Chapter 3
Common Representational Format

Abstract. The subject of this chapter is the common representational format. Conversion of all sensor observations to a common format is a basic requirement for image fusion. The reason for this is that only after conversion to a common format are the input images compatible, i. e. the input images "speak a common language" and image fusion may be performed. In this chapter we shall consider the principal theories and techniques which underlie the concept of a common representational format.

3.1 Introduction

Conversion of all input images to a common representational format is a basic requirement for image fusion. The reason for this is that only after conversion to a common format are the input images compatible and fusion may take place.

The following example illustrates the concept of a common representational format in brain research.

Example 3.1. A Standardized Brain Atlas: A Common Representational Format for Brain Research [14]. In order to compare different brains and, to facilitate comparisons on a voxel-by-voxel basis, we use a standardized anatomically-based coordinate system or *brain atlas*. The idea is that, in the new coordinate system, all brains have the same orientation and size. The transformation to this coordinate system also gives us the means to enhance weak, or noisy, signals by averaging the transformed images. The standardized brain atlas allows us to catalogue the anatomical, metabolic, electrophysiological, and chemical architecture of different brains into the same coordinate systems.

The process of converting the input images into a common representational format involves many different processes. As listed in Chapt. 1, the principal processes are:

Spatial Alignment. Transformation of the local spatial positions (x,y) to a common coordinate system. The process is often performed using a statistical matching technique and is considered in Chapt. 4.

Temporal Alignment. Transformation of the local times t to a common time axis. The process is often performed using a dynamic time warping algorithm. In many image fusion applications, the spatial and temporal algnments are performed simultaneously in a joint spatial-temporal alignment algorithm.

Feature Extraction. Transformation of the input images into corresponding feature maps. The process is performed by extracting characteristic features from the input images. Different feature extraction processes are considered in Chapts. 8, 9, 12, 13 and 16.

Decision labeling. Transformation of the input image into a corresponding decision, or label, map. The process is performed by applying decision operators on the input images or on the feature maps. The process of decision labeling is considered in Chapts. 9, 12, 16 and 17.

Semantic Equivalence. Transformation of the input images so the pixel values all refer to the same object or phenomena. This is considered in Chapt. 5.

Radiometric Calibration. Transformation of the input images so the pixel values are all measured on a common scale. The process is considered in Chapt. 6.

In many image fusion applications, the construction of a common coordinate system is the primary fusion algorithm. The following example illustrates the construction of a common representational format for an environment which is essentially static and in which the sensors are all of the same type. In this case, temporal alignment, semantic equivalence and radiometric calibration are not required and the construction of a common representational format reduces to the construction of a common spatial coordinate system.

Example 3.2. A Distributed Surveillance System [15]. The demand for surveillance activities for safety and security purposes has received particular attention for remote sensing in transportation applications (such as airports, maritine environments, railways, motorways) and in public places (such as banks, supermarkets, department stores and parking lots). Such systems typically consist of a number of video-based television cameras located in multiple locations. Consider a sequence of M narrow field-of-view "spot" images $I_m, m \in \{1,2,\ldots,M\}$, taken of a wide surveillance area.

We establish a common coordinate system by building a panoramic or "mosaic" image I^* from the sequence of images I_m (Fig. 3.1). For each image I_m, we find a geometric transformation T_m which maps the local "camera-centered" coordinate system of I_m to the common "object-centered" coordinate system of I^*. We then form the mosaic image I^* by "stitching" or "compositing" together the transformed images $T_m(I_m)$. In this case, the aim of a stitching algorithm is to produce a visually plausible mosaic image I^* in

