Studies in Computational Intelligence 552

Yen-Wei Chen
Lakhmi C. Jain
Editors

Subspace Methods for Pattern Recognition in Intelligent Environment

Springer

Studies in Computational Intelligence

Volume 552

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Yen-Wei Chen · Lakhmi C. Jain
Editors

Subspace Methods for Pattern Recognition in Intelligent Environment

Springer

Editors

Yen-Wei Chen
College of Science and Engineering
Ritsumeikan University
Kusuatsu
Japan

Lakhmi C. Jain
Faculty of Education, Science, Technology
and Mathematics
University of Canberra
ACT 2601
Australia

ISSN 1860-949X ISSN 1860-9503 (electronic)
ISBN 978-3-642-54850-5 ISBN 978-3-642-54851-2 (eBook)
DOI 10.1007/978-3-642-54851-2
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014934176

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

The book is devoted to the contemporary methods for pattern recognition, based on the use of the lower-dimensional feature subspace. The actuality of this scientific area is indisputable and is defined by the impetuous development of the information theories and the huge amounts of visual data, which have to be processed and analyzed. The large challenge for the pattern recognition systems is the necessity of efficient processing of multidimensional signals and images, obtained from various sources, such as (for example), the EEG and ECG devices, traffic surveillance systems, systems for video control, admission systems, remote control systems, medical systems for image diagnostics, and systems for content-based retrieval in large image databases. The book comprises 8 chapters, which present various new approaches aimed at solving some important problems in the theory of the pattern recognition, related to the feature space reduction, and also many interesting applications in the pattern recognition area. Each chapter is a small monograph, which represents the research of the authors in the corresponding area.

In the *first chapter* are presented and investigated new approaches, based on the Active Shape Model (ASM) for face recognition. Here is also proposed the so-called Point Distribution Model, used for the description of a set of landmarks, which characterize the form of the recognized face. By using the landmarks alignment and the Principal Component Analysis (PCA), is developed the multi-resolution gray-level face profile, used for the image search. The presented modeling results prove the improvement achieved, when compared to the classical ASM.

The second chapter is focused on one new approach for features space reduction through the conditional Statistical Shape Model (SSM), used in the medical image analysis. In the chapter are given the advantages and the disadvantages of this model, and are discussed its basic varieties: the non-conditional (NC-SSM), the conventional conditional (C-SSM), and the relaxed conditional Statistical Shape Model (RC-SSM). Here is also given a comparison among various SSM models, used for the segmentation of computer tomography images. The obtained experimental results for the evaluation of the shape of the segmented areas show, that the models RC-SSM and RC-SSM-E (RC-SSM with integrated conditional features error model) surpass C-SSM and NC-SSM.

In the *third chapter* is offered feature extraction technique based on the method for Independent Component Analysis (ICA), aimed at the classification of High-resolution Remote Sensing Multispectral (MS) Images. The transform of the color components R, G, B of the investigated 3-channel MS images into the corresponding 3 independent components IC1, IC2, and IC3 results in the obtaining of nonoverlapping distributions in the ICA features space. The evaluation of the classification results for the objects detected in the MS images through the K-means algorithm shows higher accuracy for the ICA features space, compared to that, based on the PCA.

In the *fourth chapter* is presented the so-called Generative learning method, through which, on the basis of a small number of real images of traffic signs, obtained from TV cameras, are generated many artificial images of these signs, which have various degradations in respect to the originals. For the signs recognition is used the subspace method. The presented framework is applicable for any traffic sign by combining it with conventional traffic sign detection methods. The use of the presented results is aimed at the driver support systems.

In the *fifth chapter* are presented two novel subspace analysis methods for face recognition and image clustering tasks. One is a nonlinear subspace method obtained by using an algebraic approach, and the other is a probabilistic subspace analysis method derived from the topic model. The experiments on face recognition and image clustering show that the proposed subspace analysis methods are resistant in respect of faces variations, such as noise, pose, and lighting.

The *sixth chapter* is devoted to the problem for the restoration of the high-resolution image from a single low-resolution input image, by using the sparse signal representation, based on the combined K-SVD algorithm and the Orthogonal Matching Pursuit (OMP) for learning the adaptive dictionary and achieving the sparse coefficients. The OMP is an extended orthogonal version of matching pursuit, which is a type of numerical technique which involves finding the ”best matching” projections of multidimensional data onto an over-complete dictionary, and can be combined into the K-SVD strategy for achieving sparse representation and the best adaptive dictionary. The effectiveness of the proposed new dictionary propagation in sparse coding for super-resolution is demonstrated by comparison with the conventional super-resolution approaches such as sparse coding and bicubic interpolation.

The *seventh chapter* is a short tutorial on the sampling and recovery of continu ously-defined sparse signals and its application to image feature extraction. To sample a signal at low frequency compared with its Nyquist frequency, the signal is characterized using how frequently unknown parameters appear in its parametric expression, instead of the classical frequency approach. The new frequency of the parameter appearance is defined as the rate of innovation. In the chapter are analyzed some example signals of this kind: a sequence of Diracs and piecewise polynomials, and is also given the application of the proposed algorithm for object edge detection.

In the *eighth chapter* is developed the approach for human face representation through so-called „tensorfaces“. Here is proposed the tensor-based subspace learning method (TSL) for synthesizing human multi-pose facial images from a single two-dimensional (2D) image. The facial pose synthesis is applied to generate much required information for several applications, such as public security, facial cosmetology, etc. In the proposed TSL method, 2D multi-pose images in the database are previously organized into a tensor form and a tensor decomposition technique is applied to build the projection subspaces. The experimental results show the effectiveness of proposed method for facial pose synthesis.

Each chapter comprises a theoretical part, followed by experimental results and comparison with other similar techniques. The authors are researchers from different universities and R&D centers in Japan, China, and USA. The book will be very useful for university and PhD students, researchers and software developers, who work in the area of the digital processing, analysis and recognition of signals and images.

Prof. D. Sc. PhD. Roumen Kountchev
Technical University of Sofia
Bulgaria

Preface

With the fast development of internet and computer technologies, the amount of available data is now rapidly increasing in our daily life. How to extract core information or useful features is an important issue. Subspace methods are widely used for dimension reduction and feature extraction in pattern recognition. They transform high dimensional data to a lower dimensional space (subspace), that focuses on the relevant information only. A lot of methods have been proposed for data transformation, such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) and so on. PCA is a linear transform that projects the data into a new coordinate system (subspace) with bases where the data varies the most, while ICA finds a linear representation of non-Gaussian data so that the components are statistically independent, or as independent as possible. Kernel PCA and Kernel ICA are modifications of the original PCA and ICA, facilitating nonlinear transformations. A lot of other nonlinear subspace methods are also proposed in the literature, such as ISOMAP, Locally Linear Embedding (LLE) and so on. Recently, sparse coding and sparse sampling are hot topics in pattern recognition and signal processing. For multidimensional data such as medical volumes, multi-view facial images, tensor based subspace learning methods are also proposed, which are based on a multi-linear algebra framework.

This book focuses on major trends and new techniques in subspace methods and their applications in pattern recognition. There are eight chapters written by experts in this book. The area of interest of the chapters covers a broad spectrum of subspace learning methods with application to pattern recognition in intelligent environment.

Chapter 1 focuses on principal component analysis (PCA) and its application to construct an active shape model of facial images. The variations of facial shape and texture can be represented by a few leading edge modes. Through this model, image interpretation can be formulated as a fitting problem.

Chapter 2 introduces a conditional statistical shape model, which is a valuable subspace method in medical image analysis. During training of the model, the relationship between the shape of the object of interest and a set of conditional features is established. Subsequently, while analyzing an unseen image, a measured condition is matched with this conditional distribution and then a subspace of the training data is marked as relevant and used for the desired reconstruction of the object shape.

Chapter 3 focuses on independent component analysis (ICA) and its application to classification of high-resolution remote sensing images. ICA tries to find a linear representation of non-Gaussian data so that the components are statistically independent, or as independent as possible. The three independent components are in opponent-color model by which the responses of R, G and B cones are combined in opponent fashions. This is consistent with the principle of many color systems.

Chapter 4 presents a training method for subspace construction from artificially generated images for traffic sign recognition, which is one of the important tasks for driver support systems. Conventional approaches used camera-captured images as training data, which required exhaustive collection of captured samples. The generative learning method, instead, allows to obtain these training images based on a small set of actual images.

Chapter 5 presents local structure preserving methods based on subspace analysis. Two novel subspace methods are proposed for face recognition and image clustering tasks. The first is named Supervised Kernel Locality Preserving Projections (SKLPP) for face recognition tasks, in which geometric relations are preserved according to prior class-label information and complex nonlinear variations of real face images are represented by nonlinear kernel mapping. The second is a novel probabilistic topic model for image clustering task, named Dual Local Consistency Probabilistic Latent Semantic Analysis (DLC-PLSA). The proposed DLC-PLSA model can learn an effective and robust mid-level representation in the latent semantic space for image analysis.

Chapter 6 introduces the sparse signal representation, and a popular implementation: KSVD algorithm combining orthogonal matching pursuit (OMP) for learning the adaptive dictionary and achieving the sparse coefficients. The sparse representation is used for learning-based image super-resolution for recovering the high-resolution image from only single low-resolution one. Based on the couple dictionary learning for super-resolution, we proposed a HR2LR (high-resolution to low-resolution) dictionary propagation algorithm in sparse coding for image super-resolution.

Chapter 7 presents a technique for efficient sampling and recovery of continuously-defined sparse signals that is known as sparse sampling. To sample a signal at low frequency compared with its Nyquist frequency, the signal was characterized using how frequently the unknown parameters appear in its parametric expression, instead of the classical frequency. The new frequency of the parameter appearance was defined as the rate of innovation. The chapter focused on two typical examples of signals with finite rate of innovation: the sequence of Diracs and piecewise polynomials.

Chapter 8 presents a tensor-based subspace learning method (TSL) for synthesizing human multi-pose facial images from a single two-dimensional image. In the proposed TSL method, two-dimensional multi-pose images in the database are previously organized into a tensor form and a tensor decomposition technique is applied to build projection subspaces. In synthesis processing, the input two-dimensional image is first projected into its corresponding projection subspace to get an identity vector and then the identity vector is used to generate other novel pose images.

Although the above chapters do not make a complete coverage of the subspace learning methods for pattern recognition, it provides a flavor of the important issues and the benefits of applying subspace learning methods to pattern recognition.

We are grateful to the authors and reviewers for their contribution. We would like to thank the editors of the *Springer* for hosting this book and for their advice during the editorial process of the book.

January 2014

Yen-Wei Chen, Ritsumeikan University, Japan
Lakhmi Jain, University of Canberra, Australia

Contents

Chapter 1
Active Shape Model and Its Application to Face Alignment

Huchuan Lu and Fan Yang

Abstract. Active Shape Model (ASM) is a model-based method, which makes use of a prior model of what is expected in the image, and typically attempts to find the best match position between the model and the data in a new image. It has been successfully applied to many problems and we apply ASM to the face recognition. We represent all shapes with a set of landmarks to form a Point Distribution Model (PDM) respectively. After landmarks alignment and Principal Component Analysis, we construct gray-level profile for each landmark in all multi-resolution versions of a training image. In search procedure, we give the model's position an initial estimate. We adopt a lot of improvements to the classical ASM, such as increasing the width of search profile to reduce the effect of noise, grouping landmarks to avoid mouth shape distort in the search procedure and altering the direction of search profile.

1 Introduction

The ultimate aim of machine vision is to make machines understand and respond to what they see, in typical such as applications in medical image interpretation, face recognition, and many other aspects. Practical applications need to be typically characterized by the ability to deal with complex and variable structures and images which contain noise and possibly incomplete evidence. Most extraordinary applications also include the challenges in image structure recovery and interpretation by automated systems. Therefore, it is necessary to develop models which can describe and label the expected structure of the image.

Huchuan Lu · Fan Yang
School of Electronic and Information Engineering, Dalian University of Technology, China

Fan Yang
Department of Computer Science, University of Maryland, College Park, USA

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment,* Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_1, © Springer-Verlag Berlin Heidelberg 2014

Model-based methods offer effective solutions to the difficulties above. First of all, a prior knowledge of the problem is used to resolve the difficulties caused by structural complexity. Then we apply knowledge of the expected shapes of structures, their spatial relationships, and their grey-level appearance to interpret the desired images.

To achieve this purpose, our task focuses on generating sufficiently complete models that produce authentic images of target objects. For instance, we need a face model capable of generating conformed images of any individual with changes in expressions or postures. Through this model, image interpretation can be formulated as a fitting problem: given a new image, the target object can be located by adjusting several parameters, which deform the model into a plausible object that closely approximates the object of interest in the unseen image.

In actual applications, we usually need to deal with objects that possess a large variability in shape and appearance. This leads to the idea of deformable models, which maintain the essential characteristics of the class of objects and can be deformed to fit a range of examples. Such models have to possess two main characteristics.

First, they should be general, which means that they should be capable of generating any plausible example of the class they represent. Second, and crucially, they should be target-oriented. They are only allowed to generate suitable shape. We obtain specific models of variable objects by knowledge of the way they vary.

Model-based methods make use of a prior model with what is expected in the image, and typically attempt to find the best match position between the model and the data in a new image. One can measure whether the target is actually present after matching the model.

This approach is a "top-down" strategy, and differs significantly from the "bottom-up" or "data-driven" methods. In the latter they examine the image data at a low level, looking for local structures such as edges or regions, which are assembled into groups in an attempt to identify objects of interest. Without a global model of the object of interest, this approach is difficult to realize and inclined to failure. Thus, a wide variety of model-based approaches have been proposed. For instance, a statistical approach is explored, in which a model is built by analyzing the statistical characteristics of a set of manually annotated images. It is possible to distinguish plausible variations from those that are not. One can interpret a new image by finding the best matching position of the model to the image data.

The advantages of such a method are those [1]:

1. It is widely applicable. The same algorithm can be applied to many different problems, only to collect different training images.
2. Prior knowledge can be captured in the annotation procedure of the training images.
3. The model not only gives a compact representation of allowable variation, but also is specific enough to avoid unacceptable shapes generated.
4. The algorithm need make few prior assumptions about the objects being modeled, other than what it learns from the training set. For instance, there are no boundary smoothness parameters to be set.

Human faces can vary widely in size, shape and appearance due to changes in expression, perspective, and illumination. It makes it difficult to automatically identify and segment structures we are interested in. Thus, to interpret images of faces, a suitable model of appearance of faces is necessary.

Then a model [2] will be introduced next, which requires a user to be able to label points called landmarks on each image of a training set. For instance, when building a model of the appearance of an eye in a face image, good landmarks would be the corners of the eye, as these would be easy to identify and mark in each image. Unfortunately, some things such as cells or simple organisms which exhibit large changes in shape are so amorphous that the model cannot be applied well, which is the limitations of this model.

2 Statistical Shape Models

The model described above is a statistical model, which is used to represent objects in images. We adopt a statistical approach. The face shape represented by annotating a set of feature points to define correspondence across the set varies across a range of images. Commonly the points are in two or three dimensions. Shape is usually defined as the position of points which is invariant under some transformation. In two or three dimensions, alignment of training shapes (translation, rotation and scaling) is usually considered.

Our aim is to derive models which not only allow us to analyze new shapes, but also to synthesize shapes similar to those in a training set, which typically comes from manual annotation of a set of training images, though automatic or semi-automatic landmarking systems are being developed. By analyzing the variations in shape over the training set, a model is built to generalize this variation. The patterns of intensities are then analyzed to learn the ways in which the texture can vary. The final model is one capable of capturing and generalizing the statistical characteristics of training images, but it is also specific enough to generate face-like shape.

How to build a shape model under a similarity transformation in an arbitrary d-dimensional space will be discussed below. Most illustrations will be given for two dimensional shapes under the similarity transformation with parameters of scaling, rotation and translation.

However, it is not necessary that the dimensions are always in space. They can also be time or intensity in an image [1]. For instance

1. 3D Shapes can either be composed of points in 3D space, or points in 2D with a time dimension.
2. 2D Shapes can either be composed of points in 2D space, or one space and one time dimension.
3. 1D Shapes can either be composed of points along a line, or intensity values sampled at particular positions in an image.

There are numerous other possibilities. In each case a corresponding transformation must be defined.

2.1 Point Distribution Model

As mentioned above, our aim is to build a model describing shapes and variations of an object. To make the model capable of capturing typical shape and typical variability, we collect a large amount of images of the object, which should cover all the types of variation we wish the model to represent. For instance, if we are only interested in frontal faces, we should include only frontal faces in the model. If, however, we want to model faces with different perspectives, the images we collect should contain people faces with a wide range of pose angles. Figure 1 shows some images of the training set. After choosing enough interesting images, we get a set of images, and name it training set.

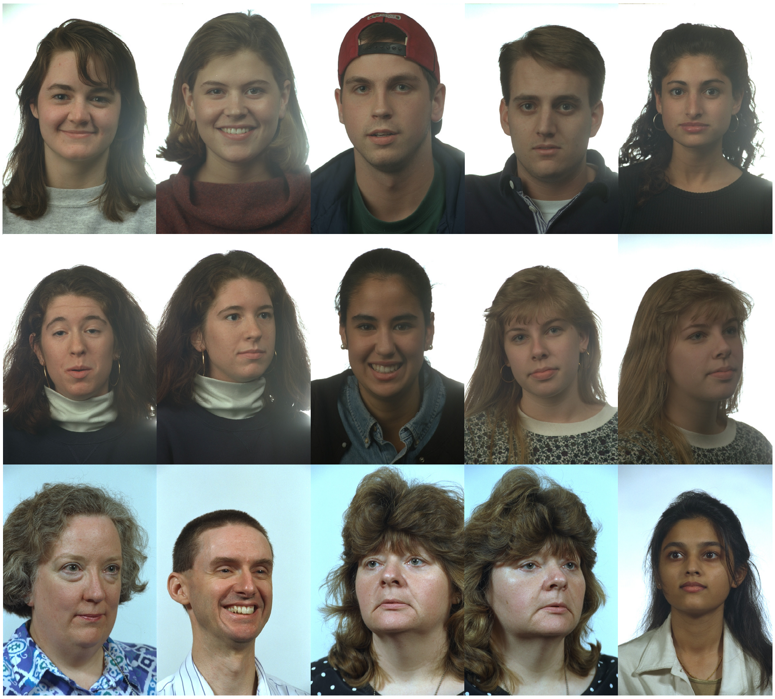
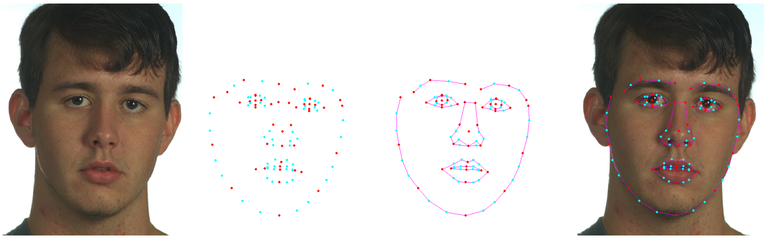


Fig. 1 Some images from the training set

Labeling the Training Set

To model a shape, we should represent it with a set of landmark points or landmarks. Good choices for landmarks are points which can be consistently located from one image to another. Each object must be annotated with landmarks defining the key facial features. Each landmark represents a particular part of the object or its boundary, and thus has a certain distribution in the image space. The procedure is called labeling the training set. It is important but in practice it may be



Input face Landmarks Landmarks(connected) Labeled face

Fig. 2 Example of 103 landmarks defining facial features

very time consuming, and semi-automatic or automatic method is being developed to aid the annotation. After this, we get a model called Point Distribution Model (PDM). Figure 2 shows a set of 103 landmarks used to labeling frontal faces.

Some examples of shape obtained from the training set are illustrated in Figure 3.

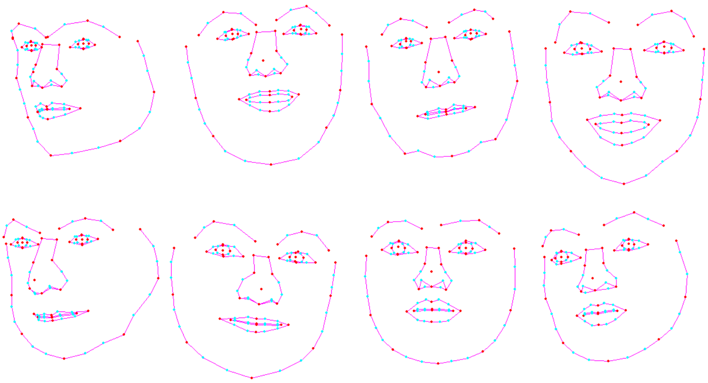


Fig. 3 PDM samples from the training images

However, a significant issue should be paid attention before starting to place points on the images. We should decide on the number of landmark points that adequately represent the shape. The number of the landmarks depends on the complexity of shapes and the desired detail level. Adequate landmarks can show the overall shape and details that we need. The same number of landmarks should be placed on each image of the whole training set. Exact correspondence in the sequence of labeling a shape in one image and in another is also important. The location of the landmark point should be located as accurately as possible, since these locations will govern the resulting point variations and the intended PDM. If the labeling is incorrect, with a particular landmarks placed at different sites, the algorithm will result in failure in capturing the shape variability reliably.

It is important to choose the suitable locality to make landmark points represent the key features. Generally, there are three types of landmarks [3] as follow:

1. Application-dependent landmarks, such as the center of an eye.
2. Application-independent landmarks, such as the highest point of an object in a certain orientation.
3. Landmarks interpolated from the above two type.

The decisions on the number of landmarks and choices in the locality of them are both significant, and they could have a great impact on building model later. Supposing we have labeled each of the *N* images in the training set with *n* landmarks. Now we get a landmark set to represent the shape. For the i^{th} image, we denote the j^{th} landmark coordinate point by (x_{ij}, y_{ij}). And the *2n* element vector describing the *n* point of the i^{th} image can be written as

$$X_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, \ldots, x_{in-1}, y_{in-1}]^T \qquad (1)$$

where $1 \le i \le N$. As a result, we generate *N* such representative vectors from *N* training images. Before carrying out statistical analysis on those vectors, we should ensure that the shapes represented are in the same coordinate frame.

Aligning the Training Set

As mentioned above, in order to study the statistical characteristics of the coordinates of the landmark points, it is important that the shapes represented should be in a common coordinate frame. To achieve this, all the shapes must be aligned to each other to remove variation that could affect the statistical analysis result. The aligning can be done by scaling, rotating and translating the shapes of the training set to meet the requirement. The aim of alignment is to minimize a weighted sum of squares of distances between equivalent landmarks on different images. In other words, we wish to make points as close to the corresponding ones as possible.

Procrustes analysis is the most popular method to align shapes into the same coordinate. The spirit of the algorithm is a weighted least-squares approach. By using this, the pose parameters needed to align one vector to another can be obtained. Given two vectors X_i and X_k, we need to align X_i and X_k, so we need to find the scaling value s, the rotation angle θ, and the value of translation in both dimensions (t_x, t_y).

A weighted diagonal matrix *W* is used to give points that are more stable over the set more significance. Stable point is defined as having less movements respect to other points in a shape.

To calculate such weights we first calculate the distances between every pair of points in all the shapes. Then we calculate the variance of the distance between every pair of points over all the shapes. Then for a specific point, the sum of the variances of the distances from this point to all others, would measure the instability of the point, we thus take the weight to be the inverse of this summation. Mathematically, let R_{ikl} be the distance between the landmark points *k* and *l* in the i^{th} shape. By denoting the variance of the distance between the landmark points *k* and *l* by $V_{R_{kl}}$, we get the weight for the k^{th} landmark

$$w_k = \left(\sum_{l=0}^{n-1} V_{R_{kl}} \right)^{-1} \tag{2}$$

where $0 \le k \le n-1$ and *n* is the number of landmark points. And the weighting matrix will then be the following diagonal matrix.

$$W = diag(w_1, w_1, w_2, w_2, ..., w_{n-1}, w_{n-1}) \tag{3}$$

The following algorithm is an easy handling iterative approach to align the set of *N* shapes to each other [3].

1. Align each shape to one of the shapes, for instance, the first one or the mean shape.
2. Calculate the mean shape from the aligned shape.
3. Normalize the pose of the current mean shape.
4. Realign the each shape with the normalized mean.
5. If converged, stop the process, or else go to step 2 and repeat.

When the shapes are not changing more than a pre-defined threshold after iteration, we can claim that the convergence is reached.

The meaning of the normalization [4] is

1. Scale the shape so that the distance between two points becomes a certain constant.
2. Rotate the shape so that the line joining two pre-specified landmarks is directed in a certain direction.
3. Translate the shape so that it becomes centered at a certain coordinate.

Normalization is carried out in order to make sure the algorithm converges. Without this the mean shape may translate to expand or shrink indefinitely. Different approaches to alignment can produce different distributions of the aligned shapes, so we can choose other methods to satisfy various requirements.

Figure 4 shows shapes before alignment and after alignment. We pick up three of the alignment results and carry out a comparison. In the figure, the green line and the blue line respectively represent shapes before and after alignment. We can see that the position of face shapes are more concentrated, so all the shapes are closer to each other.

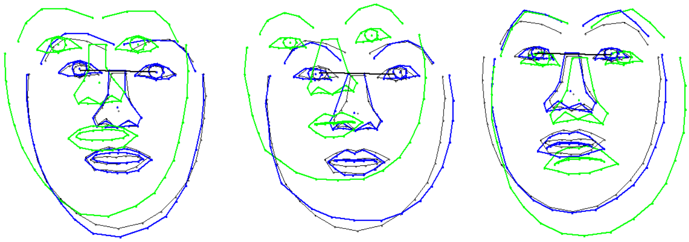


Fig. 4 Examples of face shapes before and after alignment

Statistical Model of Variation

After alignment, *N* sets of vector X_i representing the images of training set now contain the new coordinate. These vectors form a distribution from which we can generate new examples similar to the original ones. Also, we can determine clearly whether a new shape produced by the model during search procedure is allowable or acceptable.

We aim to build a parameterized model, which can generate new shapes with the formula $X = M(b)$, where *b* is a vector of parameters of the model. If we can find the distribution of *b*, we can control to produce new shapes similar to those in the training set.

Each vector is in a *2n*-dimensional space. The *N* vectors representing the *N* aligned shapes will then map to a “cloud” of *N* landmarks in the same *2n*-dimensional space. Also, these *N* landmarks are contained within a region of this *2n*-dimentional space. This region is called the Allowable Shape Domain (ASD) [3], where every landmark in this region gives a shape that is similar to the other ones. We can use the Euclidean distance between two landmarks representing two shapes in the *2n*-dimentional as a measure of similarity.

However, a problem still exists. If the vector consists of too many elements, the computation complexity could increase significantly. It can be unacceptable in some particular situation. On the other hand, the majority of variations are determined by a few elements of the vector X . To deal with this, we can reduce the dimensionality of the data from *2n-D* to a lower space. An effective approach is applying Principal Component Analysis (PCA) to the data. By doing this, we generate a new set called principal component. Each of that is a linear combination of the original variables. All the principal components are orthogonal to each other. The whole principal components form an orthogonal basis for the space of data. PCA allows us to use a model with fewer than *2n* parameters instead of original data. So it reduces the computation complexity and more practical. The following is the detailed procedure.

1. $\overline{X}$ represents the mean value of vector X_i, and the derivation of each shape from the mean can be denoted by dX_i. Then we have

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{4}$$

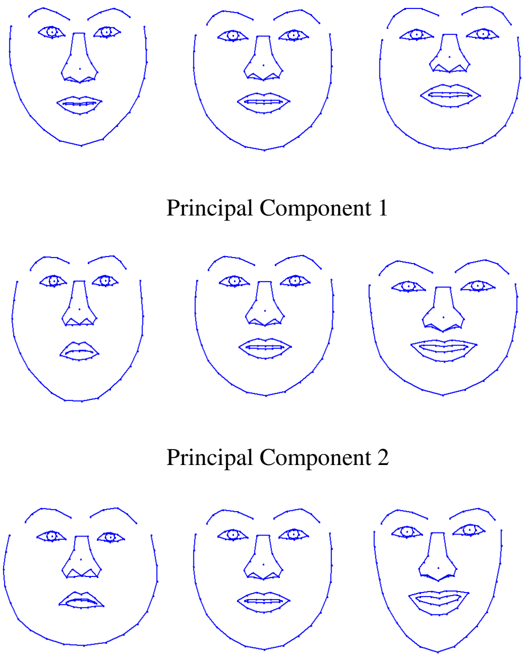
and

$$dX_i = X_i - \overline{X} \tag{5}$$

2. Calculate the $2n \times 2n$ covariance matrix of the data, and write it as

$$S = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(X_i - \overline{X})^T \tag{6}$$

3. Calculate the eigenvectors and corresponding eigenvalues of S.



Principal Component 1

Principal Component 2

Principal Component 3

Fig. 5 The effect of the first three principal components (negative to positive)

The largest eigenvalue corresponding to the eigenvector derived from the covariance matrix describes the most significant shape variation within the training data set. The variance explained by each eigenvector is equal to the corresponding eigenvalue. And from the covariance matrix, we can see that the eigenvalues are in descending order, namely $\lambda_i \geq \lambda_{i+1}$, thus the first eigenvalue is the largest one. Actually, most variation can be explained by a small number of the eigenvalue, i.e., t ($t < 2n$). It means that the $2n$-D space can be approximated by a t-D space. That is dimension reduction.

Figure 5 illustrates the effect of the first three principal components on the face shape changing. We can see that those principal components determine the most variations of face shapes. Thus, it is reasonable that only a few principal components are used for shape modeling.

To obtain the suitable number of the eigenvalues, *t*, there is a simple approach for most of the practical applications. That is accumulating all the eigenvalues from the first one. When the sum of the first *t* eigenvalues explains a sufficient proportion of the total variance of the original data, the number *t* is found. Expressed mathematically

$$\sum_{i=1}^{t} \lambda_i \geq f_v V_T \tag{7}$$

where V_T is the total variance, $V_T = \sum \lambda_i$. f defines the proportion of the total variance that we need. It usually ranges from 90% to 99%.

Then we can approximate an example in the training set using the mean shape plus a weighted sum of the first *t* principal component like this

$$X \approx \overline{X} + Pb \tag{8}$$

where $P = (p_1, p_2, ...p_t)$ is the first t eigenvector matrix, and $b = (b_1, b_2, ...b_t)$ is the weights vector. Transform the equation into the following form

$$b = P^{-1}\left(X_i - \overline{X}\right) \tag{9}$$

where b is derived from the eigenvalues. By changing the elements of b, we can vary the shape.

But arbitrary b could result in an unallowable shape. To avoid the problem, we can impose constrains on b . By giving limits, we can control the changing of shape in order to generate plausible shapes. Since the variance of *b* of the training set has a relationship with the eigenvalue λ . The typical limits are

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \tag{10}$$

where λ_i is the eigenvalue corresponding to the i^{th} eigenvector. A series of experiments has proved that the Gaussian assumption is a good approximation to the face shape distribution, provided the training set contains only modest viewpoint variation, as large viewpoint variation may introduce nonlinear changes into the shape so that the model can't be applied well.

2.2 Modeling Local Structure

It is not enough to obtain only the statistical characteristics of the position of landmarks. In order to find desired movement and make a good estimation of model position during the image search and classification procedure, besides shape information, a model containing gray-level information of the images in the training set [1, 7, 8] should also be established. The core idea of the gray-level information modeling method is to collect pixels around each landmark and try to put the pixels' gray information in a compact form so we can use it for image search. Generally, the region around the landmark can be considered, but for simplicity, we only use the points along the line passing through the landmark and perpendicular to the line connecting the landmark and its neighbors, as Figure 6 shows.

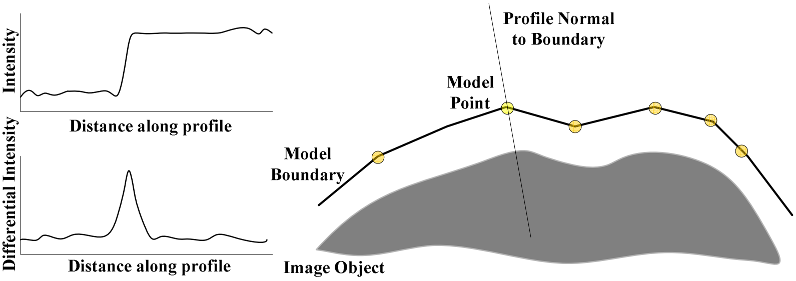


Fig. 6 Sample along the line normal to the boundary

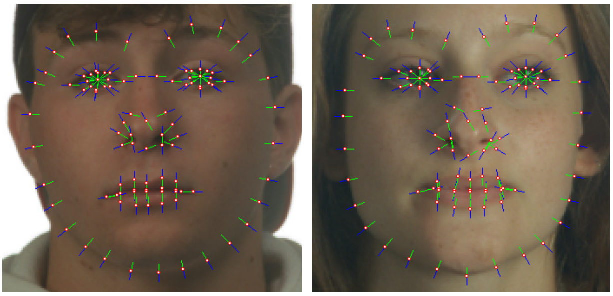


Fig. 7 Samples of gray-level profile

For each landmark, we can sample *k* pixels on either side of the landmark along a profile (as Figure 7 shows). Then we obtain a gray-level profile of *2k+1* (include the landmark itself) length. We describe it by a vector *g*. To reduce the effect of global intensity changes, we do not use the actual vector *g* but use the normalized derivative instead. It reflects the change of gray-level along the profile.

The gray-level profile of the j^{th} landmark in the i^{th} image is a vector of *2k+1* element

$$g_{ij} = \left[g_{ij0}, g_{ij1}, \ldots, g_{ij2k}, g_{ij(2k+1)} \right]^T \tag{11}$$

And its differential form is of *2k* length

$$dg_{ij} = \left[g_{ij1} - g_{ij0}, g_{ij2} - g_{ij1}, \ldots, g_{ij(2k+1)} - g_{ij2k} \right]^T \tag{12}$$

The normalized form is

$$y_{ij} = \frac{dg_{ij}}{\sum_{m=0}^{2k-1} \left| dg_{ijm} \right|} \tag{13}$$

Then the mean of the normalized derivative profiles of each landmark in the whole training set can be calculated, and for the j^{th} landmark

$$\overline{y_j} = \frac{1}{N} \sum_{i=1}^{N} y_{ij} \tag{14}$$

The covariance matrix of the normalized derivative is given by

$$C_{y_j} = \frac{1}{N} \sum_{i=1}^{N} (y_{ij} - \overline{y_j})(y_{ij} - \overline{y_j})^T \tag{15}$$

The process above is carried out until all landmarks in the training images have their own normalized derivative profiles. We assume they have multivariate Gaussian distribution and calculate their mean $\overline{y_j}$ and covariance matrix C_{y_j} .

Until now, we have built statistical models of the gray-level profile for all the landmarks in the training images. In the image searching step, we will use the profile for better search.

2.3 Multi-resolution Active Shape Model

There exists a problem that how can we optimize the accuracy and the complexity of the algorithm. On one hand, we wish to choose as many pixels as possible to obtain enough gray information in order to fit the model well onto the new image. On the other hand, the length of the profile should not be too long; otherwise it may result in significant increasing of the computation complexity, which cannot be tolerant in practical applications. And if the search profile is long but the target point is close to the current position of the landmark then it will be more probable to move to a far away point and miss the target.

Based on the analysis above, the algorithm is implemented in a multi-resolution approach [9], which is involved in searching firstly in a coarse image for remote points with large jumps and refining the location in a series of finer resolution images by limiting the jump to only close points.

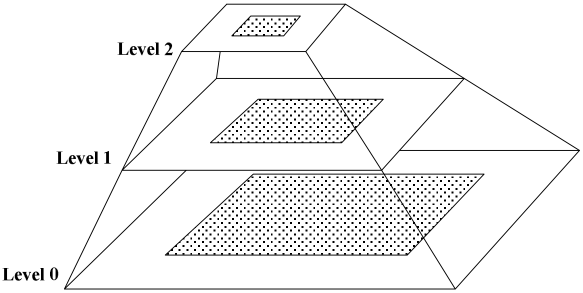


Fig. 8 Image pyramid

The structure of multi-resolution images is like a pyramid, so we call it image pyramid. At the base of the pyramid it is the original image and the level is the lowest (level 0). The image in the higher level is formed by subsampling the former image then we obtain a lower resolution version of the image with half number of the pixels along each dimension. Subsequent levels are obtained by further subsampling (as Figure 8 shows) until each training image has its own pyramid.

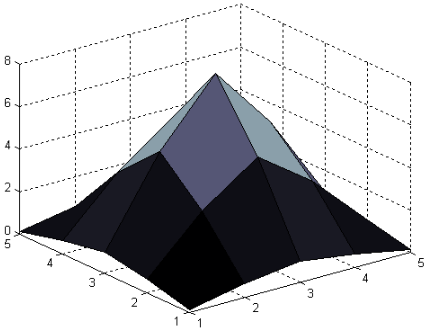


Fig. 9 Gaussian filter

The procedure of getting image pyramid is as follows [9, 10].

1. Smooth the original image with a Gaussian filter, which is linearly decomposed into two 1-5-8-5-1 convolutions as Figure 9 shows. The reason of using Gaussian filter is that jagged edge will be produced in the sub-image if we directly sample the original image without a filter, and it will go against the gray-level modeling of landmarks.

2. Sub-sample the image every other pixel in each dimension. Then we get a new image of level 1, which is 1/4 of the original image.

3. From level 1, repeat step 1 and step 2 to obtain the higher level of the pyramid.

4. Until we have got the highest level pre-defined for all the training images, terminate the process.

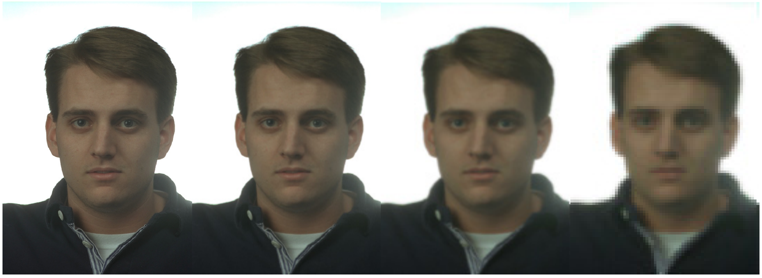


Fig. 10 An example of multi-resolution image

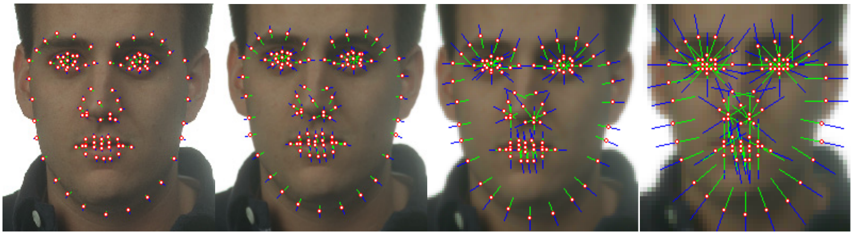


Fig. 11 Multi-resolution gray-level profile

The final result is shown in Figure 10. In this procedure we generate a 4-level pyramid. Level 0 is the original image, while higher level is the coarser image with low resolution. Such images with different resolution versions compose an image pyramid.

During the training, we build statistical models of gray-level along profile through each landmark, at each level of the image pyramid. We usually use the same length in the gray profile, regardless of level. Since the pixels at level L are $1/2^L$ times the size of those of the original image. During search we need only to search n pixels either side of the current landmark in each level. Thus, at coarse level this will allow large movement and convergence can established quickly, whereas at finer resolution the amounts of the movement could be small. We can see that from Figure 11.

During the search, we start our searching from the top level of the pyramid. In each image, the initial search position of the model is the search output of its upper level, until the lowest level is reached. We mainly consider when to stop searching at a level and move to the next level. We record the number of times that the best found pixel along a search profile is within the central 50% of the profile [7]. A convergence criterion is that a sufficient number of landmarks are reached. A constraint on iteration number can also be added to prevent the search getting stuck on one level.

By applying the multi-resolution method on training images, we have improved the efficiency and robustness of the previous algorithm.

3 Image Search Using Active Shape Model

Till now, we have established a model with multi-resolution training images and gray-level information around the landmarks in each level of the image pyramid. Now, based on the model, we will locate a new example of the object in an image. The idea is: first giving the model an initial position through some prior knowledge; second, we examine landmarks and their neighbors along gray-level profile to find better location of landmarks; third, we update the pose and shape parameters with suitable constraints and move the model to the new location and produce a plausible shape. The fact that the shapes are modeled so that they can only vary in a controllable way by constraining the weights of the principal components explains why such model is named Active Shape model or ASM.

We assume that an instance of an object is described as the mean shape obtained from the training set plus a weighted sum of the t principal components, with the possibility of this sum being scaled, rotated and translated.

3.1 Initial Estimate

For an unknown image, if we wish to find out a specific matching object, the first step is to give the model an initial position, which is obtained by prior knowledge and should not be too far away from the target object, as initial estimate of ASM. We can express the estimate X_i of the shape as a scaled, rotated and translated version of a shape X_l.

$$X_i = M(s_i, \theta_i)[X_l] + t_i \tag{16}$$

Where s_i, θ_i and t_i is respectively scale, rotation and translation parameters. $t_i = \left[t_{xi}, t_{yi}, t_{xi}, t_{yi}, \ldots, t_{xi}, t_{yi}\right]^T$ with a length of *2n*. X_l can also be expressed as $X_l = \bar{X} + dX_l$, with $dX_l = Pb_l$. X_l is mean shape of the model.

$$X_i = M(s_i, \theta_i)\left[\bar{X} + Pb_l\right] + t_i \qquad (17)$$

3.2 Compute the Movements of Landmarks

Putting X_i onto the image which we will search for the object, we can examine the gray-level information of the landmarks along the normal of the shape boundary for the strongest image edge, whose magnitude proportional to the strength of the edge. So we can find a new position of landmarks to make the model closer to the target object.

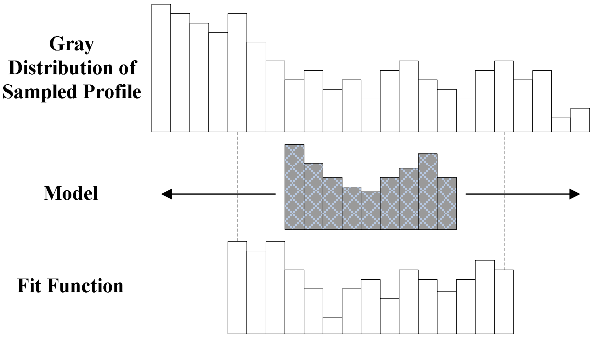


Fig. 12 Search along the profile to find the best fit

Denote the search profile of landmark *j* by S_j, which is a vector generated by sampling *m* pixels either side of the current point, so its length is *2m+1*. Obviously, we should make sure that *m>k* so the search profile can cover the model profile completely. For instance, *m* is 15 and *k* is 5. Put symbolically

$$S_j = \left[s_{j0}, s_{j1}, \ldots, s_{j2m}, s_{j(2m+1)}\right]^T \qquad (18)$$

And the differential form of S_j is

$$dS_j = \left[s_{j1} - s_{j0}, s_{j2} - s_{j1}, \ldots, s_{j(2m+1)} - s_{j2m}\right]^T \qquad (19)$$

The normalized form of dS_j is

$$y_{si} = \frac{dS}{\sum_{k=0}^{2m-1} \left| dS_{jk} \right|} \tag{20}$$

The degree of similarity of the target object, y_{s_i}, to the model $\overline{y_j}$ is given by

$$f(d) = (h(d) - \overline{y_j})^T (h(d) - \overline{y_j}) \tag{21}$$

where $h(d)$ is the sub-profile centered at the d^{th} pixel of y_{si}. $\overline{y_j}$ is mean value of gray-level information. Given a *d*, if $h(d)$ is the most similar to $\overline{y_j}$, then the d^{th} pixel is where the position that landmark *j* should be moved toward. $f(d)$ is linearly related to the log of the probability to which $h(d)$ observes, and $h(d)$ shares the same probability distribution with $\overline{y_j}$. Minimizing $f(d)$ is equivalent to maximizing the probability of $h(d)$.

Using the algorithm above, the best fit position for the landmark *j* can be found. For the shape X_i, this process should be repeated to find a suggested new position for each landmark, so we get a position offset dX_i. Thus, we finally obtain an "ideal" shape $X_i^{'} = X_i + dX_i$.

Generally, $X_i^{'}$ obtained through gray-level profile search is closer to the shape of the target object. Then we can adjust the pose parameters, namely scale, rotation and translation, to move the initial estimate as close as possible to the target object.

But usually we do not update the X_i to $X_i^{'}$ directly, because $X_i^{'}$ maybe doesn't satisfy the shape constraints and generate an unallowable shape.

With this problem into consideration, the shape parameter *b* should also be updated to make X_i as close as possible to $X_i^{'}$. Meanwhile, constraints should be imposed on *b* to ensure that $X_i^{'}$ is a plausible shape.

$$X_i = M(s_i, \theta_i)\left[\overline{X} + Pb_l\right] + t_i \xrightarrow{(1+ds),d\theta,dt} X_i^{'}$$

or

$$X_i = M(s_i, \theta_i)\left[\overline{X} + Pb_l\right] + t_i \xrightarrow{(1+ds),d\theta,dt} X_i + dX_i$$

We can get the additional scale $1 + ds$, rotation $d\theta$ and translation (dt_x, dt_y). But they just belong to the similarity transformation which won't change the shape. There will remain residual adjustments which can only be satisfied by deforming the shape X_l.

Compute db_l by solving the following equation

$$M(s_i(1+ds),\theta_i+d\theta)\left[\bar{X}+P(b_l+db_l)\right]+t_i+dt=X_i+dX_i \tag{22}$$

Let $X_i=M(s_i,\theta_i)\left[\bar{X}+Pb_l\right]+t_i$, we get

$$M(s_i(1+ds),\theta_i+d\theta)\left[\bar{X}+P(b_l+db_l)\right]=M(s_i,\theta_i)\left[\bar{X}+Pb_l\right]-(t_i+dt) \tag{23}$$

and since

$$M^{-1}(s,\theta)[\ldots]=M(s^{-1},-\theta)[\ldots] \tag{24}$$

We obtain

$$db_l=P^TM((s_i(1+ds))^{-1},-(\theta_i+d\theta))\left[M(s_i,\theta_i)\left[\bar{X}+Pb_l\right]+dX-dt\right] \tag{25}$$

where db_l is the parameter controlling the shape change. It determines to which degree that X_i is close to $X_i^{'}$ with parameters s,θ,t .

Now, we have enough information to form a new shape estimate using the parameters above.

$$X_i^{(1)}=M(s_i(1+ds),\theta_i+d\theta)\left[X_l+Pdb^{'}\right]+t_i+dt \tag{26}$$

Then we start this procedure from $X_i^{(1)}$ to produce $X_i^{(2)}$, until there is no significant change of the shape. Parameters then can be updated, and we can also add weights to them

$$t_{xi}\rightarrow t_{xi}+w_tdt_x$$

$$t_{yi}\rightarrow t_{yi}+w_tdt_y$$

$$s_i\rightarrow s_i(1+w_sds)$$

$$\theta_i\rightarrow\theta_i+w_\theta d\theta$$

$$b_l\rightarrow b_l+W_bdb^{'}$$

where w_t , w_s , w_θ are scalar, and W_b is a diagonal of weights. These weights are used to speed up the convergence and give the stable points more importance. It is important that constraints should always be imposed on *b* during the search in order to produce an allowable shape.

The procedure of searching for a target object in a new image can be summarized as follows

1. Let $X_i = M(s_i, \theta_i)[X_l] + t_i$ be the initial estimate of the model in the new image, where $X_l = \bar{X} + Pb_l$, and P contains the first t principal component.

2. Use the gray-level profile of landmarks to search for the suggested movements of landmarks. By doing this, we can get a position offset dX_i for each landmarks of the model. Then move the initial shape to a new plausible position $X_i + dX_i$.

3. Calculate the additional pose parameters s, θ, t.

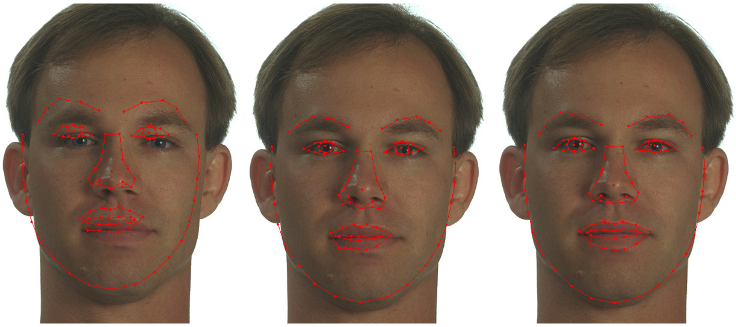
4. Calculate the additional shape parameter db_l and notice that a suitable constraint should imposed on b to avoid unallowable shapes appear.

5. Update the pose and shape parameters on X_i to obtain a new shape $X_i^{(1)}$ close to the target. Then use $X_i^{(1)}$ as the estimate, repeat from step 2.

6. Stop the iteration until no significant change is found.

3.3 *Example of Search*

This section will illustrate an example of image search using ASM algorithm. Given an image, we place the model near the human face. A coarse-fine search is carried out using the image pyramid. As we already have a 4-level pyramid, the search starts from level 4 of the pyramid, obtaining the approximate position of the model with large movements. When the pre-defined iteration number reached, the search moves to the lower level, aiming to find more subtle adjustment, until it reaches to level 0. The final convergence gives a good result. The procedure is shown in Figure 13.



Initial In search Converged

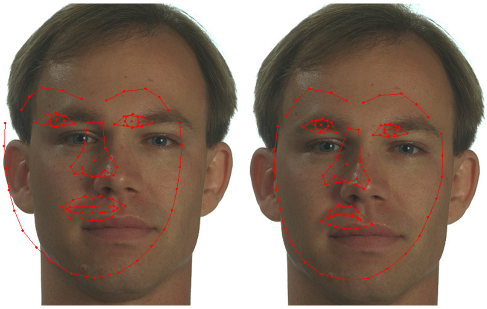
Fig. 13 Example of image search

3.4 *Application and Problems*

The basic idea of Active Shape Model is collecting the interesting images to form a training set, labeling landmarks, and obtaining the shape and gray information. For efficient and fast search, an image pyramid should also be generated. In the training procedure, we represent the shape with weighted principal components, using PCA algorithm to reduce dimension of landmark set. In addition, we obtain a series of images with different resolution versions in the training set.

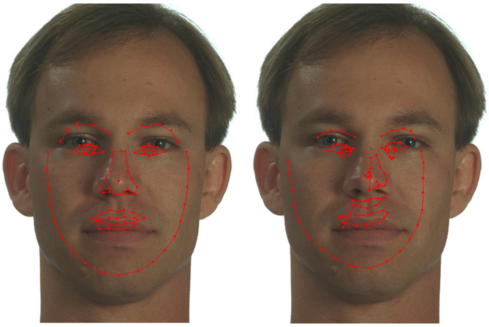
When given a new image, we can use the model to search for the desired object. During the search, we use gray-level information to obtain the suggested movements of the model, also impose constraints on shape parameter *b* to generate allowable shapes and fit the model to the target object well.

ASM is a geometry statistical model [13]. It analyzes a lot of shape information to obtain the corresponding mathematical model by statistical method. The model can cover contour subspace and texture subspace of training images, also has a strong discriminative power on the non-training objects. These characteristics make ASM has generality and specificity. Generality means that the model can cover various conditions. As to human face modeling, it means that the face model must consist of plenty of information on different people, different expressions and different poses. While pertinence means that in specific conditions, the model should only take the current object into consideration, and can distinguish illegal information from the current object. The reason that ASM outperforms other deformable model is that it only produces reasonable shape as the final segmentation result.



Initial Converged

Fig. 14 ASM failure (Model is too far away from the target.)



Initial Converged

Fig. 15 ASM failure (Model is too small.)

However, the classical ASM is a local search technique [8]. It only uses relatively sparse local information around the landmarks. And it assumes that the information at each landmark is independence, which is not often the case. It usually suffers from the local minima problem during the search, which should be solved by advanced improvements. Figure 14 and Figure 15 demonstrates the ASM failing. In Figure 14, since the initial position of the model is too far away from the target object, and in Figure 15, the model is too small for the target face, ASM search has failed to locate the correct position of the target face.

4 Improvements on Classical Active Shape Model

Owing to the existing disadvantages of classical ASM, much related work has been done to improve it, which has achieved remarkable results.

4.1 Constraint on b

As mentioned above, the shape parameter b should be constrained in a suitable range to ensure an acceptable shape to be generated. Typically, the range of b is between $-3\sqrt{\lambda_i}$ and $3\sqrt{\lambda_i}$, where λ_i is the eigen value of the covariance matrix corresponding to the i^{th} principal component. In practical applications, it may be not suitable, and results in shape distortion. In an image pyramid, because of subsampling, each dimension of the image on the L^{th} level is $1/2^L$ of the original one. x and y coordinate of landmarks are also $1/2^L$. Therefore, all elements of the covariance matrix become $1/4^L$ comparing to those in the original image. When applying PCA now, we obtain the eigen value with $1/4^L$ times of that on the level 0. If we retain the constraint as $\pm 3\sqrt{\lambda_i}$, that leads to an excessive broad limit. Therefore, a distort shape will be generated finally.

To solve the problem, we must modify the range limit on b. In the lower level, the restriction should be relaxed, whereas it should be narrower in the higher level. Shown mathematically, in the level L, the constraint should be

$$-3\sqrt{\frac{\lambda_i}{4^L}} \le b_i \le 3\sqrt{\frac{\lambda_i}{4^L}} \tag{27}$$

where λ_i is the eigenvalue of the covariance matrix of the i^{th} principal component. According to the actual conditions, constraint could be stricter in order to get better search result over the higher level.

Images in the high level are blurred due to subsampling. Thus, the main task of search in the high level is to determine the approximate position of model, while the specific deformation should be obtained by searching in the lower image.

4.2 Width of Search Profile

In classical algorithm, search profile is only along a single line when searching for the suggested movements of landmarks. It could be affected largely by noises, which leads to inaccurate search result. To improve the anti-noise performance of the model, an improvement [14] is shown as follows.

In Figure 16, p_0 is the current landmark, p_1 and p_2 are the nearest pixels on the boundary of p_0. Denote the search profile of p_0, p_1 and p_2 by $g_0^{'}$, $g_1^{'}$ and $g_2^{'}$, where $g_0^{'}$, $g_1^{'}$ and $g_2^{'}$ is acquired by the same method above. The following formula can be used for computing the search profile of p_0 .

$$g_0 = 0.25 g_1^{'} + 0.5 g_0^{'} + 0.25 g_2^{'} \tag{28}$$

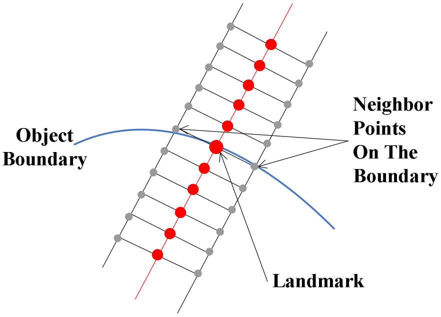


Fig. 16 Search profile with width modification

Then we use this g_0 as current gray-level search profile of landmark p_0 instead of the former profile.

By expanding the width of search profile, the effect of noises can be reduced in some degree. Thus, it can be used to improve the robustness of classical ASM.

4.3 Landmarks Grouping

In Active Shape Model, a new face shape can be linearly represented with a mean shape and some principal component shapes. In equation, it is $X = \bar{X} + Pb$. With the same Training set, mean shape $\bar{X}$ and principal component P is constant. During searching, a new shape is constrained as a face shape by limiting the range of b. Most of the time, $-3\sqrt{\lambda} \leq b \leq 3\sqrt{\lambda}$, where λ is the eigenvalue result of the principal component analysis.

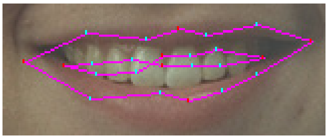


Fig. 17 Mouth shape distort

Some experiments have been done on observing each principal component's effect on face shape's variation. The result shows that some of the organic shape variation isn't decided by a single principal component. For instance, the variety of mouth shape is primarily decided by the some certain components. In this situation, although range of *b* can be limited narrower, unexpected shape with distort mouth shape like Figure 17 will still appear in search procedure. This will lead to an inaccurate searching result.

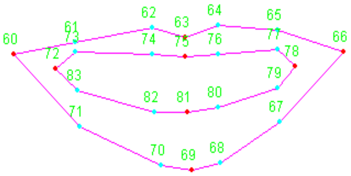


Fig. 18 Mouth shape landmarks

To avoid this problem, a landmark grouping method is proposed [14]. Landmark points of the mouth shape in the face alignment system are shown as the example in Figure 18.

There are 24 landmark points in total, from Number 60 to Number 83, which can be separated into 5 groups. These groups are:

Group 1: [61, 73, 83, 71].
Group 2: [62, 74, 82, 70].
Group 3: [63, 75, 81, 69].
Group 4: [64, 76, 80, 68].
Group 5: [65, 77, 79, 67].

The following segment will explained with Group 1. In Group 1, the 61th landmark point is considered as Group_From, while the 71th landmark point as Group_To. In a proper shape, the right vertical order of these 4 landmark points is: 61--73--83--71. Once the order is wrong, the shape of the mouth is distorted. So retaining the order of the group is necessary for getting a more accurate search result.

In order to make sure that all landmark points in a group are located along a line, the normalized gray derivate will be calculated using the following method:

1. For Group_From and Group_To in the group, the computation of normalized gray derivate is similar to a traditional one.

2. For the others like 73th and 83th in Group 1, landmark point will be vertically mapped on the line connecting Group_From and Group_To first. Then the normalized derivate will be calculated around the mapped point. (Figure 19)

The mapping procedure is illuminated in Figure 16. Denote the group point as N_1, the mapped point of N_1 on line formed by point G_From and G_To as N_1', the slop of the line formed by G_From and G_To as s_1. The coordinate of N_1 and G_From are (x_1, y_1) and (x_2, y_2), the coordinate of N_1' is (x_1', y_1'), $s_1' = \frac{-1}{s_1}$, Then we can get:

$$x' = \frac{(s_1' \times x_1 - s_1 \times x_2 + y_2 - y_1)}{(s_1' - s_1)} \tag{29}$$

$$y' = y_1 + (x' - x_1) \times s_1 \tag{30}$$

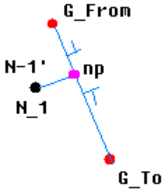


Fig. 19 Special group profile search

Desired movement for a group should obey the following Rule:

1. After calculating the desired movement, new coordinate for each landmark point will be got. In Group 1, they are: (x_{61}, y_{61}), (x_{73}, y_{73}), (x_{83}, y_{83}), (x_{71}, y_{71}).

2. If $(y_{63} \leq y_{73} \leq y_{83} \leq y_{71})$, then location of the landmark points in group 1 will be updated with the new coordinate. Otherwise, distort of mouth has appeared and update will not be done for landmark points in this group. That means all the landmark points in this group will stay at the old position.

Figure 20 shows the search result with a better mouth shape using the group method.

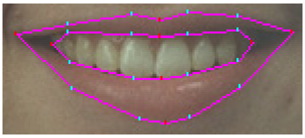


Fig. 20 Search result using landmarks grouping

4.4 Direction of Search Profile

In the search iteration procedure of classical ASM, search profile is obtained by searching along a line passing through the landmark in question and perpendicular to the boundary formed by the landmark and its neighbors [4]. But for some of the landmark points in a face shape, the strongest profile doesn't along the normal direction. Instead, it exists in some other directions. So using the search profile with same length is not appropriate. Owing to this, a more suitable search profile direction for each landmark point is used. We make a difference between inhomogeneous landmarks, by setting more suitable search direction according to the area where the landmark lies. So the search procedure will be more accurate and quicker.

For most landmarks, the normal line passing through the point in the face contour contains the most abundant information and distinguishing characteristics. Whereas the traditional ASM sets the search profile's direction along the line formed by the landmarks and one of its neighbors for convenience. But that should be modified. Instead, we use the normal of the line connecting the two nearest landmarks, which are located on both sides of current landmark respectively, and the normal line passes through the landmark. This improved search direction [14] can make the local gray-level profile of landmarks on the boundary more reasonable.

Take the eyeball shape in a face for example. In traditional ASM the search profile's direction of 29th landmark point is along the line perpendicular to the boundary formed by landmark point Number 28 and number 29 (shown with yellow line in Figure 21). In the improved algorithm, we set the search profile's direction of number 29 along the line formed by 29th and 26th landmark points (shown with blue line in Figure 21), because the profile information along this line seems to be more abundant than the previous one.

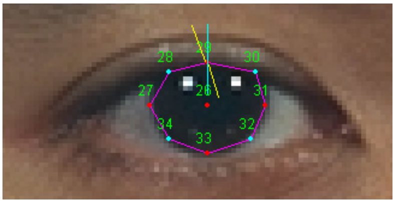


Fig. 21 Improvement on search direction

4.5 Skin-Color Model

In classical ASM, the local gray-level information around the landmarks is used for modeling. We also use it in calculating the suggested movements of landmarks during image search. Despite this, there is still a problem that should be considered.

Human faces are usually affected by hair, ornament, race and light conditions. Such effect makes that the gray distribution in the same position may vary significantly, which leads to that some landmarks lose their significance. For instance, the brightness of women's hair is no obvious difference from the brightness of face areas; illumination can vary the intensity of faces largely. These conditions can greatly affect the modeling and applying ASM.

Skin color is one of the features on human faces. Skin colors cluster in a small region in a color space [15]. Also their study shows that human skin colors differ more in intensity than in colors. At the same time, different skin color has a same *2D* Gaussian distribution model $G\left(Mean, Cov^2\right)$.

In order to improve the performance of ASM based on gray-level information, we can use a skin color likelihood transformation method to make the profile focus more on the skin color. ASM using skin color likelihood transformation method is called Skin-ASM.

The transformation method is composed of 3 steps: (1) color space translation; (2) likelihood calculation; (3) likelihood to gray image projection.

Most color face images use a RGB representation; However, RGB is not necessarily the best color representation for characterizing skin-color [16]. Because tricolor not only represents color, but also represents intensity which may vary largely due to surrounding environments and other reasons. So we first translate RGB into a new color space that separate colors from intensity. In our model, we choose the YCrCb color space as the skin color representation space. Space translation formula is showed below:

Y=0.299R+0.587G+0.114B

Cb=0.169R+0.331G+0.500B

Cr=0.500R+0.419G+0.081B

Then we obtain new images in the YCrCb color space.

