Unsupervised Learning of Models for Recognition

M. Weber[1] M. Welling[2] P. Perona[1,2,3]

[1] Dept. of Computation and Neural Systems
[2] Dept. of Electrical Engineering
California Institute of Technology
MC 136–93, Pasadena, CA 91125, U.S.A.
[3] Università di Padova, Italy
{mweber,welling,perona}@vision.caltech.edu

Abstract. We present a method to learn object class models from unlabeled and unsegmented cluttered scenes for the purpose of visual object recognition. We focus on a particular type of model where objects are represented as flexible constellations of rigid parts (features). The variability within a class is represented by a joint probability density function (pdf) on the shape of the constellation and the output of part detectors. In a first stage, the method automatically identifies distinctive parts in the training set by applying a clustering algorithm to patterns selected by an interest operator. It then learns the statistical shape model using expectation maximization. The method achieves very good classification results on human faces and rear views of cars.

1 Introduction and Related Work

We are interested in the problem of recognizing members of object classes, where we define an *object class* as a collection of objects which share characteristic features or *parts* that are visually similar and occur in similar spatial configurations. When building models for object classes of this type, one is faced with three problems (see Fig. 1). *Segmentation or registration of training images*: Which objects are to be recognized and where do they appear in the training images? *Part selection*: Which object parts are distinctive and stable? *Estimation of model parameters*: What are the parameters of the global geometry or *shape* and of the appearance of the individual parts that best describe the training data?

Although solutions to the model learning problem have been proposed, they typically require that one of the first two questions, if not both, be answered by a human supervisor. For example, features in training images might need to be hand-labeled. Oftentimes training images showing objects in front of a uniform background are required. Objects might need to be positioned in the same way throughout the training images so that a common reference frame can be established.

Amit and Geman have developed a method for visual selection which learns a hierarchical model with a simple type of feature detector (edge elements) as its front end [1]. The method assumes that training images are registered with respect to a reference grid. After an exhaustive search through all possbile local feature detectors, a global model is built, under which shape variability is encoded in the form of small regions in which local features can move freely.

Fig. 1. Which objects appear consistently in the left images, but not on the right side? Can a machine learn to recognize instances of the two object classes (*faces* and *cars*) without any further information provided?

Burl et al. have proposed a statistical model in which shape variability is modeled in a probabilistic setting using Dryden-Mardia shape space densities [2, 11, 3, 4]. Their method requires labeled part positions in the training images.

Similar approaches to object recognition include the active appearance models of Taylor et al. [5, 8] who model global deformations using Eigenspace methods as well as the Dynamic Link Architecture of v. der Malsburg and colleagues, who consider deformation energy of a grid that links landmark points on the surface of objects [10]. Also Yuille has proposed a recognition method based on gradient descent on a deformation energy function in [15]. It is not obvious how these methods could be trained without supervision.

The problem of automatic *part selection* is important, since it is generally not established that parts that appear distinctive to the human observer will also lend themselves to successful detection by a machine. Walker et al. address this problem in [14], albeit outside the realm of statistical shape models. They emphasize “distinctiveness” of a part as criteion of selection. As we will argue below, we believe that part selection has to be done in the context of model formation.

A completely unsupervised solution of the three problems introduced at the beginning, in particular the first one, may seem out of reach. Intuition suggests that a good deal of knowledge about the objects in question is required in order to know where and what to look for in the cluttered training images. However, a solution is provided by the expectation maximization framework which allows simultaneous estimation of unknown data and probability densities over the same unknown data. Under this framework, all three problems are solved simultaneously.

Another compelling reason to treat these problems jointly, is the existing trade-off between localizability and distinctiveness of parts. A very distinctive part can be a strong cue, even if it appears in an arbitrary location on the surface of an object—think e.g. of a manufacturer's logo on a car. On the other hand, a less distinctive part can only contribute information if it occurs in a stable spatial relationship *relative* to other parts.

2 Approach

We model object classes following the work of Burl et al. [2, 4]. An object is composed of *parts* and *shape*, where 'parts' are image patches which may be detected and characterized by appropriate detectors, and 'shape' describes the geometry of the mutual position of the parts in a way that is invariant with respect to rigid and, possibly, affine transformations [12]. A joint probability density on part appearance and shape models the object class. Object detection is performed by first running part detectors on the image, thus obtaining a set of candidate part locations. The second stage consists of forming likely object hypotheses, i.e. constellations of appropriate parts (e.g. eyes, nose, mouth, ears); both complete and partial constellations are considered, in order to allow for partial occlusion. The third stage consists of using the object's joint probability density for either calculating the likelihood that any hypothesis arises from an object (object detection), or the likelihood that one specific hypothesis arises from an object (object localization). In order to train a model we need to decide on the key parts of the object, select corresponding parts (e.g. eyes, nose etc) on a number of training images, and lastly we need to estimate the joint probability density function on part appearance and shape. Burl et al. [3] perform the first and second act by hand, only estimating the joint probability density function automatically. In the following, we propose methods for automating the first and second steps as well.