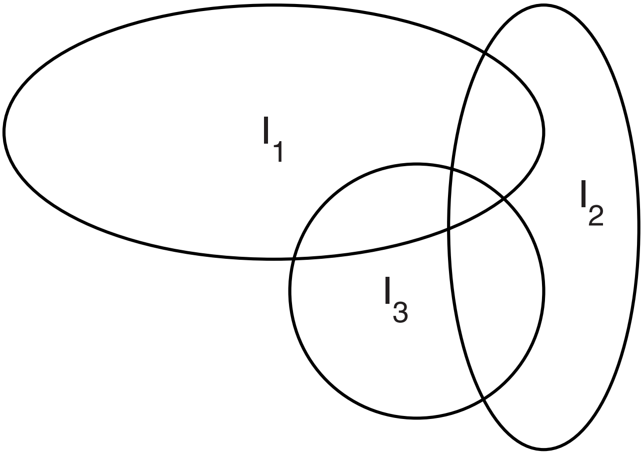


Fig. 3.1 Shows the surveillance of a wide-area site with a sequence of "spot" images I_1, I_2 and I_3. Each spot image I_m is transformed to a common coordinate system using a transformation T_m. The union of three three images define the mosaic image I^*. The stitching algorithm is used in those regions where two, or more, transformed images $T_m(I_m)$ overlap.

which, geometrically and photometrically, I^* is as similar as possible to the input images $T_m(I_m)$ and the seams between the stitched images are invisible.

3.2 Geographical Information System

An important example of a common representational format is a *Geographical Information System*. In a Geographic Information System (GIS) we combine multiple images of the earth obtained from many different sensors and maps, including demographic and infrastructure maps, into a common coordinate system.

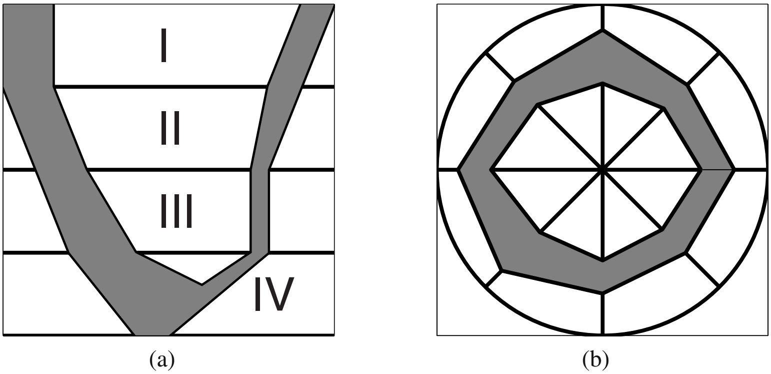
3.3 Choosing a Common Representational Format

The common representational format plays a crucial role in image fusion. In fact the choice of common representational format will often govern the fusion algorithm which is used and its performance. In the sections which follow we consider some of the issues involved in choosing an appropriate common representational format.

3.3.1 Human Fusion

In some applications image fusion is performed by a human observer. In this case, the common representational format is chosen as an aid to the human observer.

Example 3.3. Myocardial Imaging [3]. Polar maps, or “bull’s-eye” images, are a standard way of displaying myocardial functions and are well established in clinical settings. Fig. 3.2 shows the polar image representation of a left ventricle. The polar maps are constructed by combining images from multiple planes so that information about the entire myocardium can be displayed in a single image. Polar maps can be compared to a three-dimensional cone-shaped heart activity image projected onto a single plane. Each image plane forms a ring in the polar map. Although the rings may be divided into an arbitrary number of sectors, in practice, a clinician uses four (anterior, lateral, inferior and septal) or six (anterior, anterior-lateral, inferior-lateral, inferior, inferior-septal and anterior-septal) sectors for his visual interpretation of the image.



(a) (b)

Fig. 3.2 Shows a polar image of a left ventricle. **(a)** Shows the the left ventricle divided into four slices: I (Basal), II (Mid-Basal), III (Mid-Apical) and IV (Apical). **(b)** Shows a given slice divided into 8 sectors.

3.3.2 Sparseness

In some applications only a small number of image pixels are transformed into the common representational format. The result is a sparse representation. The following example illustrates the conversion of an input image into a sparse representation which is scale and rotation invariant and is invariant across a wide range of distortions and changes in illumination.