The distribution of skin colors can be considered as a 2D Gaussian model $G(Mean, Cov^2)$ [15]. So, we can calculate the likelihood value *P* between the pixel's color value *X* and the mean skin color using equation:

$$P = e^{(-\alpha(X-\bar{X})^T C^{-1}(X-\bar{X}))} \tag{31}$$

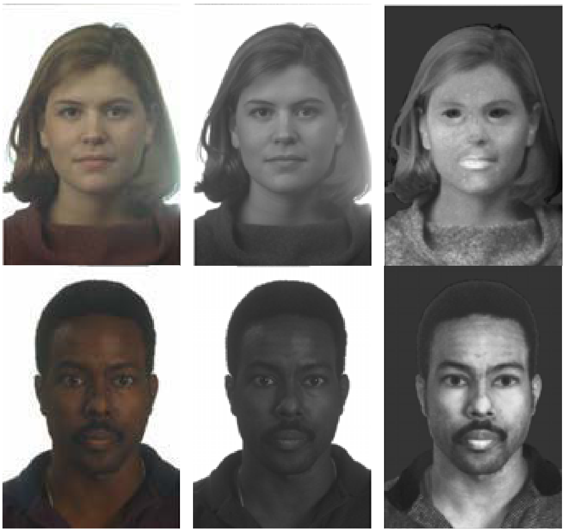
where $X = (C_r, C_b)^T$ is the color value of the pixel. $\bar{X} = (\bar{C}_r, \bar{C}_b)^T$ is the mean matrix of skin color's 2D Gaussian model. $C = E\left[(X-\bar{X})(X-\bar{X})^T\right]$ is the Covariance matrix of 2D Gaussian model. α is a constant value between [0, 1].

The calculated result above is a probability value between [0, 1]. However, it cannot be displayed in grayscale. In step 3, we will project it into the value between [0, 255] as the gray level using the following equation.

$$g_{ij} = \frac{P_{ij}}{P_{\max}} * 255 \tag{32}$$

where max $P_{\max}$ is the maximum likelihood value of all the pixels in an image.

We choose several representative images and get their skin color likelihood transformation results as Figure 22 shows. The gray distribution of human faces becomes more concentrated, also has less relationship with race and illumination. Comparing with original gray level image, the skin likelihood image has more abundant contour information, especially in eyes and mouth. Therefore, it is more reliable for ASM modeling.



(a) color image (b) gray image (c) skin likely image

Fig. 22 Comparisons of color image, gray image and skin likely image

5 Related Work

Besides the improvements we discussed above, other research [17-19] is devoted to improve the performance of classical ASM. Van Ginneken *et al.* [20] proposed a non-linear gray-level appearance model to improve ASM, which is called Active Shape Model with optimal features (OF-ASM). It uses a non-linear kNN-classifier instead of sampling along the normal to the object boundary. For each landmark a square grid of N×N points is defined with n grid an odd integer and the landmark point at the center of the grid with weight imposed on each neighbor point. Features are extracted by taking the first few moments of the local distribution of image intensities around each location, selected by using the training image and sequential feature forward and backward selection. In the search procedure, using a suitable measurement, the optimal feature set is fed into a kNN classifier to determine the probability that the pixel is inside or outside the target object. It outperforms traditional ASM but is computationally expensive. Based on OF-ASM, Ordas *et al.* [21] proposed an extension to the non-linear appearance approach, incorporating a reduced set of differential invariant features as local image descriptors. The new method is Active Shape Models with invariant optimal features (IOF-ASM). It is invariant to Euclidean transformations given that only Cartesian differential Euclidean invariants are chosen as local image descriptors. These improvements are used in medical image segmentation.

Yan *et al.* [22] proposed a Texture-Constrained Active Shape Model (TC-ASM). It inherits the local appearance model in ASM for the robustness of varying lighting combined with global texture, which acts as a constraint over shape and provides an optimization criterion for determining the shape parameters. In order to formula the correlations between shape and texture, texture is mapped onto the shape space linearly by a projection matrix pre-computed by SVD. In each step of optimization, a better shape is found under Bayesian framework.

Wang *et al.* [23] introduced an improved ASM used for generic face alignment in the case that new faces are not in the training set, which includes three improvements. First, random forest classifiers are trained to recognize local appearance around each landmark instead of gray-level and add weight matrix derived form the outputs of random forest classifiers to the optimization. Second, shape vectors are restricted to the vector space spanned by the training database. Third, data augment algorithm is used. The authors claimed that the improvements can achieve good performance.

Romdhani *et al.* [24] proposed a multi-view nonlinear ASM using Kernel PCA. It utilizes 2D view-dependant constraint without explicit reference to 3D structures. Such a model captures all possible 2D shape variations in a training set and performs a nonlinear transformation of the model during matching. For nonlinear transformation, Kernel PCA based on SVM is used. The improved model can deal with large pose variations and nonlinear shape space.

Hamarneh and Gustavsson [25] extend 2D Active Shape Models to 2D+time by presenting a method for modeling and segmenting spatio-temporal shapes (ST-shapes). The modeling part consists of constructing a statistical model of ST-shape parameters and describes the principal modes of variation of the ST-shape in addition to constraints on the allowed variations. Segmentation results on both synthetic and real data are presented in their paper.

Seshadri, K *et al* [26] proposed an improved method for locating facial landmarks in images containing frontal faces using a modified active shape model, which includes the use of an optimal number of facial landmark points, better profiling methods during the fitting stage and the development of a more suitable optimization metric to determine the best location of the landmarks compared to the simplistic minimum Mahalanobis distance criteria used to date.

Pengfei Xiong *et al* [27] propose a new algorithm for shape initialization and 3D pose alignment in Active Shape Model, instead of initializing with average shape in previous works, they build a scatter data interpolation model from key points to obtain the initial shape, which ensures shape initialized around face organs. These key points are chosen from organs of face shape and located with a strong classifier firstly. Then they are utilized to build a Radial Basis Function (RBF) model to deform the average shape as initial shape.

In [28], inspired by physics, the boundary moment invariants are employed to resolve the difficulty that during the process of shape fitting, distortions and displacements often occur when the target is not clear or with defects, and there is a lack of effective amendment strategies in ASM. Moment invariants have been introduced into ASM for the first time for distortion detection and shape amendment.

Shanhui Sun *et al* [29] presented a new fully automated approach for segmentation of lungs with such high-density pathologies. Their method consists of two main processing steps. First, a novel robust active shape model (RASM) matching method is utilized to roughly segment the outline of the lungs. Second, an optimal surface finding approach is utilized to further adapt the initial segmentation result to the lung.

Lee, Yong-Hwan *et al* [30] addressed issues related to face detection and implements an efficient extraction algorithm for facial landmarks suitable for use on mobile devices. The original ASM was modified to enhance its performance firstly improving the initialization model using the center of the eyes by utilizing a feature ma of RGB color information, secondly building a modified model definition and fitting more landmarks than the classical ASM, and also extending and building a 2-D profile model for detecting faces in input images.

Other researchers apply ASM to 3D objects [31], for outlier detection [32], facial feature tracking [33], and MR image segmentation [34]. Zhao *et al.* [35] use a 3D Partitioned Active Shape Model (PASM) to deal with small training set. They use curve alignment to fit models during deformations. Each training sample and deformed model is represented as a curve. ASM is even used in road network structure recognition [36].

6 Conclusions

In this chapter, we introduced ASM as a data match technique and its application to face recognition. We collect training images and represent all shapes with a set of landmarks, to form a Point Distribution Model (PDM) respectively. After landmarks alignment and Principal Component Analysis, we construct gray-level profile for each landmark in all multi-resolution versions of a training image. In search procedure, we give the model's position an initial estimate. Then it can compute the suggested movements through an iteration approach using the gray-level profile. When convergence is established, we get a final matching result. We adopt a lot of improvements to the classical ASM, such as increasing the width of search profile to reduce the effect of noise, grouping landmarks to avoid mouth shape distort in the search procedure and altering the direction of search profile.

References

1. Cootes, T.F., Taylor, C.J.: Statistical Models of Appearance for Computer Vision. Technical Report (2004)
2. Cootes, T.F., Taylor, C.J.: Active Shape Models - 'Smart Snakes'. In: Proc. British Machine Vision Conference, pp. 266–275 (1992)

3. Cootes, T.F., Taylor, C.J., Cooper, D., Graham, J.: Active Shape Models – Their Training and Application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
4. Hamarneh, G.: Active Shape Models, Modeling Shape Variations and Gray Level Information and an Application to Image Search and Classification. Technical Report R005/1998 (S2-IAG-98-1). Chalmers University of Technology, Sweden (1998)
5. Lanitis, A., Taylor, C.J., Cootes, T.F.: A Unified Approach to Coding and Interpretting Faces. In: Proc. 5th International Conference on Computer Vision, pp. 368–373 (1995)
6. Lanitis, A., Taylor, C.J., Cootes, T.F.: Automatic Interpretation and Coding of Face Images Using Flexible Models. IEEE PAMI 19(7), 743–756 (1997)
7. Cootes, T.F., Page, G.J., Jackson, C.B., Taylor, C.J.: Statistical Grey-Level Models for Object Location and Identification. Image and Vision Computing 14(8), 533–540 (1996)
8. Cootes, T.F., Taylor, C.J.: Using Grey-Level Models to Improve Active Shape Model Search. In: Proc. International Conference on Pattern Recognition, vol. 1, pp. 63–67 (1994)
9. Cootes, T.F., Taylor, C.J., Lanitis, A.: Multi-Resolution Search with Active Shape Models. In: Proc. International Conference on Pattern Recognition, vol. 1, pp. 610–612 (1994)
10. Cootes, T.F., Taylor, C.J., Lanitis, A.: Active Shape Models: Evaluation of a Multi-Resolution Method for Improving Image Search. In: Proc. the British Machine Vision Conference, pp. 327–336 (1994)
11. Edwards, G.J., Lanitis, A., Taylor, C.J., Cootes, T.F.: Face recognition using statistical models. IEE Colloquium on Image Processing for Security Applications, No. 1997/074, 2/1-2/6 (1997)
12. Hill, A., Cootes, T.F., Taylor, C.J.: Active Shape Models and the shape approximation problem. Image and Vision Computing 14(8), 601–608 (1996)
13. Cootes, T.F., Edwards, G., Taylor, C.J.: Comparing Active Shape Models with Active Appearance Models. In: Proc. the British Machine Vision Conference, pp. 173–182 (1999)
14. Lu, H., Shi, W.: Accurate Active Shape Model for Face Alignment. In: 17th IEEE International Conference on Tools with Artificial Intelligence, pp.642–646 (2005)
15. Yang, J., Lu, W., Waibel, A.: Skin-color Modeling and Adaptation. In: Chin, R., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1352, pp. 687–694. Springer, Heidelberg (1997)
16. Yang, J., Waibel, A.: A Real-time Face Tracker: Applications of Computer Vision. In: Proc. 3rd IEEE Workshop on WAC 1996, pp. 142–147 (1996)
17. van Ginneken, B., Frangi, A.F., Staal, J.J., ter Haar Romeny, B.M., Viergever, M.A.: A non-linear gray-level appearance model improves active shape model segmentation. In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 205–212 (2001)
18. Zhang, L., Ai, H.: Multi-View Active Shape Model with Robust Parameter Estimation. In: 18th International Conference on Pattern Recognition, pp. 465–468 (2006)
19. Xin, S., Ai, H.: Face Alignment under Various Poses and Expressions. In: Proc. SPIE, vol. 5286(558), pp. 40–47 (2003)

20. van Ginneken, B., Frangi, A.F., Staal, J.J., ter Haar Romeny, B.M., Viergever, M.A.: Active Shape Model Segmentation with Optimal Features. IEEE Trans. on Medical Imaging 21(8), 924–933 (2002)
21. Ordas, S., Boisrobert, L., Huguet, M., Frangi, A.F.: Active Shape Models with Invariant Optimal Features (IOF-ASM) Application to Cardiac MRI Segmentation. Computers in Cardiology, 633–636 (2003)
22. Yan, S., Liu, C., Li, S.Z., Zhang, H., Shum, H.-Y., Cheng, Q.: Face Alignment Using Texture-constrained Active Shape Models. Image and Vision Computing 21, 69–75 (2003)
23. Wang, L., Ding, X., Fang, C.: Generic Face Alignment Using an Improved Active Shape Model. In: International Conference on Audio, Language and Image Processing, pp. 317–321 (2008)
24. Romdhani, S., Gong, S., Psarrou, A.: A Multi-View Nonlinear Active Shape Model Using Kernel PCA. In: Proc. the British Machine Vision Conference, pp. 483–492 (1999)
25. Hamarneh, G., Gustavsson, T.: Deformable Spatio-temporal Shape Models: Extending Active Shape Models to 2D+time. Image and Vision Computing 22, 461–470 (2004)
26. Seshadri, K., Savvides, M.: Robust modified Active Shape Model for automatic facial landmark annotation of frontal faces. Biometrics: Theory, Applications, and Systems. In: IEEE 3rd International Conference on BTAS, pp. 1–8 (2009)
27. Xiong, P., Huang, L., Liu, C.: Initialization and Pose Alignment in Active Shape Model. In: 20th International Conference on Pattern Recognition, pp. 3971–3974 (2010)
28. Chen, S.Y., Zhang, J., Guan, Q., Liu, S.: Detection and amendment of shape distortions based on moment invariants for active shape models. IET Image Processing 5(3), 273–285 (2011)
29. Sun, S., Bauer, C., Beichel, R.: Automated 3-D Segmentation of Lungs With Lung Cancer in CT Data Using a Novel Robust Active Shape Model Approach. IEEE Trans. on Medical Imaging 31(2), 449–460 (2011)
30. Lee, Y.-H., Yang, D.-S., Lim, J.-K., Lee, Y., Kim, B.: Improved Active Shape Model for Efficient Extraction of Facial Feature Points on Mobile Devices. In: Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 256–259 (2013)
31. Lekadir, K., Merrifield, R., Yang, G.-Z.: Outlier Detection and Handling for Robust 3-D Active Shape Models Search. IEEE Trans. on Medical Imaging 26(2), 212–222 (2007)
32. Tong, Y., Wang, Y., Zhu, Z., Ji, Q.: Robust Facial Feature Tracking under Varying Face Pose and Facial Expression. Pattern Recognition 40, 3195–3208 (2007)
33. Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: The Use of Active Shape Models for Locating Structures in Medical Images. Image and Vision Computing 12, 355–366 (1994)
34. Zhao, Z., Teoh, E.K.: Robust MR Image Segmentation Using 3D Partitioned Active Shape Models. In: International Conference on Control, Automation, Robotics and Vision, pp. 1–6 (2006)
35. Koutaki, G., Uchimura, K., Hu, Z.: Network Active Shape Model for Updating Road Map from Aerial Images. In: IEEE Intelligent Vehicles Symposium, pp. 325–330 (2006)
36. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face Recognition Algorithms. IEEE PAMI 22, 1090–1104 (2000)

Chapter 2
Condition Relaxation in Conditional Statistical Shape Models

Elco Oost, Sho Tomoshige, and Akinobu Shimizu

Abstract. A conditional statistical shape model is a valuable subspace method in medical image analysis. During training of the model, the relationship between the shape of the object of interest and a set of conditional features is established. Subsequently, while analyzing an unseen image, a measured condition is matched with this conditional distribution and then a subspace of the training data is marked as relevant and used for the desired reconstruction of the object shape. This approach can work properly only in the case when the conditional term is reliable. Unfortunately, the reliability of the conditional term is not always sufficiently high and in such situation, instead of being beneficial, the conditional term is hampering the statistical shape model. This chapter describes the advantages and disadvantages of conditional statistical shape models and discusses how relaxation of the conditional term can help to deal with possible unreliability of the conditional term. The requirements for the construction of a properly functioning relaxed conditional shape model are defined and the optimal design is tested against various alternative (relaxed) conditional shape models, showing the superiority of the optimally designed relaxed conditional shape model.

1 Introduction

One of the main objectives in medical image processing is to develop tools for automated image interpretation, either fully automatic or otherwise with minimal user interaction. In the past years a multitude of diagnostic tools, such as Computer Aided Diagnosis (CAD) and Computer Aided Surgery (CAS) have been developed, all with the same aims: reducing the workload of the clinician, speeding up the diagnosis and standardizing the diagnosis. In modern clinical practice where

E. Oost · S. Tomoshige · A. Shimizu
Tokyo University of Agriculture and Technology
Nakacho 2-24-16, Koganei, Tokyo, 184-8588 Japan
e-mail: simiz@cc.tuat.ac.jp

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_2, © Springer-Verlag Berlin Heidelberg 2014

state-of-the-art high-resolution medical imaging devices produce increasingly large data sets, achieving these three goals is essential.

The backbone of automated image interpretation tools is the incorporation of a priori knowledge on how a medical doctor would interpret the image. By supplying numerous examples, the diagnostic tool should be able to mimic the interpretation of a medical doctor, and for that purpose, statistical models have been developed. A major contribution has been made by Cootes *et al.*, with the development of the Active Shape Model (ASM) (Cootes *et al.* 1995). An ASM is a Point Distribution Model (PDM) based approach that is trained on a large data set of manually drawn shapes of an object. It models the shape of an object in terms of the average object shape and a series of orthogonal shape variations, obtained by eigenvector decomposition. During segmentation of the object in an unseen image, the delineated object shape is iteratively updated using the underlying image information and subsequently constrained by the statistical shape model (SSM), to guarantee that the final outcome is a realistic shape. In this process, the definition of a realistic shape is defined by the medical doctor(s) who supplied the manually drawn training shape examples. Active Shape Models can be applied either to 2D, such as for example shown by (van Ginneken *et al.* 2002), 2D+time (Hamarneh and Gustavsson 2004), or to 3D, such as for example presented by (van Assen *et al.* 2003).Furthermore, the introduction of the Active Appearance Model (Cootes *et al.* 1998) has extended the modeling concept to also incorporate the texture of the object, as seen in the image. A comprehensive overview of PDM based modeling in the field of medical image processing is provided by (Heimann and Meinzer 2009).

A different approach to mimic the image interpretation of a medical doctor is level set based modeling, introduced by (Leventon *et al.* 2000). Instead of manual delineations it is trained on labeled data sets, in which the labels denote the object of interest. level set based models produce a slightly less accurate shape representation, but they have a major advantage that they do not require point correspondence between the training shapes. (Cremers *et al.* 2007) elaborately discussed the usage of level set based modeling.

Regarding the previously defined objectives of automated image interpretation, (Oost *et al.* 2009) have shown in a clinical validation study on left ventricular angiography, that automated image analysis algorithms are capable of reducing the workload of a medical doctor. Instead of a full delineation of the object of interest, only occasional editing of the automatically generated shape is needed. Consequently, the average analysis time required per case can be significantly reduced. Besides these two practical improvements, automatic image interpretation can also lead to improvements in diagnostic accuracy. (Oost *et al.* 2009) have shown that automated segmentation accuracy is between inter-observer and intra-observer variability, hence creating a reliable standardization of the diagnosis. This standardization can be further strengthened by training the statistical model on manual delineations of multiple medical doctors.

Despite the promising results that have been obtained with the various modeling approaches, there are a number of drawbacks and limitations in statistical modeling. One of them is the overfitting to the training data. Particularly when the number of training samples is limited, the segmentation result will remain close to the model's average shape representation, leading to suboptimal object delineation. Another drawback is that the statistical models generally assume a single Gaussian distribution of the training data, while in practice the data is best represented by multiple Gaussian kernels. For example, when modeling the texture of the myocardium in delayed enhanced MRI scans, myocardial regions are either dark, representing healthy tissue, or bright, representing infarcted tissue. Although a 'half-bright' representation of local tissue is physically impossible, combining healthy tissue training samples and infarcted tissue training samples in a single Gaussian model representation will allow 'half-bright' as a valid texture representation.

Most statistical (shape or texture) models are constructed from the training data that is available in daily clinical practice. Hence, it involves a large partition of healthy 'normals', plus a series of smaller groups of various pathological cases. Because the various pathological sub-sets are generally too small to create individual statistical models per pathology, all training data is combined in one statistical model, with the erroneous assumption that the data fits a single Gaussian distribution. Consequently, the relatively large sub-set of healthy normals will result in a bias towards the average model representation, while the single Gaussian assumption allows unnatural representations that average between normal and pathology. The overall segmentation therefore becomes sub-optimal: the global result is somewhat acceptable, but local detail is lacking.

One approach to improve segmentation results is by finding the optimal balance between healthy and pathological training samples. In a cardiac MRI segmentation study, (Angelie *et al.* 2007) have shown for example that optimal segmentation results are obtained when the training data set is consisted of 80% healthy data samples and 20% pathological samples. A more common segmentation improvement approach is a postprocessing of the model generated segmentation. The globally acceptable model segmentation result is refined, using information from the underlying image. In a 2D application (x-ray left ventricle angiography) (Oost *et al.* 2006) have shown that a refinement of the model based delineation, by using dynamic programming, results in a higher segmentation accuracy. The combination of an Active Appearance Model with dynamic programming outperformed these two approaches, when applied individually.

Recognizing that the segmentation result obtained from a statistical model might not always be the optimal shape delineation, statistical models have recently been employed differently. Using a statistical shape model, for an unseen image, the optimal model-based shape representation of the object of interest is constructed. Subsequently this estimated shape is used as a regulating term in the optimization of the energy function of a graph cut segmentation (Boykov and Funka-Lea 2006). Using this approach, successful segmentation was obtained in single-shape graph cut segmentation by (Shimizu *et al.* 2010) and in multi-shape segmentation by (Nakagomi *et al.* 2013).

Alternatively, conditional statistical models (de Bruijne *et al.* 2007) can increase the segmentation performance by focusing on the relevant shape subspace in which the segmentation result should be searched for. By using a priori information as a condition, a conditional SSM imposes an additional restriction on the shape subspace that is constructed by a conventional SSM. Hence, the model will focus on only part of the distribution, and this subset of the data might be expressed by fewer Gaussians, or even by a single Gaussian. This way, the conditional statistical model can be an effective approach to solve the limitations of standard statistical models, as described above. In practice, when, for example from a set of shape and/or image features, it can be derived in which sub-group (healthy, pathology A, pathology B, etcetera) the unseen image fits best, the segmentation process can focus on the subspace that corresponds to this sub-group of the data.

The remainder of this chapter will further focus on conditional statistical shape models, and how they can be employed to improve segmentation results. Firstly, in Section 2 the formulation of a general conditional statistical shape model is described, followed by the benefits (Section 3) and limitations (Section 4) that arise when the underlying features are not fully reliable. In Sections 5 and 6 level set based conditional SSMs and relaxation of the conditional term are discussed. Subsequently, in Sections 7 and 8, two approaches are presented to relax the influence of the conditions: A relaxed conditional statistical shape model, based on the selection formula (Lord and Novick 1968), and a relaxed conditional statistical shape model with integrated conditional error estimation. The performance of these algorithms will be presented for automatic liver segmentation in non-contrast abdominal CT images in Section 9.

2 Conditional Statistical Shape Models

The general formulation of a conditional statistical shape model (SSM) is given by:

$$\mu_{b|x_0} = \mu_b + \Sigma_{bx}\Sigma_{xx}^{-1}(x_0 - \mu_x) \tag{1}$$

$$\Sigma_{bb|x_0} = \Sigma_{bb} - \Sigma_{bx}\Sigma_{xx}^{-1}\Sigma_{xb} \tag{2}$$

in which b is a set of shape parameters (Cootes *et al.* 1995), μ_b is the average parametric shape representation, Σ_{bb} is the covariance matrix for the parameterized training shape data samples, μ_x is the average set of conditional features, Σ_{xx} is the covariance matrix for the conditional data matrix X, Σ_{xb} and Σ_{bx} are mutual covariance matrices, x_0 is the measured set of conditional features for the unseen image and $\mu_{b|x_0}$ and $\Sigma_{bb|x_0}$ are respectively the conditional average (shape) and the conditional covariance matrix, given the condition x_0.

Similar to a non-conditional model, the conditional SSM has an average representation of the training data (1) and a covariance matrix (2), describing the variation within the training samples. For the training of a conditional SSM, a set of training shape samples and a set of conditional features are required. The parameterized training shape samples are combined in matrix b: all individual training shapes (represented either as a set of landmark points, or as a signed distance map of the object) are placed in column vectors and Principle Component Analysis (PCA) is applied on these vectors to create an SSM. Projection of the individual training shapes onto the SSM leads to the (N-1) by N sized principle component score matrix b, with N denoting the number of training samples. Note that generally PCA is applied for model construction, but that other approaches, such as Independent Component Analysis (ICA) (Üzümcü et al. 2003) or manifold learning (Pless and Souvenir 2009) can also be used.

The definition of the conditional features can, for example, be the landmark points of an adjacent object, or features derived from the underlying image in which the object of interest is embedded. Any feature can be selected, as long as the feature acquisition is performed identically for all individual training samples. All conditional features are accumulated in the conditional data matrix *X*, whose dimensionality is F by N, with F signifying the number of selected features. The third and final input for the conditional SSM is a set of conditional features, extracted from a new, unseen image and/or shape. These features should correspond to the features that are stored in matrix X, and are defined as the condition x_0. Summarizing the behavior of (1) and (2), the conditional average $\mu_{b|x_0}$ is calculated from the average training shape μ_b, the average set of conditional features μ_x, (derived from matrix X), the condition x_0 (obtained from the unseen sample), the covariance matrix of *X* (denoted by Σ_{xx}) and the mutual covariance matrix of *X* and *b* (denoted by Σ_{bx}). The conditional covariance matrix $\Sigma_{bb|x_0}$ of (2) is calculated from the covariance matrices for *X* (Σ_{xx}) and *b* (Σ_{bb}), and the mutual covariance matrices Σ_{xb} and Σ_{bx}. Subsequent eigenvalue decomposition of $\Sigma_{bb|x0}$ leads to construction of the conditional SSM.

3 The Benefit of Conditional SSMs

The benefit of conditional statistical shape models is shown in various applications. (de Bruijne *et al.* 2007) compare a non-conditional SSM and a conventional conditional SSM when applied to vertebra fracture quantification. For a fractured vertebra, the reconstructed (un-fractured) shape is estimated. Comparison of the fractured vertebra shape and the reconstructed shape results in a measure of the severity of the fracture. Because a non-conditional SSM cannot use knowledge on neighboring vertebrae to predict the reconstructed shape, the average vertebra shape is used to represent the reconstructed shape. In their experiments, using a large data set of 282 lateral lumbar spine radiographs, (de Bruijne *et al.* 2007) show that the reconstructed shape produced by the conditional SSM, based on the shape of neighboring vertebrae, is more accurate than using the average vertebra shape for the quantification of vertebra fracture severity.

(Syrkina *et al.* 2011) apply their multivariate-Gaussian distribution based conditional SSM to the prediction of the proximal tibia shape, based on knowledge of the shape of the distal femur. Using a data set of 184 left leg tibia and femur pairs (125 for training and 59 for evaluation), they show that solely based on the shape of the distal femur, the proximal tibia shape can be estimated with a reconstruction error of only a few millimeters. Because this application requires a strong relationship between the shape of the predictor and the shape of the unseen part, PDM based modeling (with the landmarks of a neighboring object as the predictor) is probably more useful than level set based modeling (with image derived conditional features).

(Baka *et al.* 2010) show that their approach for dense shape reconstruction from a sparse point cloud results in a proper reconstruction of the unknown landmarks. Experiments on a MRI data set of 114 combined left and right cardiac ventricle shapes, show that their proposed method outperforms a conventional conditional SSM.

All these three papers describe PDM based conditional models, in which the conditional term is simply the set of landmark points of a neighboring object. As a consequence, the relationship between the predictor and the unseen part is strong. A more challenging test for conditional SSMs is when the model is level set based and the conditional features are image derived. In such a setup, the relationship between the conditional term and the desired shape is weak.

4 Reliability of the Conditional Term

The main concept of the conditional SSM is that the addition of conditional data will improve the segmentation performance of the model. More a priori knowledge leads to better shape description. This assumption is true when the condition is reliable. (de Bruijne *et al.* 2007) for example show the power of the conditional SSM in predicting the reconstructed (unfractured) shape for a fractured vertebra, based on the shape information of neighboring vertebrae. However, when the reliability of the condition is not particularly high, the conditional SSM will not be superior to a standard, non-conditional SSM. An unreliable condition might even mislead the model and deteriorate segmentation results.

In some applications, the reliability of the conditional features can be expected to be high. In (de Bruijne *et al.* 2007) for example, it can be expected that shapes of neighboring vertebrae are strongly correlated. The PDM based conditional modeling of a vertebra, based on the shape of neighboring vertebrae, can therefore be judged as having a reliable conditional term. Also the shape of the distal femur is a reliable condition, when estimating the shape of the proximal tibia, because human anatomy dictates that the head of the tibia should fit the base of the femur in order to have a properly functioning joint.

Less reliable are image extracted conditional features, as used for example in the level set based conditional SSM as proposed by (Tomoshige *et al.* 2012). As they are derived either directly from the underlying image or from the outcome of low-level image processing algorithms, there is no a priori knowledge involved. Thus, the reliability of the conditional features is expected to be low.

The remainder of this chapter will concentrate on the construction and application of level-set based conditional statistical shape models, with a strong focus on how to deal with the unreliability of the condition. Central topic will be the relaxation of the condition such that the additional conditional knowledge that is inputted into the model is beneficial instead of detrimental.

5 Level Set Based Conditional SSMs

The difficulty of 3D PDM based shape modeling lies in obtaining point correspondence among the training samples. In level set based SSMs point correspondence is not an issue. Hence, it is relatively easy to implement a segmentation pipeline in which the object of interest is roughly delineated by a (conditional) level set based SSM, after which the resultant shape is used as the shape prior for a graph cut segmentation. This approach will be elaborated on in the following paragraphs.

Graph cuts, introduced by (Boykov and Funka-Lea 2006), have become a popular segmentation tool over the last decade. Also in the field of medical image processing, several successful graph cut based segmentation algorithms have been published (Freedman and Zhang, 2005; Shimizu *et al.*, 2010; Linguraru *et al.*, 2012; Nakagomi *et al.*, 2013). To reproduce a natural shape of the object of interest, the graph cut segmentation requires a shape prior that will be used as a regulating term while optimizing the energy function of the graph cut segmentation. The final delineation should not differ too much from the initial shape prior. Using a level set based SSM segmentation result as the shape prior will guarantee that the final object delineation has a natural shape. To further improve the shape prior, estimated features of the target object can be used as conditional data in a conditional SSM. (Tomoshige *et al.* 2012) for example start with a Maximum A Posteriori (MAP) segmentation of the object of interest and derive a set of conditional features from the roughly segmented MAP volume. Features can be the total object volume, the area of the projected object in the axial, sagittal or coronal plane, the nth percentile point of the x-, y-, or z-coordinate, and so on. Based on these features, the conditional SSM creates an intermediate delineation, which is offered to the graph cut algorithm in order to create a final, accurate object segmentation.

6 Relaxation of the Conditional Term

The performance of the graph cut segmentation strongly depends on the quality of the shape prior estimation by the conditional SSM, which in turn is dependent on the quality of the estimated conditional features. Because the features are extracted without a priori knowledge, it is difficult to improve this part of the segmentation pipeline. It is close to impossible to make the error of any (feature extraction) process zero, in particular for the segmentation of organs with an atypical shape. Improvements can however be obtained in how the conditional SSM processes the conditional features. Instead of using a fixed conditional term as input for the conditional SSM, the conditional term must be relaxed, such that the condition gives direction towards the optimal shape prior, instead of providing a rigid shape constraint. In the ideal situation, the influence of the condition should be parametrically positioned on a linear trajectory in the shape parameter subspace, connecting a conventional conditional SSM (with a fixed constraint) and a non-conditional SSM. By creating a seamless transition between these two extremities, the degree of relaxation of the conditional term can be chosen according to the reliability of the conditional features. Constructing the domain of the relaxed conditional SSM, as depicted in figure 1, requires the calculation of the conditional average as well as the calculation of the conditional covariance matrix. Only when both can be calculated, a seamless and natural transition between the conventional conditional SSM and the non-conditional SSM can be realized.

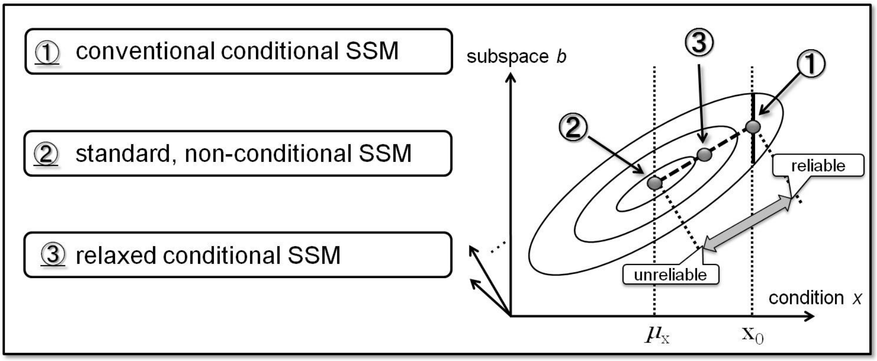

① conventional conditional SSM
② standard, non-conditional SSM
③ relaxed conditional SSM
subspace b
③
①
②
reliable
unreliable
condition x
μ_x
x_0

Fig. 1 The desired relaxed conditional SSM bridges seamlessly between the non-conditional SSM and the conventional conditional SSM

In literature, several approaches for the relaxation of the conditional term have been proposed. A prominent contribution is the work by (de Bruijne *et al.* 2007) in which the shape of a vertebra is estimated using a conditional statistical shape model that uses the shape information of neighboring vertebrae as conditional term. To avoid matrix singularity, a regulating term ρI is added to (1) and (2), resulting in:

$$\mu_{b|x_0} = \mu_b + \Sigma_{bx}(\Sigma_{xx} + \rho I)^{-1}(x_0 - \mu_x) \tag{3}$$

$$\Sigma_{bb|x_0} = \Sigma_{bb} - \Sigma_{bx}(\Sigma_{xx} + \rho I)^{-1}\Sigma_{xb} \tag{4}$$

with I denoting the identity matrix and ρ signifying a value between zero and infinity, taking usually a small value. This technique, introduced by (Hoerl and Kennard 1970), is known as ridge regression and incidentally supports the desired seamless transition between the non-conditional SSM and conventional conditional SSM. If ρ is zero, (3) and (4) are identical to (1) and (2), signifying the conventional conditional SSM. If ρ approaches infinity, the influence of the conditional term is reduced to zero, resulting in the non-conditional SSM. One drawback of ridge regression is that, because the regulating term ρI is a constant absolute value, added to all conditional variations that are contained in Σ_{xx}, the weak shape variations in Σ_{bb} are disproportionally strongly affected by the regulating term. This implies that, the closer the model tends to be towards a non-conditional SSM, the more marginal the effect of the conditional term on the weak shape variations becomes. In other words, when the model is close to position 2 in figure 1, the (relaxed) conditional term will effectively only influence the strongest shape variations.

Other PDM based conditional SSMs have been proposed for example by (Syrkina *et al.* 2011) and (Baka *et al.* 2010). In essence similar to the conditional SSM by (de Bruijne *et al.* 2007), (Syrkina *et al.* 2011) select a subset of the landmark points of the object (or objects) of interest and use this subset as the conditional term to predict the shape representation of the set of remaining (‘unseen’) landmark points. The conditional shape of the set of remaining points is calculated from the joint Gaussian distribution of the predictors and the unseen points. During training the number of shape variation modes for both the predictor shape model and the unseen part shape model is optimized such that the average prediction error is minimized. This implies that when the relationship between predictor and unseen part is low, a large portion of the statistical shape information is rejected from the model and only the larger eigenmodes are retained. Although the optimization of shape modes can be seen as a form of relaxation, a true relaxation of the condition is not integrated in the approach by (Syrkina *et al.* 2011).

(Baka *et al.* 2010) also divide the shape vector into two parts: an unknown part and a constrained part that has only limited freedom to deform. The constrained points then are used as the conditional term to predict the position of the unknown landmarks, while the uncertainty of the condition is incorporated into the conditional model. In this approach the conditional covariance matrix is defined, but the conditional average is not calculated. Consequently, the desired seamless transition between the non-conditional SSM and the conventional conditional SSM cannot be achieved.

7 Employing the Selection Formula for Relaxation

(Tomoshige *et al.* 2012) propose a relaxed conditional SSM that does allow this seamless transition and design it such that the drawbacks of the ridge regression approach are overcome. Furthermore, since their approach is a level set based conditional SSM, and the conditional features are derived from a MAP segmentation of the underlying image, the reliability of the conditions is not too high. Consequently, their objective is to create a relaxed conditional SSM with a set of reliability parameters, to easily optimize the performance of the model.

Using the selection formula from (Lord and Novick 1968), only a limited range of the conditional features can be selected, softening the influence of the conditions on the shape estimation. Starting with the regular set of covariance matrices for the conditional SSM

$$\begin{pmatrix} \Sigma_{xx} & \Sigma_{xb} \\ \Sigma_{bx} & \Sigma_{bb} \end{pmatrix} \tag{5}$$

and defining V_{xx} as a limited range of the conditional data of Σ_{xx}, the set of covariance matrices can be rewritten as:

$$\begin{pmatrix} V_{xx} & V_{xx}\Sigma_{xx}{}^{-1}\Sigma_{xb} \\ \Sigma_{bx}\Sigma_{xx}{}^{-1}V_{xx} & \Sigma_{bb} - \Sigma_{bx}\left(\Sigma_{xx}{}^{-1} - \Sigma_{xx}{}^{-1}V_{xx}\Sigma_{xx}{}^{-1}\right)\Sigma_{xb} \end{pmatrix} \tag{6}$$

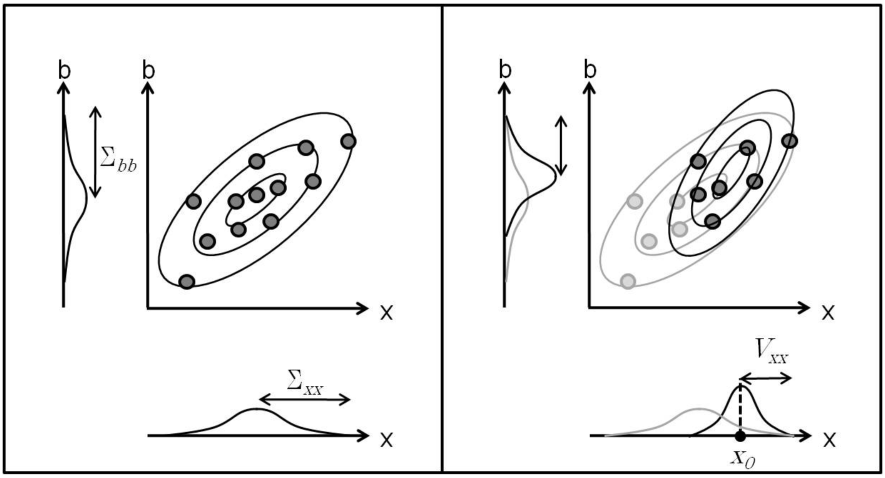


Fig. 2 By employing the selection formula, a sub-space of the data is used to construct the conditional SSM. The left hand side of this figure represents the distribution for a non-conditional SSM, the right hand side shows the selection of a limited range of the data by employing the selection formula

Figure 2 illustrates the concept of the selection formula. By selecting a range of the conditional data around a measured set of conditional features x_0, such that the new covariance matrix for the conditional features becomes V_{xx}, the covariance matrix for the shape data also changes. Instead of Σ_{bb}, the new conditional covariance matrix of b (given the condition x_0) becomes:

$$\Sigma_{bb|x_0} = \Sigma_{bb} - \Sigma_{bx}\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}V_{xx}\Sigma_{xx}^{-1}\right)\Sigma_{xb} \tag{7}$$

which corresponds with the bottom right element of (6). Comparing (7) with the regular equation (2) for the conditional covariance matrix, it can be observed that the term Σ_{xx}^{-1} is replaced by $\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}V_{xx}\Sigma_{xx}^{-1}\right)$. Analogue substitution of Σ_{xx}^{-1} by $\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}V_{xx}\Sigma_{xx}^{-1}\right)$ in (1), results in the equation for the selection formula based relaxed conditional average:

$$\mu_{b|x_0} = \mu_b + \Sigma_{bx}\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}V_{xx}\Sigma_{xx}^{-1}\right)\left(x_0 - \mu_x\right) \tag{8}$$

To allow the seamless transition between the non-conditional SSM and the conventional conditional SSM, the reliability parameters $\{\gamma_1, \gamma_2, \ldots, \gamma_F\}$ are introduced, with $0 \leqq \gamma_i \leqq 1$ and F denoting the number of conditional features. The set of reliability parameters are used to define V_{xx} as a limited range of Σ_{xx} according to (9) and (10).

$$V_{xx} = \left((I - \Gamma)^{\frac{1}{2}}\right)^T \Sigma_{xx} (I - \Gamma)^{\frac{1}{2}} \tag{9}$$

$$\textit{with} \qquad \Gamma = \begin{pmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \gamma_F \end{pmatrix} \tag{10}$$

Substitution of (9) and (10) into (7) and (8) results in the equations for the conditional average and the conditional covariance matrix of the relaxed conditional SSM:

$$\mu_{b|x_0} = \mu_b + \Sigma_{bx}\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}\left(\left((I - \Gamma)^{\frac{1}{2}}\right)^T \Sigma_{xx} (I - \Gamma)^{\frac{1}{2}}\right)\Sigma_{xx}^{-1}\right)(x_0 - \mu_x) \tag{11}$$

$$\Sigma_{bb\,|\,x_0} = \Sigma_{bb} - \Sigma_{bx}\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}\left(\left((I - \Gamma)^{\frac{1}{2}}\right)^T \Sigma_{xx} (I - \Gamma)^{\frac{1}{2}}\right)\Sigma_{xx}^{-1}\right)\Sigma_{xb} \tag{12}$$

Let's, for simplicity, assume that all reliability parameters γ_i are equal, meaning $\gamma_1 = \gamma_2 = \ldots = \gamma_F = \gamma$. When substituting $\gamma = 0$ into (11) and (12), the term $\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}\left(\left((I - \Gamma)^{\frac{1}{2}}\right)^T \Sigma_{xx} (I - \Gamma)^{\frac{1}{2}}\right)\Sigma_{xx}^{-1}\right)$ results in $\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}(\Sigma_{xx})\Sigma_{xx}^{-1}\right)$, which is zero, and hence (11) and (12) resemble the non-conditional SSM representation. With the substitution of the other extremity, $\gamma = 1$, into (11) and (12), the term $\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}\left(\left((I - \Gamma)^{\frac{1}{2}}\right)^T \Sigma_{xx} (I - \Gamma)^{\frac{1}{2}}\right)\Sigma_{xx}^{-1}\right)$ results in Σ_{xx}^{-1} and consequently (11) and (12) become identical to (1) and (2) respectively, signifying the conventional conditional SSM. Hence, with the introduction of the reliability parameters, it is possible to bridge between the conventional conditional SSM and the non-conditional SSM. Because the parameters γ_i can take any value between 0 and 1, the transition between these two models becomes continuous and seamless. Depending on the reliability of the conditional features the relaxed model behaves either more like a conditional SSM or like a non-conditional SSM. For small values of γ_i (assuming an unreliable condition), the range around the measured condition x_0 becomes large, and the model tends more toward a non-conditional SSM. For large values of γ_i (assuming a reliable condition), the range around the measured condition x_0 becomes small, and the model tends more toward a conventional conditional SSM.

Because in this approach the manipulations of Σ_{xx} are based on multiplication instead of addition (as is the case for ridge regression), the weak shape variations in Σ_{bb} are not disproportionally strongly affected by large values of the regulating term γ_i. Furthermore, the domain of the regulating term (between 0 and 1) is more practical and elegant than the domain of the ridge parameter (between 0 and infinity).

Although the mathematical design of this relaxed conditional SSM is solid, there is one flaw in terms of practicality: How to optimize the parameters γ_i? Ideally, extensive tuning of all combinations of reliability parameter values should be executed to obtain the best performing relaxed conditional SSM. However, because of the huge time requirement for such an optimization, a rudimentary optimization is performed in (Tomoshige *et al.* 2012), using the simplification $\gamma_1 = \gamma_2 = \ldots = \gamma_F = \gamma$.

8 Automatic Estimation of the Reliability of the Conditional Features

To eliminate this practical flaw, (Tomoshige *et al.* 2013) propose yet another relaxed conditional SSM, in which (during the model training phase) the relationship between the measured condition and the true condition is modeled. This way, they attempt to incorporate a priori knowledge of the reliability of the measured conditional features into the framework of the relaxed conditional SSM. To construct this so called conditional features error model, two sets of binary labeled volumes are required: the manually drawn labels of the object of interest and the calculated MAP result. For every training sample, from both these two binary volumes a set of corresponding conditional features are extracted. Identical to the conditional SSMs described above, the true label based conditional features are combined in the conditional data matrix X. The conditional features derived from the MAP estimations of the training data are stored in matrix M. Using these two matrices, a conventional conditional model is constructed, with equations for the conditional average and the conditional covariance matrix as follows:

$$\mu_{x|x_0} = \mu_x + \Sigma_{xm}\Sigma_{mm}^{-1}\left(x_0 - \mu_m\right) \tag{13}$$

$$V_{xx|x_0} = \Sigma_{xx} - \Sigma_{xm}\Sigma_{mm}^{-1}\Sigma_{mx} \tag{14}$$

with μ_x denoting the average set of true features, μ_m signifying the average set of MAP estimated features, Σ_{xx} and Σ_{mm} respectively representing the covariance matrices for X and M and Σ_{xm} and Σ_{mx} denoting the mutual covariance matrices. Note that this conditional model is a feature model and not a shape model. Figure 3 visualizes how the conditional features error model is employed to estimate the reliability of the set of measured conditional features. Starting from the extracted a set of conditional features x_0 from an unseen image, the conditional feature model calculates the expected representation of the conditional term $\mu_{x|x_0}$ and the variance $V_{xx|x_0}$ around $\mu_{x|x_0}$.

Contrary to (Tomoshige *et al.* 2012), in which the relaxation of the condition is regulated by the reliability parameters Γ, relaxation of the conditional term by using the conditional features error model is model-driven, using a priori knowledge, and does not require any parameter tuning.

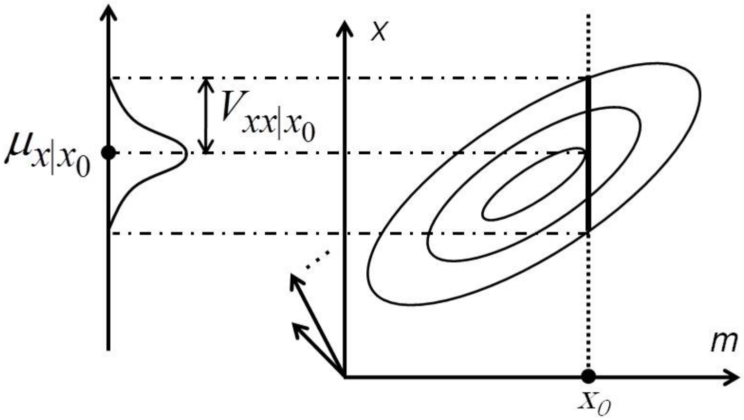


Fig. 3 Using the distribution describing the relationship between estimated condition *m* and true condition *x*, a measured condition x_0 results in an expected condition $\mu_{x|x_0}$ and accompanying covariance matrix $V_{xx|x_0}$ (Reproduced from Tomoshige *et al.* 2013)

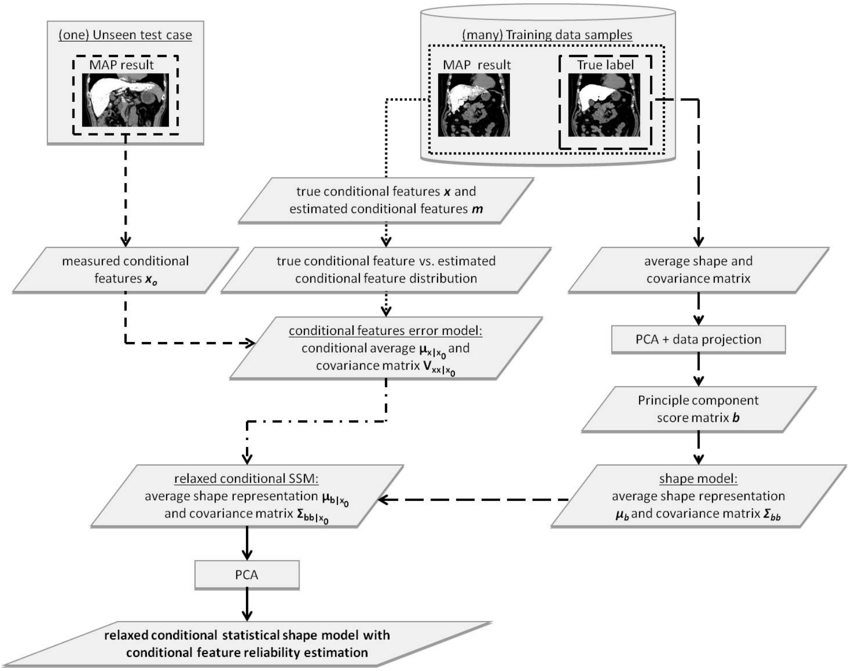

(one) Unseen test case
MAP result
(many) Training data samples
MAP result
True label
true conditional features x and estimated conditional features m
measured conditional features x_o
true conditional feature vs. estimated conditional feature distribution
average shape and covariance matrix
conditional features error model: conditional average $\mu_{x|x_0}$ and covariance matrix $V_{xx|x_0}$
PCA + data projection
Principle component score matrix b
relaxed conditional SSM: average shape representation $\mu_{b|x_0}$ and covariance matrix $\Sigma_{bb|x_0}$
shape model: average shape representation μ_b and covariance matrix Σ_{bb}
PCA
relaxed conditional statistical shape model with conditional feature reliability estimation

Fig. 4 Schematic overview of the construction of the relaxed conditional SSM with integrated conditional features error model. Note that the flow for the shape statistics is denoted by the long-dashed lines, the construction of the distribution describing the relationship between estimated condition *m* and true condition *x* is plotted with the dotted lines, and the input from the unseen test image is depicted by the dashed lines. (Reproduced from Tomoshige *et al.* 2013)

The construction and integration of the conditional features error model into the conditional SSM framework is depicted by figure 4. Integration of the conditional features error model into the framework of conditional shape modeling is straightforward. In the equation for the conventional conditional average (1), x_0 is replaced by $\mu_{x|x_0}$, resulting in:

$$\mu_{b|x_0} = \mu_b + \Sigma_{bx}\Sigma_{xx}^{-1}\left(\mu_{x|x_0} - \mu_x\right) \tag{15}$$

The formulation of the conditional covariance matrix of the relaxed conditional SSM is derived from (7), in which V_{xx} is replaced by the calculated covariance matrix of the conditional features error model $V_{xx|x_0}$, as calculated by (14):

$$\Sigma_{bb|x_0} = \Sigma_{bb} - \Sigma_{bx}\left(\Sigma_{xx}^{-1} - \Sigma_{xx}^{-1}V_{xx|x_0}\Sigma_{xx}^{-1}\right)\Sigma_{xb} \tag{16}$$

Now let's investigate the behavior of the conditional features error model and how it influences the conditional SSM. In case of, during training of the conditional features error model, a perfect correspondence between the true labels and the labels derived from the MAP estimation, all covariance matrices would be identical, and furthermore the both averages would be identical. Or in equations: $\Sigma_{xx} =\Sigma_{mm} = \Sigma_{xm} = \Sigma_{mx}$, and $\mu_x = \mu_m$. Following (13) and (14), this would lead to $\mu_{x|x_0} = x_0$ and $V_{xx|x_0} = 0$. Consequently, using these settings, (15) and (16) will become identical to (1) and (2), representing the conventional conditional SSM with a fixed condition. In other words, the extracted conditional features are completely reliable and therefore the usage of a fixed constraint is the most logical approach. The other extremity is the situation where the true labels and the MAP based labels are completely uncorrelated. Mathematically, this is denoted by $\Sigma_{xm} = \Sigma_{mx} = 0$. (13) and (14) then become $\mu_{x|x_0} = \mu_x$ and $V_{xx|x_0} = \Sigma_{xx}$ respectively. Substitution into (15) and (16) results in $\mu_{b|x_0} = \mu_b$ and $V_{bb|x_0} = \Sigma_{bb}$. Hence, the relaxed conditional SSM will behave like a non-conditional SSM. In practice, the reliability of the conditional term, and thereby the actual degree of relaxation, is automatically determined by the trained relationship between the true labels and the MAP based labels.

Summarizing, the method proposed in (Tomoshige *et al.* 2013) contains every element that a conditional statistical shape model should possess:

- ✓ integration of conditional data to improve the performance of the statistical shape model,
- ✓ relaxation of the conditional term, in case the reliability of the conditional term is not 100%,
- ✓ a framework in which the relaxed conditional SSM is defined as a seamless transition between the non-conditional SSM and the conventional conditional SSM,
- ✓ an automatic, knowledge based estimation of the reliability of the conditional term

9 Performance Comparison of Various Conditional SSMs

Following the experiments of (Tomoshige *et al.* 2012, 2013), five different level set based SSMs are compared: a non-conditional SSM (NC-SSM), a conventional conditional SSM (C-SSM), a ridge regression based conditional SSM (RC-SSM-R) (de Bruijne *et al.* 2007), a relaxed conditional SSM (RC-SSM) (Tomoshige *et al.* 2012) and a relaxed conditional SSM with integrated conditional features error model (RC-SSM-E) (Tomoshige *et al.* 2013).

For all (level set based) conditional SSMs there is one risk during construction of the model. If there are multi-colinearities between the conditional features, Σ_{xx} will become singular. To ensure matrix non-singularity, the training data set size should be substantially larger than the number of conditional features. Furthermore, it is advisory to select a set of conditional features such that the mutual correlation between features does not exceed a certain limit. (Tomoshige *et al.* 2013) ensure that the mutual correlation between features is below 0.95.

The comparison of the conditional models is based on a segmentation pipeline for non-contrast abdominal CT volumes. Liver segmentation from abdominal CT volumes is one of the most popular and challenging segmentation problems in medical image analysis. A multitude of liver segmentation papers has been published, such as (Soler *et al.* 2001), (Kainmuller *et al.* 2007), (Okada *et al.* 2007), (Ruskó *et al.* 2009) and (Heimann *et al.* 2009), among which the fully automatic method by (Kainmuller *et al.* 2007) shows the best performance. The vast majority of liver segmentation literature is based on contrast-enhanced CT volumes, which provides a relatively clear appearance of the liver. However, due to radiation dose issues, in many clinical situations only non-contrast imaging is available. Liver segmentation in such images is still challenging, especially when the liver has large pathological lesions or when the liver has an atypical shape, which is difficult to be accounted for by the SSM. The comparison of the different conditional SSMs, reported in this section, is based on a non-contrast abdominal CT data set.

The segmentation pipeline starts with a MAP estimation on the unseen image. Subsequently, conditional features x_0 are extracted from the resultant MAP volume to serve as conditional term in the conditional SSM. Furthermore the MAP result is projected onto the ((relaxed) conditional) SSM. Using Powell's method (Press *et al.* 2007), with the Jaccard Index (J.I.) as overlap measure and the model representation as the objective function, the model's shape parameters are optimized to best fit the MAP estimation. To dodge local optima, this optimization is performed in three steps, first using approximately 30% of the shape variation, then using approximately 60% of the shape variation and finally using approximately 90% of the shape variation. Finally, the optimized shape representation is used as the shape prior for a graph cut segmentation. Figure 5 presents an overview of the segmentation pipeline. Details on parameters and settings used in the graph cut segmentation are elaborately described in (Tomoshige *et al.* 2013) and (Shimizu *et al.* 2010). Note that the dashed ellipsoids in figure 5 clearly show the error propagation in this segmentation process: Inaccuracies in the estimation of the shape prior are propagated to the subsequent graph cut segmentation, leading to an unsatisfactory segmentation result. The need for an optimal shape estimation is apparent.

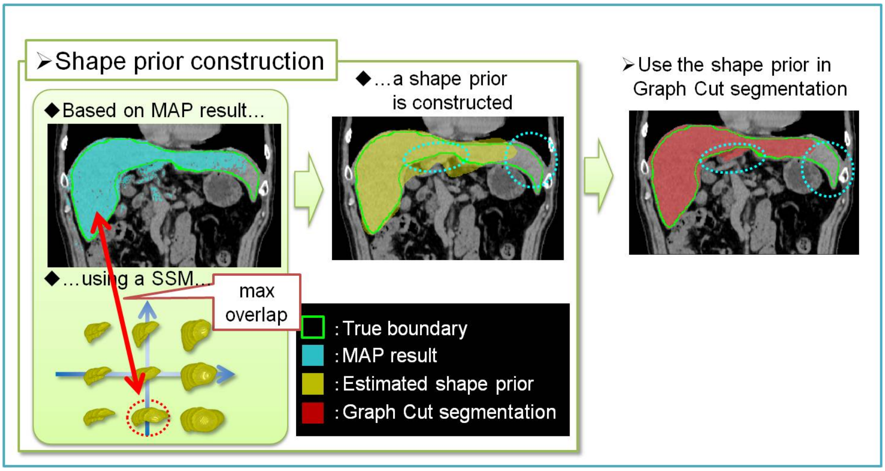


Fig. 5 Schematic overview of the entire segmentation pipeline, including MAP estimation, shape prior estimation and graph cut segmentation

Figure 6 shows the results for shape estimation and subsequent graph cut segmentation, when comparing the RC-SSM with the NC-SSM, the C-SSM and the RC-SSM-R. Note that the framework for the construction of the RC-SSM, allows to construct the NC-SSM by setting all γ_i to 0, and similarly allows to construct the C-SSM by setting all γ_i to 1. The results in figure 6 represent 20 difficult to segment abdominal CT volumes, taken from a test set of in total 48 cases. Training is performed on another set of 48 cases and a third set of 48 cases is used to optimize the graph cut parameters, the ridge regression parameter and the reliability parameters Γ.

A first striking result is the difference between the NC-SSM and the C-SSM. Mainly in terms of shape estimation, but also in terms of subsequent graph cut segmentation, the NC-SSM appears to provide much better results than the C-SSM. This is exactly the justification for relaxation of the conditional features: Due to the low reliability of the conditional features, and the hard constraint that is imposed by the C-SSM, the features deteriorate the model and effectively do more harm than good.

With the necessity of relaxation of the conditional term being established, the next step is to investigate the performance of the various relaxed conditional SSMs, by comparing the RC-SSM and the RC-SSM-R. As mentioned above, during the training phase both the ridge regression parameter and the reliability parameters are optimized, with the restriction for the RC-SSM that $\gamma_1 = \gamma_2 = \ldots = \gamma_F = \gamma$. In that respect the two approaches are treated identically in the experiments: one optimized parameter manipulates all conditional features. The ridge regression parameter ρ is optimized in the range $\rho = \{0.01; 0.1; 1; 10; 100; 1000; 10000\}$, with $\rho = 1000$ being the optimal value. Similarly, the relaxation parameter γ is optimized in the range $\gamma = \{0.0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1.0\}$, with $\gamma = 0.5$ being the optimal value. As figure 6 indicates, the RC-SSM-R does not outperform the standard

NC-SSM. It does appear to perform better than the C-SSM, which is corroborating the finding that relaxation of conditional features is essential. The RC-SSM does outperform the NC-SSM, as well as C-SSM and the RC-SSM-R. Both in terms of shape estimation and in terms of subsequent graph cut segmentation, the RC-SSM shows statistically significant improved performance, with p<0.05 when compared to the RC-SSM-R and the NC-SSM and with p<0.01 when compared to the C-SSM.

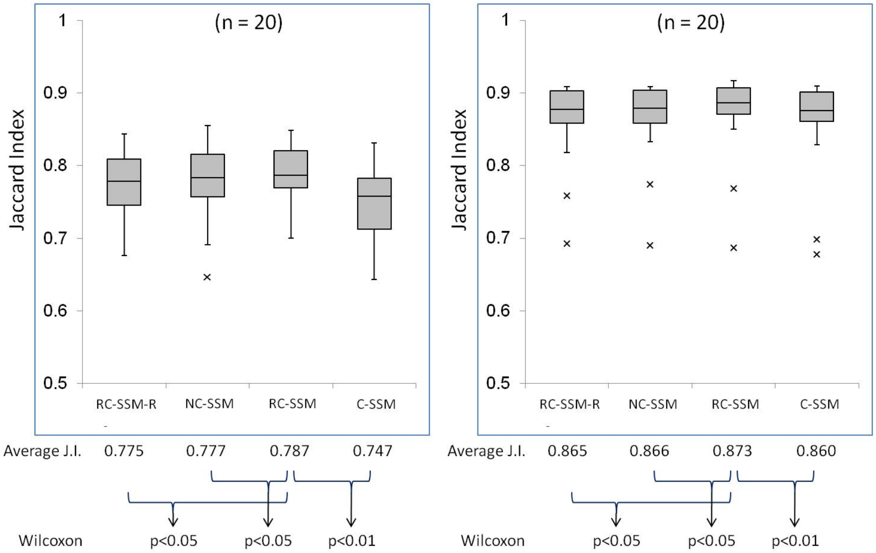


Fig. 6 Experimental results for four different (conditional) statistical shape models, with on the left hand side the shape estimation performance and on the right hand side the subsequent graph cut segmentation results (Reproduced from Tomoshige *et al.* 2012)

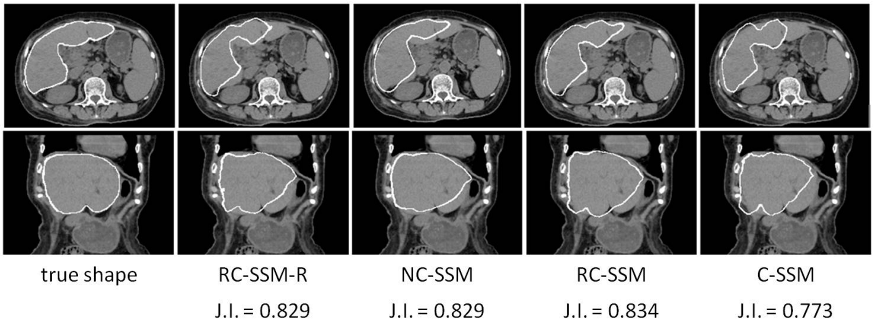


Fig. 7 Example results (shape estimation) showing the performance of four different (conditional) SSMs (Reproduced from Tomoshige *et al.* 2012)

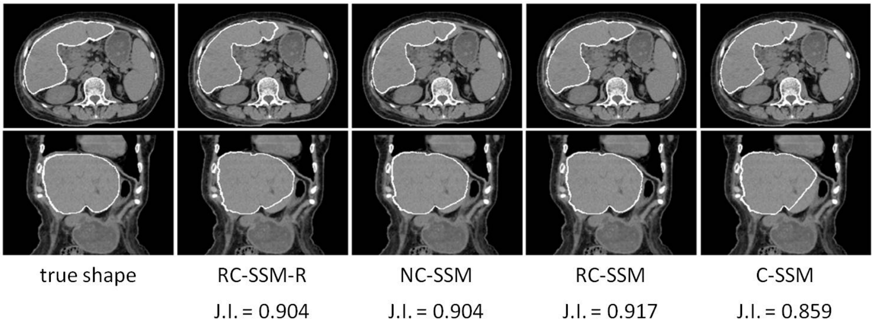


Fig. 8 Example results (graph cut segmentation) showing the performance of four different (conditional) SSMs (Reproduced from Tomoshige *et al.* 2012)

A typical example of a segmented case is displayed in figures 7 and 8, showing the shape estimation and the subsequent graph cut segmentation respectively. During shape estimation, especially the lobes of the liver are better delineated by the RC-SSM, resulting also in optimal graph cut segmentation.

