Semi-supervised learning and recognition of object classes

[1]R. Fergus, [2]P. Perona, and [1]A. Zisserman

[1] Dept. of Engineering Science
University of Oxford
Parks Road, Oxford
OX1 3PJ, U.K.
[2] Dept. of Electrical Engineering,
California Institute of Technology,
MC 136-93, Pasadena,
CA 91125, U.S.A.

Draft Version: November 14, 2004 – 6:29 P.M.

Abstract. We present a method for learning object classes without supervision: learning is scale and translation-invariant, does not require alignment nor correspondence between the training images, and is robust to clutter and occlusion. Class models are probabilistic constellations of parts, and their parameters are estimated by maximizing the likelihood of the training data. The appearance of the parts, as well as their mutual position, relative scale and probability of detection are explicitly described in the model. We investigate two models differing in their representation of the part's configuration. Parts may be regions, representing object appearance patches, or curves representing the local object shape.

Recognition takes place in two stages. First, a feature-finder identifies promising locations for the models' part. Second, the category model is used to compare the likelihood that the observed features are generated by the model, or is generated by background clutter. The flexible nature of the model is demonstrated by excellent results over six diverse object classes including geometrically constrained classes (e.g. faces, cars) and flexible objects (such as animals). We also describe an application of a learnt model to object based retrieval of frames from the situation comedy 'Fawlty Towers'.

Table of Contents

Semi-supervised learning and recognition of object classes 1
[1]*R. Fergus,* [2]*P. Perona,* [1]*A. Zisserman*

1 Introduction

Representation, detection and learning are the main issues that need to be tackled in designing a visual system for recognizing object classes. The first challenge is coming up with models that can capture the 'essence' of a category, i.e. what is common to the objects that belong to it, and yet are flexible enough to accommodate object variability (e.g. presence/absence of distinctive parts such as mustache and glasses, variability in overall configuration, changing appearance due to lighting conditions, viewpoint etc). The challenge of detection is defining metrics and inventing algorithms that are suitable for matching models to images efficiently in the presence of occlusion and clutter. Learning is the ultimate challenge. If we wish to be able to design visual systems that can recognize, say, 10,000 object classes, then effortless learning is a crucial step. This means that those training steps that require a human operator (e.g. collection of good quality training exemplars of the class; elimination of clutter; correspondence and scale normalization of the training examples) should be reduced to a minimum or eliminated.

The problem of describing and recognizing classes, as opposed to specific objects (e.g. [10,16,19]), has recently gained some attention in the machine vision literature [1,2,4,5,7,9,11,14,18,21,22,24,28] with an emphasis on the detection of faces [20,23,25], handwritten characters [3,13] and automobiles [14,22]. There is broad agreement on the issue of representation: object classes are represented as collection of features, or parts, each part has a distinctive appearance and spatial position. Different authors vary widely on the details: the number of parts they envisage (from a few to thousands of parts), how these parts are detected and represented, how their position is represented, whether the variability in part appearance and position is represented explicitly or is implicit in the details of the matching algorithm. The issue of learning is perhaps the least well understood. Most authors rely on manual steps to eliminate background clutter and normalize the pose of the training examples. Recognition often proceeds by an exhaustive search over image position and scale. In exploring only a few classes, many of the challenges and difficulties in tacking thousands of classes have yet to be encountered.

We focus our attention on the probabilistic approach proposed by Burl *et al.* [5] which models objects as a constellations of parts. This approach presents several advantages: the model explicitly accounts for configuration variations and for the randomness in the presence/absence of features due to occlusion and detector errors. It accounts explicitly for image clutter. It yields principled and efficient detection methods. Weber *et al.* [27,28] proposed a maximum likelihood unsupervised learning algorithm for the "constellation model" which successfully learns object classes from cluttered data with minimal human intervention. We propose here a number of substantial improvement to the constellation model and to its maximum likelihood learning algorithm. First: we model configuration variability using two different models, a "fully connected" model and a "star" model. Second, while Burl *et al.* and Weber *et al.* explicitly model configuration variability, they do not model the variability of appearance. We extend their