Our technique for selecting potentially informative parts is composed of two steps (see Fig. 2). First, small highly textured regions are detected in the training images by means of a standard 'interest operator' or keypoint detector. Since our training images are not segmented, this step will select regions of interest both in the image areas corresponding to the training objects and on the clutter of the background. If the objects in the training set have similar appearance then the textured regions corresponding to the objects will frequently be similar to each other as opposed to the textured regions corresponding to the background which will be mostly uncorrelated. An unsupervised clustering step favoring large clusters will therefore tend to select parts that correspond to the objects of interest rather than the background. Appropriate part detectors may be trained using these clusters.

The second step of our proposed model learning algorithm chooses, out of these most promising parts, the most informative ones and simultaneously estimates the remaining model parameters. This is done by iteratively trying different combinations of a small number of parts. At each iteration, the parameters of the underlying probabilistic model are estimated. Depending on the performance of the model on a validation data set, the choice of parts is modified. This process is iterated until the final model is obtained when no further improvements are possible.

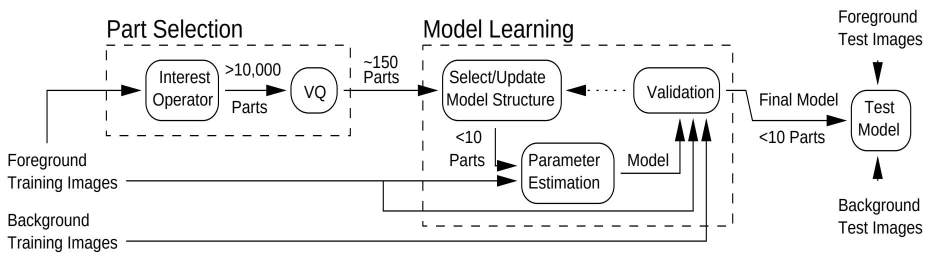

Fig. 2. Block diagram of our method. "Foreground images" are images containing the target objects in cluttered background. "Background images" contain background only.

Outline of the Paper In Section 3, we present our statistical object model. Section 4 discusses automatic part selection. Section 5 is dedicated to the second step of model formation and training. Section 6 demonstrates the method through experiments on two datasets: cars and faces.

3 Modeling Objects in Images

Our model is based on the work by Burl et al. [3]. Important differences are that we model the positions of the background parts through a uniform density, while they used a Gaussian with large covariance. The probability distribution of the *number* of background parts, which Burl et al. ignored, is modeled in our case as a Poisson distribution.

3.1 Generative Object Model

We model objects as collections of rigid parts, each of which is detected by a corresponding detector during recognition. The part detection stage therefore transforms an entire image into a collection of parts. Some of those parts might correspond to an instance of the target object class (the *foreground*), while others stem from background clutter or are simply false detections (the *background*). Throughout this paper, the only information associated with an object part is its position in the image and its identity or part *type*. We assume that there are T different types of parts. The positions of all parts extracted from one image can be summarized in a matrix-like form,

$$X^o = \left(\begin{array}{c} x_{11} x_{12}, \ldots, x_{1N_1} \\ x_{21} x_{22}, \ldots, x_{2N_2} \\ \vdots \\ x_{T1} x_{T2}, \ldots, x_{TN_T} \end{array} \right.,$$

where the superscript 'o' indicates that these positions are *observable* in an image, as opposed to being unobservable or *missing*, which will be denoted by 'm.' Thus, the t^{th}

row contains the locations of detections of part type t, where every entry, x_{ij}, is a two-dimensional vector. If we now assume that an object is composed of F different parts,[1] we need to be able to indicate which parts in X^o correspond to the foreground (the object of interest). For this we use the vector $\mathbf{h}$, a set of indices, with $h_i = j,\ j > 0$, indicating that point x_{ij} is a foreground point. If an object part is not contained in X^o, because it is occluded or otherwise undetected, the corresponding entry in $\mathbf{h}$ will be zero. When presented with an unsegmented and unlabeled image, we do not know which parts correspond to the foregound. Therefore, $\mathbf{h}$ is not observable and we will treat it as *hidden* or *missing* data. We call $\mathbf{h}$ a *hypothesis*, since we will use it to hypothesize that certain parts of X^o belong to the foreground object. It is also convenient to represent the positions of any unobserved object parts in a separate vector $\mathbf{x}^m$ which is, of course, hidden as well. The dimension of $\mathbf{x}^m$ will vary, depending on the number of missed parts.

We can now define a generative probabilistic model through the joint probability density

$$p(X^o, \mathbf{x}^m, \mathbf{h}). \tag{1}$$

Note that not only the entries of X^o and $\mathbf{x}^m$ are random variables, but also their dimensions.

3.2 Model Details

In order to provide a detailed parametrization of (1), we introduce two auxiliary variables, $\mathbf{b}$ and $\mathbf{n}$. The binary vector $\mathbf{b}$ encodes information about which parts have been detected and which have been missed or occluded. Hence, $b_f = 1$ if $h_f > 0$ and $b_f = 0$ otherwise. The variable $\mathbf{n}$ is also a vector, where n_t shall denote the number of *background* candidates included in the t^{th} row of X^o. Since both variables are completely determined by $\mathbf{h}$ and the size of X^o, we have $p(X^o, \mathbf{x}^m, \mathbf{h}) = p(X^o, \mathbf{x}^m, \mathbf{h}, \mathbf{n}, \mathbf{b})$. Since we assume independence between foreground and background, and, thus, between $p(\mathbf{n})$ and $p(\mathbf{b})$, we decompose in the following way

$$p(X^o, \mathbf{x}^m, \mathbf{h}, \mathbf{n}, \mathbf{b}) = p(X^o, \mathbf{x}^m | \mathbf{h}, \mathbf{n})\, p(\mathbf{h} | \mathbf{n}, \mathbf{b})\, p(\mathbf{n})\, p(\mathbf{b}). \tag{2}$$