For completeness, it should be mentioned that the experiments on the 28 remaining easy to segment cases also show statistically significant performance improvement in terms of shape estimation. In terms of graph cut segmentation, only the improved performance with respect to the NC-SSM was not statistically significant.

In (Tomoshige *et al.* 2013) a further evaluation of the various conditional SSMs is performed, including the RC-SSM-E, but excluding the RC-SSM-R, which did not prove to be the optimal approach for relaxation of the conditional term. Using the same data set as in (Tomoshige *et al.* 2012), now 24 cases are used for parameter optimization for the graph cut segmentation and optimization of the reliability parameters γ. The latter is now optimized at $\gamma = 0.9$. The remaining 120 cases were divided in two groups of 60 cases which were mutually evaluated through cross validation.

Regarding shape estimation, for these 120 cases, the two relaxed conditional models, RC-SSM and RC-SSM-E, do outperform the C-SSM and the NC-SSM, but do not statistically differ in their mutual performance. Nonetheless, the results for subsequent graph cut segmentation show a statistically significant difference in performance for the RC-SSM-E, outperforming the RC-SSM. It is suggested by the authors that the former approach outperforms the latter approach when estimating the shape of difficult to segment areas of the liver, such as the tip of the liver lobes. Although not expressed as a significant difference in the Jaccard Index for the shape estimation by both methods, the superior shape estimation from the RC-SSM-E is a better shape prior for the graph cut, resulting in a statistically significant final segmentation result. Average differences in Jaccard Index, although statistically significant, however remain low.

In a second experiment, only 23 difficult to segment cases are examined. To distinguish between easy to segment and difficult to segment cases, the difference between the true conditional features and the measured conditional features is analyzed. For these 23 difficult to segment cases, the RC-SSM-E outperforms the RC-SSM (as well as the C-SSM and the NC-SSM), both in shape estimation and in subsequent graph cut segmentation. All mutual differences between the two relaxed conditional models prove statistically significant, and are of considerable magnitude, with a difference of 0.012 for the shape estimation, and 0.026 for the graph cut segmentation. The most probable explanation for the significant difference in performance is that the Γ parameters are rudimentary optimized during the training phase. This means that the degree of relaxation that is used by the RC-SSM is determined during training and will remain the same, regardless of the representation of the unseen image during the testing phase. On the contrary, the RC-SSM-E tries to assess the global representation of the unseen image and uses the extracted information to determine the degree of relaxation of the conditional features. Thus, the degree of relaxation is determined during testing phase, instead of during the training phase. This way the RC-SSM-E has much more flexibility in deciding the degree of relaxation of the conditional features for the individual case. This is especially useful for difficult to segment images, as is illustrated by the results for the 23 difficult to segment cases.

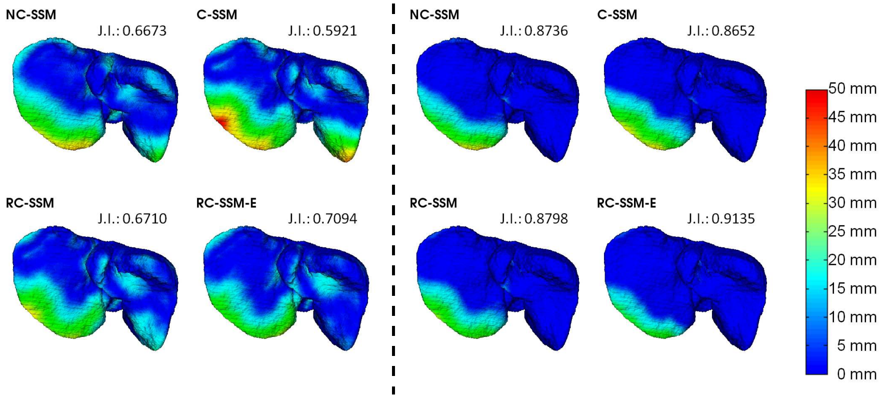


Fig. 9 Example results showing the performance of the NC-SSM, the C-SSM, the RC-SSM and the RC-SSM-E. Left from the dashed line, the shape estimation results are shown, on the right the subsequent graph cut segmentation results are shown. The performance of the RC-SSM-E is superior, both in local border delineation and in global segmentation performance, as expressed by the Jaccard Index. (Reproduced from Tomoshige *et al.* 2013)

Figure 9 presents an example result (shape estimation and graph cut segmentation), visualizing the benefit of the RC-SSM-E. It is clearly visible that, for the shape estimation, the local border delineation error at the extremities of the liver lobes is least when the RC-SSM-E is used. This superior shape estimate subsequently results in a superior graph cut segmentation. Note that for this case the C-SSM shows the worst performance. Apparently the reliability of the conditional term is low, with the consequence that applying the conditional term as a hard

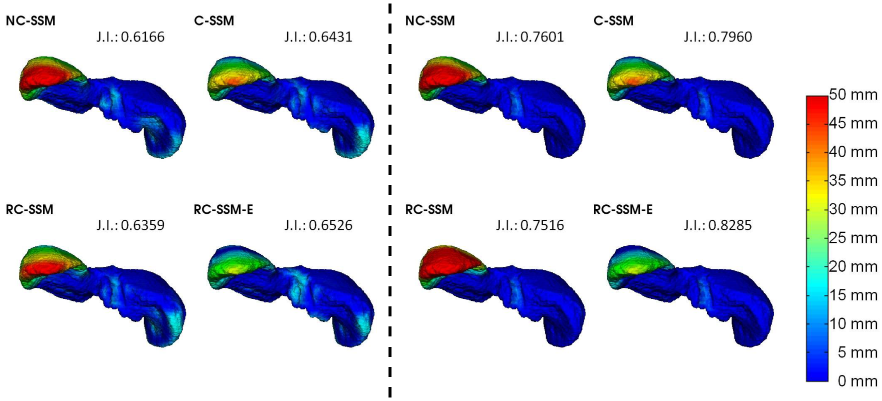


Fig. 10 Example results showing the performance of the NC-SSM, the C-SSM, the RC-SSM and the RC-SSM-E. Left from the dashed line, the shape estimation results are shown, on the right the subsequent graph cut segmentation results are shown. Both visual inspection and quantitative evaluation using the Jaccard Index, demonstrate the superior shape estimation and subsequent graph cut segmentation for the RC-SSM-E. Note that this example liver displays an atypical shape. (Reproduced from Tomoshige *et al.* 2013)

constraint is more harmful than beneficial. Results for a second example, a liver with an atypical shape, are shown in figure 10. Similar to figure 9, the RC-SSM-E provides the best shape estimation, with an obviously superior local border delineation of the liver lobes' extremities. Again this improved shape estimation results in an superior graph cut segmentation compared to the other three methods. Surprisingly, in this example the RC-SSM shows the worst performance. Probably, for this atypical shape, it is better to determine the degree of relaxation of the conditional features during the testing phase (RC-SSM-E) than during the training phase (RC-SSM).

10 Conclusions

Due to large variations in training data samples, convergence to a correct model-based shape representation for an unseen image is not always straightforward. A sub-space approach, focusing on a limited range of the training data to model the unseen shape, can solve such convergence issues, resulting in a more robust and more accurate delineation of the object of interest. One sub-space modeling approach that is growing in popularity, is the conditional statistical shape model. Given a measured condition x_0, the conditional SSM identifies a sub-space of the model training data that corresponds to this condition, and a more dedicated shape delineation can be obtained. However, the performance of the conditional SSM depends strongly on the reliability of the condition x_0. If the reliability of the condition is high, the conventional conditional SSM will perform properly. On the contrary, if the condition is completely unreliable, it is better to use a non-conditional SSM instead, because the condition will rather mislead the conditional SSM instead of directing it toward the desired segmentation. It can be concluded that relaxation of the conditional term is essential for proper shape estimation, especially when the reliability of the conditional term is low. The ideal definition of a relaxed conditional SSM is when it bridges between the non-conditional SSM and the hard constrained conventional conditional SSM, allowing a seamless transition between the two extremities. A comparison study of two of such bridging conditional SSMs shows that the RC-SSM is superior to the ridge regression based RC-SSM-R, both in mathematical formulation as well as in performance. However, both these two models are outperformed by the relaxed conditional SSM with integrated conditional features error model (RC-SSM-E). Because the latter model determines the degree of relaxation for the conditional term during the segmentation of the unseen image, instead of during the model training phase, it is much more flexible to find the optimal sub-space of data, best representing the object of interest in the unseen image. Performing an evaluation in non-contrast abdominal CT images, this approach proves particularly beneficial for difficult to segment images, showing a considerable and statistically significant improvement in terms of shape estimation and subsequent graph cut segmentation.

As defined earlier in this chapter, the optimal conditional SSM should:

- ✓ integrate conditional data to improve the performance of the statistical shape model,
- ✓ relax the conditional term, in case the reliability of the conditional term is not 100%,
- ✓ define the relaxed conditional SSM as a seamless transition between the non-conditional SSM and the conventional conditional SSM,
- ✓ contain an automatic, knowledge based estimation of the reliability of the conditional term.

Future Direction of Conditional Shape Models. With this list of requirements for the construction of conditional SSMs, the modeling and delineation of single organs appears to be properly addressed. Future challenges remain in the simultaneous modeling and delineation of multiple organs. By introducing conditional SSMs for adjacent organs, for example, the full abdomen can be modeled, while mutually employing conditional features (both shape and appearance) of neighboring organs as the conditional term to delineate the organ of interest.

References

1. Angelié, E., Oost, E.R., Hendriksen, D., Lelieveldt, B.P.F., van der Geest, R.J., Reiber, J.H.C.: Automated Contour Detection in Cardiac MRI Using Active Appearance Models: The Effect of the Composition of the Training Set. Investigative Radiology 42(10), 697–703 (2007)
2. van Assen, H.C., Danilouchkine, M.G., Behloul, F., Lamb, H.J., van der Geest, R.J., Reiber, J.H.C., Lelieveldt, B.P.F.: Cardiac LV segmentation using a 3D active shape model driven by fuzzy inference. In: Ellis, R.E., Peters, T.M. (eds.) MICCAI 2003. LNCS, vol. 2878, pp. 533–540. Springer, Heidelberg (2003)

3. Baka, N., de Bruijne, M., Reiber, J.H.C., Niessen, W., Lelieveldt, B.P.F.: Confidence of model based shape reconstruction from sparse data. In: Proceedings of ISBI 2011, pp. 1077–1080 (2010)
4. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision 70(2), 109–131 (2006)
5. de Bruijne, M., Lund, M.T., Tanko, L.B., Pettersen, P.C., Nielsen, M.: Quantitative vertebral morphometry using neighbor-conditional shape models. Medical Image Analysis 11, 503–512 (2007)
6. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - their training and application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
8. Cremers, D., Rousson, M., Deriche, R.: A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. International Journal of Computer Vision 72(2), 195–215 (2007)
9. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 755–762 (2005)
10. van Ginneken, B., Frangi, A.F., Staal, J.J., ter Haar Romeny, B.M., Viergever, M.A.: Active shape model segmentation with optimal features. IEEE Transactions on Medical Imaging 21(8), 924–933 (2002)
11. Hamarneh, G., Gustavsson, T.: Deformable spatio-temporal shape models: extending active shape models to 2D+time. Image and Vision Computing 22(6), 461–470 (2004)
12. Heimann, T., van Ginneken, B., Styner, M.A., Arzhaeva, Y., Aurich, V., Bauer, C., Beck, A., Becker, C., Beichel, R., Bekes, G., Bello, F., Binnig, G., Bischof, H., Bornik, A., Cashman, P.M.M., Chi, Y., Córdova, A., Dawant, B.M., Fidrich, M., Furst, J.D., Furukawa, D., Grenacher, L., Hornegger, J., Kainmüller, D., Kitney, R.I., Kobatake, H., Lamecker, H., Lange, T., Lee, J., Lennon, B., Li, R., Li, S., Meinzer, H.P., Németh, G., Raicu, D.S., Rau, A.M., van Rikxoort, E.M., Rousson, M., Ruskó, L., Saddi, K.A., Schmidt, G., Seghers, D., Shimizu, A., Slagmolen, P., Sorantin, E., Soza, G., Susomboon, R., Waite, J.M., Wimmer, A., Wolf, I.: Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Transactions on Medical Imaging 28(8), 1251–1265 (2009)
13. Heimann, T., Meinzer, H.P.: Statistical shape models for 3D medical image segmentation: A review. Medical Image Analysis 13, 543–563 (2009)
14. Hoerl, A., Kennard, R.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1), 55–67 (1970)
15. Kainmüller, D., Lange, T., Lamecker, H.: Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: Proc. MICCAI Workshop 3-D Segmentation Clinic: A Grand Challenge, pp. 109–116 (2007)
16. Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: IEEE CVPR, pp. 316–323 (2000)
17. Linguraru, M.G., Pura, J.A., Pamulapati, V., Summers, R.M.: Statistical 4D graphs for multi-organ abdominal segmentation from multiphase CT. Medical Image Analysis 16(4), 904–914 (2012)

18. Lord, F.M., Novick, M.R.: Statistical theories of mental test scores, pp. 146–147. Addison-Wesley Publishing Company Inc. (1968)
19. Nakagomi, K., Shimizu, A., Kobatake, H., Yakami, M., Fujimoto, K., Togashi, K.: Multi-shape graph cuts with neighbor prior constraints and its application to lung segmentation from a chest CT volume. Medical Image Analysis 17(1), 62–77 (2013)
20. Okada, T., Shimada, R., Sato, Y., Hori, M., Yokota, K., Nakamoto, M., Chen, Y.-W., Nakamura, H., Tamura, S.: Automated segmentation of the liver from 3D CT images using probabilistic atlas and multi-level statistical shape model. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 86–93. Springer, Heidelberg (2007)
21. Oost, E., Koning, G., Sonka, M., Oemrawsingh, P.V., Reiber, J.H.C., Lelieveldt, B.P.F.: Automated Contour Detection in X-Ray Left Ventricular Angiograms Using Multi-View Active Appearance Models and Dynamic Programming. IEEE Transactions on Medical Imaging 25(9), 1158–1171 (2006)
22. Oost, E., Oemrawsingh, P.V., Reiber, J.H.C., Lelieveldt, B.P.F.: Automated left ventricular delineation in x-ray angiograms: A validation study. Catheter. Cardiovasc. Interv. 73(2), 231–240 (2009)
23. Pless, R., Souvenir, R.: A survey of manifold learning for images. IPSJ Transactions on Computer Vision and Applications 1, 83–94 (2009)
24. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes, pp. 509–514. Cambridge University Press (2007)
25. Ruskó, L., Bekes, G., Fidrich, M.: Automatic segmentation of the liver from multi- and single-phase contrast-enhanced CT images. Medical Image Analysis 13, 871–882 (2009)
26. Shimizu, A., Nakagomi, K., Narihira, T., Kobatake, H., Nawano, S., Shinozaki, K., Ishizu, K., Togashi, K.: Automated segmentation of 3D CT images based on statistical atlas and graph cuts. In: International Conference on Medical Image Computing and Computer Assisted Intervention, Proceedings of Workshop on Medical Computer Vision, pp. 127–138 (2010)
27. Soler, L., Delingette, H., Malandain, G., Montagnat, J., Ayache, N., Koehl, C., Dourthe, O., Malassagne, B., Smith, M., Mutter, D., Marescaux, J.: Fully automatic anatomical pathological and functional segmentation from CT scans for hepatic surgery. Comput. Aided Surg. 6(3), 131–142 (2001)
28. Syrkina, E., Blanc, R., Szekely, G.: Propagating uncertainties in statistical model based shape prediction. In: Proceedings of SPIE, vol. 7962, p. 796240 (2011)
29. Tomoshige, S., Oost, E., Shimizu, A., Watanabe, H., Kobatake, H., Nawano, S.: Relaxed conditional statistical shape models and their application to non-contrast liver segmentation. In: Yoshida, H., Hawkes, D., Vannier, M.W. (eds.) Abdominal Imaging 2012. LNCS, vol. 7601, pp. 126–136. Springer, Heidelberg (2012)
30. Tomoshige, S., Oost, E., Shimizu, A., Watanabe, H., Nawano, S.: A conditional statistical shape model with integrated error estimation of the conditions; application to liver segmentation in non-contrast CT images. Medical Image Analysis (2013) (in press)
31. Üzümcü, M., Frangi, A.F., Sonka, M., Reiber, J.H.C., Lelieveldt, B.P.F.: ICA vs. PCA active appearance models: Application to cardiac MR segmentation. In: Ellis, R.E., Peters, T.M. (eds.) MICCAI 2003. LNCS, vol. 2878, pp. 451–458. Springer, Heidelberg (2003)

Condition Relaxation in Conditional Statistical Shape Models

List of Acronyms

ASM	Active Shape Model
CAD	Computer Aided Diagnosis
CAS	Computer Aided Surgery
C-SSM	conventional conditional SSM
ICA	Independent Component Analysis
J.I.	Jaccard Index
MAP	Maximum A Posteriori
NC-SSM	non-conditional SSM
PCA	Principle Component Analysis
PDM	Point Distribution Model
RC-SSM	relaxed conditional SSM
RC-SSM-E	relaxed conditional SSM with integrated conditional features error model
RC-SSM-R	ridge regression based conditional SSM
SSM	Statistical Shape Model

Chapter 3
Independent Component Analysis and Its Application to Classification of High-resolution Remote Sensing Images

Xiang-Yan Zeng and Yen-Wei Chen

Abstract. Independent component analysis (ICA) finds a linear representation of non-Gaussian data so that the components are statistically independent, or as independent as possible. It has been successfully applied to many problems, such as blind source separation. We apply ICA to high-resolution remote sensing images to obtain an efficient representation of color information. The three independent components are in opponent-color model by which the responses of R, G and B cones are combined in opponent fashions. This is consistent with the principle of many color systems. The interesting point is that there is no summation component that responds to the luminance channel in other transformations such as principal component analysis (PCA). The spectral independent components are then used for classification of high-resolution remote sensing images. The classification map of the independent components exhibits somewhat spatial consistency, which indicates that reduction of spectral correlation may lead to increase of spatial correlation.

1 Introduction

A critical problem in the signal processing area is finding a suitable representation of data, by means of a suitable transformation. It is important for subsequent process of data, whether it be pattern recognition or other tasks, that the data is represented in a way that facilitate the process. In this area, many statistical analysis techniques have been developed, such as principal component analysis (PCA) and independent component analysis (ICA). Historically, PCA was developed before ICA and have been widely used for the same type of problems, such as blind source separation

Xiang-Yan Zeng
Fort Valley State University, 1005 State University Dr., Fort Valley, GA 31030, USA
e-mail: zengx@fvsu.edu

Yen-Wei Chen
Ritsumeikan University, 1-1-1, NojiHigashi, Kusatus-Shi, Shiga-ken, 525-8577, Japan
e-mail: chen@is.ritsumei.ac.jp

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_3, © Springer-Verlag Berlin Heidelberg 2014

and feature extraction. The main difference between ICA and PCA is that ICA decomposes a set of observations into a set of non-gaussian and independent source signals, whereas PCA decomposes a set of observations into a set of uncorrelated signals. This distinction has the consequences of far-reaching capabilities of ICA methods relative to PCA methods.

ICA was first introduced in early 1980's in the context of artificial neural networks[13]. In mid-1990's, some highly successful algorithms were developed with impressive results and applications in signal processing and pattern recognition[2, 7, 14]. ICA uses a model that the input data or observations are linear mixtures of unknown latent variables (sources) [8, 15, 16]. The mixing coefficients are unknown. The only assumption is that the sources (with at most one exception) are non-Gaussian and they are independent of each other. To recover the sources, ICA rotates the input space so that the output are as statistically independent as possible. The typical application of ICA is blind source separation. Another category of successful applications is feature extraction, which is motivated by the results in neural sciences that suggest the principle of redundancy reduction explain some aspects of the early processing of sensory data in the brain. In this chapter, we introduce ICA and its application to feature extraction of remote sensing images .

Satellite images have been the subject of extensive research in a broad range of applications, such as planning and management of public transportation systems and environment investigation. Remote sensing images come in different types, including visible, hyperspectral and others; they differ from each other in the number and the wavelength range of band measurements in each pixel. Visible data consists of pixels composed of three color values of red, green, and blue. Hyperspectral images include up to hundreds of bands of data collected over narrow bandwidths of the electromagnetic spectrum. The spatial resolution of hyperspectral images can vary from a few to tens of meters, which means a pixel may contain different ground materials and the spectrum measured by a sensor is a *composite* or *mixed* spectrum[22,25]. Assuming each material's contribution to the mixed spectrum is proportional to its area within the pixel, a linear mixture model is suitable for describing the composite spectra. ICA has been successfully applied to hyperspectral data for demixing the spectra and finding the abundance fraction of the materials within each pixel. Several reports have covered unsupervised classification of hyperspectral images by ICA [5, 10, 24, 27]. In the meantime, however, the application of ICA to Red-Green-Blue (RGB) color images typically available in high-resolution remote sensing has received much less attention.

Color information has long been used for classification of remote sensing images [11,19]. Finding efficient color representation is important for classification. There are many transformations that convert an RGB color space into a new color space [26], but these are general transformations independent of actual images. A more reasonable way is to find the color encoding information by analyzing the target images. Over the last few decades, substantial research has been done on the application of statistical methods to color information analysis. Buchsbaum et al. has conducted a systematic analysis of the role of opponent type processing in color vision and noticed that efficient information transmission is achieved by a transformation of the three color mechanisms into an achromatic and two opponent chromatic channels[4]. Ruderman et al. analyzed hyperspectral images of natural scenes and had similar findings [20]. Given data represented in a logarithmic response space, an orthogonal decorrelation produced three principal axes, one corresponding to changes in radiance and the other two representing blue-yellow and red-green chromatic-opponent mechanisms. Ruderman et al. also pointed out that the chromatic mechanisms are not uniquely defined and depend on experimental data. The basic idea behind these color analysis approaches coincides with Barlow's theory that the goal of vision information processing is to reduce redundancy between input signals [1].

In this chapter, we use ICA to analyze color encoding information of high-resolution remote sensing images, namely IKONOS multispectral images. The image has three bands but in many cases contains more than three material categories. Therefor, the independent components cannot directly represent the classes. To perform classification, we use the k-means algorithm to cluster the independent components of RGB data. Typical methods of classification of remote sensing imagery include pixel-based and region-based. Pixel-based classification results commonly exhibit "salt and pepper"appearance. Region-based methods have been proposed to overcome this problem and achieve better spatial consistency. In these approaches, region splitting and region growth are implemented by a homogeneity criterion [17,21]. We here use a simple technique to achieve spatial consistency. A penalty term is added to the clustering algorithm to incorporate the spatial consistency into the classification. The incorporation of spatial consistency tends to produce smoother classification maps.

This chapter is organized as follows. Section 2 introduces the background of ICA, including the data model and two algorithms. Section 3 presents the application of ICA to classification of remote sensing images. Section 4 concludes with brief remarks.

2 Background of Independent Component Analysis

2.1 Linear Transformation of Multivariate Data

In signal processing, pattern recognition and other related areas, it is important to transform data from one space to another space so that its essential structure becomes more accessible. Linear transformations are simple and sufficient in most cases.

Let us assume that the data consists of a number of variables $x_1, x_2, \ldots, x_m$ that are observed. Further we denote the number of observations by P. We can then denote the data by $x_i(t)$ where $i = 1, 2, \ldots, m$ and $t = 1, 2, \ldots, P$. We transform the data from an m-dimensional space to an n-dimensional space, and denote the transformed data by $y_j(t), j = 1,2,\ldots,n$. For a linear transformation, each component of the transformed data is a weighted combination of the observed variables:

$$y_i(t) = \sum_{j=1}^{m} w_{ij} x_j(t), \qquad i = 1,2,\ldots,n \tag{1}$$

where w_{ij} is the weight of the jth variable in the ith component. The equation can be written as a matrix multiplication:

$$\begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix} = \mathbf{W} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_m(t) \end{pmatrix} \tag{2}$$

The $n \times m$ matrix **W** defines a linear transformation. it is desirable for the linear transformation to produce a set of components that correspond to some physical causes otherwise hidden in the original data variables. Therefore, one could determine the matrix **W** by the expected statistical properties of the transformed components y_i, such as being uncorrelated or independent.

2.2 Blind Source Separation

To better understand why we need linear transformation, or why we are not satisfied with the observed data, let us look at the same problem from a different viewpoint. Blind source separation (BSS) is a good example to illustrate the necessity of an appropriate transformation. BSS is the separation of a set of source signals from a set of mixed signals, without information about the source signals and the mixing process.

A typical example of BSS is *cocktail party problem*, where a number of people are talking simultaneously in a room, and one is trying to follow the speech of individual speakers. The same number of microphones are placed in various locations to record the signals, which turn out to be different mixtures of the same sources (speakers' conversations). The BSS problem is to recover the conversations from the recorded mixed signals. The unknown mixing process is a linear operation and depends on the microphone locations. To simplify the problem, we assume the number of mixtures is the same as the number of sources and hereafter hold this assumption in the discussions.

The process of mixing n source signals $s_1(t), s_2(t), \ldots, s_n(t)$ can be represented by a linear equation:

$$x_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + \ldots + a_{in}s_n(t), \qquad i = 1,2,\ldots,n \tag{3}$$

where $a_{i1}, a_{i2}, \ldots, a_{in}$ are mixing coefficients and $x_i(t)$ is the ith recorded signal. In this set of linear equations, the sources and the mixing coefficients are unknown.

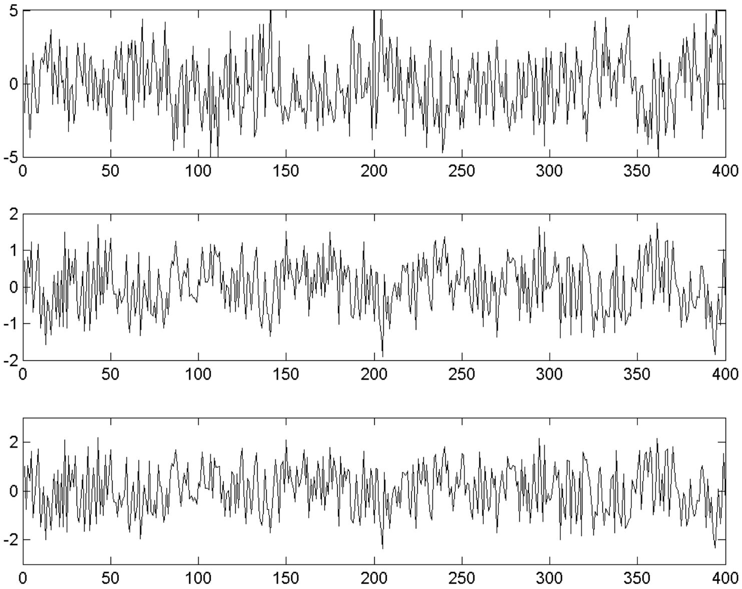


Fig. 1 The observed signals that are mixtures of the underlying source signals $s_1(t)$, $s_2(t)$, and $s_3(t)$

An example is given as follows. The three sources are synthetic signals

$$\begin{pmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{pmatrix} = \begin{pmatrix} sin(0.1t)cos(0.2t) \\ random() \\ sin(2t) + cos(3t) \end{pmatrix}$$

where *random*() is a noise source with a uniform distribution. The source signals are mixed by a unknown 3 × 3 matrix and the mixtures are shown in Fig. 1. The observed signals appear to be pure noise and make it difficult to perform further operations such as pattern recognition . Although the structured source signals are barely detectable, they are hidden in the observed signals and can be fully or partially recovered.

The BSS problem is to recover the structured source signals from the observed signals, which is in general highly underdetermined due to the limited information about the mixing matrix. To narrow down the possible solutions, it is essential to make assumptions or impose constraints on the source signals. Example approaches are principal and independent component analysis, where one seeks source signals that are minimally correlated or maximally independent in a probabilistic or information-theoretic sense.

2.3 Independent Components Analysis

2.3.1 Data Model

ICA is a computational method for solving the BSS problem. It is based on the assumption that the source signals are mutually independent. The following assumptions are typically made: (1) the source signals are statistically independent or nearly independent; (2) at most one signal has a Gaussian distribution. The ICA data model is shown in Fig. 2 and defined as follows.

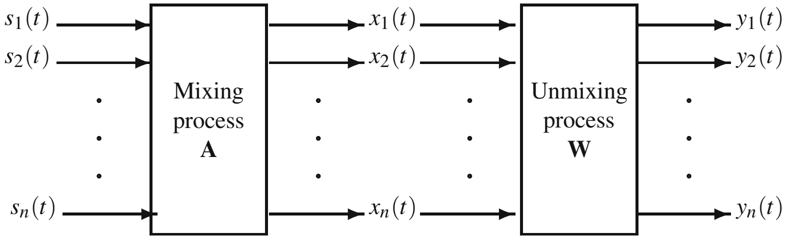


Fig. 2 The mixing and unmixing processes in independent component analysis

Assume that there exist unknown source signals $s_i(t), i = 1,2,\ldots,n$, which have zero mean and are mutually independent. The sources are mixed by an unknown $n \times n$ matrix, and the mixing process can be described by a matrix multiplication:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{4}$$

where $\mathbf{s} = (\, s_1(t),\, s_2(t),\, \ldots,\, s_n(t)\,)^T$ is the source, and $\mathbf{x} = (\, x_1(t),\, x_2(t),\, \ldots,\, x_n(t)\,)^T$ is the observation, and $\mathbf{A} = (\, \mathbf{a}_1,\, \mathbf{a}_2,\, \ldots,\, \mathbf{a}_n\,)$ is the mixing matrix. T stands for the transpose of a matrix. The column vectors $\mathbf{a}_i$ are called basis functions.

The above equation can be rewritten to describe two different types of applications. For the BSS problem, each observation is represented as a linear combination of the n sources

$$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t), \qquad i = 1,2,\ldots,n \tag{5}$$

where a_{ij} is the weight of the jth source in the ith observation. In the meantime, the feature extraction problem in many cases focuses on finding the basis functions that encode the statistical characteristics of data. The equation is rewritten so that $\mathbf{x}$ can be represented in terms of a linear superposition of basis functions:

$$\mathbf{x}(t) = \mathbf{a}_1 \times s_1(t) + \ldots + \mathbf{a}_N \times s_N(t) \tag{6}$$

The goal of unmixing is to obtain a set of variables that are statistically as independent as possible. Therefore, the linear analysis process is to find a matrix W to separate the mixed signals in $\mathbf{x}$

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{7}$$

where $\mathbf{y} = (y_1(t), y_2(t), \ldots, y_n(t))^T$ is an estimate of $\mathbf{s}$ in the sense that each component of $\mathbf{y}$ resembles a component of $\mathbf{s}$, possibly permuted and rescaled. The rows of the matrix $\mathbf{W}$ are called ICA filters and are used for linear transformation of data.

2.3.2 Why ICA?

Many techniques can find uncorrelated sources for the blind source separation problem. For instance, PCA is a widely used technique in data analysis. It is a linear transformation that removes the correlation among the elements of a random vector and concentrates the variances in a few components. An example is shown in Fig. 3, which demonstrates two-dimensional correlated data with coordinates x_1 and x_2 . It can be seen that x_1 and x_2 are related, which here means that if we know x_1 we can make a reasonable predication of x_2 and vice versa. If we rotate the axes by $\pi/4$, we get a new space with coordinates of e_1 and e_2 in which data are uncorrelated. Another property of this rotation is that the variance of the transformed data is maximized along the e_1 axis. The matrix that decorrelates the components is constructed in a way that its columns are the eigenvectors of the covariance matrix of the data.

However, producing uncorrelated components may be not enough for applications like blind source separation. Let us illustrate this with an example using two independent variables. Fig.4 shows the sources, the mixtures, and the components recovered by PCA and ICA. The data are plotted using one variable as the x-axis coordinate and another one as the y-axis coordinate. The two independent sources have a uniform distribution with a range of [-1,1]. The data is uniformly distributed in a square due to the independence of the sources. The two sources are linearly mixed. PCA rotates the plane but cannot recover the independent sources. ICA recovers the sources with a different scale. This example suggests that blind source separation require more than decorrelation. In addition, it is fair to say that ICA may recover some information that PCA cannot in the case of feature extraction.

2.4 ICA Algorithms

A large number of ICA algorithms have been proposed from different perspectives. In this section, we introduce two algorithms that are used in a wide range of applications and also in our study of remotely sensed image data.

2.4.1 Whitening the Data

The first step in many ICA algorithms is to whiten (or sphere) the data [3]. The observed data $\hat{\mathbf{x}}$ is sphered by first subtracting the mean $\mathbf{m}_x$

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{m}_x \tag{8}$$

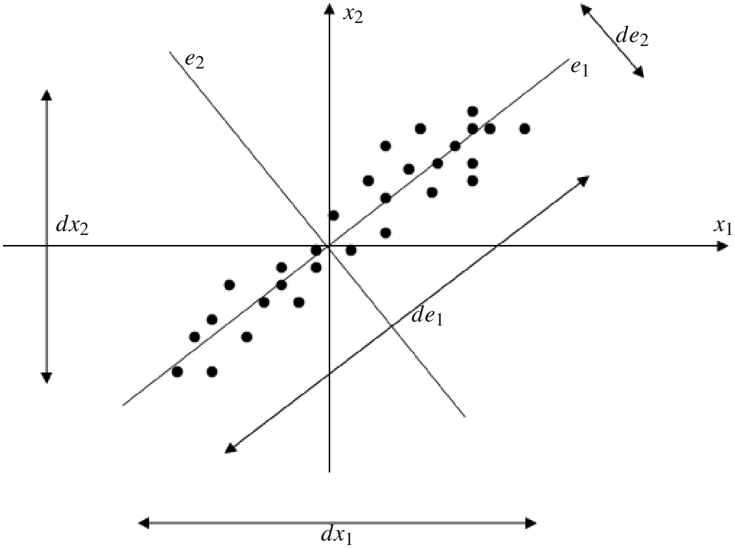
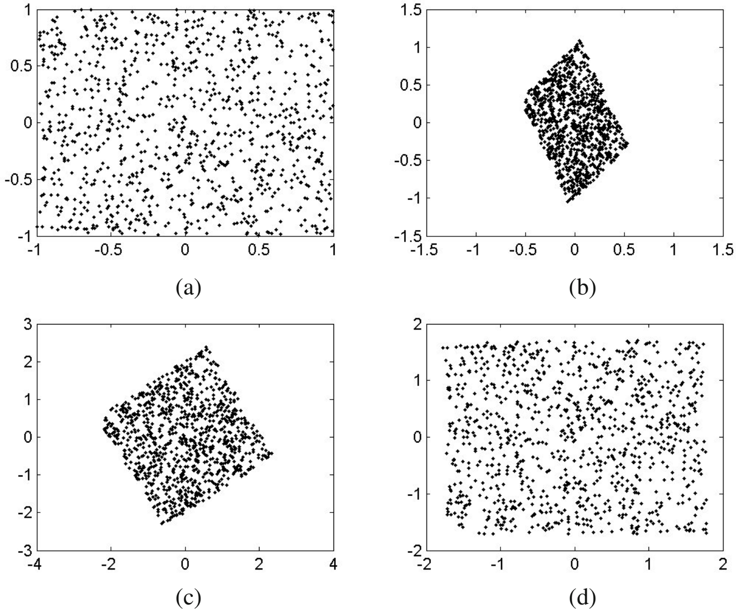


Fig. 3 Two correlated data components are decorrelated by PCA



(a)

(b)

(c)

(d)

Fig. 4 PCA and ICA in blind source separation. (a) two independent sources , (b) two mixtures, (c) sources recovered by PCA, (d)sources recovered by ICA.

and then multiplying by a whitening matrix

$$\mathbf{x} = \mathbf{W}_0\tilde{\mathbf{x}} \tag{9}$$

so that its components x_i are mutually uncorrelated and all have unit variance, i.e. the covariance matrix of $\mathbf{x}$ is an identity matrix.

$$E[\mathbf{x}\mathbf{x}^T] = \mathbf{I} \tag{10}$$

There are many solutions for the whitening matrix. One is the ZCA-whitening defined by

$$\mathbf{W}_z = \left(E\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right]\right)^{-\frac{1}{2}} \tag{11}$$

The whitening matrix $\mathbf{W}_z$ is symmetrical, which gives the name zero-phase component analysis(ZCA). The rows of $\mathbf{w}_z$ are not orthogonal. Another common whitening solution is PCA, which in the meantime can reduce dimensions if necessary. The PCA solution comes from the eigenvalue decomposition of the data covariance matrix

$$\mathbf{E}\mathbf{D}\mathbf{E}^{-1} = E\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] \tag{12}$$

where $\mathbf{D}$ is the diagonal matrix of eigenvalues, and $\mathbf{E}$ is the orthogonal matrix of eigenvectors of the covariance matrix. The PCA whitening matrix is given by

$$\mathbf{W}_P = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T \tag{13}$$

The rows of $\mathbf{W}_P$ are orthogonal, because $\mathbf{W}_P\mathbf{W}_P^T = \mathbf{D}^{-1}$ is a scale matrix. Whitening is a standard preprocessing step for the ICA algorithms discussed below.

2.4.2 ICA by Information Maximization

Bell & Sejnowski have proposed a neural learning algorithm for ICA[2]. The approach is to maximize the mutual information between input and output by a stochastic gradient ascent learning rule. The motivation behind this approach is that the neural network that maximizes information trasnfer can also reduce redundancy among the output units. Consider a neural network with an input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, a weight matrix $\mathbf{w}$, and an output vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$. The goal here is to find the weight matrix $\mathbf{w}$ that can maximally transfer information from input to output, which means to maximize the mutual information between input and output:

$$I(\mathbf{y};\mathbf{x}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}) \tag{14}$$

where $H(\mathbf{y})$ is the entropy of the outputs, $H(\mathbf{y}|\mathbf{x})$ is the conditional entropy and does not depend on the matrix $\mathbf{w}$. The partial derivative of the mutual information with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}} I(\mathbf{y};\, \mathbf{x}) = \frac{\partial}{\partial \mathbf{w}} H(\mathbf{y}) \tag{15}$$

Therefore, one can maximize the mutual information by maximizing the joint entropy of the outputs alone.

Before we obtain the learning rule for the matrix **w**, we first consider the case of one input x and one output y. The entropy is given by

$$H(y) = -E[ln(f_y(y)] = -\int_{-\infty}^{\infty} f_y(y) ln f_y(y) dy \tag{16}$$

where E is the expected value operator, and $f_y(y)$ is the probability density function (pdf) of the output.

When the transforming function of the neural network is monotonically increasing or decreasing (ie. has a unique inverse), the pdf $f_y(y)$ can be written as a function of the pdf of the input $f_x(x)$:

$$f_y(y) = \frac{f_x(x)}{|\partial y/\partial x|} \tag{17}$$

where the bars denote absolute values. Substituting $f_y(y)$ into the entropy yields:

$$H(y) = E\left[ln\left|\frac{\partial y}{\partial x}\right|\right] - E[ln f_x(x)] \tag{18}$$

the second term on the right side doesn't depend on w. Therefore, w that maximizes $H(y)$ can be obtained by a stochastic gradient ascent learning rule:

$$\triangle w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) \tag{19}$$

In the case of a sigmoid transfer function:

$$y = g(x) = \frac{1}{1+e^{-wx}} \tag{20}$$

The partial derivatives evaulate as follows:

$$\frac{\partial y}{\partial x} = wy(1-y) \tag{21}$$

$$\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) = y(1-y)(1+wx(1-2y)) \tag{22}$$

Therefore, the stochastic gradient ascent rule becomes:

$$\triangle w \propto \frac{1}{w} + x(1-2y) \tag{23}$$

The above derivations can be extended to a neural network with an input vector **x** and an output vector **y**. The pdf of a single variable is replaced by a joint pdf of a random vector. The joint pdf of the output is:

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{f_{\mathbf{x}}(\mathbf{x})}{|J|} \tag{24}$$

where $|J|$ is the absolve value of the Jacobian of the transformation, and $f_{\mathbf{y}}(\mathbf{y})$ is a simplified notation for the join pdf $f_{y_1,y_2,\ldots,y_n}(y_1,y_2,\cdots,y_n)$. The Jacobian is the determinant of the matrix of partial derivatives

$$J = det \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{vmatrix} \tag{25}$$

For a sigmod neural network, the resulting learning rule for **w** is similar in the form:

$$\Delta\mathbf{w} \propto [\mathbf{w}^T]^{-1} + (\mathbf{I} - 2\mathbf{y})\mathbf{x}^T \tag{26}$$

where **I** is an $n \times n$ identity matrix. A natural gradient is utilized to improve the convergence properties of the gradient ascent learning:

$$\Delta\mathbf{w} \propto \frac{\partial H(\mathbf{w})}{\partial \mathbf{w}} \mathbf{w}^T \mathbf{w} = \left(\mathbf{I} + f(\mathbf{y})\,\mathbf{y}^T\right)\mathbf{w} \tag{27}$$

where $\mathbf{y} = \mathbf{wx}$, and $f(y) = 1 - 2/(1 + e^{-y})$ is calculated for each component of **y**.

2.4.3 ICA by Maximization of Non-gaussianity

FastICA is an efficient ICA algorithm invented by Aapo Hyvarinen [14]. The algorithm is based on a fixed point iteration scheme maximizing non-gaussianity as a measure of statistical independence.

The natural objective of ICA is to minimize the mutual information among the output signals. Consider a simple case of two outputs y_1 and y_2, the mutual information is given by

$$I(y_1;y_2) = H(y_1) - H(y_1|y_2) = H(y_1) + H(y_2) - H(y_1,y_2) \tag{28}$$

where H denotes the entropy. To minimize the mutual information, one can maximize the joint entropy or minimize the summation of the marginal entropies. The information maximization algorithm coincides with the first consideration. Minimizing the marginal entropies leads the ICA mode to non-gaussianity, because the gaussian distribution has maximum entropy among all real-valued distributions with specified mean and standard deviation.

The classical measure of non-gaussianity is the absolute value of kurtosis. For a random variable y, the kurtosis is defined by

$$kurt(y) = E[y^4] - 3(E[y^2])^2 \tag{29}$$

If y is a gaussian variable, its kurtosis is zero. Sub-gaussian varaibles have negative kurtosis and a flat pdf. Super-gaussian variables have positive kurtosis and a"spiky" pdf with heavy tails.

To illustrate in a simple example how independent components could be found by kurtosis minimization or maximization, let us look at one output $y = \mathbf{w}^T\mathbf{x}$. ICA attempts to produce outputs that are as non-gaussian as possible, which means to minimize or maximize the kurtosis

$$kurt(\mathbf{w}^T\mathbf{x}) = E[(\mathbf{w}^T\mathbf{x})^4] - 3(E[(\mathbf{w}^T\mathbf{x})^2])^2 \tag{30}$$

The maximization or minimization of kurtosis is meaningful only if the norm of $\mathbf{w}$ is bounded, let us assume $\|\mathbf{w}\| = 1$. This can be easily done by dividing $\mathbf{w}$ by its norm. The gradient of the absolute value of kurtosis can be computed as

$$\frac{\partial |kurt(\mathbf{w}^T\mathbf{x})|}{\partial \mathbf{w}} = 4sign(kurt(\mathbf{w}^T\mathbf{x}))(E[\mathbf{x}(\mathbf{w}^T\mathbf{x})^3] - 3\mathbf{w}\|\mathbf{w}\|^2) \tag{31}$$

which suggests

$$\mathbf{w} \propto E[\mathbf{x}(\mathbf{w}^T\mathbf{x})^3] - 3\mathbf{w}\|\mathbf{w}\|^2 \tag{32}$$

The convengence of gradient descent or ascent algorithms can be slow and highly depends on the choice of the learning rate. The fixed point algorithms are alternatives with faster and more reliable convergence. A fixed point algorithm updates $\mathbf{w}$ by

$$\mathbf{w} \leftarrow E[\mathbf{x}(\mathbf{w}^T\mathbf{x})^3] - 3\mathbf{w} \tag{33}$$

where $\|\mathbf{w}\|$ is omitted because the norm is 1.

Measuring Non-gaussianity by kurtosis generates a simple ICA algorithm. However, kurtosis has its drawback of being sensitive to outlier data. This becomes a problem in practice when the value has to be estimated from sample data. Erroneous or irrelevant observations may severely affect the kurtosis calculation in some cases. Negentropy is an alternative for measuring non-gaussianity. For a random variable y , the negentropy is defined as

$$J(y) = H(y_{gaussian}) - H(y) \tag{34}$$

where $y_{gaussian}$ is a gaussian random variable of the same mean and variance as y. Due to the fact that the gaussian distribution has maximum entropy, negentropy is always nonnegative, and it is zero if and only if s is a gaussian variable. Negentropy has reliable statistical performance, but the calculation is difficult. We can approximate negentropy using higher-order cumulants:

$$J(y) \approx \frac{1}{12}(E[y^3])^2 + \frac{1}{48}(kurt(y))^2 \tag{35}$$

When the random variable y has an approximately symmetric distribution, the right side is equivalent to the square of kurtosis. Maximization of this approximation is equivalent to maximization of the absolute value of kurtosis. This approximation suffers from the same problems that kurtosis has. Therefore, more general nonquadratic functions are used to replace the polynomial functions y^3 and y^4 in the high-order cumulant approximation. A new approximation is obtained as

$$J(y) \propto (E[G(y)] - E[G(v)])^2 \tag{36}$$

where G is a nonquadratic function and v is a gaussian variable of zero mean and unit variance. A fixed point algorithm that maximizes the negentropy approximation is developed, which iteratively generates new $\mathbf{w}$

$$\mathbf{w} \leftarrow E[\mathbf{x}g(\mathbf{w}^T\mathbf{x})] - E[g'(\mathbf{w}^T x)]\mathbf{w} \tag{37}$$

where the function g is the derivative of the function G. The performance of the algorithm depends on the choice of the function G, or rather the function g. The following two functions have proved useful:

$$g_1(y) = tanh(a_1 y) \tag{38}$$

$$g_2(y) = ye^{(-y^2/2)} \tag{39}$$

where $1 \leq a_1 \leq 2$ is a constant. It is noticed that maximization of negentropy could lead to the same algorithm as maximization of the absolute value of kurtosis if we use the function

$$g_3(y) = y^3 \tag{40}$$

Therefore, eq. (37) is a general algorithm for maximizing non-gaussianity.

The iterative fixed point algorithm for obtaining one independent component constitutes of the following steps:

1. Take a random initial vector $\mathbf{w}(0)$ of norm 1. Let k=1.
2. Let $\mathbf{w}(k) \leftarrow E[\mathbf{x}g(\mathbf{w}^T(k-1)\mathbf{x})] - E[g'(\mathbf{w}^T(k-1)\mathbf{x})]\mathbf{w}(k-1)$
3. Let $\mathbf{w}(k) = \mathbf{w}(k)/\|\mathbf{w}(k)\|$.
4. if $|\mathbf{w}^T(k)\mathbf{w}(k-1)|$ is not close enough to 1, let $k = k+1$, go back to step 2.

The convergence of this algorithm means the dot product of the old and new $\mathbf{w}$ is close to one when the two vectors are almost the same.

The above algorithm is called “one-unit” algorithm, because it only estimate one independent component. To estimate n independent components, we can run the algorithm n times using different initial vectors. But this is not a reliable way to obtain distinct independent components. To ensure that a different independent component is estimated in each run, we need to add a constraint that the vectors $\mathbf{w}_i$ corresponding to the independent components are orthogonal. Assume the independent components are sequentially estimated one by one, when we estimate the p-th vector $\mathbf{w}_p$, we add a orthogonalization operation in step 3 before the normalization:

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{i=1}^{p-1} (\mathbf{w}_p^T \mathbf{w}_i)\mathbf{w}_i \tag{41}$$

3 ICA for Remote Sensing Study

3.1 ICA for Hyperspectral Remote Sensing

ICA has been successfully applied to remote sensing image classification in the past decade. Much of the work has been done with the hyperspectral image data, which have high spectral resolution but relatively low spatial resolution. An example of this is when the pixel size is comparable or larger than the natural size of ground object units. In these cases, pixels in the hyperspectral image are “mixed” pixels, because the area in a pixel may contain different materials and objects. The mixing process is described by a linear mixture model. Let n be the number of spectral bands, and $\mathbf{x}$ be the spectrum vector of a pixel. The element x_i is the reflectance in the ith wavelength band. Assume there are m types of ground materials, and the reflectance of a pixel is a linear mixture of all the different materials in that pixel. Let $\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \ldots, \mu_{in})^T$ denote the ith material spectral signature (referred to as an endmember in the linear mixture model [22]), and the unit vector $\mathbf{f} = (f_1, f_2, \ldots, f_m)^T$ denote the proportions of each ground category in the pixel. The spectrum vector of a pixel is given by

$$\mathbf{x} = f_1\boldsymbol{\mu}_1 + f_2\boldsymbol{\mu}_2 + \ldots + f_m\boldsymbol{\mu}_m = \mathbf{Mf} \tag{42}$$

$$\sum_{i=1}^{m} f_i = 1 \tag{43}$$

where $\mathbf{M} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_m)$ is an $n \times m$ matrix containing the spectral signatures as the columns. In the presence of noise, the spectrum vector of a pixel becomes

$$\mathbf{x} = \mathbf{Mf} + \mathbf{e} \tag{44}$$

where $\mathbf{e}$ is a noise vector. The spectrum unmixing analysis is to estimate the proportion vector $\mathbf{f}$ for each pixel, which also leads to the image classification results. The classification task here is not to seek a single map of material categories, but a series of images, each giving a map of the concentration of a different ground material across the scene.

The linear mixture model fits the ICA data model in eq.(6), where $\mathbf{f}$ corresponds to the source $\mathbf{s}$, $\boldsymbol{\mu}_i$ corresponds to the basis function $\mathbf{a}_i$. As a special case of this model, when the resolution is high and/or the pixel size is small, the vector $\mathbf{f}$ is sparse and there is only one dominant class for each pixel. In this case, a single class map is sufficient for representing the classification result.

One problem we need to address is that the number of ground material categories is not, in most cases, same as the number of bands. The hyperspectral image data provides sufficient spectral resolution so that there are more bands than ground categories ($n > m$). A dimension reduction technique, such as PCA, is generally used in these cases. There are very few reports about how to use the linear mixture model when $n < m$.

3.2 ICA for High-resolution Remote Sensing

In this study, we explore the application of ICA to classification of IKONOS high-resolution satellite images with three bands of red, green, and blue. In many instances a remotely sensed scene may include more than three ground categories, which means applying ICA in the same way as the above unmixing analysis is difficult. In the meantime, the pixels in the image are less "mixed" due to the high spatial resolution; image classification that assigns each pixel to a single category yields a fair estimate of the ground truth. In the following, we use ICA to transform the data into independent components and classify the transformed data using the k-means method.

3.2.1 Independent Components of RGB Remote Sensing Images

The three band IKONOS images can be assigned respectively to the R, G, B colors for display. In this way, the image is represented in a RGB color space. There are other color spaces that are suitable for specific purposes. For example, YIQ is the color space used by the NTSC color TV system. Color space conversion is the translation of the representation of a color from one basis to another; the goal is to make the translated image suitable for data processing or graphical display. The

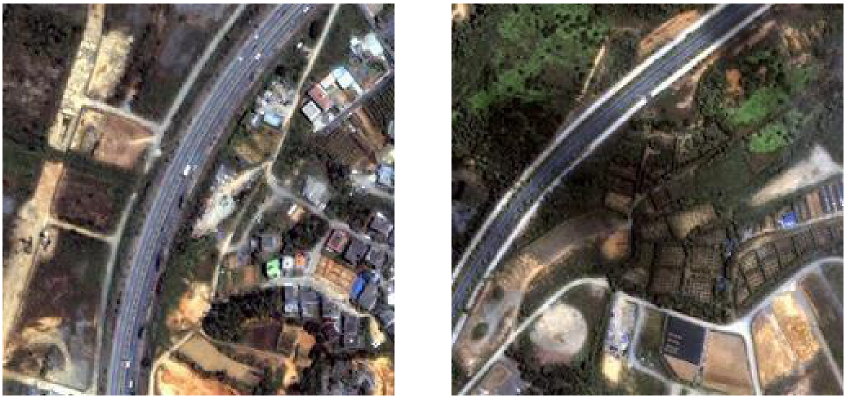


image1 image2

Fig. 5 RGB remotely sensed images Copyright(C) Japan Space Imaging Corporation

conversion in many cases is implemented by a linear transformation. For instance, the following formula approximate the conversion from a RGB color space to YIQ:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5957 & -0.2746 & -0.3213 \\ 0.2114 & -0.5226 & 0.3111 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

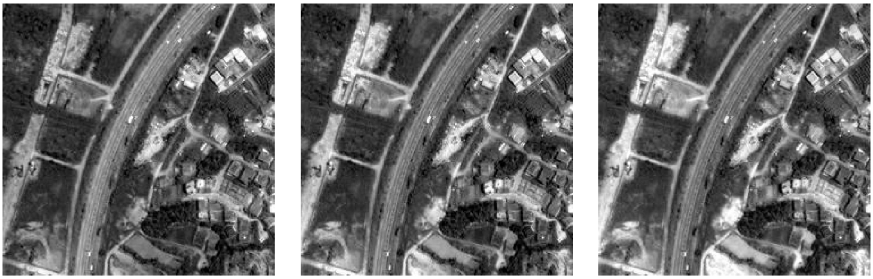
There are many transformations that convert an RGB color space into a new color space, but these are general transformations independent of the actual images. There has been a growing interest in the application of statistical methods for color data analysis. ICA has been used to reveal the structure of color information in natural images[18,23]. Tailor et al. applied ICA to color images of natural scenes. The resulting ICA filters are either luminance or color filters. The color filters resemble either blue-yellow or red-green double-opponent receptive fields.

We here applied ICA to high-resolution remote sensing images. In our experiment, the training data set consisted of 10000 samples, with 5000 samples randomly selected from each of the images shown in Fig. 5. The two images were taken under similar conditions containing similar ground objects. As an example, The three band images of image1 are shown in Fig. 6, which exhibit strong correlations between the three bands. The histograms of the three bands are shown in Fig. 7, which also have a similar range and shape.

A sample pixel had 3 values of R, G, B. We focused on the color information and didn't use any spatial information. However, the later classification results suggested that the independent components include some spatial relationship information, because the neighboring pixels were more consistently classified than in the RGB space.

The fastICA algorithm was applied to the RGB sample data, and the resulting transformation matrix is

$$\mathbf{W}_{ICA} = \begin{bmatrix} -0.0767 & 0.1325 & -0.0564 \\ 0.0359 & 0.0079 & -0.041 \\ 0.0106 & -0.0075 & -0.0225 \end{bmatrix}$$



(red) (green) (blue)

Fig. 6 The three band images of image1

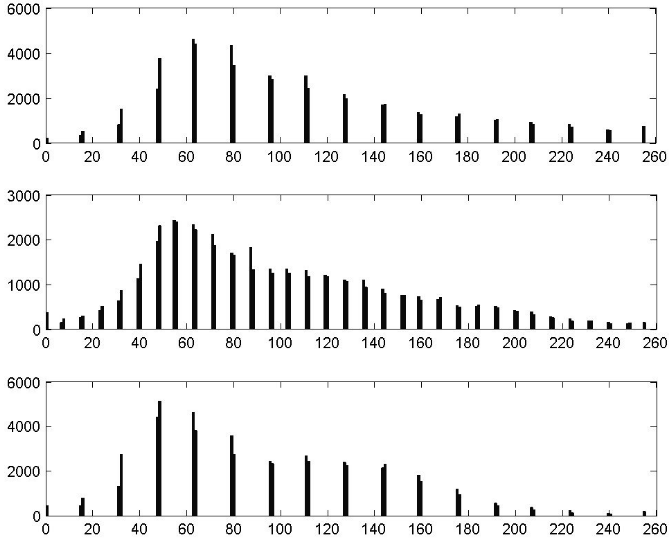


Fig. 7 The histograms of image1. From top to bottom the three histograms represent the distributions of red, green, and blue bands.

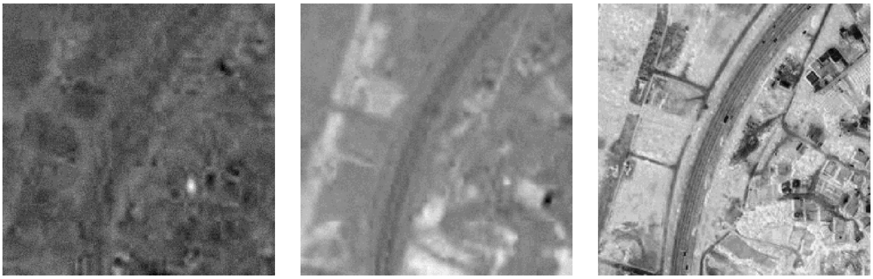
The spectral independent components of an RGB pixel are given by:

$$\begin{bmatrix} IC_1 \\ IC_2 \\ IC_3 \end{bmatrix} = \begin{bmatrix} -0.0767 & 0.1325 & -0.0564 \\ 0.0359 & 0.0079 & -0.041 \\ 0.0106 & -0.0075 & -0.0225 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The coordinates of the ICA transformation indicate an approximate opponent-color model by which R, G, and B are combined into opponent color components. IC_1 is referred to as the violet-green channel, and IC_2 is referred to as the yellow-blue channel. The interesting point is that there is no summation component that usually appears in many transformations, such as the PCA transformation. It should be noted that the transformation matrix is not unique and depend on the data. However, the presence of three chrominance channels without luminance is consistent.

The independent components of image1 and their histograms are shown in Fig. 8, and Fig. 9. Unlike the RGB histograms that have the similar distribution and overlap each other, the independent component histograms are much less overlapped and are concentrated in different ranges. The first component is mainly distributed in the range of [70,120], the second component is in [120, 150], and the third one is in [140,210].

We also applied the information maximization ICA algorithm and principal component analysis to the same training data set. The information maximization



(IC_1) (IC_2) (IC_3)

Fig. 8 The three independent component (IC) images of image1

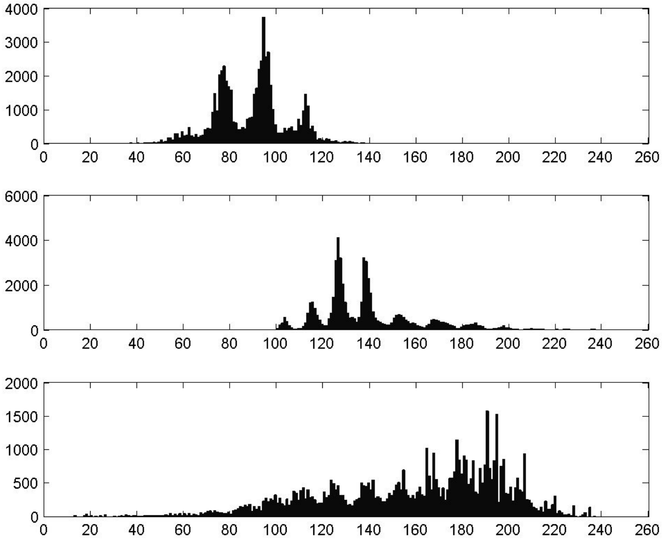


Fig. 9 The histograms of the three independent components of image1. From top to bottom the three histograms represent the distributions of IC_1, IC_2, and IC_3.

algorithm produced a similar transformation matrix without any summation components. However, principal component analysis gave a slightly different matrix:

$$\mathbf{W}_{PCA} = \begin{bmatrix} 0.6136 & 0.5871 & 0.5280 \\ 0.6384 & 0.02451 & -0.7693 \\ 0.4645 & -0.8091 & 0.3598 \end{bmatrix}$$

where the first coordinate is a summation component. This came as no surprise because the first principal component corresponds to a line that passes through the multidimensional mean and minimizes the sum of squares of the distances of the points from the line. In many applications, the first principal component resembles a gaussian or a mean of the original data.

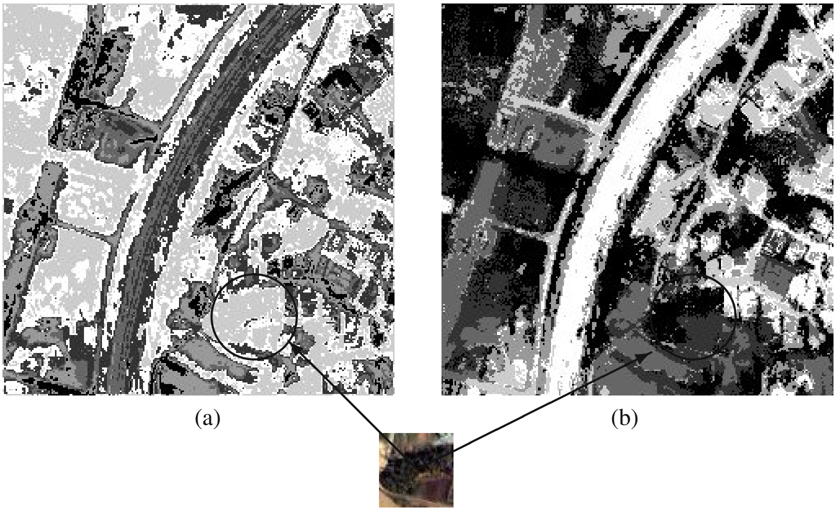
3.3 *Classification of High-resolution Remote Sensing Images*

3.3.1 Pixel Classification by Spectral Information

We first compared the RGB data and the independent components for classifying pixels of the two IKONOS images. The K-means algorithm was used for clustering spectral features. K-means is an unsupervised clustering method and has proven to be an effective technique for many applications [12,6,9]. It uses centroids to represent clusters by optimizing the squared error function. Given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ where each observation is a real vector, K-means clustering aims to partition the n observations into K clusters $\mathbf{S} = \{S_1, S_2, \ldots, S_K\}$ so as to minimize the within-cluster sum of squared Euclidean distance:

$$\sum_{i=1}^{K}\sum_{x_j \in S_i} dist(\mathbf{x}_j, \mathbf{m}_i) = \sum_{i=1}^{K}\sum_{x_j \in S_i} \|\mathbf{x}_j - \mathbf{m}_i\|^2 \tag{45}$$

where $\mathbf{m}_i$ is the centroid of S_i.