Example 3.4. Scale Invariant Feature Transform (SIFT) [9, 12]. SIFT key-points (see Chapt. 13) are invariant to image scale and rotation and provide robust matching across a substantial range of affine distortion changes in three-dimensional viewpoint, occlusion, noise and changes in illumination. In addition, the features are highly distinctive in the sense that a single feature can be reliably matched with only a few other SIFT key-points. The SIFT common representational format is a vector containing $L = 128$ components:

$$\mathbf{s} = (s(1), s(2), \ldots, s(128))^T \ .$$

It is computed by partitioning the image region surrounding each key-point into a 4×4 grid of sub-regions, and computing an 8-bin orientation histogram in each subregion. The 128-component vector is then formed by concatenating the 16 orientation histograms.

3.3.3 Object Recognition

In object recognition applications we require a common representational format which is invariant to translation, rotation and scaling and is insensitive to variations due to articulation, occlusion and noise (Fig. 3.3). The following example illustrates shape-context. This is a common representational format which is used for comparing two-dimensional shapes and contours.

Example 3.5. Shape Context [4]. Let C denote a closed contour in a given input image. Its shape context S_c is defined as follows. Let $z_i = (x_i, y_i), i \in \{1, 2, \ldots, M\}$, denote a set of M sample points on the contour C (Fig. 3.4). Then $S_c = (S_1, S_2, \ldots, S_M)$, where S_i is a two-dimensional histogram which describes the distribution of the distances and orientations of the points $z_j, j \neq i$, relative to z_i.

Let r_{ij} and θ_{ij} denote the distance and orientation of the point z_j relative to z_i. To define θ_{ij} we require an axis. Two rotationally invariant axes which are commonly used for this purpose are:

(1) The tangent of the contour at z_i,
(2) The line which joins the center of mass of the contour and the point z_i.

In Fig. 3.4 we show the first option, i. e. we measure the angles θ_{ij} relative to the tangent of the contour at z_i.

We divide the (r, θ) space into L vertical columns $\Theta_l, l \in \{1, 2, \ldots, L\}$, and K horizontal rows $R_k, k \in \{1, 2, \ldots, K\}$. If $h(k, l)$ is the number of points $z_j, j \neq i$, for which r_{ij} lies in the kth row and θ_{ij} lies in the lth column, then

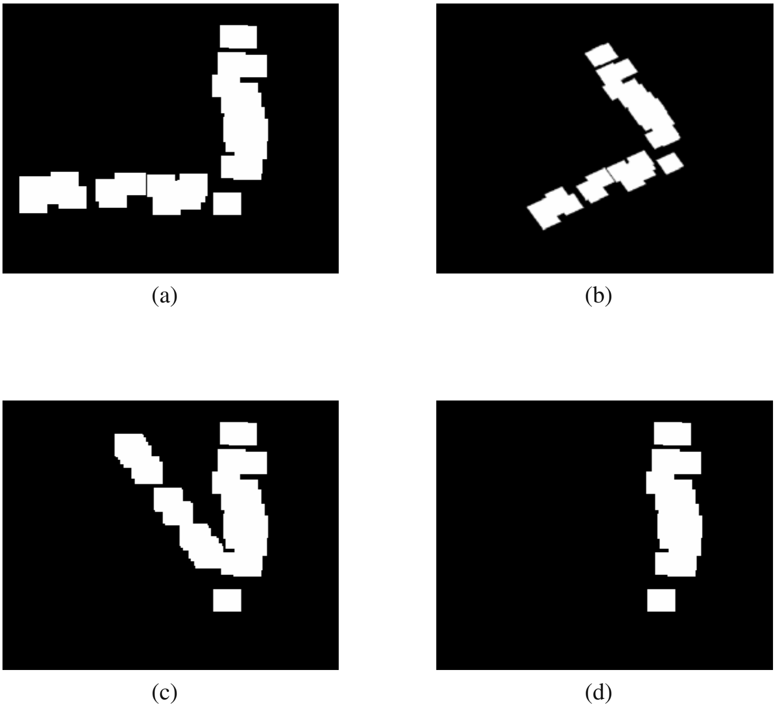


Fig. 3.3 **(a)** Shows the original image. **(b)** Shows the rotation and scaling of the original image.**(c)** Shows the articulation of the original image.**(d)** Shows the occlusion of the original image.

