Generative Models for Labeling Multi-object Configurations in Images

Yali Amit[1] and Alain Trouvé[2]

[1] Department of Statistics, University of Chicago, Chicago, IL 60637
amit@marx.uchicago.edu
[2] CMLA, ENS-Cachan,
61 Ave du President Wilson, 94235 Cachan cedex, France
trouve@cmla.ens-cachan.fr

Abstract. We propose a generative approach to the problem of labeling images containing configurations of objects from multiple classes. The main building blocks are dense statistical models for individual objects. The models assume conditional independence of binary oriented edge variables conditional on a hidden instantiation parameter, which also determines an *object support*. These models are then be composed to form models for object configurations with various interactions including occlusion. Choosing the optimal configuration is entirely likelihood based and no decision boundaries need to be pre-learned. Training involves estimation of model parameters for each class separately. Both training and classification involve estimation of hidden pose variables which can be computationally intensive. We describe two levels of approximation which facilitate these computations: the Patchwork of Parts (POP) model and the coarse part based models (CPM). A concrete implementation of the approach is illustrated on the problem of reading zip-codes.

1 Introduction

Work in object recognition has focused on two main areas. The first area involves classifying images of segmented objects or images known to contain only one object. The problems are formulated in different ways, sometimes a decision among several classes [3,7], and at times a decision class vs. background [15]. The second area involves the detection of instances of an object class in large images, which may contain any number of these objects or none at all, [24,4,16,27,26]. In [25] several objects are detected simultaneously. These two areas are of course closely related, and raise important issues such as how is photometric and geometric variability handled? How is the background defined? What type of training is used?

There is a rather clear dividing line in the literature between those that emphasize non-parametric discriminative learning of decision boundaries and those that employ parametric modeling of the different object classes. For example in handwritten digit recognition the work in [11,3] involves discriminative learning

J. Ponce et al. (Eds.): Toward Category-Level Object Recognition, LNCS 4170, pp. 362–381, 2006.

using different algorithms, whereas a generative modeling approach is proposed in [23]. For object detection the work in [27] uses large numbers of examples of faces and massive numbers of non-face examples to train a classifier between face and non-face. These ideas are extended in [25]. On the other hand in [17] training is performed on several hundred object examples alone. Although the method there is described as a cascade of classifiers, it is shown in [5] that these can be viewed as approximations to an underlying stochastic model for face images. The approach in [12,15] is also generative.

Yet both detection and classification are in fact reductions of the real goal of labeling images with multiple instances of different object classes, with various types of interactions between the objects. If we put aside the approach of bottom up segmentation and subsequent classification, we need to be able to combine detection and classification for multi-object configurations. This issue arises even when detecting a single object class, say faces. When several faces are present close to each other, or even occluding each other, or when trying to determine how to prune clusters of very close detections, one encounters the issue of object configurations which are not accounted for with simple object/background discriminative boundaries. All the more so when multiple object classes are present.

One interesting example of a coherent *discriminative* framework for dealing with object configurations, in the context of reading handwritten digits, is found in [20]. A well defined cost function is proposed involving an interaction between segmentations and outputs of classification. However for the system to work the authors needed to train the network with massive numbers of digits presented with *flanking* digits so that the pretrained classifiers would be robust to clutter in the subwindow being processed. It does not appear that such an approach can scale to multiple objects and novel types of configurations. Moreover the requirements on the training set size are prohibitive.

In terms of generative approaches [18,9] provide an overall *theoretical* proposal for compositional scene models involving hierarchies of parts/objects that are successively composed, ultimately to provide an explanation of the entire scene. In [5] a concrete attempt is made to compose object models into scene models. The notion of an *object support* is defined in terms of the model and the object instantiation. This concept is crucial in composing object models, defining object configurations, occlusion and other forms of interactions. In [5] it is assumed the object supports do not overlap, and the range of poses is rather limited. The main challenge comes from the presence of clutter and noise. Object supports can be defined naturally when one employs dense data models, such as the Bernoulli edge based model proposed in [2] and used in [5]. Sparse models such as the constellation models of [15,13], or [14] do not provide an object support, and could in fact be viewed as approximations to dense models. In [21] object supports are derived from constellation models.A related non-generative approach to computing object supports is proposed in [10].

In this chapter the ideas of [5] are extended to highly deformable objects, e.g. handwritten digits. We start with the formulation of single object deformable Bernoulli models and their composition into scene models (section 2). In section 3 we outline the patchwork of parts (POP) approximation to the Bernoulli model, which allows for tractable and efficient training and testing. In section 4 we describe a further coarse part based approximation which can be used to efficiently discover clusters within object classes, as well as quickly scan a large image for candidate detections. Results on combining the two approximations for isolated digit classification are given in section 5. In section 6 we explain how an image containing multiple objects is processed using the above models and how the optimal scene labelling is computed. Finally in 7 we provide some experimental results on hand written zip-codes from the US postal CEDAR database.

2 The Deformable Bernoulli Model

2.1 Oriented Edge Features

The data models defined below are all based on a set of eight binary oriented edge features defined originally in [4], and employed in multiple applications see e.g. [2,5,6]. The edge features are binary and computed at each point in the image which is defined on a grid L. Several edges can be present at one location - they are not mutually exclusive. These features are highly robust to intensity variations. Each detected edge is spread to its immediate 3×3 neighborhood. This spreading operation providing robustness to small local deformations which are very difficult to model, and greatly improves performance of any classifier implemented on the data. We write the binary data (after spreading) as $X = \{X_e(x) \mid x \in L, e = 1, \cdots, E\}$, where $E = 8$, corresponding to 8 orientations at increments of 45 degrees. In figure 1 we show the edges extracted on a typical zip-code for two different orientations. The darker points are the original edges and the gray areas the spreading regions.

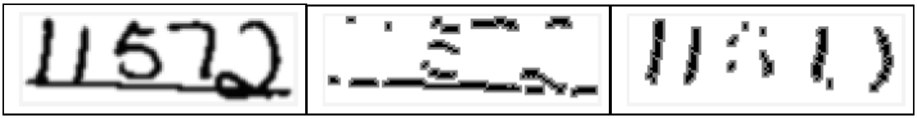


Fig. 1. Left: A sample zip-code. Middle: Horizontal edges. Right: Vertical edges. Dark points original edges, gray points after spreading.

2.2 One Object

We start with a data model for the edge features in an image containing only one object.

There are several components in the description of the model.