(a)

(b)

Fig. 10 The classification map of the remote sensing image1 by (a)RGB, and (b)ICA

The predetermined number of object classes was 6 for image1 and 5 for image2. Fig.10 and Fig.11 show the classification results, where different gray scale levels represent different classes. Although these results may be preliminary because we used an unsupervised clustering algorithm and only spectral information, we here

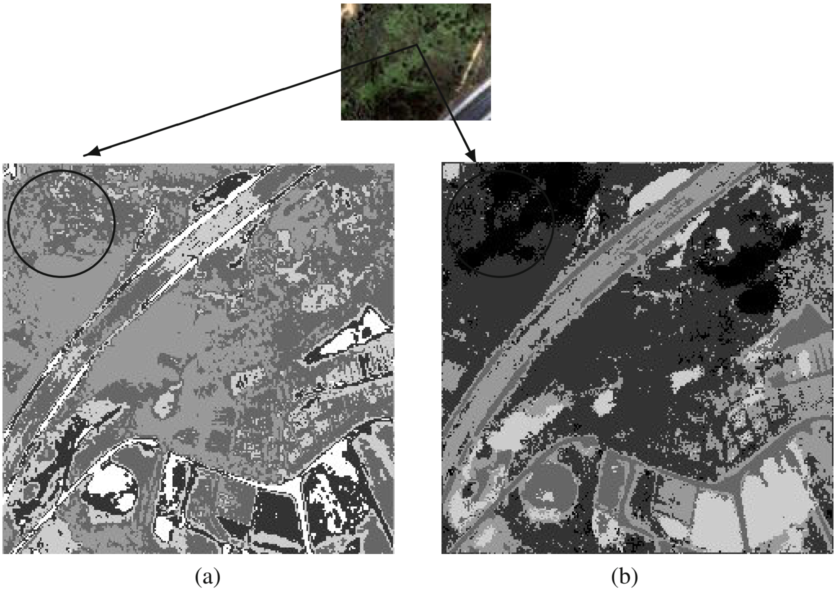


Fig. 11 The classification map of the remote sensing image2 by (a)RGB, and (b)ICA

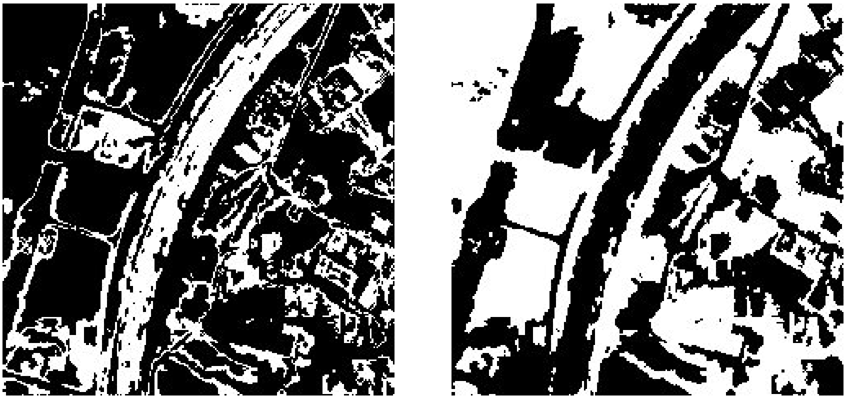
focused on the efficient representation of color information. Note that road and grass pixels in both images were better classified by independent components. For instance, in Fig. 10, the marked area includes dirt and green plants, which was mostly classified by ICA and misclassified as one class by RGB. An interesting observation is that the independent components exhibits some spatial consistency, because the classification map of the independent components is much more smoother than that of the RGB. In Fig. 11, the marked area of a ground cover category has a very accurate map in the classification of independent components, whereas the same area has a noisy map in the RGB classification. We conclude cautiously that the reduction of spectral correlation may have the effect of increasing spatial correlation.

3.3.2 Classification by Spectral Information and Spatial Consistency

Pixel-based classification does not include spatial information and classification maps tend to be noisy. Region-based methods are proposed to overcome this problem and achieve better spatial consistency. We here add a penalty term on the distance of a pixel and a cluster center in the K-means algorithm,

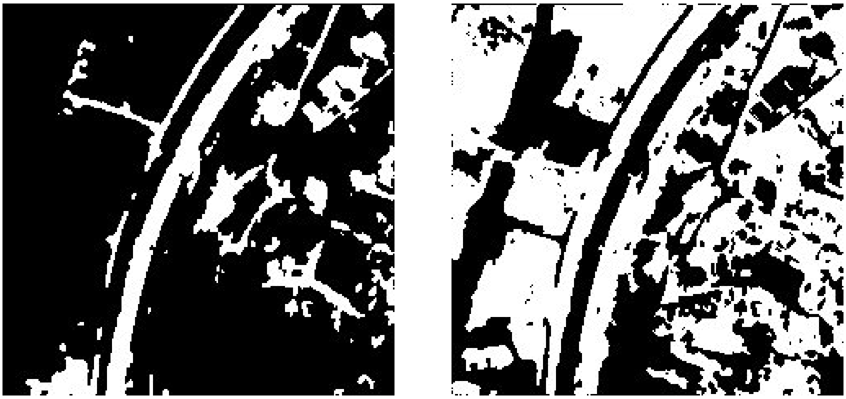
$$dist_{\mathbf{x}_j,\mathbf{m}_i} = \|\mathbf{x}_j - \mathbf{m}_i\| + \alpha / neighbor(\mathbf{x}_j, i) \tag{46}$$

where $neighbor(\mathbf{x}_j, i)$ is the function to compute how many neighboring pixels of $\mathbf{x}_j$ belong to the ith class, and $0 < \alpha < 1$ is a parameter. The second term on the right side of the equation introduces spatial consistency. The class of a pixel is determined not only by the Euclidean distance between the pixel and the centroids but also by



concrete grass

Fig. 12 The map of the concrete class and the green plant class of image1 classified by the RGB data and spatial consistency



concrete grass

Fig. 13 The map of the concrete class and the green plant class of image1 classified by the principal components and spatial consistency

the neighboring pixel's classes. This aims to remove isolated noises and produce smoother classification maps.

The new clustering algorithm was applied to the RGB images, the principal components, and the independent components. The two main classes in these images are concrete objects and green plants that are important in agriculture and transportation applications. The two classes in image1 are shown in Fig.12, Fig. 13 and Fig.14, where the target object is shown in white color. These maps, in particular the RGB classification maps, are smoother than the classification maps obtained from the spectral information without spatial consistency.

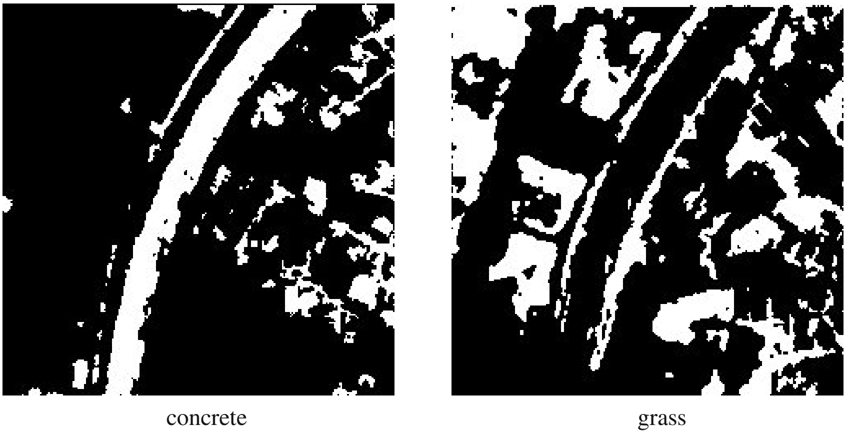


Fig. 14 The map of the concrete class and the green plant class of image1 classified by the independent components and spatial consistency

Table 1 Classification accuracy of the road class

	false negative rate (%)		false positive rate (%)	
	image1	image2	Image1	Image2
RGB	25.3	43.7	42.5	67.6
PCs	47.3	6.1	21.9	28.5
ICs	7.3	4.6	3.5	21.7

To compare the results quantitatively, we manually select random samples from the concrete class and analyze their classifications. The "ground truth" was approximately determined by human observation. The classification performance is measured by two criteria of false negative rate δ_p and false positive rate δ_n defined by:

$$\delta_p = N_p/C \tag{47}$$

and

$$\delta_n = N_n/C \tag{48}$$

where N_p is the number of pixels that have color similar to roads but are not classified into the road class, N_n is the number of pixels that have different color but are misclassified into the road class, and C is the number of road pixels in the ground truth. Table 1 summarized the classification results of the road class in the two test images. Note that generally independent components yield the best results in terms of positive and negative false rate. Although the image2 has shadows in the road area, its classification by the principal components or the independent components has a low false negative rate, which means the majority of the road pixels are correctly classified.

4 Conclusions

In this chapter, we introduced ICA as a feature extraction technique and its application to classification of high-resolution remote sensing images. Linear transformation is normally used in signal processing and pattern recognition to acquire meaningful features from observed data. There are many statistical methods for feature extraction, including PCA and ICA. ICA uses higher-order statistics and have had superior results in many applications. For instance, ICA has been widely applied to hyperspectral remote sensing images for spectral unmixing, where ICA basis functions represent the spectral signatures of ground materials and independent components indicate the proportions of different material categories in mixed pixels. High-resolution remote sensing images have much fewer spectral bands, which means using ICA for spectral unmixing analysis is impractical. We used ICA to learn an efficient color representation of RGB remote sensing images. The obtained independent components were the opponent combination of R, G, and B, and had nonoverlapping distributions. The k-means clustering algorithm was used to classify the independent components. The classification map of the independent components was much smoother than that of the RGB data, which suggested that reduction of spectral correlation may lead to increase of spatial correlation. This is an interesting finding about the encoding of spectral and spatial information of images and worth further study.

References

1. Barlow, H.: Sensory Communication. MIT Press, Boston (1961)
2. Bell, A., Sejnowski, T.J.: An information-maximasation approach to blind separation and blind deconvolution. Neural Computation 7, 1129–1159 (1995)
3. Bell, A., Sejnowski, T.J.: The "Independent Components" of natural scenes are edge filters. Vision Res. 37(23), 3327–3338 (1997)
4. Buchsbaum, G., Gottschalk, A.: Trichromacy, opponent colours coding and optimum colour information transmission in the retina. Proceedings of the Royal Society London, B 220, 89–113 (1983)
5. Chang, C.I., Chiang, S.-S., Smith, J.A., Ginsberg, I.W.: Linear spectral random mixture analysis for hyperspectral imagery. IEEE Trans. on Geoscience and Remote Sensing 40(2), 375–392 (2002)
6. Coates, A., Ng, A.Y.: Learning feature representations with K-means. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, 2nd edn. LNCS, vol. 7700, pp. 561–580. Springer, Heidelberg (2012)
7. Comon, P.: Independent component analysis- a new concept? Signal Processing 36, 287–314 (1994)
8. Deco, G., Obradovic, D.: Independent Component Analysis: General Formulation and Linear Case, An Information-theoretic Approach to Neural Computing. Springer, New York (1996)

9. Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of International Conference on Machine Learning, pp. 225–232 (2004)
10. Du, Q., Kopriva, I., Szu, H.: Independent-component analysis for hyperspectral remote sensing imagery classification. Opt. Eng. 45(1), 1–13 (2006)
11. Dubuisson-Jolly, M.P., Gupta, A.: Color and texture fusion: application to aerial image segmentation and GIS updating. Image and Vision Computing 18, 823–832 (2000)
12. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Join Wiley and Sons, Inc., New York (1973)
13. Herault, J., Jutten, C.: Space or time adaptive signal processing by neural network models. In: Denker, J.S. (ed.) Neural Network for Computing. In: Proceedings of AIP Conference, pp. 206–211. American Institute of Physics, New York (1985)
14. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. on Neural Networks 10(3), 626–634 (1999)
15. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and application. Neural Networks 13(4-5), 411–430 (2000)
16. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiely and Sons, Inc., New York (2001)
17. Klein, U., Sester, M., Strunz, G.: Segmentation of remotely sensed image based on the uncertainty of multispectral classification. IAPRS, GIS-between Vision and Applications, Stuttgart. 32(4), 299–305 (1998)
18. Lee, T.W., Achtler, T., Sejnowski, T.J.: The spectral independent components of natural scenes. Institute for Neural Computation at UCSD Technical Report Series No. INC-9901, San Diego (1999)
19. Murai, H., Omatsu, S., OE, S.: Principal component analysis for remotely sensed data classified by kohonens feature mapping preprocessor and multi-layered neural network classifier. IEICE Trans. Commun. E78-B (12), 1604-1610 (1995)
20. Ruderman, D.L.: Statistics of cone responses to natural images: implications for visual coding. J. Opt. Soc. Am. 15(8), 2036–2045 (1998)
21. Schiewe, J.: Segmentation of high-resolution remotely sensed data-concept, application and problem. In: Proceedings of Symposium on Geospatial Theory, Processing and Applications, Ottawa (2002)
22. Settle, J.J., Drake, N.A.: Linear mixing and estimation of ground cover proportions. Int. J. Remote Sensing 14, 1159–1177 (1993)
23. Tailor, D.R., Finkel, L.H., Buchsbaum, G.: Color-opponent receptive fields derived from independent component analysis of natural images. Vision Res. 40(19), 2671–2676 (2000)
24. Tu, T.: Unsupervised signature extraction and separation in hyperspectral images: a noise-adjusted independent component analysis approach. Opt. Eng. 39(4), 897–906 (2000)
25. Villa, A., Chanussot, J., Benediktsson, J.A., Jutten, C., Dambreville, R.: Unsupervised methods for the classification of hyperspectral images with low spatial resolution. Pattern Recognition 46(6), 1556–1568 (2013)
26. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formula, 2nd edn. Wiley, New York (1982)
27. Zhang, X., Chen, C.H.: New independent component analysis method using higher order statistics with application to remote sensing images. Opt. Eng. 41(7), 1717–1728 (2002)

List of Acronyms

BSS	Blind Source Separation
IC	Independent Component
ICA	Independent Component Analysis
PCA	Principal Component Analysis
RGB	Red-Green-Blue
ZCA	Zero-phase Component Analysis

Chapter 4
Subspace Construction from Artificially Generated Images for Traffic Sign Recognition

Hiroyuki Ishida, Ichiro Ide, and Hiroshi Murase

Abstract. Recognition technologies using digital cameras have gained considerable interest in recent years. However, even with the improvements of digital cameras, the quality of captured images often can be insufficient for the recognition in many practical cases. In order to recognize low-quality images, similarly degraded images should be used for training classifiers. This chapter presents a training method for the subspace method. It is named "Generative learning method," since the training images are generated artificially from an original image. Conventional approaches used camera-captured images as training data, which required exhaustive collection of captured samples. The generative learning method, instead, allows to obtain these training images based on a small set of actual images. Since the training images need to be generated on the basis of actual degradation characteristics, the estimation step of the degradation characteristics is introduced. This framework is applied to traffic sign recognition that is one of the important tasks for driver support systems.

1 Introduction to the Generative Learning

High-resolution digital cameras have come into widespread use in recent years. Recognition technologies using such digital equipments are especially of practical concern. However, objects in distant places still tend to be captured in low-resolution and blurred, which has a serious effect on the recognition performance.

The generative learning method [1] is developed to solve the problem of degradation. It was originally proposed for the recognition of hand-written characters [2, 3].

Hiroyuki Ishida
Toyota Central R&D Labs. Inc., 41-1, Yokomichi, Nagakute, Aichi, Japan
e-mail: h-ishida@mosk.tytlabs.co.jp

Ichiro Ide · Hiroshi Murase
Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan
e-mail: {murase,ide}@is.nagoya-u.ac.jp

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_4, © Springer-Verlag Berlin Heidelberg 2014

It generates artificially degraded samples, and allows to make classifiers trained by them. Traditionally, training images ought to be collected from actual images taken in the real world. Such a collection-based approach may be the most straightforward approach to obtain a set of training samples. In many practical cases, however, camera-based collection of a sufficient number of training images in various conditions is unrealistic. Let us consider collecting the training images for many categories. In the character recognition task [4], for instance, the number of categories tends to be large, and at the same time, printed text may even contain various types of fonts. This diversity of characters makes the collection difficult. Moreover, various conditions that cause respective distortions in captured images should be taken into account.

In contrast, the generative learning method eliminates the necessity of capturing an exhaustive collection of training images. All the training images are generated artificially from a smaller number of original images. However, if such artificial generation is performed regardless to realistic models, this method might not be sufficient as a “training” method; it is important to simulate the actual degradation systems. Thus, models are initially defined as corresponding to the actual degradation factors. The generative learning method consists of two main parts: (1) estimation of actual degradation systems and (2) generation of training images based on the estimated degradation models. Details of each part are described below.

1.1 Modeling of Degradation Characteristics

Degradation characteristics need to be modelled before working on the generation of training data. They can be optical blur, motion blur, segmentation errors, and so on. For each of these models, parameters to control the degree of degradations are defined. Let p be a vector containing parameters from all the models, a training image is generated from the original image using it, and then a set of training images is obtained by applying a set of different parameter vectors. These parameters are applicable to all categories, therefore training images for all categories can be obtained. Fig. 1 illustrates an example of the degradation model, where the parameter σ controls the standard deviation of the Gaussian blur function.

1.2 Estimation of Degradation Characteristics

Once the degradation model is defined, it becomes possible to generate a wide variety of training images. However, it is the parameters that actually determine the properties of the generated samples. This is why parameter estimation is necessary for the reproduction of actual degradation characteristics. In the example of the blur model in Fig. 1, it is important to estimate the value of σ in general cases.

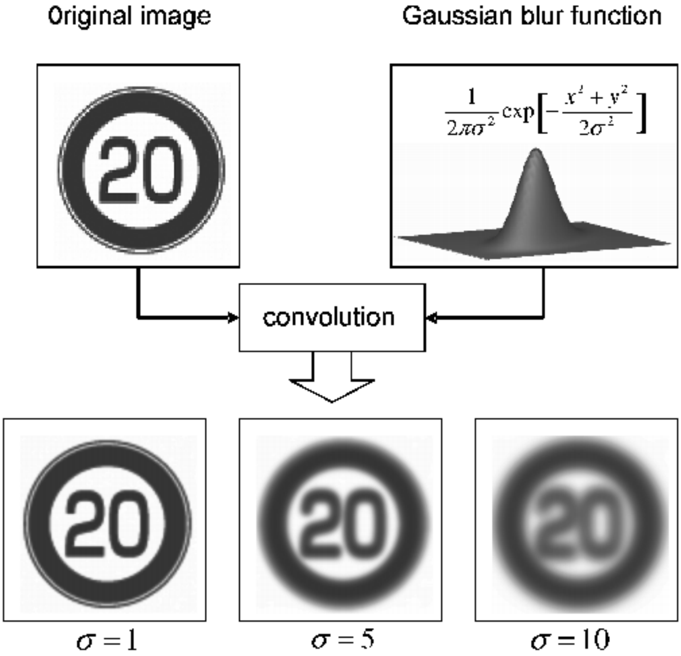


Fig. 1 The Gaussian blur model. The level of blurring is controlled by a parameter σ.

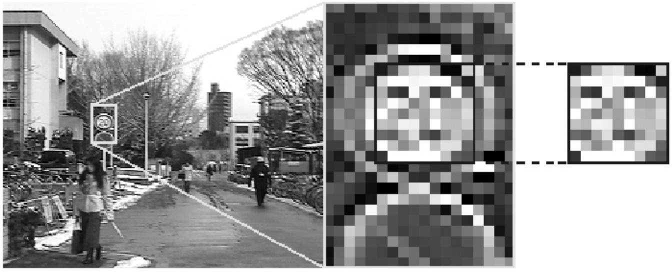


Fig. 2 A traffic sign symbol extracted from an actual image taken by a digital video camera

If the blur function cannot be assumed to be a Gaussian, then σ should be replaced by a point spread function (PSF) [5]. This PSF is used when the degradation characteristic of a camera is unknown.

2 Generative Learning for Traffic Sign Recognition

The traffic sign recognition is one of the important tasks for supporting drivers. The subspace method [6] is used for the classification of traffic sign symbols. To construct subspaces, the abovementioned generative learning method is employed to generate various training images of traffic signs. Collection of all training images under various conditions is especially difficult for the traffic sign recognition, therefore the generative learning method is useful. This training step includes an estimation step of parameters.

Technologies for supporting drivers with in-vehicle cameras have gained considerable industrial interests in recent years. Many studies have been conducted on pedestrian detection, traffic signal recognition, road marking recognition, and road environment understanding [7, 8]. Traffic sign recognition is another important task. If such a recognition system comes into practice, it could support drivers by informing them of the current speed limit, for instance. Also, it could be applicable for periodically updating a road map database [9] used for navigation systems. Two main issues in the traffic sign recognition are detection and classification. Various attempts have been carried out on the detection of traffic signs: edge detection mask [10], hierarchical template [11], shape information [12], and color information [13]. There are methods proposed specifically for circular sign detection [14, 15]. Works [16] and [17] present methods for shape classification. The category classification of extracted signs also is an important task. Some studies have been conducted on the category classification of extracted signs. In [18], results from high-quality images are preferentially used for avoiding degradations. A method for speed sign classification [19] copes with the rotation of traffic sign symbols. It is also important to focus on the various degradations appearing in camera-captured images. This Chapter focuses on the classification of variously degraded traffic sign symbols, and introduces a method [20] for constructing classifiers. First of all, generation models are introduced in Section 2.1. They are defined as corresponding to actual degradation factors. In Section 2.2, the strategies for generating training images are described, together with the estimation step of generation parameters.

2.1 Generation Models of Traffic Signs

Training images are generated from an original image by three degradation models: rotation, blurring, and segmentation error. These models are defined with generation parameters, as shown in Fig. 3. The parameters are listed in Table 1. Given an original CG image P_0 of a traffic sign symbol, a degraded image P_3 is generated from P_0 as described below:

1. Rotation
   This model simulates the rotation of traffic signs. Assume that the original traffic sign plate exists in plane $z = 0$, and its center is at point $(0,0,0)$. Rotation angle

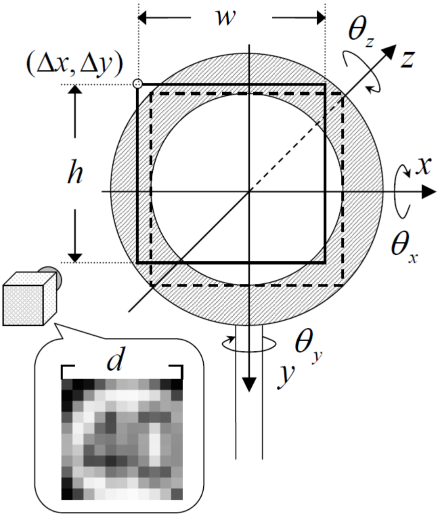


Fig. 3 Proposed generation model

Table 1 Parameters for the generation models

θ_x	Rotation angle around the *x*-axis
θ_y	Rotation angle around the *y*-axis
θ_z	Rotation angle around the *z*-axis
γ	Gauss parameter of focus blur
Δx	Horizontal gap
Δy	Vertical gap
w	Width of the segmentation area
h	Height of the segmentation area
d	Segmented image size

parameters are denoted by $\theta_x,\ \theta_y$, and θ_z. The rotation matrices around each axis are denoted by R_x, R_y, and R_z. The operation of rotating the traffic sign plate is represented as

$$P_1(x,y) = P_0(x',y'). \tag{1}$$

Values x' and y' are determined by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \left(R_z(\theta_z)R_y(\theta_y)R_x(\theta_x)\right)^{-1} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}, \tag{2}$$

where

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{3}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{4}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

2. Blurring
   This model is used to simulate focus blur. For simplicity, the blurring function is assumed to be a Gaussian function. The level of blurring is controlled by a single Gaussian parameter γ. This blurring operation is represented using convolution $(*)$ as

$$P_2(x,y) = P_1(x,y) * \left[\frac{1}{2\pi\gamma^2}\exp\left(-\frac{x^2+y^2}{2\gamma^2}\right)\right]. \tag{6}$$

3. Segmentation error
   This model is used to simulate incorrectly segmented symbol images. Horizontal and vertical gap parameters $(\Delta x, \Delta y)$, segmented area parameters (w,h), and segmented image size d are introduced. Resolution transformation is performed together in this model. P_3 is obtained by

$$P_3(i,j) = \frac{1}{|D_{(i,j)}|}\sum_{x,y\in D_{(i,j)}} P_2(x,y), \tag{7}$$

   where $D_{(i,j)}$ is a set of pixels projected on pixel $P_3(i,j)$, and $|D_{(i,j)}|$ is the number of pixels in $D_{(i,j)}$ It is represented as

$$D_{(i,j)} = \left\{(x,y) \;\middle|\; \frac{i}{d+1}w \le x - \Delta x < \frac{i+1}{d+1}w, \right. \\ \left. \frac{j}{d+1}h \le y - \Delta y < \frac{j+1}{d+1}h\right\}. \tag{8}$$

The size of the generated image P_3 is d $(0 < i,j \le d)$.

2.2 Training by Generative Learning

From the viewpoint of constructing training sets, images with various levels of degradation should be obtained. Specifically, a range of the degradation levels should be adequately determined. This is especially needed for the application using an in-vehicle camera because the image degradation tends to be serious due to camera movement.

This method estimates the parameter range from captured images, since it can be considered that the estimated parameter range is suited to recognize traffic signs captured in similar conditions. To represent the parameter range, a multi-variational normal distribution is used for approximation. It provides a simple and general framework, in which the parameter range can be controlled by mean and variance. Once they are obtained, the degradation characteristics in the generation step of the training images can be reproduced. This is possible for any category of traffic sign symbols because the degradation models are applicable universally to them. Recall that capturing the training images of all categories is extremely difficult for traffic sign recognition. The major advantage of the generative learning method is that the training images of all categories can be obtained completely by the generation.

This method consists of two steps. The first is the parameter estimation step introduced in 2.2.1. The second is the generation step introduced in 2.2.2.

2.2.1 Parameter Estimation Step

The distribution of generation parameters is estimated from actual images. Before that, however, parameters need to be estimated for each image.[1]

As introduced in Section 1.1, a parameter vector p consisting of the generation parameters is defined as

$$p = (\theta_x, \theta_y, \theta_z, \gamma, \Delta x, \Delta y, w, h). \tag{9}$$

Using this vector, degraded traffic sign images are generated from an original image. Fig. 4 illustrates this estimation step. Let T be one of the captured images for parameter estimation and Q be an image generated from the original image "speed limit 30 km/h" using p. A parameter vector $\hat{p}$, which maximizes the similarity between Q and T, should be found and regarded as the optimal representation of the degradation characteristics of T. The similarity between these two images is given by an inner product $\langle q, t \rangle$, where vectors q and t consist of the pixel values of images Q and T, respectively.[2] The maximization of this similarity is achieved by the Genetic Algorithm (GA) [22]. Fig. 5 illustrates the operations of crossover and mutation in GA. A detailed description of the GA-based parameter estimation algorithm is given in Table 3. Table 2 lists parameters which are used in the algorithm of GA. Fig. 6 shows an example of a captured image t and images simulated by GA.

[1] These images should be captured by the same camera as that used in the recognition step. It is also required to exclude the images which look obviously unsuitable for the parameter estimation. If the degradation characteristics of the images are dissimilar to the general ones, the performance of the generative learning method will not be satisfactory.

[2] Each vector is normalized such that the mean of its elements is 0 and the norm is 1, namely, $\langle q, q \rangle = \langle t, t \rangle = 1$.

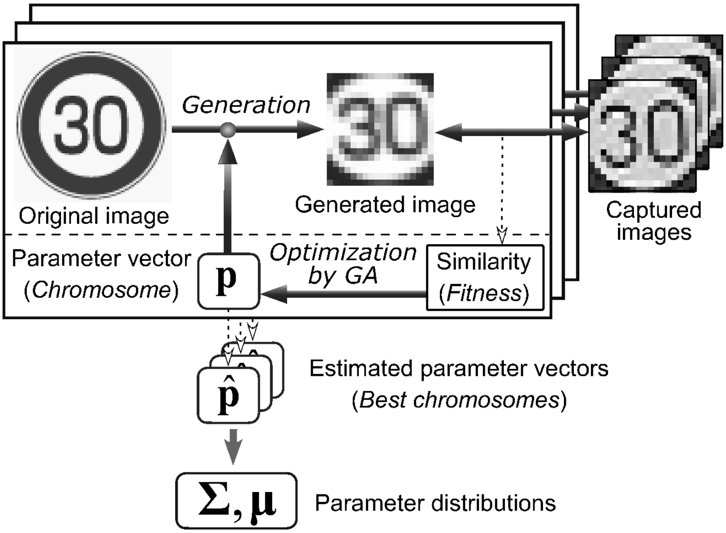


Fig. 4 Parameter estimation step

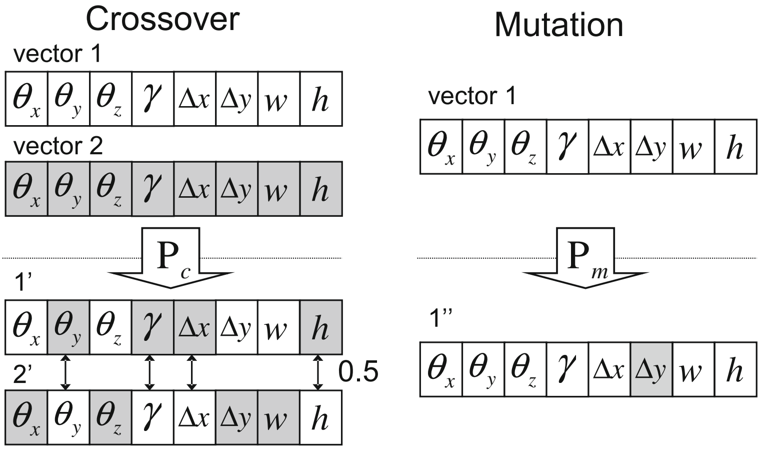


Fig. 5 Operations used in Genetic Algorithm

The parameter distribution is estimated from multiple parameter vectors $\hat{p}$ computed from the captured images. The mean vector μ and covariance matrix Σ are then obtained from the multiple vectors $\hat{p}$ by

$$\mu = \mathscr{E}[\hat{p}], \tag{10}$$

$$\Sigma = \mathscr{E}\left[(\hat{p}-\mu)(\hat{p}-\mu)^{\top}\right]. \tag{11}$$

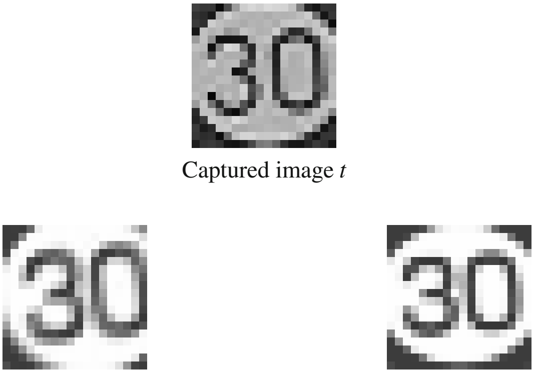
Table 2 Parameters for the Genetic Algorithm

N_c	Population size
G	Number of generations
P_c	Crossover rate
P_m	Mutation rate

Table 3 Parameter estimation algorithm based on Genetic Algorithm [22]

```
Algorithm
// $C_p$: Parents set
// $C_c$: Children set
// $t$: Normalized captured image $T$
// $q$: Normalized generated image $Q$
initialize set $C_p$ and its $N_c$ chromosomes $p_i$
do
    for all $p_i \in C_p$
        generate $q_i$ from the original image of $t$ with $p_i$
        calculate fitness $s_i = \langle q_i, t \rangle$
    next
    do
        select chromosomes $p_a, p_b$ by roulette selection
        reproduce $p_a \to p'_a$, $p_b \to p'_b$
        /* Crossover */
        if Rand$[0,1) < P_c$ then cross $p'_a$ with $p'_b$
        add $p'_a$, $p'_b$ to $C_c$
    until $|C_p| = |C_c|$
    for each chromosome $p_i$ of $C_c$
        /* Mutation */
        if Rand$[0,1) < P_m$ then
            randomly initialize one of the elements of $p_i$
    next
    copy $C_c \to C_p$
    empty $C_c$
until generation reaches $G$
$\hat{p} := p_i$ with the largest fitness $s_i$
return $\hat{p}$
```

Note that size parameter d does not appear in Eqs. (9)–(11) because d can be obtained directly from each captured image itself. While the other parameters of p are estimated by the algorithm in Table 3, the value of d is set equal to the size of the captured image. Also for the sake of simplicity, it is assumed that d is independent of the other parameters; μ and Σ are computed regardless to d.



Captured image *t*

Image of the first generation. (similarity 0.742)

Image of the 100th generation. (similarity 0.919)

Fig. 6 Images generated to reproduce a captured image as similar as possible

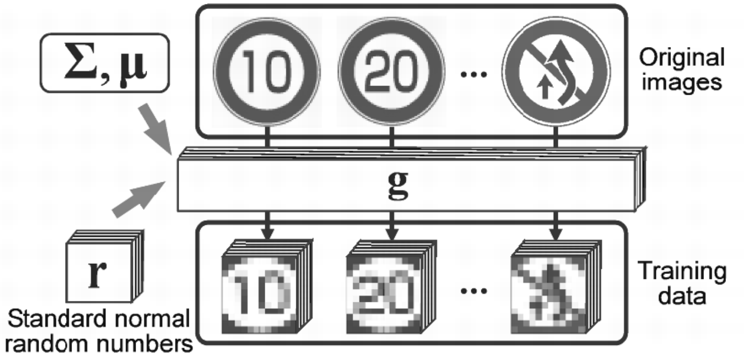


Fig. 7 Generation of training dataset

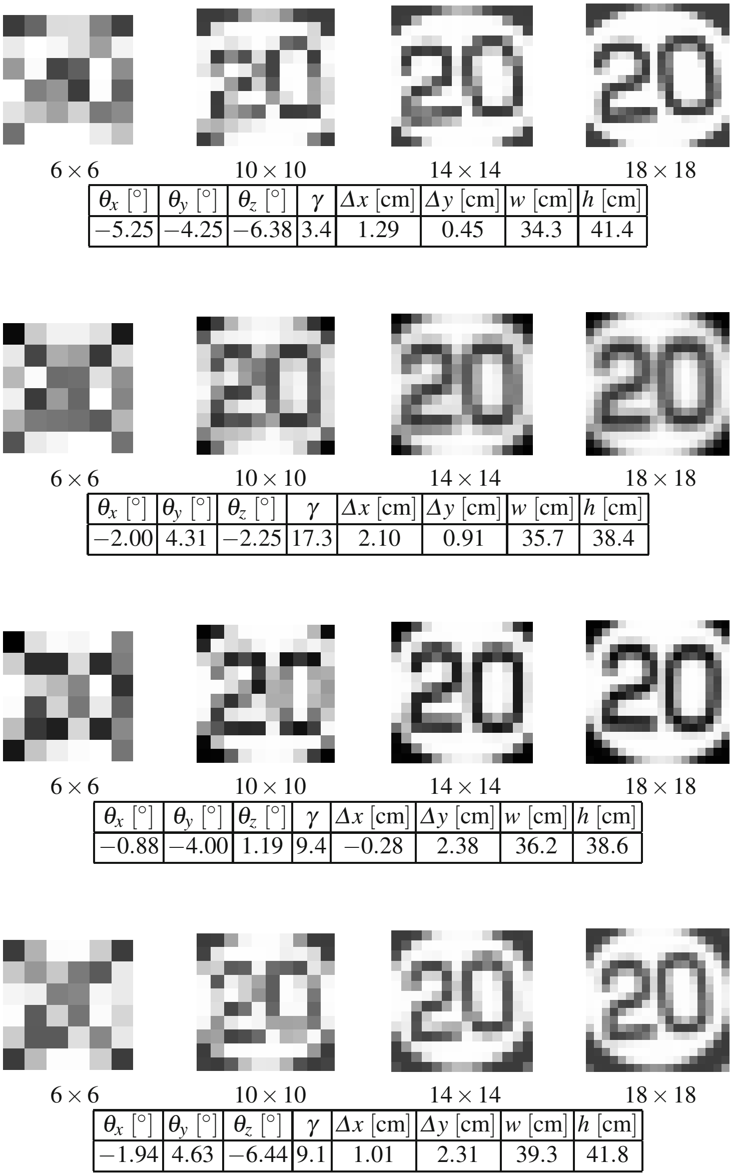
2.2.2 Generation of Training Images

Once the parameter distribution is estimated, a parameter vector g, which follows the estimated distribution (μ, Σ), is reproduced by the following parameter-producing function:

$$g = \Sigma^{1/2} r + \mu, \tag{12}$$

where r denotes a vector composed of standard normal random numbers[3] [23] and $\Sigma^{1/2}$ denotes the Cholesky decomposition [24] of Σ. Fig. 7 illustrates this generation step. Various parameter vectors are produced, and correspondingly, various training images of all categories are generated. Some examples of the generated training images are shown in Fig. 8.

[3] Generator of standard normal random numbers and the Cholesky decomposition are available in MIST libraries [25].



6 × 6 10 × 10 14 × 14 18 × 18

θ_x [°]	θ_y [°]	θ_z [°]	γ	Δx [cm]	Δy [cm]	w [cm]	h [cm]
−5.25	−4.25	−6.38	3.4	1.29	0.45	34.3	41.4

6 × 6 10 × 10 14 × 14 18 × 18

θ_x [°]	θ_y [°]	θ_z [°]	γ	Δx [cm]	Δy [cm]	w [cm]	h [cm]
−2.00	4.31	−2.25	17.3	2.10	0.91	35.7	38.4

6 × 6 10 × 10 14 × 14 18 × 18

θ_x [°]	θ_y [°]	θ_z [°]	γ	Δx [cm]	Δy [cm]	w [cm]	h [cm]
−0.88	−4.00	1.19	9.4	−0.28	2.38	36.2	38.6

6 × 6 10 × 10 14 × 14 18 × 18

θ_x [°]	θ_y [°]	θ_z [°]	γ	Δx [cm]	Δy [cm]	w [cm]	h [cm]
−1.94	4.63	−6.44	9.1	1.01	2.31	39.3	41.8

Fig. 8 Examples of generated images for "speed limit 20 km/h" in various resolutions [pixels]

3 Recognition by the Subspace Method

The subspace method [6] is used in the recognition step. The process of constructing a subspace is described in 3.1, followed by a description of the recognition step using multiple-frame integration in 3.2. A simple algorithm to extract circular traffic signs is outlined in 3.3.

3.1 Construction of a Subspace

A subspace is constructed from various training images for each category and also for each size.

Let $\mathscr{P}$ be a set of N different parameter vectors p_n $(n = 1, 2, \cdots, N)$, where N is the number of training images used for constructing a subspace of a category. N training images are generated from parameter vectors $p_n \in \mathscr{P}$. For each $d \times d$ training image $\mathrm{x}^{(c)}_{p_n,d}$ of category c, a vector $x^{(c)}_{p_n,d}$ is constructed from pixel values of the image as described below. First, an image $\mathrm{x}^{(c)}_{p_n,d}$ is converted to a vector $\tilde{x}^{(c)}_{p_n,d}$ such that the mean of its elements becomes 0 by

$$\begin{aligned}\tilde{x}^{(c)}_{p_n,d} = \Big[\quad & \mathrm{x}^{(c)}_{p_n,d}(0,0) - \bar{x}^{(c)}_{p_n,d} \quad \cdots \quad \mathrm{x}^{(c)}_{p_n,d}(d-1,0) - \bar{x}^{(c)}_{p_n,d} \\ \cdots \quad & \mathrm{x}^{(c)}_{p_n,d}(0,d-1) - \bar{x}^{(c)}_{p_n,d} \quad \cdots \quad \mathrm{x}^{(c)}_{p_n,d}(d-1,d-1) - \bar{x}^{(c)}_{p_n,d} \Big]^\top,\end{aligned} \tag{13}$$

where the mean $\bar{x}^{(c)}_{p_n,d}$ is calculated by

$$\bar{x}^{(c)}_{p_n,d} = \frac{1}{d^2} \sum_{x=0}^{d-1} \sum_{y=0}^{d-1} \mathrm{x}^{(c)}_{p_n,d}(x,y).$$

Secondly, this vector is normalized to $x^{(c)}_{p_n,d}$ whose norm is 1 by

$$x^{(c)}_{p_n,d} = \frac{\tilde{x}^{(c)}_{p_n,d}}{\left\| \tilde{x}^{(c)}_{p_n,d} \right\|}. \tag{14}$$

Next, a matrix $X^{(c)}_d$ whose size is $d^2 \times N$ is constructed from N normalized vectors $x^{(c)}_{p_n,d}$ by

$$X^{(c)}_d = \left[x^{(c)}_{p_1,d} \;\; \cdots \;\; x^{(c)}_{p_N,d} \right]. \tag{15}$$

An auto-correlation matrix $Q^{(c)}_d$ whose size is $d^2 \times d^2$ is computed by

$$Q^{(c)}_d = X^{(c)}_d \left(X^{(c)}_d \right)^\top. \tag{16}$$

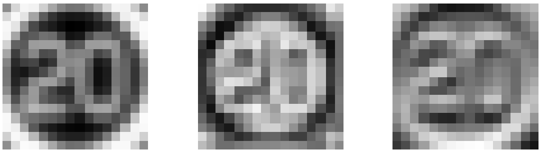


Fig. 9 Top three eigenvectors (Speed limit 20 km/h, size 16×16 pixels)

Eigenvectors are derived from $Q_d^{(c)}$, of which $e_{\{l,d\}}^{(c)}$ $(l = 1, \cdots, L)$ with the largest L $(L < N)$ eigenvalues are used for recognition. Fig. 9 shows examples of the eigenvectors. The reason why the subspaces are constructed for each size d is that size normalization can have an undesirable effect on the matching process. If the image size is changed, the influence of pixel interpolation on very small images is not negligible.

3.2 Multiple Frame Integration

An input image is classified to a category c that maximizes the similarity. In the subspace method, the similarity is given by the sum of the squared inner product between the given image and the eigenvectors. Yanadume et al. demonstrated that integrating similarities from multiple frames improves recognition accuracy [21]. Given M image frames of the same target, let z_m be the m-th input image $(m = 1, \cdots, M)$ converted in vector form; the recognition result is obtained by

$$\hat{c} = \arg\max_{c} \sum_{m=1}^{M} \sum_{l=1}^{L} \left(e_{\{l,\bar{d}_m\}}^{(c)\top} z_m \right)^2 , \tag{17}$$

where $\bar{d}_m$ represents the size of the segmented image z_m. In order to distinguish it from the generation parameter d, the size of the captured images is denoted by $\bar{d}_m$.

3.3 Circular Sign Detection

HSV color space [26] is useful for the extraction of symbol regions in circular signs, since H and S are nearly uniform in respect to changes of illumination. A discriminant function for finding the red circumference is defined as

$$\mathrm{red}(x,y) = \begin{cases} 1 & \begin{pmatrix} -\pi/9 < H(x,y) < \pi/9 \\ \text{and } 0.2 < S(x,y) \leq 1 \\ \text{and } 30 \leq V(x,y) \leq 255 \end{pmatrix} \\ 0 & \text{otherwise} \end{cases} . \tag{18}$$

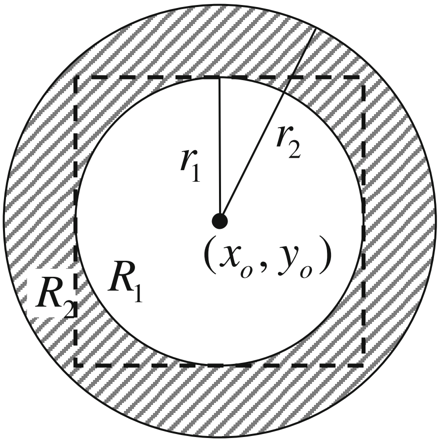


Fig. 10 Extraction parameters defined for a circular sign

Circular signs is detected by matching a doughnut-shaped structure shown in Fig. 10 with segmentation parameters. Here (x_0, y_0) is the center point, R_1 is the symbol area, R_2 is the red circumferential area, and r_1 and r_2 are the radii of R_1 and R_2, respectively. They are represented as

$$R_1 = \left\{ (x,y) \,\middle|\, \sqrt{(x-x_0)^2 + (y-y_0)^2} < r_1 \right\} \tag{19}$$

and

$$R_2 = \left\{ (x,y) \,\middle|\, r_1 < \sqrt{(x-x_0)^2 + (y-y_0)^2} < r_2 \right\}. \tag{20}$$

The extracted region is the smallest square that includes the entire symbol area. Using Eqs. (18), (19), and (20), segmentation parameters (x_0, y_0) and segmented image size $\bar{d}$ are obtained by

$$\left\{ x_0, y_0, \frac{\bar{d}}{2} \right\} = \arg \max_{\{x,y,r_1\}} \left[\sum_{(x,y)\in R_2} \frac{\mathrm{red}(x,y)}{\pi(r_2^2 - r_1^2)} - \sum_{(x,y)\in R_1} \frac{\mathrm{red}(x,y)}{\pi r_1^2} \right]. \tag{21}$$

This segmentation algorithm is applied to the input video stream. Searching only neighborhoods of (x_0, y_0) obtained from previous frames is effective for the reduction of computational complexity and false recognition.

4 Experiment

An experiment was performed using video streams captured by an in-vehicle camera (Table 4) during one run on a sunny morning. Using the detection algorithm in 3.3, the symbol images were cropped from 1,073 images in the video stream. The number of categories contained in this test data was five (No. 2, 4, 5, 12, and 20),

Table 4 Specifications of the in-vehicle camera

Product model	Sony DCR-PC105
Resolution	720 × 480pixels
Frame rate	30 fps
Focus length	3.7 mm

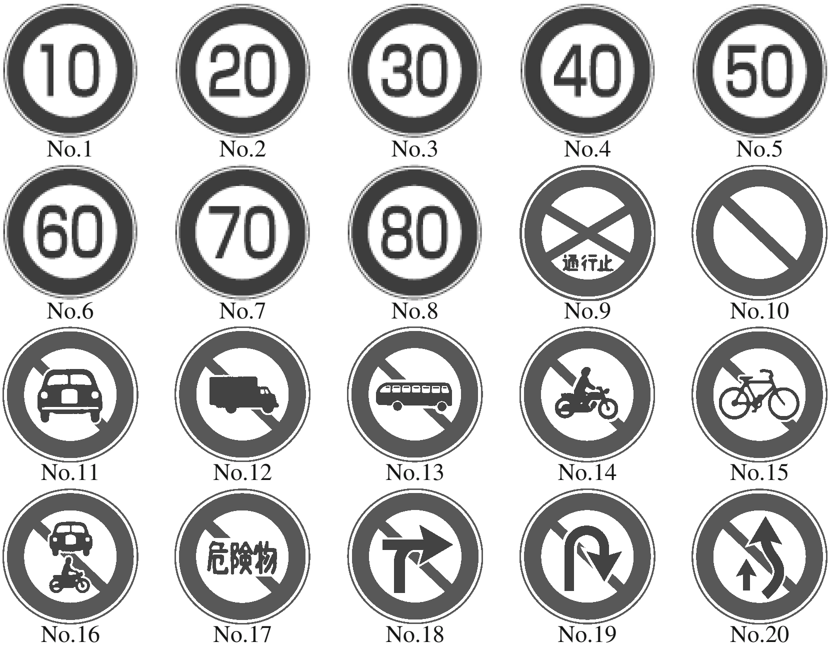


Fig. 11 Traffic sign categories

where Fig. 11 illustrates twenty circular traffic signs used in Japan. These symbol images were divided into five data sets (sets A–E) by their category as shown in Table 5. In this experiment, a variant of 5-fold cross validation was used; each data set was chosen once for the parameter estimation, and the remaining four sets were used for testing. It was to ascertain whether parameters estimated from a single category were valid for constructing classifiers of other categories [4]. Fig. 12 shows the size distribution of the segmented images, and Fig. 13 shows examples of the images.

In the training step, the parameter distribution was estimated using the algorithm in Table 3 with $N_c = 100$, $G = 100$, $P_c = 0.7$, and $P_m = 0.01$. Instead of Eq. (12), training images were generated by a parameter producing function in which $\Sigma^{1/2}$

[4] In the actual application, we would want to train parameters from a limited number of samples.

Table 5 Number of symbol images in each data set

Set	Category	Number of symbol images
A	No. 2	174
B	No. 4	356
C	No. 5	214
D	No. 12	214
E	No. 20	115

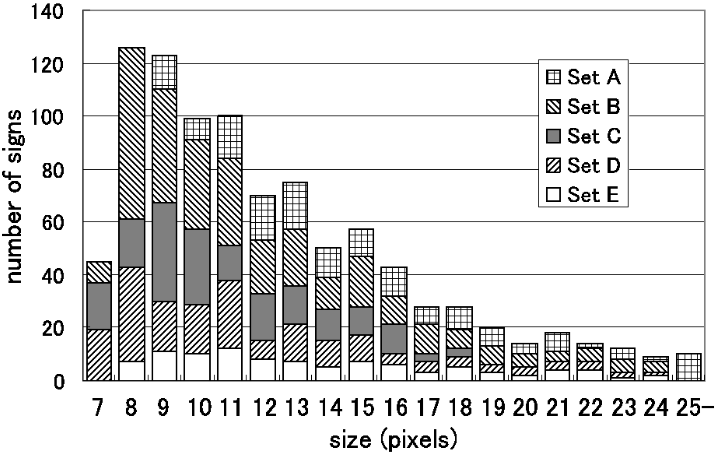


Fig. 12 Distribution of traffic sign size

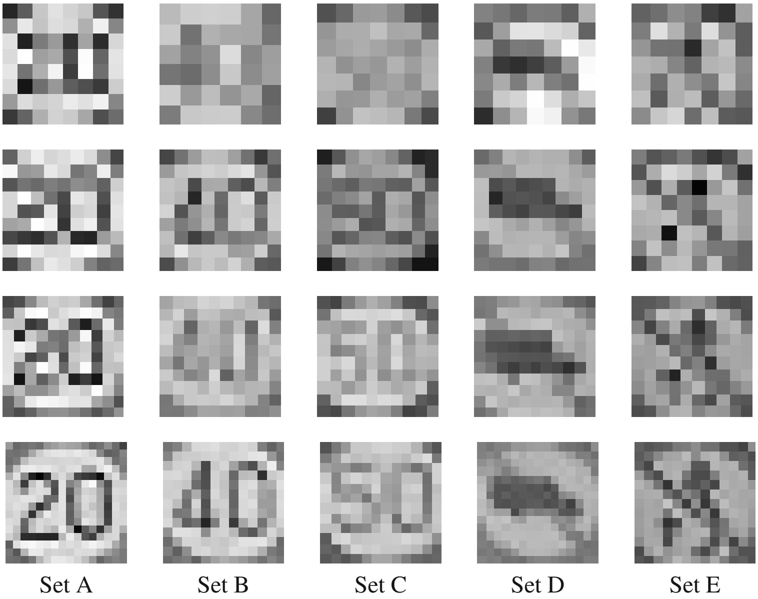
was weighted on as

$$g = k\Sigma^{1/2}r + \mu, \tag{22}$$

where k is considered as a factor that controls the parameter range by weighting on the estimated $\Sigma^{1/2}$. The number of the generated training images was $N = 200$. Recognition rates in six cases $(k = 0, 1/4, 1/2, 1, 2, 4)$ were compared. In the case of $k = 0$, however, only a single training image $(g = \mu)$ was obtained from Eq. (22). Hence in this case, the input images were classified by

$$\hat{c} = \arg\max_{c} \sum_{m=1}^{M} \left({x_{\bar{d}_m}^{(c)}}^{\top} z_m \right) \tag{23}$$

with a single training image $x_{\bar{d}_m}^{(c)}$. In the other cases, recognition results were obtained by Eq. (17). The case of $k = 1$ was identical to the presented generative learning method, since Eq. (22) equals Eq. (12). In the recognition step, ten successive frames were integrated $(M = 10)$, and ten eigenvectors were used $(L = 10)$.



Set A Set B Set C Set D Set E

Fig. 13 Examples of test images

Table 6 Average recognition [%] rates from single frame ($M = 1$) and multiple frame integration ($M = 10$)

Weight k	0	1/4	1/2	1	2	4
Single frame	48.0	81.7	83.4	84.3	82.7	82.4
Multiple frames	57.4	89.2	91.7	92.9	91.4	91.2

4.1 Results

Recognition rates are presented in Fig. 14, where the horizontal axis in the graph represents the maximum symbol size $\bar{d}_{\max}$ within the integrated M frames. As shown in the results, the recognition rates have strong relationships with the image sizes. The generative learning method exhibited high recognition rates; the recognition rate of relatively large symbols ($\bar{d}_{\max} \geq 20$) was 100%. For small symbols ($\bar{d}_{\max} < 10$), it was 84.4%. In Table 6, overall recognition rates are presented together with rates from single frame recognition ($M = 1$). Compared with the case of $k = 0$, in which an average pattern was learned, the other cases improved drastically in terms of recognition rates. Although the cases where $k = 1/4, 1/2, 2$, and 4 also exhibited high recognition rates, the case of $k = 1$ was the most effective. Fig. 15 presents some examples of the recognition results where $k = 1$. Even small symbols were recognized if the generative learning method was combined with multiple frame integration.

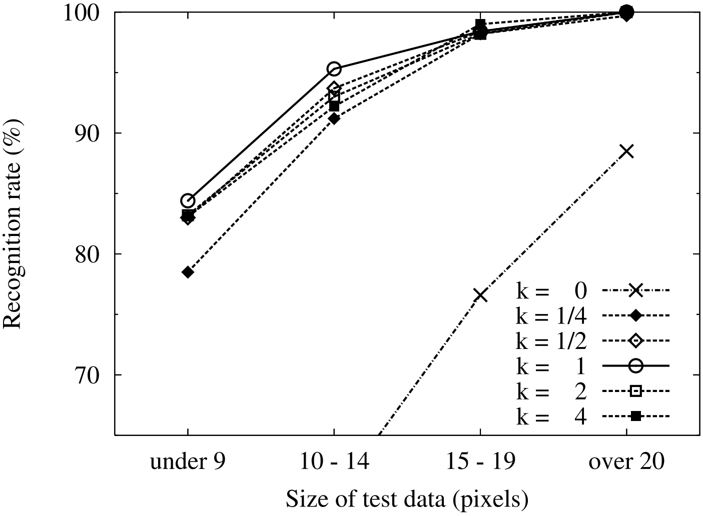


Fig. 14 Recognition results according to the maximum size of traffic sign symbol images in multiple frames

4.2 Discussion

The case using the estimated distribution ($k = 1$) was the most appropriate for recognizing traffic signs captured in similar conditions. This result indicates that the GA-based parameter estimation successfully worked.

Since most of the available traffic sign images are small as presented in Fig. 12, robustness to low-resolution images is important for real-world applications. Nevertheless, the recognition rate was not high enough when the image size was very small ($\bar{d}_{\max} < 10$). One reason is that small signs are especially sensitive to the degradation factors. It implies the dependency of parameters, which are listed in Eq. (9), on size parameter d. For the sake of simplicity, the current method assumes the independence of d from the other parameters. A better representation for parameter distribution should be discussed in future works.

Table 7 shows the recognition rates of the generative learning method ($k = 1$) for each data set. A sufficient performance should be obtained also from the case where different sets were used for estimation and testing. Non-diagonal elements in Table 7 show the results of such cases. However, sets A and C were not recognized with high accuracy, compared with the case where the same set was used both for estimation and testing (see Table 7 column-by-column). This is partly due to the distribution of traffic sign size in Fig. 12; set A was composed mostly of large images, and set C was composed mostly of small images. Moreover, the recognition rates were lower when sets D and E were used for parameter estimation (see Table 7 row-by-row). One explanation is that the parameter distribution was not satisfactorily estimated because of structural simplicity of the original traffic sign symbols.

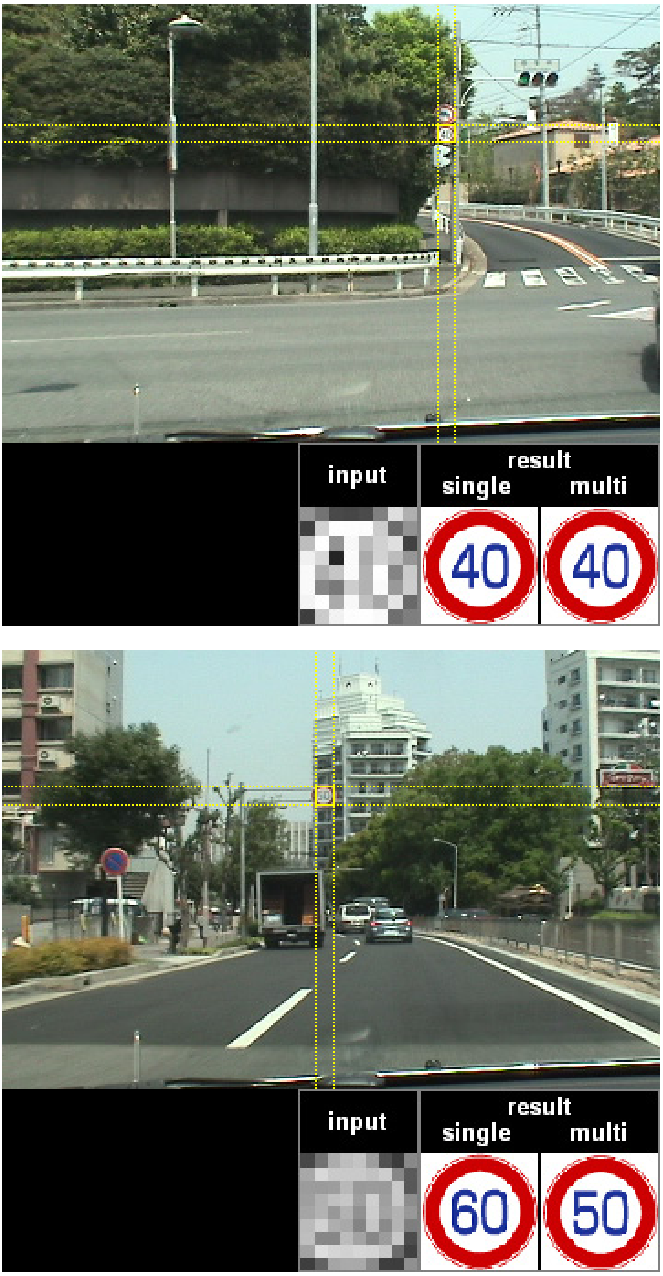


Fig. 15 Video stream demonstrating the recognition results. Traffic signs shown at the bottom of the images are the extracted symbol, the result of single-frame recognition ($M = 1$), and the result of multiple-frame recognition ($M = 10$), from left to right. Whereas the single-frame recognition sometimes gave incorrect results, the multiple-frame recognition gave the correct result at higher rates.

Table 7 Recognition rates of the generative learning method ($k = 1$) for each training and test set

	Recognition rates for test data [%]						
	Single frame						Multiple frames
Training data	Set A	Set B	Set C	Set D	Set E	Average	Average
Set A	97.1	68.0	68.2	98.6	100	82.3	93.5
Set B	97.7	78.4	72.0	100	100	86.9	95.8
Set C	93.7	82.6	82.7	100	100	89.7	98.5
Set D	91.4	83.1	65.4	98.6	100	85.8	91.5
Set E	89.1	76.4	56.1	98.1	100	81.3	90.0

Table 8 Edge density measured from original traffic sign symbol images of 56 × 56 pixels

Set	A	B	C	D	E
Category	No. 2	No. 4	No. 5	No. 12	No. 20
Edge density	0.064	0.061	0.065	0.046	0.055

Table 8 shows the complexities calculated for the traffic sign symbols, where the complexity is defined by edge density as introduced in [27]. Altogether, parameters should preferably be estimated from images of various sizes using structurally complex symbols.

5 Summary

In this chapter, a method for recognizing traffic sign symbols was introduced. Degradation parameters were defined in order to generate variously degraded training images. Based on the generated models, degradation characteristics were estimated from a small number of captured images. The estimated characteristics were trained via the generated images.

The presented framework is applicable for any traffic sign by combining it with conventional traffic sign detection methods [10]–[19]. In future works, the effectiveness under various weather conditions and at various times of day should be evaluated. Application to the recognition of other on-road objects and pedestrians will be interesting.

References

1. Murase, H.: Generative learning for image recognition. Trans. IPS Japan 46(SIG15, CVIM-12), 35–42 (2005) (in Japanese)
2. Ishii, K.: Generation of distorted characters and its applications. Systems and Computers in Japan 14(6), 19–27 (1983)
3. Horiuchi, T., Toraichi, K., Yamamoto, L., Yamada, H.: On method of training dictionaries for handwritten character recognition using relaxation matching. In: Proc. 2nd Int. Conf.

on Document Analysis and Recognition, Tsukuba, Japan, pp. 638–641 (October 1993)

4. Doermann, D., Liang, J., Li, H.: Progress in camera-based document image analysis. In: Proc. 5th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland, UK, pp. 606–616 (August 2003)
5. Andrew, H., Hunt, B.: Digital image restoration. Prentice-Hall, Englewood Cliffs (1977)
6. Oja, E.: Subspace methods of pattern recognition. Research Studies, Hertfordshire, UK (1983)
7. Bishop, R.: A survey of intelligent vehicle applications worldwide. In: Proc. IEEE 2000 Intelligent Vehicles Symp., Dearborn, MI, USA, pp. 25–30 (October 2000)
8. Geronimo, D., Lopez, A.M., Sappa, A.D., Graf, T.: Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans. PAMI 32(7), 1239–1258 (2010)
9. Kamijo, S., Okumura, K., Kitamura, A.: Digital road map database for vehicle navigation and road information systems. In: Proc. Int. Conf. on Vehicle Navigation and Information Systems, Toronto, ON, Canada, pp. 319–323 (September 1989)
10. Escalera, A., Salichs, M.: Road traffic sign detection and classification. IEEE Trans. Industrial Electronics 44(12), 848–859 (1997)
11. Gavrila, D.: Multi-feature hierarchical template matching using distance transforms. In: Proc. 14th IAPR Int. Conf. on Pattern Recognition, Brisbane, QLD, Australia, vol. 1, pp. 439–444 (August 1998)
12. Miura, J., Kanda, T., Shirai, Y.: An active vision system for real-time traffic sign recognition. In: Proc. IEEE 2000 Intelligent Transportation Systems, Dearborn, MO, USA, pp. 52–57 (June 2000)
13. Mo, G., Aoki, Y.: A recognition method for traffic sign in color image. IEICE Trans. J87-D-II(12), 2124–2135 (2004) (in Japanese)
14. Uchimura, K., Kimura, H., Wakiyama, S.: Extraction and recognition of circular road signs using road scene color images. IEICE Trans. J81-A(4), 546–553 (1998) (in Japanese)
15. Matsuura, D., Yamauchi, H., Takahashi, H.: Extracting circular road signs using specific color distinction and region limitation. IEICE Trans. J85-D-II(6), 1075–1083 (2002) (in Japanese)
16. Lafuente-Arroyo, S., Gil-Jimenez, P., Maldonado-Bascon, R., Lopez-Ferreras, F.: Traffic sign shape classification evaluation I: SVM using distance to borders. In: Proc. IEEE 2005 Intelligent Vehicles Symp., Las Vegas, NV, USA, pp. 557–562 (June 2005)
17. Gil-Jimenez, P., Lafuente-Arroyo, S., Gomez-Moreno, H., Lopez-Ferreras, F., Maldonado-Bascon, S.: Traffic sign shape classification evaluation II: FFT applied to the signature of blobs. In: Proc. IEEE 2005 Intelligent Vehicles Symp., Las Vegas, NV, USA, pp. 607–612 (June 2005)
18. Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., Koehler, T.: A system for traffic sign recognition, tracking, and recognition using color, shape, and motion information. In: Proc. IEEE 2005 Intelligent Vehicles Symp., Las Vegas, NV, USA, pp. 255–260 (June 2005)
19. Johansson, B.: Road sign recognition from a moving vehicle. Master's thesis, Center for Image Analysis, Swedish University of Agricultural Science (2002)
20. Ishida, H., Takahashi, T., Ide, I., Mekada, Y., Murase, H.: Generation of training data by degradation models for traffic sign symbol recognition. IEICE Trans. E90-D(8), 1134–1141 (2007)
21. Yanadume, S., Mekada, Y., Ide, I., Murase, H.: Recognition of very low-resolution characters from motion images captured by a portable digital camera. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3331, pp. 247–254. Springer, Heidelberg (2004)

22. Daris, L.: Handbook of genetic algorithms. Van Nostrand Reinhold, New York (1991)
23. von Neumann, J.: Various techniques used in connection with random digits. National Bureau of Standards Series (12), 36–38 (1951)
24. Golub, G., Loan, C.: Matrix computations, 2nd edn. Cambridge Univ. Press, Cambridge (1992)
25. MIST project, `http://mist.murase.m.is.nagoya-u.ac.jp/trac-en/`
26. Smith, A.: Color Gamut transform pairs. Computer Graphics 12(3), 12–19 (1978)
27. Hirayama, K., Omachi, S., Aso, H.: String extraction from scene images using color and luminance information. IEICE Trans. J-89-D(4), 893–896 (2006) (in Japanese)

List of Acronyms

PSF	Point Spread Function
GLM	Generative Learning Method
GA	Genetic Algorithm
CD	Cholesky Decomposition

Chapter 5
Local Structure Preserving Based Subspace Analysis Methods and Applications

Jian Cheng and Hanqing Lu

Abstract. Subspace analysis is an effective approach for image representation. Local structure preserving has been widely adopted to learn subspace which reflects the intrinsic attributes of samples. In this chapter, inspired by the idea of local structure preserving, we propose two novel subspace methods for face recognition and image clustering tasks. The first is named Supervised Kernel Locality Preserving Projections (SKLPP) for face recognition task, in which geometric relations are preserved according to prior class-label information and complex nonlinear variations of real face images are represented by nonlinear kernel mapping. The second is a novel probabilistic topic model for image clustering task, named Dual Local Consistency Probabilistic Latent Semantic Analysis (DLC-PLSA), The proposed DLC-PLSA model can learn an effective and robust mid-level representation in the latent semantic space for image analysis. As our model considers both the local image structure and local word consistency simultaneously when estimating the probabilistic topic distributions, the image representations can have more powerful description ability in the learned latent semantic space. The extensive experiments on face recognition and image clustering show that the proposed subspace analysis methods are promising.