$$S_i = \begin{pmatrix} h(1,1) & h(1,2) & \dots & h(1,K) \\ h(2,1) & h(2,2) & \dots & h(2,K) \\ \vdots & \vdots & \ddots & \vdots \\ h(K,1) & h(K,2) & \dots & h(K,L) \end{pmatrix} .$$

The shape context $S_c = (S_1, S_2, \dots, S_M)$ is invariant to translation and rotation. To a good approximation it is also invariant to occlusion and to noise. Although S_c is not scale invariant it may be made scale invariant by dividing the distances r_{ij} through with the mean distance $\bar{r}$, where

$$\bar{r} = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1, j \neq i}^{M} r_{ij} .$$

Although widely used for comparing two-dimensional shapes and contours (see Ex. 5.6) it is less successful when comparing articulated objects. In this case we may use a modified shape context common representational representation known as the inner distance [8].

Example 3.6. Inner Distance [8]. Suppose the object shown in Fig. 3.4 is in fact an articulated object consisting of two parts A and B which are hinged together at X. Then the inner distance is defined as follows: For any two points z_i and z_j the inner distance is equal to the shape context except we replace r_{ij} with R_{ij}, where R_{ij} is the length of the shortest jointed line which joins z_i and z_j and which lies within the given contour. To a good approximation the inner distance is invariant as we change the angle between A and B.

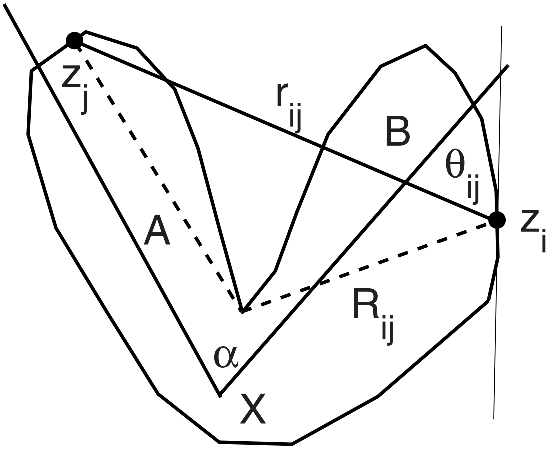


Fig. 3.4 Shows a closed contour defined by M points $z_i, i \in \{1,2,\ldots,M\}$. The line joining the points z_i and z_j has a Euclidean length r_{ij} and it makes an angle θ_{ij} with the tangent at z_i. We use the distance r_{ij} and the angle θ_{ij} to define the shape context of the contour C. We obtain the inner distance by replacing r_{ij} with R_{ij}, the length of the shortest jointed line which joins z_i and z_j and which lies inside the closed contour.

3.3.4 Uncertainty

In choosing an appropriate common representational format we must take into account how the measurement uncertainty will propagate in a given format. The following example illustrates these concerns.

Example 3.7. Object Recognition Based on Photometric Color Invariants [5]. A simple and effective scheme for three-dimensional object recognition is to represent and match images on the basis of color histograms. For effective object recognition we should use a color space which is invariant to changes in viewing direction, object orientation and illumination.

In Table 3.1 we list several color spaces which are commonly used for this purpose (see Chapt. 16). We observe that measurement uncertainty is propagated differently in each space: the normalized *rg* space is unstable around $R = G = B = 0$ ($\sigma_r, \sigma_g \rightarrow \infty$) and hue H is unstable around $R = G = B$ ($\sigma_H \rightarrow \infty$) while the opponent color space o_1, o_2 is relatively stable at all *RGB* values [6].