Instantiation set. A set $\Theta = L \times \Theta_0$ describing the possible instantiations of the object, where L is the image lattice and indicates all possible locations of the object, and Θ_0 describes the linear and non-linear deformations of the object. We write $\theta = (\xi, \vartheta)$ where $\xi \in L, \vartheta \in \Theta_0$. There will be a product prior distribution $P(\theta) = P(\xi)P(\vartheta)$, indicating that the deformation of the object is independent of its location. Typically with one object $P(\xi)$ is uniform on L.

Probability maps. A probability map on a reference grid G.
$p_e \equiv p_e(z), z \in G, e = 1, \ldots, 8$.

Probability instantiation. For any $\theta \in \Theta$ and any point $x \in L$ define an operator $\theta p_e(x)$ which assigns a probability of finding edge e at x as a function of the instantiation θ, and the probability map p_e. For example if θ is a map of $G \rightarrow L$ one reasonable form would be

$$\theta p_e(x) = \begin{cases} p_e(\theta^{-1}x) & \text{if} x \in \theta G \\ p_{e,bgd} & x \notin \theta G \end{cases}, \tag{1}$$

where $p_{e,bgd}$ is a generic background probability for edge e, and $\theta^{-1}x = \vartheta^{-1}(x - \xi)$. We will propose a different form in the context of the POP models below.

Given only one object of this class is present in the image at instantiation (ξ, ϑ) we assume the edges in the image are independent and have marginal probabilities at each point x given by $\theta p_e(x)$. Specifically we write

$$P(X|\theta) = \prod_e \prod_{x \in L} \theta p_e(x)^{X_e(x)} (1 - \theta p_e(x))^{1 - X_e(x)}. \tag{2}$$

Let the *object support* for edge type e be defined as

$$S_{\theta,e} = \{x \in L : \theta p_e(x) \neq p_{e,bgd}\}, \tag{3}$$

namely the set of pixels with probabilities different from the generic background probability, which bear some information regarding the presence of the object. The probability model is rewritten as

$$\begin{aligned} P(X|\theta) &= \prod_e \left[\prod_{x \in S_{\theta,e}} \theta p_e(x)^{X_e(x)} (1 - \theta p_e(x))^{1 - X_e(x)} \cdot \prod_{x \notin S_{\theta,e}} p_{e,bgd}^{X_e(x)} (1 - p_{e,bgd})^{1 - X_e(x)} \right] \\ &= P_{bgd}^{-1} \prod_e \prod_{x \in S_{\theta,e}} \left(\frac{\theta p_e(x)}{p_{e,bgd}} \right)^{X_e(x)} \left(\frac{1 - \theta p_e(x)}{1 - p_{e,bgd}} \right)^{1 - X_e(x)}, \end{aligned} \tag{4}$$

where P_{bgd} can be viewed as the probability of the data given *no object* is present in the image.

If Θ_0 consists of smooth mappings of G into L we have described a deformable template model. However since typically a semantic object class can contain more than several distinct smoothly deformable structures we model classes as *mixtures* of Bernoulli models. In other words introduce a discrete variable $m \in \{1, \ldots, K\}$ denoting the component and a distribution $P(m)$. For each

m we have a specific distribution on instantiations Θ, denoted $P(\theta|m)$, and a specific probability map $p_{e,m}$. Write the joint distribution of observables, deformation and component as

$$P(X,\theta,m) = P(X|\theta,m)P(\theta|m)P(m), \tag{5}$$

where $P(X|\theta,m)$ has the same form as equation (4) with the probability maps $p_{e,m}$.

For each component of each object class we denote the probability maps as $p_{e,m,c}, m = 1,\dots,K$. Given an image with a single object of unknown class we may ask for the maximum posterior on class

$$\begin{aligned}\tilde{c} &= \mathrm{argmax}_c P(c|X) = \mathrm{argmax}_c P(c)P(X|c) \\ &= \mathrm{argmax}_c P(c)\sum_m \int_\Theta P(X|\theta,m,c)P(\theta|m,c)d\theta P(m|c),\end{aligned} \tag{6}$$

where the key data term $P(X|\theta,m,c)$ is given in equation (4). The integration above is very difficult to compute so we substitute a maximization for integration and summation and define the classifier as

$$\hat{c} = \mathrm{argmax}_c P(c) \max_{\theta,m} P(X|\theta,m,c)P(\theta|m,c)P(m|c). \tag{7}$$

There may be an advantage to computing $\hat{c}$ since is comes together with an estimate of the instantiation. Note that in the standard classification problems with segmented data it is assumed that $\xi = 0$.

2.3 Scene Models

Define a scene as a set of objects $c_1,\dots,c_k$ with their instantiations and components $\theta_1, m_1,\dots,\theta_k, m_k$, and a partial ordering determining an occlusion relation between the objects. For simplicity we can assume that if $i < j$ than c_j can not occlude c_i. Denote a scene as

$$\mathbf{D} = \{k, c_1, m_1, \theta_1, \dots, c_k, m_k, \theta_k\}. \tag{8}$$

Let S_i denote the support of object i (equation (3)). Let the occluding region of object c_i for edge type e be the union of the supports of all previous objects,

$$O_{i,e} = \cup_{j=1}^{i-1} S_{j,e}. \tag{9}$$

The likelihood of the data given a scene $\mathbf{D}$ is then

$$P(X|\mathbf{D}) = P_{bgd}^{-1} \prod_e \prod_{i=1}^k \prod_{x \in S_{i,e} \setminus O_{i,e}} \left(\frac{\theta_i p_{e,c_i,m_i}(x)}{p_{e,bgd}}\right)^{X_e(x)} \left(\frac{1-\theta p_{e,c_i,m_i}(x)}{1-p_{e,bgd}}\right)^{1-X_e(x)}. \tag{10}$$

We introduce a prior on scenes with a probability distribution $P(k)$ on the number of elements in the scene and an interaction term between the objects involving their instantiation parameters. Assuming no interaction between the class and component labels we have

$$P(\mathbf{D}) = P(k)\left(\exp[U_k(\theta_1, \ldots, \theta_k)] \prod_{i=1}^{k} P(\theta_i, m_i, c_i)\right)/Z_k, \tag{11}$$

where $P(\theta_i|m, c)$ are the original distributions on Θ for the component m of class c. Again given the edge data of an image the scene label is obtained by maximizing the posterior on the entire *scenes* parameter

$$\hat{\mathbf{D}} = \mathrm{argmax}_{\mathbf{D}\in\mathcal{D}} P(\mathbf{D})P(X|\mathbf{D}), \tag{12}$$

yielding a set of pose parameters in addition to the labels of the objects.