1 Introduction

As one of the most critical stages in pattern recognition tasks, feature extraction and representation has attracted much effort in the last decades. Many feature descriptors have been introduced, such as color, texture, shape, SIFT, LBP [1,2]. As a result, the dimensionality of feature space increases to the scale of hundreds even thousands, namely the curse of dimensionality. To address the curse of dimensionality, subspace

Jian Cheng · Hanqing Lu
National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences
e-mail: {jcheng,luhq}@nlpr.ia.ac.cn

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_5, © Springer-Verlag Berlin Heidelberg 2014

analysis method is utilized to seek a much lower dimensional feature space with approximative description capability to the original feature space.

There are many subspace analysis algorithms in literature. Principal Component Analysis (PCA) is a popular subspace analysis method which assuming an image can be decomposed as a linear combination of basis images [3,4]. The basic idea of PCA is to maximally retain variance of samples in learned subspace. In viewpoint of statistics, PCA only satisfies the second-order statistical information independent, but high-order statistical dependencies still exist and cannot be properly separated. Independent Component Analysis (ICA) aims to seek for a set of basis images to make the image coordinates high-order statistically independent in this basis besides the second-order statistical independence used in PCA [5,6]. In this sense, PCA is a special case of ICA with Gaussian prior, and ICA can be considered as a generalization of PCA to non-gaussian scenario . There are many other algorithms, such as Linear Discriminant Analysis (LDA), also named Fisher's linear discriminant, which maximizes the variance of between-class while minimizing the variance of within-class [7]. These subspace analysis algorithms can be categorized by different criteria, such as linear or nonlinear, global or local. Most of the above-mentioned subspace methods can be solved in algebraic approach, e.g. matrix factorization. Another hot feature representation approach is probabilistic topic models, such as probability Latent Semantic Analysis (pLSA) [8], Latent Dirichlet Allocation (LDA) [9]. In contrast to algebraic method, these probability topic model methods can be solved in probability approach.

Recent studies show that there may exist a compact subspace (i.e. manifold) embedding in the original feature space for certain object or scene images, which can reflect the intrinsic distribution of object or scene images [10,11,12]. The representative manifold learning algorithms include Multidimensional scaling (MDS) [13], Locally Linear Embedding (LLE) [10], Isomap [11], Laplacian Eigenmap [12], Hessian LLE [14], etc. The most prominent characteristic of these manifold learning algorithms is the capability of preserving the structure information among samples when map the samples into the compact subspace. Among them, the structure preserving can be grouped into local structure preserving and global structure preserving. MDS and Isomap preserve the global structure in Euclidean space and geodesic space, respectively. In contrast, LLE and Laplacian Eigenmap are local structure preserving approaches. In this chapter, we focus on local structure preserving approach and present two new subspace analysis methods. One is nonlinear subspace method from algebraic approach, the other is probabilistic subspace analysis method derived from topic model. The extensive experiments on face recognition and image clustering show that the proposed subspace analysis methods are promising.

The rest of this chapter is organized as follows: section 2 will introduce a representative local structure preserving method LPP. Section 3 will apply local structure preserving to face recognition. The applications in image clustering will be presented in section 4. Finally, section 5 is conclusions.

2 Local Structure Preserving

Subspace analysis is an effective approach for feature representation. Specially learning a compact manifold (subspace) that can preserve local structure of image distribution has attracted a great deal of attention in the past few years. There are three popular manifold learning methods, i.e., Locally Linear Embedding (LLE) [10], Isomap [11], Laplacian Eigenmap [12], but these methods are not suitable for pattern recognition problems, because they cannot give an explicit subspace mapping for a new test sample. In order to overcome this drawback, He et al proposed a method, named Locality Preserving Projections (LPP) [15,16], to approximate the eigenfunctions of the Laplace Beltrami operator on the manifold, and new sample can be easily mapped to the learned low-dimensional feature subspace.

Locality Preserving Projections (LPP) is a linear approximation of Laplacian Eigenmap. It seeks a transformation $\mathbf{P}$ to project high-dimensional input data $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}]$ into a low-dimensional subspace $\mathbf{Y}$ in which the local structure of the input data can be preserved. The linear transformation can be obtained by minimizing an objective function as follows [15]:

$$\min_P \sum_{i,j=1}^{n} ||\mathbf{y}_i - \mathbf{y}_j||^2 \mathbf{S}(i,j) \tag{1}$$

where $y_i = \mathbf{P}^T x_i$, the weight matrix $\mathbf{S}$ (called heat kernel) is constructed through the nearest-neighbor graph. If $\mathbf{x}_i$ is among the l nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the l nearest neighbors of $\mathbf{x}_i$, then

$$\mathbf{S}(i,j) = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}} \tag{2}$$

where parameter t is a suitable constant. Otherwise, $\mathbf{S}(i,j) = 0$. Alternatively, the weight matrix can be simply set as: $\mathbf{S}(i,j) = 1$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ are the nearest neighbors, otherwise $\mathbf{S}(i,j) = 0$. The minimization problem can be converted to solve a generalized eigenvalue problem as follows:

$$\mathbf{XLX}^T\mathbf{P} = \lambda \mathbf{XDX}^T\mathbf{P} \tag{3}$$

where D is a diagonal matrix with the i-th diagonal element $\mathbf{D}_{ii} = \sum_j \mathbf{S}(i,j)$, and $\mathbf{L} = \mathbf{D} - \mathbf{S}$.

3 Local Structure Preserving for Face Recognition

Although LPP has achieved significant success in face recognition [16], it often fails to deliver good performance when face images are subject to complex nonlinear changes due to large pose, expression or illumination variations, for it is a linear method in nature. In this chapter, a novel subspace analysis method named Supervised Kernel Locality Preserving Projections (SKLPP) is proposed for face recognition[17]. Firstly, we use nonlinear kernel mapping to map the data into an implicit feature space F, which is successfully used in Support Vector Machine (SVM)[18]. Then we seek a linear transformation that can preserve within-class geometric structures in F. Thus, we can gain a nonlinear subspace that can approximate the intrinsic geometric structure of the face manifold. Though He, et al mentioned that LPP could be generalized into a reproducing kernel Hilbert space through a nonlinear mapping, it was not further discussed [15]. Moreover, LPP seeks to preserve local structure defined by the nearest neighbors. So it fails to preserve within-class local structure, which is very important for object recognition, because the nearest neighbors may belong to different classes due to influence of complex variations, such as lighting, expression and pose.

3.1 Supervised Kernel Locality Preserving Projections

LPP is a linear method in nature, and it is inadequate to represent the nonlinear face space. Moreover, LPP seeks to preserve local structure defined by the nearest neighbors. It often fails to preserve within-class local structure, which is very important for object recognition, because the nearest neighbors may belong to different classes due to influence of complex variations, such as lighting, expression, and pose. In this chapter, we propose a novel subspace method for face recognition, i.e., SKLPP. First, the nonlinear kernel mapping is used to map the data into an implicit feature space **F**, which is successfully used in Support Vector Machine (SVM), and then a linear transformation is performed to preserve within-class geometric structures in **F**. Thus, we can gain a nonlinear subspace that can approximate the intrinsic geometric structure of the face manifold.

Assuming a set of face images $\mathbf{X} = [x_1, x_2, \cdots, x_n]$, x_i is a N-dimensional face image. Firstly, we use a nonlinear function ϕ to map the data into a high-dimensional feature space $\mathbf{F} : \phi(X) = [\phi(x_1), \phi(x_2), \cdots, \phi(x_n)]$. Then in feature space **F**, we seek a projecting transformation $\mathbf{P}_\phi$ that can preserve the within-class geometric structure of the data $\phi(X)$ by minimizing the sum of the weighted distance of samples. The minimization problem can be expressed as:

$$\min_{\mathbf{P}_\phi} \sum_{i,j=1}^{n} ||Z_i - Z_j||^2 \mathbf{S}(i,j) \tag{4}$$

where $Z_i = \mathbf{P}_\phi^T \phi(x_i)$ is the projection of $\phi(x_i)$ onto $\mathbf{P}_\phi$, and the weight $\mathbf{S}(i,j)$ represents the relations of x_i and x_j. The objective function (4) can be simplified as:

$$\sum_{i,j=1}^{n} ||Z_i - Z_j||^2 S(i,j) = \sum_{i,j=1}^{n} ||P_\phi^T \phi(x_i) - P_\phi^T \phi(x_j)||^2 S(i,j) = 2P_\phi^T \phi(X)(D-S)\phi(X)^T P_\phi \tag{5}$$

where **D** is a diagonal matrix with i-th element $\mathbf{D}_{ii} = \sum_j \mathbf{S}(i,j)$. Because the linear transformation $\mathbf{P}_\phi$ should lie in the span of $\{\phi(x_1), \phi(x_2), \cdots, \phi(x_n)\}$, there exists a coefficient vector $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T$ such that

$$\mathbf{P}_{\phi} = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \phi(\mathbf{X})\alpha \tag{6}$$

Substituting (6) into (5), we can obtain:

$$\sum_{i,j=1}^{n} ||Z_i - Z_j||^2 \mathbf{S}(i,j) = 2\alpha^T \mathbf{K}(\mathbf{D} - \mathbf{S})\mathbf{K}\alpha \tag{7}$$

where the matrix $\mathbf{K}(i,j) = \phi(x_i) \cdot \phi(x_j)$ is a positive definite and symmetric matrix. According to the kernel trick, the dot produce of two vectors in $\mathbf{F}$ is calculated by a kernel function $k(x,y) = \phi(x) \cdot \phi(y)$ without knowing the nonlinear mapping ϕ explicitly.

Thus, this minimization problem can be converted to a generalized eigenvalue problem with a constraint condition $\alpha^T KDK\alpha = 1$. The eigenvectors corresponding to the smallest eigenvalues are the solution:

$$\mathbf{K}(\mathbf{D} - \mathbf{S})\mathbf{K}\alpha = \lambda \mathbf{KDK}\alpha \tag{8}$$

Up to now, the weight matrix **S** is still unknown. In [15,16], the weight matrix **S** is just defined by the nearest-neighbor relations. Here, with prior class label information, we define the **S** using a supervised approach. In fact, each entry of the weight matrix **S** can be regarded as the similarity metric of a pair of samples. The dot product between two samples is in a sense a similarity measure. So we define the weight matrix **S** as follows:

$$\mathbf{S}(i,j) = \begin{cases} \phi(x_i) \cdot \phi(x_j), & \text{if} x_i \text{ and } x_j \text{belong to the same class} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

It means that the within-class geometric information is emphasized, and we set the similarity between two samples to zero, if they belong to different classes. From (9), we find that the matrix **S** and **K** are unified into a consistent dot product form except that the matrix **S** has a strong constraint.

3.2 Experimental Results on Face Recognition

To verify the proposed method, SKLPP is applied to face recognition compared with LPP, PCA, LDA, KPCA[19], and KLDA[20]. The experiments are performed on two publicly available databases: Yale [3] and ORL [21]. The Yale database contains 165 grayscale face images from 15 persons. All face images are cropped into 80×90 . Both expression and lighting variations exist in the Yale database. The ORL database contains 40 persons, and each person has 10 different grayscale face images that include variations in pose and scale. The size of face images is 92×112. The gray values of all images are rescaled to $[0,1]$ and the norm of each image vector is normalized to 1. Figure 1 provides some samples from the Yale database and ORL database.

Table 1 Comparisons on Yale and ORL databases

Method	Dims	accuracy(Yale)	Dims	accuracy(ORL)
PCA	90	76.36	40	98.00
LPP	60	85.45	100	92.75
LDA	14	96.96	39	94.75
KPCA	100	76.96	30	97.25
KLDA	14	98.78	39	98.50
SKLPP	40	99.39	20	98.75

Fig. 1 The first row is sample of the Yale database, the second row is sample of the ORL database

It is well known that the kernel selection is still an open problem till now. In our experiments, we empirically adopts polynomial kernel as kernel function:

$$k(x,y) = \phi(x) \cdot \phi(y) = (\alpha(x \cdot y))^d \tag{10}$$

The parameters of the polynomial kernel are empirically set as: a = 0.1, d = 2. All experiments were conducted using the Leave-One-Out strategy. For simplicity, the nearest-neighbor classifier based on the Cosine distance metric is used.

$$d(z_i, z_j) = 1 - \frac{z_i^T \cdot z_j}{||z_i|| \cdot ||z_j||} \tag{11}$$

The compared recognition rates are shown in Table 1. SKLPP achieves the best recognition rate 99.39% with 40 dimensions while LPP only gets 85.45% with 60 dimensions on the YALE database. On the ORL database, SKLPP achieves 98.75% with 20 dimensions while LPP gets 92.75% with 100 dimensions. Because the maximum dimensions of LDA and KLDA are no more than c (c is the number of classes), the best recognition rates of LDA and KLDA are obtained with 14 dimensions on YALE and 39 dimensions on ORL, respectively. Experimental results suggest that SKLPP also outperforms the other methods. It demonstrates that the performance is significantly improved because SKLPP preserves the local structure information in kernel feature space.

4 Local Structure Preserving for Image Clustering

Image representation is one of the most fundamental components for image clustering and classification tasks. A large variety of features have been proposed to characterize the content of images [23,24]. However, these low-level features cannot correctly represent the semantic content of images in many situations due to the so-called “semantic gap”. Therefore, mid-level features are exploited by many researchers to bridge the gap. In recent years, the latent topic models, such as probabilistic Latent Semantic Analysis (pLSA)[8], and Latent Dirichlet Allocation (LDA)[9], have been proposed to address this problem. The above-mentioned topic models can discover hidden topics in the latent semantic space based on a bag-of-words representation for the images, which can connect the low-level features and high-level semantic content. Due to the success of topic models, they have been widely adopted in many applications such as image clustering, classification, and retrieval [24,25,26].

However, most applications treat the topic model as a black box and each image or word is treated independently in topic modeling. There are still few efforts paid to explore how these hidden topics distribute and what correlations exist among them. Therefore, the topic distributions estimated by the traditional models may not be accurate enough in some cases. According to recent research[10,11,12] , data from images or texts are often found to lie on a low-rank non-linear manifold embedded in the high-dimensional space of the original data. Therefore, exploiting the intrinsic structure concealed in the data can help discover more accurate latent topics[27,28]. Moreover, the words frequently co-occured in one image or text often have similar meanings and should be related to similar latent topics with a high probability. Thus, the word co-occurring information is also very essential to reveal the hidden semantics in the data.

To address the above issues, in this chapter, we present a novel probabilistic topic model, named Dual Local Consistency Probabilistic Latent Semantic Analysis (DLC-PLSA)[29], to model the latent topics with sparse neighborhood preserving embedding and local word consistency. Compared with the traditional models, our model has the following characteristics: (1) ℓ^1-graph is constructed to model the sparse neighborhood structure of images and embedded into topic modeling. (2) the word co-occurring information is first incorporated into topic models to help discover more accurate latent topics. In this way, the topic model can estimate the probabilistic topic distributions and simultaneously consider the image neighborhood structure as well as the local word consistency in a uniform formulation. Therefore, our model is less sensitive to noise and has more discriminative power in the latent semantic space.

4.1 pLSA with Local Structure Preserving

In most of the traditional topic models, the images (or visual words) are treated individually and there are few efforts to explore the intrinsic structure existed among them. However, such local structure preserving is very important and helpful in discovering more accurate latent topics. In this section, we present a novel topic model, named Dual Local Consistency Probabilistic Latent Semantic Analysis (DLC-PLSA). Different from the traditional topic models, our model considers the sparse image neighborhood structure and local word consistency simultaneously when estimating the latent topic distributions. Therefore, it can preserve more structure semantic information in the latent semantic space. Next, we will introduce how to embed the local structure of images and words into topic discovering in this subsection.

4.1.1 Sparse Neighborhood Consistency

In this part, we present a novel manifold learning approach based on traditional NPE method [30]. Our method is motivated by the limitations of classical graph construction methods on robustness to data noise and data-adaptiveness, and recent advances in sparse coding [31,32,33]. With sparse representation, each sample can be reconstructed by the sparse linear superposition of the training data. The sparse reconstruction coefficients, used to deduce the weights of the ℓ^1-graph, are derived by solving an ℓ^1 optimization problem on sparse representation. Recent work in [34] has shown the ℓ^1-graph is superior to the classical graphs in various machine learning tasks such as image clustering and subspace learning.

Suppose we have an underdetermined system of linear equations: $x = D\alpha$, where $x \in R^m$ is the vector to be approximated, $\alpha \in R^n$ is the vector for unknown reconstruction coefficients, and $D \in R^{m\times n}$ is the overcomplete dictionary with n bases. Generally, a sparse solution is more robust and is able to facilitate the consequent identification of the test sample x. We seek the sparse solution to $x = D\alpha$ by solving the following optimization problem:

$$\min_{\alpha} ||\alpha||_1, \quad s.t. \quad x = \mathbf{D}\alpha \tag{12}$$

This problem can be solved in polynomial time by standard linear programming method. In practice, there may exist noises on certain elements of x, and a natural way to recover these elements and provide a robust estimation of α is to formulate

$$x = \mathbf{D}\alpha + \zeta = \begin{bmatrix}\mathbf{D}\ I\end{bmatrix}\begin{bmatrix}\alpha \\ \zeta\end{bmatrix} \tag{13}$$

where $\zeta \in R^m$ is the noise term. Then by setting $\mathbf{B} = \begin{bmatrix}\mathbf{D}\ \mathbf{I}\end{bmatrix} \in R^{m\times(m+n)}$ and $\alpha' = \begin{bmatrix}\alpha \\ \zeta\end{bmatrix}$, we can solve the following ℓ^1-norm minimization problem with respect to both reconstruction coefficients and data noises:

$$\min_{\alpha'} \|\alpha'\|_1, \quad s.t. \quad x = \mathbf{B}\alpha' \tag{14}$$

An ℓ^1-graph summarizes the overall behavior of the whole dataset in sparse representation. The construction process is stated as follows.

1) **Inputs:** The image set denoted as the matrix $X = [x_1, x_2, \ldots, x_N]$, where $x_i \in \mathbf{R}^M$

2) **Robust sparse representation:** For each image x_i in the dataset, its robust sparse representation is achieved by solving the ℓ^1-norm optimization problem

$$\min_{\alpha^i} \|\alpha^i\|_1, \quad s.t. \quad x_i = B^i \alpha^i \tag{15}$$

where $B^i = [x_1, \ldots x_{i-1}, x_{i+1} \ldots, x_N, I] \in \mathbf{R}^{M \times (M+N-1)}$ and $\alpha^i \in \mathbf{R}^{M+N-1}$

3) **Graph weight setting:** Denote the $G = \{X, W\}$ as the ℓ^1-graph with the image set X as graph vertices and W as the graph weight matrix. We set $W_{ij} = \alpha^i_j$ if $i > j$, and $W_{ij} = \alpha^i_{j-1}$ if $i < j$.

After the ℓ^1-graph is constructed, the neighborhood structure of the image dataset as well as the graph weights is derived simultaneously in a parameter-free manner. Then, similar to NPE, sparse neighborhood preserving embedding aims to preserve the neighborhood structure of the dataset in the latent topic space by minimizing

$$R_1 = \sum_{k=1}^{K} R_{1k} = \sum_{k=1}^{K} \sum_{i=1}^{N} (P(z_k|x_i) - \sum_{j=1}^{N} W_{ij} P(z_k|x_j))^2 \tag{16}$$

An intuitive explanation of minimizing R_1 is that if the image x_i can be reconstructed by its neighbors in the feature space, the intrinsic structure should also be preserved in the latent topic space.

4.1.2 Local Word Consistency

Besides the image-level local structure, the word consistency is usually ignored and each word is treated individually in the existing topic models. However, the local word consistency is also very important for topic modeling. For example, it is a natural and intuitive assumption that frequently co-occurring words should share similar topics in the latent space. In this part, we will introduce how to maintain the local word consistency in our topic model.

We first compute the co-occurrence information C_{ij} between word w_i and word w_j as follows:

$$C_{ij} = \frac{f_{ij}}{\sqrt{f_i} * \sqrt{f_j}} \tag{17}$$

where f_{ij} is the number of images in which both word w_i and word w_j appeared and f_i is the number of images in which word w_i appeared.

After we get the co-occurrence matrix C, we maintain the local word consistency in the latent topic space by minimizing

$$R_2 = \sum_{k=1}^{K} R_{2k} = \sum_{k=1}^{K} \sum_{i,j=1}^{M} (P(w_i|z_k) - P(w_j|z_k))^2 C_{ij} \tag{18}$$

An intuitive explanation of minimizing R_2 is that if the word w_i often co-occurred with w_j, their conditional distributions related to the latent topic z_k should also be similar in the latent topic space.

4.1.3 The Regularized Model

In order to consider the local image and word structure simultaneously, we add R_1 and R_2 as regularized terms to the log-likelihood of PLSA model. Then we get our new latent topic model which aims to maximize the regularized log-likelihood as follows:

$$L = l - \lambda_1 R_1 - \lambda_2 R_2 = l - \lambda_1 \sum_{k=1}^{K} R_{1k} - \lambda_2 \sum_{k=1}^{K} R_{2k} \tag{19}$$

where $n(x_i, w_j)$ specifies the number of times the word w_j occurred in image x_i, and $\lambda_{1,2}$ are the regularized parameters. When $\lambda_1 = \lambda_2 = 0$, our model degenerates to the traditional pLSA model. When $\lambda_1 = 0$, our model only considers the local word consistency. When $\lambda_2 = 0$, only the sparse neighborhood structure is preserved.

4.2 Model Fitting

When a probabilistic model involves unobserved latent variables, the EM algorithm is generally used for the maximum likelihood estimation of the model. EM alternates two steps: (i) an expectation (E) step where posterior probabilities are computed for the latent variables, based on the current estimates of the parameters, (ii) a maximization (M) step, where parameters are updated based on maximizing the so-called expected complete data log-likelihood which depends on the posterior probabilities computed in the E-step.

As there are regularization terms in the log-likelihood of our model, the traditional EM algorithm cannot be applied directly. Here we use the generalized EM algorithm [35] for parameter estimation. The main difference between generalized EM and traditional EM is that generalized EM algorithm finds parameters that "improve" the expected value of the log-likelihood function rather than "maximizing" it.

Let $\phi = \{P(w_j|z_k)\}$ and $\theta = \{P(z_k|x_i)\}$ denote the parameters in our model.

E-step:

Our model adopts the same generative scheme as that of pLSA. Thus, we have the same E-step as that of pLSA. The posterior probabilities for latent variables are $P(z_k|x_i, w_j)$, which can be computed as follows:

$$P(z_k|x_i, w_j) = \frac{P(w_j|z_k)P(z_k|x_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|x_i)} \tag{20}$$

M-step:

The relevant part of the expected complete log-likelihood for our model is

$$\begin{aligned} Q(\phi, \theta) &= Q_1(\phi, \theta) - \lambda_1 R_1(\theta) - \lambda_2 R_2(\phi) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} n(x_i, w_j) \log \sum_{k=1}^{K} P(w_j|z_k)P(z_k|x_i) \end{aligned}$$

$$-\lambda_1 \sum_{k=1}^{K} \sum_{i=1}^{N} (P(z_k|x_i) - \sum_{j=1}^{N} W_{ij} P(z_k|x_j))^2$$

$$-\lambda_2 \sum_{k=1}^{K} \sum_{i,j=1}^{M} (P(w_i|z_k) - P(w_j|z_k))^2 C_{ij}$$

In the M-step, we improve the expected value of the log-likelihood function $Q(\phi,\theta)$. We have parameter values $\{\phi_r,\theta_r\}$ and try to find $\{\phi_{r+1},\theta_{r+1}\}$ which satisfy $Q(\phi_{r+1},\theta_{r+1}) \geq Q(\phi_r,\theta_r)$ in each step.

We first find $\{\phi_{r+1}^{(1)},\theta_{r+1}^{(1)}\}$ which maximizes $Q_1(\phi,\theta)$ instead of the whole $Q(\phi,\theta)$. This can be done by the following equations which are the M-step reestimation of pLSA:

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(x_i,w_j) P(z_k|x_i,w_j)}{\sum_{i=1}^{N} \sum_{m=1}^{M} n(x_i,w_m) P(z_k|x_i,w_m)} \tag{21}$$

$$P(z_k|x_i) = \frac{\sum_{j=1}^{M} n(x_i,w_j) P(z_k|x_i,w_j)}{\sum_{j=1}^{M} n(x_i,w_j)} \tag{22}$$

Clearly, $Q(\phi_{r+1}^{(1)},\theta_{r+1}^{(1)}) \geq Q(\phi_r,\theta_r)$ does not necessarily hold. We then try to start from $\{\phi_{r+1}^{(1)},\theta_{r+1}^{(1)}\}$ and decrease R_1 and R_2, which can be done through Newton-Raphson method [38]. Note that R_1 only involves parameters $P(z_k|x_i)$ while R_2 only involves parameters $P(w_j|z_k)$, we can update ϕ_{r+1} and θ_{r+1} respectively.

Given a function $f(x)$ and the initial value $x^{(t)}$, the Newton-Raphson updating formula to decrease (or increase) $f(x)$ is as follows:

$$x^{(t+1)} = x^{(t)} - \gamma \frac{f'(x^{(t)})}{f''(x^{(t)})} \tag{23}$$

where $0 \leq \gamma \leq 1$ is the step parameter. Since we have $R_{1k} \geq 0, R_{2k} \geq 0$, the Newton-Raphson method will decrease R_{1k} and R_{2k} in each updating step. With $\phi_{r+1}^{(1)}$ and put R_{1k} into the Newton-Raphson updating formula in Eqn. (25), we can get the closed form solution for $\phi_{r+1}^{(2)}, \phi_{r+1}^{(3)}, \ldots, \phi_{r+1}^{(m)}$, where

$$P(z_k|x_i)_{r+1}^{(t+1)} = (1-\gamma_1) P(z_k|x_i)_{r+1}^{(t)} + \gamma_1 \sum_{j=1}^{N} W_{ij} P(z_k|x_j)_{r+1}^{(t)} \tag{24}$$

Similarly, we can also get the updating equation for θ_{r+1} as follows:

$$P(w_i|z_k)_{r+1}^{(t+1)} = (1-\gamma_2) P(w_i|z_k)_{r+1}^{(t)} + \gamma_2 \frac{\sum_{j=1}^{M} C_{ij} P(w_j|z_k)_{r+1}^{(t)}}{\sum_{j=1}^{M} C_{ij}} \tag{25}$$

Every iteration of Eqn. (24) and (25) will make the topic distribution smoother. We continue the iteration of Eqn. (24) and (25) until $Q(\phi_{r+1}^{(t+1)},\theta_{r+1}^{(t+1)}) \leq Q(\phi_{r+1}^{(t)},\theta_{r+1}^{(t)})$. Then we test whether $Q(\phi_{r+1}^{(t)},\theta_{r+1}^{(t)}) \geq Q(\phi_r,\theta_r)$. If not, we reject the values of $\{\phi_{r+1}^{(t)},\theta_{r+1}^{(t)}\}$, and return the $\{\phi_r,\theta_r\}$ as the result of the M-step, and continue with the next E-step. The E-step and M-step are iteratively performed until the probability values are stable.

4.3 Experimental Results on Image Clustering

In this section, we evaluate the performance of our model by comparing it with the state-of-the-art methods on image clustering task. Clustering is one of the most crucial techniques to organize the data samples. The latent topics discovered by the topic modeling approaches can be regarded as clusters. By representing the images in the latent space, topic models can assign each image to the most probable latent topic according to the estimated conditional probability distributions $P(z_k|x_i)$. Our experiments are conducted on two publicly available datasets: the Binary Alphadigits [1], and the Caltech-101 dataset [36]. The weighting parameters λ_1 and λ_2 are tuned with cross validation from intervals [1,100] and [1000,1500] respectively. The values of the Newton step parameter γ_1 and γ_2 are both set to 0.1 in our experiment.

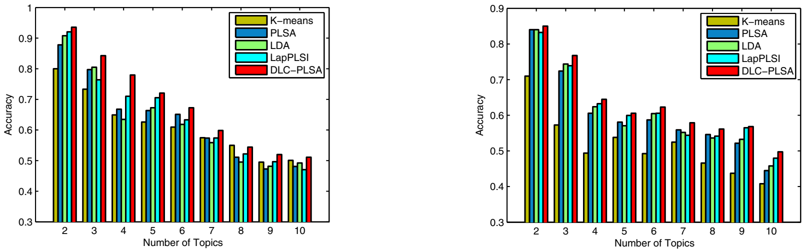


Fig. 2 Clustering results on (a) the Binary Alphadigits, and (b) the Caltech-101 dataset

The Binary Alphadigits contains binary 20x16 digits of ‘0’ through ‘9’ and capital ‘A’ through ‘Z’ where there are 39 examples of each class. Thus we have 1404 images from 36 classes in total with each image represented by a 320-dimensional binary pixel vector. The topic models are applied to the images by representing each binary pixel as a word and each image as a document to generate K clusters. The Caltech-101 dataset involves 9144 images from 101 object categories and a background category. A unique label has been assigned to each image in the datasets to indicate which category it belongs to, which serves as the ground truth for performance evaluation. SIFT [2] features are extracted and a 1000-D bag-of-words representation is generated for Caltech-101 dataset. Then all the models are performed on the bag-of-words to generate K clusters. The clustering accuracy (AC) is used to measure the clustering performance [37].

We evaluated the proposed DLC-PLSA model and compared it with the following algorithms: K-means clustering algorithm (K-means), Probabilistic Latent Semantic Analysis (pLSA) [8], Latent Dirichlet Allocation (LDA) [9], Laplacian Probabilistic Latent Semantic Indexing (LapPLSI) [27].

[1] http://www.cs.nyu.edu/~roweis/data.html

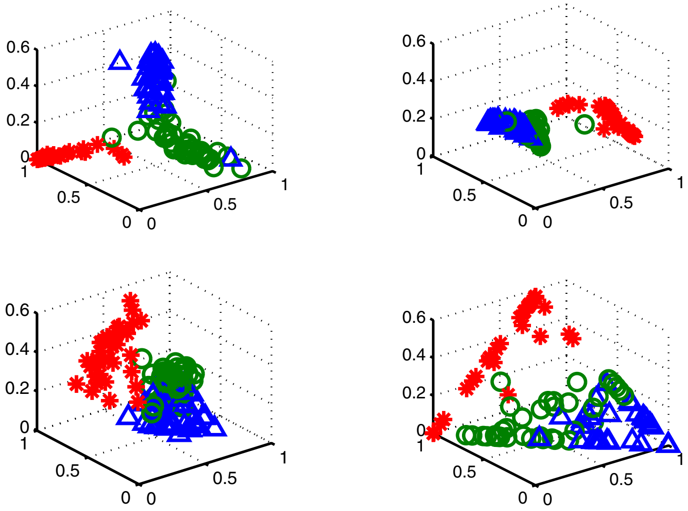


Fig. 3 Visualization view of image distribution in the latent topic space learned by different topic models. (a)The DLC-PLSA model. (b) The LapPLSI model. (c)The LDA model. (d)The PLSA model. (digit characters 'A' - 'C' in the Binary Alphadigits dataset, 'A': blue, 'B': green, 'C': red).

Table 2 The influence of different regularized terms on the clustering accuracy of the Binary Alphadigits

Topic number	2	4	6	8
$\lambda_1 \neq 0, \lambda_2 = 0$	0.926	0.722	0.648	0.528
$\lambda_1 = 0, \lambda_2 \neq 0$	0.915	0.751	0.669	0.510
$\lambda_1 \neq 0, \lambda_2 \neq 0$	**0.935**	**0.780**	**0.673**	**0.544**

Table 3 The influence of different regularized terms on the clustering accuracy of the Caltech-101 dataset

Topic number	2	4	6	8
$\lambda_1 \neq 0, \lambda_2 = 0$	**0.850**	0.617	0.601	0.553
$\lambda_1 = 0, \lambda_2 \neq 0$	0.844	0.626	0.613	0.559
$\lambda_1 \neq 0, \lambda_2 \neq 0$	**0.850**	**0.644**	**0.623**	**0.562**

In order to make the experiments statistically meaningful, we conducted the evaluations with the cluster numbers ranging from two to ten. For each given cluster number k, k different categories were randomly selected from the datasets and provided to the clustering algorithms. Five test runs were conducted for each k, and the final performance scores were obtained by averaging the sores over the five test runs.

Figure 2 shows the clustering performance of all the algorithms on the Binary Alphadigits and the Caltech-101 dataset, respectively. We can see that the topic models achieve better performance than the traditional K-means clustering method on the whole. But the pLSA and LDA model show lower performance than LapPLSI and DLC-PLSA because they do not consider any local discriminant structure when discovering the latent topics. Although the LapPLSI model considers the proximity between image pairs, it is not robust enough and sometimes even gets worse results than pLSA and LDA. The reason is that a K-NN graph is simply constructed to model the image structure, which is very sensitive to noise, and the local word consistency is also ignored. Therefore, it cannot reach full discriminant power. Our DLC-PLSA model, which constructs the ℓ^1-graph to model the neighborhood structure of images and incorporates the local word consistency into the model at the same time, can perform consistently better than other models.

Table 4 The impact of noisy images on clustering accuracy (digit characters ‘A’, ‘B’, ‘C’) of different models

	DLC-PLSA	LapPLSI	LDA	PLSA
Without removing the noisy images	0.8718	0.6667	0.7949	0.7179
Removing the noisy images	0.8803	0.8034	0.8291	0.8205

In order to evaluate the performance of different regularized terms, we also compare the results of our model with different regularized terms by setting the regularization parameter λ_1=0 or λ_2=0 respectively on different topic numbers. The comparison results are shown in Table 2-4, from which we can see that the model with both the regularized terms always perform as well as or outperform the better one with only one regularized term. Moreover, the model with only the word regularized term also performs very consistently and sometimes get better results than the image regularized term, which proves that the local word consistency is also very important in topic modeling.

The visualization comparison of image distribution (digit characters ‘A’ - ‘C’ in the Binary Alphadigits dataset) in the latent topic space is shown in Figure 3. The comparison results show that the embedded representations of DLC-PLSA, which models the hidden topics with sparse neighborhood and local word consistency, have the best separability. Although LapPLSI considers the proximity of image pairs in topic modeling, it cannot separate characters ‘A’ and ‘B’ very well. We analyzed the reason and found that three ‘noisy’ images of character ‘B’ were very easily considered as neighbors by most images of character ‘A’ when constructing the K-NN graph, which affects the whole local structure significantly. In order to further show the impact of the noisy images on the performance of different models, we conduct an experiment by manually removing some noisy images in these three classes and the comparison clustering results are given in Table 4. We can see that the performance of LapPLSI is affected greatly by the noisy images, because it models the image structure on the K-NN graph. After removing the noisy images, its performance has a big improvement. In contrast, our model constructs ℓ^1-graph to model the neighborhood structure of images and it can perform very consistently in both cases, which indicates that our model is more robust to noise and can discover more accurate latent topics.

5 Conclusions

Subspace analysis is an efficient and effective approach for compact feature representation. In this chapter, inspired by the idea of local structure preserving, we propose two local structure preserving based subspace analysis methods, SKLPP and DLC-PLSA. The two methods are utilized to face recognition and image clustering, respectively. The experimental results show that the two methods are promising to handle some common variations, such as noise, pose, lighting.

References

1. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transaction on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)
2. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
3. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
4. Turk, M., Pentland: Pentland Face recognition using eigenfaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)
5. Hyvarinen, A.: Survey on independent component analysis. Neural Computing Surveys 2, 94–128 (1999)
6. Comon, P.: Independent component analysis: A new concept. Signal Processing 36(3), 287–314 (1994)
7. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7(2), 179–188 (1936)
8. Hoffman, T.: Probabilistic latent semantic analysis. In: Uncertainty in Artificial Intelligence, pp. 289–296 (1999)
9. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
10. Roweis, S., Saul, L.: Nonlinear dimensionalityreduction by locally linear embedding. Science 290, 2323–2326 (2000)
11. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
12. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of Advances in Neural Information Processing System, vol. 14, pp. 585–591 (2001)
13. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman and Hall (2001)
14. Donoho, D.L., Grimes, C.: Hessian Eigenmaps: Locally Linear Embedding Techniques for High Dimensional Data. Proceedings of the National Academy of Sciences of the United States of America 100(10), 5591–5596 (2003)

15. He, X., Niyogi, P.: Locality preserving projections. In: Proceedings of Advances in Neural Information Processing Systems, Vancouver, Canada, vol. 16 (2003)
16. He, X., Yan, S., Hu, Y., Zhang, H.-J.: Learning a locality preserving subspace for visual recognition. In: Proceedings of Ninth International Conference on Computer Vision, pp. 385–392 (2003)
17. Cheng, J., Liu, Q., Lu, H., Chen, Y.-W.: Supervised kernel localitypreserving projections for face recognition. Neuro Computing 67, 443–449 (2005)
18. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
19. Yang, M.H., Ahuja, N., Kriegman, D.: Face recognition using kernel eigenfaces. In: Proceedings of International Conference on Image Processing, Vancouver, Canada, pp. 37–40 (2000)
20. Liu, Q., Huang, R., Lu, H., Ma, S.: Face recognition using kernel based fisher discriminant analysis. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Washington, pp. 197–201 (2002)
21. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, Sarasota, USA (December 1994)
22. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)
23. Li, Z., Yang, Y., Liu, J., Zhou, X., Lu, H.: Unsupervised feature selection using nonnegative spectral analysis. In: Proceeding of Association for the Advancement of Artificial Intelligence (2012)
24. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via PLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
25. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: Proceeding of ACM International Conference on Multimedia, pp. 348–351 (2004)
26. Cao, L., Li, F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: Proceeding of IEEE International Conference on Computer Vision, pp. 1–8 (2007)
27. Cai, D., Mei, Q., Han, J., Zhai, C.: Modeling hidden topics on document manifold. In: Proceeding of Conference on Information and Knowledge Management, pp. 911–920 (2008)
28. Cai, D., Wang, X., He, X.: Probabilistic dyadic data analysis with local and global consistency. In: Proceeding of International Conference on Machine Learning, pp. 105–112 (2009)
29. Li, P., Cheng, J., Lu, H.: Modeling Hidden Topics with Dual Local Consistency for Image Analysis. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 648–659. Springer, Heidelberg (2013)
30. He, X., Cai, D., Yan, S., Zhang, H.: Neighborhood preserving embedding. In: Proceeding of IEEE International Conference on Computer Vision, pp. 1208–1213 (2005)
31. Donoho, D.: For most large underdetermined systems of linear equations the minimal.1-norm solution is also the sparsest solution. Communications on Pure and applied Mathematics 59, 797–829 (2004)
32. Meinshansen, N., Buhlmann, P.: High-dimensional graphs and variable selection with the lasso. The Annals of Statistics 34, 1436–1462 (2006)
33. Wright, J., Genesh, A., Yang, A., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 210–227 (2009)

34. Cheng, B., Yang, J., Yan, S., Fu, Y., Huang, T.: Learning with.1-graph for image analysis. IEEE Transactions on Image Processing 19, 858–866 (2010)
35. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in Graphical Models. Kluwer Academic Publishers (1998)
36. Li, F., Rob, F., Pietro, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceeding of CVPR Workshop on Generative Model Based Vision, pp. 178–178 (2004)
37. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceeding of ACM Special Interest Group on Information Retrieval, pp. 267–273 (2003)
38. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical recipes in C: the art of scientific computing. Cambridge University Press (1992)

List of Acronyms

DLC-PLSA	Dual Local Consistency Probabilistic Latent Semantic Analysis
EM	Estimation Maximization
ICA	Independent Component Analysis
KPCA	Kernel Principal Component Analysis
KLDA	Kernel Linear Discriminant Analysis
LapPLSI	Laplacian Probabilistic Latent Semantic Indexing
LBP	Local Binary Patterns
LDA	Latent Dirichlet Allocation
LDA	Linear Discriminant Analysis
LLE	Locally Linear Embedding
LPP	Locality Preserving Projections
MDS	Multidimensional Scaling
PCA	Principal Component Analysis
pLSA	probability Latent Semantic Analysis
SIFT	Scale-Invariant Feature Transform
SKLPP	Supervised Kernel Locality Preserving Projections
SVM	Support Vector Machine

Chapter 6
Sparse Representation for Image Super-Resolution

Xian-Hua Han and Yen-Wei Chen

Abstract. This chapter concentrates the problem of recovery a high-resolution (HR) image from a single low-resolution input image. Recent research proposed to deal with the image super-resolution problem with sparse coding, which is based on the well reconstruction of any local image patch by a sparse linear combination of an appropriately chosen over-complete dictionary. Therein the chosen LR (Low-resolution) and HR (High-resolution) dictionaries have to be exactly corresponding for well reconstructing the local image patterns. However, the conventional sparse coding based image super-resolution usually achieves a global dictionary $\mathbf{D}$=[$\mathbf{D}_l$; $\mathbf{D}_h$] by jointly training the concatenated LR and HR local image patches, and then reconstruct the LR and HR image as a linear combination of the separated $\mathbf{D}_l$ and $\mathbf{D}_h$. This strategy only can achieve the global minimum reconstructing error of LR and HR local patches, and usually cannot obtain the exactly corresponding LR and HR dictionaries. In addition, the accurate coefficients for reconstructing the HR image patch using HR dictionary $\mathbf{D}_h$ are also unable to be estimated using only the LR input and the LR dictionary $\mathbf{D}_l$. Therefore, this paper proposes to firstly learn the HR dictionary $\mathbf{D}_h$ from the features of the training HR local patches, and then propagates the HR dictionary to the LR one, called as HR2LR dictionary propagation, by mathematical proving and statistical analysis. The effectiveness of the proposed HR2LR dictionary propagation in sparse coding for super-resolution is demonstrated by comparison with the conventional super-resolution approaches such as sparse coding and interpolation.

1 Introduction

Sparse signal representation[1-8] has been proven to be a greatly powerful algorithm for coding, representing, and compressing high-dimensional signals. It is well

Xian-Hua Han · Yen-Wei Chen
Ritsumeikan University, 1-1-1, NojiHigashi, Kusatus-Shi, Shiga-ken, 525-8577, Japan
e-mail: hanxhua@fc.ritsumei.ac.jp, chen@is.ritsumei.ac.jp

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_6, © Springer-Verlag Berlin Heidelberg 2014

known that the important properties of signals such as audio and images have naturally sparse representations with respect to fixed basis (i.e., Fourier, Wavelet), or concatenations of such basis. However, the Fourier, Wavelet basis etc. are mathematically fixed, and universal to any signal, and then cannot be adaptive to the processed signal. Therefore, researches on adaptively learning basis from the processed signals are actively taken on since 1990's. The most popular strategies for achieving adaptive basis to represent data mainly include Principle Component Analysis (PCA)[9-10], Independent Component Analysis (ICA)[11-14] and so on. PCA is a mathematical procedure that learns an orthogonal transformation from the proposed signal to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. ICA [14-17] is a method to find a linear nonorthogonal coordinate system in any multivariate data. The directions of the axes in this ICA coordinate system are determined by not only the second but also higher order statistics of the original data, unlike the principle component analysis (PCA), which considers only the second order statistics and can only deal with the variables that have Gaussian distributions. In computer vision, it is more preferable to extract the source signals produced by independent causes or obtained from different sensors; such signals are easily solved using ICA. These two classical adaptive base-learning strategies (PCA and ICA based) usually only produce non-overcomplete (the number of basis equals or is less than the dimension of the processed signal) basis, and then require to use all the learned basis for well representing the observed signal. In the other hand, understand processes in retina and primary visual cortex (V1) of human being [18] has been elucidated that early visual processes compress input into a more efficient form by activating only a few receptive fields in millions, which in mathematical theory can transfer this mechanism into sparse coding by learning an over-complete basis and only using a few of basis (sparsity) to represent the observed signal.

Thanks to the success of sparse coding strategy on representing, compressing high-dimensional signal, it is widely applied to pattern recognition, computer vision, image representation and so on, and has been proven its powerful advantage over conventional adaptive basis learning approaches such as PCA and ICA. Given only unlabeled input data in sparse coding, it learns basis functions that capture higher-level features in the data. When sparse coding is applied for natural image representation, the learned basis resemble the receptive fields of neurons in the visual cortex [1, 2]; in addition, sparse coding can also produce localized basis when applied to other natural stimuli such as speech and video [3, 4]. Different to the conventional unsupervised learning techniques such as PCA, sparse coding can learn overcomplete basis sets, in which the number of basis is larger than the dimensionality of the feature space. In order to learn the adaptive basis functions and achieve the sparse coefficients from the observed data, we use the popular strategy: the K-SVD algorithm [6, 19], a generalized algorithm from the K-Means clustering process. K-SVD is an iterative method that alternates between sparse coefficient calculation (sparse coding) of the observed signal based on the current dictionary, and a process of updating the dictionary atoms to better represent the data. The update of the dictionary columns is combined with an update of the sparse representations, thereby accelerating convergence. Furthermore, the K-SVD algorithm is flexible and can work with any pursuit method (e.g., basis pursuit, matching pursuit) for achieving adaptive basis and sparse coefficients. In this chapter, we use K-SVD and orthogonal matching pursuit (OMP) [20-23]- a smart improved version of matching pursuit [24-27] for dictionary updating and sparse coefficient calculation, and then apply the sparse representation for image super-resolution.

Super-Resolution (SR) is to generate a high resolution image from one or more low resolution input images. The super-resolution techniques are recently becoming a hot research topic due to many demanding applications such as biometric identification [27-28], medical imaging [29-30], remote sensing [31-32], etc.. There are mainly two types of super-resolution frameworks: the multiple-image super-resolution, which has several available low-resolution images with sub-pixel translation and rotation; and the single-image super-resolution, which has only one LR image. In this chapter, we focus on image super-resolution for a single image using the learning-based method, which can recover the lost information in LR images by exploring the co-occurrence prior between lots of available existing LR and HR image patches. The basic idea of learning-based super-resolution is to deduce the lost information by learning from training samples, which comprise HR and LR image pairs. In [33], Freeman et al. proposed an example-based super-resolution method to infer the HR images by the corresponding relationship of the prepared training HR and LR images pair, whose LR one is most similar to the input LR one. Stephenson extended this approach to predict the HR image from the LR one using Markov Random Field (MRF) solved by belief propagation [34]. However, the above methods typically require a huge amount of HR and LR training patch pair as prepared database, which makes ineffectiveness and un-efficiency for generating a HR image from the LR one. In [35], Locally Linear Embedding (LLE) [36] as a famous manifold learning was adopted to reconstruct the HR image patch as a linear combination of the training HR ones by mapping the local geometry of the LR space to the HR one, assuming similarity between two manifolds in the HR and LR patch spaces. With this strategy, more patch patterns can be represented using a moderate amount of training database, but usually results in blurring effect using the linear combination of the nearest K raw neighborhood patches due to the large variation of raw image patches. Then, Yang etc. [37-38] proposed to learn a structural dictionary using sparse coding, which can well reconstruct any image patch as a linear combination of several similar structural atoms in the learned dictionary. However, the conventional sparse coding based image super-resolution usually achieves a global dictionary $\mathbf{D}$=[$\mathbf{D}_l$; $\mathbf{D}_h$] by jointly training the concatenated LR and HR local image patches, and then reconstruct the LR and HR image as a linear combination of the separated $\mathbf{D}_l$ and $\mathbf{D}_h$. This strategy only can achieve the global minimum reconstructing error of LR and HR local patches, and usually cannot obtain the exactly corresponding LR and HR dictionaries. In addition, the accurate coefficients for reconstructing the HR image patch using HR dictionary $\mathbf{D}_h$ are also unable to be estimated using only the LR input and the LR dictionary $\mathbf{D}_l$. Therefore, this chapter proposes to firstly learn the HR dictionary $\mathbf{D}_h$ from the features of the training HR local patches, and then propagates the HR dictionary to the LR one, called as

HR2LR dictionary propagation, by mathematical proving and statistical analysis. The effectiveness of the proposed HR2LR dictionary propagation in sparse coding for super-resolution is demonstrated by comparison with the conventional super-resolution approaches such as sparse coding and LLE. Furthermore, we validate that the proposed algorithm is robust to noisy for generating the HR image from a noisy LR image.

The remaining parts of this chapter are organized as follows. We introduce the basic sparse coding for image representation in Sec. 2, and give a detail descriptors of the SC-based image super-resolution in Sec. 3. Sec. 4 describes the used LR and HR features for learning procedure, and explores the relationship between the used LR and HR features. The proposed HR2LR dictionary propagation strategy in sparse coding is given in Sec. 5, and experimental results are shown in Sec. 6. Finally, we conclude and summarize in Sec. 7.

2 Sparse Coding

In statistical analysis of image representation, recent works [39-40] show that any image local patch can be represented by a sparse linear combination of the atoms in an over-complete dictionary. Assuming $\mathbf{D} \in \Re^{n \times K}$ be an over-complete dictionary of K prototype atoms by statistical learning from some reshaped image patches, a reshaped vector $\mathbf{x}$ from one image patch can be written as $\mathbf{x} = \mathbf{D}\alpha_\mathbf{0}$, where $\alpha_0 \in \Re^K$ is a vector with very few $(\ll K)$ nonzero entries.

Problem Formulation: Suppose that there are N data samples $\{\mathbf{y}_i \in \Re^n : i = 1, 2, \cdots, N\}$ of dimension n, and the collection of these N samples forms an n-by-N data matrix $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N)$ with each column as one sample vector. The goal is to construct a representative dictionary for $\mathbf{Y}$ in the form of an n-by-K matrix $\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K)$, which consists of K (usually $K \ll N$ and $K > n$) key features $\{\mathbf{d}_i \in \Re^n : i = 1, 2, \cdots, K\}$ extracted from $\mathbf{Y}$. In the dictionary context, $\mathbf{d}_i$ is also called an atom that represents one prototype feature for well-representing any input data. This dictionary $\mathbf{D}$ needs to be trained from Y, and should be capable to sparsely represent all the samples in Y. In other words, we want to find a dictionary $\mathbf{D}$ and corresponding coefficient matrix $\mathbf{A} = (\alpha_1, \alpha_2, \cdots, \alpha_N) \in \Re^{K \times N}$ such that $\mathbf{y}_i = \mathbf{D}\alpha_i$ and $\|\alpha_i\|_0 \ll K$ for all $i = 1, 2, \cdots, N$. This can be intuitively formulated as the following minimization problem:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{y}_i - \mathbf{D}\alpha_i\|_2 \quad \mathbf{s.t.} \quad \|\alpha_i\|_0 < T \tag{1}$$

where T is the predefined threshold which controls the sparseness of the representation and $\| \bullet \|_0$ denotes the l_0 norm which counts the number of non-zero element in the vector. The equation can also alternately be formulated as:

$$\min_{\mathbf{D},\mathbf{A}} \sum_{i=1}^{n} \|\alpha_i\|_0 \quad \mathbf{s.t.} \quad \|\mathbf{y}_i - \mathbf{D}\alpha_i\|_2 \leq \varepsilon \tag{2}$$

where $\varepsilon > 0$ is the predefined tolerance of representation error. The solution $(\mathbf{D}, \mathbf{X})$ of Eq. 3 yields a dictionary $\mathbf{D}$ which extracts the representative features $\{\mathbf{d}_i \in \Re^n : i = 1, 2, \cdots, K\}$ from samples in $\mathbf{Y}$ and a coefficient matrix $\mathbf{A}$ with each column α_i representing the similarity between the sample $\mathbf{y}_i$ and the dictionary atoms in $\mathbf{D}$.

Since the optimization problem in Eq. 2 is NP-hard in general, recent results suggest that several algorithms are able to be used for well approximating the solutions of Eq. 3 [39-40]. In this chapter, we use the recently developed K-SVD algorithm , which has proved to be very robust to solve Eq. 2, by iterating exact K times of Singular Value Decomposition (SVD). With an initial dictionary, K-SVD algorithm solves Eq. 2 by alternating the following two steps: the minimization with respect to $\mathbf{A}$ with the fixed $\mathbf{D}$, and atoms updating in $\mathbf{D}$ using the current $\mathbf{A}$. The formulate of the first step is same to Eq. 2 with the fixed $\mathbf{D}$, called the "sparse coding", which can be approximated by the orthogonal matching pursuit (OMP) [41]. in the following subsection, we will introduce how to calculate sparse coefficient using OMP strategy with initially selected dictionary, and update the dictionary $\mathbf{D}$ using K-SVD with the calculated sparse coefficients in the previous step.

2.1 Orthogonal Matching Pursuit

OMP is an extended orthogonal version of matching pursuit (MP), which is a type of numerical technique which involves finding the "best matching" projections of multidimensional data onto an over-complete dictionary $\mathbf{D}$. The OMP algorithm attempts to achieve the projected coefficients of the selected best basis vectors (atoms) iteratively to minimize the representation error, where the main difference from MP is that after every step, all the coefficients extracted so far are updated, by computing the orthogonal projection of the signal onto the set of selected atoms. Let $\mathbf{y}$ denotes an observed signal, and $\mathbf{D}$ denotes the fixed dictionary, the OMP algorithm attempts to find the sparse code vector α in four steps:

Step 1. Initialize the residual $\mathbf{r}_0 = \mathbf{y}$, and initialize the selected dictionary $\mathbf{D}' = \phi$ and the corresponding coefficients $\alpha_0(\mathbf{D}') = \phi$. Let iteration counter $i = 1$, and the dictionary candidate $\mathbf{D}_0(\mathbf{D}_i) = \mathbf{D}$, from which one best basis (atom) is needed to be selected in following.

Step 2. Project the residual vector $\mathbf{r}_i$ to the dictionary candidate $\mathbf{D}_i$, and find the atom with the maximum projection value:

$$\mathbf{d} \Leftarrow \max_{\mathbf{k}} \mathbf{D}_i \mathbf{r}_i \tag{3}$$

Delete $\mathbf{d}$ from the dictionary candidate $\mathbf{D}_i$, and add it to the selected dictionary $\mathbf{D}' = [\mathbf{D}', d]$.

Step 3. Update the coefficients $\alpha_i \Leftarrow \mathbf{D}_i^{\dagger}\mathbf{y}$ using the following equation:

$$\alpha_i(\mathbf{D}') = \min_{\alpha_i} \|\mathbf{y} - \mathbf{D}_i \alpha_i\|^2 \tag{4}$$

Step 4. Update the residual $\mathbf{r}_i = \mathbf{y} - \mathbf{D}_i \alpha_i$.

The stop rule for OMP algorithm can be tuned to solve for either of the problem defined in Eq. 1, which iterates the step 2∼4 T times and Eq. 2, which would quit the iteration when $\|\mathbf{r}_i\|_2^2 < \epsilon$.

2.2 K-SVD Algorithm

As introduced in the above section, the sparse representation problem can be formulated by either Eq. 1 or Eq. 2. Let's assume the sparse representation problem formulated as Eq. 1, and the goal is to train the adaptive dictionary $\mathbf{D} \in R^{n*K}$ and the corresponding sparse coefficients $\alpha \in R^{K*N}$ from the observed dataset $\mathbf{Y} \in R^{n*N}$, where n is the dimension of the observed signal, N is the sample number, and K is the number of atoms or the dimension of the output sparse vector with $K >> n$. We introduce the K-SVD algorithm for extracting the adaptive dictionary $\mathbf{D}$, which is flexible and works in conjunction with any pursuit algorithm. The K-SVD algorithm is simply designed to be a truly direct generalization of the K-means. When forced to work with one atom per signal, it can train a dictionary for the gain-shape VQ. When forced to have a unit coefficient for this atom, it exactly reproduces the K-means algorithm. We start our discussion with a description of the K-means, and then derive the K-SVD algorithm as its direct extension.

A. K-means algorithm for vector quantization

K-means is to produce a codebook including codewords (representatives), which is used to represent a wide family of observed vectors (signals) by nearest neighbor assignment [42-48]. It can lead to efficient compression or description of those observed signals as clusters in surrounding the chosen codewords. Generally, K-means can be implemented based on the expectation maximization procedure, and intuitively it can be extended to the fuzzy assignment using similarity between an input signal and the codeword or normalized similarity by the covariance matrix per each cluster, where that the signal are modeled as a mixture of Gaussians [49]. Let's introduce the general K-means algorithm for learning the codebook matrix (dictionary) $\mathbf{D}$ with the codeword being in the columns from a set of observation signal $\mathbf{Y}$. In k-means, a signal $\mathbf{y}_i$ is represented as its closest codeword (under l^2-norm distance), and then its coded vector α_i include only one non-zero element with value 1, and all others zeros. Therefore, the objective function is to minimize the within-cluster sum of squares (WCSS):

$$\begin{aligned} &\arg\min_{\alpha} \sum_{i=1}^{N} \|\mathbf{y}_i - \alpha_i \mathbf{D}\| \\ &s.t. \|\alpha_i\|_{l^0} = 1, \sum_{j=1}^{K} \alpha_{ij} = 1, for\, all\, i \end{aligned} \tag{5}$$

Table 1 K-means algorithm

Goal: Find the best possible codebook to represent the observed signals $\mathbf{Y}=\{\mathbf{y}_i\}_{i=1}^N$ **using nearest neighbor**

$$\arg\min_\alpha \sum_{i=1}^N \|\mathbf{y}_i-\alpha_i\mathbf{D}\|$$

$$s.t.\|\alpha_i\|_{l^0}=1,\sum_{j=1}^K\alpha_{ij}=1, for\,all\,i$$

Initialization: initially the codebook $\mathbf{D}^{(0)}\in R^{n\times K}$ by randomly selecting K samples.

Set iteration number t=1, and repeat until convergence

Sparse coding step: Assign the observed sample to one of K codewords, and K cluster set can be achieved

$$(\mathbf{C}_1^{(t-1)},\mathbf{C}_2^{(t-1)},\cdots,\mathbf{C}_K^{(t-1)})$$

where the sample $\mathbf{y}_i$ index i in k cluster $\mathbf{C}_k^{(t-1)}$ should satisfies the following condition:

$$\mathbf{C}_k^{(t-1)}=\{i\mid\forall_{l\neq k}\parallel\mathbf{y}_i-\mathbf{d}_k\parallel_2<\parallel\mathbf{y}_i-\mathbf{d}_l\parallel_2\}$$

Codebook update step: Update the k^{th} column $\mathbf{d}_k$ in the codebook by calculating the mean vector

in k^{th} cluster:

$$\mathbf{d}_k^{(t)}=\frac{1}{|\mathbf{C}_k|}\sum_{i\in\mathbf{C}_k}\mathbf{y}_i$$

set $t=t+1$

where $\alpha=[\alpha_1,\alpha_2,\cdots,\alpha_N]$ is the set of code vectors for the observed signal set $\mathbf{Y}$. The cardinality constraint $\|\alpha_i\|_{l^0}=1$ means there will be only one non-zero element in each code vector $\mathbf{y}_i$, which corresponds to the most sparsity representation for the observed signal. The summation constraint $\sum_{j=1}^K\alpha_{ij}=1$ imposes the coding weight for $\mathbf{y}_i$ is 1.

The K-means algorithm is generally implemented in an iterative strategy for designing the optimal codebook for vector quantization [39]. In each iteration there are two steps: one for assigning the observed signal to the codewords which can be called as sparse coding step, and one for updating the codebook by calculating the mean vector in each cluster, which can be considered as dictionary update. Table. 1 gives a more detailed description of these steps. The sparse coding step assumes a known codebook $\mathbf{D}^{(t-1)}$ and computes the coded coefficient αi that minimizes the representation error of Eq. (5). Similarly, the dictionary update step seeks an update of $\mathbf{D}$ by minimizing Eq. (5) with a fixed αi as known.

B. K-SVD: a generalized version of K-means

As introduced in the above, K-means algorithm quantize an observed signal to a codeword by vector quantization (VQ), which means that only one codeword is selected for representing the observed signal. The VQ strategy would results in a lot of representation error especially for the samples in the boundary areas of clusters. The sparse representation problem can be viewed as a generalization of the VQ problem (Eq. (5)), in which each observed signal is represented by a linear combination of codewords, called dictionary elements (atoms) in sparse coding (SC). Then, the coded coefficients vector is now allowed more than one nonzero entry, and these can have arbitrary values. In SC, the cost function can be relaxed as Eq. (1) or Eq. (2) mentioned in the Sec. 2.