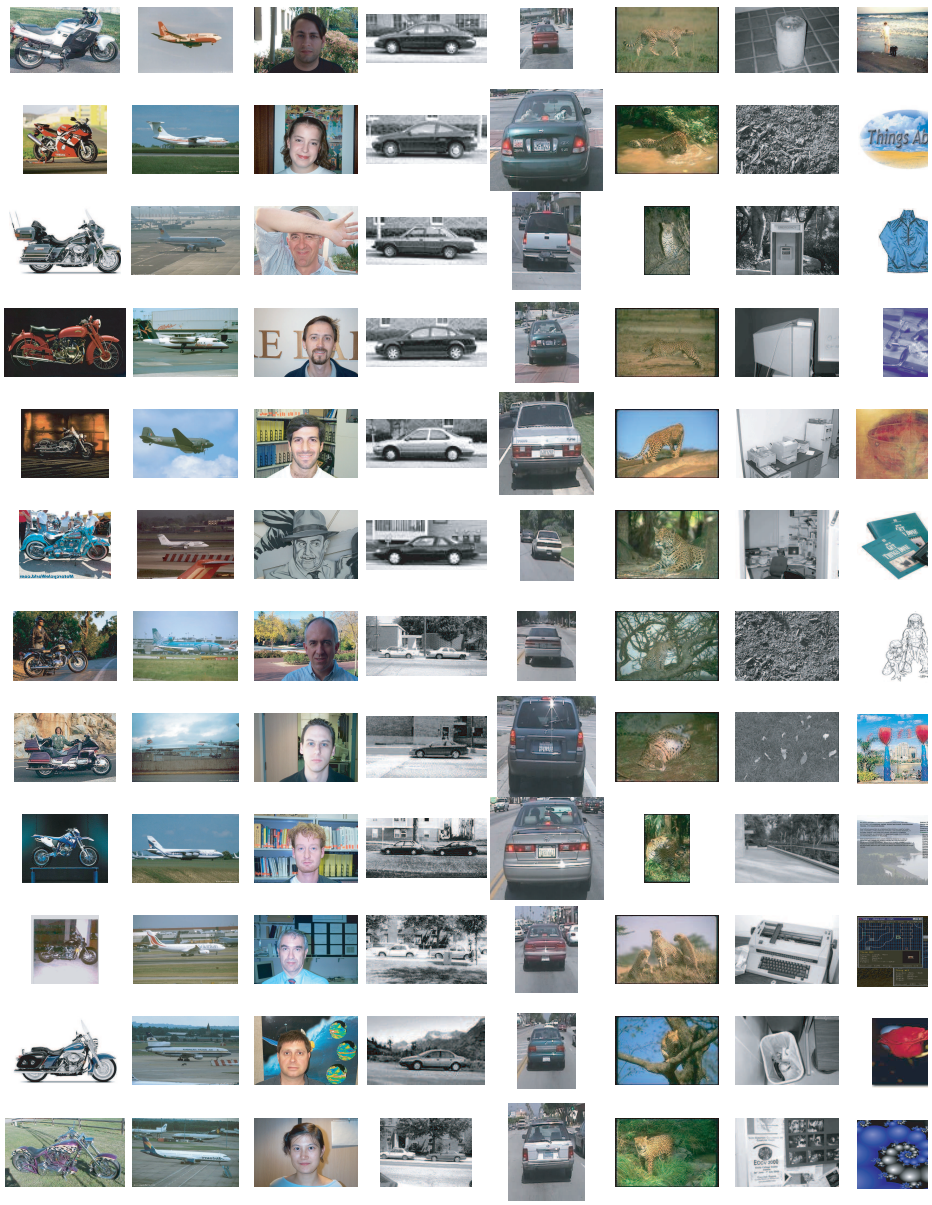


Fig. 1. Some sample images from the datasets. Note the large variation in scale in, for example, the cars (rear) database. These datasets are from both http://www.vision.caltech.edu/html-files/archive.html and http://www.robots.ox.ac.uk/∼vgg/data/, except for the Cars (Side) from (http://l2r.cs.uiuc.edu/∼cogcomp/index_research.html) and Spotted Cats from the Corel Image library. A Powerpoint presentation of the figures in this chapter can be found at http://www.robots.ox.ac.uk/∼vgg/presentations.html

model to take this aspect into account. Third, appearance here is learnt simultaneously with configuration, whereas in their work the appearance of a part is fixed before configuration learning. Fourth: they use correlation to detect their parts. We substitute their front end with an interest operator, which detects regions and their scale in the manner of [15,17], and a curve detector. Fifth, Weber *et al.* did not experiment extensively with scale-invariant learning, most of their training sets are collected in such a way that the scale is approximately normalized. We extend their learning algorithm so that new object classes may be learnt efficiently, without supervision, from training sets where the object examples have large variability in scale. A final contribution is experimenting with a number of new image datasets to validate the overall approach over several object classes. Examples images from these datasets are shown in figure 1.

In summary, the outcome is that an object category constellation model can be learnt in a *semi-supervised* manner from a set of training examples: it is only necessary that the training examples contain instances of the object category, the position and segmentation of the instance in each image are not required. Furthermore, both learning and recognition are translation and scale invariant. In recognition the localization of the object is determined in the image.

The aim of this chapter is to describe our algorithm in sufficient detail to make implementation possible as well as giving an insight into its design. In section 2 we describe the structure of our probabilistic object model and feature detectors and their representation. In section 3 we show how to estimate the model's parameters given a set of training images. Section 4 describes the use of the model in recognition. Our approach is then tested on a wide variety of data in section 5, including an assessment of recognition and confusion performance, and an application to image retrieval. Finally, conclusions are drawn in section 6.

2 Model structure

As in Burl *et al.* and Weber *et al.* [5,26,27,28] an object model consists of a configuration of parts. Each part has an appearance, relative scale and a probability of being occluded or not, or erroneously detected in the background clutter. The entire model is generative and probabilistic, so appearance, scale, configuration of parts, and occlusion are all modeled by probability density functions, which here are Gaussians. The process of learning an object category is one of first detecting regions and their scales, and then estimating the parameters of the above densities from these regions, such that the model gives a maximum-likelihood description of the training data. Recognition is performed on a query image by again first detecting regions and their scales, and then evaluating the regions in a Bayesian manner, using the model parameters estimated in the learning. Note that parts refers to model, while features refer to detection in the image.

The model is best explained by first considering recognition. We have learnt a generative object class model, with P parts and parameters θ_{fg}. We also assume that all non-object images can also be modeled by a background with a single,

fixed, set of parameters θ_{bg}. We are then presented with a new image and we must decide if it contains an instance of our object class or not. In this query image we have identified N interesting features with locations $\mathbf{X}$, scales $\mathbf{S}$, and appearances $\mathbf{A}$. We now make a decision as to the presence/absence of the object by comparing the ratio of class posterior densities, R, to a threshold T:

$$R = \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \tag{1}$$

$$= \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{Object})\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{No object})\,p(\text{No object})} \approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta_{fg})\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})\,p(\text{No object})} \tag{2}$$

The last expression is an approximation since we represent the class with its (imperfect) model, parameterized by θ. The ratio of the priors may be estimated from the training set or set by hand (usually to 1).

Since our model only has P (typically 3-7) parts but there are N (up to 30) features in the image, we use an indexing variable $\mathbf{h}$ (as introduced in [5]) which we call a *hypothesis*. $\mathbf{h}$ is a vector of length P, where each entry is between 0 and N, which allocates a particular feature to a model part. The unallocated features are assumed to be part of the background, with 0 indicating the part is unavailable (e.g. because of occlusion). The set H is all valid allocations of features to the parts; consequently $|H|$ is $O(N^P)$. Computing R in (1) requires the calculation of the ratio of the two likelihood functions. In order to do this, the likelihoods are factored as follows:

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta_{fg}) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\,\theta_{fg}) \tag{3}$$

$$= \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta_{fg})}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta_{fg})}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta_{fg})}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta_{fg})}_{Other} \tag{4}$$

If we believe no object to be present, then all features in the image belong to the background. Thus we only have one possible hypothesis: $\mathbf{h}_0 = \mathbf{0}$, the null hypothesis. So the likelihood in this case becomes:

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta_{bg}) = p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h}_0,\theta_{bg})p(\mathbf{X}|\mathbf{S},\mathbf{h}_0,\theta_{bg})p(\mathbf{S}|\mathbf{h}_0,\theta_{bg})p(\mathbf{h}_0|\theta_{bg}) \tag{5}$$

As we will see below, $p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta_{bg})$ is a constant for a given image. This simplifies the computation of the likelihood ratio in (1), since $p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta_{bg})$ can be moved inside the summation over all hypotheses in (3), to cancel with the foreground terms.