The introduction of interactions between the instantiations introduces significant complications in the form of the distribution $P(\mathbf{D})$. For example in our application these interactions involve constraints on the intersections of the supports of objects. Thus Z_k involves the normalization of the product on the right on a subset of admissible k-tuples. In general computing Z_k is a challenge but it is essential for comparing the posterior on scenes with different numbers of elements. This is the fundamental challenge of compositional models (see [9].)

In our particular setting of reading zip-codes we have $k = 5$ so that Z_k is irrelevant in comparing different admissible instantiations. Another simpler setting is where the interaction term involves only the locations ξ_i of the objects:

$$U_k(\theta_1, \ldots, \theta_k) = U_k(\xi_1, \ldots, \xi_k).$$

Since $P(\theta_i|c, m)$ is independent of ξ_i the normalization constant Z_k is computed independently in terms of the ξ. In other circumstances if there is a very good data model, the likelihood component of the posterior should overwhelmingly point towards a particular value of k in which case there is no problem. But in general this issue remains a challenge.

3 Approximations I: Patchwork of Parts (POP) Models

So far we have considered Θ_0 as a set of smooth maps of the reference grid G into the lattice L, with the operator $\theta p_e(x)$ defined in terms of equation (1). The main problem in this formulation is the complex form of the posterior distribution on θ conditional on the data. This presents computational challenges which effect both training and labeling. In [6] a convenient approximation is introduced which we describe in brief.

3.1 POP Model Formulation

Instead of describing a full map of the reference grid the instantiation is summarized as the mapping of a moderate number of reference points $y_1, \ldots, y_n$ in the

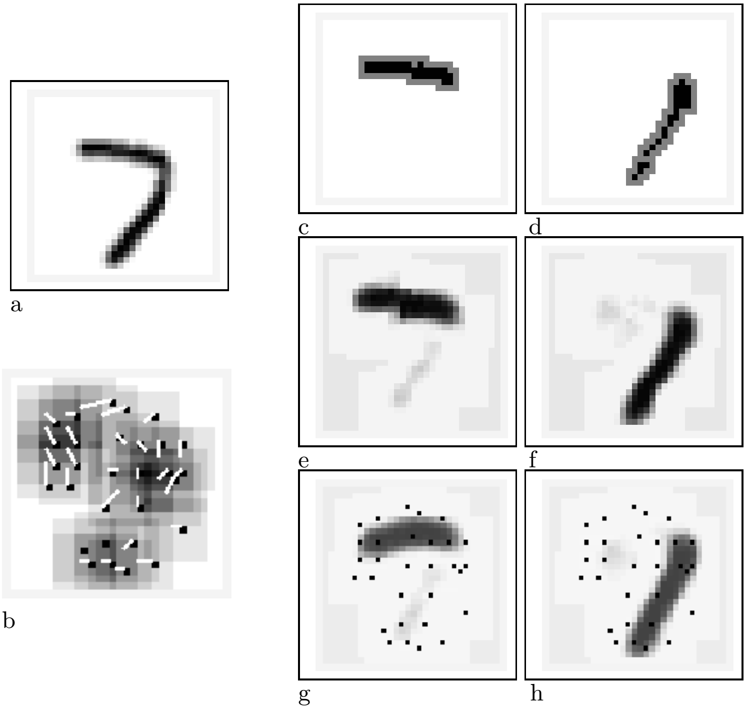


Fig. 2. a: Sample seven. b. The function $\mathcal{I}(x)$ for instantiation θ. Black dots are the reference points y_i. White arrows show the shifts $z_i - y_i$. Darker areas are correspond to higher values. (c,d) Extracted horizontal and vertical edges. (e,f) The global POP model for the two edge types given θ. (g,h) The model probability map for the two edge types (mean global POP), black dots are the reference points.

reference grid into the image lattice. Let $z_i = \theta y_i, i = 1, \ldots, n$ and with some abuse of notation write $\theta = (z_1, \ldots, z_n)$. Define *parts* Q_i of the full probability map

$$Q_{i,e}(s) \doteq p_e(y_i + s), e = 1, \ldots, 8, s \in W, \tag{13}$$

where W is some fixed size subwindow around the origin. Now imagine that the part Q_i is 'moved' to the point z_i. Edges at points in the image lattice that are not covered by any of the windows $z_i + W$ get assigned background probabilities. At points covered by one or more of the translated parts edges are assigned the average of the contributed probabilities. Specifically for each $x \in L$ let $\mathcal{I}(x) = \{i : x \in z_i + W\}$ be the set of shifted reference points whose W neighborhood covers x. The marginal probability at each point x is then given by the following average of the contributions of the parts:

$$\theta p_e(x) = P(X_e(x) = 1|\theta) = \begin{cases} \frac{1}{|\mathcal{I}(x)|} \sum_{i \in I(x)} Q_{i,e,x-z_i} & \text{if } |\mathcal{I}(x)| > 0 \\ p_{e,bgd} & \text{if } |\mathcal{I}(x)| = 0, \end{cases} \tag{14}$$

where $p_{e,bgd}$ is a generic background probability for edge type e.

This *patchwork* of the local models using the pointwise average of all local submodels covering the point x motivates the term *patchwork of parts (POP)* model. With this new definition of $\theta p_e(x)$, and staying with the conditional independence assumption, we write the global POP model conditional on the points θ as

$$P(X|\theta) = P(X|z_1, \ldots, z_n) = \prod_x \prod_e [\theta p_e(x)]^{X_e(x)}[1 - \theta p_e(x)]^{(1-X_e(x))}, \quad (15)$$

with $\theta p_e(x)$ defined in (14). Let $\bar{\theta} = (y_i)_{i=1,\ldots,n}$, i.e. the original reference points. The original probability map $p_e(y), y \in G$ is given by $\bar{\theta} p_e(y)$. Since the windows have not been moved the probabilities in the average in (14) are all the same and equal to $p_{e,y}$.

In 2(a) we show a sample '7', with the function $\mathcal{I}(x)$ in 2(b), together with white arrows connecting the reference points y_i to the instantiation points z_i. In panels (c,d) we show two edge types extracted from the image, in (e,f) we show the global POP model conditional on $z_1, \ldots, z_n$, and in (g,h) the original probability map. The gray areas in panels (g,h) show areas in the reference grid where $p_e = p_{bgd}$, the remaining areas are the *object support* at reference pose. The same holds for (e,f) - the object support includes all pixels outside the gray areas.

