International Journal of Computer Vision 75(2), 267–282, 2007

DOI: 10.1007/s11263-006-0033-9

POP: Patchwork of Parts Models for Object Recognition

YALI AMIT
Department of Statistics and the Department of Computer Science, University of Chicago, Chicago, IL 60637
amit@marx.uchicago.edu

ALAIN TROUVÉ
CMLA at the Ecole Normale Superieur, Cachan

Received April 4, 2005; Accepted December 27, 2006

First online version published in January, 2007

Abstract. We formulate a deformable template model for objects with an efficient mechanism for computation and parameter estimation. The data consists of binary oriented edge features, robust to photometric variation and small local deformations. The template is defined in terms of probability arrays for each edge type. A primary contribution of this paper is the definition of the instantiation of an object in terms of shifts of a moderate number local submodels—*parts*—which are subsequently recombined using a patchwork operation, to define a coherent statistical model of the data. Object classes are modeled as mixtures of patchwork of parts (POP) models that are discovered *sequentially* as more class data is observed. We define the notion of the *support* associated to an instantiation, and use this to formulate statistical models for multi-object configurations including possible occlusions. All decisions on the labeling of the objects in the image are based on comparing likelihoods. The combination of a deformable model with an efficient estimation procedure yields competitive results in a variety of applications with very small training sets, without need to train decision boundaries—only data from the class being trained is used. Experiments are presented on the MNIST database, reading zipcodes, and face detection.

Keywords: deformable models, model estimation, multi-object configurations, object detection

1. Introduction

Two directions of research—categorization and detection—have dominated the field of shape and view based object recognition. The first, categorization, refers to the classification between several object classes based on segmented data (see Vapnik, 1995; Amit and Geman, 1997; LeCun et al., 1998; Hastie and Simard, 1998; Belongie et al., 2002), and the second, detection, to finding instances of a particular object class in large images (see Leung et al., 1995; Rowley et al., 1998; Viola and Jones, 2004; Amit and Geman, 1999; Burl et al., 1998; Torralba et al., 2004). The latter is often considered as a problem of classification between object and background. Both subjects are viewed as building blocks towards more general algorithms for the analysis of complex scenes containing multiple objects.

The challenge of computer vision is the analysis of images with multiple interacting objects and clutter, requiring some methodology for integrating the different detectors and classifiers in one framework, as well as sequentially learning additional object classes from new examples, without access to earlier training sets.

Imagine running detectors for each object class at low false negative rates. This will typically yield quite a large number of false positives as well as multiple hits (for different detectors) in the same region. It is then necessary to classify among these and eliminate false positives. Furthermore, if several objects can be present in the scene, one needs to choose among multiple candidate *interpretations*, i.e. different assignments of labels, locations, and instantiations for a number of objects, possibly occluding each other. This can not be performed based on pre-trained classifiers among the virtually infinite number of possible configurations, and requires *online* procedures. The same issue would arise if bottom-up segmentation, or saliency detection are used to determine candidate regions or locations of the objects of interest. Competing segmentations/classifications need to be resolved.

We propose to address these challenges in a coherent statistical framework, based on a novel family of deformable object models, which can be composed to define models for multi-object configurations. The data at each pixel, in our case binary oriented edges, is assumed *independent* conditional on the instantiation, which consists of a non-linear deformation of the model. The basic idea is to describe the deformation in terms of shifts of a moderate number of local submodels, parts, which are subsequently recombined using a patchwork operation, to define a coherent model of the data—hence the name patchwork of parts (POP) model. The optimal deformation and associated likelihood of the data can be efficiently computed through iterative optimization on the shifts.

Training is a challenge in models with high dimensional instantiation parameters, because these are typically *unobserved*. The specific form of the proposed deformable object model motivates an approximate estimation procedure, where each of the parts is estimated separately and for each part the only unobserved variable is a local shift. This procedure is only approximate, however it is very fast and yields very good estimates.

Given an instantiated object model we introduce the notion of the *support*, and the *visible support*—the non-occluded subset of the support. This leads to another contribution of this paper: a well defined mechanism for composing instantiated objects, *online*, into a data model for an interpretation, i.e. a configuration of objects with occlusions (see Fig. 3.) All decisions are then based on likelihood ratios between competing classes or competing interpretations. Most existing object detection or categorization approaches do not have this modular capability (see Section 1.1).

An important advantage of using statistical models is that training can be performed one class at a time. There is no need to see all the classes ahead of time in order to compute decision boundaries. Moreover due to the explicit modeling of object deformations, state of the art performance can be achieved with much smaller training sets.

1.1. Other Work

1.1.1. Deformable Models. In the object recognition literature, most statistically formulated deformable models are 'constellation' type models such as Burl et al. (1998), Crandall et al. (2005). This consists of a distribution on the geometric arrangement of 'rigid' parts, and the assumption that conditional on the arrangement, the distribution of the data at the different parts is independent. In Burl et al. (1998), the distribution of the grey level data on the support of a part has the form of a Gaussian and the data off these supports is assumed i.i.d. Gaussian. In Crandall et al. (2005), the data model is defined in terms of oriented edges, with the same type of conditional independence assumption used here and in Amit (2002). However, in both models, the statistical distribution on the data is well defined only if the parts do not overlap. This constraint is a drawback of both approaches in that large areas of the object are modeled as background, leading to a loss in precision and discriminatory power. Furthermore, the 'gaps' render unsupervised training problematic. Indeed, in Crandall et al. (2005), the centers of the parts are given by the user on the training images.

In the constellation model in Fei-Fei et al. (2003), the data are no longer modeled as a dense set of features, rather the image data is transformed to a sparse point process using local filters that fire with low probability on generic background. The instantiation has the form of a correspondence between the model points and a subset of the point process. In Fei-Fei et al. (2003), a principled probabilistic model is proposed for the transformed data together with a well formulated EM type estimation procedure, which is needed to overcome the fact that the correspondence is unobserved. Detection and classification are performed by computing the maximum posterior on constellations. In these models, the average number of points detected in an image as well as the number of interest points in the object models need to stay very small to avoid a combinatorial explosion in the learning process and in the detection and classification steps. This can be problematic for discriminating between very similar classes, as is the case in character recognition problems, or to achieve very low false positive rates in detection problems.

In this context the main contribution of the POP models is threefold: (i) the formulation of a dense data model explaining all edge data on the object allowing for fine discrimination between similar shapes, (ii) a simple and efficient training procedure for the models, (iii) the definition of object supports and the ability to compose object models to scene models.

1.1.2. Deformable Nearest Neighbor Approaches. The work in Hastie and Simard (1998), Wiskott et al. (1997), Belongie et al. (2002) involves explicit modeling of the deformations of objects but classification is based on nearest neighbors. These nearest neighbor approaches, each of which has been highly successful, can be viewed as assigning a template to each training example, thus requiring intensive computation and extensive memory. The distances are not explicitly formulated in a statistical framework and are somewhat ad-hoc. One conclusion of this paper is that statistical modeling and estimation procedures yield compact and efficient representations of the shape ensembles (e.g. handwritten digits, faces, etc.) where distances are defined in a principled manner in terms of likelihoods.

1.1.3. Dense Representations. As indicated, ours is a dense deformable template model. Since classification or detection require the estimation of the deformation, the end result is not only a class label or a location of the object in the scene but an explicit map of the model into the image. A by product is the identification of an *object support*, at the level of edges. On areas where the object is 'flat' edges are not detected and thus do not get included in the support.

In Borenstein et al. (2004) object representations are explicitly learned in order to accurately define a support or a figure ground segmentation. Their representation is also defined in terms of a collection of overlapping parts, and in each part the region corresponding to object or background is learned. The authors use a gray level data representation so that the object support includes all pixels on the object. In Leibe and Schiele (2003) and Liebe and Schiele (2004), a probabilistic Hough transform based on scale-invariant interest points is proposed for object detection and object/background segmentation. The use of predetected interest points puts this algorithm in the sparse category described above. However the authors also propose a method for determining a dense object support. The interest points on a detected object use a learned 'support probability map' relative to the point location to cast votes for points as object supports. These approaches do not offer a clearly defined statistical model for the image data (object + background) and it is therefore unclear how classification among several classes is performed, nor how object instantiations can be composed to model multi-object configurations.

1.1.4. Comprehensive Image Models. The idea of composing similar types of object models into interpretation models was initially explored in Amit et al. (2004), in the context of reading license plates. However there the object variation was limited to a small range of linear transformations, the objects had disjoint supports and no training was needed since the object classes were predefined in terms of a binary template.

The models described above mainly describe the data around one or several known objects, assuming at best a very simple model for data off the object. Others have attempted to develop more comprehensive and complex models for the 'background', see Tu et al. (2004), at a significant computational cost both in estimation and in recognition.

Our work is motivated in part by the philosophy proposed in Geman et al. (2002) where the authors argue for a a hierarchy of compositions of increasingly complex elements—reusable parts—leading to a likelihood based choice of the optimal interpretation. In their proposal, an interpretation involves not only the objects and their poses, but an assignment of part labels to structures in the background that are not associated to any object. Here also, there remain however significant challenges in terms of training and computation.