The probability density over the number of background detections can be modeled by a Poisson distribution,

$$p(\mathbf{n}) = \prod_{t=1}^{T} \frac{1}{n_t!} (M_t)^{n_t} e^{-M_t},$$

where M_t is the average number of background detections of type t per image. This conveys the assumption of independence between part types in the background and the idea that background detections can arise at any location in the image with equal probability, independently of other locations. For a discrete grid of pixels, $p(\mathbf{n})$ should be

[1] To simplify notation, we only consider the case where $F = T$. The extension to the general case ($F \geq T$) is straightforward.

modeled as a binomial distribution. However, since we will model the foreground detections over a continuous range of positions, we chose the Poisson distribution, which is the continuous limit of the binomial distribution. Admitting a different M_f for every part type allows us to model the different detector statistics.

Depending on the number of parts, F, we can model the probability $p(\mathbf{b})$ either as an explicit table (of length 2^F) of joint probabilities, or, if F is large, as F independent probabilities, governing the presence or absence of an individual model part. The joint treatment could lead to a more powerful model, e.g., if certain parts are often occluded simultaneously.

The density $p(\mathbf{h}|\mathbf{n}, \mathbf{b})$ is modeled by,

$$p(\mathbf{h}|\mathbf{n}, \mathbf{b}) = \begin{cases} \frac{1}{\prod_{f=1}^{F} N_f^{b_f}} & \mathbf{h} \in \mathcal{H}(\mathbf{b}, \mathbf{n}) \\ 0 & \text{other } \mathbf{h} \end{cases}$$

where $\mathcal{H}(\mathbf{b}, \mathbf{n})$ denotes the set of all hypotheses consistent with $\mathbf{b}$ and $\mathbf{n}$, and N_f denotes the total number of detections of the type of part f. This expresses the fact that all consistent hypotheses, the number of which is $\prod_{f=1}^{F} N_f^{b_f}$, are equally likely in the absence of information on the part locations.

Finally, we use

$$p(X^o, \mathbf{x}^m|\mathbf{h}, \mathbf{n}) = p_{\mathbf{fg}}(\mathbf{z})\, p_{\mathbf{bg}}(\mathbf{x}_{bg}),$$

where we defined $\mathbf{z}^T = (\mathbf{x}^{oT}\, \mathbf{x}^{mT})$ as the coordinates of all foreground detections (observed and missing) and $\mathbf{x}_{bg}$ as the coordinates of all background detections. Here we have made the important assumption that the foreground detections are independent of the background. In our experiments, $p_{\mathbf{fg}}(\mathbf{z})$ is modeled as a joint Gaussian with mean μ and covariance Σ.

Note that, so far, we have modeled only *absolute* part positions in the image. This is of little use, unless the foreground object is in the same position in every image. We can, however, obtain a translation invariant formulation of our algorithm (as used in the experiments in this paper) by describing all part positions relative to the position of one reference part. Under this modification, $p_{\mathbf{fg}}$ will remain a Gaussian density, and therefore not introduce any fundamental difficulties. However, the formulation is somewhat intricate, especially when considering missing parts. Hence, for further discussion of invariance the reader is referred to [3].

The positions of the background detections are modeled by a uniform density,

$$p_{\mathbf{bg}}(\mathbf{x}_{bg}) = \prod_{t=1}^{T} \frac{1}{A^{n_t}},$$

where A is the total image area.

3.3 Classification

Throughout the experiments presented in this paper, our objective is to classify images into the classes "object present" (class $\mathcal{C}_1$) and "object absent" (class $\mathcal{C}_0$). Given the observed data, X^o, the optimal decision—minimizing the expected total classification error—is made by choosing the class with maximum a posteriori probability (MAP approach, see e.g. [7]). It is therefore convenient to consider the following ratio,

$$\frac{p(\mathcal{C}_1|X^o)}{p(\mathcal{C}_0|X^o)} \propto \frac{\sum_{\mathbf{h}} p(X^o, \mathbf{h}|\mathcal{C}_1)}{p(X^o, \mathbf{h}_0|\mathcal{C}_0)}, \tag{3}$$

where $\mathbf{h}_0$ denotes the *null hypothesis* which explains all parts as background noise. Notice that the ratio $\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_0)}$ is omitted, since it can be absorbed into a decision threshold. The sum in the numerator includes all hypotheses, also the null hypothesis, since the object could be present but remain undetected by any part detector. In the denominator, the only consistent hypothesis to explain "object absent" is the null hypothesis.

Although we are here concerned with classification only, our framework is by no means restricted to this problem. For instance, object localization is possible by identifying those foreground parts in an image, which have the highest probability of corresponding to an occurrence of the target object.