3.2 Training

This simplified model lends itself to a very simple and effective approximate estimation procedure. Given a fixed collection of start points x_i on a coarse subgrid of the reference grid, separately estimate a Bernoulli model $\tilde{Q}_i$ supported on a window of size W, for the data around x_i. For each local model, the unobserved variable - the instantiation - is simply a shift τ of the start point x_i, constrained within a fixed window V. For estimation assume that conditional on the shift τ the data is generated independently according to $\tilde{Q}_i$ inside $x_i + \tau + W$ and according to the homogeneous background model *everywhere else* in the image. Since we can enumerate all the shifts in V a full EM algorithm can be implemented. Some of the local Bernoulli models $\tilde{Q}_i$ end up being very close to a homogeneous background model and are eliminated.

The reference points y_i are obtained from this procedure as $x_i + \bar{\tau}_i$ where $\bar{\tau}_i$ is the average shift, estimated through the EM procedure, over all training points. Finally the full probability map is created by patching together the local models using equations (14),(15) with $\theta = (y_1, \ldots, y_n)$. The probability maps shown in figure 2 (g,h) were estimated in this manner. Despite the separate training of each part Q_i the data imposes consistency between models estimated at neighboring windows and the final probability map is smooth and has the form of a seven. For more details see [6].

3.3 Computing an Instantiation

Once the probability map and the reference points of the POP model have been estimated it is possible to run the model on a test image. Around each reference point y_i, find the optimal shift τ_i^* for the submodel Q_i defined in equation (13), in terms of the likelihood ratio to the background model, within the range V of shifts. This is done independently of all other shifts. Setting $z_i = y_i + \tau_i^*$, compute the likelihood under the global POP model $P(X|z_1, \dots, z_n)$. The instantiation shown in 2 (b) was obtained in this manner. Joint optimization of the shifts τ_i to optimize the full likelihood is computationally very intensive.

3.4 Training Additional Parameters

Once the probability map is estimated other parameters of the model can be estimated by computing an instantiation for each training data using the method outlined in 3.3. One can obtain the distribution of the computed likelihoods, which are assumed to be Gaussian and summarized with a mean and standard deviation μ, σ. Furthermore we estimate a joint distribution $p(\theta)$ for the computed instantiations. Assuming a joint Gaussian we take the means to be $y_i, i = 1, \dots, n$ and a $2n \times 2n$ covariance matrix $\varGamma$, whose dimension is twice the number of reference points. A POP model for a class c can be summarized as the collection

$$\mathcal{M}_c^{pop} = \{Q_{i,c}, y_{i,c}, i = 1, \dots, n, \varGamma_c, \mu_c, \sigma_c\}, \tag{16}$$

where each $Q_{i,c}$ is the local model in the window W around point y_i.

4 Approximations II: Coarse Part Based Mixture Models

Whereas the estimation of the probability maps with a POP model proves to be rather simple, estimating a mixture of POP models is quite a challenge. One can formulate a more complex EM procedure that involves both the unobserved instantiation parameters and the discrete component parameter, see for example [1]. However this is quite computationally demanding. We propose the following simplification which involves introducing a further approximation of the POP model in terms of part models on a coarse grid.

4.1 Generic Part Library with Rotational Symmetry

It is intuitively clear that the local Bernoulli models, i.e. the restriction of the full model to small windows, can be well approximated by a moderate number of fixed models - a fixed library of parts. We thus consider local edge maps in a window W arbitrarily placed in the image as coming from a mixture distribution of local Bernoulli models. Since it is sensible to assume that any local structure occurs at all rotations we assume the mixture includes a discrete set of A equally spaced rotations of a small number of base components. This both simplifies the problem of chosing the number of components and provides a means for rotating models.

We do not want to model 'background' windows in this mixture, i.e. windows with no real structure. These are assumed to be distributed according to a

Bernoulli model with homogeneous probabilities $p_{e,bgd}$ for each edge. We reject the null background hypothesis on a subwindow if its probability under the background model is less than .01. For example if $p_{e,bgd} \equiv p_{bgd}$ this reduces to setting a minimal number of edges τ_e in the window. The training sample then consists of windows $W+x$ around random points in a set of images, where the background hypothesis has been rejected. Write the mixture as

$$P(X_W) = \sum_{f=1}^{K_F}\sum_{\alpha=0}^{A-1} \tau_{f,\alpha} P_{f,\alpha}(X_W) \tag{17}$$

$$P_{f,\alpha}(X_W) = \prod_{s\in W}\prod_{e} p_{e,f,\alpha}(s)^{X_e(s)}(1-p_{e,f,\alpha}(s))^{(1-X_e(s))}.$$

Theoretically one would want to write $p_{e,f,\alpha}(s) = p_{\alpha^{-1}e,f}(\alpha^{-1}s)$ for some base probabilities $p_{e,f}$. This however is problematic since it is unclear how to rotate the edge by angles that are not multiples of $\pi/4$ and the square domain W is not invariant under rotations. Instead we assume that if the edge map - X_{W+x} - in a subwindow is from component (f,α), then after rotation of the *original image* around x by angle a the resulting edge map in the same window is a sample from component $(f,\alpha+a)$, i.e. it is distributed according to $P_{f,\alpha+a}$. We take the addition of the angle indices to be *modulo* A.

Thus for each point x which is the center of a valid 'non-background' subwindow we rotate the original gray level image *around* x at the A angles and compute the edge maps $X^{(a)}_{x+W}, a=0,\dots,A-1$ from the rotated images. We denote the resulting training set as $X^{t,a}_W, t=1,\dots,T, a=0,\dots,A-1$. Suppose $X^{t,0}_W$ is a sample from P_{f_t,α_t} then $X^{t,a}_W$ is a sample from P_{f,α_t+a}. But f_t,α_t are unobserved and are dealt with in the framework of the EM algorithm. The estimate of $p_{e,f,\alpha}$ with fully observed data (i.e. knowing α_t, f_t for each training sample $X^{(t)}_W$) would reduce to

$$\hat{p}_{e,f,\alpha}(s) = \frac{1}{T_f}\sum_{t:f_t=f} X^{(t,\alpha-\alpha_t)}(s), \alpha=0,\dots,A-1 \tag{18}$$

where T_f is the number of training subwindows from component f. Instead, denoting by $\pi_{f,\alpha,t}$ the estimated conditional expectation on (f,α) for training sample t, the EM algorithm produces the following estimate

$$\hat{p}_{e,f,\alpha}(s) = \frac{1}{w_f}\sum_{t=1}^{T}\sum_{a=0}^{A-1} \pi_{f,a,t} X_e^{(t,\alpha-a)}(s),\ \ \alpha=0,\dots,A-1$$

$$w_f = \frac{1}{T}\sum_{t=1}^{T}\sum_{a=0}^{A-1}\pi_{f,a,t}. \tag{19}$$

The resulting features are very easy to interpret. In the first column of figure 3 we show the mean gray level images with $K_F=6$ parts at angle $\alpha=0$ found on the MNIST data base. In other words we show the mean of all subwindows