1.2. Summary of Results

The proposed models allow for a simple and efficient training procedure from small sample sizes and yield high classification rates on isolated hand written digits. For example with 30 examples per class on the MNIST dataset we achieve 3% error on the test set compared to 6% error with SVM's on the same edge features. We reach 1.52% error with 500 examples per class, where in effect only 80–100 examples were actually used to update the model parameters through sequential learning. Using a different clustering mechanism and with 1000 examples per class we achieve **.8%** error, reaching **.68%** error with the full training set.

The models trained for isolated digits are applied to zipcode reading by defining interpretation models through the composition of object instantiations. No additional training is performed and a dynamic programming algorithm is used to compute the most likely interpretation. We achieve a recognition rate of **88.7%**, with the correct zipcode being in the top 10 interpretations for 94% of the zipcodes. These rates are higher than results reported in the literature, and are of particular interest since no presegmentation or preprocessing is performed.

To test the relevance of these models to other object types, we train face models on 400 faces from the Ollivetti data base images. The likelihood ratio with respect to an adaptive background model is used as a filter on detections of the algorithm described in Amit (2002). The reduction in false positives is by a factor of 30–40. The resulting false positive rate at **12%** false negatives is under **1** false positive per image on the CMU dataset including rotated images, which is somewhat higher than the state of the art (see e.g. Viola and Jones, 2004; Schneiderman and Kanade, 2004) but given the simplicity of the test and the lack of training on any background images, we believe it is evidence of the generality and usefulness of the proposed models.

2. The Patchworks of Parts (POP) Model

The data model we propose is based on coarse binary oriented edge features (see Amit and Geman, 1999), computed at each point in the image which is defined on a grid L. We write $X = \{X_e(x) \mid x \in L, e = 1, \ldots, E\}$, where $E = 8$, corresponding to 8 orientations at increments of 45 degrees. This can also be viewed as 4 orientations with 2 polarities per orientation. These features are highly robust to intensity variations. Each detected edge is spread

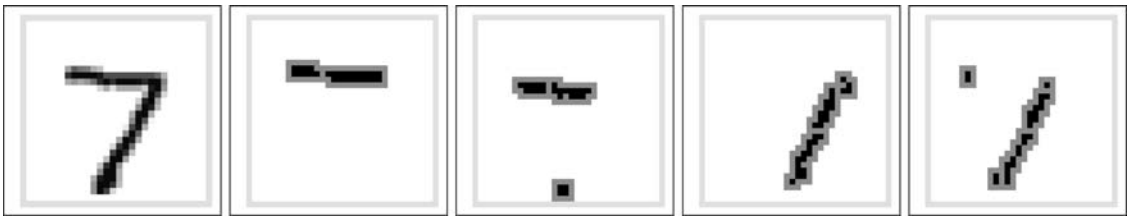

Figure 1. A sample digit image with edge maps for four of the eight orientations. The first two are horizontal with opposite polarities, and the last two are vertical with opposite polarities. In black are the original edge locations in gray are the locations after spreading.

to its immediate 3×3 neighborhood. This spreading operation is crucial in providing robustness to small local deformations, and greatly improves performance of any classifier implemented on the data. In Fig. 1, we show edge maps for four orientations on a sample image from MNIST. In black are the original edges and in gray are the locations after spreading. It is important to note that the features are not mutually exclusive, several edges can be found at the same locations. These edge features have proved useful in a large number of applications such as character recognition, detection and recognition of 3d objects, medical imaging (see Amit, 2002) as well as reading license plates from rear photos of cars (see Amit et al., 2004).

2.1. A Rigid Model

To motivate and introduce notation, we start with a rigid model for an object in which the object instantiation θ is defined only by its location. A probability array $(p_e(y))_{y\in\mathbb{Z}^2}$ is defined on the 2d lattice. Given the object is at location r and the rest of the image is empty, the edges observed in the image are assumed independent with marginal probabilities given by

$$P(X_e(x) = 1 \,|\, \theta) = p_e(x;\theta) \doteq p_e(x - r),\ x \in L.$$

The probability array is simply shifted to r. By the notation $p_e(x;\theta)$ we mean the probability of edge type e at x given the instantiation θ. Clearly, outside some region around the origin, the probabilities p_e are zero. To model the possibility that edges are observed outside the object, we define the object *support* $S(\theta)$ and assume a background model outside $S(\theta)$, where the edges are still independent but with some non-zero homogeneous marginal probabilities $p_{e,\text{bgd}}$. The support $S(\theta)$ is defined as the set of points where at least one of the marginal probabilities is greater than some fixed threshold ρ

$$S(\theta) = \{x \in L : \max_e p_e(x;\theta) \geq \rho\}. \tag{1}$$

The idea is that locations where all probabilities are low do not represent areas with 'edge activity' on the object and hence can not really be distinguished from background. Here the support at $\theta = r$ is simply the shift of the support at 0, i.e. $S(\theta) = S(0) + r$. The more complex definition in (1) is needed in the more general setting below.

2.2. A Deformable Model

The rigid model is too constrained and does not accommodate object variability. Indeed, without taking into consideration this variability, the assumption of conditional independence is grossly inadequate. As a simple extension, we assume n reference points $(y_i)_{i=1,\ldots,n}$ in the 2d lattice, and define an instantiation as a location r together with a sequence of shifts: $\theta = (r, \mathbf{v}) \doteq (r, v_1, \ldots, v_n)$. Each reference point y_i is mapped to $z_i = r + y_i + v_i$, and each vector $r + v_i$ represents a rigid shift of the model as described above. However, unless all the v_i are equal, these shifts are not consistent. To reconcile the different shifted models we recombine them as follows. Pick a non-negative kernel $K(x, y) = K(x - y)$, which decays quickly to zero as $|x - y| \to \infty$, and perform an averaging operation at each point:

$$p_e(x;\theta) = \frac{\sum_{i=1}^n p_e(x - r - v_i)K(x - z_i)}{\sum_{i=1}^n K(x - z_i)}. \tag{2}$$

(We assume $p_e(x;\theta) = 0$ if $K(x - z_i) = 0$ for all i.) In other words, the contribution of the i'th shift of the probability map to the marginal probability at point x depends on the quantity $K(x - z_i)$, it is most affected by shifted models centered around points z_i that are close by. The influence on x decreases to zero as x moves further away from z_i.

Many choices are possible for the kernel K. For simplicity, for computational efficiency and to motivate the estimation procedure, we choose $K(u)$ to be the indicator for a square neighborhood W of the origin: $K(x - y) = \mathbf{1}_W(x - y)$. In this case define the *part* Q_i associated to the reference point y_i as the subarray of p_e around y_i:

$$Q_i \doteq (p_e(y_i + s))_{s\in W}, \quad e = 1, \ldots, E,$$

Let $\mathcal{I}(x) = \{i : x \in z_i + W\}$ be the set of shifted windows covering x. Equation (2) reduces to

$$p(x;\theta) = P(X_e(x) = 1 \mid \theta) = \begin{cases} \dfrac{1}{|\mathcal{I}(x)|} \displaystyle\sum_{i \in I(x)} p_e(x - z_i + y_i) & \text{if } \mathcal{I}(x) \neq \emptyset \\ 0 & \text{if } \mathcal{I} = \emptyset. \end{cases} \tag{3}$$

which can be thought of as a *patchwork* of parts (POP) model.

Given the object is present in the image at instantiation θ, we assume again that the edges in the image are conditionally independent with marginal probabilities given by (2). As in the rigid case, we define the instantiated object support as

$$S(\theta) = \{x \in L : \max p_e(x;\theta) \geq \rho\}. \tag{4}$$

Here the support cannot be expressed in a simple form in terms of θ and the support $S(0)$ corresponding to $r = 0$, $\mathbf{v} = 0$. A background model outside the support is defined as above and the distribution of the data assuming one object in the image at instantiation θ and background edges outside the object is given by

$$P(X \mid \theta) = \prod_{x \in S(\theta)} \prod_e [p(x;\theta)]^{X_e(x)} [1 - p(x;\theta)]^{(1 - X_e(x))} \times \prod_{x \notin S(\theta)} \prod_e p_{e,\text{bgd}}^{X_e(x)} [1 - p_{e,\text{bgd}}]^{(1 - X_e(x))} \tag{5}$$

Dividing the expression in (5) by the likelihood of the data assuming the background model everywhere in the image we obtain a product restricted to the support of the object:

$$\frac{P(X \mid \theta)}{P(X \mid \text{bgd})} = \prod_{x \in S(\theta)} \prod_e \left(\frac{p(x;\theta)}{p_{e,\text{bgd}}} \right)^{X_e(x)} \times \left(\frac{1 - p(x;\theta)}{1 - p_{e,\text{bgd}}} \right)^{(1 - X_e(x))}. \tag{6}$$

In other words, the conditional independence model allows us to express the likelihood of all the data given an instantiation, up to a constant factor, in terms of a product limited to the data observed on the support.