We now look at each of the likelihood terms and derive their actual form. The likelihood terms model not only the properties of the features assigned to the models parts (the foreground) but also the statistics of features in the background of the image (those not picked out by the hypothesis). It will be helpful to define the following notation: let $\mathbf{d}$ be a binary vector giving the state of occlusion for each part, i.e. $d_p = 1$ if part p is present and $d_p = 0$ if absent), $n_{fg} = sum(\mathbf{d})$ (the number of foreground features under the current hypothesis) and $n_{bg} = N - n_{fg}$ (the number of background features).

2.1 Appearance

Here we describe the form of $p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)$ which is the appearance term of the object likelihood. We simplify the expression to $p(\mathbf{A}|\mathbf{h},\theta)$ since given the detected features, we assume their appearance and location to be independent. Each feature's appearance is represented as a point in an appearance space, defined below. Each part p has a Gaussian density within this space, with mean and covariance parameters $\theta^{app}_{fg,p} = \{\mathbf{c}_p, V_p\}$ which is independent of other parts' densities. The background model has fixed parameters $\theta^{app}_{bg} = \{\mathbf{c}_{bg}, V_{bg}\}$. Both V_p and V_{bg} are assumed to be diagonal. The appearance density is computed over all features: each feature selected by the hypothesis is evaluated under the appropriate part density while all features not selected by the hypothesis are evaluated under the background density:

$$p(\mathbf{A}|\mathbf{h},\theta_{fg}) = \prod_{p=1}^{P} \mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_p, V_p)^{d_p} \prod_{j=1, j\notin\mathbf{h}}^{N} \mathrm{G}(\mathbf{A}(j)|\mathbf{c}_{bg}, V_{bg}) \tag{6}$$

where G is the Gaussian distribution, and d_p is the pth entry of the vector $\mathbf{d}$, i.e. $d_p = \mathbf{d}(p)$. If no object is present, then all features are evaluated under the background density:

$$p(\mathbf{A}|\mathbf{h}_0,\theta_{bg}) = \prod_{j=1}^{N} \mathrm{G}(\mathbf{A}(j)|\mathbf{c}_{bg}, V_{bg}) \tag{7}$$

As $p(\mathbf{A}|\mathbf{h}_0,\theta_{bg})$ is a constant and so is not dependent on $\mathbf{h}$, so we can cancel terms between (6) and (7) when computing the likelihood ratio in (1):

$$\frac{p(\mathbf{A}|\mathbf{h},\theta_{fg})}{p(\mathbf{A}|\mathbf{h}_0,\theta_{bg})} = \prod_{p=1}^{P} \left(\frac{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_p, V_p)}{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_{bg}, V_{bg})} \right)^{d_p} \tag{8}$$

So the appearance of each feature in the hypothesis is evaluated under foreground and background densities and the ratio taken. If the part is occluded, the ratio is 1 ($d_p = 0$).

2.2 Configuration

Here we describe the form of $p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)$ which is the configuration term of the object likelihood. We investigate two models: the first is a *fully connected* model, as previously studied by Weber *et al.* and Burl *et al.* [5,28], where the configuration is represented by a joint Gaussian density of the locations of features within a hypothesis, once they have been transformed into a scale and translation-invariant space. This representation allows the modeling of both inter and intra part variability: interactions between the parts (both attractive and repulsive) as well as uncertainty in location of the part itself. The second is a *star model* where only the Gaussian density between parts and a distinguished landmark part are represented.

In both cases translation invariance is achieved by using the location of the feature assigned to the landmark (this landmark is the first non-occluded part in the fully connected model). We then model the configuration of the remaining features in the hypothesis relative to this landmark feature. Scale invariance is achieved by using the scale of the landmark part to normalize the locations of the other features in the constellation. This approach avoids an exhaustive search over scale that other methods use. If the index of the landmark part is l, then the landmark feature's location is $\mathbf{X}(h_l)$ and its scale is $S(h_l)$.

$\mathbf{X}(\mathbf{h})$ is a $2P$ vector holding the x and y coordinates of each feature in hypothesis h, i.e. $\mathbf{X}(\mathbf{h}) = \{x_{h_1}, \ldots, x_{h_P}, y_{h_1}, \ldots, y_{h_P}\}$. To obtain translation invariance, we subtract the location of the landmark from $\mathbf{X}(\mathbf{h})$: $\mathbf{X}^*(\mathbf{h}) = \{x_{h_1} - x_{h_l}, \ldots, x_{h_P} - x_{h_l}, y_{h_1} - y_{h_l}, \ldots, y_{h_P} - y_{h_l}\}$. A scale invariant representation is obtained by dividing through by $S(h_l)$: $\mathbf{X}^{**}(\mathbf{h}) = \frac{X^*(\mathbf{h})}{S(h_l)}$.

Fully connected model: We model $\mathbf{X}^{**}(\mathbf{h})$ with a Gaussian density which has parameters $\theta_{fg}^{shape} = \{\boldsymbol{\mu}, \Sigma\}$. Since any of the P parts can act as the landmark, $\boldsymbol{\mu}, \Sigma$ are in fact a set of P $\boldsymbol{\mu}_l$'s and Σ_l's to evaluate $\mathbf{X}^{**}(\mathbf{h})$ with. However, the set members are equivalent to one another since changing landmark just involves a translation of $\boldsymbol{\mu}$ and the equivalent transformation of Σ (a referral of variances between the old and new landmark).