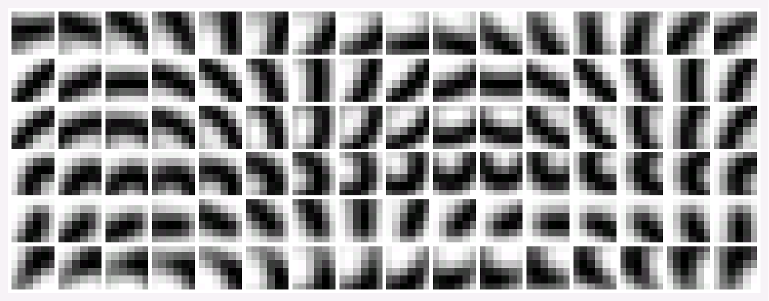


Fig. 3. Left column $K_F = 6$ parts at angle $\alpha = 0$. Subsequent columns rotations of parts at multiples of $\pi/8$. All together 96 parts.

assigned to each cluster. In the subsequent columns are the parts corresponding to the rotations $\alpha = 1, \ldots, 16 (A = 16)$. The mean images are easier to visualize than the actual probability maps, but the clustering based on edge maps is essential for photometric invariance. The same process can be performed on generic gray level images with widely varying lighting and gray scale maps. The unsupervised clustering process has discovered several basic local structures - curves with different curvatures, 'junctions' and 'endings'. We have not yet developed a rigorous framework for choosing the number of components K_F, but experiments show that the results are not very sensitive to this choice if a sufficient number of angles is used. The only price for using more angles is computational. In this chapter, since the rotation information of the features is not used, we relabel the $A \cdot K_F$ features with the index $f = 1, \ldots, F$.

4.2 Feature Labeling, Spreading and Subsampling

Having estimated a set of local features, a local feature map $Y_f(x)$, $f = 1, \ldots, F$ is computed. At each point x for which the local edge data X_{x+W} is found to be *non-background*, the most likely feature under the mixture model is recorded, i.e.

$$Y_f(x) = \begin{cases} 1 & \text{if } f = \text{argmax}_{f'} \, \log P_{f'}(X_{x+W}) \\ 0 & \text{otherwise} \end{cases}$$

Note that the computation of the log-likelihood at all locations is simply a linear convolution on the *binary edge data*, not the original image data.

The result is a new set of feature maps on the image lattice L. Since each feature encodes an entire local structure, its exact position is no longer as important as the exact edge positions. We take advantage of this fact by *spreading* the detected features to a neighborhood B of the original location and subsampling to a sublattice L_b at spacing b of the original lattice L.

$$Y^s_f(x) = \max_{\xi \in B + b \cdot x} Y_f(\xi) \tag{20}$$

for $x \in L_b$. Note that after subsampling several features can occur at the same point $x \in L_b$.

4.3 Part Based Object Models -CPM's

The local features - parts - on the coarse subgrid were motivated as approximations of the original POP model. After spreading and subsampling one assumes that the local deformations are accounted for and there is no need for a deformation variable ϑ and the instantiation is determined by the location ξ. Thus each class is modeled as a mixture of K_c Bernoulli models based on the new features in a coarse reference grid G_b. That is, conditional on the model component m, each feature Y^s_f is assumed to occur independently at each location z in L with some probability $p_{f,m,c}(z)$:

$$\begin{aligned} &P(Y^s|M=m, C=c, \xi=x) \\ &\quad = \prod_{y\in x+G_b}\prod_f p_{f,m,c}(y-x)^{Y^s_f(y)}(1-p_{f,m,c}(y-x))^{(1-Y^s_f(y))} \end{aligned} \tag{21}$$

In this context the independence assumption is blatantly wrong unless one uses a very large number of components; after all if one conditions on ‘enough’ all variables become independent. Nonetheless these mixture models give rise to a well defined estimation procedure based on the EM algorithm. Our experience is that due to the simplicity of the model - the parameters involve simple proportions - the EM algorithm is very stable and does not depend heavily on the initialization. For each component we also estimate the mean and standard deviation of the log-likelihood - $\mu^{coarse}_{c,m}, \sigma^{coarse}_{c,m}$. We denote the final coarse part based model (CPM) for a class c as

$$\mathcal{M}^{coarse}_c = \{p_{f,m,c}(z), z\in G_b, \mu^{coarse}_{c,m}, \sigma^{coarse}_{c,m}, f=1,\ldots,F, m=1,\ldots,K_c\}. \tag{22}$$

The value K_c is chosen so that on average there would be approximately 20 samples per component which is sufficient to provide good estimates of each of the marginal probabilities.

In [8] we show that such coarse models yield very powerful likelihood based classifiers on the MNIST dataset, as well as good single object detectors, see also section 5.2. In training they are used to obtain class clusters which become the components of the POP models. Finally for the purpose of scene analysis these models will be used as an indexing mechanism to prune the number of locations and classes on which to compute the more detailed POP models.

5 Combining CPM's and POP Models

5.1 Mixtures of POP Models

Given K_c mixture components of the coarse object models each training image will have one component with highest likelihood. It is almost always the case that the likelihood of one component is much higher than all the rest and there is no ambiguity. In the top row of figure 4 we show the mean image for each of the components of the coarse model. It is already clear that the estimation procedure using the coarse features discovers interesting subclasses of each digit class. The same phenomenon is observed with a dataset of face images.

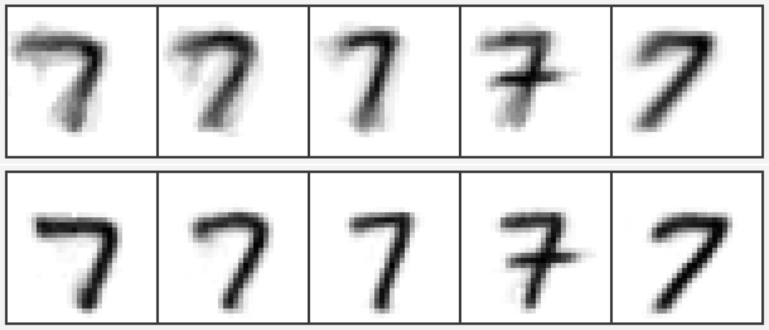


Fig. 4. Top: Mean images of training data in each cluster estimated from coarse feature based models. Bottom: *Mean global image* for the POP model estimated from the data points in each cluster.