These ideas are illustrated in Fig. 2. The probability array for horizontal edges of one polarity is shown for class '2' is shown in (A), with two example subarrays—parts—in (B). For example '2'-s we show the original images in (C). In (D) we show the instantiations in terms of the shifts v_i pointing from the original reference point y_i to the new location z_i. The overlaps of the shifted parts, the function $\mathcal{I}(x)$, are also shown in (D) where darker regions are covered by more parts. The instantiated probability maps $p_e(x;\theta)$, $x \in L$, for the two instantiations, computed using the patchwork operation (3), are shown in (E). Note how a combination of shifts of local models can accommodate a variety of deformations of the original probability array .

2.3. *The Geometric Component, Mixture Models and Classification*

We assume the geometric component, namely the distribution on instantiations $\theta = (r, \mathbf{v})$ has a density $f(\theta)$ which is the product of a joint Gaussian density $g(\mathbf{v}) = g(v_1, \ldots, v_n)$ on the shifts with 0 means, and a uniform distribution on the location r. The conditional distribution on instantiations given the observations, also called the posterior, is then proportional to $P(X \mid \theta) f(\theta)$.

One POP model may not be sufficient to describe the population of a given class. For example there are qualitative differences in shape between different instances of the digit 7 that can not be accommodated through local deformations of the parts. We thus model each class $c = 1, \ldots, C$ as a mixture of M_c POP models:

$$P_c(X) = \sum_{m=1}^{M_c} P_c(m) \int P_{c,m}(X \mid \theta) f_{c,m}(\theta) d\theta, \tag{7}$$

of POP models $P_{c,m}(\cdot \mid \theta)$, $m = 1, \ldots, M_c$, each with a different distribution $f_{c,m}$ on instantiations.

For images of isolated objects that are properly centered, we assume $r = 0$ and classification reduces to maximizing

$$\hat{Y} = \operatorname*{argmax}_c \max_{1 \leq m \leq M_c} \max_{\mathbf{v}} P_{c,m}(X \mid \theta = (0, \mathbf{v})) g_{c,m}(\mathbf{v}). \tag{8}$$

In other words, assuming a uniform prior on classes, classification is obtained by taking the class label from the maximum a-posteriori on class model and instantiation.

2.4. *Modeling Object Configurations*

An object configuration consists of a list of object classes and subclass clusters (c_i, m_i) and their instantiations θ_i.

$$\mathbf{I} = (c_i, m_i, \theta_i)_{i=1,\ldots,k}.$$

Let S_i denote $S_{c_i,m_i}(\theta_i)$. The edge data of the entire image conditional on such an interpretation is modeled by composing the individual data models on the union of the supports $S = \cup_i S_i$. On the complement S^c, the edges are again assumed independent with background probabilities. In the present setting, we assume the objects are ordered according to occlusion, namely an object with

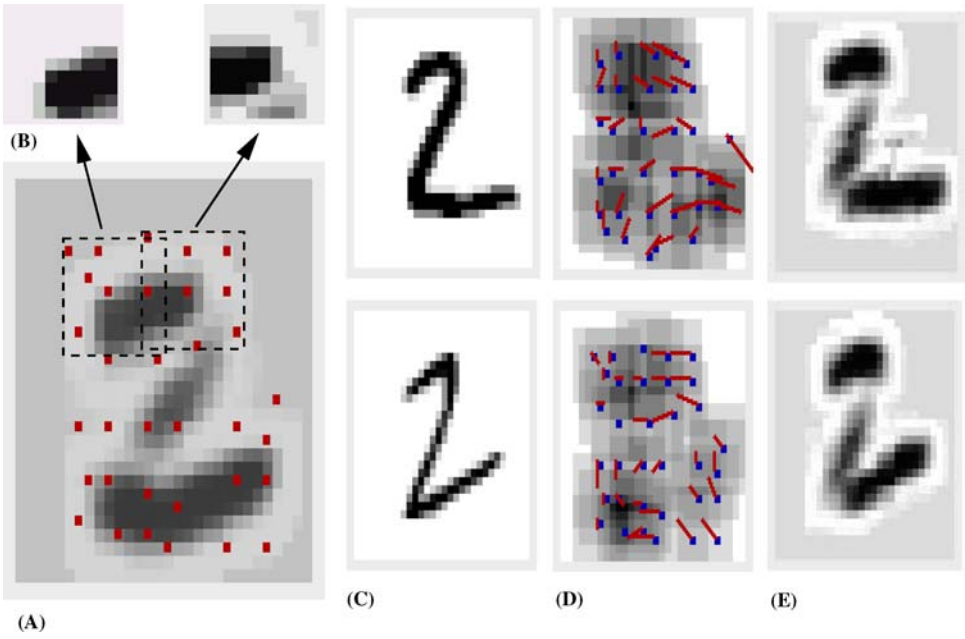

Figure 2. (A) The probability array for horizontal edges of one polarity for the class '2' (dark corresponds to high probability). In red are the reference points. (B) Two subwindows of the probability array centered at two of the reference points. (C) Two images of a '2'. (D) The instantiations: the shifts of the reference points denoted by red arrows together with the support of the subwindows—darker pixels are covered by more subwindows. (E) The probability array on horizontal edges determined by the patchwork operation and the shifts given in (D).

higher index can not occlude and object with lower index. Defining $T_i = \cup_{j=1}^{i} S_j$, to be the support of the first i objects, one expects to observe the data for object i only on the *visible support* $S_i \setminus T_{i-1}$. The ratio of the likelihood of the data given the interpretation to the likelihood given background is:

$$\frac{P(X \mid \mathbf{I})}{P(X \mid \text{bgd})} = \prod_{i=1}^{k} \prod_{e} \prod_{x \in S_i \setminus T_{i-1}} \left(\frac{p_{i,e}(x;\theta_i)}{p_{e,\text{bgd}}} \right)^{X_e(x)} \times \left(\frac{1 - p_{i,e}(x;\theta_i)}{1 - p_{e,\text{bgd}}} \right)^{1-X_e(x)}, \quad (9)$$

where $p_{i,e} = p_{c_i,m_i,e}(x)$. Again the likelihood of an interpretation can be computed up to a constant on the union of the supports of the constituent objects.

We assume that, conditional on the location components r_i of the instantiations, the displacements $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are independent; however, given k objects in the image, there is some joint prior distribution $h(r_1, \ldots, r_k)$ on locations. The posterior on interpretations with k objects is then given by

$$P(\mathbf{I} \mid X) \propto h(r_1, \ldots, r_k) \prod_{i=1}^{k} g_{c_i,m_i}(\mathbf{v}_i) \cdot \frac{P(X \mid \mathbf{I})}{P(X \mid \text{bgd})}. \quad (10)$$

The goal then is to find the interpretation of highest posterior. It is straightforward to extend this model when the number of objects is unknown.

As an illustration, we show in Fig. 3 two competing interpretations of a configuration of two horses detected in an image. Since these supports overlap we need to compare the likelihood of two interpretations: one which puts the left horse in front and the other which puts the right horse in front. These two interpretations involve a different ordering of the objects, and hence a different data model in Eq. (9). In (B) the left horse is assumed in front, with support in red, and the visible support of the right horse is in blue. Putting the right horse in front (C) yields a higher likelihood. The visible support of the left horse is in red, much of the back part is removed. Note that the supports are defined in terms of the edge data and therefore do not cover the entire object, rather the areas where edges may occur on or around objects.

3. Computation

3.1. Classification

For classification we assume $r = 0$. Computing the global maximum in Eq. (8) is difficult due to the inner maximization over $\mathbf{v}$. This is approximated using one of the following two procedures.

Iterative Maximization. Initialize $v_{i,0} = 0, i = 1, \ldots, n$. Choose a small neighborhood N of the origin. At step t loop through the reference points. For each i, fix all other points at their current shift:

$$v_{j,t+1},\ j < i, \text{ and } v_{j,t},\ j > i.$$

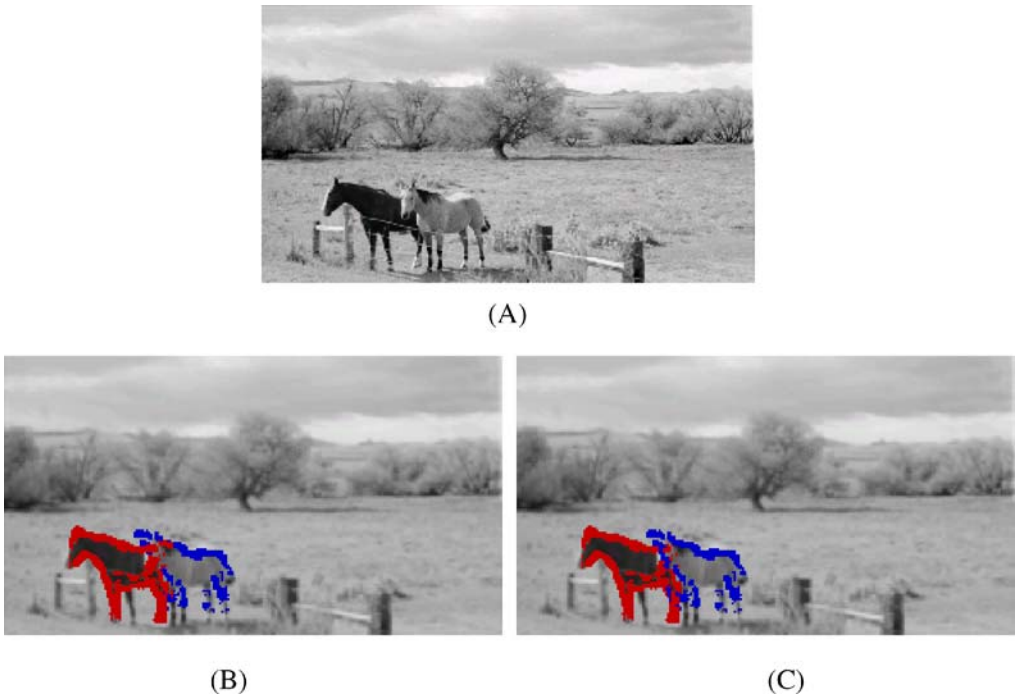

Figure 3. (A) Original image. (B) Wrong interpretation left horse in front of right horse. In red the support of the left horse and in blue the visible support of the right horse. (C) Correct interpretation. In blue the support of the left horse and in red the visible support of the right horse. separately.

For each $v \in v_{i,t} + N$ place the part Q_i at $y_i + v$ and recompute (6). This implies recomputing the patchwork (3) only at points covered by $(y_i + v_{i,t} + W) \cup (y_i + v + W)$. Then multiply by the instantiation distribution $g_{c,m}(v_{1,t+1}, \ldots, v_{i-1,t+1}, v, v_{i+1,t}, \ldots, v_{n,t})$ to obtain the posterior on the proposed instantiation (modulo a constant). Set $v_{i,t+1}$ to be the shift at which the largest posterior is found. After a full loop through all points this procedure is repeated for a small number of iterations.

Independent Maximization. A coarser approximation consists of choosing $v_{i,t+1}$ by maximizing the likelihood ratio of the model Q_i to background over points in $v_{i,t} + N$, *ignoring* all the other parts and the patchwork operation, and iterating several times. The full patchwork and $g_{c,m}(\mathbf{v})$, are computed *only at the end.* This is much faster to compute, and proves to be a very good approximation, if there is not much clutter in the neighborhood of the object. We return to this issue in the experimental section.

Recall that an outside loop over classes $c = 1, \ldots, C$ and cluster labels $m = 1, \ldots, M_c$ is needed to complete the classification.

3.2. Detection

To detect instantiations of a particular class c, we loop over locations $r \in L$ and compute

$$J(r) = \max_{1 \le m \le M_c} \max_{\mathbf{v}} P_{c,m}(X \mid r, \mathbf{v}) g_{c,m}(\mathbf{v}),$$

using one of the above two methods for the maximization over $\mathbf{v}$. Denote the values at which the maximum is attained as $m(r), \mathbf{v}(r)$. We declare a detection at r if $J(r) > \tau_c$ for some predetermined threshold. Each such detection comes with an associated cluster label $m(r)$ and the instantiation $\theta(r) = (r, \mathbf{v}(r))$.

3.3. Multi-Object Configurations

Here we perform the preceding computation for each of the possible object classes $c = 1, \ldots, C$, with conservative thresholds τ_c. This yields a set of candidate class detections $\mathcal{D}$, each with a class label c, a cluster label m and an instantiation θ. Assuming we know that there are k objects in the image, our task is to extract an *ordered* sequence of k elements from $\mathcal{D}$ which maximize Eq. (10). In most cases, due to the combinatorial explosion of possible configurations, it is essentially impossible to find the true maximum. Various greedy iterations can be designed to find a local maximum, however in the specific setting of zipcodes, due to the linear nature of the configuration it is possible under certain assumptions to find the global maximum using dynamic programming. This is described in detail in Section 5.2.

3.4. Pruning the Computation

In all three settings described above, one has to loop through quite a number of maximizations of POP models.

A massive reduction in the number of such maximizations can be obtained using a variety of coarse approximations that perform very fast tests to determine if the candidate POP model has any chance to have posterior above threshold. Then, for only a small subset of all candidate locations, classes and class clusters, is the iterative maximization actually performed. Here we describe a very simple pruning mechanism, which is used extensively in the face and zipcode experiments.

For a given class cluster pair c, m let $\tilde{p}_{c,m,e}$ be one rigid probability model (see 2.1), with no hidden shift variables, estimated on the entire reference grid. The training images are simply stacked up on the reference grid and the frequency of each edge type at each location provides the estimated marginal probability. Because no hidden deformation variables were used in training the marginal probabilities account for the geometric variability as well, and the conditional independence assumption is much less plausible. Now, choose a sample $B_{c,m}$ of edge/location pairs $(z, e) \in S_{c,m}(0)$ from the model support, in such a way that two elements in B are separated in location by d pixels. Since the features are now separated by some distance, it is more reasonable to assume that conditional on the presence of an image from this class cluster at location r, the variables $X_e(z + r)$ for $(z, e) \in B_{c,m}$ are independent with probabilities $\tilde{p}_{c,m,e}(z)$. Let

$$T_{c,m}(r) = \sum_{(e,z)\in B_{c,m}} X_e(z + r).$$

Assuming independence we can write the mean and standard deviation of $T_{c,m}(r)$ as

$$\mu = \sum_{(e,z)\in B_{c,m}} \tilde{p}_{c,m,e}, \quad \sigma^2 = \sum_{(e,z)\in B_{c,m}} \tilde{p}_{c,m,e}(1 - \tilde{p}_{c,m,e}).$$

Since $T_{c,m}$ is approximately Gaussian on the cluster population set a conservative threshold $t_{c,m} = \mu - 3\sigma$, and reject any location r for which $T_{c,m}(r) < t_{c,m}$. Computing the statistic $T_{c,m}(r)$ is just a summation of several tens of binary variables and is very fast. Typically this pruning eliminates over 95% of the locations for each class cluster.

4. Training a POP Model

It is difficult to simultaneously estimate the full probability array p_e and the geometric distribution $g(\mathbf{v})$, in large part due to the unobserved instantiation parameters. One example can be found in Allassonnière et al. (2006) for a related type of deformable model, using the EM algorithm, but the computation is very intensive, and typically one can not carry out the full integration needed in the expectation step. Here we describe an approximate estimation procedure which is motivated by the structure of the model, is very efficient and yields excellent results.

We assume all the training data is located at the origin (i.e. $r = 0$). The idea is to estimate each part *separately* assuming a *rigid* model for the data with instantiations, i.e. shifts, limited to a square region V around the origin. Pick a point x and assume a priori that the support $S(0)$ of this rigid model is given by $S(0) = x + W$. The instantiation of each training point is unobserved so that estimation of the probability array $Q = (p_e(z))_{z\in x+W}$, and the distribution $\pi(v)$ on shifts $v \in V$, is performed using an EM procedure as detailed below. The mean of π then yields a reference point $y \in x + V$. These estimates are only affected by data in the neighborhood of x, so that at different points x different probability arrays are obtained. For this constrained estimation problem, the EM algorithm can be performed *in full*, since the state space of the unobserved variable—the set of possible shifts—is not very large.

The procedure is carried out at each point of a regular grid $x_i, i = 1, \ldots, n$ yielding probability arrays Q_i—the parts—and reference points $y_i \in x_i + V$. Using the patchwork operation in Eq. (3), where each part is placed at the reference point, we obtain an estimate of the full probability array, also denoted the mean global model. Estimation time of a POP model for several 10's of training data is on the order of several seconds.