Due to translation invariance, $\boldsymbol{\mu}_l$ is a $2(P-1)$ vector (x and y coordinates of the non-landmark parts). Correspondingly, Σ_l is a $2(P-1) \times 2(P-1)$ matrix. Note that, unlike appearance whose covariance matrices V_p, V_{bg} are diagonal, Σ_l is a full matrix. All features not included in the hypothesis are considered as arising from the background. The model for the background assumes features to be spread uniformly over the image (which has area α), with locations independent of the foreground locations. We also assume that the landmark feature can occur anywhere in the image, so it's location is modeled by a uniform density of $1/\alpha$.

$$p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \theta_{fg}) = \left(\frac{1}{\alpha}\mathrm{G}(\mathbf{X}^{**}(\mathbf{h})|\boldsymbol{\mu}_l, \Sigma_l)\right)\left(\frac{1}{\alpha}\right)^{n_{bg}} \tag{9}$$

If a part is occluded then we marginalize it out, which for a Gaussian entails deleting the appropriate dimensions from the mean and covariance matrix. See [26] for more details.

If no object is present, then all detections are in the background and are consequently modeled by a uniform distribution:

$$p(\mathbf{X}|\mathbf{S}, \mathbf{h}_0, \theta_{bg}) = \left(\frac{1}{\alpha}\right)^N \tag{10}$$

Again, this is a constant, so we can cancel between (9) and (10) for the likelihood ratio in (1) to give:

$$\frac{p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \theta_{fg})}{p(\mathbf{X}|\mathbf{S}, \mathbf{h}_0, \theta_{bg})} = \mathrm{G}(\mathbf{X}^{**}(\mathbf{h})|\boldsymbol{\mu}_l, \Sigma_l)\, \alpha^{n_{fg}-1} \tag{11}$$

Star model: While the joint Gaussian model on the part locations gives the most thorough description, it makes all parts dependent on one another. The EM learning scheme, described in Section 3, requires the computation of the posterior density on $\mathbf{h}$: $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$. The inter-dependency of the parts makes the calculation of this density $O(N^P)$. Despite the use of efficient heuristics to give accurate approximations, it becomes computationally intractable for $P > 7$ and $N > 30$.

A way around this is to reduce the dependencies in the model by only conditioning on a single model part, as illustrated in Figure 2. Under this model the non-landmark parts are independent of one another given the landmark. In graphical model terms, this is a tree of depth one, with the landmark part being the root node. We call this the "star" model.

In the star model the joint probability of the configuration aspect of the model may be factored as:

$$p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \theta) = p(\mathbf{x}_L|h_L) \prod_{j \neq L} p(\mathbf{x}_j|\mathbf{x}_L, s_L, h_j, \theta_j) \tag{12}$$

where $\mathbf{x}_j$ is the position of part j and L is the landmark part. We adopt a Gaussian model on the relative position between the parts $p(\mathbf{x}_j|\mathbf{x}_L, s_L, h_j, \theta_j)$. The reduced dependencies of this model mean that the marginalization is $O(N^2P)$, enabling us to cope with far larger N and P in learning and recognition. One drawback however is that the landmark part must always be present, which may not always be the case, leading to artificially large configuration variances.

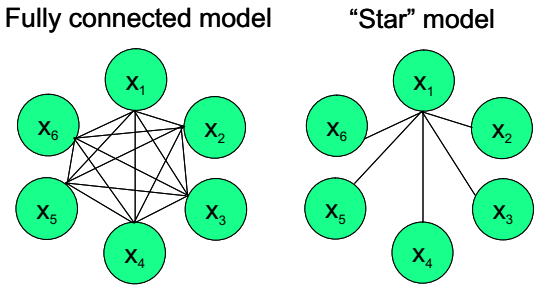


Fig. 2. (a) Fully-connected six part configuration model. Each node is a model part while the edges represent the dependencies between parts. **(b)** A six part Star model. The former has complexity $O(N^P)$ while the latter has complexity $O(N^2P)$

2.3 Relative scale

Here we describe the form of $p(\mathbf{S}|\mathbf{h}, \theta)$ which is the relative scale term of the object likelihood. This term has the same structure as the configuration term.

The scale of parts relative to the scale of the landmark feature is modeled by a Gaussian density in log space which has parameters $\theta_{fg}^{scale} = \{\mathbf{t}, U\}$. Again, since the landmark feature could belong to any of the P parts, these parameters are really a set of equivalent $\mathbf{t}_l, U_l$'s. The parts are assumed to be independent of one another, thus U_l is a diagonal $(P-1) \times (P-1)$ matrix, with $\mathbf{t}_l$ being a $(P-1)$ vector. The background model assumes a uniform distribution over scale (within a range r).

$$p(\mathbf{S}|\mathbf{h}, \theta_{fg}) = \left(\frac{1}{r} \mathrm{G}(\log \mathbf{S}^*(\mathbf{h})|\mathbf{t}_l, U_l)\right) \left(\frac{1}{r}\right)^{n_{bg}} \tag{13}$$

If the object is not present, all detections are modeled by the uniform distribution:

$$p(\mathbf{S}|\mathbf{h}_0, \theta_{bg}) = \left(\frac{1}{r}\right)^{N} \tag{14}$$

Thus the ratio of likelihood becomes:

$$\frac{p(\mathbf{S}|\mathbf{h}, \theta_{fg})}{p(\mathbf{S}|\mathbf{h}_0, \theta_{bg})} = \mathrm{G}(\log \mathbf{S}^*(\mathbf{h})|\mathbf{t}_l, U_l)\, r^{n_{fg}-1} \tag{15}$$

2.4 Occlusion and Statistics of the feature finder

$$p(\mathbf{h}|\theta_{fg}) = p_{Poiss}(n_{bg}|M)\, \frac{1}{{}^{n}C_r(N, n_{fg})}\, p(\mathbf{d}|\mathbf{D}) \tag{16}$$