4 Automatic Part Selection

The problem of selecting distinctive and well localizeable object parts is intimately related to the method used to detect these parts when the recognition system is finally put to work. Since we need to evaluate a large number of potential parts and thus, detectors, we settled on normalized correlation as an efficient part detection method. Furthermore, extensive experiments lead us to believe that this method offers comparable performance over many more elaborate detection methods.

With correlation based detection, every pattern in a small neighborhood in the training images could be used as a template for a prospective part detector. The purpose of the procedure described here is to reduce this potentially huge set of parts to a reasonable number, such that the model learning algorithm described in the next section can then select a few most useful parts. We use a two-step procedure to accomplish this.

In the first step, we identify *points of interest* in the training images (see Fig. 3). This is done using the interest operator proposed by Förstner [9], which is capable of detecting corner points, intersections of two or more lines, as well as center points of circular patterns. This step produces about 150 part candidates per training image.

A significant reduction of the number of parts can be achieved by the second step of the selection process, which consists in performing vector quantization on the patterns (a similar procedure was used by Leung and Malik in [13]). To this end, we use a standard k-means clustering algorithm [7], which we tuned to produce a set of about 100 patterns. Each of these patterns represents the center of a cluster and is obtained as the average of all patterns in the cluster. We only retain clusters with at least 10 patterns. We impose this limit, since clusters composed of very few examples tend to represent patterns which do not appear in a significant number of training images. Thus, we obtain parts which are averaged across the entire set of training images.

In order to further eliminate redundancies, we remove patterns which are simlilar to others after a small shift (1–2 pixels) in any an arbitrary direction.

Due to the restriction to points of interest the set of remaining patterns exhibits interesting structure, as can be seen in Figure 3. Some parts, such as human eyes, can

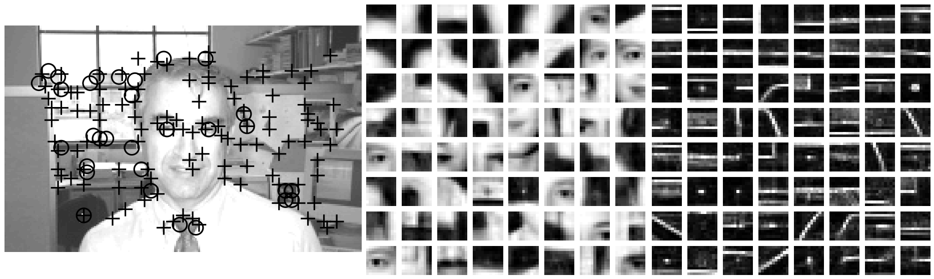

Fig. 3. Points of interest (left) identified on a training image of a human face in cluttered background using Förstner's method. Crosses denote corner-type patterns while circles mark circle-type patterns. A sample of the patterns obtained using k-means clustering of small image patches is shown for faces (center) and cars (right). The car images were high-pass filtered before the part selection process. The total number of patterns selected were 81 for faces and 80 for cars.

be readily identified. Other parts, such as simple corners, result as averages of larger clusters, often containing thousands of patterns.

This procedure dramatically reduces the number of candidate parts. However, at this point, parts corresponding to the background portions of the training images are still present.

5 Model Learning

In order to train an object model on a set of images, we need to solve two problems. Firstly, we need to decide on a small subset of the selected part candidates to be used in the model, i.e. define the *model configuration*. Secondly, we need to learn the parameters underlying the probability densities. We solve the first problem using an iterative, "greedy" strategy, under which we try different configurations. At each iteration, the pdfs are estimated using *expectation maximization* (EM).

5.1 Greedy Model Configuration Search

An important question to answer is with *how many parts* to endow our model. As the number of parts increases, models gain complexity and discriminatory power. It is therefore a good strategy to start with models comprised of few parts and add parts while monitoring the generalization error and, possibly, a criterion penalizing complexity.

If we start the learning process with few parts, say $F = 3$, we are still facing the problem of selecting the best out of N^F possible sets of parts, where N is the number of part candidates produced as described in Sec. 4. We do this iteratively, starting with a random selection. At every iteration, we test whether replacing one model part with a randomly selected one, improves the model. We therefore first estimate all remaining model parameters from the training images, as explained in the next section, and then assess the classification performance on a validation set of positive and negatives examples. If the performance improves, the replacement part is kept. This process is stopped when no more improvements are possible. We might then start over after increasing the total number of parts in the model.

It is possible to render this process more efficient, in particular for models with many parts, by prioritizing parts which have previously shown a good validation performance when used in smaller models.

5.2 Estimating Model Parameters through Expectation Maximization

We now address the problem of estimating the model pdfs with a given set of model parts, from a set of I training images.