4.1. Training one Part

Since we are dealing with one part around one point $\tilde{x}$ we remove subscripts and to further simplify notation we assume only one binary feature type $X(x)$ at each pixel. The data observed in each training image is modeled in terms of a probability array $Q = (p(s))_{s\in W}$ on a window W, placed at an unobserved random location $z = \tilde{x} + v$, where v is distributed according to an unknown distribution π defined on the set V. A background probability p_{bgd}, which we assume known, is assigned everywhere else. Thus only data around $\tilde{x}$ affects the estimates. Given the instantiation $\theta = v$, using the same ratio trick as in Eq. (6), we get

$$\begin{aligned} P(X \mid v, Q) &= C \cdot \prod_{x\in\tilde{x}+v+W} \left(\frac{p(x - \tilde{x} - v)}{p_{\text{bgd}}}\right)^{X(x)} \\ &\quad \times \left(\frac{1 - p(x - \tilde{x} - v)}{1 - p_{\text{bgd}}}\right)^{1-X(x)} \\ &= C \cdot \prod_{s\in W} \left(\frac{p(s)}{p_{\text{bgd}}}\right)^{X(\tilde{x}+v+s)} \\ &\quad \times \left(\frac{1 - p(s)}{1 - p_{\text{bgd}}}\right)^{1-X(\tilde{x}+v+s)} \end{aligned} \tag{11}$$

where C does not depend on the unknown parameters. Note that since we are only translating the models, one can either translate the probability map or translate the

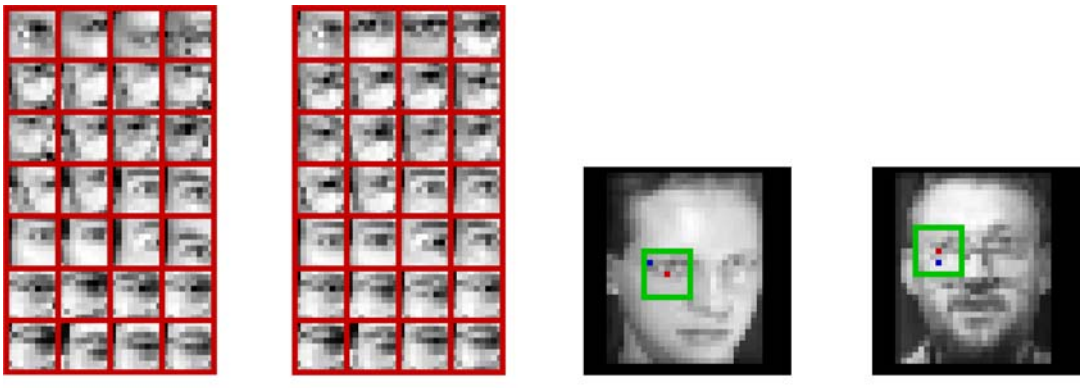

Figure 4. (A) Training subwindows at start point. (B) Subwindows centered at most likely shift for each image. (C) For two training images, the location of the start point in blue, and the subwindow around the shifted point.

observations. This becomes more difficult for more complex instantiations.

The log-likelihood of a set of m training images $X^{(j)}$ with *observed* instantiations $v^{(j)}$ has a unique maximizer at

$$\hat{\pi}(v) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}_{\{v^{(j)}=v\}}, \quad v \in V$$

$$\hat{p}(s) = \frac{1}{m}\sum_{j=1}^{m} X^{(j)}\big(\tilde{x} + v^{(j)} + s\big), \quad s \in W.$$

The corresponding reference point is set as $y = \tilde{x} + \sum_{v\in V} v\hat{\pi}(v)$.

Since we do not observe $v^{(j)}$, the likelihood of the observed data $X^{(j)}$ has the form

$$P(X \mid Q, \pi) = \sum_{v\in V} \pi(v) P(X \mid v, Q).$$

We are now in the classical setting of estimating the parameters of a mixture distribution. A unique feature of the present setting is that the distributions of the components of the mixture are 'shifts' of each other, thus more data can be pooled to estimate the parameters. The standard method for finding a local maximum of the likelihood is the EM algorithm, see Dempster et al. (1977), which involves generating iterative estimates Q^{ℓ}, π^{ℓ} as follows:

1. Initialize

$$p^{(0)}(s) = \frac{1}{m}\sum_{j} X^{(j)}(\tilde{x} + s), \quad s \in W,$$

$$\pi^{(0)}(v) = 1/|V|, \quad v \in V.$$

2. For each training point j and $v \in V$, compute

$$P\big(v \mid X^{(j)}, Q^{(\ell)}, \pi^{(\ell)}\big) = \frac{P(X^{(j)} \mid v, Q^{(\ell)})\pi^{(\ell)}(v)}{\sum_{z'} P(X^{(j)} \mid v', Q^{(\ell)})\pi^{(\ell)}(v')},$$

using (11) in the numerator and denominator.

3. Compute new estimates

$$\pi^{(\ell+1)}(v) = \frac{1}{m}\sum_{j} P\big(v|X^{(j)}, Q^{(\ell)}, \pi^{(\ell)}\big).$$

$$p^{(\ell+1)}(s) = \frac{1}{m}\sum_{v\in V}\sum_{j} P\big(v|X^{(j)}, Q^{(\ell)}, \pi^{(\ell)}\big) \times X^{(j)}(\tilde{x} + v + s), \; s \in W$$

$\ell \to \ell + 1$, goto 2.

After a small number of iterations the probabilities $p^{(\ell)}(s), s \in W$ stabilize and are recorded. The reference point is set as $y = \tilde{x} + \sum_{v\in V} \hat{v}\pi^{(\ell)}(v)$. If the estimated array is too close to a homogeneous map (p_s are all very similar), we eliminate the associated part. That is why in the various figures one finds reference points only near the object support although the start points are regularly spaced on the entire grid.

These ideas are illustrated in Fig. 4. On the left we show for a sample of face images, the subimages of size 9×9 around a certain point $\tilde{x}$ which is in the neighborhood of the left eye. On the right we show the 9×9 windows around $\tilde{x} + v_*^{(j)}$ for each image, where $v_*^{(j)}$ is the mode of the conditional distribution $P(v|X^{(j)}, Q^{(\ell)}, \pi^{(\ell)})$, which after convergence is typically peaked at one particular shift. Note how the eyes are now located in the same place in the subwindow. Furthermore for two training images the location of the start point is shown in blue and the subwindow around the most likely location is shown in green.

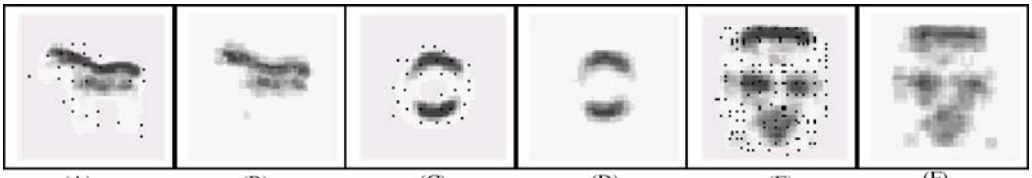

Figure 5. Probability arrays for horizontal edge type of one polarity. High probability areas are darker. Reference points in red. (A) Mean global model for a horse model. (B) Mean global model without iterations. (C),(D) Same the '0' digit in MNIST. (E)(F) Same for face model.

4.2. Estimating the Distribution on Instantiations

We estimate the joint distribution $g(\mathbf{v})$ on shifts as follows. Re-loop through the training data with the estimated POP model. For each example compute the optimal instantiation $\theta_*^{(j)} = (0, \mathbf{v}_*^{(j)})$, as detailed in Section 3.1 above. Use this sample to estimate the full covariance matrix for the joint Gaussian centered at zero. We use a Bayesian estimate with an inverse-Wishart conjugate prior. Specifically let $\mathcal{C}(x, x') = bq(|x - x'|/s)$ define a positive definite symmetric kernel, for some function q. Let $C_{j,j'} = \mathcal{C}(y_j, y_{j'})$ be the positive definite matrix obtained by evaluating the kernel at all pairs of reference points of the model. We assume an inverse-Wishart prior with matrix parameter M and scale parameter a on the joint covariance matrix of the shifts. If $\hat{\Sigma}$ denotes the empirical covariance matrix computed from the m samples, the Bayesian estimate is simply a weighted average of $\hat{\Sigma}$ and C:

$$\tilde{\Sigma} = \frac{m\hat{\Sigma} + aC}{m + a}.$$

We take q to be e^{-x^2}. The parameter s reflects our prior assumptions on the degree of dependence between the shifts at the different reference points, we use $s = 1$ (in pixel units). The smaller it is the larger the dependence. The parameter b is a scale parameter reflecting prior assumptions on the range of variance of the shifts, we use $b = 2$. Finally a is the weight assigned to the prior, and should be proportional to the dimensionality of the problem, i.e. the number of reference points (see Allassonnière et al., 2006 for details). As will be seen in the experimental results, the distribution on shifts has a small positive effect on the error rates for isolated digits, but plays an important role in improving recognition rates for zipcodes and detection rates for faces.

4.3. Mean Global Model (MGM) and Inter-Part Consistency

The final estimate of the probability map is the mean global model, obtained by applying the patchwork operation with the estimated parts Q_i at the reference points y_i. The mean global model for horses, the zero character and faces are shown in Fig. 5. On the left for one type of edge (horizontal) is the result with the EM iterations. On the right is the result with no iterations. Each part is obtained by taking the initial $p^{(0)}(s)$ obtained with no shifting of the windows, namely the frequencies of the edges at pixel s.

The alignment generated by the training procedure produces more concentrated models where local variability has been factored out. As shown in the results section, this leads to significant improvements in performance since the likelihood contrasts become sharper.

Note that even though the local models are trained separately, placing the parts Q_i at the estimated reference points y_i yields a consistent model in the sense that the distributions induced on the overlap regions by several overlapping parts are very similar. Had this not been the case, the model would appear blurred and diffuse.

4.4. Learning Mixture Models

Our goal is to enable any model we develop to evolve as more data gets processed. The idea is to envisage each class as a *mixture* of POP models, and have the number of mixture components and the parameters of the mixture components evolve as additional data is introduced. We describe a simple approach that has performed remarkably well.