Table 3.1 Photometric Invariant Color Space

Color Space	Definition	Uncertainty
Normalized *rg*	$r = R/S$, $g = G/S$	$\sigma_r = \sqrt{R^2(\sigma_B^2 + \sigma_G^2) + (G+B)^2\sigma_R^2}/S.$ $\sigma_g = \sqrt{G^2(\sigma_B^2 + \sigma_R^2) + (R+B)^2\sigma_G^2}/S.$
Opponent o_1o_2	$o_1 = (R-G)/2,$ $o_2 = (2B-R-G)/4$	$\sigma_1 = \sqrt{\sigma_G^2 + \sigma_R^2}/2$, $\sigma_2 = \sqrt{4\sigma_B^2 + \sigma_G^2 + \sigma_R^2}/4.$
Hue H	$\tan(H) = \sqrt{3}(G-B)/$ $(2R-G-B)$	$\sigma_H^2 = 3(\sigma_G^2(-2BR+1) + \sigma_B^2(G^2-2GR)/\Delta + \sigma_R^2(1$ $+(G^2-2GB)/\Delta) + R^2\sigma_B^2\sigma_G^2)/4.$

$S = (R+G+B)$, $\Delta = R^2 + B^2 + G^2 - GR - B(G+R)^2$.

3.4 Textures

The local binary pattern (LBP) operator [2, 10] is an efficient method for representing image textures. The operator takes a local neighborhood around each pixel and thresholds the pixels in the neighborhood according to the value of the center pixel. The resulting binary valued image patch forms a normalized local texture descriptor of the image A. For a 3×3 neighborhood centered on the pixel (m,n), the LBP operator is defined as

$$LBP(m,n) = \sum_{k=0}^{7} s(A(m,n), A(i_k, j_k))2^{(k-1)} ,$$

where (i_k, j_k) are the coordinates of the kth pixel in the 3×3 neighborhood of (m,n) and

$$s(A(m,n), A(i_k, j_k)) = \begin{cases} 1 \text{ if } A(i_k, j_k) > A(m,n) , \\ 0 \text{ otherwise} . \end{cases}$$

The LBP encoding process is illustrated in Fig. 3.5.

98	85	32
94	70	43
87	27	44

(a)

1	1	0
1	–	0
1	0	0

(b)

Fig. 3.5 Shows a local binary pattern operator. **(a)** Shows the gray-levels in a 3×3 neighborhood in the image A. **(b)** Shows the thresholded values. Starting from the top left-hand corner and moving in a clockwise direction, the LBP for the 3×3 local neighborhood is LBP=$\sum_{k=0}^{7} s(A(m,n),A(i_k,j_k))2^{(k-1)} = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 = 195$.

An important extension of the LBP is the uniform LBP.

Example 3.8. Uniform Local Binary Pattern [11]. A LBP is "uniform" if it contains at most one $0-1$ and one $1-0$ transition when viewed as a circular string. For example, the LBP code in Fig. 3.5 is uniform. In a 3×3 window, only 58 of the $2^8 = 256$ patterns are uniform.

3.5 Multi-scale Representation

In some applications a multi-scale representation of the input image I is required. A wavelet decomposition (see Chapt. 8) is often used for this purpose. Another example is the SIFT key-points (Chapt. 13) which are defined as local extrema in the following multi-scale representation of I:

$$\{D(m,n|\sigma_1), D(m,n|\sigma_2), \ldots, D(m,n|\sigma_L)\} ,$$

where $D(m,n|\sigma_l)$ is the difference-of-Gaussian (DoG) representation of I at a scale σ_l:

$$D(m,n|\sigma_l) = I(m,n) \otimes G(\sigma_l) - I(m,n) \otimes G(\sigma_{l+1}) ,$$

and $I(m,n) \otimes G(\sigma_l)$ is the convolution of $I(m,n)$ with the two-dimensional zero-mean Gaussian $G(\sigma_l)$.

3.6 Sub-space Methods

In many image fusion applications an important consideration is to keep the computational load and/or the storage requirements low. This may be achieved by using a *low-dimensional* common representational format. One way of producing such a format is to apply a dimension-reducing, or sub-space, technique to the input images.

Table 3.2 lists some of the principal sub-space techniques which are commonly used for this purpose.