Since our detection method relies on the *maximum a posteriori probability* (MAP) principle, it is our goal to model the class conditional densities as accurately as possible. We therefore employ the expectation maximization (EM) algorithm to produce maximum likelihood estimates of the model parameters, $\theta = \{\mu,\ \Sigma,\ p(\mathbf{b}),\ \mathbf{M}\}$. EM is well suited for our problem, since the variables $\mathbf{h}$ and $\mathbf{x}^m$ are missing and must be inferred from the observed data, $\{X_i^o\}$. In standard EM fashion, we proceed by maximizing the likelihood of the observed data,

$$L(\{X_i^o\}|\theta) = \sum_{i=1}^{I} \log \sum_{\mathbf{h}_i} \int p(X_i^o, \mathbf{x}_i^m, \mathbf{h}_i|\theta)\, d\mathbf{x}_i^m,$$

with respect to the model parameters. Since this is difficult to achieve in practice, EM iteratively maximizes a sequence of functions,

$$Q(\tilde{\theta}|\theta) = \sum_{i=1}^{I} E[\log p(X_i^o, \mathbf{x}_i^m, \mathbf{h}_i|\tilde{\theta})],$$

where $E[.]$ refers to the expectation with respect to the posterior $p(\mathbf{h}_i, \mathbf{x}_i^m|X_i^o, \theta)$. Throughout this section, a tilde denotes parameters we are optimizing for, while no tilde implies that the values from the previous iteration are substituted. EM theory [6] guarantees that subsequent maximization of the $Q(\tilde{\theta}|\theta)$ converges to a local maximum of L.

We now derive update rules that will be used in the M-step of the EM algorithm. The parameters we need to consider are those of the Gaussian governing the distribution of the foreground parts, i.e. μ and Σ, the table representing $p(\mathbf{b})$ and the parameter, $\mathbf{M}$, governing the background densities. It will be helpful to decompose Q into four parts, following the factorization in Equation (2).

$$\begin{aligned} Q(\tilde{\theta}|\theta) &= Q_1(\tilde{\theta}|\theta) + Q_2(\tilde{\theta}|\theta) + Q_3(\tilde{\theta}|\theta) + Q_4 \\ &= \sum_{i=1}^{I} E[\log p(\mathbf{n}_i|\theta)] + \sum_{i=1}^{I} E[\log p(\mathbf{b}_i|\theta)] + \sum_{i=1}^{I} E[\log p(X_i^o, \mathbf{x}_i^m|\mathbf{h}_i, \mathbf{n}_i, \theta)] \\ &+ \sum_{i=1}^{I} E[\log p(\mathbf{h}_i|\mathbf{n}_i, \mathbf{b}_i)] \end{aligned}$$

Only the first three terms depend on parameters that will be updated during EM.

Update rule for $\boldsymbol{\mu}$ Since only Q_3 depends on $\tilde{\mu}$, taking the derivative of the expected likelihood yields

$$\frac{\partial}{\partial\tilde{\mu}}Q_3(\tilde{\theta}|\theta) = \sum_{i=1}^{I} E\left[\tilde{\Sigma}^{-1}(\mathbf{z}_i - \tilde{\mu})\right],$$

where $\mathbf{z}^T = (\mathbf{x}^{oT}\,\mathbf{x}^{mT})$ according to our definition above. Setting the derivative to zero yields the following update rule

$$\tilde{\mu} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{z}_i].$$

Update rule for $\boldsymbol{\Sigma}$ Similarly, we obtain for the derivative with respect to the inverse covariance matrix

$$\frac{\partial}{\partial\tilde{\Sigma}^{-1}}Q_3(\tilde{\theta}|\theta) = \sum_{i=1}^{I} E\left[\frac{1}{2}\tilde{\Sigma} - \frac{1}{2}(\mathbf{z}_i - \tilde{\mu})(\mathbf{z}_i - \tilde{\mu})^T\right].$$

Equating with zero leads to

$$\tilde{\Sigma} = \frac{1}{I}\sum_{i=1}^{I} E[(\mathbf{z}_i - \tilde{\mu})(\mathbf{z}_i - \tilde{\mu})^T] = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{z}_i\mathbf{z}_i^T] - \tilde{\mu}\tilde{\mu}^T.$$

Update rule for $\boldsymbol{p(\mathbf{b})}$ To find the update rule for the 2^F probability masses of $p(\mathbf{b})$, we need to consider $Q_2(\tilde{\theta}|\theta)$, the only term depending on these parameters. Taking the derivative with respect to $\tilde{p}(\bar{\mathbf{b}})$, the probability of observing one *specific* vector, $\bar{\mathbf{b}}$, we obtain

$$\frac{\partial}{\partial\tilde{p}(\bar{\mathbf{b}})}Q_2(\tilde{\theta}|\theta) = \sum_{i=1}^{I} \frac{E[\delta_{\mathbf{b},\bar{\mathbf{b}}}]}{\tilde{p}(\bar{\mathbf{b}})},$$

where δ shall denote the Kronecker delta. Imposing the constraint $\sum_{\bar{\mathbf{b}}} \tilde{p}(\bar{\mathbf{b}}) = 1$, for instance by adding a Lagrange multiplier term, we find the following update rule for $\tilde{p}(\bar{\mathbf{b}})$,

$$\tilde{p}(\bar{\mathbf{b}}) = \frac{1}{I}\sum_{i=1}^{I} E[\delta_{\mathbf{b},\bar{\mathbf{b}}}].$$

Update rule for $\mathbf{M}$ Finally, we notice that $Q_1(\tilde{\theta}|\theta)$ is the only term containing information about the mean number of background points per part type M_f. Differentiating $Q_1(\tilde{\theta}|\theta)$ with respect to $\tilde{\mathbf{M}}$ we find,

$$\frac{\partial}{\partial\tilde{\mathbf{M}}}Q_1(\tilde{\theta}|\theta) = \sum_{i=1}^{I} \frac{E[\mathbf{n}_i]}{\tilde{\mathbf{M}}} - I.$$