Given data from one class, we train initially on a small training set $\mathcal{T}_0$ of size $M_{\max}$ and produce one POP model $\mathcal{P}_0$ from this dataset. For this model, we compute the mean and standard deviation μ_0, σ_0 of the log-likelihoods $\ell_0(X)$, $X \in \mathcal{T}_0$. Any 'problematic' data point in $\mathcal{T}_0$ with log-likelihood one standard deviation below the mean—$\ell_0(X) < \mu_0 - \sigma_0$—is added to a new list $\mathcal{T}_1$. Now as additional data points X arrive (not from the original set $\mathcal{T}_0$), we evaluate $\ell_0(X)$ and add to $\mathcal{T}_1$ only those for which $\ell_0(X) < \mu_0 - \sigma_0$. Once the size of this list $|\mathcal{T}_1| = M_{\max}$ estimate a POP model $\mathcal{P}_1$ from this data set and estimate μ_1, σ_1. All points in $\mathcal{T}_1$ are already below threshold for the model $\mathcal{P}_0$. Those that also fall below threshold for $\mathcal{P}_1$ are used to start a new list $\mathcal{T}_2$. New data points that fall below threshold on *all* existing models (in this case $\mathcal{P}_0$, $\mathcal{P}_1$), are

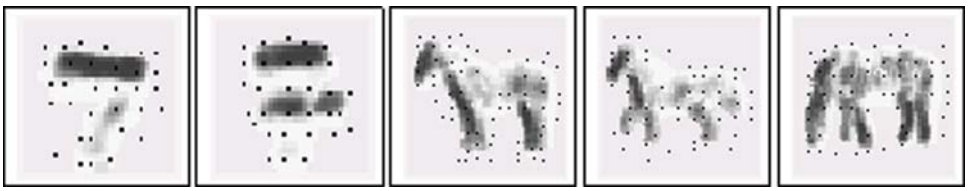

Figure 6. Probability arrays for horizontal edges for two seven clusters and three horse clusters. Dots show the reference points.

added to $\mathcal{T}_2$, and so on. In this manner additional models are added once a sufficient number of points has accumulated whose likelihood is below threshold for each of the current models.

Clearly as the number of models grow the rate at which the 'bad' set grows is slower and slower. Effectively we are training with only a very small subset of the data points, since already after the first set most points are above threshold.

In Fig. 6 we show two of the models estimated for the class 7, and three of the models estimated for horses using the database in Borenstein (2006). It is encouraging to see that the second seven cluster has picked up sevens that are qualitatively different from those represented by the first, i.e. European sevens with a cross bar. Among the horses the clustering process has picked up discretely different poses.

5. Experimental Results

In this section we illustrate the usefulness of the proposed data models and associated training procedure in a number of applications. Due to space limitations not all details of the implementation can be provided.

5.1. *Classifying Segmented Digits*

We present detailed experiments on the MNIST data set to explore the dependence of the algorithm on several of the model parameters. The default parameters are defined in Table 1. In this setting we use the independent maximization procedure explained in Section 3.1. We note that the error rates are estimated on a test set of size 10,000 and the corresponding standard error for rates under 3%, is .17%.

Table 1. Default setting of algorithm parameters.

Edge spreading window – 3×3
Part size (W) – 9×9
Points x_i for part estimation every 4 pixels.
Neighborhood of shifts in training (V) – 11×11
Number of iterations in iterative maximization – 5.
Number of iterations in EM – 10
Number of data points at which additional model is estimated – $M_{\max} = 10$

5.1.1. Computing the Affine Component—Normalization. There is quite a wide range of variability in the MNIST dataset in terms of the affine pose, in particular object slant. This is not easy to incorporate in the POP setting since large slants create significant changes in orientation which are not accommodated by simple shifts of the parts. In the context of isolated digits this is easily addressed by a simple slant correction and scaling procedure. This preprocessing step depends heavily on having cleanly segmented data, it is sensitive to noise and clutter, and is viewed as a computational shortcut. When dealing with more complex images with several adjacent objects, such as zipcodes, reliable presegmentation is not a stable option.

Instead define a discrete set of affine maps covering the desired range. After training the mixture model with the normalized data, take the training images assigned to a cluster, apply one of the affine maps, and estimate a new POP model. This is done for each model and each of the affine maps. If N_A affine maps are used and there are M_c clusters for class c on the normalized data, we end up with $N_A \cdot M_c$ clusters in the mixture model for each class. The price for lack of cleanly segmented objects, is a larger number of components in the mixture model for each class. For zipcodes we used 5 scales at 0.75, 1., 1.2, 1.5, 1.8 relative to the scale of the training set images, and 3 slants: $x = y + sx$, $s = -.4, 0, .4$.

5.1.2. Error Rates as Function of Training Set Size. The first question of interest is the evolution of the error rates with the training set size. This is summarized in Table 2. The classification results are for a test set of 10,000 where the margin of error is about .17%. The error rate starts at 6% with 100 training data, i.e. 10 per class with 1 cluster per class, to 1.5% with 5000 training data, i.e. 500 per class with on average 8 models per class. Note that this means that the models were estimated with about 80 samples per class of the 500 available. Ignoring the joint distribution on shifts the rate is 1.85%. In this experiment the estimated joint distribution f on shifts seems to have a small effect.

Table 2. Right: Classification rates as function of number of training data per class. Middle column indicates number of clusters found in each class with the sequential clustering algorithm. We report error rates with the prior on θ and without, as well as the best rates achieved with SVM's on the same edge features (Using a quadratic kernel).

Training data per class	Avg. clusters per class	Error rate with f	Error rate without f	SVM error rate
10	1	6.5	6.05	12.61
30	2.6	3	3	6.17
50	3.4	2.46	2.58	4.18
100	4.1	1.96	2.14	3.02
500	8	1.52	1.85	1.47

5.1.3. Stability With Respect to the Training Set. The models reported in Table 2 were trained with the first (100,300,1000,5000) training examples of the MNIST training set. Of interest is the stability of the results with respect to variations in the training set. For sample size 300 - 30 per class—we trained 25 classifiers on disjoint subsets of the training set. The mean error rate was 3.1% with standard deviation .3%. This is an encouraging finding. Despite the very small training set size, the variance of the final classification rate is very small.

5.1.4. Comparison to Non-Parametric Classifiers. For the smaller size datasets 10–100 per class, the results are far better than anything we were able to achieve with non-parametric classification methods such as SVM's or boosted randomized decision trees on the same edge features, see last column of Table 2. The 3% error rate reported for 30 examples per class, and the 2% error rate reported for 100 examples per class are competitive with many algorithms listed in LeCun (2004) that have been trained on 6000 per class. The results become indistinguishable as the sample sizes increase.

5.1.5. Non-Sequential Clustering. The clustering algorithm described in Section 4.4 is appealing because of its sequential nature and the ability to update the model as more data is observed. However for optimal results it may be preferable to estimate the clusters simultaneously from all the available data. This is difficult to do in our context because of the fact that the instantiation parameter θ is unobserved. However using a coarse approximation to the POP model in terms of a fixed library of local parts as proposed in Bernstein and Amit (2005) we can implement an EM type algorithm to estimate a predefined number of clusters. Then using the data assigned to each such cluster we estimate a POP model. This improves the results for the larger training set sizes as summarized in Table 3.

With only 1000 training points per class, using likelihood ratio based classification with no discrimination boundaries, we achieve a state of the art error rate of 0.8% going up to 0.68% with the full training set.

Table 3. Classification results with non-sequential clustering using all available data.

Training data	No. of Clusters	POP Error rate
500	20	1.11
1000	30	**.8**
6000	80	**.68**

5.1.6. Computation Time. With pruning of the form described in Section 3.4 the computation time is about .001 seconds per image per cluster on a Pentium IV 3 Ghz, for example 100 images per second with five clusters per class. Thus as the number of clusters grows classification slows down. One remedy is to use simpler models to detect the top 2,3 classes. For example with the models made from 30 examples per class, with 2–3 clusters per class the top 3 classes are correctly identified for 99.6% of the data. Using the simpler models in an initial run with some confidence threshold, the more intense computations using more complex models can be performed just on 'uncertain' examples. Ultimately this classification method should be incorporated in a comprehensive coarse-to-fine computation.

5.1.7. Varying Parameter Settings. At 100 examples per class, we experimented with some of the parameter settings. We summarize the results in Table 4 where the modified parameter value is indicated all others being at default value.

First we show the importance of performing the maximization on the shifts v_i. if the likelihoods are computed directly at $v_i = 0$, i.e. placing each part Q_i at the original reference point y_i the error rate increases to 10.1%. The classification is highly dependent on estimating the deformation variable.

It is also possible to estimate the model with no shifting in training This reduces to a straightforward estimation of marginal probabilities of the edges at each location. For classification we still maximize over the shifts. This yields cruder models, (see Fig. 5) with higher error rates—3% (instead of 2%). This is significantly lower than the rate obtained with the original model (the std. on error is .17%).