The first term models the number of features in the background, using a Poisson distribution, which has a mean M. The second is a book-keeping term for the hypothesis variable and the last is a joint distribution on the occlusions of model parts. It is a multinomial density (of size 2^P), modeling all possible occlusion patterns $\mathbf{d}$, having a parameter, $\mathbf{D}$. This joint distribution allows the modeling of correlations in occlusion: nearby parts are more often occluded together than far apart things. In the null case, we only have only possible hypothesis, $\mathbf{h}_0$, so the only term from (16) that remains is the Poisson which now has to account for all features belonging to the background:

$$p(\mathbf{h}_0|\theta_{bg}) = p_{Poiss}(N|M) \tag{17}$$

Thus the ratio becomes:

$$\frac{p(\mathbf{h}|\theta_{fg})}{p(\mathbf{h}|\theta_{bg})} = \frac{p_{Poiss}(n_{bg}|M)}{p_{Poiss}(N|M)}\, \frac{1}{{}^{n}C_r(N, n_{fg})}\, p(\mathbf{d}|\mathbf{D}) \tag{18}$$

These terms were introduced by Weber *et al.* [28].

2.5 Model structure summary

The model encompasses many of the properties of an object, all in a probabilistic way, so this model can represent both geometrically constrained objects (where

the configuration density would have a small covariance) and objects with distinctive appearance but lacking geometric form (the appearance densities would be tight, but the configuration density would now be looser). Some additional assumptions inherent in our chosen model structure include: given a set of detected features, their appearance and location are independent; the foreground features' appearances are independent of one another; the background features' are independent of the foreground and each other. Using (8),(11),(15) and (18) we can write the likelihood ratio from (1) as:

$$\frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{fg})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})} = \sum_{\mathbf{h}\in H} \frac{p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\theta_{fg})}{p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}_0|\theta_{bg})} \tag{19}$$

$$= \sum_{\mathbf{h}\in H} \prod_{p=1}^{P} \left(\frac{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_p, V_p)}{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_{bg}, V_{bg})} \right)^{d_p} \tag{20}$$

$$\frac{\mathrm{G}(\mathbf{X}^{**}(\mathbf{h})|\boldsymbol{\mu}_l, \Sigma_l)\ \ \mathrm{G}(\log \mathbf{S}^{*}(\mathbf{h})|\mathbf{t}_l, U_l)\,(\alpha r)^{n_{fg}-1}\ \ p_{Poiss}(n_{bg}|M)p(\mathbf{d}|\mathbf{D})}{p_{Poiss}(N|M)\ \ {}^{n}C_r(N, n_{fg})}$$

The intuition is that the majority of the hypotheses will be low scoring as they will be picking up features from background clutter on the image but hopefully a few features will genuinely be part of the object and hypotheses using these will score highly. However, we must be able to locate features over many different instances of the object and over a range of scales in order for this approach to work.

2.6 Feature detection

The model may be learnt from any feature detector that returns a position and a scale. In the sequel we employ two complementary detectors: a region detector and a curve detector. One emphasizes the appearance, the other the geometry.

Region detector: Features are found using the detector of Kadir and Brady [12][3]. This method finds regions that are salient over both location and scale. For each point in the image a histogram $P(I)$ is made of the intensities in a circular region of radius (scale) s. The entropy $H(s)$ of this histogram is then calculated and the local maxima of $H(s)$ are candidate scales for the region. The saliency of each of these candidates is measured by $H\frac{dP}{ds}$ (with appropriate normalization for scale [12,15]).

This gives a 3-D saliency map (over x,y and s). Regions of high saliency are clustered over both location and scale, with a bias toward clusters of large scale, since they tend to be more stable between object instances. The centroids of the clusters then provide the features for learning and recognition, their coordinates within the saliency map defining the centre and radius of each feature.

[3] An implementation of this feature detector is available at http://www.robots.ox.ac.uk/~timork/salscale.html

A good example illustrating the saliency principal is that of a bright circle on a dark background. If the scale is too small then only the white circle is seen, and there is no extrema in entropy. There is an entropy extrema when the scale is slightly larger than the radius of the bright circle, and thereafter the entropy decreases as the scale increases.

In practice this method gives stable identification of features over a variety of sizes and copes well with intra-class variability. The saliency measure is designed to be invariant to scaling, although experimental tests show that this is not entirely the case due to aliasing and other effects. Note, only monochrome information is used to detect and represent features. Figure 3 illustrates this detector on six typical images from the motorbike dataset.

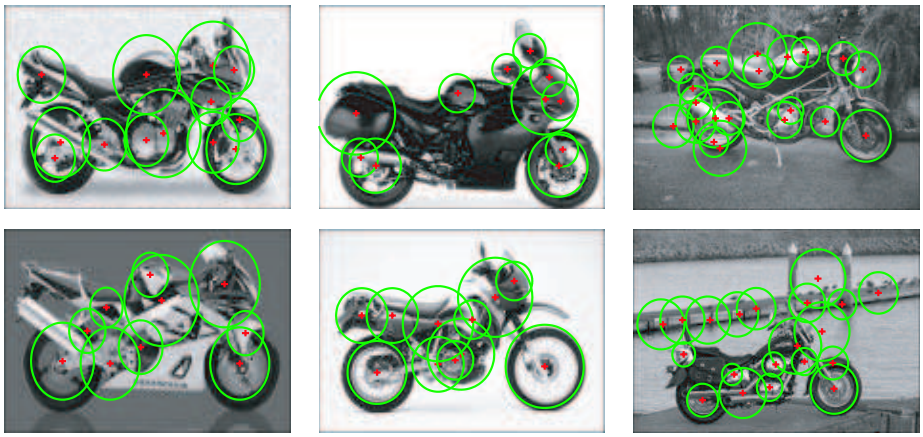


Fig. 3. Six typical motorbikes images with the output of the Kadir-Brady operator overlaid. The +'s illustrate the centre of the salient region, while the circles show the scale of the region. Notice how the operator fires more frequently on more salient regions, ignoring the uniform background present in of some of the images.

Curve detector: Rather than only consider very local spatial arrangements of edge points (as in [2]), extended edge chains are used, detected by the Canny edge operator [6]. The chains are then divided into segments between bitangent points, i.e. points at which a line has two points of tangency with the curve. Figure 4(b) shows an example.