Equating with zero leads to the intuitively appealing result

$$\tilde{\mathbf{M}} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{n}_i].$$

Computing the Sufficient Statistics All update rules derived above are expressed in terms of *sufficient statistics*, $E[\mathbf{z}]$, $E[\mathbf{z}\mathbf{z}^T]$, $E[\delta_{\mathbf{b},\bar{\mathbf{b}}}]$ and $E[\mathbf{n}]$ which are calculated in the E-step of the EM algorithm. We therefore consider the *posterior* density,

$$p(\mathbf{h}_i, \mathbf{x}_i^m | X_i^o, \theta) = \frac{p(\mathbf{h}_i, \mathbf{x}_i^m, X_i^o|\theta)}{\sum_{\mathbf{h}_i \in \mathcal{H}_\mathbf{b}} \int p(\mathbf{h}_i, \mathbf{x}_i^m, X_i^o|\theta)\, d\mathbf{x}_i^m}$$

The denominator in the expression above, which is equal to $p(X_i^o)$, is calculated by explicitly generating and summing over all hypotheses, while integrating out[2] the missing data of each hypothesis. The expectations are calculated in a similar fashion: $E[\delta_{\mathbf{b},\bar{\mathbf{b}}}]$ is calculated by summing only over those hypotheses consistent with $\bar{\mathbf{b}}$ and dividing by $p(X_i^o)$. Similarly, $E[\mathbf{n}_i]$ is calculated by averaging $\mathbf{n}(\mathbf{h})$ over all hypotheses. The case of $E[\mathbf{z}]$ is slightly more complicated. For every hypothesis we regroup $\mathbf{z}^T = (\mathbf{x}^{oT}\ \mathbf{x}^{mT})$ and note that $E[\mathbf{x}^o] = \mathbf{x}^o$. For $E[\mathbf{x}^m]$ one needs to calculate,

$$\int \mathbf{x}^m\, G(\mathbf{z}|\mu, \Sigma)\, d\mathbf{x}^m = \mu^m + \Sigma^{mo}\Sigma^{oo-1}(\mathbf{x}^o - \mu^o),$$

where we defined,

$$\mu = \begin{pmatrix} \mu^o \\ \mu^m \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma^{oo} & \Sigma^{om} \\ \Sigma^{mo} & \Sigma^{mm} \end{pmatrix}$$

Summing over all hypothesis and dividing by $p(X^o)$ establishes the result. Finally we need to calculate

$$E[\mathbf{z}\mathbf{z}^T] = \begin{pmatrix} \mathbf{x}^o\mathbf{x}^{oT} & \mathbf{x}^o E[\mathbf{x}^m]^T \\ E[\mathbf{x}^m]\mathbf{x}^{oT} & E[\mathbf{x}^m\mathbf{x}^{mT}] \end{pmatrix}.$$

Here, only the part $E[\mathbf{x}^m\mathbf{x}^{mT}]$ has not yet been considered. Integrating out the missing dimensions, $\mathbf{x}^m$, now involves,

$$\int \mathbf{x}^m\mathbf{x}^{mT}\, G(\mathbf{z}|\mu, \Sigma)\, d\mathbf{x}^m = \Sigma^{mm} - \Sigma^{mo}\Sigma^{oo-1}\Sigma^{moT} + E[\mathbf{x}^m]E[\mathbf{x}^m]^T.$$

Looping through all possible hypotheses and dividing by $p(X_i^o)$ again provides the desired result. This concludes the E-step of the EM algorithm.

6 Experiments

In order to validate our method, we tested the performance, under the classification task described in Sect. 3.3, on two data sets: images of rear views of cars and images of human faces. As mentioned in Sec. 3, the experiments described below have been performed with a translation invariant extension of our learning method. All parameters of the learning algorithm were set to the same values in both experiments.

[2] Integrating out dimensions of a Gaussian is simply done by deleting the means and covariances of those dimensions and multiplying by the suitable normalization constant.

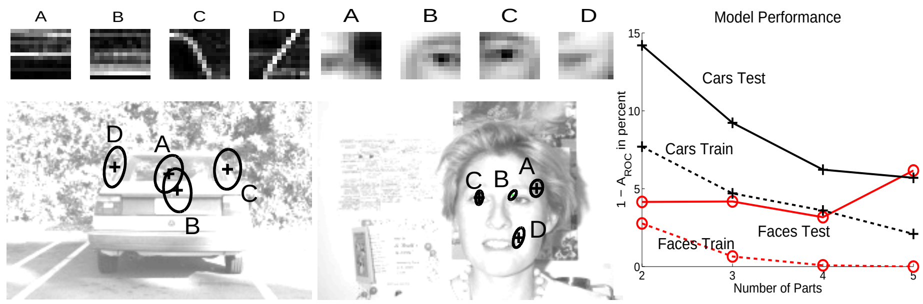

Fig. 4. Results of the learning experiments. On the left we show the best performing car model with four parts. The selected parts are shown on top. Below, ellipses indicating a one-std deviation distance from the mean part positions, according to the foreground pdf have been superimposed on a typical test image. They have been aligned by hand for illustrative purposes, since the models are translation invariant. In the center we show the best four-part face model. The plot on the right shows average training and testing errors measured as $1 - A_{ROC}$, where A_{ROC} is the area under the corresponding ROC curve. For both models, one observes moderate overfitting. For faces, the smallest test error occurs at 4 parts. Hence, for the given amount of training data, this is the optimal number of parts. For cars, 5 or more parts should be used.