Table 3.2 Sub-space Techniques

Technique	Description
Principal Component Analysis (PCA)	Linear transformation chosen so the projected components have maximum variance.
Linear Discriminant Analysis (LDA)	Linear transformation for $K \geq 2$ classes. Transformation is chosen so the projected components for each class are maximally separated from the projected components of the other classes.
Independent Component Analysis (ICA)	Linear transformation chosen so the projected components have maximized independence.
Non-Negative Matrix Factorization (NMF)	Finds factors with non-negative elements.
Canonical Correlation Analysis (CCA)	For $K = 2$ finds the two transformations, one for each class, in which the projected components of the two classes are maximally correlated.

The following example illustrates the classic dimension reducing method of principal component analysis (PCA).

Example 3.9. Principal Component Analysis (PCA) [7]. The aim of principal component analysis (PCA) is to find a L-dimensional *linear* projection that best represents the input data in a least squares sense. Let the input data be K $M \times N$ input images $A_k, k \in \{1,2,\ldots,K\}$. In classical PCA we first write each A_k as a column vector $\mathbf{a}_k$:

$$\mathbf{a}_k = (a_k(1), a_k(2), \ldots, a_k(MN))^T .$$

Then we use a set of orthonormal axes $\mathbf{u}_l, l \in \{1,2,\ldots,L\}$, to obtain a L-dimensional representation of $\mathbf{a}_k$:

$$\widetilde{\mathbf{a}}_k = U^T(\mathbf{a}_k - \bar{\mathbf{a}}) ,$$

where

$$\begin{aligned} U &= (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_L) \ , \\ \mathbf{u}_l &= (u_l(1), u_l(2), \ldots, u_l(MN))^T \ , \\ \widetilde{\mathbf{a}}_k &= (\widetilde{a}_k(1), \widetilde{a}_k(2), \ldots, \widetilde{a}_k(L))^T \ . \end{aligned}$$

Mathematically, the orthornormal axes $\mathbf{u}_l, l \in \{1,2,\ldots,L\}$, are given by the L dominant eigenvectors of the sample covariance matrix S:

$$S\mathbf{u}_l = \lambda_l \mathbf{u}_l \ ,$$

where

$$\begin{aligned} S &= \frac{1}{K} \sum_{k=1}^{K} (\mathbf{a}_k - \bar{\mathbf{a}})(\mathbf{a}_k - \bar{\mathbf{a}})^T \ , \\ \bar{\mathbf{a}} &= \frac{1}{K} \sum_{k=1}^{K} \mathbf{a}_k \ . \end{aligned}$$

3.7 Multiple Training Sets

A recent development in image fusion is *ensemble learning* (see Chapt. 10) in which we employ an ensemble, or collection, of *multiple* decision maps $D_k, k \in \{1,2,\ldots,K\}$, where each function D_k is learnt on its own training set T_k. Given a common (base) training set T^* we may generate an ensemble of training sets, $T_k, k \in \{1,2,\ldots,K\}$, which share the same common representational format by simply sub-sampling T^*.

Example 3.10. Bootstrapping. Given a base training set T^* of N measurements $T^*_m, m \in \{1,2,\ldots,M\}$, we create a set of K bootstrapped training sets $T_k, k \in \{1,2,\ldots,K\}$: Each bootstrapped training set T_k consists of M measurements which are selected by randomly sampling T^* with replacement. The following matlab code can be used to create a bootstrapped training set T_k.

```
for $m = 1:M$
    $index = floor(rand(1,1) * (M-1) + 1)$
    $T_k(m) = T^*(index)$
end
```

Table 3.3 Methods for Ensemble Sampling a Base Training Set T^*

Method	Description
Sub-sampling	Partition the base training set T^* into K disjoint slices (similar to that used in cross-validation). Each classifier S_k is trained on a training set T_k, where T_k is the base training set T^* *less* the examples in the k slice.
Bootstrapping	Perturb T^* by randomly sampling T^* with replacement. The sampling is made with a uniform probability random selection procedure. The entire procedure is repeated K times to create K different, although overlapping, training sets T_k. Each T_k contains N samples. On average each perturbed training set will have 63.2% of the samples in T^*, the rest being duplicates.
Boosting	We use the classification results obtained with the kth classifier, C_k, to learn T_{k+1}. The classifier C_k is itself learnt on T_k. The training set T_{k+1} is created by re-sampling T^* such that samples which are misclassified by S_k have a higher chance of being chosen than samples which were correctly classified by C_k.
Class Switching	Perturb T^* by randomly swapping a small percentage of class labels. The procedure is repeated K times to create K training sets $T_k, k \in \{1,2,\ldots,K\}$.