K-SVD is proposed to combine an approximation pursuit method to solve the minimization problem of Eq. (1). First, with a initialized fixed dictionary $\mathbf{D}$, a best sparse coefficient matrix is solved using a pursuit method by minimizing Eq. (1), called sparse coding step. With the calculated coefficient in sparse coding step, the second step is performed to search for a better dictionary. This step updates one column at a time, fixing all columns $\mathbf{D}$ except one, $\mathbf{d}_k$, which attempt to find the new column $\mathbf{d}_k$ and the new values for its coefficients that best reduce the mean square error (MSE). The process of updating only one column of at a time can lead to a straightforward solution based on the singular value decomposition (SVD), and allowing a change in the coefficient values while updating the dictionary columns accelerates convergence, since the subsequent column updates will be based on more relevant coefficients. Next we will give the detail description of K-SVD algorithm for dictionary update. Assuming we have already extract the sparse coefficients in an iteration step with an fixed dictionary in the preview step, let's update only one column $\mathbf{d}_k$ in the dictionary and the coefficients that correspond to it: the k^{th} row in α, denoted as α_R^k (not the k_{th} column vector α_k in α). Then the objective function can be rewritten as:

$$\begin{aligned} \|\mathbf{Y} - \mathbf{D}\alpha\|^2 &= \|\mathbf{Y} - \sum_{i=1}^{K} \mathbf{d}_i \alpha_R^i\|^2 \\ &= \|\mathbf{Y} - \sum_{i \neq k} \mathbf{d}_i \alpha_R^i - \mathbf{d}_k \alpha_R^k\|^2 \\ &= \|\mathbf{E}_k - \mathbf{d}_k \alpha_R^k\|^2 \end{aligned} \tag{6}$$

The above equation separates the error term into two parts: error when the atom $\mathbf{d}_k$ is not taken into account, and the error reduction due to its induction for representation. This also decompose the matrix multiplication $\mathbf{D}\alpha$ to the sum of K rank-1 matrices, among which $K - 1$ terms are assumed fixed, and the k^{th} one if left for updating. Then the problem of minimizing the total error thus boils down to finding a rank-1 matrix which best approximates the error matrix $\mathbf{E}_k$. Estimation of such a matrix could simply be done by performing a singular value decomposition on $\mathbf{E}_k$ and using the largest singular value and its corresponding vector for this task. However, such a step will lead to an update of the coefficients: the row vector α_R^k being very likely to be filled, which would not be sparse any more. An intuitive remedy of this problem is to form the matrix $\mathbf{E}_k$ as the reconstruction error resembles, denoted as $\mathbf{E}_{+k}$, of the observed signals which use the k^{th} atom of the dictionary for reconstruction, since the reconstruction errors of the other samples do not any change due to deleting the atom $\mathbf{d}_k$. Therefore, in order to achieve the updated atom $\mathbf{d}_k$ and the sparse coefficient, SVD decomposition of $\mathbf{E}_{+k}$ can be directly conducted, where Eigenvector of the largest Eigenvalue is used for updating the k^{th} atom $\mathbf{d}_k$ with only updating the coefficients which used the k^{th} atom so far. To implement, we first identify all the observed signals that use the k^{th} atom of the dictionary for reconstruction. Then the total error term of Eq. 6 can be split into two terms, where one term is the resulted error of representation of those signals due to the $\mathbf{d}_k$ atom

Table 2 K-SVD algorithm

```
Goal: Find the best dictionary to represent the observed signals $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$
    as a linear sparse combination by solving
        $\arg\min_{\alpha} \sum_{i=1}^{N} \|\mathbf{y}_i - \alpha_i \mathbf{D}\|$
        $s.t. \|\alpha_i\|_0 < T, for\, all\, i$
Initialization: initially the dictionary $\mathbf{D}^{(0)} \in R^{n \times K}$ with $l^2$ normalized column.
        Set iteration number t=1. Repeat until convergence.
Sparse coding step: Using any pursuit method (such the OMP algorithm ) to calculate the sparse vector $\alpha_i$
        for each sample $\mathbf{y}_i$, by solving the following Equation with the fixed dictionary $\mathbf{D}$
            $\arg\min_{\alpha_i} \sum_{i=1}^{N} \|\mathbf{y}_i - \alpha_i \mathbf{D}\|$
            $s.t. \|\alpha_i\|_0 < T_0, \quad i = 1, 2, \cdots, N$
Dictionary update step: For each atom (each column) in Dictionary $\mathbf{D}^{(t-1)}$, update it by,
    · Obtaining the index set $idx \Longleftarrow$ all non-zero indices of $\alpha_R^k$
      or the sample indices that use the $k^{th}$ atom
    · Calculating the reconstruction error $\mathbf{E}_{+k}$ of the sample with indices $idx$ that use the $k^{th}$ atom,
      when remove the $k^{th}$ atom
            $\mathbf{E}_{+k} = \mathbf{Y}_{+k} - \sum_{i \in idx} \mathbf{d}_i \alpha_R^i - \mathbf{d}_k \alpha_R^k$
    · Doing SVD decomposition on $\mathbf{E}_{+k}$, update the $k^{th}$ atom $\mathbf{d}_k$ using the eigenvector with the largest Eigenvalue.
            $\mathbf{E}_{+k} = \mathbf{U} \triangle \mathbf{V}^T$
    · updating the coefficient vector $\alpha_R^k$ using the first column of $\mathbf{V}$ multiplied with $\triangle(1,1)$.
set $t = t + 1$
```

being removed, and the other is the un-varied reconstruction error of the observed signals which do not use the k^{th} atom for reconstruction. The reconstruction error can be written as:

$$\begin{aligned} \|\mathbf{Y} - \mathbf{D}\alpha\|^2 &= \|\mathbf{Y} - \sum_{i=1}^{K} \mathbf{d}_i \alpha_R^i\|^2 \\ &= \|\mathbf{E}_{-k} + \mathbf{E}_{+k} - \mathbf{d}_k \alpha_R^k\|^2 \end{aligned} \tag{7}$$

where $\mathbf{E}_{-k}$ is the unchanged reconstruction error due to the deleting of the k^{th} atom, $\mathbf{E}_{+k}$ is the reconstruction error matrix with zero-vector for the observed signal without using the k^{th} atom but some reconstruction residual for the ones with using the k^{th} atom for representation. Let's firstly remove the zero-vector from the error matrix $\mathbf{E}_{+k}$, and decompose it using SVD for achieving the Eigenvector of the largest eigenvalue to update the k^{th} atom, and the corresponding vector to update the observed signals using the k^{th} atom. For all atoms, the procedure is iterated K times for updating each atom. Therefore, this procedure for dictionary updating is called 'K-SVD' to parallel the name K-means. While K-means applies computations of means to update the codebook, K-SVD obtains the updated dictionary by SVD computations, each determining one column. A detail description of the algorithm is given in Table. 2. Fig. 1 shows some 2-dimensional 8*8 DCT basis, and the learned K-SVD basis from some observed natural image patches.

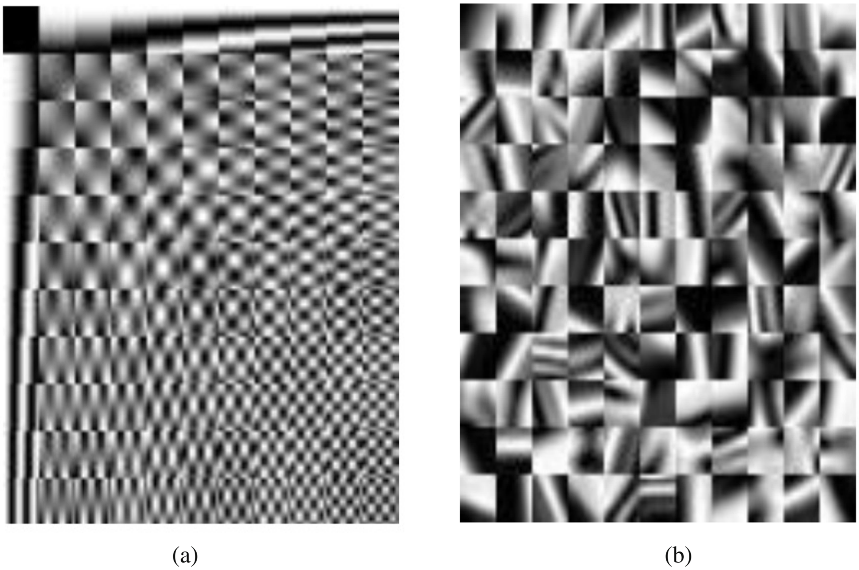


Fig. 1 Basis functions. (a) DCT basis; (b) the learned basis with K-SVD from the 8*8 natural image patches.

3 Sparse Coding Based Super-Resolution

The single-image super-resolution is to recover a high-resolution (HR) image $\mathbf{X}$ from a observed low-resolution image $\mathbf{Y}$, which is a blurred and downsampled version of the HR one $\mathbf{X}$:

$$\mathbf{Y} = \mathbf{LHX} \tag{8}$$

where $\mathbf{H}$ represents a blurring (smooth) filter, and $\mathbf{L}$ is the sown-sampling operator. The degradation model of the imaging procedure is shown in Fig. 2. In the learning-based super-resolution, the lost information in any test LR image can be recovered by learning using the corresponding relationship of the raw patches in the available LR and HR images. With same philosophy, given any LR image patch $\mathbf{y}$ well reconstructed by a sparse linear combination of an over-complete LR dictionary $\mathbf{D}_l$, the HR corresponding image patch $\mathbf{y}$ can also be approximated by the liner combination of corresponding HR dictionary $\mathbf{D}_h$ with the same sparse coefficients as the following:

$$\mathbf{y} = \mathbf{D}_l\alpha_0, \qquad \mathbf{x} \approx \mathbf{D}_h\alpha_0 \tag{9}$$

where α_0 is a vector with very few ($\ll K$) nonzero entries. In the conventional super-resolution using sparse coding, the image local patches are firstly reconstructed by the sparse linear combination of the pre-learned dictionary, and then remove the artifacts in the recovered global HR images formed from the local patches based on reconstruction constraints. Next, we will mainly introduce sparse representation of image local patches for super-resolution.

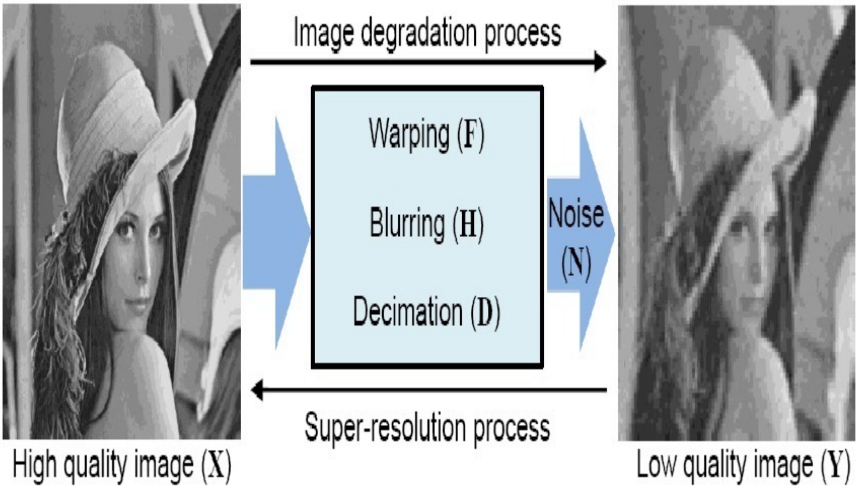


Fig. 2 The degradation model of imaging procedure

Similar to the conventional learning-based super-resolution framework which is shown in Fig. 2, the sparse coding based one also tries to infer the high-resolution patch from each low-resolution image patch of the input. In the sparse representation of the image local patches, there are two dictionaries $\mathbf{D}_h$ and $\mathbf{D}_l$, which are trained to have the similar sparse representations for each high-resolution and low-resolution image patch pair. Given any input low-resolution patch $\mathbf{y}$, we can find a sparse representation with respect to $\mathbf{D}_l$. The estimation of the corresponding high-resolution patch $\mathbf{x}$ can also be reconstructed by the sparse combination of these same coefficients but replacing $\mathbf{D}_l$ with $\mathbf{D}_h$.

For sparse coding based super-resolution, the corresponding LR and HR dictionaries need to be learned from the training LR and HR image patches $\mathbf{Y}$ and $\mathbf{X}$, respectively. [37] modifies Eq. 3 as the following optimization formulation:

$$\begin{aligned} \min_{\mathbf{D}_l,\mathbf{D}_h,\mathbf{A}} \sum_{i=1}^{n} \|\alpha_i\|_0 \quad \mathbf{s.t.} \quad & \|F\mathbf{y}_i - F\mathbf{D}_l\alpha_i\|_2^2 \leq \varepsilon_1 \\ & \|P\mathbf{x}_i - P\mathbf{D}_h\alpha_i\|_2^2 \leq \varepsilon_2 \end{aligned} \tag{10}$$

where F is a linear feature extraction operator, which is to provide a perceptually meaningful constraint on how closely the coefficients α approximates the input patch $\mathbf{x}_l$. In [38], The 1-order derivative operator are used for F. Sec. 3 will explore the choice of F in our proposed HR2LR dictionary propagation. The $\mathbf{P}$ is the operator for subtracting mean intensity of all pixels from the HR image patch.

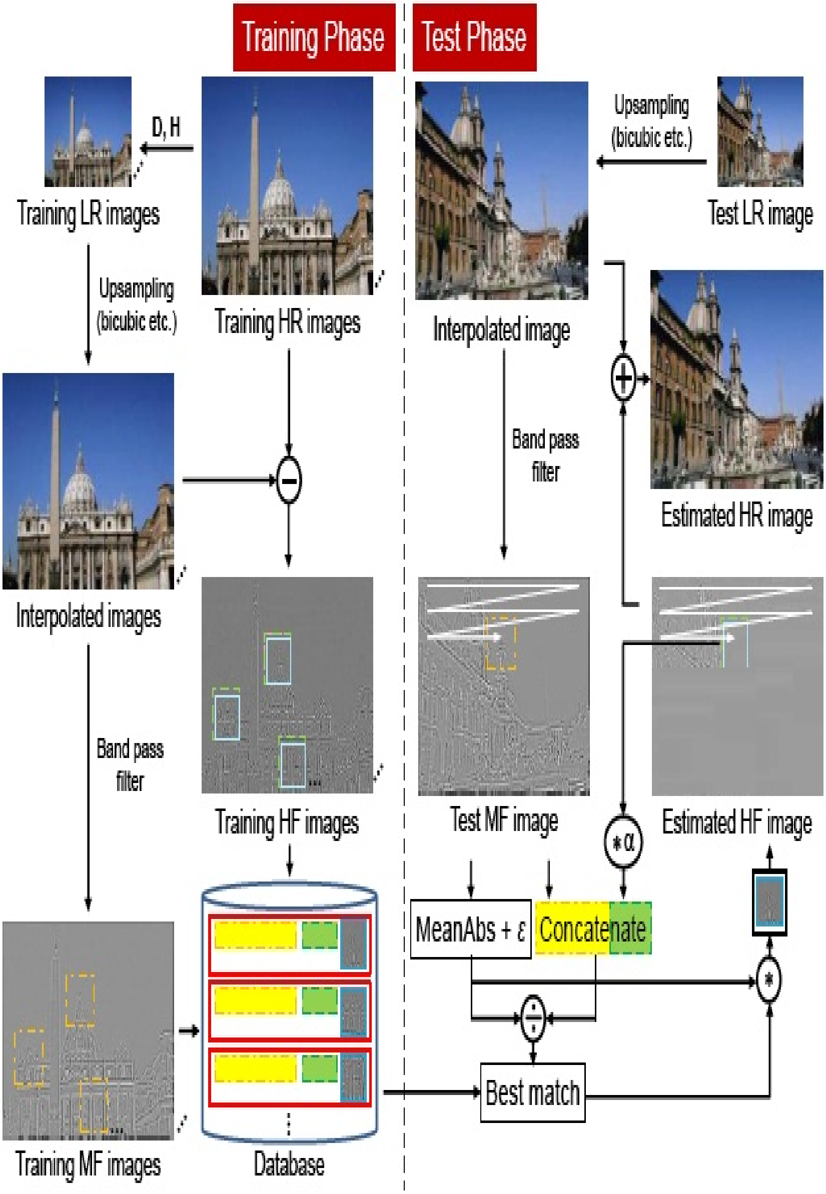


Fig. 3 The framework of learning-based Super-Resulution

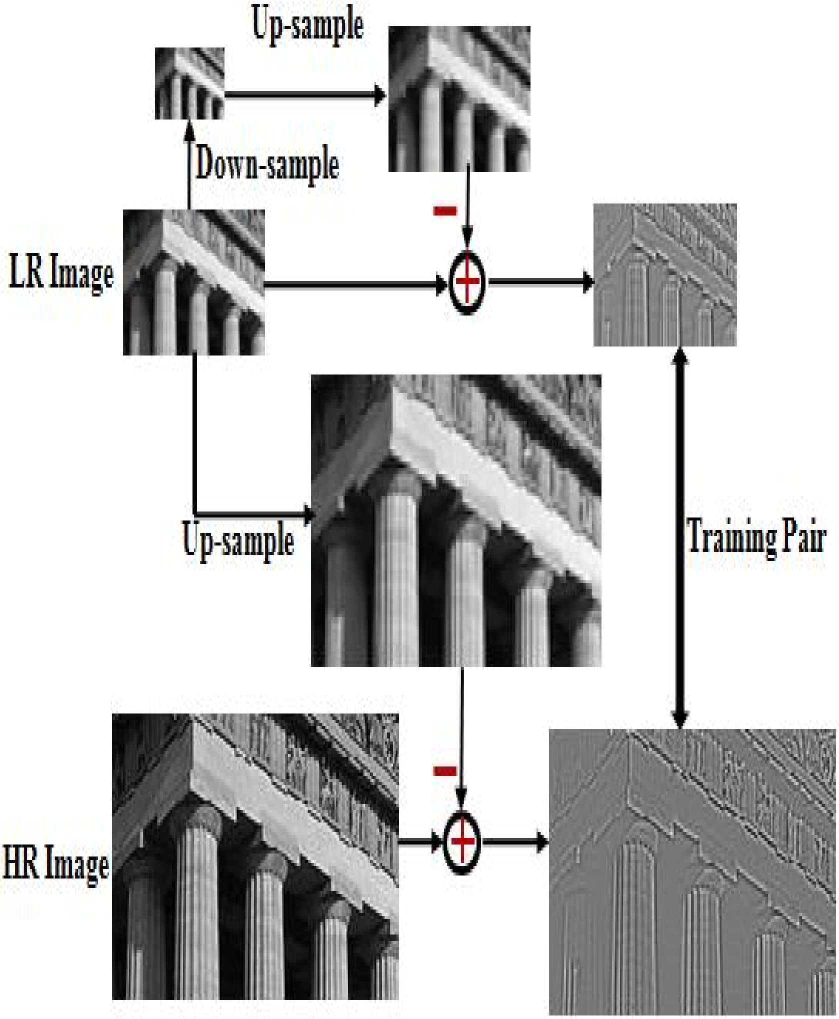


Fig. 4 The LR and HR feature extraction procedure

The constrained optimization (6) can be similarly reformulated as a jointly learning procedure for $\mathbf{D}_l$ and $\mathbf{D}_h$ [38]:

$$\min_{\bar{\mathbf{D}},\mathbf{A}} \sum_{i=1}^{n} \|\alpha_i\|_0 \quad \mathbf{s.t.} \quad \|\bar{\mathbf{y}}_i - \bar{\mathbf{D}}\alpha_i\|_2^2 \leq \varepsilon \tag{11}$$

where $\bar{\mathbf{D}} = [F\mathbf{D}_l; \beta P\mathbf{D}_h]$ and $\bar{\mathbf{y}} = [F\mathbf{y}_i; \beta P\mathbf{x}_i]$. The parameter β controls the tradeoff between reconstructing the LR and HR patches. With any input LR image patch $\mathbf{x}_t$, the sparse coefficient α_t can be achieved with the learned LR dictionary

$\mathbf{D}_l$, and then, the corresponding HR patch can be estimated with the same coefficients α_t and the learned HR dictionary $\mathbf{D}_h$. However, The above jointly learning procedure for $\mathbf{D}_l$ and $\mathbf{D}_h$ usually cannot achieve the accurate corresponding LR and HR dictionaries, and the sparse coefficients are also approximated estimation with only the available input LR feature $F\mathbf{y}_t$. Therefore, the following section investigates the corresponding LR and HR features for image patch representation, which invokes the proposed HR2LR dictionary propagation for achieving the accurate corresponding LR and HR dictionaries.

4 Analysis of the Represented Features for Local Image Patches

In Eq. 5, some features are needed to be extracted for image representation. The conventional sparse coding based super-resolution algorithm [37] uses the first order derivative as F in Eq. 5 for LR image representation, and the subtracted pixel intensity from the mean of the HR patch as P. It is obvious that the used features for LR and HR image patch have no accurate correspondence even after some pre-normalization for removing scale variance [37]. As mentioned in Sec. 2, the low-resolution image $\mathbf{Y}$ is a blurred and down-sampled version of the high-resolution image $\mathbf{X}$: $\mathbf{Y} = SH\mathbf{X}$ with the blurring filter H and the down-sampling operator S. The most intuitive method for achieve the same size version of $\mathbf{X}$ from $\mathbf{Y}$ is to use up-sampling interpolation operator U: $\bar{\mathbf{X}} = U\mathbf{Y}$. Based on the interpolated version $\bar{\mathbf{X}}$ of $\mathbf{Y}$, the lost information of the high-resolution $\mathbf{X}$ can be considered as $\mathbf{X} - \bar{\mathbf{X}}$, which is the used feature for the training HR image, and at the same time, also is the estimated lost information of any LR input for achieving the HR one. The feature extraction, as linear operator P, for the HR image can be formulated as:

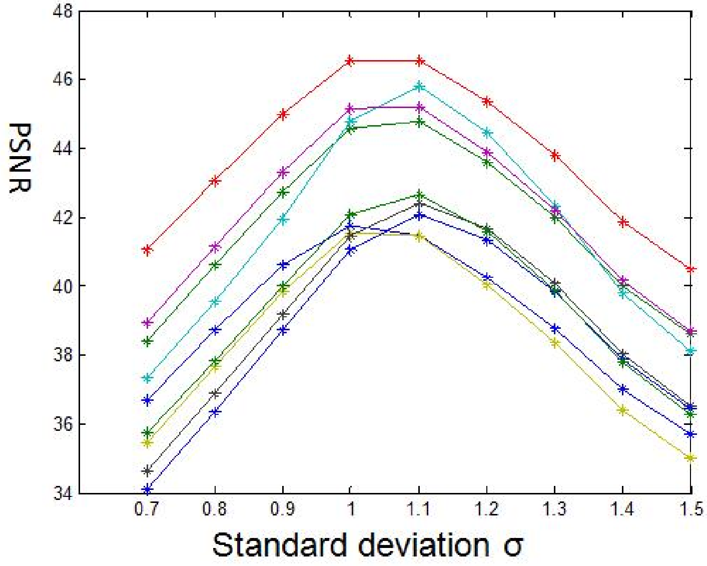
$$P\mathbf{X} = \mathbf{X} - \bar{\mathbf{X}} = \mathbf{X} - U\mathbf{Y} = \mathbf{X} - USH\mathbf{X} \tag{12}$$

In order to obtain the corresponding features of the LR image to those of the HR one, we impose the blurring and down-sampling operator on $\mathbf{Y}$: $\mathbf{Z} = SH\mathbf{Y}$, which is same on $\mathbf{X}$ to produce $\mathbf{Y}$, and then extract the LR feature by subtracting the interpolated version $\bar{\mathbf{Y}} = U\mathbf{Z}$ from the LR image $\mathbf{Y}$. Then the operator F for extracting feature from the LR image can be formulated as:

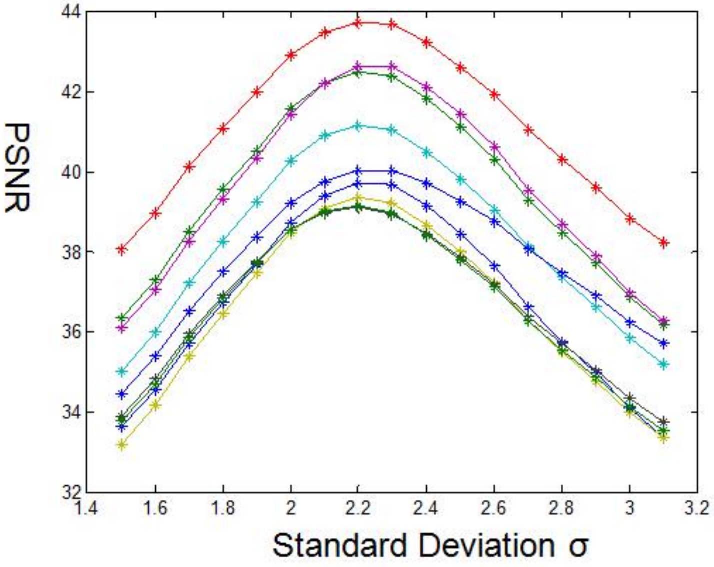
$$F\mathbf{Y} = \mathbf{Y} - \bar{\mathbf{Y}} = \mathbf{Y} - U\mathbf{Z} = \mathbf{Y} - USH\mathbf{Y} \tag{13}$$

The feature extraction procedures for the LR and HR image are shown in Fig. 3. From Eq. 7 and 8, it is obvious that the feature extractions for the LR and HR images follow the same process, prospecting corresponding relation between $F\mathbf{Y}$ and $P\mathbf{X}$. In addition, with $\mathbf{Y} = SH\mathbf{X}$ being the blurred and down-sampled version of the $\mathbf{X}$, the transformation from $P\mathbf{X}$ and $F\mathbf{Y}$ can be formulated as:

$$\begin{aligned} F\mathbf{Y} &= \mathbf{Y} - USH\mathbf{Y} = SH\mathbf{X} - USH(SH\mathbf{X}) \\ &= SH\{\mathbf{X} - USH\mathbf{X}\} = SH\{P\mathbf{X}\} \end{aligned} \tag{14}$$

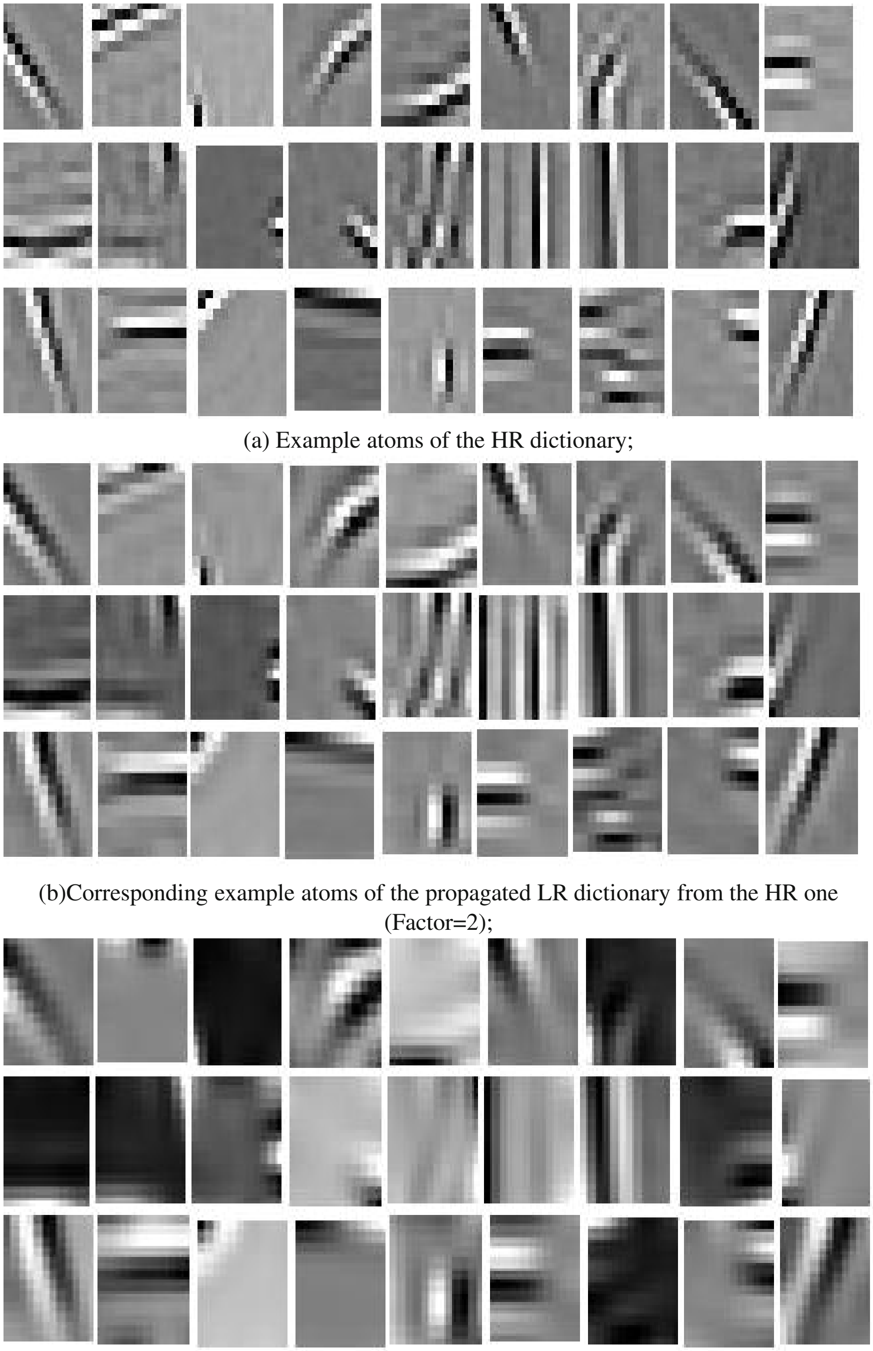


(a)The PSNR values between the interpolation versions of the LR images and the blurred versions of the HR images with different σ (Factor=2).



(b)The PSNR values between the interpolation versions of the LR images and the blurred versions of the HR images with different σ (Factor=4).

Fig. 5 The statistical analysis of the LR and HR images



(a) Example atoms of the HR dictionary;

(b)Corresponding example atoms of the propagated LR dictionary from the HR one (Factor=2);

(b)Corresponding example atoms of the propagated LR dictionary from the HR one (Factor=4);

Fig. 6 Example atoms of the LR and HR dictionaries

Fig. 7 The used example images for PSNR calculation

Therefore, the LR feature $F\mathbf{Y}$ is also a blurred and down-sampled version from the HR feature $P\mathbf{X}$. This means that the un-downsampled version of the LR features can be approximated by some suitable blurring version of the HR feature. If the HR feature is blurred by some suitable low-pass filter, the smoothed version should have high similarity with the un-downsampled version $H\{P\mathbf{X}\}$, which can be obtained by up-sampling the LR feature $F\mathbf{Y}$ using interpolation. Next, we investigate the similarity with PSNR (peak signal-to-noise ratio) between the interpolation version $\bar{\mathbf{Y}}$ of the LR images $\mathbf{Y}$ and the blurred version $H\mathbf{X}$ of the HR images $\mathbf{X}$ with low-pass filter.

We use the Gaussian kernel as the low-pass filter H with different standard deviation $\sigma = [0.7, 0.8, \cdots, 1.5]$, and utilize bilinear as the interpolation operator. The used 9 example images are shown in Fig. 4, and the PSNR values between between the interpolation version of the LR images and the blurred version of the HR image with different σ are shown in Fig. 5. It can be seen that the PSNR values of all images are larger than 40 with the largest one: more than 47 with about 1 or 1.1 σ value for expanding factor 2 (Fig. 5(a)), which means enough similarity and be difficult for distinguish from subjective assessment; larger than 37 with the largest one: more than 43 with about 2.1 or 2.3 σ value for expanding factor 4 (Fig. 5(b)). Based the statistical analysis, we will introduce the proposed HR2LR dictionary propagation approach of sparse coding for super-resolution.

5 HR2LR Dictionary Propagation of SC

SC based image super-resolution requires two corresponding dictionaries $\mathbf{D}_l$ and $\mathbf{D}_h$ to be pre-learned for reconstructing the LR and HR image patches using the sparse combination of their atoms. A given feature of a HR image patch $\mathbf{x}_i$ is reconstructed as a sparse combination of atoms taken from the HR dictionary $\mathbf{D}_h$ as follows:

$$\mathbf{x}_i \approx \sum_{j=1}^{K} \alpha_{ij} \mathbf{D}_h^j, \quad \mathbf{s.t.} \|\alpha_i\|_0 \leq L \tag{15}$$

where L is a positive integer, meaning that the non-zero numbers of α_i are less than L. As analyzed in Section 3, the LR feature is a down-sampled version of the corresponding HR feature, formulated as

$$\begin{aligned} \mathbf{y}_i = SH\mathbf{x}_i \approx SH \sum_{j=1}^{K} \alpha_{ij} \mathbf{D}_h^j &= \sum_{j=1}^{K} \alpha_{ij} [SH\mathbf{D}_h^j] \\ \mathbf{s.t.} \|\alpha_i\|_0 &\leq L \end{aligned} \tag{16}$$

From Eq. 11, we conclude that the accurate corresponding LR dictionary can be propagated by the mathematical transformation if the HR dictionary is available. Because the corresponding HR and LR training images are available, we can first learn the HR dictionary $\mathbf{D}_h$ using SC strategy as follows:

$$\min_{\mathbf{D}_h, \mathbf{A}} \sum_{i=1}^{n} \|\alpha_i\|_0 \quad \mathbf{s.t.} \quad \|\mathbf{x}_i - \mathbf{D}_h \alpha_i\|_2 \leq \varepsilon \tag{17}$$

With the HR dictionary $\mathbf{D}_h$ obtained, the LR dictionary can then be simply propagated using $\mathbf{D}_l = SH\mathbf{D}_h$. In real applications, because of the boundary effects in small image patches, the blurred version $\bar{\mathbf{D}}_h$ of the HR dictionary $\mathbf{D}_h$, which corresponds to the interpolated up-sampled version of the LR image, is used to obtain the sparse coefficient of any LR image input $\mathbf{y}_t$ as follows:

$$\begin{aligned} \min \|\alpha_t\|_0 \quad \mathbf{s.t.} \quad & \|\mathbf{y}_t - \mathbf{D}_l \alpha_t\|_2^2 \leq \varepsilon \\ & \|U\mathbf{y}_t - U\mathbf{D}_l \alpha_t\|_2^2 \approx \|U\mathbf{y}_t - \bar{\mathbf{D}}_h \alpha_t\|_2^2 \leq \varepsilon \end{aligned} \tag{18}$$

where U is the up-sampling operator, and $\bar{\mathbf{D}}_h$ is the blurred version of $\mathbf{D}_h$, which in turn is the approximation of the up-sampling of $\mathbf{D}_l$. With the obtained α_t value for sparse reconstruction of the LR input $\mathbf{y}_t$, the HR estimation can be reconstructed with the same α_t but by replacing $\mathbf{D}_l$ with $\mathbf{D}_h$. Figure 6 shows a learned HR dictionary and the corresponding propagated LR dictionary for the magnification factors 2 and 4.

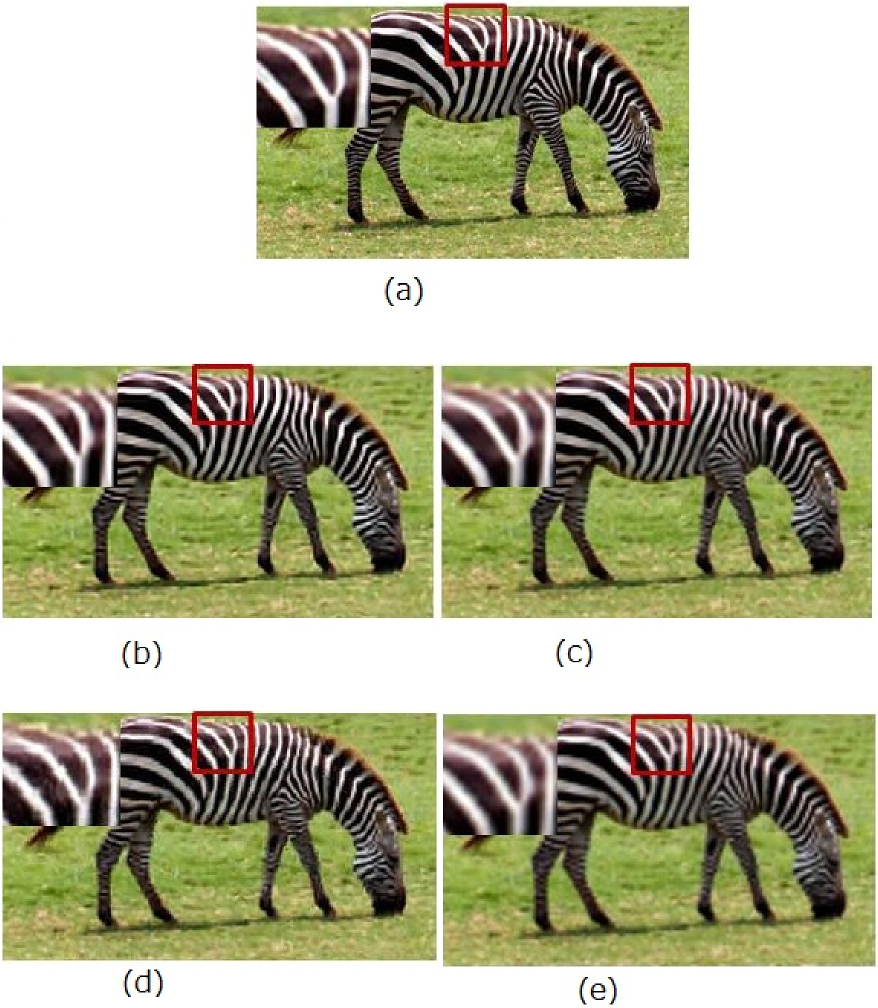


Fig. 8 Comparison of HR images of a zebra, reconstructed by different methods (magnification factor=2) with (a) the original HR image. Recovered images were obtained by (b) our proposed method, (c) the conventional SC-based method, (d) the NE-based method, and (e) the bicubic interpolation-based method.

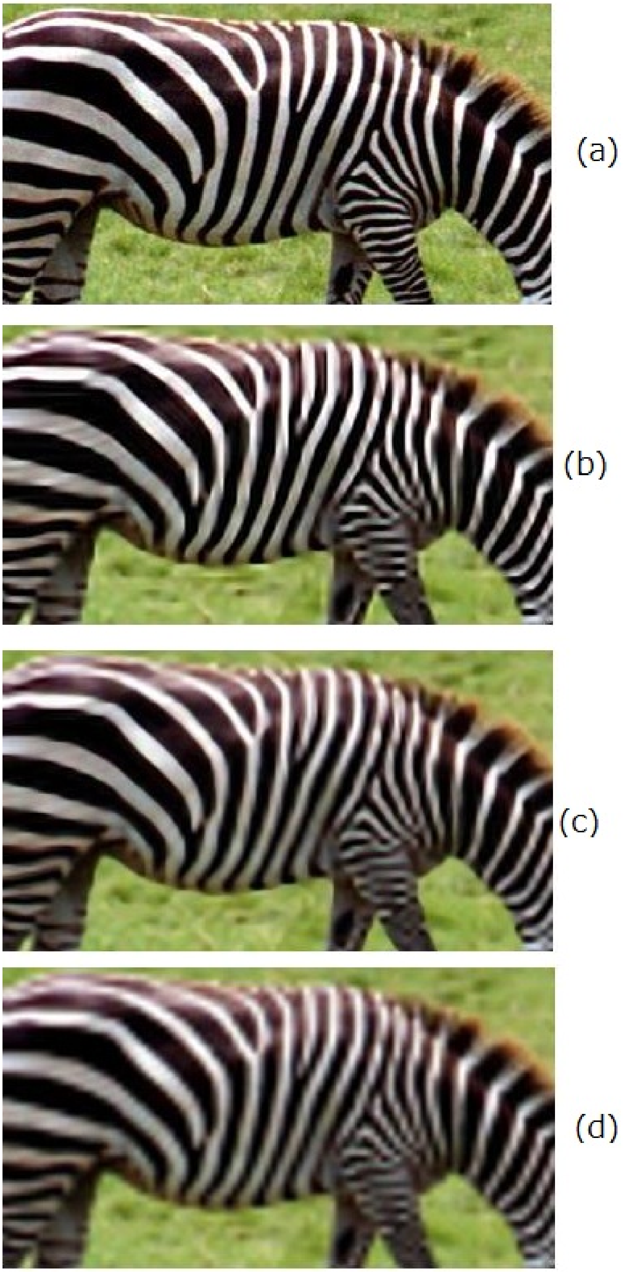
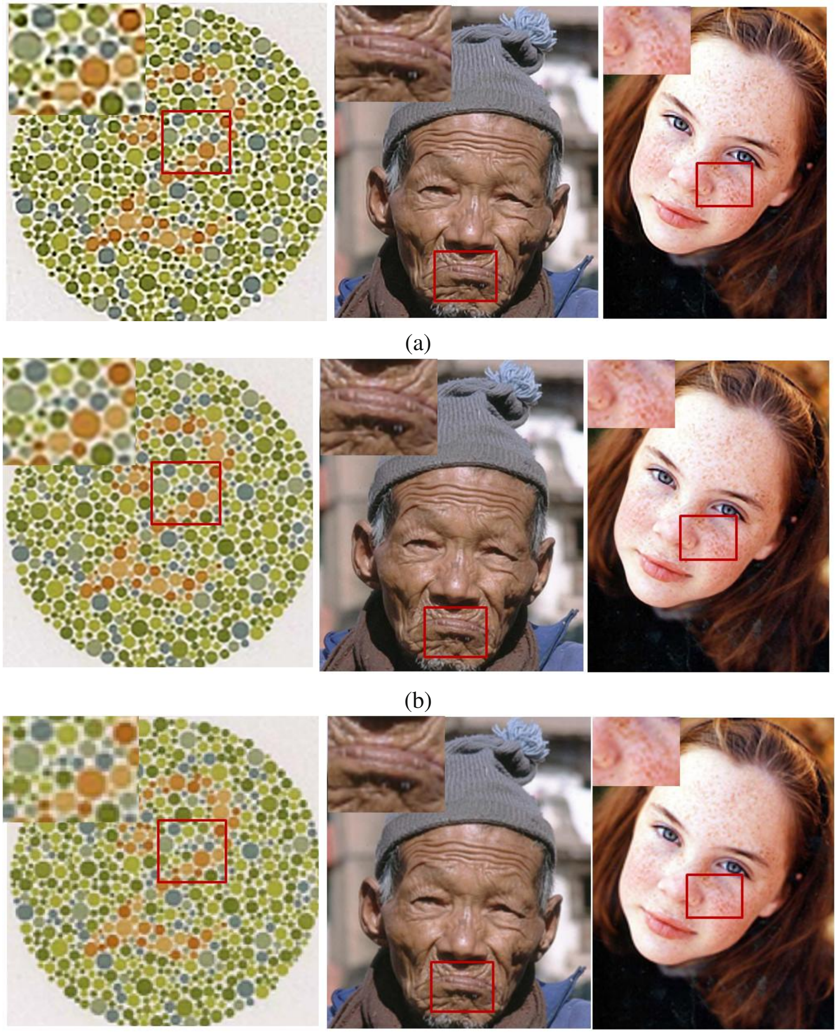


Fig. 9 Comparison of HR images of the zebra (magnification factor=4) reconstructed by different methods. (a) The Original HR zebra image and the HR recovered by (b) our proposed method (RMSE: 14.84), (c) the conventional SC-based method (RMSE: 15.40), and (d) the bicubic interpolation-based method (RMSE: 16.46).



(a)

(b)

(c)

Fig. 10 Other examples of HR images (magnification factor=2 recovered by (a) our proposed method, (b) the conventional SC-based method and (c) the bicubic interpolation method

Table 3 Comparison of the RMSE and PSNR for the zebra images in Fig. 8 recovered using different SR methods

Evaluated measures	RMSE	PSNR
Our method	11.21	27.14
Conventional SC	11.71	26.76
NE-based method	15.13	23.97
Interpolation	13.58	25.47

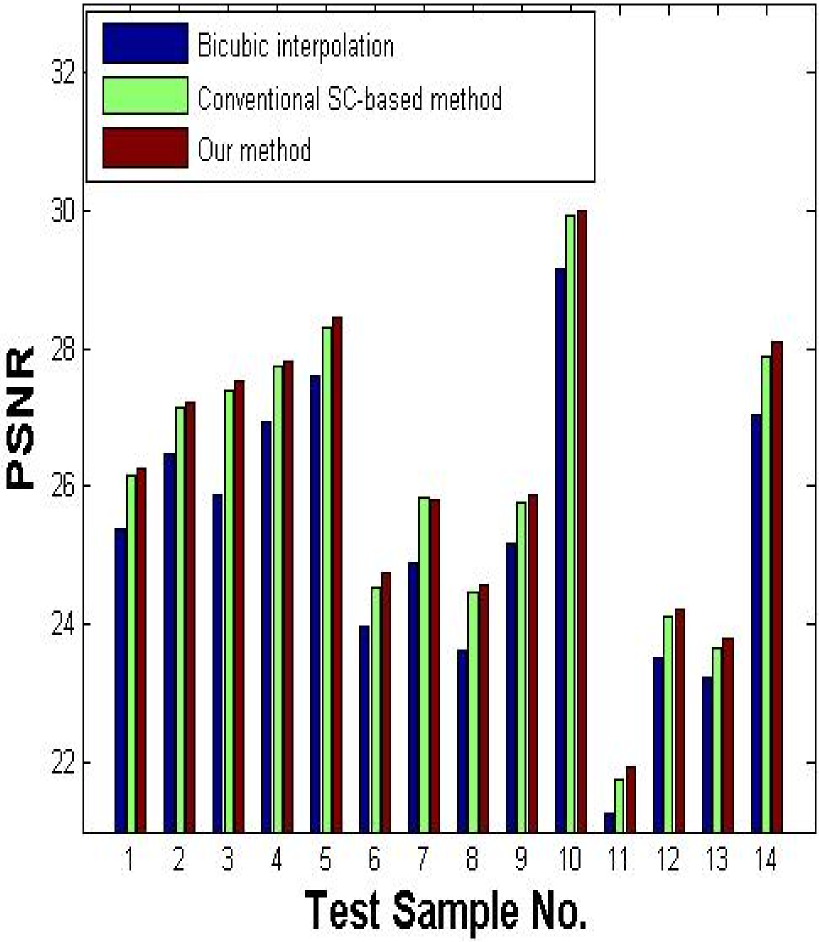


Fig. 11 The compared PSNRs of 14 test samples

6 Experiments

In our experiments, we magnified the LR input image by a factor of 2 or 4. We first interpolated the LR input to the same size. In the interpolated LR image and the corresponding HR image, we always use patches of size 12 × 12, with adjacent patches overlapping by 3 pixels. The features were then extracted as shown in Fig. 4. For color images, we only applied the SR strategy to the illuminance component, and the interpolated color components were used for reconstructing the HR final color image. To propagate the HR dictionary to the LR one, we used a Gaussian filter with a standard deviation $\sigma = 1.0$ for magnification factor 2, and $\sigma = 2.0$ for

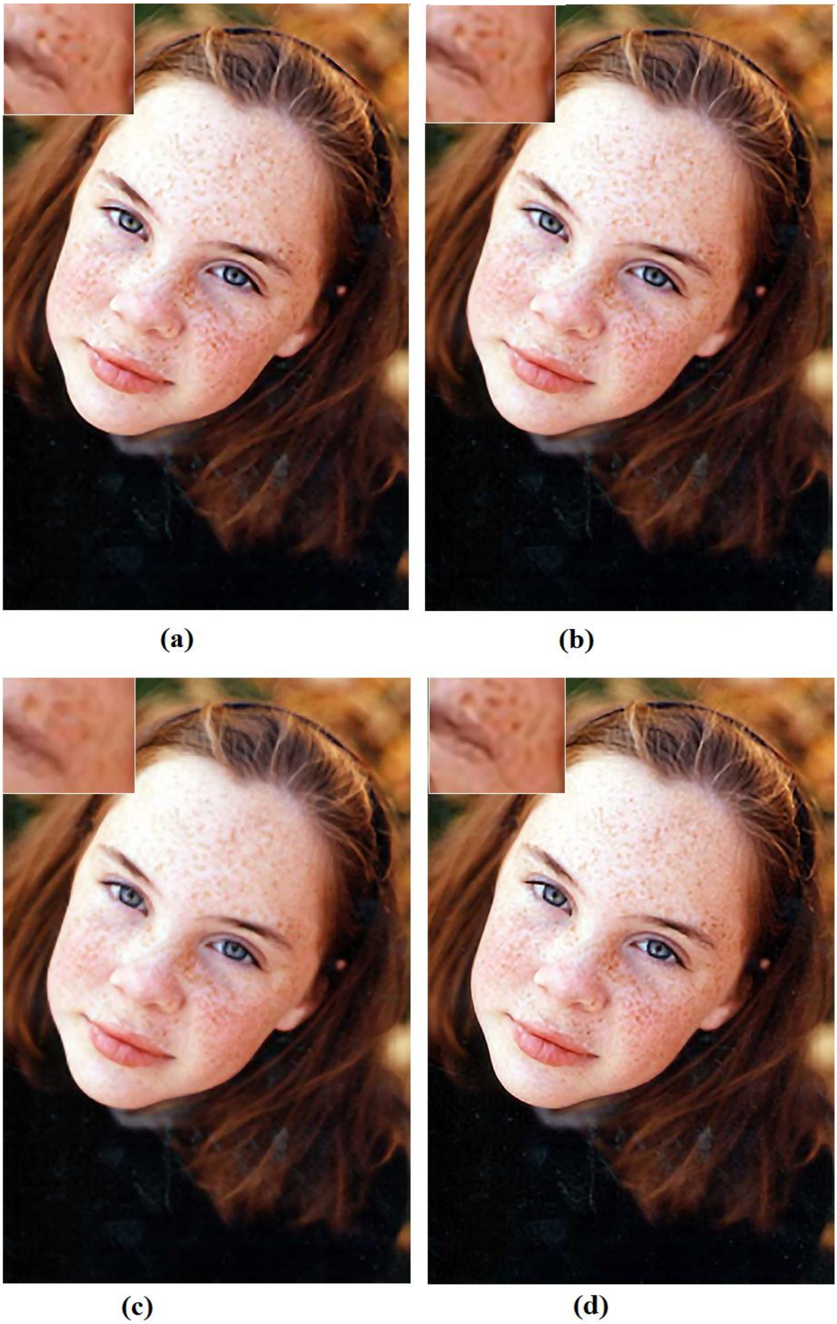


Fig. 12 The recovered HR images with other state-the-art methods. (a) our method, (b) Freedman's method [**?**], (c) Genuine Fractals (a state-of-the-art commercial product), (d) Glasner's method [**?**].

magnification factor 4, as shown in Fig. 6. Fig. 8 shows the HR images of the zebra recovered by our proposed strategy, the conventional SC-based [37], NE-based [35] and interpolation-based methods for magnification factor 2. Figure 9 shows a section of the HR images of the zebra (magnification factor 4) reconstructed by our proposed algorithm and by the conventional SC-based and interpolation-based methods. Figures 4 and 5 demonstrate that the proposed HR2LR dictionary propagation method in SC can yield much clearer HR images than yielded by the conventional SC-based, NE-based, and interpolation-based methods. We also evaluate the quantitative quality of the recovered HR images in Figs. 4 using root mean square error (RMSE) and the peak signal-to-noise ratio (PSNR) in Table 3. Figure 10 compares the reconstructed HR images derived from other LR inputs using our proposed strategy, the conventional SC-based and bicubic-interpolation-based methods. Here again, our proposed approach yields great clarity. In addition, in Fig. 11, we show the compared PSNR of the recovered HR images for other 14 test samples, which obviously validate most test images by our method can achieve better PSNR than the conventional SC-based method except for a similar PSNR for one sample. In order to validate effectiveness of the proposed strategy compared with other the state-of-the-art method [50-51], we also use the recovered HR images with expand factor 3 in Fig. 12. It is obvious that the recovered HR image is much better than the ones by Glasner's method [50], and has similar performance visually but sharper in some detail regions compared with Freedman's work [51].

7 Conclusions

This chapter introduces the sparse signal representation, and a popular implementation: K-SVD algorithm combining orthogonal matching pursuit (OMP) for learning the adaptive dictionary and achieving the sparse coefficients. OMP is an extended orthogonal version of matching pursuit (MP), which is a type of numerical technique which involves finding the "best matching" projections of multidimensional data onto an over-complete dictionary $\mathbf{D}$, and can be combined into the K-SVD strategy for achieving sparse representation and the best adaptive dictionary. K-SVD is popularly used for solving the optimization problem in sparse coding. The procedure of K-SVD mainly include two steps: first, with a initialized fixed dictionary $\mathbf{D}$, a best sparse coefficient matrix is solved using a pursuit method, called sparse coding step. With the calculated coefficient in sparse coding step is achieved, the second step is performed to search for a better dictionary. This step updates one column at a time, fixing all columns $\mathbf{D}$ in except one, $\mathbf{d}_k$, which attempt to find the new column $\mathbf{d}_k$ and the new values for its coefficients that best reduce the MSE.

Next, we apply the sparse representation for learning-based image super-resolution for recovering the high-resolution image from only single LR one. Based on the couple dictionary learning for super-resolution, we proposed a HR2LR dictionary propagation algorithm in SC for image super-resolution. Conventional SC-based image SR usually yields a global dictionary $\mathbf{D}$=[$\mathbf{D}_l$; $\mathbf{D}_h$] by jointly training the concatenated LR and HR local image patches and then reconstructing the LR and HR images as sparse combinations of atoms taken from $\mathbf{D}_l$ and $\mathbf{D}_h$. This strategy can only achieve the global minimum reconstruction error of LR and HR local patches, and cannot usually obtain exactly corresponding LR and HR dictionaries. In addition, accurate-coefficients for reconstructing the HR image patch using $\mathbf{D}_h$ cannot be estimated using only the LR input and the $\mathbf{D}_l$. This chapter proposes an algorithm called HR2LR dictionary propagation that involves learning the HR dictionary $\mathbf{D}_h$ from the features of the HR training local patches and then propagating the HR dictionary to the LR one by mathematical proofs and statistical analysis. The experimental results for image SR demonstrate that the proposed HR2LR dictionary propagation yields much clearer HR images than those obtained using conventional SR approaches such as those based on SC, NE and bicubic interpolation.

Acknowledgement. This work was supported in part by the Grant-in Aid for Scientific Research from the Japanese MEXT under the Grant No. 2430076, 24103710, 24700179 and in part by the R-GIRO Research fund from Ritsumeikan Univ.

References

1. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
2. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research 37, 3311–3325 (1997)
3. Lewicki, M.S., Sejnowski, T.J.: Learning overcomplete representations. Neural Comp. 12(2) (2000)
4. Olshausen, B.A.: Sparse coding of time-varying natural images. Journal of Vision 2(7), Article 130 (2002)
5. Olshausen, B.A., Field, D.J.: Sparse coding of sensory inputs. Cur. Op. Neurobiology 14(4) (2004)
6. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)
7. Aharon, M., Elad, M.: Image denoising via sparse and re- dundant representations over learned dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
8. Donoho, D.L., Elad, M., Temlyakov, V.: Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Transactions on Information Theory 52(1), 6–18 (2006)
9. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2, 433–459 (2010)
10. Roweis, S.: M Algorithms for PCA and SPCA. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) Advances in Neural Information Processing Systems. The MIT Press (1998)
11. Bell, A.J., Sejnowski, T.J.: The ‘Independent Components’ of natural scenes are edge filters. Vision Research 37, 3327–3338 (1997)
12. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7, 1129–1159 (1995)
13. Common, P.: Independent component analysis-a new concept? Signal Processing 36, 287–314 (1994)
14. Hyvarinen, A., Oja, E.: A fast fixed-point algorithm for indepepndent component analysis. Neural Computation 9, 1483–1492 (1997)

15. Hyvarinen, A., Oja, E., Hoyer, P.: Image Denoising by Sparse Code Shrinkage. In: Haykin, S., Kosko, B. (eds.) Intelligent Signal Processing. IEEE Press (2000)
16. Han, X.-H., Nakao, Z., Chen, Y.-W.: An ICA-Domain Shrinkage based Poisson-Noise Reduction Algorithm and Its Application to Penumbral Imaging. IEICE Trans. Inf. & Syst. E88-D(4), 750–757 (2005)
17. Han, X.-H., Chen, Y.-W., Nakao, Z.: Robust Edge Detection by Independent Component Analysis in Noisy Images. IEICE Trans. Inf. & Syst. E87-D(9), 2204–2211 (2004)
18. Sceniak, M.P., Hawken, M.J., Shapley, R.: Visual spatial characterization of macaque V1 neurons. The Journal of Neurophysiology 85(5), 1873–1887 (2001)
19. Aharon, M., Elad, M.: Image denoising via sparse and re- dundant representations over learned dictionaries. IEEE Transactions on Image Processing 15(12), 3736–3745 (2006)
20. Cai, T.T., Wang, L.: Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise. IEEE Transactions on Information Theory 57(7), 4680–4688 (2011)
21. Tropp, J.A., Gilbert, A.C.: Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. IEEE Transactions on Information Theory 53(12) (December 2007)
22. Donoho, D.L., Tsaig, Y., Drori, I., Starck, J.-l.: Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, Technique Report (2006)
23. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput., vol. 1 (1993)
24. Bergeaud, F., Mallat, S.: Matching pursuit of images. In: Proc. International Conference on Image Processing, vol. 1, pp. 53–56 (1995)
25. Neff, R., Zakhor, A.: Very low bit-rate video coding based on matching pursuits. IEEE Transactions on Circuits and Systems for Video Technology 7(1), 158–171 (1997)
26. Mallat, S.G., Zhang, Z.: Matching Pursuits with Time-Frequency Dictionaries. IEEE Transactions on Signal Processing, 3397–3415 (December 1993)
27. Gunturk, B., Batur, A.U., Altunbasak, Y., Hayes, M.H., Mersereau, R.M.: Eigenface-domain super-resolution for face recognition. IEEE Transaction on Image Processing 12(5), 137–147 (2003)
28. Zhang, J., Pu, J., Chen, C., Fleischer, R.: Low-resolution gait recognition. IEEE Transaction on Systems, Man, and Cybernetic–Part B: Cybernetics 40(4), 986–996 (2010)
29. Galbraith, A., Theiler, J., Thome, K., Ziolkowski, R.: Resolution enhancement of multilook imagery for the multispectral thermal image. IEEE Transaction on Geoscience and Remote Sensing 43(9), 1964–1977 (2005)
30. Boucher, A., Kyriakidis, C., Collin, C.: Geo-statistical solutions for super-resolution land cover mapping. IEEE Transaction on Geoscience and Remote Sensing 46(1), 272–283 (2008)
31. Greenspan, H.: Super-resolution in medical imaging. The Computer Journal 52, 43–63 (2009)
32. Kennedy, J.A., Israel, O., Frenkel, A., Bar-Shalom, R., Azhari, H.: Super-resolution in pet imaging. IEEE Transaction on Medical Imaging 25(2), 137–147 (2006)
33. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision IJCV (2000)
34. Sun, J., Zheng, N.-N., Tao, H., Shum, H.: Image hallucinationwith primalsketch priors. In: Proc. CVPR (2003)
35. Chang, H., Yeung, D.-Y., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR (2004)
36. Roweis, S.T., Saul, L.K.: Nonlinear dimentionality reduction by locally linear embedding. In: Proc. CVPR (2003)

37. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: Proc. CVPR (2008)
38. Yang, J., Wright, J., Huang, T., Ma, Y.: Image Super-resolution via Sparse Representation. IEEE Transaction on Image Processing 19 (2010)
39. Elad, M., Aharon, M.: Image denoising via sparse and redundant representation over learned dictionaries. IEEE Transaction on Image Processing 15, 3736–3745 (2006)
40. Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representation for image and video restoration. Multiscale Modeling and Simulation 7, 214–241 (2008)
41. Tropp, J.: Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inf. Theory 50, 2231–2242 (2004)
42. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic, Norwell (1991)
43. Hamerly, G., Elkan, C.: Alternatives to the k-means algorithm that find better clusterings. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM) (2002)
44. Vattani, A.: k-means requires exponentially many iterations even in the plane. Discrete and Computational Geometry 45(4), 596–616 (2011)
45. Arthur, D., Manthey, B., Roeglin, H.: k-means has polynomial smoothed complexity. In: Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS) (2009)
46. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society, Series C 28(1), 100–108 (1979)
47. Dasgupta, S., Freund, Y.: Random Projection Trees for Vector Quantization. IEEE Transactions on Information Theory 55, 3229–3242 (2009)
48. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The Planar k-Means Problem is NP-Hard. In: Das, S., Uehara, R. (eds.) WALCOM 2009. LNCS, vol. 5431, pp. 274–285. Springer, Heidelberg (2009)
49. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc., Ser. B 39(1), 1–38 (1977)
50. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proc. ICCV (2009), `http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html`
51. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM Trans. Graph. 28(3), 1–10 (2010)

List of Acronyms

DCT	Discrete Cosine Transform
HR	High-Resolution
ICA	Independent Component Analysis
K-SVD	K-Singular Value Decomposition
LEE	Locally Linear Embedding
LR	Low-Resolution
MP	Matching Pursuit
MSE	Mean Square Error
MRF	Markov Random Field
NE	Neighborhood Embedding
OMP	Orthogonal Matching Pursuit

PCA	Principle Component Analysis
PSNR	Peak Signal-to-Noise Ratio
RMSE	Root Mean Square Error
SC	Sparse Coding
PSNR	Super-Resolution
RMSE	Singular Value Decomposition
VQ	Vector Quantization
WCSSR	Within Cluster Sum of Squares

Chapter 7
Sampling and Recovery of Continuously-Defined Sparse Signals and Its Applications

Akira Hirabayashi

Abstract. The common guideline for sampling continuously-defined signals has been provided by the Nyquist frequency for long. Recently it was clarified that even though signals in radar, echo, and sonar are wide-band with high Nyquist frequency, they can be sampled at extremely low frequency compared with the Nyquist frequency by taking the fact into account that such signals are sparse linear combinations of time-delayed versions of a transmitted (known) pulse. Such sampling scheme can also be applied to signals defined by piecewise polynomials or exponentials, in spite that they are not band-limited. In this article, we introduce a class of signals called signals with finite rate of innovation that covers not only the band-limited signals but also aforementioned non band-limited signals, and review sampling and reconstruction schemes for those signals in noiseless and noisy scenarios. This is followed by the more stable approach based on maximum likelihood estimation, which is connected to the so-called structured low-rank estimation. We further briefly introduce an application of these techniques to image feature extraction.

1 Introduction

Recently the low-cost of large storage devices has caused us to pay less attention to the size of the files we make. Yet, we have to pay attention to the size of the file when we wish to send or receive them. For example, let us consider a remote monitoring of EEG or ECG signals. To detect unexpected singular signals of epilepsy or arrhythmia, we need continuous monitoring of the signals for more than twenty four hours. Since the bandwidth of these signals is at least 100Hz, the Nyquist interval, which is the inverse of double of the bandwidth, amounts to 1/(2*100)=0.005 second. If we record 256 samples every second ($\sim$ 0.004 second apart) from sixty four channels of the EEG signals for twenty four hours, then the total amount of

Akira Hirabayashi
Ritsumeikan University, 1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan
e-mail: akirahrb@media.ritsumei.ac.jp

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_7, © Springer-Verlag Berlin Heidelberg 2014

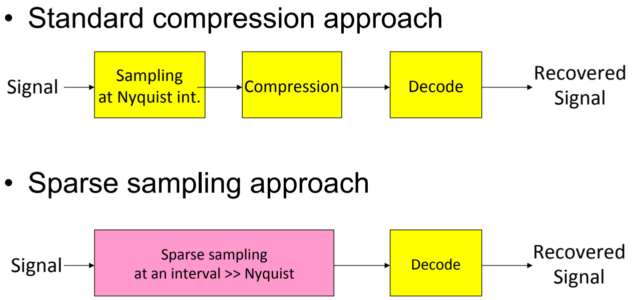

• Standard compression approach
Signal
Sampling at Nyquist int.
Compression
Decode
Recovered Signal
• Sparse sampling approach
Signal
Sparse sampling at an interval >> Nyquist
Decode
Recovered Signal

Fig. 1 Standard compression vs. sparse sampling

the data becomes approximately 2.8 giga bytes, which is equivalent to the data size of four compact discs. To send this amount of data from a patient to a server might be a considerable load to communication facility. Compression reduces the amount of data, but an encoder is necessary besides sampler. The equipment on the patient side should be as compact as possible. In sparse sampling techniques, a signal is measured at an interval much wider than the Nyquist one and the measurements are directly send to a receiver. Then, the original signal is recovered from the measurements at almost same quality as is in the standard compression approach. Fig. 1 illustrates the comparison of these two approaches.