Training and Test Images For each of the two object classes we took 200 images showing a target object at an arbitrary location in cluttered background (Fig. 1, left). We also took 200 images of background scenes from the same environment, excluding the target object (Fig. 1, right). No images were discarded by hand prior to the experiments. The face images were taken indoors as well as outdoors and contained 30 different people (male and female). The car images were taking on public streets and parking lots where we photographed vehicles of different sizes, colors and types, such as sedans, sport utility vehicles, and pick-up trucks. The car images were high-pass filtered in order to promote invariance with respect to the different car colors and lighting conditions. All images were taken with a digital camera; they were converted to a grayscale representation and downsampled to a resolution of 240×160 pixels.

Each image set was randomly split into two disjoint sets of training and test images. In the face experiment, no single person was present in both sets.

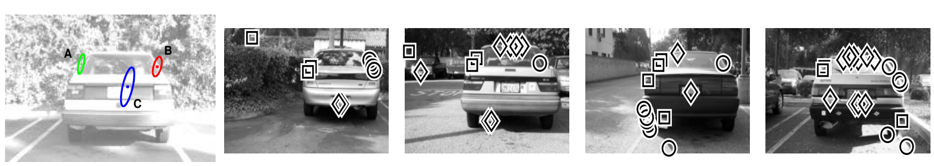

Fig. 5. Multiple use of parts: The three-part model on the left correctly classified the four images on the right. Part labels are: ◯ = ‘A’, ☐ = ‘B’, ◇ = ‘C’. Note that the middle part (C) exhibits a high variance along the vertical direction. It matches several locations in the images, such as the bumper, license plate and roof. In our probabilistic framework, no decision is made as to the *correct match*. Rather, evidence is accumulated across all possible matches.

Automatically Selected Parts Parts were automatically selected according to the procedure described in Sec. 4. The Förstner interest operator was applied to the 100 unlabeled and unsegmented training images containing instances of the target object class. We performed vector quantization on grayscale patches of size 11×11 pixels, extracted around the points of interest. A different set of patterns was produced for each object class, as shown in Figure 3.

Model Learning We learned models with 2, 3, 4, and 5 parts for both data sets. Since the greedy configuration search as well as the EM algorithm can potentially converge to local extrema, we learned each model up to 100 times, recording the average classification error.

All models were learned from the entire set of selected parts. Hence, no knowledge from the training of small models about the usefulness of parts was applied during the training of the larger models. This was done in order to investigate to what extent the same parts were chosen across model sizes.

We found the EM algorithm to converge in about 100 iterations, which corresponds to less than 10s for a model with two parts and about 2min for a five-part model. We used a Matlab implementation with subroutines written in 'C' and a PC with $450\mathrm{MHz}$ Pentium II processor. The number of different part configurations evaluated varied from about 80–150 (2 parts) to 300–400 (5 parts).

Results Instead of classifying every image by applying a fixed decision threshold according to (3), we computed receiver operating characteristics (ROCs) based on the ratio of posterior probabilities. In order to reduce sensitivity to noise due to the limited number of training images and to average across all possible values for the decision threshold, we used the area under the ROC curve as a measure of the classification performance driving the optimization of the model configuration. In Figure 4, we show two learned models as well as this error measure as a function of the number of parts.

Examples of successfully and wrongly classified images from the test sets are shown in Fig. 6.

When inspecting the models produced, we were able to make several interesting observations. For example, in the case of faces, we found confirmation that eye corners are very good parts. But our intuition was not always correct. Features along the hairline turned out to be very stable, while parts containing noses were almost never used in the models.

Before we introduced a high-pass filter as a preprocessing step, the car models concentrated on the dark shadow underneath the cars as most stable feature. Researchers familiar with the problem of tracking cars on freeways confirmed that the shadow is often the easiest way to detect a car.

Oftentimes the learning algorithm took advantage of the fact that some part detectors respond well at multiple locations on the target objects (Fig. 5). This effect was most pronounced for models with few parts. It would be difficult to predict and exploit this behavior when building a model "by hand."

Since we ran the learning process many times, we were able to assess the likelihood of converging to local extrema. For each size, models with different part choices were produced. However, each choice was produced at least a few times. Regarding the EM

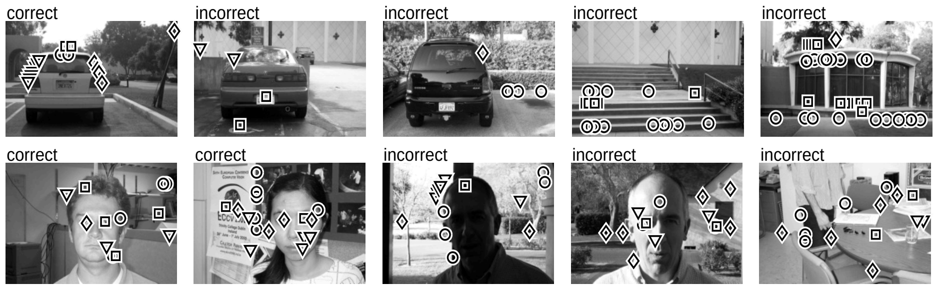

Fig. 6. Examples of correctly and incorrectly classified images from the test sets, based on the models in Fig. 4. Part labels are: $\bigcirc$ = 'A', $\square$ = 'B', $\diamond$ = 'C', $\bigtriangledown$ = 'D'. 100 foreground and 100 background images were classified in each case. The decision threshold was set to yield equal error rate on foreground and background images. In the case of faces, 93.5% of all images were classified correctly, compared to 86.5% in the more difficult car experiment.