In Table 3.3 we list some methods for ensemble sampling the training set T^*.

Sometimes we require each training set T_k to have its own common representational format. This is a case of *multiple common representational formats*. Given a common (base) training set T^*, we may generate an ensemble of training sets $T_k, k \in \{1,2,\ldots,K\}$, where each T_k has a different common representational format, by applying a sub-space technique to T^* and then sub-sampling (with, or without, replacement) the result. For further details see Chapts. 10 and 11.

3.8 Software

The following matlab toolboxes are of general utility in creating a common representational format.

LIBRA. A matlab toolbox for classical and robust statistics. Authors: Sabine Verboven and Mia Hubert [16].

LBP, GETMAPPING. Two matlab m-files for the local binary pattern operator. Author: T. Ahonen [1].

MATLAB IMAGE PROCESSING, STATISTICAL AND WAVELET TOOLBOXS. Matlab toolboxes.

STPRTOOL. A statistical pattern recognition toolbox. Authors: Vojtech Franc and Vaclav Hlovac.

3.9 Further Reading

The local binary pattern (LBP) operator has been intensely investigated by the Machine Vision Group at the University of Oulu. Two important references on the subject are: [1, 11]. A modern reference which provides many pointers to the literature on the subject of invariant color spaces is [17]. For references on sub-space methods (see Chapt. 9). For object recognition we require a method for extracting two-dimensional shape enclosing contours. A modern reference on this subject is [13].

References

1. Ahonen, T.: Face and texture image analysis with quantized filter response statistics. PhD thesis, University of Oulu, Finland (2009)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Trans Patt. Anal. Mach. Intell. 28, 2037–2041 (2006)
3. Behloul, F., Lelieveldt, B.P.E., Boudraa, A., Janier, M., Revel, D., Reiber, J.H.C.: Neurofuzzy systems for computer-aided myocardial viability assessment. IEEE Trans. Medical Imag. 20, 1302–1313 (2001)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans Patt. Anal. Mach. Intell. 24, 509–522 (2002)
5. Gevers, T., Smeulders, A.W.M.: Color-based object recognition. Patt. Recogn. 32, 453–464 (1999)
6. Gevers, T., Stockman, H.M.G.: Robust photometric invariant region detection in multispectral images. Int. J. Comp. Vis. 53, 135–151 (2003)
7. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
8. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Trans. Patt. Anal. Mach. Intell. 29, 286–299 (2007)
9. Lowe, D.G.: Distinctive image features from scale-invariant key-points. Int. J. Comp. Vis. 60, 91–100 (2004)
10. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Patt. Recogn. 29, 51–59 (1999)
11. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Patt. Anal. Mach. Intell. 24, 971–987 (2002)
12. Mikolajczek, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans Patt. Anal. Mach. Intell. 27, 1615–1630 (2005)
13. Schlei, B.R.: A new computational framework for 2D shape-enclosing contours. Image Vis. Comp. 27, 637–647 (2009)
14. Thompson, P.M., Mega, M.S., Narr, K.L., Sowell, E.R., Blanton, R.E., Toga, A.W.: Brain image analysis and atlas construction. In: Handbook of Medical Imaging, Washington, USA, vol. 2. SPIE Press, Bellington
15. Valera Espina, M., Velastin, S.A.: Intelligent distributed surveillance systems: a review. IEE Proc. Vis. Image Sig. Process. 152, 192–204 (2005)
16. Verboven, S., Hubert, M.: Libra: A matlab library for robust analysis. Chemometrics and Intell. Laboratory Syst. 75, 127–136 (2005)
17. van de Weijer, J., Gevers, T., Smeulders, A.W.M.: Robust photometric invariant features from the color tensor. IEEE Trans. Image Process. 15, 118–127 (2006)