There are two streams of sparse sampling. One is the so-called compressed sensing, which is a theory for discrete vectors [1]. On the other hand, if target signals are essentially continuous, and the Nyquist frequencies for these signals are very high, we wish to sample them as widely as possible and recover them at reasonable quality. A theory that enables to do so is that for signals with finite rate of innovation [2]. Typical example of the signal comes in radar, echo, and sonar. In these techniques, a pulse is transmitted to objects and time delays are estimated from reflected signals. Since the transmitted pulse is wide-band, the standard approach for time-delay estimation requires samples at the Nyquist frequency, which is very high. Since the waveform of the transmitted pulse is known, however, unknown parameters in the reflected signal are only time-delays and attenuation coefficients. It is redundant to sample the reflected signal at its Nyquist frequency only for the estimation of these parameters. If possible, we wish to sample the signals at a frequency which is close to the rate of the unknown parameters appearance. Then, the numbers of unknown parameters and balanced with that of conditions (measurements), and we can compute the parameters from the measurements. The rate of the unknown parameters appearance is the rate of innovation.

In this article, we define the class of signals with finite rate of innovation as a natural extension of band-limited signals, and review standard sampling and reconstruction schemes for those signals in noiseless and noisy scenarios. This is followed by the more stable approach based on maximum likelihood estimation.

In particular, we focus on two types of signals with finite rate of innovation: the stream of Diracs and piecewise polynomials. Further, we briefly introduce an application of these techniques to image feature extraction.

2 Signals with Finite Rate of Innovation as an Extension of Band-Limited Signals

As is well-known, if a signal $s(t)$ has a Fourier transform

$$\hat{s}(\omega) = \int_{-\infty}^{\infty} s(t)e^{-i\omega t}dt,$$

and if its support satisfies a condition that

$$\hat{s}(\omega) = 0 \quad (|\omega| \geq \omega_c), \tag{1}$$

then the signal can be completely reconstructed from measurements acquired by an interval T less than π/ω_c by

$$s(t) = \frac{\omega_c T}{\pi} \sum_{k=-\infty}^{\infty} f(kT)\mathrm{sinc}\left\{\frac{\omega_c(t-kT)}{\pi}\right\}, \tag{2}$$

[3],[4][1], where

$$\mathrm{sinc}(t) = \begin{cases} \sin(\pi t)/(\pi t) & (t \neq 0), \\ 1 & (t = 0). \end{cases}$$

The condition (1) is called band-limitation of lowpass type. The interval π/ω_c is nothing but the Nyquist interval, which is mostly an only guideline for sampling analog signals. If ω_c is small, the Nyquist interval takes a moderate value. If ω_c is large, however, the Nyquist interval gets a small value, which causes various problems including huge amount of data, computational cost, and data acquisition time, or hardware cost to implement it. To avoid these problems, we wish to sample the signal at a wide interval even if ω_c is very large.

To this end, signal treatment was revisited and a new class of signals was defined. Let us start with the generalization of (2), which means that the signal is expressed by a linear combination of the shifted version of $\frac{\omega_c T}{\pi}\mathrm{sinc}(\frac{\omega_c t}{\pi})$. The shift amount in (2) is kT while the coefficient is the sample value $f(kT)$. This can be generalized as follows: a signal is expressed by a linear combination of shifted versions of a known waveform $\varphi(t)$, but the shifted amount t_k and coefficients c_k are unknown. Without loss of generality, we suppose that $t_k < t_l$ if $k < l$. Then, the signal $s(t)$ is represented by

$$s(t) = \sum_{k=-\infty}^{\infty} c_k \varphi(t - t_k). \tag{3}$$

[1] The contribution by Someya to the sampling theory is summarized in [5].

More generally, consider a signal represented by linear combination of arbitrary shifts of R known functions $\varphi_r(t)$ $(r = 0, \ldots, R-1)$, but the shift amounts t_k and the coefficients $c_{k,r}$ are unknown. Then, the signal $s(t)$ is represented by

$$s(t) = \sum_{k=-\infty}^{\infty} \sum_{r=0}^{R-1} c_{k,r} \varphi_r(t - t_k). \tag{4}$$

The total number of t_k in period $[t_a, t_b]$ and $c_{k,r}$ with the identical k is denoted by $C_s(t_a, t_b)$. Then, we define a rate of innovation ρ as

$$\rho = \lim_{\tau \to \infty} \frac{1}{\tau} C_s(-\tau/2, \tau/2). \tag{5}$$

Definition 1. [2] A signal with a finite rate of innovation is a signal whose parametric representation is given in (4) and with a finite ρ, as defined in (5).

We can also define a local rate of innovation with respect to a moving (yet fixed) window size τ, as

$$\rho_\tau(t) = \frac{1}{\tau} C_s(t - \tau/2, t + \tau/2) \tag{6}$$

In this case, one is often interested in its maximum:

$$\rho_{\max}(\tau) = \max_{t \in \mathrm{IR}} \rho_\tau(t)$$

If a signal has a period τ, the local rate of innovation $\rho_\tau(t)$ is useful because it does not depend on t and gets a constant ρ. This article also discusses periodic signals $s(t)$, defined by

$$s(t) = \sum_{k' \in \mathbb{Z}} s_0(t - k'\tau), \tag{7}$$

where $s_0(t)$ is the signal in the interval $[0, \tau)$, given as

$$s_0(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} c_{k,r} \varphi_r(t - t_k). \tag{8}$$

In this case, we enforce the condition that $0 \leq t_0 < \cdots < t_{K-1} < \tau$.

The sequence of Diracs is $s(t)$ in (7) and (8) with $R = 1$ and $\varphi_0(t) = \delta(t)$. This is typically sparse, because its value is mostly zero except at positions t_k. Further, this ideal pulse sequence produces the general pulse sequence by convolving with $\varphi(t) \neq \delta(t)$. One generalization of the sequence of Diracs is the sequence of derivative of Diracs. This is $s(t)$ with $\varphi_r(t) = \delta^{(r)}(t)$, where the derivative of the Dirac is defined by

$$\int_{-\infty}^{\infty} \delta^{(r)}(t) \phi(t) dt = (-1)^r \phi^{(r)}(0),$$

with $\phi(t)$ an arbitrary function that has derivatives of any order and tends to zero more rapidly than any power of t, as $|t|$ tends to infinity [6]. This signal is mapped from piecewise polynomials by $R+1$th derivatives.

3 Sampling and Recovery of the Sequence of Diracs

Let us focus on the sampling and recovery of the sequence of Diracs. This signal is at the heart of the theory for signals with finite rate of innovation, because sequences of general pulses can be expressed by convolution with the sequence of Diracs and the general pulses. We quickly review how the signal is recovered from compressive measurements in noiseless and noisy cases. Since we discuss the periodic case, the Fourier coefficients of the sequence of Diracs are well-defined and given by

$$\hat{d}_p = \frac{1}{\tau}\int_0^{\tau} s(t)e^{-i2p\pi t/\tau}dt = \frac{1}{\tau}\sum_{k=0}^{K-1} c_k u_k^p, \tag{9}$$

where $u_k = e^{-i2\pi t_k/\tau}$.

3.1 Noiseless Case

Direct sampling $s(t)$ of the sequence of Diracs yields mostly zero measurements, which are useless. Instead, we sample it after filtering by $\psi(t)$ at $t = nT$. This can be expressed by the inner product as

$$d_n = \langle s, \psi_n \rangle = \int_{-\infty}^{\infty} s(t)\overline{\psi(t-nT)}dt, \tag{10}$$

where $n = 0 \sim N-1$ and $T = \tau/N$. Strictly speaking, this sample is called a generalized sample or measurement, which is distinguished from $s(t_n)$ called an ideal sample [7, 8]. There are various types of sampling kernels $\psi(t)$. For example, $\psi(t) = B\mathrm{sinc}(Bt)$ is an ideal lowpass filter. In this case, for perfect reconstruction, we suppose that

$$B \geq \rho = \frac{2K}{\tau}. \tag{11}$$

Other kernels of compact support, such as B-spline [9] and E-spline [10], are also used [11]. The kernel called sum-of-sincs is also of compact support with degrees of freedom for designing its waveform [12]. In this article, we use $\psi(t) = B\mathrm{sinc}(Bt)$ for periodic signals in Sections 3 and 4, while spline kernels are exploited in the image processing scenario in Section 5.

We wish to obtain $2K$ unknown parameters of $\{t_k\}_{k=0}^{K-1}$ and $\{c_k\}_{k=0}^{K-1}$ from the measurements $\{d_n\}_{n=0}^{N-1}$. In a noiseless case, this problem is solved elegantly by the so-called annihilating filter method (a.k.a. Prony's method) [2], as follows. Let P be an integer which does not exceed $B\tau/2$: $P = \lfloor B\tau/2 \rfloor$. Then, from the Poisson summation formula for the sinc function

$$B\sum_{k'=-\infty}^{\infty} \mathrm{sinc}\{B(t+k'\tau)\} = \frac{1}{\tau}\sum_{p=-P}^{P} e^{-i2p\pi t/\tau},$$

it holds that

$$d_n = \sum_{p=-P}^{P} \hat{d}_p e^{i2pn\pi/N}.$$

This equation implies that the Fourier coefficients $\hat{d}_p$ are exactly related to the measurements d_n acquired with the sinc kernel by the inverse discrete Fourier transform. Hence, if we assume that

$$N \geq 2P+1, \tag{12}$$

we can obtain $\hat{d}_p$ from d_n by the discrete Fourier transform (DFT), as

$$\hat{d}_p = \frac{1}{N} \sum_{n=0}^{N-1} d_n e^{-i2pn\pi/N}. \tag{13}$$

Its vector form is

$$\hat{\mathbf{d}} = F\mathbf{d}, \tag{14}$$

where $\mathbf{d}$ and $\hat{\mathbf{d}}$ are N and $2P+1$ dimensional vectors whose n and p elements are d_n and $\hat{d}_{p-P}$, respectively, and F is an accordingly defined DFT matrix.

The sequence $\{\hat{d}_p\}_{p=-P}^{P}$ in (9) is annihilated by a filter $[a_0, a_1, \ldots, a_K]$ whose z transform is

$$A(z) = \sum_{k=0}^{K} a_k z^{-k} = a_0 \prod_{k=0}^{K-1} (1 - u_k z^{-1}),$$

as

$$a_0 \hat{d}_p + a_1 \hat{d}_{p-1} + \ldots + a_K \hat{d}_{p-K} = 0 \quad (p = K-P, \ldots, P). \tag{15}$$

Simultaneously solving these equations provides the filter coefficients a_k. Concretely, let T be a $(2P-K+1) \times (K+1)$ matrix given by

$$T = \begin{pmatrix} \hat{d}_{K-P} & \hat{d}_{K-P-1} & \ldots & \hat{d}_{-P} \\ \hat{d}_{K-P+1} & \hat{d}_{K-P} & \ldots & \hat{d}_{-P+1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_P & \hat{d}_{P-1} & \ldots & \hat{d}_{P-K} \end{pmatrix}. \tag{16}$$

Then, the vector $\mathbf{a} = [a_0, a_1, \ldots, a_K]^T$ is a solution to the equation

$$T\mathbf{a} = 0. \tag{17}$$

To solve this, we note that the definition of P implies $P \leq B\tau/2 < P+1$ and (11) stands for $K \leq B\tau/2$. Therefore, it holds that

$$P \geq K. \tag{18}$$

Hence, the matrix T is square when $P = K$ and vertically longer rectangular when $P > K$. The number of columns implies that the rank of T can be $K+1$, but the number of t_k enforces that $\mathrm{rank}(T) = K$. Hence, in the singular value decomposition (SVD) USV of T, there is one singular value of zero and its corresponding column vector of V gives the filter $\mathbf{a}$. It is also important to note that T has a Toeplitz structure, which plays an essential role in the noisy scenario with $\mathrm{rank}(T) = K$.

Once t_k are fixed, (9) reduces to linear equations in terms of c_k. That is, let a Vandermonde matrix be

$$U_t = \begin{pmatrix} u_0^{-P} & u_1^{-P} & \dots & u_{K-1}^{-P} \\ u_0^{-P+1} & u_1^{-P+1} & \dots & u_{K-1}^{-P+1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ u_0^{P} & u_1^{P} & \dots & u_{K-1}^{P} \end{pmatrix}.$$

By solving the equation

$$U_t\mathbf{c} = \hat{\mathbf{d}}, \tag{19}$$

we finally obtain the coefficients $\mathbf{c} = [c_0, \dots, c_{K-1}]^T/\tau$, which is unique because t_k are distinct each other. We can summarize the above discussion as follows:

Theorem 1. [2] *If the sampling kernel $\psi(t) = B\mathrm{sinc}(Bt)$ satisfies the condition (11) and N is greater than $2P+1$, then the measurements $\{d_n\}_{n=0}^{N-1}$ obtained by (10) is a sufficient characterization of a τ-periodic sequence of Diracs.*

From (12) and (18), it holds that

$$N \geq 2K+1,$$

which implies that $s(t)$ can be perfectly reconstructed from $2K+1$ measurements per period. This is one more than the number of unknown parameters $2K$. This is because we need to know $2K$ Fourier coefficients $\hat{d}_p$, which we do not have direct access. Instead, we have to compute them from the measurements d_n using (13). Because of conjugate symmetry of the Fourier series, we need an odd number of the coefficients to express real values. As a result, we need an odd number of measurements to invert it. It is not novel to obtain t_k and c_k from $\hat{d}_p$ of the form of (9). Mathematically same problems can be found in spectral estimation and direction of arrival (DoA) estimation. Conventional methods for these problems, such as MUSIC [13], ESPRIT [14], and Matrix Pencil [15], exploit redundant number of measurements. On the other hand, we used the annihilating filter approach to clarify the minimum number of measurements for perfect reconstruction. As the great contributions of [2], it was shown that $\hat{d}_p$ can be exactly computed from d_n by DFT and that the unknown parameters are determined using the annihilating filter method as well as those conventional methods in the spectral or DoA estimations.

3.2 Cadzow Denoising

Let y_n be a noisy measurements, as $y_n = d_n + e_n$. Its vector form is

$$\mathbf{y} = \mathbf{d} + \mathbf{e}. \tag{20}$$

We assume that the probability density function of $p(\mathbf{e})$ is known. Let us denote the DFT of $\mathbf{y}$ by $\hat{\mathbf{y}} = (\hat{y}_{-P}, \hat{y}_{-P+1}, \ldots, \hat{y}_P)^T = F\mathbf{y}$. Generally, there does not exist $\mathbf{a}$ that annihilates (17), in which $\hat{d_p}$ is replaced by $\hat{y}_p$. A simple remedy to this situation is to use $\mathbf{a}$ that minimizes the squared norm of the residue $T\mathbf{a}$. Its solution is provided by the singular value decomposition USV^* of T as well as in the noiseless case: the column vector of V corresponding to the smallest singular value gives the filter coefficients. Note that, because of the noise, the smallest singular value does not yield zero. This method is called the least square (LS) method.

To improve the quality of the LS method, the so-called Cadzow denoising [16] is usually used [17]. This algorithm exploits the two facts mentioned above: one is that the rank of T should be K and the other is that T has a Toeplitz structure. In general, T can be rectangular ($P > K$), but it has been empirically shown that the algorithm works the best when T is chosen to be square ($P = K$) [17]. The algorithm proceeds as follows:

1. Compute the DFT $\hat{y}_p$ of y_n.
2. Set a square matrix Y as

$$Y = \begin{pmatrix} \hat{y}_0 & \hat{y}_{-1} & \cdots & \hat{y}_{-P} \\ \hat{y}_1 & \hat{y}_0 & \cdots & \hat{y}_{-P+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_P & \hat{y}_{P-1} & \cdots & \hat{y}_0 \end{pmatrix}.$$

3. Repeat the following procedure until a termination condition is met,
   a. Compute the SVD of $Y = USV^*$, where U, S, and V are $P+1$ square matrices.
   b. Compute $Y' = US'V^*$ using S', in which $P-K+1$ smallest singular values of S are replaced by zero. Now, Y' does not have the Toeplitz structure anymore.
   c. To recover the structure, compute averages along diagonal parallel elements and replace all of the elements by the corresponding average. For example, if $P = 2$, do

$$Y' = \begin{pmatrix} y'_{0,0} & y'_{0,1} & y'_{0,2} \\ y'_{1,0} & y'_{1,1} & y'_{1,2} \\ y'_{2,0} & y'_{2,1} & y'_{2,2} \end{pmatrix} \quad \Rightarrow \quad Y := \begin{pmatrix} \hat{y}'_0 & \hat{y}'_{-1} & y'_{0,2} \\ \hat{y}'_1 & \hat{y}'_0 & \hat{y}'_{-1} \\ y'_{2,0} & \hat{y}'_1 & \hat{y}'_0 \end{pmatrix},$$

   where $\hat{y}'_1$, $\hat{y}'_0$ and $\hat{y}'_{-1}$ are given as

$$\hat{y}'_1 = \frac{y'_{1,0} + y'_{2,1}}{2}, \quad \hat{y}'_0 = \frac{y'_{0,0} + y'_{1,1} + y'_{2,2}}{3}, \quad \hat{y}'_{-1} = \frac{y'_{0,1} + y'_{1,2}}{2}.$$

The termination condition can be that the ratio of the Kth singular value to the $K+1$th one is greater than a threshold, but normally ten or twenty times of repetition provides a Toeplitz matrix with rank of K. This type of problem, in which a matrix is approximated by another one with a structure and low-rank, is called Structured Low-Rank Approximation (SLRA) and a hot topic in a signal/image processing and optimization [18]. In the end, we should note that, even though the set of all Toeplitz matrices is convex, that of rank of K is not. Hence, any optimality and convergence are not guaranteed in this algorithm.

3.3 Maximum Likelihood Estimation

To resolve the difficulties mentioned in the previous section, we can exploit the formalism of maximum likelihood estimation. Equations (14) and (19) yield

$$\mathbf{d} = F^{-1}U_t\mathbf{c}. \tag{21}$$

Then, we can define the log-likelihood function as $L(\mathbf{t},\mathbf{c}) = \log p(\mathbf{y} - F^{-1}U_t\mathbf{c})$, where $\mathbf{t} = [t_0\ t_1\ \cdots\ t_{K-1}]^{\mathrm{T}}$. Assume that $p(\mathbf{e})$ is the Gaussian distribution with zero mean and covariance matrix $\sigma^2 I$, where σ is a known positive real and I is the identity matrix. Then, the log-likelihood function reads

$$L(\mathbf{t},\mathbf{c}) = -\frac{\|\mathbf{y} - F^{-1}U_t\mathbf{c}\|^2}{2\sigma^2} - N\log(\sqrt{2\pi}\sigma). \tag{22}$$

This implies that the maximization of the log-likelihood function is equivalent to the minimization of the norm $\|\mathbf{y} - F^{-1}U_t\mathbf{c}\|^2$. Further on, F is unitary up to constant. Hence, this minimization is equivalent to that of

$$f_o(\mathbf{t},\mathbf{c}) = \|\hat{\mathbf{y}} - U_t\mathbf{c}\|^2. \tag{23}$$

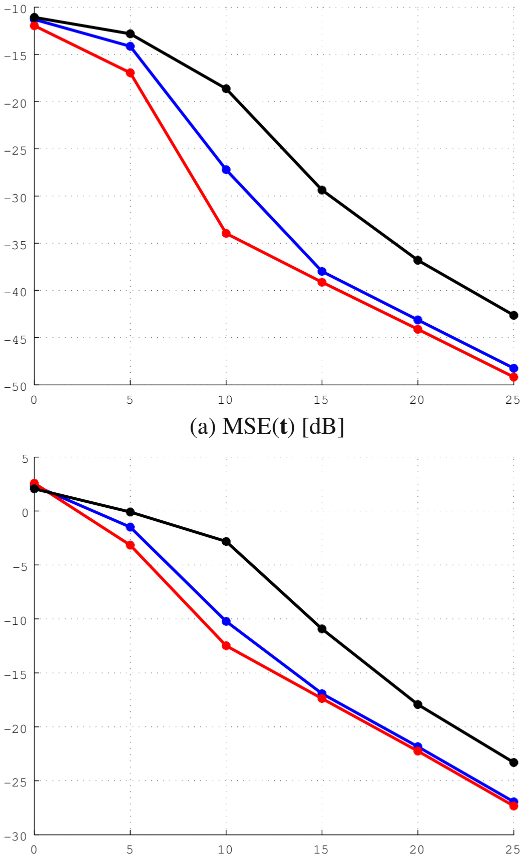
Finally, the maximum likelihood estimation amounts to estimating the vector $U_t\mathbf{c}$, which is the closest to $\hat{\mathbf{y}}$ in the least-squares sense, in Fourier domain.

Equation (23) is quadratic with respect to $\mathbf{c}$, when $\mathbf{t}$ is fixed. Therefore, the optimal $\mathbf{c}$ for a fixed $\mathbf{t}$ is obtained analytically as $\mathbf{c} = U_t^{\dagger}\hat{\mathbf{y}}$, where $(\cdot)^{\dagger}$ stands for the Moore-Penrose generalized inverse of the bounded operator [19]. Hence, the minimizer of $f_o(\mathbf{t},\mathbf{c})$ is found by searching $\mathbf{t}$ that minimizes

$$f(\mathbf{t}) = f_o(\mathbf{t}, U_t^{\dagger}\hat{\mathbf{y}}) = \|\hat{\mathbf{y}} - U_tU_t^{\dagger}\hat{\mathbf{y}}\|^2,$$

and then by computing $\mathbf{c} = U_t^{\dagger}\hat{\mathbf{y}}$.

The criterion $f(\mathbf{t})$ is non-convex and it is very difficult to find the global minimum solution. We thus exploit the so-called particle swarm optimization (PSO) algorithm [20]. The particles model the parameter $\mathbf{t}$ to be optimized. For each particle $j = 1,\ldots,J$, we first initialize the position $\mathbf{t}_j$ and its velocity $\dot{\mathbf{t}}_j$ with uniformly distributed random vectors in the domain. We use the particle's and swarm's best known positions $\mathbf{b}_j^{(p)}$ and $\mathbf{b}^{(s)}$, which are initialized by $\mathbf{b}_j$ and the best among the



(a) MSE(**t**) [dB]

(b) MSE(**c**) [dB]

Fig. 2 Mean square errors (MSE) of estimated parameters for **t** and **c** with respect to the SNR. The number of measurements is 11. The red, blue and black lines show the results by the proposed method, by LS with and without Cadzow denoising, respectively.

initial positions, respectively. Until a termination criterion is met, the particle's velocity $\dot{\mathbf{t}}_j$ and position $\mathbf{t}_j$ are updated by

$$\dot{\mathbf{t}}_j \leftarrow w\dot{\mathbf{t}}_j + c_1 r_1(\mathbf{b}_j^{(p)} - \mathbf{t}_j) + c_2 r_2(\mathbf{b}^{(s)} - \mathbf{t}_j),$$
$$\mathbf{t}_j \leftarrow \mathbf{t}_j + \dot{\mathbf{t}}_j,$$

respectively, where c_1 and c_2 are pre-defined constants near 1 and r_1, r_2 are uniform random variables within 0 and 1. If $f(\mathbf{t}_j) < f(\mathbf{b}_j^{(p)})$, then $\mathbf{b}_j^{(p)}$ is updated by $\mathbf{t}_j$. If $f(\mathbf{b}_j^{(p)}) < f(\mathbf{b}^{(s)})$, then $\mathbf{b}^{(s)}$ is replaced by $\mathbf{b}_j^{(p)}$. Finally, $\mathbf{b}^{(s)}$ gives the best found solution. Because of its global and random nature, PSO is more robust than gradient approaches, against getting trapped in local minima. The drawback is a relatively high computational cost.

In simulations, the parameters are set as $\tau = 1$, $b_p = 1$, $K = 2$ and $N = 11$. The unknown parameters are $\mathbf{t} = (t_0, t_1) = (0.42, 0.52)$, and $\mathbf{c} = (c_0, c_1) = (1.00, 1.00)$. For PSO, we used $J = 150$ particles and $(w, c_1, c_2) = (0.4, 0, 9, 0.4)$, $(0.9, 0.4, 0.4)$ and $(0.4, 0.4, 0.9)$ for 75, 45 and 30 particles, respectively. Thousand of noise vector $\mathbf{e}$ were generated from the Gaussian distribution in which σ was determined so that SNR[2] becomes 0, 5, ..., 25[dB]. For each experiment, we computed estimates $\hat{\mathbf{t}}$ and $\hat{\mathbf{c}}$ of $\mathbf{t}$ and $\mathbf{c}$, for 1,000 different noise realizations. Accordingly, the mean square errors MSE($\mathbf{t}$) and MSE($\mathbf{c}$) were defined as the average over the 1,000 trials of $||\hat{\mathbf{t}} - \mathbf{t}||^2$ and $||\hat{\mathbf{c}} - \mathbf{c}||^2$, respectively. The results are shown in Fig. 2, where the red, blue and black lines show the results by the proposed method, by the LS methods with and without Cadzow denoising, respectively. We can see that the proposed method outperforms the conventional methods for every value of SNR except for 0dB of MSE($\mathbf{c}$), in which case three approaches mostly give the same result because of large noise level.

4 Sampling and Recovery of Signals of Piecewise Polynomials

Piecewise polynomial is a powerful tool for signal and image processing. Its special case is the polynomial spline, which is a standard tool for interpolation. A τ-Periodic piecewise polynomial with K jointing points is defined as follows. For every $k = 0, \ldots, K-2$, let us define the function $\varphi_k(t)$ as

$$\varphi_k(t) = \begin{cases} v_k(t) & (t_k < t < t_{k+1}), \\ 0 & (\text{otherwise}), \end{cases}$$

and the function $\varphi_{K-1}(t)$ as

$$\varphi_{K-1}(t) = \begin{cases} v_{K-1}(t+\tau) & (0 \le t < t_0), \\ v_{K-1}(t) & (t_{K-1} < t < \tau), \\ 0 & (\text{otherwise}), \end{cases}$$

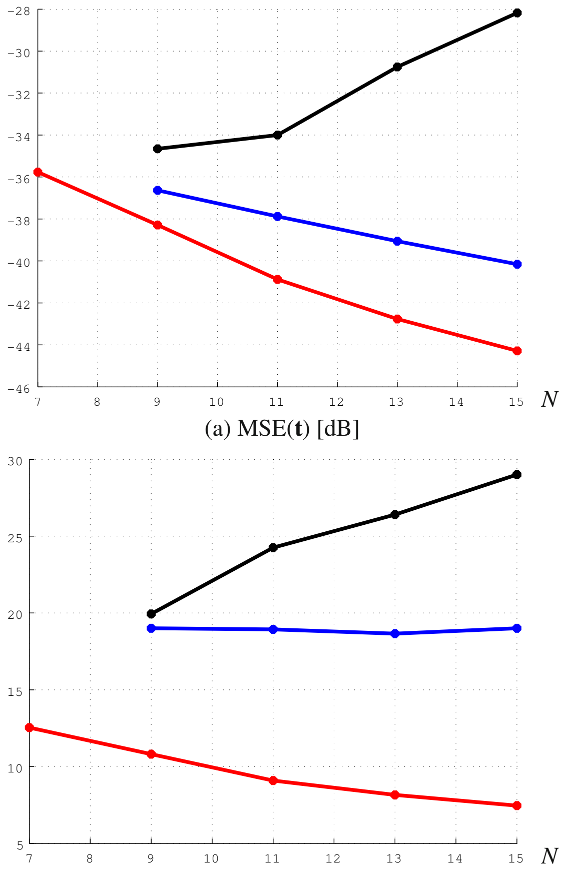
where $v_k(t) = \sum_{r=0}^{R} \alpha_{k,r} t^r$. Then, a τ-periodic piecewise polynomial $s(t)$ of degree R is defined by $s(t)$ in (7) with

$$s_0(t) = \sum_{k=0}^{K-1} \varphi_k(t).$$

The piecewise polynomials are signals with finite rate of innovation, because $s(t)$ has K degrees of freedom from the positions t_k and $(R+1)K$ from the coefficients $\alpha_{k,r}$ per period. This implies that the rate of innovation is $\rho = K(R+2)/\tau$.

The measurements of piecewise polynomials acquired with the sinc kernel can be expressed by the parameters t_k and $\alpha_{k,r}$ as follows. Let us introduce matrices D, V_t, and $\tilde{V}_t$ as

[2] The SNR is defined by $10\log_{10} \dfrac{||\mathbf{d}||^2}{\sigma^2 N}$.



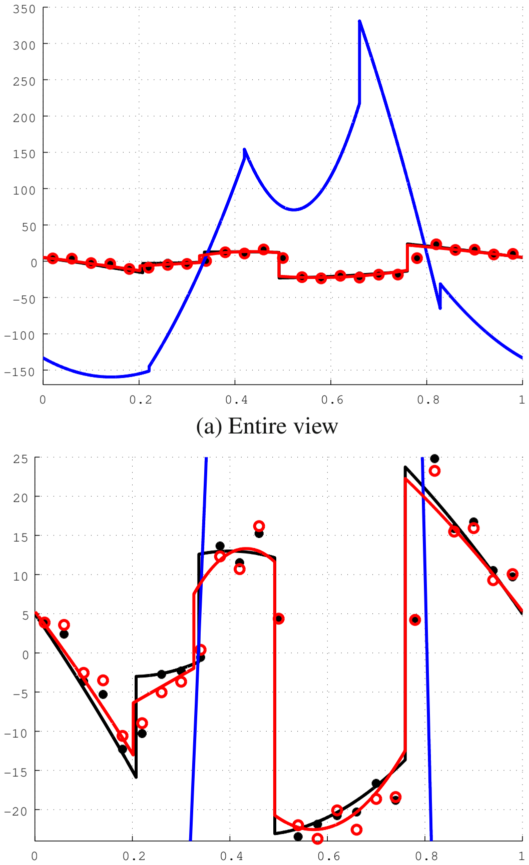
(a) MSE(**t**) [dB]

(b) MSE(α) [dB]

Fig. 3 MSE [dB] of the estimated parameters for **t** and α of a piecewise polynomial with respect to the number N of measurements. The legends are the same as in Fig. 2.

$$D = \left(\left(\frac{\tau}{i2\pi} \mathrm{diag} \left(\frac{1}{-P}, \frac{1}{-P+1}, \ldots, \frac{1}{P} \right) \right)^{R+1} \begin{matrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{matrix} \right),$$

$$V_t = \begin{pmatrix} u_0^{-P} & \cdots & (-P)^R u_{K-1}^{-P} \\ u_0^{-P+1} & \cdots & (-P+1)^R u_{K-1}^{-P+1} \\ \vdots & \ddots & \vdots \\ u_0^{P} & \cdots & (P)^R u_{K-1}^{P} \end{pmatrix}, \quad \tilde{V}_t = \begin{pmatrix} V_t & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix},$$



(a) Entire view

(b) Enlarged view

Fig. 4 A simulation example with $K = 4$ and $R = 2$. The black line shows the target signal and the red circles and black dots are measurements with and without 20dB noise. The red and blue lines are reconstructed signals by the proposed method and LS with Cadzow denoising, respectively.

with $\mathbf{0}$ indicating the zero vector. Note that the $R+1$th derivative of the piecewise polynomial in the sense of distribution is a sequence of derivatives of Diracs [2]:

$$s_0(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} c_{k,r} \delta^{(r)}(t - t_k),$$

where $\delta^{(r)}(t)$ is the r th derivative of the Dirac. This relation gives a meaning to the matrix D as the mapping from the Fourier coefficients of the sequence of derivatives of Diracs to those of the piecewise polynomial. Further, we introduce a matrix W_t which maps $\alpha_{k,r}$ to the coefficients $c_{k,r}$ of the sequence of derivatives of Diracs.

For instance when $R = 1$ and $K = 2$, W_t is given as

$$W_t = \begin{pmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & t_0 & -(t_0 + \tau) \\ 0 & 0 & -1 & 1 \\ -1 & 1 & -t_1 & t_1 \\ \frac{t_1 - t_0}{\tau} & \frac{t_0 + \tau - t_1}{\tau} & \frac{t_1^2 - t_0^2}{2\tau} & \frac{(t_0+\tau)^2 - t_1^2}{2\tau} \end{pmatrix}.$$

We refer to [21] for further details on the matrix W_t. Then, it holds that [21]

$$\mathbf{d} = F^{-1} D \tilde{V}_t W_t \alpha,$$

where $\alpha = (\alpha_{0,0} \ \cdots \ \alpha_{K-1,R})^{\mathrm{T}}$. That is, the noiseless measurements of the piecewise polynomial are expressed by using the locations t_k and the coefficients $\alpha_{k,r}$. Because of this expression, the log-likelihood function is defined similarly as in (22) and its maximization is equivalent to the minimization of $\|\mathbf{y} - \Phi_t \alpha\|^2$. We find the minimizer of this term by searching $\mathbf{t}$ that minimizes $\|\mathbf{y} - \Phi_t \Phi_t^{\dagger} \mathbf{y}\|^2$, and then calculating $\alpha = \Phi_t^{\dagger} \hat{\mathbf{y}}$. The search of the minimizer was again conducted by PSO.

The performance of the proposed method was evaluated by simulations. The target signal is a $\tau = 1$-periodic piecewise polynomial of degree $R = 1$ with $K = 2$ discontinuities. The unknown parameters are $\mathbf{t} = (0.20, 0.65)$ and $\alpha = (\alpha_{0,0}, \alpha_{0,1}, \alpha_{1,0}, \alpha_{1,1}) = (-1.00, -3.00, 2.00, 4.00)$. We reconstructed the signal from 7, 9, ..., 15 measurements with 20dB noise. The estimation errors MSE($\mathbf{t}$) and MSE(α) were obtained by averaging $\|\hat{\mathbf{t}} - \mathbf{t}\|^2$ and $\|\hat{\alpha} - \alpha\|^2$ over 1,000 noise realizations, respectively. The results are shown in Fig. 3, with same legends as in Fig. 2. We can see that the proposed method outperforms the conventional methods in all cases.

A simulation example with $K = 4$, $R = 2$, and $N = 25$ is shown in Fig. 4. We can see that the proposed method gives much better results than the classical approach. We should note that $N = 25$ is the minimum for the classical approach while the proposed method can reconstruct the signal from fewer samples. It took 19.12s for the proposed method to reconstruct the signal, while LS with Cadzow denoising required 0.06s only. It should be noted that Matlab is far from optimal for the implementation of algorithms like PSO, whose potential for parallelization is not exploited at all.

5 Application to Image Feature Extraction

Straight line-edge is one of the most important image feature used in many applications including registration or vehicle navigation. The standard method to extract straight lines is the Hough transform and its extensions [22, 23]. Such techniques, however, have limitations including the fact that many parameters need to be adjusted or the fact that they require high computational costs. Further, the Hough transform uses center position of a detected pixel as location of the line. Since this is not true generally, preciseness of the method degrades as the resolution of image decreases. This difficulty was reduced by using more precise acquisition model [24],

[25], [21]. Since step line edge can be expressed by three parameters of orientation, offset, and amplitude, it can be regarded as a signal with finite rate of innovation. Hence, the techniques developed there can be exploited here as well.

By using orientation θ, offset γ, and amplitude λ, as defined in Fig. 5, a step line-edge can be expressed as

$$f(x,y) = \lambda u(-x\sin\theta + y\cos\theta + \gamma\sin\theta), \tag{24}$$

where $u(t)$ is the unit step function whose value is 1 if $t \geq 0$ and 0 if $t < 0$. This continuously-defined step line-edge is sampled by the integer-shifted version of a sampling kernel $\psi(x)\psi(y)$ as

$$g[m,n] = \langle f(x,y), \psi(x-m)\psi(y-n)\rangle + \varepsilon[m,n],$$

where $\varepsilon[m,n]$ is additive noise. The sampling kernel $\psi(t)$ is modeled by the trigonometric E-spline of the first order ($P = 1$) [10], which is given by

$$\beta_\alpha(t) = \begin{cases} \sin\omega_0(t+1)/\omega_0 & (-1 \leq t < 0), \\ -\sin\omega_0(t-1)/\omega_0 & (0 \leq t < 1), \\ 0 & (t \leq -1, t > 1). \end{cases}$$

When ω_0 tends to zero, the trigonometric E-spline converges to the B-spline of the first order.

The algorithm proceeds as follows. First, edge pixels are detected by a conventional method like Canny operator. Then, for each pixel detected as an edge, the surrounding pixel area is extracted, and the three parameters are computed from the pixels in the area. To suppress extraction errors, similar edges are merged, while other edges are discarded. Within these steps, we mainly discuss the second one. Therefore, the indices m and n are assigned in a local manner: the focused detected pixel is set to $m = n = 0$. The local area size is chosen as 8 × 8 pixels since those affected by the focused edge are mostly within this area. This is because the sampling kernel is modeled by the E-spline of the first order (its support width is two). Hence, the indices m and n are from -3 to 4 (see Fig. 5).

To retrieve the parameters θ, γ, and λ from the pixel values $g[m,n]$, we first compute a horizontal differentiated sample $d_H[m,n]$ which is given by $g[m+1,n] - g[m,n]$. We then compute product-sum of $d_H[m,n]$ and coefficients $C_m^{(\alpha_p)}$:

$$\tau_{n,p}^{(H)} = \sum_{m=-3}^{3} C_m^{(\alpha_p)} d_H[m,n]. \tag{25}$$

The coefficients $C_m^{(\alpha_p)}$ are determined so that they satisfy

$$\sum_{m=-\infty}^{\infty} C_m^{(\alpha_p)} (\beta_{\alpha_2} * \psi)(t-m) = e^{\alpha_p t} \tag{26}$$

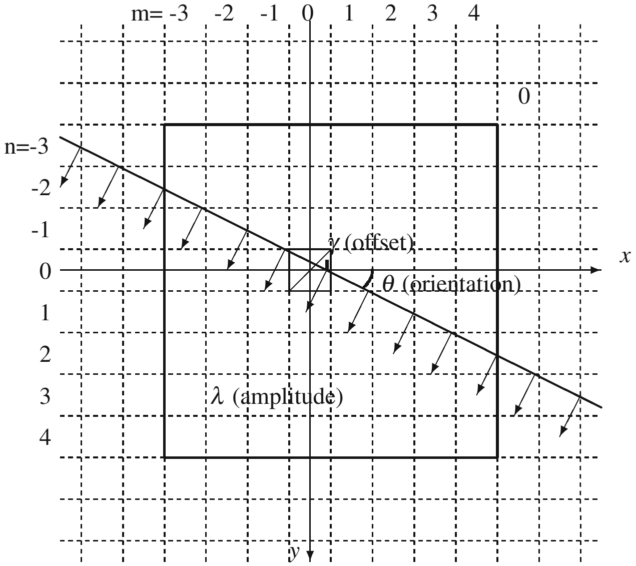


Fig. 5 Description parameters for a step line-edge. The *xy* coordinates are local ones whose origin is the center of the pixel detected as an edge. The grid shows sampled pixels.

for $p = 0,1,2$, where $\beta_{\alpha_p}(t)$ is defined by

$$\beta_{\alpha_p}(t) = \begin{cases} e^{\alpha_p t} & (-0.5 \le t < 0.5), \\ 0 & (t < -0.5, t \ge 0.5), \end{cases} \tag{27}$$

with $\alpha_2 = 0$. The coefficients can be computed by

$$C_m^{(\alpha_p)} = e^{m\alpha_p} \Big/ \left\{ \sum_{k=-P'}^{P'} e^{k\alpha_p} (\beta_{\alpha_2} * \psi)(-k) \right\}, \tag{28}$$

where P' is the maximum integer not exceeding $(P+2)/2$. Note that the convolved sampling kernel $(\beta_{\alpha_2} * \psi)(t)$ can produce $e^{\alpha_p t}$ for $p = 0,1,2$.

To show a closed form of $\tau_{n,p}^{(H)}$, let us define

$$\mu_{n,p}^{(H)}(\theta,\gamma) = -\mathrm{sgn}(\sin\theta) e^{\alpha_p(\gamma + \frac{n}{\tan\theta} - \frac{1}{2})} \Psi\left(\frac{\alpha_p}{\tan\theta}\right),$$

where $\mathrm{sgn}(t)$ is the function whose value is 1 if $t > 0$, 0 if $t = 0$, and -1 if $t < 0$ and $\Psi(s) = \int_{-\infty}^{\infty} \psi(t) e^{st} dt$. Assume that $d_H[m,n]$ is equal to zero for $|m| \ge 4$. Then, as shown in [25], it holds for $p = 0,1,2$ that

$$\tau_{n,p}^{(H)} = \lambda\, \mu_{n,p}^{(H)}(\theta,\gamma). \tag{29}$$

This equation yields closed formulas for $\tan\theta$, γ, and λ [25]. as

$$\lambda = \frac{|\tau_{0,2}^{(H)}|}{\Psi(0)}, \ \ \tan\theta = \frac{\omega_0}{\angle(\tau_{1,0}^{(H)}/\tau_{0,0}^{(H)})},$$

$$\gamma = \frac{1}{\omega_0}\angle\left(\frac{\tau_{0,0}^{(H)}\operatorname{sgn}(\tau_{0,2}^{(H)})}{\lambda\Psi(\alpha_0/\tan\theta)}\right) + \frac{1}{2}, \tag{30}$$

where $\angle(z)$ is the phase angle of complex number z. If model mismatch is not too severe, then $\tau_{n,p}^{(H)}$ can be computed by (25) and the closed formulas can provide good estimates for λ, θ, and γ. However, if model mismatch cannot be ignored, it is getting hard for the formulas to work precisely. To overcome this limitation, we search for θ, γ, and λ by which the right-hand side in (29) best approximates the left-hand side for all $n = -1,0,1$ and $p = 0,1,2$.

As well as the horizontal one, we can do the same processing vertically. That is, let us compute differentiated samples vertically as $d_V[m,n] = g[m,n+1] - g[m,n]$. Then, the product-sum of $d_V[m,n]$ and coefficients $C_n^{(\alpha_p)}$ is computed by

$$\tau_{m,p}^{(V)} = \sum_{n=-\infty}^{\infty} C_n^{(\alpha_p)} d_V[m,n] \tag{31}$$

and $\tau_{m,p}^{(V)}$ has the following closed form:

$$\tau_{m,p}^{(V)} = \lambda\,\mu_{m,p}^{(V)}(\theta,\gamma), \tag{32}$$

where

$$\mu_{m,p}^{(V)}(\theta,\gamma) = \operatorname{sgn}(\cos\theta)e^{\alpha_p\{-(\gamma-m)\tan\theta-\frac{1}{2}\}}\Psi(\alpha_p\tan\theta).$$

Let us define eighteen dimensional vectors τ and $\mu(\theta,\gamma)$ as

$$\tau = (\tau_{-1,0}^{(H)}, \tau_{-1,1}^{(H)}, \tau_{-1,2}^{(H)}, \tau_{0,0}^{(H)}, \dots \tau_{1,1}^{(V)}, \tau_{1,2}^{(V)})^T,$$

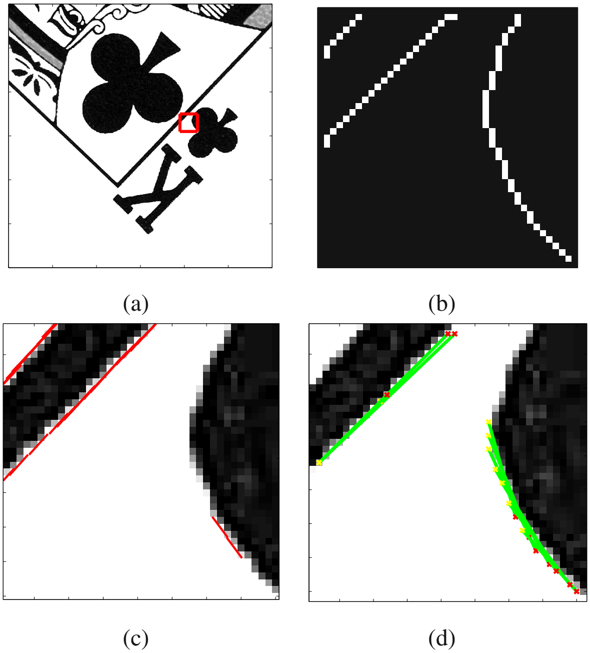
$$\mu(\theta,\gamma) = (\mu_{-1,0}^{(H)}(\theta,\gamma), \mu_{-1,1}^{(H)}(\theta,\gamma), \mu_{-1,2}^{(H)}(\theta,\gamma),$$
$$\mu_{0,0}^{(H)}(\theta,\gamma), \dots, \mu_{1,1}^{(V)}(\theta,\gamma), \mu_{1,2}^{(V)}(\theta,\gamma))^T.$$

Then, the differences between the left and right hand sides in (29) and (32) can be simultaneously evaluated by

$$J_o(\theta,\gamma,\lambda) = \|\lambda\mu(\theta,\gamma) - \tau\|^2.$$

For fixed θ and γ, the optimal λ is obviously given by

$$\lambda_{\text{opt}}(\theta,\gamma) = \left\langle \tau, \frac{\mu(\theta,\gamma)}{\|\mu(\theta,\gamma)\|^2} \right\rangle.$$


(a) (b)

(c) (d)

Fig. 6 Edge extraction results by the proposed algorithm: (a) original image, (b) Canny edge detection results, (c) results by the proposed method, (d) results by the Hough transform.

Hence, θ, γ, and λ which minimize J_o are given by θ and γ which minimize

$$J(\theta,\gamma) = J_o(\theta,\gamma,\lambda_{\text{opt}}(\theta,\gamma)), \tag{33}$$

and then $\lambda_{\text{opt}}(\theta,\gamma)$ with the resultant values.

We applied the proposed method to the real image obtained by a Nikon D50 SLR camera. Its point spread function (PSF) is simply approximated by the trigonometric E-spline of the first order with $\omega_0 = \pi/8$ without any calibration. Fig. 6 (a) shows the original image, Fig. 6 (b) shows the Canny edge detection results. Figs. 6 (c) and (d) show the extracted edges for the small area in the box indicated in Fig. (a) by the proposed method and the Hough transform, respectively. Even though it is difficult to evaluate these results quantitatively, we can see that the Hough transform extracts many wrong straight line edges along the curve on the right while the proposed method does more precise results and less wrong ones. We also note that the Hough transform could not extract the top left straight line edge because the area shown in Fig. (d) was not sufficient. Even though the PSF for the camera is unknown, the proposed method showed the good performance. This means that the proposed approach is robust against the PSF model mismatch. The computational time for the simulation by the proposed method was 0.7[s] which can be accelerated by more dexterous initial values.

6 Conclusion

We provided a short tutorial on the theory for signals with finite rate of innovation and its application to image feature extraction. To sample a signal at low frequency compared with its Nyquist frequency, the signal was characterized using how frequently unknown parameters appear in its parametric expression, instead of the classical frequency. The new frequency of the parameter appearance was defined as the rate of innovation. We focused on the two typical examples of signals with finite rate of innovation: the sequence of Diracs and piecewise polynomials. Measurements of both signals were acquired with an appropriate sampling kernel and recovered stably using a maximum likelihood estimation with the so-called particle swarm optimization (PSO). Using the similar technique, we showed that step line-edges can be extracted very precisely. Interesting applications, which we could not mention here, include compressive sensing of the EEG signals [26], vehicular signals [27], and more.

References

1. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory 52(4), 1289–1306 (2006)
2. Vetterli, M., Marziliano, P., Blu, T.: Sampling signals with finite rate of innovation. IEEE Transactions on Signal Processing 50(6), 1417–1428 (2002)
3. Shannon, C.: Communications in the presence of noise. In: Proc. IRE, vol. 37, pp. 10–21 (1949)
4. Someya, I.: Hakei Denso. Shukyosha, Tokyo (1949)
5. Ogawa, H.: Sampling theory and Isao Someya: A historical note. Sampling Theory in Signal and Image Processing 5(3), 247–256 (2006)
6. Papoulis, A.: The Fourier Integral and Its Applications. McGraw Hill, New York (1962)
7. Ogawa, H.: A generalized sampling theorem. Electronics and Communications in Japan, Part. 3 72(3), 97–105 (1989)
8. Eldar, Y., Dvorkind, T.: A minimum squared-error framework for generalized sampling. IEEE Transactions on Signal Processing 54(6), 2155–2167 (2006)
9. Unser, M.: Splines: A perfect fit for signal and image processing. IEEE Signal Processing Magazine 16(6), 22–38 (1999)
10. Unser, M., Blu, T.: Cardinal exponential splines: Part I—Theory and filtering algorithms. IEEE Transactions on Signal Processing 53(4), 1425–1438 (2005)
11. Dragotti, P.L., Vetterli, M., Blu, T.: Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix. IEEE Transactions on Signal Processing 55(5), 1741–1757 (2007)
12. Tur, R., Eldar, Y., Friedman, Z.: Innovation rate sampling of pulse streams with application to ultrasound imaging. IEEE Transactions on Signal Processing 59(4), 1827–1842 (2011)
13. Schmidt, R.: Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas Propagation 34(3), 276–280 (1986)
14. Roy, R., Paulraj, A., Kailath, T.: ESPRIT a subspace rotation approach to estimation of parameters of cisoids in noise. IEEE Transactions on Acoustic, Speech, Signal Processing 34(5), 1340–1342 (1986)
15. Hua, Y., Sakar, T.K.: On SVD for estimating generalized eigenvalues of singular matrix pencil in noise. IEEE Transactions on Signal Processing 39(4), 892–900 (1991)

16. Cadzow, J.: Signal enhancement a composite property mapping algorithm. IEEE Transactions on Acoustic, Speech, and Signal Processing 36(1), 49–62 (1988)
17. Blu, T., Dragotti, P.L., Vetterli, M., Marziliano, P., Coulot, L.: Sparse sampling of signal innovations. IEEE Signal Processing Magazine 25(2), 31–40 (2008)
18. Markovsky, I.: Structured low-rank approximation and its applications. Automatica J. IFAC 44(4), 891–909 (2008)
19. Albert, A.: Regression and the Moore-Penrose Pseudoinverse. Academic Press, New York (1972)
20. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
21. Hirabayashi, A.: Sampling and reconstruction of periodic piecewise polynomials using sinc kernel. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E95-A(1), 322–329 (2012)
22. Aggarwal, N., Karl, W.: Line detection in images through regularized Hough transform. IEEE Transactions on Image Processing 15(3), 582–591 (2006)
23. Shi, D., Zheng, L., Liu, J.: Advanced Hough transform using a multilayer fractional Fourier method. IEEE Transactions on Image Processing 19(6), 1558–1566 (2010)
24. Baboulaz, L., Dragotti, P.: Exact feature extraction using finite rate of innovation principles with an application to image super-resolution. IEEE Transactions on Image Processing 18(2), 281–298 (2009)
25. Hirabayashi, A., Dragotti, P.L.: E-spline sampling for precise and robust line-edge extraction. In: Proceedings of International Conference on Image Processing (ICIP 2010), Hong Kong, pp. 909–912 (2010)
26. Poh, K.K., Marziliano, P.: Compressive sampling of EEG signals with finite rate of innovation. EURASIP Journal on Advances in Signal Processing (2010)
27. Hirabayashi, A., Makido, S., Condat, L.: MAP recovery of polynomial splines from compressive samples and its applications to vehicular signals. In: Van De Ville, D., Goyal, V., Papadakis, M. (eds.) Wavelets and Sparsity XV, San Diego. Proceedings of SPIE, vol. 8858 (2013)

List of Acronyms

EEG	Electroencephalogram
ECG	Electrocardiogram
DFT	Discrete Fourier Transform
SVD	Singular Value Decomposition
DoA	Direction of Arrival
MUSIC	Multiple Signal Classifier
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Technique
LS	Least Square
SLRA	Structured Low-Rank Approximation
PSO	Particle Swarm Optimization
SNR	Signal-to-Noise Ratio
MSE	Mean Square Error
SLR	Single Lens Reflex
PSF	Point Spread Function

Chapter 8
Tensor-Based Subspace Learning for Multi-pose Face Synthesis

Xu Qiao*, Takanori Igarashi, and Yen-Wei Chen

Abstract. Facial pose synthesis is applied to generate much required information for several applications, such as public security, facial cosmetology, etc. How to synthesize facial pose images from one image accurately without spatial information is still a challenging problem. In this chapter we propose a tensor-based subspace learning method (TSL) for synthesizing human multi-pose facial images from a single two-dimensional image. In the proposed TSL method, two-dimensional multi-pose images in the database are previously organized into a tensor form and a tensor decomposition technique is applied to build projection subspaces. In synthesis processing, the input two-dimensional image is first projected into its corresponding projection subspace to get an identity vector and then the identity vector is used to generate other novel pose images. Our technique is applied on KAO-Ritsumeikan Multi-angle View, Illumination and Cosmetic Facial Database(MaVIC) and experimental results show the effectiveness of our proposed method for facial pose synthesis.

1 Introduction

Real data of natural and social sciences is often very high-dimensional. However, the underlying structure can be characterized by a small number of parameters. Reducing the dimensionality of such data is beneficial for visualizing the intrinsic structure

Xu Qiao
School of Control Science and Engineering, Shandong University, Jinan, China
e-mail: qiaoxu@sdu.edu.cn

Takanori Igarashi
Beauty Cosmetic Research Lab, Kao Corporation, Tokyo, Japan
e-mail: igarashi.takanori@kao.co.jp

Yen-Wei Chen
College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
e-mail: chen@is.ritsumei.ac.jp

* This work was mainly contributed in Ritsumeikan University when he studied for his PhD Degree.

Y.-W. Chen and L.C. Jain (eds.), *Subspace Methods for Pattern Recognition in Intelligent Environment*, Studies in Computational Intelligence 552,
DOI: 10.1007/978-3-642-54851-2_8, © Springer-Verlag Berlin Heidelberg 2014

and it is also an import pre-processing step in many statistical pattern classification problems, such as face recognition and image retrieval.

Recently, multilinear algebra was applied for analyzing the multifactor structure of image ensembles. Vasilesu and Terzopoulos have proposed a novel face representation algorithm called tensorface [1, 2]. Tensorface represents the set of face images by a multi-dimensional tensor and extends traditional PCA to tensor-based subspace learning with tensor decompositions. In this way, the multiple factors related to expression, illumination and pose can be separated from different dimensions of tensor. These multiple factors can be used for recognition and synthesis. Following we use tensor-based subspace learning for facial pose synthesis.

When we obtain a person's profile facial image, can we generate this person's frontal facial image or other poses? The method to solve this kind of problem is properly called ”facial pose synthesis” [3]. The essential idea of image synthesizing is extracting information from exist images and generating an accurate and detailed facial model. It has been an active topic in computer vision, computer graphics and related fields.

Facial pose synthesis has a number of useful applications, such as for social security and cosmetology. In social security, the facial image synthesis method can be applied to assist law enforcement. Sometimes, due to the limits of circumstances, the police take suspect photos with some feature parts, such as half of the face, invisible. It is difficult for the police to recognize the suspect without having front facial information. Facial pose synthesis techniques will help the police generate a frontal facial portrait and other poses. It can also be applied into the field of cosmetology for skin appearance[32]. If one human nature facial image is obtained, the person's cosmetic facial images under some conditions (such as mutative illuminations, mutative view-angles) will be generated by using synthesis methods.

Synthesis methods can mainly be classified into three categories: anatomy based methods, geometry based methods, and learning methods. Their advantages and limitations are analyzed and discussed following:

Anatomy based methods build face models by estimating the dynamic facial muscle contractions from a sequence of human face images [5]. It needs a lot of pre-processing, such as registration of corresponding muscle points and setting constraints of muscle contraction to every sample. Although this method is of high accuracy, it is difficult for practical applications.

Geometry based methods recover shapes from information such as shading, stereo, motion, texture, etc. For example, the shape from shading method (SFS) deals with the recovery from a gradual variation of shading in an image [6, 7]. The SFS method needs to compute the surface orientation map, such as a normal direction or gradient field from image intensity and then reconstruct the surface depth map from the orientation map. Since the SFS is highly dependant on the gradual variation of shading in the image, it is an ill-posed problem and difficult to find a unique solution without additional constraints.

Learning based methods find the related objects between pose subspaces by training samples. It is shown that linear transformations can be learned exactly from a basis set of two-dimensional prototypical views for linear object classes in [8].

Locally linear Embedding (LLE) is used to learn common hidden structures shared among different pose images to find parameters which control the pose variation [9]. A model for pose synthesis is built for a special person by training this person's images using LLE method. But this model can not be applied for other persons. A morphable three-dimensional face model (MF) has been proposed by Blanz and Vetter [10, 11]. They first built a three-dimensional face model by training a large dataset of three-dimensional face scans, which are obtained by laser scans. In the synthesis process, an input two-dimensional image is first used to estimate its three-dimensional model parameters by model fitting. The other novel pose images are obtained by projecting the three-dimensional model to two-dimensional planes. This method needs good initial values for model fitting in order to avoid the local minimum. It also suffers time-consuming problems for three-dimensional data collection and model fitting processing.

To overcome these shortcomings, we propose a tensor-based subspace learning method (TSL) for facial pose synthesis. We organize two-dimensional multi-pose images as a tensor form and apply tensor decomposition to build a projection subspace. In the synthesis process, the input two-dimensional image is projected into the corresponding projection subspace to get an identity vector. The identity vector is then used to generate other novel pose images. This is motivated by the fact that Vasilescu and Terzopoulos have noticed that the tensor decomposition method is an efficient tool for feature detection, which plays an important role in the synthesis learning method [1, 2]. The method based on tensor decomposition has also been applied to synthesize facial expressions [12], in which the authors only need a frontal facial pose and do not consider the differences in the same person's facial contour in different poses. Compared with the morphable three-dimensional face model, our proposed method constructs a statistical model by training two-dimensional multi-pose images instead of three-dimensional scans and doesn't need any model fitting processing. So it is easier to implement and can be used for on-line processing. Experimental results on Kao-Ritsumeikan Multi-angle View, Illumination and Cosmetic Facial Database(MaVIC) [13] show our proposed method is effective for facial pose synthesis.

2 Tensor and Multilinear Algebra Foundations

2.1 Definitions and Preliminaries

2.1.1 Tensor Definitions

Scalers are denoted by italic-shape letters, i.e. $(a,b,...)$ or $(A,B,...)$. Bold lower case letters, i.e. $(\mathbf{a},\mathbf{b},...)$, are used to represent vectors. Matrices are denoted by bold upper case letters, i.e. $(\mathbf{A},\mathbf{B},...)$; and higher-order tensors (more than third order tensor) are denoted by calligraphic upper case letters, i.e. $(\mathscr{A},\mathscr{B},...)$.

A tensor is a multidimensional array. The order of a tensor in the number of dimensions, as known as ways or modes. An N-th order tensor $\mathscr{A}$ is defined as a multi-array with N indices, where $\mathscr{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and $\mathbb{R}$ is the real manifold.

Elements of the tensor $\mathscr{A}$ are denoted as $a_{i_1\dots i_n\dots i_N}$, where $1 \leqslant i_n \leqslant I_n$. The space of the N-th order tensor is comprised by the N mode subspaces. From the perspective of $\mathscr{A}$, scalars, vectors and matrices can be seen as zeroth-order, first order and second order tensors, respectively.

The ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i,j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i,j,k) of a 3rd-order tensor $\mathscr{X}$ is denoted by x_{ijk}. Indices typically range from 1 to their capital version, e.g., $i = 1,\dots,I$. The nth element in a sequence is denoted by a superscript in parentheses, e.g., $\mathbf{A}^n$ denotes the nth matrix in a sequence.

Subarrays are formed when a subset of the indices is fixed. For matrices, these are the rows and columns. A colon is used to indicate all elements of a mode. Thus, the jth column of $\mathbf{A}$ is denoted by $\mathbf{a}_{:j}$, and the ith row of $\mathbf{A}$ is denoted by $\mathbf{a}_{i:}$.

Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixed every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. 3rd-order tensors have column, row, and tube fibers, denoted as $\mathbf{x}_{:jk}$, $\mathbf{x}_{i:k}$, and $\mathbf{x}_{ij:}$, respectively. Fibers of a 3rd-order tensors are shown in Fig. 1.

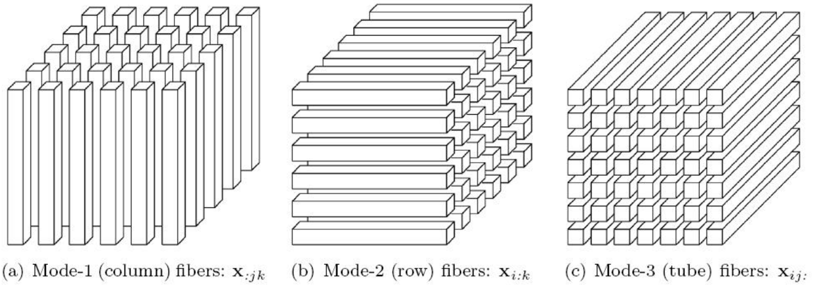

(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Fig. 1 Fibers of a 3rd-order tensor

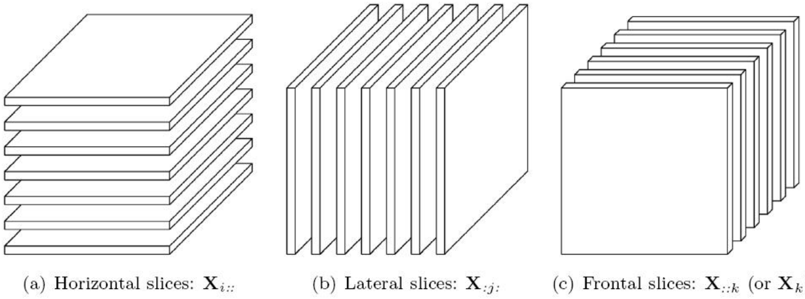
Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. Fig. 2 shows the horizontal, lateral, and frontal slides of a 3rd-order tensor $\mathscr{X}$, denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$, and $\mathbf{X}_{::k}$, respectively.

2.1.2 Tensor Norm and Rank

The norm of a tensor $\mathscr{X} \in \mathbb{R}^{I_1\times I_2\times\dots\times I_N}$ is the square root of the sum of the squares of all its elements, i.e.,

$$\| \mathscr{X} \| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} x^2_{i_1 i_2\cdots i_N}}. \tag{1}$$

This is analogous to the matrix Frobenius norm, which is denoted $\| \mathbf{A} \|$ for a matrix $\mathbf{A}$.



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

Fig. 2 Slices of a 3rd-order tensor

The inner product of two same-sized tensors $\mathscr{X},\mathscr{Y} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is the sum of the products of their entries, i.e.,

$$< \mathscr{X},\mathscr{Y} >= \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} x_{i_1 i_2\cdots i_N} y_{i_1 i_2\cdots i_N}. \qquad (2)$$

It follows immediately that $< \mathscr{X},\mathscr{X} >=\| \mathscr{X} \|^2$.