algorithm itself, we only observed one instance, where a given choice of parts resulted in several different classification performances. This leads us to conclude that the EM algorithm is extremely unlikely to get stuck in a local maximum.

Upon inspection of the different part types selected across model sizes, we noticed that about half of all parts chosen at a particular model size were also present in smaller models. This suggests that initializing the choice of parts with parts found in well performing smaller models is a good strategy. However, one should still allow the algorithm to also choose from parts not used in smaller models.

7 Discussion and Future Work

We have presented ideas for learning object models in an unsupervised setting. A set of unsegmented and unlabeled images containing examples of objects amongst clutter is supplied; our algorithm automatically selects distinctive parts of the object class, and learns the joint probability density function encoding the object's appearance. This allows the automatic construction of an efficient object detector which is robust to clutter and occlusion.

We have demonstrated that our model learning algorithm works successfully on two different data sets: frontal views of faces and rear views of motor-cars. In the case of faces, discrimination of images containing the desired object vs. background images exceeds 90% correct with simple models composed of 4 parts. Performance on cars is 87% correct. While training is computationally expensive, detection is efficient, requiring less than a second in our C-Matlab implementation. This suggests that training should be seen as an off-line process, while detection may be implemented in real-time.

The main goal of this paper is to demonstrate that it is feasible to learn object models directly from unsegmented cluttered images, and to provide ideas on how one may do so. Many aspects of our implementation are suboptimal and susceptible of improvement. To list a few: we implemented the part detectors using normalized correlation. More sophisticated detection algorithms, involving multiscale image processing, multiorientation-multiresolution filters, neural networks etc. should be considered and tested. Moreover, in our current implementation only part of the information supplied by the detectors, i.e. the candidate part's location, is used; the scale and orientation of the image patch, parameters describing the appearance of the patch, as well as its likelihood, should be incorporated. Our interest operator as well as the unsupervised clustering of the parts have not been optimized in any respect; the choice of the algorithms deserves further scrutiny as well. An important aspect where our implementation falls short of generality is invariance: the models we learned and tested are translation invariant, but not rotation, scale or affine invariant. While there is no conceptual limit to this generalization, the straightforward implementation of the EM algorithm in the rotation and scale invariant case is slow, and therefore impractical for extensive experimentation.

Acknowledgements This work was funded by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726), and an NSF National Young Investigator Award to P.P. (NSF9457618). M. Welling was supported by the Sloan Foundation.

We are also very grateful to Rob Fergus for helping with collecting the databases and to Thomas Leung, Mike Burl, Jitendra Malik and David Forsyth for many helpful comments.

References

1. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
2. M.C. Burl, T.K. Leung, and P. Perona. "Face Localization via Shape Statistics". In *Int Workshop on Automatic Face and Gesture Recognition*, 1995.
3. M.C. Burl, T.K. Leung, and P. Perona. "Recognition of Planar Object Classes". In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1996.
4. M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *proc. ECCV'98*, pages 628–641, 1998.
5. T.F. Cootes and C.J. Taylor. "Locating Objects of Varying Shape Using Statistical Feature Detectors". In *European Conf. on Computer Vision*, pages 465–474, 1996.
6. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society* **B**, 39:1–38, 1976.
7. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., 1973.
8. G.J. Edwards, T.F.Cootes, and C.J.Taylor. Face recognition using active appearance models. In *Proc. 5^{th} Europ. Conf. Comput. Vision, H. Burkhardt and B. Neumann (Eds.), LNCS-Series Vol. 1406–1407, Springer-Verlag*, pages 581–595, 1998.
9. R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision II*. Addison-Wesley, 1993.
10. M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P. Wurtz, and W. Konen. "Distortion Invariant Object Recognition in the Dynamic Link Architecture". *IEEE Trans. Comput.*, 42(3):300–311, Mar 1993.
11. T.K. Leung, M.C. Burl, and P. Perona. "Finding Faces in Cluttered Scenes using Random Labeled Graph Matching". *Proc. 5th Int. Conf. Computer Vision*, pages 637–644, June 1995.
12. T.K. Leung, M.C. Burl, and P. Perona. Probabilistic affine invariants for recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 678–684, 1998.
13. T.K. Leung and J. Malik. Reconizing surfaces using three-dimensional textons. In *Proc. 7th Int. Conf. Computer Vision*, pages 1010–1017, 1999.
14. K. N. Walker, T. F. Cootes, and C. J. Taylor. Locating salient facial features. In *Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, 1998.
15. A.L. Yuille. "Deformable Templates for Face Recognition". *J. of Cognitive Neurosci.*, 3(1):59–70, 1991.