We also tried increasing the value of $M_{\max}$ which determines the number of points needed to estimate a new POP model, the number of models per class dropped from 4.1 to 2.6. This led to a very slight decrease in performance. It is interesting that, in this setting, there was a somewhat larger drop in classification rate when the distribution on deformations is ignored. With more clusters, part of the geometric variation is covered by the different models, and the constraints on the deformation captured by the distribution f are redundant.

Table 4. Comparing error rate with default parameters to individual parameter changes.

Varied Parameter	Default	No opt.	No EM iters.	$M_{\max} = 40$ w/wo f	$W = 6$	$W = 12$
Error rates	1.96	10.1	3%	2.04/2.35	2.79	2.44

5.2. *Reading Zipcodes*

The goal here is to perform a likelihood based labeling of the zipcode avoiding any preprocessing or presegmentation. Here we are not interested in a highly dedicated algorithm for reading zipcodes, rather this setting is viewed as a simple context where the generic ideas on multiple-object configurations can be explored. The digit models are trained from the isolated and segmented MNIST dataset. Since the zipcode digits appear at widely different scales (at least 2:1)—even in the same zipcode, instead of estimating one POP model for each class cluster, we estimate a number of models where all the data in the cluster is simultaneously scaled or slanted, using 5 scales and 3 slants as described in Section 5.1. We experiment with varying size training sets: 100, 500, and 1500 per class. The number of clusters in each case is 5, 15 and 60.

An initial scan results in a set $\mathcal{D}$ of candidate detections for all 10 digits, using very conservative thresholds, see Section 3.3. Typically $\mathcal{D}$ contains 2-3 hundred instantiated detections with extensive overlaps of their supports. At this stage the instantiations are computed using the more efficient independent maximization method. For some example detections and their support on a sample zipcode see Fig. 7. Note that due to the many different scales and slants at which the digits appear there can be many detections on a particular part of the zipcode that 'make sense' unless the full context of the interpretation is taken into account.

Since the objects have to be arranged in a linear fashion, a simple prior is defined in terms of hard constraints on the locations of consecutive pairs of detections:

$$h(r_1, \ldots, r_5) = \prod_{i=1}^{4} c(r_i, r_{i+1}), \tag{12}$$

where $c(r, r')$ constrains $r'_x > r_x$ and $|r'_y - r_y| < \delta$. The goal is to maximize (10) over all candidate sequences of length 5 from $\mathcal{D}$. Assume that in a correct interpretation where the objects are ordered from left to right only supports of consecutive objects can intersect: $S_i \cap S_{i+2} = \emptyset$, $i = 1, 2, 3$. Then the log of the expression in (9) becomes a sum on functions of consecutive pairs of the form

$$\Psi(i-1, i, X) = \sum_{e} \sum_{x \in S_i \setminus S_{i-1}} X_e(x) \log\left(\frac{p_{i,e}(x;\theta_i)}{p_{e,\text{bgd}}}\right) + (1 - X_e(x)) \log\left(\frac{1 - p_{i,e}(x;\theta_i)}{1 - p_{e,\text{bgd}}}\right),$$

where we set $S_0 = \emptyset$. The log-posterior on a zipcode interpretation ordered from left to right has the form

$$L(\mathbf{I} \mid X) = \sum_{i=1}^{5} \Psi(i-1, i, X) + \log c(r_{i-1}, r_i),$$

which can easily be optimized with dynamic programming. Furthermore it is possible with little additional computational cost to obtain the top K interpretations, see Fig. 8.

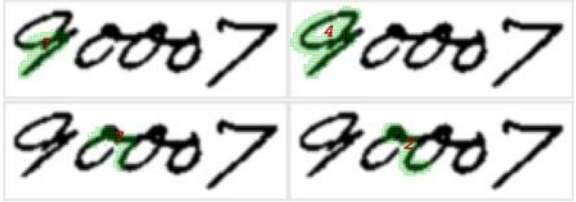

Figure 7. The support of several detections on a zipcode.

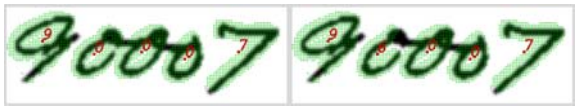

Figure 8. Interpretations with supports. Left: optimal interpretation—90007. Right: Second best—96007.

5.2.1. Reprocessing Instantiations. Recall that the instantiations are computed with the coarser independent maximization method which can lead to inaccuracies in the presence of clutter. At little additional cost it is possible to recompute the instantiations of selected objects in the top K interpretations. Recall that each interpretation is an ordered sequence of 5 instantiations. We find the index in the sequence where the label of the top interpretation differs from the second best. In all but a handful of cases, there is only *one* such index, as is the case for example in Fig. 8. For all interpretations among the top K which differ from the top interpretation in the problematic index, we recompute the instantiation of the object class at that index using the iterative maximization method described in Section 3.1. After this is done, the total log-posterior of the interpretations is recomputed

Table 5. Zipcode recognition rates and computation times, as function of size of training data with and without the reprocessing step.

No. ex. per class	No. of clusters	W.o reprocessing	With reprocessing
100	75	74.5% (4s. per zip)	77.3% (5.7 s. per zip)
500	300	84.4% (5.8 per zip)	87.2% (7.7 s. per zip)
1500	900	85.3 % (8.8 per zip)	88.7% (11s. per zip)

Table 6. Comparison of zipcode reading rates.

Author	**n**	Correct at 0 rej.	% Correct	% Rej
Ha et al. (1998)	436	85%	97%	34%
Palumbo and Srihari (1996)	1566		96.5%	32%
Wang (1998)	1000	72%	95.4%	43%
POP models	1000	88.7%	96.5%	30%

and the highest one is chosen. As shown in Table 5 the classification result improves by about 3% in all cases.

We tested the results on a set of 1000 zipcodes from the CEDAR data base. No segmentation or preprocessing of any kind is performed. We obtain a correct zipcode recognition rate of 88.7% using the models trained on 1500 examples per class, and the instantiation reprocessing procedure. For 94% of the zipcodes the correct labeling was among the top 10. Furthermore using a simple rejection criterion comparing the likelihoods of the top two interpretations, we get 96.5% correct with 30% rejection. Computation time on a Pentium IV 3Ghz is 11 seconds per zipcode. In Table 5 we summarize the results for different training set sizes and different computational regimes, including the computation time per zipcode.

There is not much literature on reading zipcodes in recent years. However, comparing to the literature from the mid to late 90's, this initial result is within the range of results obtained by very dedicated algorithms. Some results are presented in Table 6. Note that the training and testing datasets are not the same so it is hard to provide an accurate comparison.

5.3. *Faces*

To verify whether this model is applicable to gray level objects that are not line-drawings we performed a face detection experiment. Using the first 400 images of the Olivetti data set we trained 8 face POP models at .3 of the original scale—on average 10 pixels between the eyes. As in the zipcode problem, to accommodate different scales and rotations we simultaneously scaled the images in each cluster at .27, .3 and .33, and $-10, 0, 10$ degrees to create scaled versions of each model. Thus in total there are $8 \times 3 \times 3$ POP models for faces. In Fig. 5 we have shown the mean global model for one edge type for one of the face models at scale .3.

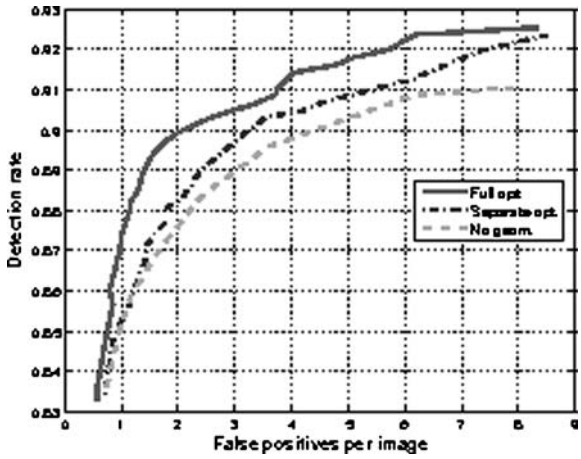

Figure 9. ROC curve for face detection. X axis—number of false positives among 130 images. Y axis, fraction of detected faces. Solid red line: full posterior. Dashed blue line: coarse approximation to optimal instantiation. Dashed cyan line: ignoring distribution on instantiation.

Using an efficient but crude face detector (see Amit, 2002) we obtain candidate windows for testing the POP models. We used very conservative thresholds and no clustering of detections yielding on average several hundred detections per image. These detectors are based on the same edges and can be viewed as very coarse approximations of the POP models. At each candidate window we compute an *adaptive* estimate of the background edge probabilities $p_{e,\mathrm{bgd}}$. Using a likelihood ratio test of the POP models at $\mathbf{v} = 0$ (no shifting) to background for each of the 72 models we pick the best. This does not involve the intensive computation of optimizing the shift of each part in the POP model. Only at the chosen model do we compute the optimal instantiation θ using the *iterative maximization* procedure (see 3.1).