A Nth-order tensor $\mathscr{X} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is rank one if it can be written as the outer product of N vectors, i.e.,

$$\mathscr{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}. \qquad (3)$$

The symbol”$\circ$” represents the vector outer product. This means that each element of the tensor is the product of the corresponding vector elements: $x_{i_1 i_2\cdots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$, for all $1 \leq i_n \leq I_N$.

Fig. 3 illustrates $\mathscr{X} = \mathbf{a}\circ\mathbf{b}\circ\mathbf{c}$, a third-order rank-one tensor.

2.1.3 Symmetry and Diagonal Tensors

A tensor is called cubical if every mode is the same size, i.e., $\mathscr{X} \in \mathbb{R}^{I\times I\times I\times\ldots\times I}$. A cubical tensor is called supersymmetric if its elements remain constant under any permutation of the indices. For instance, a 3th-order $\mathscr{X} \in \mathbb{R}^{I\times I\times I}$ is supersymmetric if $x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji}$, for all $i, j, k = 1,\cdots,I$.

Tensor can be partial symmetric in two or more modes as well. For example, a 3rd-order $\mathscr{X} \in \mathbb{R}^{I\times I\times k}$ is symmetric in modes one and two if all its frontal slices are symmetric, i.e., $\mathbf{X}_k = \mathbf{X}_k^T$, for all $k = 1,\cdots,K$.

A tensor $\mathscr{X} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is diagonal if $x_{i_1 i_2\cdots i_N} \neq 0$ only if $i_1 = i_2 = \cdots = i_N$.

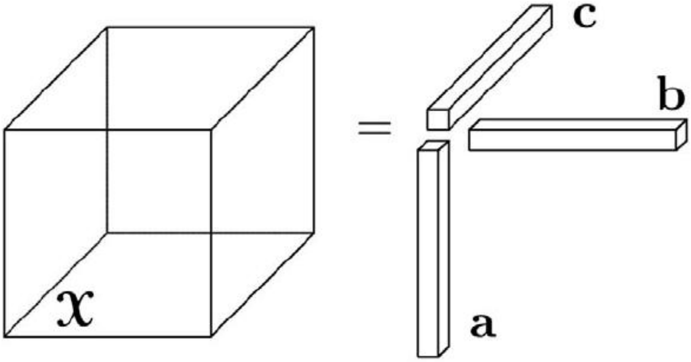


Fig. 3 Rank-one 3rd-order tensor, $\mathscr{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$

2.1.4 Matricization of Tensors

Matricization, also known as unfolding or flattening, is the process of reordering the elements of an Nth-order array into a matrix. For example. a $2 \times 3 \times 4$ tensor can be arranged as a 6×4 matrix or a 3×8 matrix, and so on. A more general treatment of matricization can be found in [20]. The mode-n matricization of a tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by and arranges the mode-n fibers to be the columns of the resulting matrix. Tensor elements $(i_1, i_2, \cdots, i_N)$ maps to matrix element(i_n, j), where

$$j = 1 + \sum_{k=1,k\neq n}^{N} (i_k - 1)J_k, with\, J_k = \prod_{m=1,m\neq n}^{k-1} I_m. \tag{4}$$

Fig. 4 illustrates a example of matricization for a 3rd-order tensor.

2.1.5 Tensor Multiplication: The n-Mode Product

Tensor can be multiplied together, though obviously the notation and symbols for this are much more complex than for matrices. Here we consider only the tensor n-mode product, i.e., multiplying a tensor by a matrix (or a vector) in mode n.

The n-mode (matrix) product of a tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathscr{X} \times_n \mathbf{U}$ and is of size $I_1 \times I_2 \times \ldots \times I_{n-1} \times J \times I_{n+1} \ldots \times I_N$. Elementwise, we have

$$(\mathscr{X} \times_n \mathbf{U})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_N} x_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

Each mode-n fiber is multiplied by the matrix $\mathbf{U}$. The idea can also be expressed in terms of unfolder tensors:

$$\mathscr{Y} = \mathscr{X} \times_n \mathbf{U} \Leftrightarrow Y_n = \mathbf{U} X_n.$$

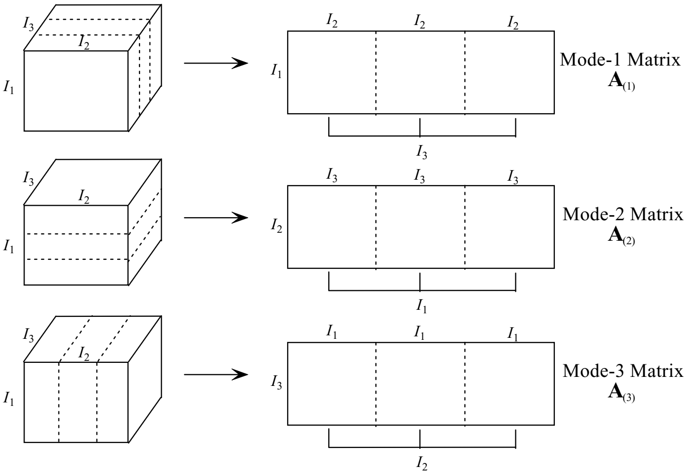


Fig. 4 Matricization of a 3rd-order tensor

A few facts regarding n-mode matrix products are in order. For distinct modes in a series of multiplications, the order of the multiplication is irrelevant, i.e.,

$$\mathscr{X} \times_m A \times_n B = \mathscr{X} \times_n B \times_m A, m \neq n.$$

If the modes are the same, then

$$\mathscr{X} \times_n A \times_n B = \mathscr{X} \times_n (BA), m = n.$$

The n-mode (vector) product of a tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathscr{X} \times_n \mathbf{v}$. The result is of order $N-1$, i.e., the size is $I_1 \times I_2 \times \ldots \times I_{n-1} \times I_{n+1} \ldots \times I_N$. Elementwise,

$$(\mathscr{X} \times_n \mathbf{v})_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_N} x_{i_1 i_2 \cdots i_N} v_{i_n}.$$

The idea is to compute the inner product of each mode-n fiber with the vector.

2.1.6 Matrix Product

We briefly introduce some important matrix products.

The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted by $\mathbf{A} \bigotimes \mathbf{B}$. The result is a matrix of size $(IK) \times (JL)$ and defined by

$$\mathbf{A} \bigotimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{pmatrix}$$

Also it can be written as the form of multiplication of column vectors:

$$\mathbf{A} \bigotimes \mathbf{B} = (\mathbf{a}_1 \bigotimes \mathbf{b}_1 \;\; \mathbf{a}_1 \bigotimes \mathbf{b}_2 \;\; \mathbf{a}_1 \bigotimes \mathbf{b}_3 \;\; \cdots \;\; \mathbf{a}_J \bigotimes \mathbf{b}_{L-1} \;\; \mathbf{a}_J \bigotimes \mathbf{b}_L).$$

The Khatri-Rao product is the ”matching columnwise” Kronecker product. Given matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$, their Kratri-Rao product is denoted by $\mathbf{A} \bigodot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ defined by

$$\mathbf{A} \bigodot \mathbf{B} = (\mathbf{a}_1 \bigotimes \mathbf{b}_1 \;\; \mathbf{a}_2 \bigotimes \mathbf{b}_2 \;\; \cdots \;\; \mathbf{a}_K \bigotimes \mathbf{b}_K).$$

If **a** and **b** are vectors, then the Khatri-Rao and Kronecker products are identical, i.e., $\mathbf{A} \bigodot \mathbf{B} = \mathbf{a} \bigotimes \mathbf{b}$.

The Hadamard product is the elementwise matrix product. Given matrices **A** and **B**, both of size $I \times J$, their Hadamard product is denoted by $\mathbf{A} * \mathbf{B}$. The result is also of size $I \times J$ and defined by

$$\mathbf{A} * \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{IJ} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{IJ} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{pmatrix}.$$

2.2 Tensor Decomposition

In linear algebra, Singular Value Decomposition (SVD) is an important factorization of a rectangular real and complex matrix, with several applications in signal processing and statistics. SVD computes the low-rank approximation of a set of one-dimensional vectors. This can be generalized to a two dimensional Singular Value Decomposition (2DSVD) to do low-rank approximation of a set of matrices such as a set of images. Higher Order Singular Value Decomposition (HOSVD) is a generalization of SVD for high dimensional tensor [16]. In the case of a three-dimensional tensor executing HOSVD in two dimensions gives the same result as 2DSVD.

N-dimensional principal component analysis(ND-PCA) [17] is based on HOSVD and the high-dimensional data is treated as a high-order tensor. This method is not only applied in data compression but also applied in multi-facial recognition(tensorface method,[1, 2]). In our previous work[18], we proposed a framework called generalized N-dimensional principal component analysis (GND-PCA) based on HOSVD and applied it to statistical appearance modeling of medical volume images. As the facial images with multiple modes can be considered as a high-dimensional tensor, we also applied it to statistical appearance modeling for facial images with multiple modes including different people, different viewpoint and different illumination[19]. Recently, tensor-based active appearance model has also proposed for multiple modes facial image modeling [23]. We propose a tensor-based subspace learning method (TSL) for novel facial pose synthesis.

2.2.1 Tucker Decomposition

The Tucker decomposition was first introduced by Tucker in 1963 [27] and refined in subsequent articles by Levin [21] and Tucker [33, 14]. The Tucker decomposition goes by several names, such as three-mode factor analysis(3MFA/Tucker3) [14],

N-mode PCA [28], higher-order singular value decomposition (HOSVD) [43], and N-mode SVD [1].

The Tucker decomposition is a form of higher-order PCA. It decomposes a tensor into a core tensor multiplied by a matrix along each mode. Thus, in the 3rd-order case where $\mathscr{X} \in \mathbb{R}^{I\times J\times K}$, we have

$$\mathscr{X} \approx \mathscr{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r = [\![\mathscr{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \qquad (5)$$

Here, $\mathbf{A} \in \mathbb{R}^{I\times P}$, $\mathbf{B} \in \mathbb{R}^{J\times Q}$, and $\mathbf{C} \in \mathbb{R}^{K\times R}$ are the factor matrices (which are usually orthogonal) and can be thought of as the principal components in each mode. The tensor $\mathscr{G} \in \mathbb{R}^{P\times Q\times R}$ is called the core tensor and its entries show the level of interaction between the different components.

Elementwise, the Tucker decomposition is

$$x_{ijk} \approx \textstyle\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}a_{ip}b_{jq}c_{kr}, for\ i = 1,\cdots, I, j = 1,\cdots,J,\ k = 1,\cdots,K.$$

Here P, Q, and R are the number of components in the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. If P, Q, R are smaller than I, J, K, the core tensor $\mathscr{G}$ can be thought of as a compressed version of $\mathscr{X}$. The Tucker decomposition is illustrated in Fig. 5.

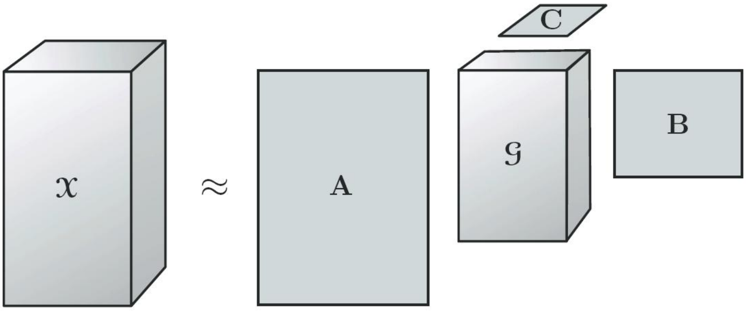


Fig. 5 Tucker decomposition of a 3rd-order tensor

The metricized forms of Eq. 5 are

$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}(\mathbf{B}\mathbf{C})^T,\ \mathbf{X}_{(2)} = \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C}\mathbf{A})^T,\ \mathbf{X}_{(3)} = \mathbf{C}\mathbf{G}_{(3)}(\mathbf{A}\mathbf{B})^T.$$

Let $\mathscr{X}$ be an Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$. Then the n-rank of $\mathscr{X}$, denoted $rank_n(\mathscr{X})$, is the column rank of $\mathbf{X}_{(n)}$. In other words, the n-rank is the dimension of the vector spanned by the mode-n fibers. If we let $R = rank_n(\mathscr{X})$ for $n = 1,2,\cdots,N$, then we can say that $\mathscr{X}$ is a rank-$(R_1,R_2,\cdots,R_N)$ tensor. Trivially, $R_n \leq I_n$ for all $n = 1,\cdots,N$.

2.2.2 CANDECOMP/PARAFAC Decomposition

Another important tensor decomposition is CANDECOMP/PARAFAC decomposition, which is shorthanded as CP decomposition. In 1927, Hitchcock [22, 24] proposed the idea of polyadic form of a tensor. The concept finally became popular after the introduction in 1970 to the psychometrics community, in the form of CANDECOMP (canonical decomposition) by Carroll and Chang [34] and PARAFAC (parallel factors) by Harshman [35].

The CP decomposition factorizes a tensor into a sum of component rank-one tensors. For example, given a 3rd-order tensor $\mathscr{X} \in \mathbb{R}^{I\times J\times K}$, we wish to write it as

$$\mathscr{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{6}$$

Where R is a positive integer and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, 2, \cdots, R$.

Elementwise, Eq. 6 is written as

$$x_{ijk} \approx \textstyle\sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}, for\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K.$$

This is illustrated in Fig 6.

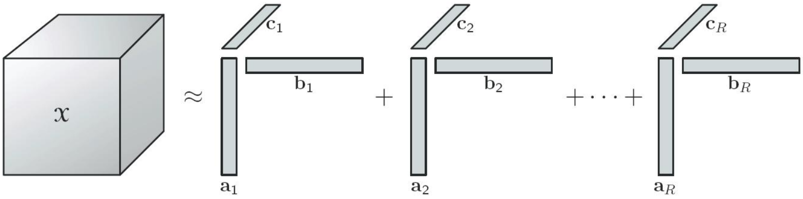


Fig. 6 CP decomposition of a 3rd-order tensor

The factor matrices refer to the combination of the vectors from the rank-one components, i.e., $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_1]$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. Using these definitions, Eq. 6 may be written in matricized form:

$$\mathbf{X}_{(1)} = \mathbf{A}(\mathbf{B} \odot \mathbf{C})^T,\ \mathbf{X}_{(2)} = \mathbf{B}(\mathbf{C} \odot \mathbf{A})^T,\ \mathbf{X}_{(3)} = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T.$$

Recall that $\odot$ denotes the Khatri-Rao product.

In fact, CP can be viewed as a special case of Tucker where the core tensor is superdiagonal and $P = Q = R$ in Eq. 5.

2.2.3 Other Decompositions

There are a number of other tensor decompositions related to Tucker and CP. Most of these decompositions which originated in the psychometrics have only recently become more widely known in other fields such as chemometrics and social network analysis. The introduction of these decompositions is not included in depth in this chapter.

3 Tensor-Based Subspace Learning Algorithm

In this section, TSL method is introduced step by step. Here suppose we have I training persons and J types of pose. Each 2D image is unfolded into a vector. The i^{th} person's j^{th} pose image vector is noted as $\mathbf{p}_{ij}$, $1 \leq i \leq I$, $1 \leq j \leq J$. Since the images' size is $M \times N$, $\mathbf{p}_{ij}$ is a MN - dimensional vector.

3.1 Image Representation

Both shape and texture (intensity) provide useful information for characterizing facial appearance. While dealing with two-dimensional images directly, variations of shape and texture interfere with each other [29, 30]. We need to separate shape information and texture information.

L physical landmarks were extracted as shape points manually, e.g., the tip of noses, the eye corner and less prominent points on the check. Each facial shape vector is represented by its landmarks' coordinates (x, y) as

$$\mathbf{s}_{ij} = (x_1, y_1, x_2, y_2, \cdots, x_L, y_L), 1 \leq i \leq I, 1 \leq j \leq J. \tag{7}$$

In order to represent the texture information, we need to normalize the facial shape previously. In our research, each facial shape is normalized to a mean shape of samples based on labeled landmarks. The normalized image of $\mathbf{p}_{ij}$ is represented by a texture vector $\widehat{\mathbf{p}}_{ij}$ which is used to construct texture subspace.

All $\mathbf{s}_{ij}$ is organized into a shape tensor $\mathscr{S}$ with the size of $I \times J \times 2L$. All $\widehat{\mathbf{P}}_{IJ}$ are arrayed into a texture tensor $\mathscr{T}$ with the size of $I \times J \times (M \cdot N)$.

3.2 Tensor Subspace Building

This step is to build a texture projection space and a shape projection space which can be considered as a filter to convert the input 2D facial image (both texture and shape) into an identify texture vector and an identify shape vector.

By performing tensor decomposition on $\mathscr{S}$ and $\mathscr{T}$, we have

$$\mathscr{S} = \mathscr{C}_S \times_1 \mathbf{U}_{person\_s} \times_2 \mathbf{U}_{pose\_s} \times_3 \mathbf{U}_{parameter}, \tag{8}$$

$$\mathscr{T} = \mathscr{C}_T \times_1 \mathbf{U}_{person} \times_2 \mathbf{U}_{pose} \times_3 \mathbf{U}_{pixel}. \tag{9}$$

The illustration of tensor decomposition is shown in Fig. 7. Here $\mathscr{C}_S$ and $\mathscr{C}_T$ are core tensors for shape and texture respectively. $\mathbf{U}_{person\_s}$, $\mathbf{U}_{pose\_s}$ and $\mathbf{U}_{parameter}$

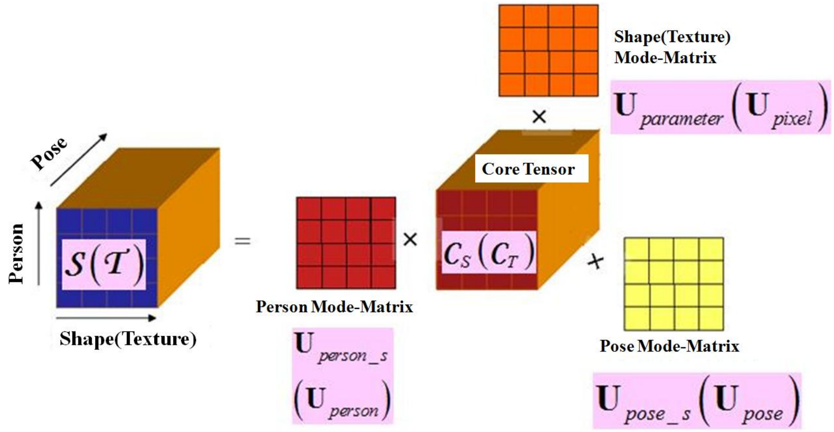


Fig. 7 Tensor decomposition on shape (texture) tensor

represent the person shape subspace, pose shape subspace, shape landmark point subspace of shape tensor respectively. Similarly, $\mathbf{U}_{person}$, $\mathbf{U}_{pose}$ and $\mathbf{U}_{pixel}$ represent the person texture subspace, pose texture subspace and texture subspace of texture tensor respectively. These matrices are all orthogonal. Each row vector in each subspace represents a specific vector in this mode. For example, in the person subspace of texture tensor, $\mathbf{U}_{person} = (\mathbf{u}_{p\_1}^T, \cdots, \mathbf{u}_{p\_i}^T, \cdots, \mathbf{u}_{p\_I}^T)$, $\mathbf{u}_{p\_i}^T$ represents the identity vector of the i^{th} person for texture. Similarly, $\mathbf{U}_{person\_s} = (\mathbf{u}_{ps\_1}^T, \cdots, \mathbf{u}_{ps\_i}^T, \cdots, \mathbf{u}_{ps\_I}^T)$, $\mathbf{u}_{ps\_i}^T$ represents the identity vector i^{th} of the person for shape. The dimension of identity vectors depends on the number of training samples.

We can build the project subspaces like Eq. 10 and Eq. 11:

$$\begin{aligned} \mathcal{S} &= (\mathcal{C}_S \times_2 \mathbf{U}_{pose\_s} \times_3 \mathbf{U}_{parameter}) \times_1 \mathbf{U}_{person\_s} \\ &= \mathcal{A}_{Shape} \times_1 \mathbf{U}_{person\_s}, \end{aligned} \tag{10}$$

$$\begin{aligned} \mathcal{T} &= (\mathcal{C}_T \times_2 \mathbf{U}_{pose} \times_3 \mathbf{U}_{pixel}) \times_1 \mathbf{U}_{person} \\ &= \mathcal{A}_{Texture} \times_1 \mathbf{U}_{person}. \end{aligned} \tag{11}$$

$\mathcal{A}_{Shape}$ is an $I \times J \times 2L$ tensor for shape projection and $\mathcal{A}_{Texture}$ is an $I \times J \times (M \cdot N)$ tensor for texture projection. The illustration of project subspace building for shape is shown in Fig. 8. $\mathcal{A}_{Shape}$ is expressed as an orderly array of matrices on pose direction as in the following equation:

$$\mathcal{A}_{Shape} \overset{\text{def}}{=} [\mathbf{A}_{pose1}^{Shape}, \mathbf{A}_{pose2}^{Shape}, \cdots, \mathbf{A}_{poseJ}^{Shape}]. \tag{12}$$

Here $\mathbf{A}_{posej}^{Shape}$ is a $I \times 2L$ matrix and is j^{th} slice of $\mathcal{A}_{Shape}$ along the direction of pose. Each $\mathbf{A}_{posej}^{Shape}$ can be considered as the j^{th} projection subspace for the pose and

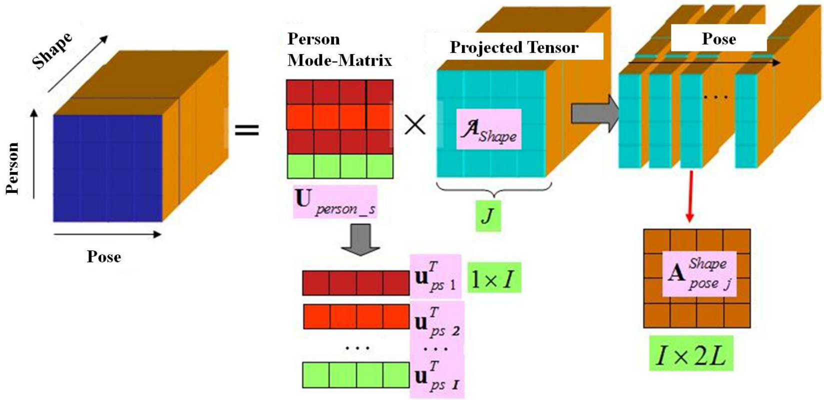


Fig. 8 Projection subspace building for shape

each shape vector is calculated by

$$\mathbf{s}_{ij} = (\mathbf{A}^{Shape}_{posej})^T \mathbf{u}_{ps\_i} \tag{13}$$

with $\mathbf{u}_{ps\_i}$ being the coefficients of the i^{th} person for shape.

Similarly, $\mathbf{A}^{Texture}_{posej}$ is j^{th} pose matrix of $\mathscr{A}_{Texture}$ with the size of $I \times (M \cdot N)$ on texture:

$$\mathscr{A}_{Texture} \overset{\text{def}}{=} [\mathbf{A}^{Texture}_{pose1}, \mathbf{A}^{Texture}_{pose2}, \cdots, \mathbf{A}^{Texture}_{poseJ}]. \tag{14}$$

For each morphed image vector

$$\widehat{\mathbf{p}}_{ij} = (\mathbf{A}^{Texture}_{posej})^T \mathbf{u}_{p\_i} \tag{15}$$

with $\mathbf{u}_{p\_i}$ being the coefficients of the i^{th} person for texture.

3.3 Synthesis Procedure

In this step, we use a 2D pose image as an input test image. An overall illustration of this step is shown in Fig. 9.

$\mathbf{s}_{T,k}$ and $\widehat{\mathbf{p}}_{T,k}$ represent the shape information and the shape normalized texture information of a single input test image $\mathbf{p}_{T,k}$, $k \in [1,J]$.

From Eq. 15, the texture identity vector of the testing image is calculated by

$$\mathbf{u}_{pT} = (\mathbf{A}^{Texture}_{posek})^{-T} \widehat{\mathbf{p}}_{T,k} \tag{16}$$

Then we project $\mathbf{u}_{pT}$ into another matrix on the pose direction to get the synthesis texture image vector with

$$\widehat{\mathbf{p}}_{T,j} \approx (\mathbf{A}^{Texture}_{posej})^T \mathbf{u}_{pT}, 1 \leq j \leq J, j \neq k. \tag{17}$$

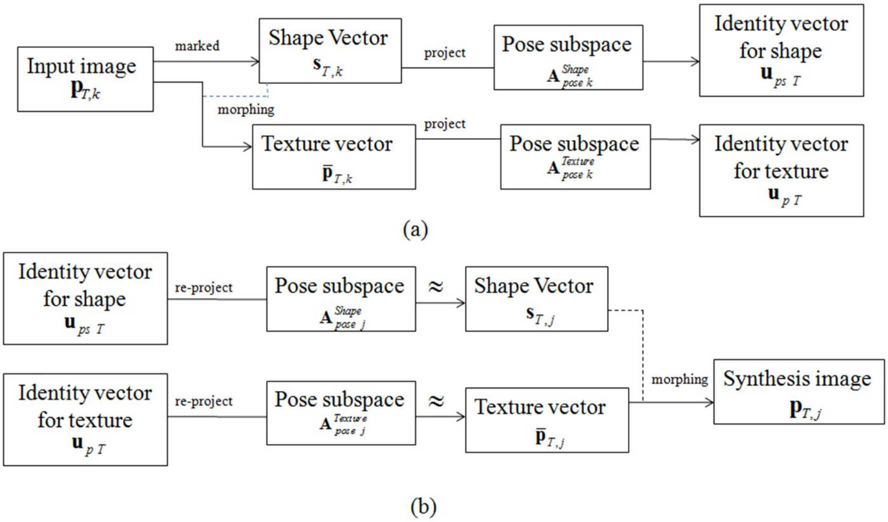


Fig. 9 Novel pose image construction. (a) Identity vector calculation. (b) Construction of a novel pose image.

Similarly, by Eq. 13, we can get the shape identity vector of the testing image by

$$\mathbf{u}_{psT} = (\mathbf{A}^{Shape}_{posek})^{-T}\mathbf{s}_{T,k} \qquad (18)$$

$$\mathbf{s}_{T,j} \approx (\mathbf{A}^{Shape}_{posej})^{T}\mathbf{u}_{psT}, 1 \leq j \leq J, j \neq k. \qquad (19)$$

Finally, we apply a reverse-morphable procedure to get the j^{th} pose image of testing person with the benefits of $\widehat{\mathbf{p}}_{T,j}$ and $\mathbf{s}_{T,j}$ calculated above.

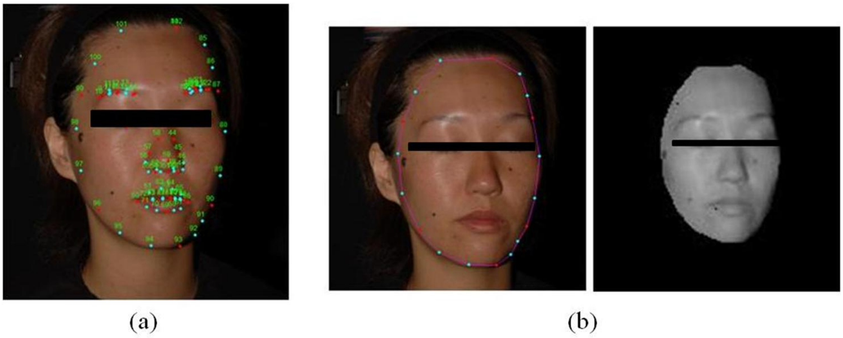
Since just given 2D images of facial poses instead of 3D scans, we don't know the spatial information and can not build the linear corresponding between the different poses of a person directly. The assumption of our method is that we consider the facial images among different persons are as much similar as possible and one pose image of a person can be represented as a linear combination of other persons'images of same pose type. With the benefit of subspace learning methods, the image can be represented as a linear combination of bases. For different pose, there exists a set of bases. By arranging the facial pose images orderly, we can get the bases for different pose by with the different pose images of one person have the "common" parameters, which we note as the identity vector. Synthesis in our method is like a generalization processing: when the object for synthesizing is much more similar like one training samples, the synthesis results will be much better. In order to increasing the synthesis results, we can apply the training samples as many as possible.

4 Experiments and Results

4.1 Data

In our experiments, we mainly deal with the image ensembles, which are formalized with our constructed database, *multi-angle view, illumination and cosmetic facial image database* (MaVIC). MaVIC is the only one and useful database for appearance studies. We chose 532 images of 76 persons in MaVIC2 with 7 poses (0°, ±15°, ±30°, ±45°) under a fixed illumination. We used 72 persons for training and left the other 4 for testing. The size of each image is 110 by 100 pixels. We manually located 103 landmark points on all of the available images for morphing and removed the background, including variable components (accessories, facial hairs and so on). The processing example is illustrated in Fig. 10.

The samples' eyes are covered by black bars in order to protect the personal privacy.


(a)

(b)

Fig. 10 (a) Extraction of landmarks. (b) Procedure of removing background.

4.2 Image Deformation

With the help of landmarks, we morphed each image using a piece-wise affine transform [31]. Each pose image is normalized to a mean shape of the corresponding poses. Some normalized examples are shown in Fig. 11.

4.3 Data Compression

The 2D shape-normalized image is unfolded into a vector with the dimension of 11000. As the dimension of the vectors is too huge to compute the tensor decomposition, we first reduce the dimension using principal component analysis method. Fig. 12 shows cumulative cover rate. The reconstructed images using 1000, 1500, 2000, 3000 and 4000 components are shown in Fig. 13, respectively. It can be seen that details in face appearance of original images can be reconstructed clearly using

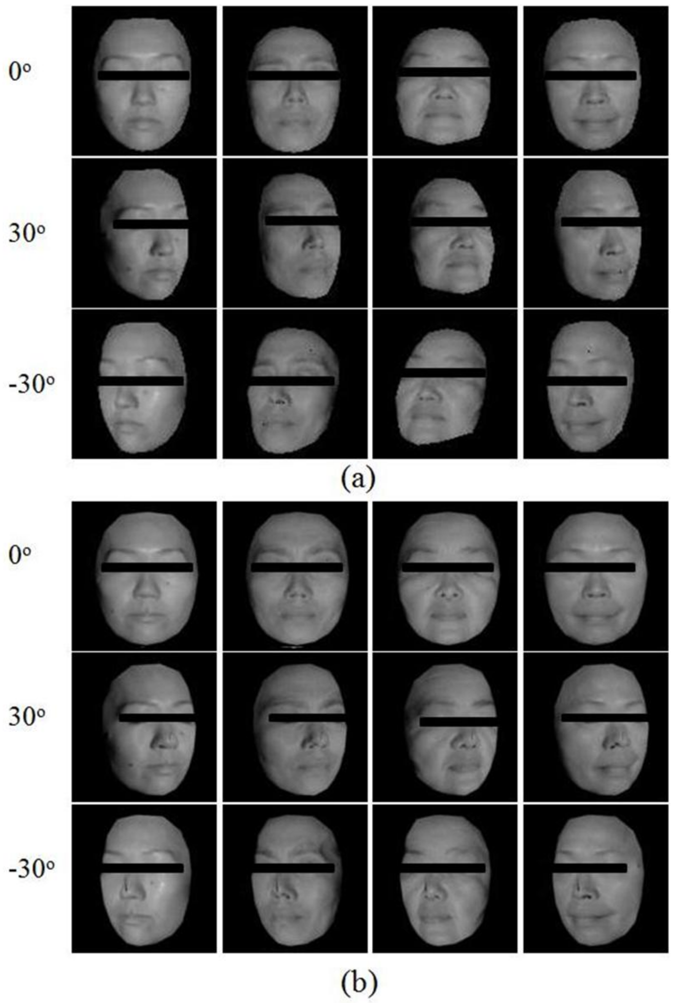


Fig. 11 Examples of shape normalization. (a) Original shape images. (b) Shape normalized images.

4000 components for reconstruction. We chose the top leading 4000 components to represent the texture vector. As shown in Fig. 12, the top leading 4000 components can keep more than 99.8% information. Then, the size of texture tensor $\mathscr{T}$ is $72 \times 7 \times 4000$. We also form the shape tensor with the size of $\mathscr{S}$ is $72 \times 7 \times 206$ as we use 103 (x, y)-coordinate landmarks.

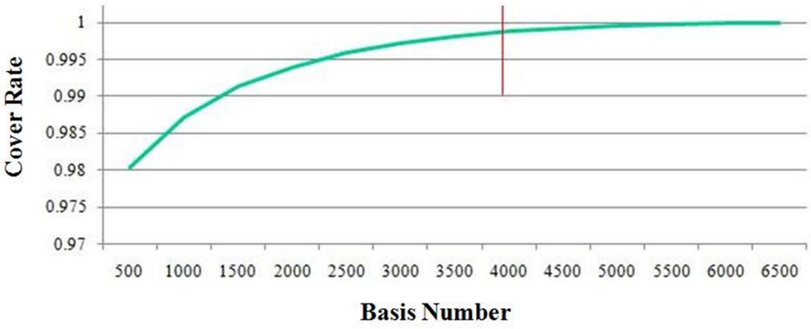


Fig. 12 Cover rate of eigenvalues while increasing number of choosing basis for reconstruction.

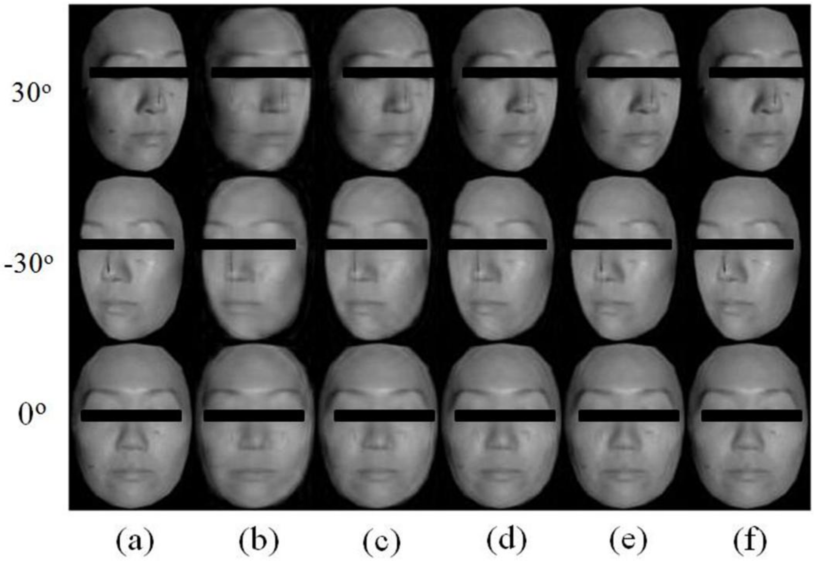


Fig. 13 Example of reconstruction results. (a) Original images. (b) Reconstruction results using 1000 basis. (c) 1500 basis. (d) 2000 basis. (e) 3000 basis. (f) 4000 basis. Red circles mark tiny features of the sample.

4.4 Synthesis Result and Evaluation

In this section, we show the synthesis results using TSL method. Fig. 14 and Fig. 15 show synthesis results of two testing samples using the TSL method. In both figures, it shows synthesis images of a pose type in each row. (b) ∼(h) exhibits the synthesis results by inputting a single image of a different pose.

In order to estimate the quality of synthesized results, we used the normalized correlation (NC) as a measurement. NC of two image data, the original $\mathbf{p}_{Ori}$ and the

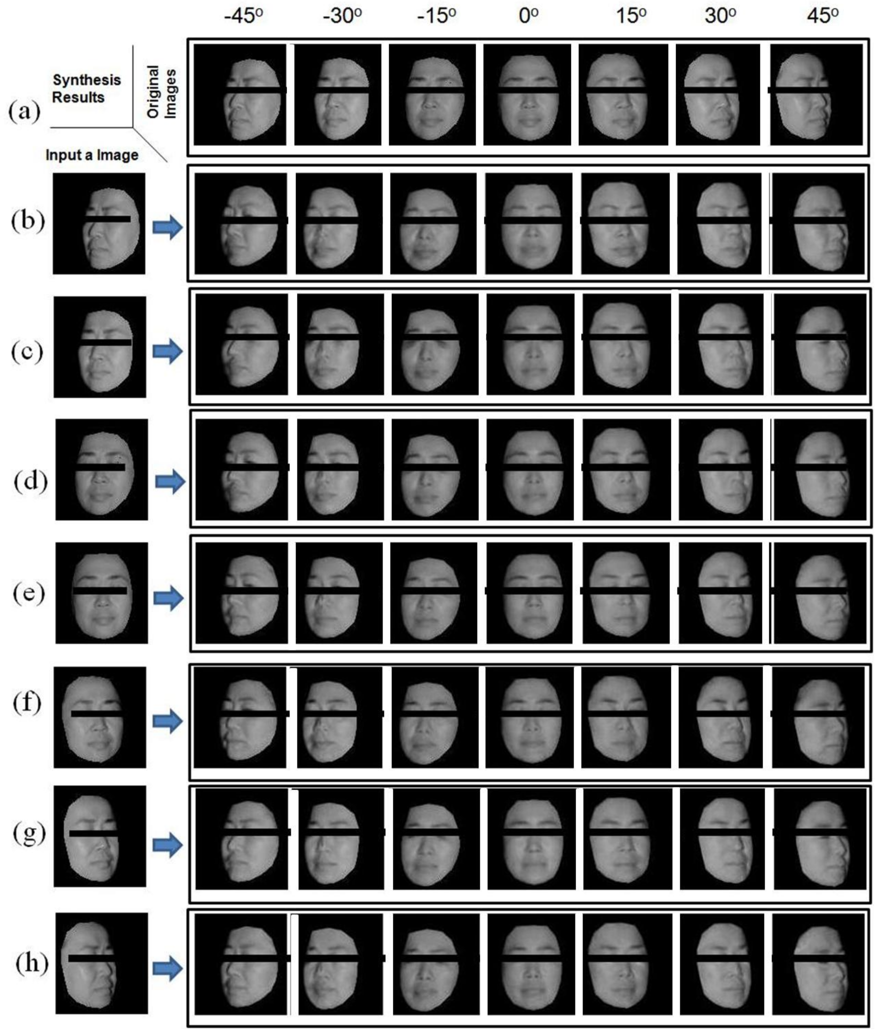


Fig. 14 Synthesis images of one test sample. (a) Original images of test sample. Synthesized images by inputting a single image of (b) -45° pose. (c) -30° pose. (d) -15° pose. (e) 0° pose. (f) 15° pose. (g) 30° pose image. (h) 45° pose.

synthesized $\mathbf{p}_{Rec}$, is defined as

$$NC \overset{\text{def}}{=} \frac{\langle \mathbf{p}_{Ori}, \mathbf{p}_{Rec} \rangle}{\sqrt{\langle \mathbf{p}_{Ori}, \mathbf{p}_{Ori} \rangle} \cdot \sqrt{\langle \mathbf{p}_{Rec}, \mathbf{p}_{Rec} \rangle}}. \tag{20}$$

$\langle \ , \ \rangle$ is represented for an inner product calculation of two vectors. The more similar the two images are, the larger the value of NC is. We only calculated the NC

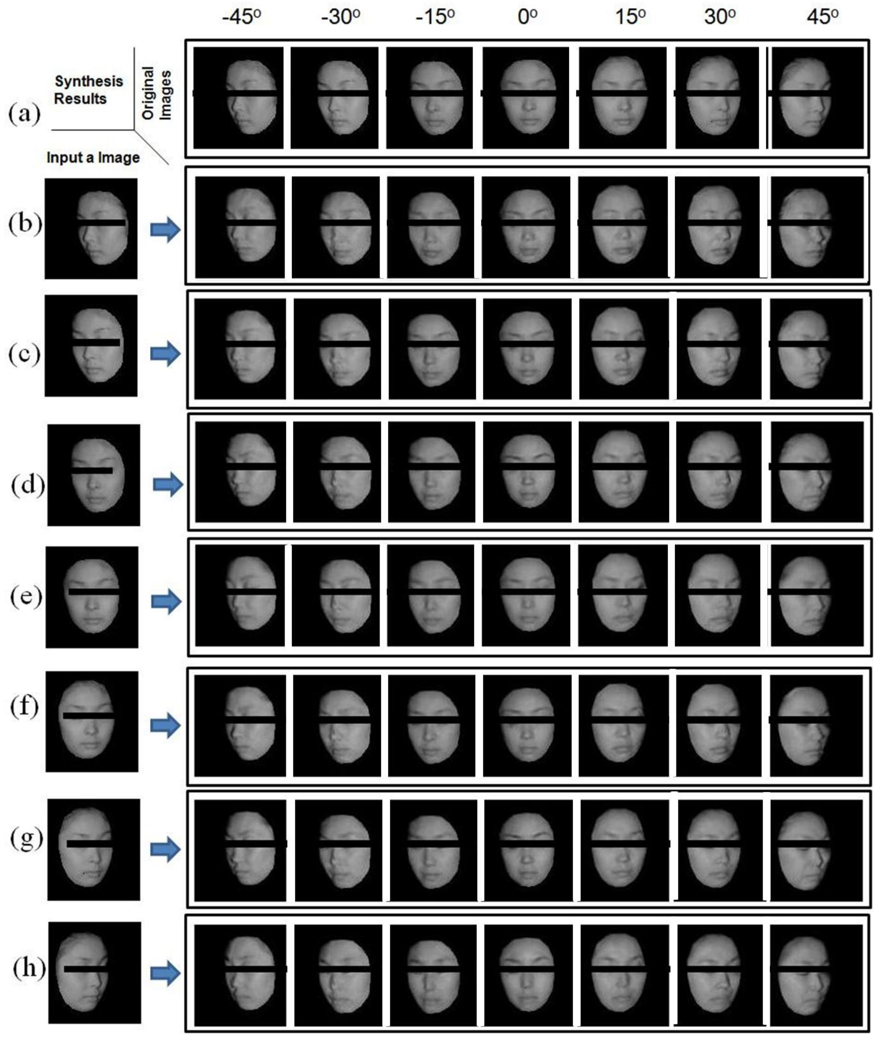
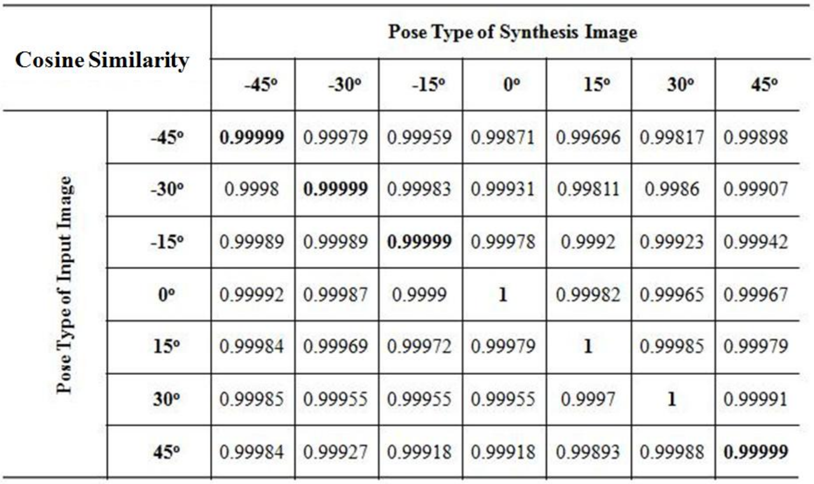


Fig. 15 Synthesis images of another test sample. (a) Original images of test sample. Synthesized images by inputting a single image of (b) −45° pose. (c) −30° pose. (d) −15° pose. (e) 0° pose. (f) 15° pose. (g) 30° pose image. (h) 45° pose.

for synthesized shape and synthesized normalized texture image in order to ignore the error caused by shape normalization.

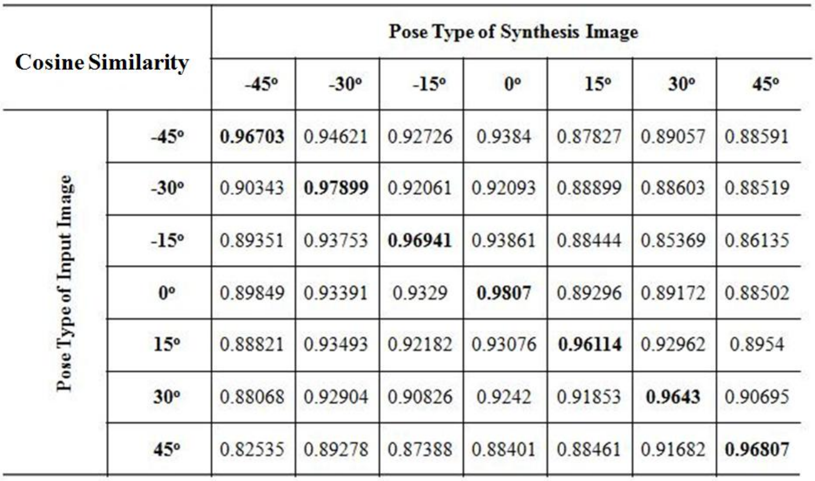
Table. 1 shows the averaged NC of shape synthesizing. From Table. 1, it can be seen that the values of NC for shape synthesizing are larger than 0.9999 and that means the shape is nearly perfectly synthesized even with larger pose difference. So we focus our discussion on the quality of texture synthesizing.

Table 1 Averaged normalized correlation of shape for all the test samples



Cosine Similarity		Pose Type of Synthesis Image						
		-45°	-30°	-15°	0°	15°	30°	45°
Pose Type of Input Image	-45°	**0.99999**	0.99979	0.99959	0.99871	0.99696	0.99817	0.99898
	-30°	0.9998	**0.99999**	0.99983	0.99931	0.99811	0.9986	0.99907
	-15°	0.99989	0.99989	**0.99999**	0.99978	0.9992	0.99923	0.99942
	0°	0.99992	0.99987	0.9999	**1**	0.99982	0.99965	0.99967
	15°	0.99984	0.99969	0.99972	0.99979	**1**	0.99985	0.99979
	30°	0.99985	0.99955	0.99955	0.99955	0.9997	**1**	0.99991
	45°	0.99984	0.99927	0.99918	0.99918	0.99893	0.99988	**0.99999**

Table 2 Averaged normalized correlation of texture for all the test samples



Cosine Similarity		Pose Type of Synthesis Image						
		-45°	-30°	-15°	0°	15°	30°	45°
Pose Type of Input Image	-45°	**0.96703**	0.94621	0.92726	0.9384	0.87827	0.89057	0.88591
	-30°	0.90343	**0.97899**	0.92061	0.92093	0.88899	0.88603	0.88519
	-15°	0.89351	0.93753	**0.96941**	0.93861	0.88444	0.85369	0.86135
	0°	0.89849	0.93391	0.9329	**0.9807**	0.89296	0.89172	0.88502
	15°	0.88821	0.93493	0.92182	0.93076	**0.96114**	0.92962	0.8954
	30°	0.88068	0.92904	0.90826	0.9242	0.91853	**0.9643**	0.90695
	45°	0.82535	0.89278	0.87388	0.88401	0.88461	0.91682	**0.96807**

Pose difference between the input image and synthesis image is noted as PD. For example, for both synthesized pose 0° from a 30° input image and synthesized pose $-15°$ from a $-45°$ input image, PD is 30°.

Table. 2 shows the averaged NC of texture synthesizing. It can be seen that the synthesizing accuracy for the same pose ($PD = 0$) is larger than 0.9600 which can be considered as generalized accuracy and it may be proved by increasing the number of training samples. The synthesizing accuracy (NC value) is decreased as increasing PD. The dependence of synthesizing accuracy of texture on pose difference is

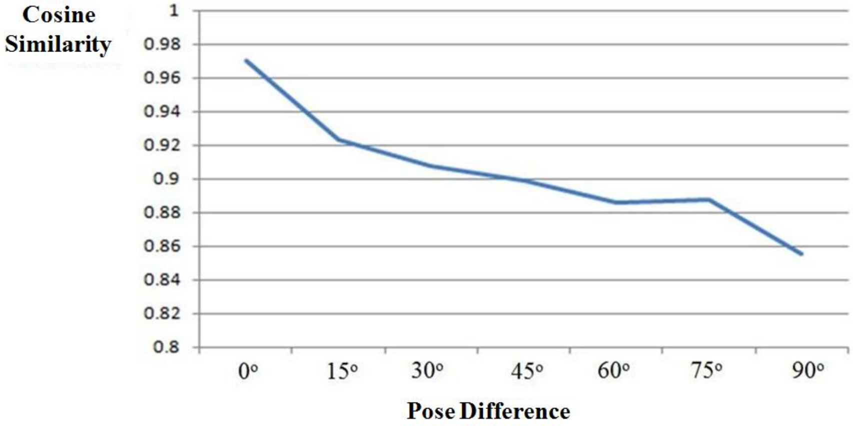


Fig. 16 NC trend of texture as PD increasing

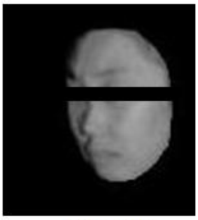
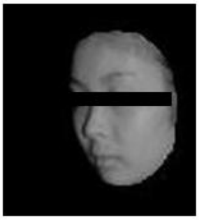
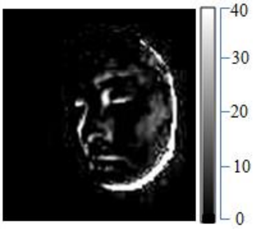
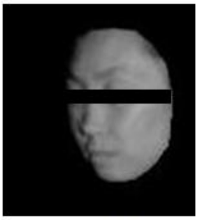
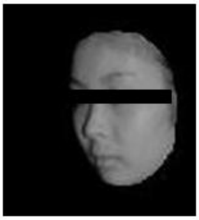
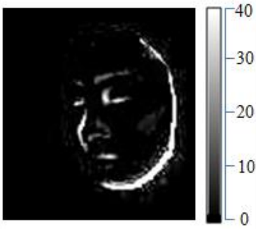
Synthesized Image	Original Image	Differential Image	Cosine Similarity
 PD=90°			0.8698
 PD=45°			0.9576

Fig. 17 Examples of synthesized results and corresponding differential images and normalized correlation

shown in Fig. 16. It is demonstrated in Fig. 16 that the average values of NC of texture are more than 0.9077 when $0 \leqslant PD \leqslant 30^\circ$ and the average value of NC is 0.8556 when $PD = 90^\circ$.

In order to discuss our experiment results qualitatively, we use Fig. 17 to show examples of synthesized results corresponding to different values of NC. The differential image, which is constructed by the absolute differential pixel values between the synthesized image and corresponding original image, is used as a reference image. Large difference in some edge regions of face is shown in the differential images for two reasons: one is that our synthesized result is generated independently with the original ones and another is that there exists small angle errors when original image is obtained. In the first row of Fig. 17, it shows synthesized result as PD is 90°(synthesize -45° pose image from 45°) and corresponding NC is 0.8698. Comparing with the original images, the synthesized result captures the most of persons' personal characteristics even there exists some divergence. In the second row of Fig. 17, it shows synthesized result as PD is 45°(synthesize -45° pose image from 0°) and corresponding NC is 0.9576. Compared the two differential images in Fig. 17, we can find that the skin appearance also can reconstructed better by applying our method for $PD = 45^\circ$ than for $PD = 90^\circ$.

5 Conclusion

Towards the problem of facial pose synthesis, this chapter presents our proposed method, tensor-based subspace learning (TSL) algorithm, for synthesizing the human multi-pose facial images. TSL has mainly two advantages. One is that multi-pose facial images are synthesized accurately only by inputting a single two-dimensional image. Another is that our method is decomposing data tensors to build learning subspaces instead of calculating the spatial information of head. This alternation leads the synthesis processing to avoid local minima solutions. It is easier to implement than previous methods and experimental results show that our method is effective for facial pose synthesis.

In the future, TSL is expected to establish a model of pose synthesis not only for a special illumination conditions but also for different illuminations or different cosmetic appearances [32].

References

1. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
2. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis for Facial Image Recognition. In: 16th International Conference on Pattern Recognition (ICPR 2002), vol. 2, pp. 511–514 (2002)
3. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and Machine Recognition of Faces: A Survey. Proceedings of the IEEE 83(5), 705–740 (1995)
4. Igarashi, T., Nishino, K., Nayar, S.K.: The Appearance of Human Skin: A Survey. Foundations and Trends in Computer Graphics and Vision 3(1), 1–95 (2007)
5. Terzopoulos, D., Waters, K.: Physically-based Facial Modeling, Analysis and Animation. Visualization and Computer Animation 1, 73–80 (1990)

6. Castelan, M., Hancock, E.R.: A Simple Coupled Statistical Model for 3d Face Shape Recovery. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 231–234 (2006)
7. Horn, B.K.P., Brooks, M.J.: Shape From Shading. NIT Press (1989)
8. Vetter, T., Poggio, T.: Linear Object Classed and Image Synthesis From A Single Example Image. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 733–742 (1995)
9. Wang, J., Zhang, C.S., Kou, Z.B.: An Analytical Mapping for LLE and Its Applications in Multi-Pose Face Synthesis. In: 14th British Machine Vision Conference, pp. 285–294 (2003)
10. Blana, V., Vetter, T.: A Morphable Model for the Synthesis of 3d Faces. In: Siggraph 1999, Computer Graphics Proceedings, pp. 187–194 (1999)
11. Tsai, Y., Lin, H.J., Yang, F.W.: Facial Expression Synthesis Based on Imitation. In: Dornaika, F. (ed.) International Journal of Advanced Robotic Systems. InTech (2012) ISBN: 1729-8806, doi:10.5772/51906, 2012
12. Wang, H.C., Ahuja, N.: Facial Expression Decomposition. In: 9th IEEE International Conference on Computer Vision (ICCV 2003), vol. 2, pp. 958–965 (2003)
13. Chen, Y.-W., Fukui, T., Qiao, X., Igarashi, T., Nakao, K., Kashimoto, A.: Multi-angle view, illumination and cosmetic facial image database (MaVIC) and quantitative analysis of facial appearance. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 411–420. Springer, Heidelberg (2008)
14. Tucker, L.R.: Some Mathematical Notes of Three-mode Factor Analysis. Psychometrika 31(3), 279–322 (1996)
15. Bader, B.W., Kolda, T.G.: Algorithm 862: Matlab Tensor Classes for Fast Algorithm Prototyping. ACM Transactions on Mathematical Software 32(4), 635–653 (2006)
16. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Journal of Matrix Analysis and Application 21, 1253–1278 (2001)
17. Yu, H.C., Bennamoun, M.: 1D-PCA,2D-PCA to nD-PCA. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 4, pp. 181–184 (2006)
18. Xu, R., Chen, Y.W.: Generalized *N*-dimensional Principal Component Analysis (GND-PCA) and Its Application on Construction of Statistical Appearance Models for Medical Volumes with Fewer Samples. Neurocomputing 72(10), 2276–2287 (2009)
19. Qiao, X., Xu, R., Chen, Y.W., Igarashi, T., Nakao, K., Kashimoto, A.: Generalized N-Dimensional Principal Component Analysis (GND-PCA) Based Statistical Appearance Modeling of Facial Images with Multipal Modes. IPSJ Transtractions on Computer Vision and Applications 1, 231–241 (2009)
20. Kolda, T.G.: Orthogonal Tensor Decompositions. SIAM Journal on Matrix Analysis and Applications 3(1), 243–255 (2001)
21. Levin, J.: Three-Mode Factor Analysis, Ph.D Thesis, University of Illinois (1963)
22. Hitchcock, F.L.: The Expression of A Tensor or A Polyadic as A Sum of Products. J. Math. Phys. 7, 164–189 (1927)
23. Lee, H.S., Tensor-Based, D.J.K.: AAM with Continuous Variation Estimation: Application to Variation-Robust Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(6), 1102–1116 (2009)
24. Hitchcock, F.L.: Multiple Invariants and Generalized Rank of a P-Way Matrix or Tensor. J. Math. Phys. 7, 39–79 (1927)
25. Cattell, R.B.: Parallel Proportional Profiles and Other Principles for Determining the Choice of Factors by Rotation. Psychometrika 9, 267–283 (1944)

26. Cattell, R.B.: The Three Basic Factor-Analytic Research Designs - Their Interrelations and Derivatives. Psych. Bull. 49, 499–520 (1952)
27. Tucker, L.R.: Implications of Factor Analysis of Three-Way Matrices for Measurement of Change. In: Problems in Measuring Change, pp. 122–137. University of Wisconsin Press (1963)
28. Kapteyn, A., Neudecker, H., Wansbeek, T.: An Approach to n-Mode Component Analysis. Psychometrika 51 (1986)
29. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
30. Xiong, P., Huang, L., Liu, C.: Initialization and pose alignment in active shape model. In: IEEE International Conference on Pattern Recognition 2010, pp. 3971–3974 (2010)
31. Terada, T., Fukui, T., Igarashi, T., Nakao, K., Kashimoto, A., Chen, Y.W.: Automatic Facial Image Manipulation System and Facial Texture Analysis. In: 5th International Conference on Natural Computation (ICNC 2009), pp. 1–5 (2009)
32. Igarashi, T., Nishino, K., Nayar, S.K.: The Appearance of Human Skin: A Survey. Foundations and Trends in Computer Graphics and Vision 3(1), 1–95 (2007)
33. Tucker, L.R.: The Extension of Factor Analysis to Three-Dimensional Matrices. In: Contributions to Mathematical Psychology, pp. 110–127 (1964)
34. Carroll, J.D., Chang, J.J.: Analysis of Individual Differences in Multidimensional Scaling via An N-Way Generalization of "Eckart-Young" Decomposition. Psychometrika 35, 283–319 (1970)
35. Harshman, R.A.: Foundations of the PARAFAC Procedure: Model and Conditions for An "Explanatory" Multi-Mode Factor Analysis. UCLA Working Papers in Phonetics 16, 1–84 (1970)
36. Henrion, R.: Body Diagonalization of Core Matrices in Three-Way Principal Components Analysis: Theoretical Bounds and Simulation. Journal of Chemometrics 7, 477–494 (1993)
37. Smilde, A.K., Wang, Y., Kowalski, B.R.: Theory of Medium-Rank Second-Order Calibration with Restricted-Tucker Models. Journal of Chemometrics 8, 21–36 (1994)
38. Bro, R.: Multi-Way Analysis in the Food Industry: Models, Algorithms, and Applications, Ph.D Thesis, University of Amsterdam (1998)
39. Bro, R., Andersson, C.A., Kiers, H.A.L.: PARAFAC2: Modeling Chromatographic Data with Retention Time Shifts. J. Chemometrics 13, 295–309 (1999)
40. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: From Matrix to Tensor: Multilinear Algebra ans Signal Processing. Mathematics in Signal Processing, 1–15 (1998)
41. Comon, P.: Tensor Decompositions: State of the Art and Applications. In: IMA Conference Mathematics in Signal Processing, pp. 18–20 (2000)
42. Chen, B., Petropolu, A., Lathauwer, L.D.: Blind Identification of Convolutive Mim systems with 3 Sources and 2 Sensors. Appl. Signal Processing 5, 487–496 (2002)
43. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications 21(4), 1253–1278 (2000)
44. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: On the Best Rank-1 and Rank-$(R_1, R_2, \cdots, R_N)$ Approximation of Higher-Order Tensors. SIAM Journal on Matrix Analysis and Applications 21(4), 1324–1342 (2000)
45. Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., Chien, L.: Text Representation: From Vector to Tensor. In: Proceedings of the 5th IEEE International Conference on Data Mining, pp. 725–728 (2005)
46. Sun, J.T., Zeng, H.J., Liu, H., Yu, Y., Chen, Z.: CubeSVD: A Novel Approach to Personalized Web Search. In: Proceedings of the 14th Internatinoal Conference on World Wide Web, pp. 382–390 (2005)

47. Beckmann, C., Smith, S.: Tensorial Extensions of Independent Component Analysis for Multisubject FMRI Analysis. NeuroImage 25, 294–311 (2005)
48. Morup, M., Harsen, L.K., Herrmann, C.S., Parnas, J., Arnfred, S.M.: Parallel Factor Analysis as An Exploratory Tool for Wavelet Transformed Event-Related EEG. SIAM Journal on Matrix Analysis and Applications 29, 938–947 (2006)
49. Kroonenberg, P.M.: Three-Mode Principal Component Anslysis: Theory and Applications. DOW Press (1983)
50. Henrion, R.: N-way Component Analysis Theory, Algorithms and Applications. Chemometrics and Intelligent Laboratory Systems 25, 1–23 (1983)
51. Coppi, R., Bolasco, S.: Multiway Data Analysis. North-Holland (1989)
52. Kroonenberg, P.M.: Applied Multiway Data Analysis. Wiley (2008)
53. Paatero, P.: The Multilinear Engine: A Table-Driven, Least Squares Program for Solving Multilinear Problems, Including the N-Way Parallel Factor Analysis Model. J. Comput. Graphical Statist. 8, 854–888 (1999)
54. Andersson, C.A., Rao, R.: The *N*-Way Toolbox for MATLAB. Chemometrics and Intelligent Laboratory Systems 52, 1–4 (2000)
55. Landry, W.: Implementing A High Performance Tensor Libernary. Sci. Programming 11, 273–290 (2003)
56. Bader, B.W., Kolda, T.G.: Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping. ACM Trans. Math. Software 32, 635–653 (2006)

Acronyms

HOSVD	High Order Singular Value Decomposition
LLE	Local Linear Embedding
MaVIC	KAO-Ritsumeikan Multi-angle View, Illumination and Cosmetic Facial Database
MF	Morphable Face Model
NC	Normalized Correlation
PCA	Principal Component Analysis
SFS	Shape from Shading Methodn
SVD	Singular Value Decomposition
TBL	Tensor-based Subspace Learning Method

Editors

Dr. Yen-Wei Chen received the B.E. degree in 1985 from Kobe University, Kobe, Japan, the M.E. degree in 1987, and the D.E. degree in 1990, both from Osaka University, Osaka, Japan. He was an Associate Professor and a Professor with the Department of Electrical and Electronic Engineering, University of the Ryukyus, Okinawa, Japan. He is currently a Professor with the college of Information Science and Engineering, Ritsumeikan University, Japan. His research interests include medical image analysis, pattern recognition and machine learning. He has published more than 300 research papers in a number of leading journals and leading conferences including IEEE Trans. Image Processing, IEEE Trans. BME, Pattern Recognition and MICCAI, ICPR, ICIP. He has received many distinguished awards including 2012 ICPR Best Scientific Paper Award, Outstanding Chinese Oversea Scholar Fund of Chinese Academy of Science.

Dr. Lakhmi C. Jain is with the Faculty of Education, Science, Technology & Mathematics at the University of Canberra, Australia and University of South Australia, Australia. He is a Fellow of the Institution of Engineers Australia.

Dr. Jain founded the KES International for providing a professional community the opportunities for publications, knowledge exchange, cooperation and teaming. Involving around 5000 researchers drawn from universities and companies world-wide, KES facilitates international cooperation and generates synergy in teaching and research. KES regularly provides networking opportunities for professional community through one of the largest conferences of its kind in the area of KES. www.kesinternational.org

His interests focus on the artificial intelligence paradigms and their applications in complex systems, security, e-education, e-healthcare, unmanned air vehicles and intelligent agents.

Author Index