Now use the images assigned to each component m to train a full POP model on the original reference grid G. To visualize the effect of the estimation of the POP models it is possible to create a *global mean image* as opposed to visualizing the probability maps of each model. Given the original images of the training data: $I^{(t)}, t = 1, \ldots, T$, and for each start point x_i, take the average of the shifted subimages:

$$J_{W,i} = 1/T \sum_{t=1}^{T} I^{(t)}_{x_i + \tau_i^{(t)} + W},$$

where $\tau_i^{(t)}$ is the most likely shift, as computed in the EM procedure. Now create a *global mean image* using the patchwork operation with the subimages $J_{W,i}$ using the estimated reference points y_i,

$$J(x) = \begin{cases} \frac{1}{|\mathcal{I}(x)|} \sum_{i \in \mathcal{I}(x)} J_{W,i}(x - y_i) & \text{if } |\mathcal{I}(x)| > 0 \\ 0 & \text{else} \end{cases}. \tag{23}$$

In figure 4 we show the global mean image for the POP models for the five clusters of sevens below the regular mean images of the sevens in each cluster. The subsequent estimation of the POP models creates a much crisper model since the local variations are accounted for.

5.2 Hierarchical Classification

The hierarchy of coarse feature based models and refined POP models leads to a natural organization of computation. For example in simple standardized classification problems one can first run a classifier based on the CPM's and only when the log-likelihood ratio between the top two classes is below some threshold run the more computationally intensive and refined POP models. The results of this procedure on the MNIST data set is summarized in table 1. We see that generative models with no discriminative training are able to obtain state of the art classification rates (under 1% error) with small training samples. However this classification problem, so intensively studied in the machine learning community is very artificial. The objects are not only cleanly segmented, they are also scaled and centered. This is hardly the case when trying to analyze unsegmented scenes even as 'simple' as a zip-code. In section 6 we approach the issue from a top down model based approach.

Table 1. Error rates on MNIST

No. of Training data per class	Components per model	CPM error rate	CPM+POP error rate	SVM Error rate
30	2	4.26	3.43	6.57
100	5	2.68	1.73	3.02
500	20	1.71	1.12	1.47
1000	30	1.51	**.9**	1.15

5.3 Scale and Slant Clusters

The characters in the MNIST dataset are well centered and scaled. In real zip-codes there is much larger variability in terms of scale, and other linear parameters such as slant or shear. In one zip-code one can find a 2:1 ratio in size of characters, some upright characters and some heavily slanted ones. In principle one could add a linear parameter to the instantiation parameter θ. However within a small neighborhood of the identity map the linear variations are easily accommodated by the configurations of the reference points. For the larger variations we produce additional components to the mixture models indexed by a linear parameter. Specifically define a discrete set of scales and slants Σ. For each $\sigma \in \Sigma$ apply σ to the training data of component m and retrain a POP model. The end result is $K_m \times |\Sigma|$ POP models covering a large range of linear variations and the non-linear variations governed by the ϑ parameter.

In addition, to expedite certain computations we also store a simple estimate $\mathcal{M}^{simple}_{c,m,\sigma}$ of marginal probabilities of the edges for the training data in each component m, *with no accounting for local shifts.* We now write

$$\mathcal{M}^{fine}_{c} = \{\mathcal{M}^{pop}_{c,m,\sigma}, \mathcal{M}^{simple}_{c,m,\sigma}, m = 1, \ldots, K_c,\ \sigma \in \Sigma\} \tag{24}$$

where each $\mathcal{M}^{pop}_{c,m,\sigma}$ is a POP model as in equation (16).

6 Analyzing a Scene

The scene labeling is defined as the optimizer in equation (12) over the set $\mathcal{D}$. This is of course an intractable computation and some short cuts need to be defined. First extract a moderate number of candidate detections, i.e. class-component-instantiation triples (c, m, θ), *ignoring their interactions.* Many of these detections may have substantial overlaps in their supports. The goal is to make sure the correct objects are among these detections at the price of having say several hundred false positives. This is done in several stages.

6.1 Stage I: Candidate Detections Using CPM's

The coarse grid L_b is labeled with the collection of part variables $Y_f, f = 1, \dots, F$ as detailed in section 4.2. At each point $x \in L_b$ run all the CPM's on Y_{x+G_b} and keep those models for which the likelihood is higher than $\mu^{coarse}_{c,m} - \alpha \cdot \sigma^{coarse}\_c, m$ where α is a parameter usually set to 2 or 3. This yields a list Δ^{coarse} of candidate detections (c, ξ), where $\xi = x \cdot b$ denotes a location on the original lattice L. In the present setting we omit the m variable denoting the component index from the detection. Note that this step is the only one involving a full scan of the image on the coarse grid.

Even though the coarse models yield good classifiers on segmented and normalized data, they do not provide precise information regarding object support and hence are not as useful when constructing probability models for object configurations. Therefore each candidate CPM detection is subsequently analyzed with the POP models.

6.2 Stage II: Refining Candidate Detections

For each detection in $(c, \xi) \in \Delta^{coarse}$ choose the most likely POP component. It is inefficient to compute the full instantiation (see section 3.3) for each pair (m, σ), which involves optimization on each reference point y_i. Instead we use $\mathcal{M}^{simple}_{c,m,\sigma}$ and compute the likelihood ratio to the background for each (m, σ) and for a range of locations around ξ. The optimal likelihood ratio then determines the preferred component (m^*, σ^*), and a location x. An optimal instantiation ϑ is only computed for the POP model $\mathcal{M}^{pop}_{c,m^*,\sigma^*}$ at location x. The result is a list Δ^{fine} of quintuples $\delta = (c, m, \sigma, \vartheta, x)$ derived from the original list of class location pairs Δ^{coarse}.

6.3 Finding the Optimal Scene Labeling

From the list of detections Δ^{fine} we now want to extract the optimal scene $\mathbf{D}$ of the image. Even if the distribution on scenes is fully specified (see discussion in section 2.3) this would be a complex computation and one can not guarantee that a global optimum will be found. Rather one would need to develop a sequence of reasonable approximations.