Finally the ratio of the posterior of the fitted POP model to the likelihood under the locally adapted background model is compared with a threshold to decide if the candidate detection is a face or not. Varying this threshold yields a ROC curve (red solid line) presented in Fig. 9. In this figure, we also show the the ROC curve obtained when ignoring the distribution on instantiations (cyan dashed line), which is significantly worse. It is clear that, in the presence of clutter, it is important to properly weight the deformation of the model. Finally we show the ROC curve obtained by maximizing the posterior on instantiations using the independent maximization method. (dashed blue line). Again, due to clutter, the results degrade although computation time is reduced.

We tested on the combined CMU MIT test sets of faces (testA, testB, testC, rotated), excluding a couple of upside-down faces, two profiles and several 'caricature' or line drawing faces leaving 537 'faces' in 160 images. At a false negative rate of 12.3% we have under 1 false positive per image The best we could achieve with

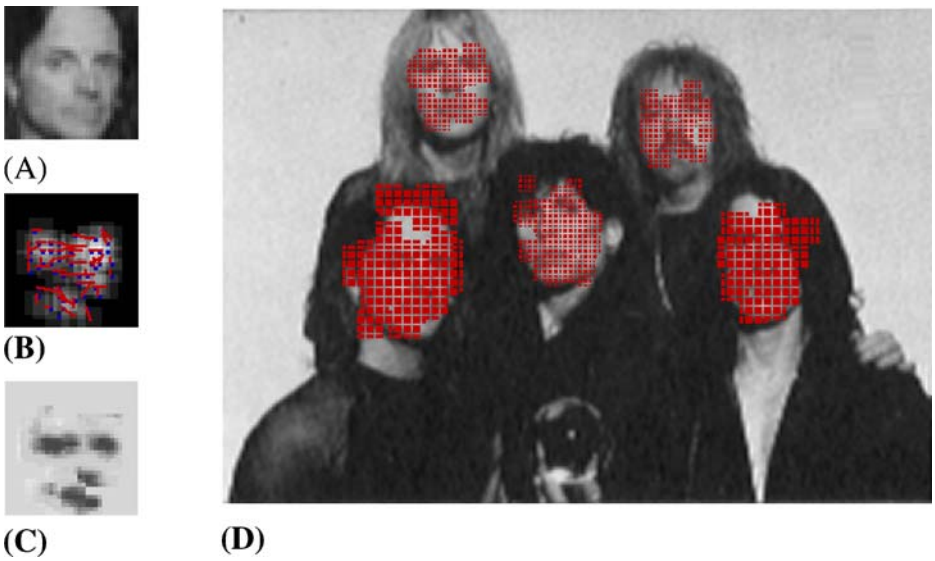

Figure 10. (A) The subimage around a face. (B) The shifts of the reference points relative to the hypothesized center of the detected face. (C) The resulting global POP probability array for horizontal edges. Bottom. Supports of global POP models for all faces in the image.

the original crude face detector at this false negative rate was around 40 false positives per image. Our results are slightly worse than those reported for example in Viola and Jones (2004) or Schneiderman and Kanade (2004). However, all other models have used explicit training with large numbers of faces and massive numbers of background images. The interest here is that the face models are trained with only 400 faces, no background, and yet a simple likelihood ratio test to an adaptive background model has so much power.

In addition to location, scale and rotation, we obtain a full instantiation of the face. As an example in Fig. 10 (A), we show the subimage of a detected face together with the shifts of the reference points (B), and the global POP model for the horizontal edges (C). Note how the deformed probability model is adjusting to the fact that the face is partially rotated. In (D) we show the support computed for each of the faces.

6. Discussion

We have introduced a new class of statistical object models with rather general applicability in a variety of data sets. These models describe the dense oriented edge maps obtained from the gray level data, and assume independence conditional on the instantiation. The advantages of statistical modeling and likelihood based classification have been demonstrated at several levels: (i) robust and efficient estimation of deformable models from small datasets, (ii) easy sequential training of new classes or new class clusters (iii) composability of object models to interpretation models for object configurations.

One inherent drawback of the current models is sensitivity to rotations beyond say $+/-15$ degrees. We allow only shifts of the parts so that when an articulated component of the object undergoes a significant rotation or skew, the probabilities of the edges at each location can no longer be represented as a shift of the original model. Currently this can be accommodated through an additional cluster in the class. This raises an important question regarding the complex tradeoff in terms of memory and computation between the number of clusters and the range of the deformations. This question becomes all the more complex when thinking of extending these ideas to modeling 3d objects from all viewpoints. Extending the range of deformations would involve a method for 'rotating' the models by estimating transition probabilities between edge types as a function of the rotation.

Other questions of interest are the possibility to have parts of different sizes depending on the degree of local variability, as well as data models for original gray level data that take photometric variability into account.

The use of interpretation models has been applied to a limited situation where the objects are arranged linearly. In more complex settings one can only hope to find suboptimal configurations using some iterative methods. It is important to see how far these ideas can be extended because they offer a systematic mechanism for sorting out the arrangement of objects in the image.

Acknowledgment

Y. Amit was supported in part by NSF ITR DMS-0219016.

References

Allassonnière, S., Amit, Y., and Trouvé, A. 2006. Toward a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Stat. Soc.,* to appear.

Amit, Y. 2002. *2d Object Detection and Recognition: Models, Algorithms and Networks*, MIT Press: Cambridge, Mass.

Amit, Y. and Geman, D. 1997. Shape quantization and recognition with randomized trees. *Neural Computation*, 9: 1545–1588.

Amit, Y. and Geman, D. 1999. A computational model for visual selection. *Neural Computation*, 11: 1691–1715.

Amit, Y., Geman, D., and Fan, X. D. 2004. A coarse-to-fine strategy for multi-class shape detection. *IEEE-PAMI*, 26: 1606–1621.

Belongie, S., Malik, J., and Puzicha, S. 2002. Shape matching and object recongition using shape context. *IEEE PAMI*, 24: 509–523.

Bernstein, E. J. and Amit, Y. 2005. Part- based models for object classification and detection, In *CVPR 2005 (2)*.

Borenstein, E. 2006. http://www.dam.brown.edu/people/eranb/.

Borenstein, E., Sharon, E., and S., U. 2004. Combining bottom up and top down segmentation, In *Proceedings CVPRW04*, Vol. 4, IEEE.

Burl, M., Weber, M., and Perona, P. 1998. A probabilistic approach to object recognition using local photometry and global geometry, In *Proc. of the 5th European Conf. on Computer Vision, ECCV 98*, pp. 628–641.

Crandall, D., Felzenszwalb, P., and Huttenlocher, D. 2005. Spatial priors for part-based recognition using statistical models, In *Proceedings CVPR 2005* to appear.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1: 1–22.

Fei-Fei, L., Fergus, R., and Perona, P. 2003. A bayesian approach to unsupervised one-shot learning of object categories, In *Proceedings of the International Conference on Computer Vision*, Vol. 1.

Geman, S., Potter, D. F., and Chi, Z. 2002. Composition systems. *Quarterly of Applied Mathematics*, **LX**: 707–736.

Ha, T. M., Zimmermann, M., and Bunke, H. 1998. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31: 257–272.

Hastie, T. and Simard, P. Y. 1998. Metrics and models for handwritten character recognition. *Statistical Science*.

LeCun, Y. 2004. The mnist database. http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Leibe, B. and Schiele, B. 2003. Interleaved object categorization and segmentation, In *BMVC'03*.

Leung, T., Burl, M., and Perona, P. 1995. Finding faces in cluttered scenes labelled random graph matching, In *Proceedings, 5th Int. Conf. on Comp. Vision*, pp. 637–644.

Liebe, B. and Schiele, B. 2004. Scale invariant object categorization using a scale-adaptative mean-shift search, In *DAGM'04 Annual Pattern Recognition Symposium*, Vol. 3175, pp. 145–153.

Palumbo, P. and Srihari, S. 1996. Postal address reading in real time. *Intr. Jour. of Imaging Science and Technology*.

Rowley, H. A., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. *IEEE Trans. PAMI*, 20: 23–38.

Schneiderman, H. and Kanade, T. 2004. Object detection using the statistics of parts. *Inter. Jour. Comp. Vis.*, 56: 151–177.

Torralba, A., Murphy, K. P., and Freeman, W. T. 2004. Sharing visual features for multiclass and multiview object detection, Technical Report AI-Memo 2004-008, MIT.

Tu, Z. W., Chen, X. R., L., Y. A., and Zhu, S. C. 2004. Image parsing: unifying segmentation, detection and recognition. *Int'l J. of Computer Vision*, to appear.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Viola, P. and Jones, M. J. 2004. Robust real time face detection. *Intl. Jour. Comp. Vis.*, 57: 137–154.

Wang, S. C. 1998. A statistical model for computer recognition of sequences of handwritten digits, with applications to zip codes, PhD thesis, University of Chicago.

Wiskott, L., Fellous, J.-M., Kruger, N., and von der Marlsburg, C. 1997. Face recognition by elastic bunch graph matching. *IEEE Trans. on Patt. Anal. and Mach. Intel.*, 7: 775–779.