This decomposition is used for two reasons: first, bitangency is covariant with projective transformations. This means that for near planar curves the segmentation is invariant to viewpoint, an important requirement if the same, or similar, objects are imaged at different scales and orientations. Second, by

segmenting curves using a bi-local property interesting segments can be found consistently despite imperfect edgel data.

Bitangent points are found on each chain using the method described in [19]. Since each pair of bitangent points defines a curve which is a sub-section of the chain, there may be multiple decompositions of the chain into curved sections as shown in figure 4(b). In practice, many curve segments are straight lines (within a threshold for noise) and these are discarded as they are intrinsically less informative than curves. In addition, the entire chain is also used, so retaining convex curve portions.

2.7 Feature representation

Region representation: The Kadir & Brady feature detector identifies regions of interest in each image. The coordinates of the centre give us $\mathbf{X}$ and the size of the region gives $\mathbf{S}$.

Once the regions are identified, they are cropped from the image and rescaled to the size of a small (typically 11×11 pixels) patch. Thus, each patch exists in a 121 dimensional space. Since the appearance densities of the model must also exist in this space, we must somehow reduce the dimensionality of each patch whilst retaining its distinctiveness – a 121-dimensional Gaussian will cause numerical problems and also the number of parameters involved (242 per model part) are too many to be estimated. This is done by using principal component analysis (PCA). In the learning stage, we collect the patches from all images and perform PCA on them. Each patch's appearance is then a vector of the coordinates within the first k (typically 10-15) principal components, so giving us the vector of appearance parameters $\mathbf{A}$. This gives a good reconstruction of the original patch whilst using a moderate number of parameters per part (20-30). Thus the appearance densities of all part exist within the same PCA basis (although this basis will vary between object classes – we return to this point in section 5). Since the principal components are orthogonal, it means that the covariance terms between components will be zero, thus V_p (the covariance of a parts appearance) is diagonal in nature. Alternative representations such as ICA and Fisher's linear discriminant were also tried, but in experiments they were shown to be inferior.

We have now computed $\mathbf{X}$, $\mathbf{S}$, and $\mathbf{A}$ for use in learning or recognition. We also retain the PCA basis so that patches from query images may be projected into the same PCA space. For a typical image, this takes 10-15 seconds (all timings given are for a 2 Ghz machine), mainly due to the unoptimized feature detector. Optimization should reduce this to a few seconds.

Curve representation: Each curve is transformed to a canonical position using a similarity transformation such that it starts at the origin and ends at the point $(1, 0)$. If the curve's centroid is below the x-axis then it is flipped both in the x-axis and the line $y = 0.5$, so that the same curve is obtained independent of the edgel ordering. The y value of the curve in this canonical position is sampled

at 13 equally spaced x intervals between $(0,0)$ and $(1,0)$. Figure 4(c) shows curve segments within this canonical space. Since the model is not orientation-invariant, the original orientation of the curve is concatenated to the 13-vector for each curve, giving a 15-vector (for robustness, orientation is represented as a normalized 2-vector). Combining the 15-vectors from all curves within the image gives $\mathbf{A}$.

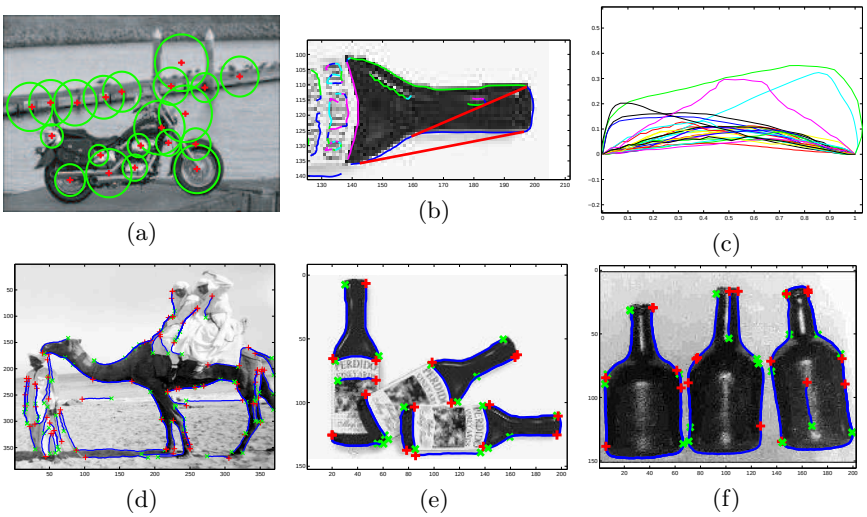


Fig. 4. **(a)** Sample output from the region detector. The circles indicate the scale of the region. **(b)** A long curve segment being decomposed at its bitangent points. **(c)** Curves within the similarity-invariant space - note the clustering. **(d)**, **(e)** & **(f)** show the curve segments identified in three images. The green and red markers indicate the start and end of the curve respectively

3 Learning

In an unsupervised learning scenario, one is presented with a collection of images containing examples of the objects amongst clutter. However the position and scale of the object with each image is unknown; no correspondence between exemplars is given; parts of the object may be missing or occluded. The challenge is to make sense from this profusion of data. Weber *et al.* [28,26] approached the problem of unsupervised learning of object classes in clutter as a maximum likelihood estimation. For this purpose they derived an EM algorithm for the constellation model. We follow their approach in deriving an EM algorithm to estimate the parameters of our improved model.

The task of learning is to estimate the parameters $\theta_{fg} = \{\boldsymbol{\mu}, \Sigma, \mathbf{c}, V, M, \mathbf{D}, \mathbf{t}, U\}$ of the model discussed above. The goal is to find the parameters $\hat{\theta}_{ML}$ which

best explain the data $\mathbf{X}, \mathbf{S}, \mathbf{A}$ from all the training images, that is maximize the likelihood: $\hat{\theta}_{ML} = \underset{\theta}{arg\,max}\ p(\mathbf{X}, \mathbf{S}, \mathbf{A}|\, \theta_{fg})$. Note that the parameters of the background, θ_{bg}, are constant during learning.