Table 2. Parameters used in zip-code experiment

Reference grid sizes	$G - 40 \times 40, G_b - 5 \times 5$, $b = 6$
Number of local features	$K = 96, K_F = 6$, 6 components, 16 angles.
Window sizes	$W - 6 \times 6$, $V - 5 \times 5$
Background prob.	$p_{e,bgd} \equiv .1$
Min. no. of edges for non-bgd. window	$\tau_e = 40$
Scales and slants	5 - scales .7 - 1.8, 3 slants.
Coarse models	Training - 100 per class, 5 components per class.
Fine models	Training - 500 per class, 20 components per class.

On the other hand in the particular problem of reading zip-codes this is not an issue since we know there are 5 objects, and these are more or less linearly organized. Thus in equation (11) $k = 5$ and the interaction term only involves hard constraints on the arrangements of the objects. First there is an upper limit on the area of the intersection of the supports of any two objects relative to the areas of each of the objects. Then assuming the objects are ordered left to right, which also determines the order of occlusion, given two consecutive objects $\delta = (c, m, \sigma, \vartheta, x), \delta' = (c', m', \sigma', \vartheta', x')$ we assume $x_1 < x'_1$ and impose an upper limit on $|x - x'|$. We also impose an upper limit on the angle between x and x'. One could add some soft constraints such as penalizing large differences in the linear pose index σ between two consecutive objects, we have not done so.

Since object $i-1$ can not occlude object $i+1$ rewrite the likelihood of equation (10) for an admissible sequence of 5 objects $\delta_1, \ldots, \delta_5$ as

$$P(X|\mathbf{D}) = P_{bgd}^{-1} \prod_e \prod_{i=1}^{k} \prod_{x \in S_{i,e} \setminus S_{i-1,e}} \left(\frac{\theta_i p_{e,i}(x)}{p_{e,bgd}} \right)^{X_e(x)} \left(\frac{1 - \theta_i p_{e,i}(x)}{1 - p_{e,bgd}} \right)^{1 - X_e(x)}, \tag{25}$$

where $p_{e,i} = p_{e,c_i,m_i,\sigma_i}$, $S_{i,e}$ is the support of object δ_i and $S_{0,e} = \emptyset$.

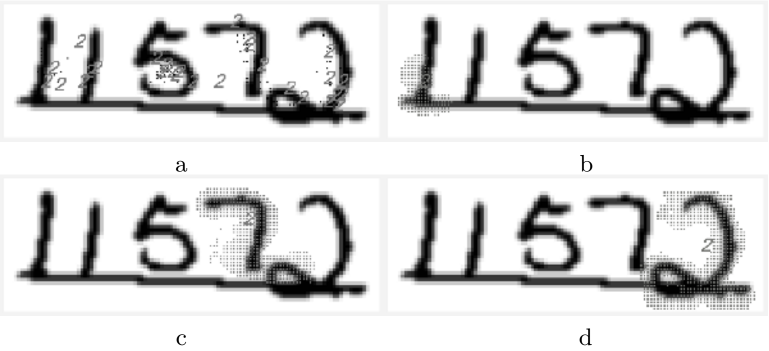
Now the likelihood is a product of terms only involving consecutive pairs of detections, and the constraints on configurations are also given in terms of pairs. Consequently optimizing the likelihood over all admissible sequences of five objects from the list of candidate detections can be efficiently done with dynamic programming. The state space for each of the five 'slots' is the set of detections in Δ^{fine}. Since the same pair (δ, δ') of detections can be entertained several times we precompute the value

$$\Phi(\delta, \delta') = \sum_e \sum_{x \in S_{\delta',e} \setminus S_{\delta,e}} X_e(x) \log \left(\frac{\theta' p_{e,\delta}(x)}{p_{e,bgd}} \right) + (1 - X_e(x)) \log \left(\frac{1 - \theta' p_{e,\delta}(x)}{1 - p_{e,bgd}} \right),$$

but only for those pairs which satisfy the pairwise hard constraints. Once Φ is precomputed dynamic programming reduces to lookups and summations. It is also an easy matter to compute L top sequences which could be further processed if needed.

7 Experimental Results

Table 2 summarizes the parameter settings in the experiments. In figure 5.a we show all locations of detections of class 2 in the coarse pass on a zip-code. In 5 (b,c,d) we show the support of a number of these detections after computation of the optimal (m, σ) and the instantiation θ of the corresponding POP model. Note that due to the range of sizes of the models, the algorithm finds 2's in strange places. There is no apriori way to know the correct size, since in some zip-codes the size of the digits ranges quite drastically.



a

b

c

d

Fig. 5. a. Coarse detections of class 2. b-d Support of some fine model detections of 2's after optimizing over model component, linear parameter and instantiation. The supports shown are the union of the supports $S_{e,\theta}$ for the different edge types.

In figure 6 we show the top two scene labelings obtained with dynamic programming for three different zip-codes. These examples illustrate several interesting aspects. First we see that due to the clutter in the form of the horizontal bar in the first zip-code there is a well formed 2 shown in figure 5(c). This instance appears in the second best labeling shown in figure 6(b). It appears with some overlap with the subsequent detection. As indicated above some percent overlap between supports is allowed and is then modeled as if one object partially occludes the other. This is necessary since indeed sometimes digits share some parts of the stroke. The dynamic programming happens to select the correct labeling in the first zip-code despite the presence of the bar. However since there is no explicit modeling of structures other than digits, other types of clutter could lead to false positives and incorrect labeling. In both the second and third zip-code one sees the large differences in object size as well as the difficulty in computing a bottom up segmentation; The flanking 4 and 6 in the second zip-code and the connected 0's in the third. Indeed any labeling obtained by the algorithm, together with the object supports provides a top down object based segmentation.

Finally table 3 shows some of the results on a set of 1000 zip-codes from the US postal CEDAR data base. For comparison we show two reported results from

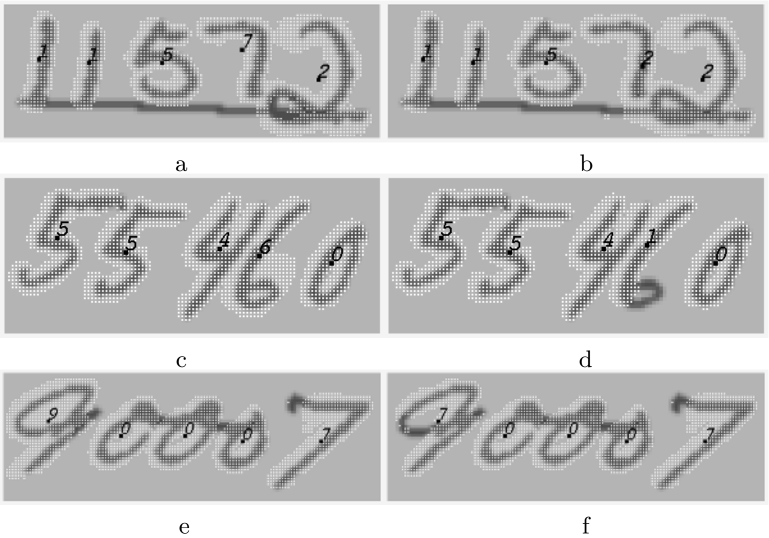


Fig. 6. Zip-code labeling. For three different zip-codes we show the top 2 labelings, together with the support (white dots) of each detection.

the literature in the mid-90's. All the reported methods used quite a number of dedicated preprocessing steps tailored to the problem. In our implementation no preprocessing or normalization is performed on the zip-code image, nor is there any presegmentation. The overall full zip-code recognition rate is 85.8, with 20% rejection the rate rises to 93.1%, reaching 97.6% at 50% rejection. The models employed in this implementation correspond to the third row of table 1 reporting an error rate just under 1% on the normalized MNIST images. One would then expect a lower error rate on the zipcodes. However as mentioned above, error rates on presegmented and centered images is misleading. If the scene labeling algorithm is run on individual MNIST images just as it is run on the zip-codes, assuming 1 object per image ($k = 1$) but assuming the location and pose are *not* known, the error rate increases significantly, to around 4%.

The computation time on a 2Ghz P-IV is approximately 10-15 seconds per zip-code where the largest computation is the massive loop over components and linear poses using the simple Bernoulli step. There are many ways one could

Table 3. Left: Comparison of zip-code classification results. Right: Scene Model classification rate against rejection rate.

Author	**n**	% corr.	% corr. at % rej
[19]	436	85%	97% - 34%
[22]	1566	*	96.5% - 32%
[28]	1000	72%	95.4% - 43%
Scene Models	1000	**85.8** %	**96.3% - 33%**

% rej.	% corr.
10%	89.1%
19%	93.1%
30%	95.5%
50%	97.6%

expedite this step in particular are more clever use of coarse to fine computational techniques as proposed in [16].

8 Discussion

We have shown that scene based models can be used to label object configurations with no preprocessing or presegmentation, yielding competitive results. The application o zipcodes is constrained since the number of objects is known and their arrangement is linear. Still, a variety of greedy algorithms can be used to find high scoring configurations in terms the proposed scene data model, such as sequentially selecting the most likely object from the remaining candidate detections *conditional* on those already selected. Of primary interest is improving the background model. The conditional independence assumption for background is very strong and the result is that clutter in the background can score very high in terms of the likelihood ratio of certain object models. One possibility is to use the fixed part library as a collection of 'background' objects whose labels and locations are incorporated in the scene annotation. The alternative to a candidate object instantiation would be the set of parts covering the same support.

Acknowledgement. Yali Amit was supported in part by NSF Grant 0427223, and by L2TI, Université Paris XIII, Villetaneuse.

References

1. S. Allassonnière, Y. Amit, and A. Trouvé. Toward a coherent statistical framework for dense deformable template estimation. Technical report, Department of Statistics, University of Chicago, 2005.
2. Y. Amit. *2d Object Detection and Recognition: Models, Algorithms and Networks.* MIT Press, Cambridge, Mass., 2002.
3. Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.
4. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11:1691–1715, 1999.
5. Y. Amit, D. Geman, and X. Fan. A coarse-to-fine strategy for multi-class shape detection. *IEEE PAMI*, 26:1606–1621, 2004.
6. Y. Amit and A. Trouvé. Pop: Patchwork of parts models for object recognition. Technical report, Department of Statistics, University of Chicago, 2004.
7. S. Belongie, J. Malik, and S. Puzicha. Shape matching and object recongition using shape context. *IEEE PAMI*, 24:509–523, 2002.
8. E. J. Bernstein and Yali Amit. Part-based statistical models for object classification and detection. In *CVPR (2)*, pages 734–740, 2005.
9. E. Bienenstock, Geman S., and D. Potter. Compositionality, mdl priors, and object recognition. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information and Processing Systems*, volume 9, pages 834–844, Cambridge, Mass., 1997. MIT Press.
10. E. Borenstein, E. Sharon, and Ullman S. Combining bottom up and top down segmentation. In *Proceedings CVPRW04*, volume 4. IEEE, 2004.

11. L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proc. IEEE Inter. Conf. on Pattern Recognition*, pages 77–82, 1994.
12. M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. of the 5th European Conf. on Computer Vision, ECCV 98*, pages 628–641, 1998.
13. Michael Burl, Markus Weber, and Perona Pietro. A probabilistic approach to object recognition using local photometrie and global geometry. In *Proc ECCV*, pages 628–641, 1998.
14. D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *to appear, Proceedings CVPR 2005*, 2005.
15. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the International Conference on Computer Vision*, volume 1, 2003.
16. F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41:85–107, 2001.
17. F. Fleuret and D. Geman. Fast face detections with precise pose estimation. In *Proceedings of ICPR2002, I*, pages 235–238, 2002.
18. S. Geman, D. Potter, and Z. Chi. Composition systems. *Quarterly J. Appl. Math.*, LX:707–737, 2002.
19. T. M. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31:257–272, 1998.
20. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
21. B Leibe and B Schiele. Interleaved object categorization and segmentation. In *BMVC'03*, 2003.
22. P Palumbo and S. Srihari. Postal address reading in real time. *Intr. Jour. of Imaging Science and Technology*, 1996.
23. M. Revow, C. K. I. Williams, and G. E. Hinton. Using generative models for handwritten digit recognition. *IEEE PAMI*, 18:592–606, 1996.
24. H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. PAMI*, 20:23–38, 1998.
25. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. Technical Report AI-Memo 2004-008, MIT, 2004.
26. S. Ullman, M. Vida-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5:682–687, 2002.
27. P Viola and M. J. Jones. Robust real time object detection. *Intl. Jour. Comp. Vis.*, 2002.
28. S. C. Wang. *A statistical model for computer recognition of sequences of handwritten digits, with applications to zip codes.* PhD thesis, University of Chicago, 1998.