Learning is carried out using the expectation-maximization (EM) algorithm [8] which iteratively converges, from some random initial value of θ_{fg} to a maximum (which might be a local one).

We now look at each stage in the learning procedure, giving practical details of its implementation and performance, using the motorbike dataset as an example. We assume that $\mathbf{X}, \mathbf{S}, \mathbf{A}$ have already been extracted from the images, examples of which are shown in figure 3.

3.1 Initialization

Initially we have no knowledge about the structure of the object to be learnt so we are forced to initialize the model parameters randomly. However, the model which has a large number of parameters, must be initialized sensibly to ensure that the parameters will converge to a reasonable maximum. For the configuration, the means are set randomly over the area of the image and the covariances to be large enough so that all hypotheses have a roughly equal weighting. The other terms are set in a similar manner. The same initialization settings are used in all experiments.

3.2 EM update equations

The algorithm has two stages: (i) the E-step in which, given the current value of θ_{fg} at iteration k, θ_{fg}^k, some sufficient statistics are computed and (ii) the M-step where we compute the parameters for the next iteration, θ_{fg}^{k+1} using these sufficient statistics.

We now give the equations for both the E-step and M-step. The E-step requires us to compute the posterior density of the hidden variables, which in our case are the hypotheses. This is calculated using the joint:

$$p(\mathbf{h}|\mathbf{X}, \mathbf{S}, \mathbf{A}, \theta_{fg}^k) = \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}|\, \theta_{fg}^k)}{\sum_{h \in H} p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}|\, \theta_{fg}^k)} = \frac{\frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}|\, \theta_{fg}^k)}{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0|\, \theta_{bg})}}{\sum_{h \in H} \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}|\, \theta + fg^k)}{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0|\, \theta_{bg})}} \quad (21)$$

We divide through by $p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0|\, \theta_{bg})$ as it is easier to compute the joint ratio rather than the joint directly. We then calculate the following sufficient statistics for each image, i from which we have previously extracted $\mathbf{X}^i, \mathbf{A}^i, \mathbf{S}^i$: $E[\mathbf{X}^{**i}]$, $E[\mathbf{X}^{**i}\mathbf{X}^{**i^T}]$, $E[\mathbf{A}_p^i]$, $E[\mathbf{A}_p^i \mathbf{A}_p^{i\,T}]$, $E[\mathbf{S}^{*i}]$, $E[\mathbf{S}^{*i}\mathbf{S}^{*i^T}]$, $E[n^i]$, $E[\mathbf{D}^i]$ where the expectation is taken with respect to the posterior, $p(\mathbf{h}|\mathbf{X}, \mathbf{S}, \mathbf{A}, \theta_{fg}^k)$, for example:

$$E[\mathbf{X}^{**i}] = \sum_{h \in H} p(\mathbf{h}|\mathbf{X}^i, \mathbf{S}^i, \mathbf{A}^i, \theta_{fg}^k)\, \mathbf{X}^{**i}(h) \quad (22)$$

Note that for simplicity we have not considered the case of missing data. The extensions to the above rules for dealing with this may be found in [26]. The general principle is to condition on the features that are present to work out the expected values of those that are missing. In the M-step we then compute: $\theta_{fg}^{k+1} = \{\boldsymbol{\mu}^{k+1}, \Sigma^{k+1}, \mathbf{c}^{k+1}, V^{k+1}, \mathbf{t}^{k+1}, U^{k+1}, M^{k+1}, \mathbf{D}^{k+1}\}$:

$$\boldsymbol{\mu}^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{X}^{**i}] \qquad \Sigma^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{X}^{**i}\mathbf{X}^{**i^T}] - \boldsymbol{\mu}^{k+1}\boldsymbol{\mu}^{k+1^T}$$

$$\mathbf{c}_p^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{A}_p^i] \quad \forall p \in P \qquad V_p^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{A}_p^i \mathbf{A}_p^{i\,T}] - \mathbf{c}_p^{k+1}\mathbf{c}_p^{k+1^T} \quad \forall p \in P$$

$$\mathbf{t}^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{S}^{*i}] \qquad U^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{S}^{*i}\mathbf{S}^{*i^T}] - \mathbf{t}^{k+1}\mathbf{t}^{k+1^T}$$

$$M^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[n^i] \qquad \mathbf{D}^{k+1} = \frac{1}{I}\sum_{i=1}^{I} E[\mathbf{D}^i]$$

where I is total number of training images. The two stages are then repeated until θ_{fg}^k converges to a stable point.

3.3 Efficient search methods

Fully connected model: In computing the sufficient statistics we need to evaluate the likelihood for every hypothesis. Since there are $O(N^P)$ per image, this is the major computational bottleneck in our approach. However, only very small portion of the hypotheses have a high probability, so efficient search methods which can quickly compute this small subset enable the learning procedure to run in a reasonable time. These methods must ensure that initially a sufficiently large set of hypotheses are considered to avoid converging on a local maximum. However, the set must be sufficiently small to ensure that learning is fast. The most difficult stage in learning is the start, when the large covariances that the model is initialized with mean that many hypotheses carry significant weight. Once the model has converged somewhat, only a very small portion of hypotheses will need to be considered.

A tree structure is used to search the space of all possible hypotheses. The leaves of the tree are complete hypotheses with each level representing a part: moving down the tree features are allocated to parts until a complete hypothesis is formed. The A^* algorithm is used to efficiently explore the tree, with a binary heap storing the list of open branches on the tree. Conditional densities for each part (i.e. conditioning on previously allocated parts) are pre-computed to minimize the computation necessary at each branch. The algorithm produces hypotheses ordered by likelihood. For each frame, we compute all hypothesis until they become less than some threshold (e^{-15}) smaller than the best hypothesis. This threshold was chosen to ensure that the learning progressed within a reasonable time while evaluating as many hypotheses as possible.

Additionally, space search methods are used to prune the tree. At a given level of the tree, the joint density of the configuration term allows the density of location of the current part to be computed by conditioning on the previously allocated parts. Only a subset of the N detections need be evaluated by this density: we assume that we can neglect detections if their probability is worse than having all remaining parts be missing. Since the occlusion probabilities are constant for a given learning iteration, this gives a threshold which truncates the density. If the covariance of the density is small, only the best few detections need to be evaluated, enabling significant numbers of hypotheses to be ignored.

Despite these methods, learning a $P = 6$-7 part fully connected model with $N = 20$-30 features per image (a practical maximum), using 400 training images, takes around 24-36 hours to run. This equates to spending 3-4 seconds per image, on average, at each iteration (given total running time of 36 hours, with 400 training images and 100 EM iterations) . It should be noted that learning only needs to be performed once per category, due to the good convergence properties.

Star model: The advantage of this model over the fully connected model is in the computational complexity of both learning and recognition. The reduced dependencies mean that the computation of the marginal density $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$ is now $O(N^2P)$. As with the fully connected model, the majority of the probability mass is concentrated at a few hypotheses hence we only consider hypotheses e^{-15} smaller than the best hypothesis. This efficient search method allows us to handle much large N and P – e.g. with 400 training images and 100 EM iterations, a $P = 6$-7 part model with $N = 20$-30 features can be learnt in 10 minutes, and a model with $P = 12$ and $N = 100$ can be learnt in 12 hours.

3.4 Implementation details

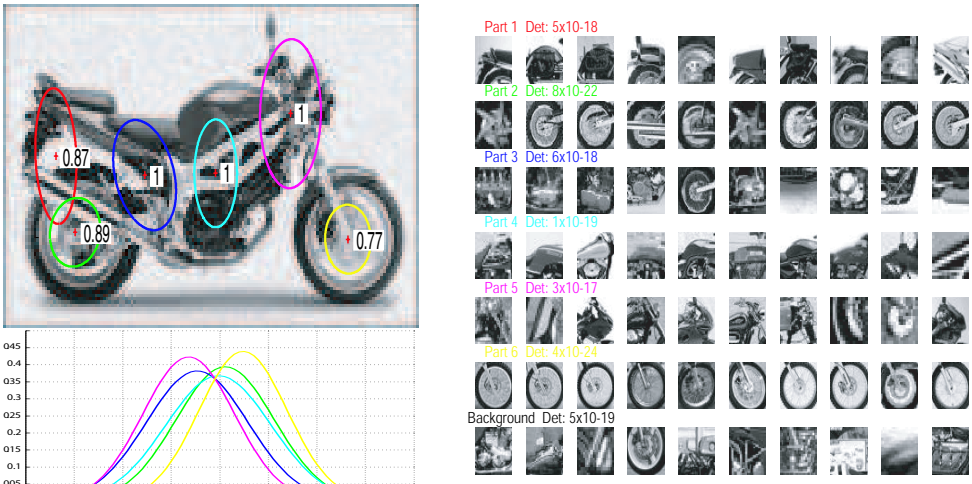
To aid both convergence and speed, an ordering constraint is placed on the x-coordinates of features allocated to parts: the features selected must have a monotonically-increasing x-coordinate. This reduces the total number of hypotheses by $P!$ but unfortunately imposes an artificial constraint upon the configuration of the object. If the object happens to be orientated vertically then this constraint can exclude the best hypothesis. Clearly in this scenario, imposing a constraint on the y-coordinate ordering would resolve the problem but it is not clear how to choose such an appropriate constraint in an unsupervised manner.

Since the model is generative, the background images are not used in learning except for one instance: the appearance model has a distribution in appearance space modeling background features. Estimating this from foreground data proved inaccurate so the parameters were estimated from a set of background images and not updated within the EM iteration.

Extracting the features for a typical image takes 10-15 seconds (all timings given are for a 2 Ghz machine) with our Matlab-C implementation.

3.5 Final model

In figure 5 we show a complete fully connected model trained on the motorbike dataset using regions only. It is pleasing to see that a sensible spatial structure has been picked out and that the appearance samples correspond to distinctive parts of the motorbike.



(a) (b)

Fig. 5. A fully connected model using only regions: (a) Top: configuration density superimposed on an image of a motorbike. The ellipses represent the covariance of each part (the inter-part covariance terms cannot easily be shown) and the probability of each part being present is shown just to the right of the mean. Bottom: Relative scale densities. The x–coordinate is log–scale, relative to the scale of the landmark part. (b) Samples belonging to each part (i.e. from the best hypothesis in a training image) which are closest to the mean of the appearance density.

4 Recognition

The recognition process is very similar in nature to learning. For query image, t, recognition proceeds by first detecting features, giving $\mathbf{X}^t$ and $\mathbf{S}^t$. These are then cropped from the image and rescaled leaving each feature as an 11×11 patch. Using the PCA basis from the training process, they are transformed into the k dimensional PCA space, giving $\mathbf{A}^t$.

Once $\mathbf{X}^t$, $\mathbf{A}^t$ and $\mathbf{S}^t$ have been obtained we then compute the likelihood ratio using (1). To determine if an object is in the image involves the summation

over all hypotheses, not the best one. The likelihood ratio, assuming we take the ratio of the priors in (1) to be 1, is the same as the ratio of posteriors, R. This is then compared to a threshold to make the object present/absent decision. This threshold is determined from the training set to give the desired balance between false positives and false negatives.

If we wish to localize the object within the image, the best hypothesis is found and a bounding box around it formed at its location. We then sum over all hypotheses which are within this box. If the total is greater than the threshold then an instance of the object placed at the centroid of the box and all features within the box are deleted. The next best hypothesis is then found and the process repeated until the sum of hypotheses within the box fall below the threshold. This procedure allows the localization of multiple object instances within the image.

The same efficient search methods described in section 3.3 are used in the recognition process to perform the summation over only those hypotheses that have a significant contribution to the overall sum. However, recognition is faster as the covariances are tight (as compared with the initial values in the learning process) so the vast majority of hypotheses may safely be ignored. For $N = 25$ and $P = 6$ the process takes around 2-3 seconds per image for the fully connected model, and 0.02 seconds for the star model.

5 Results

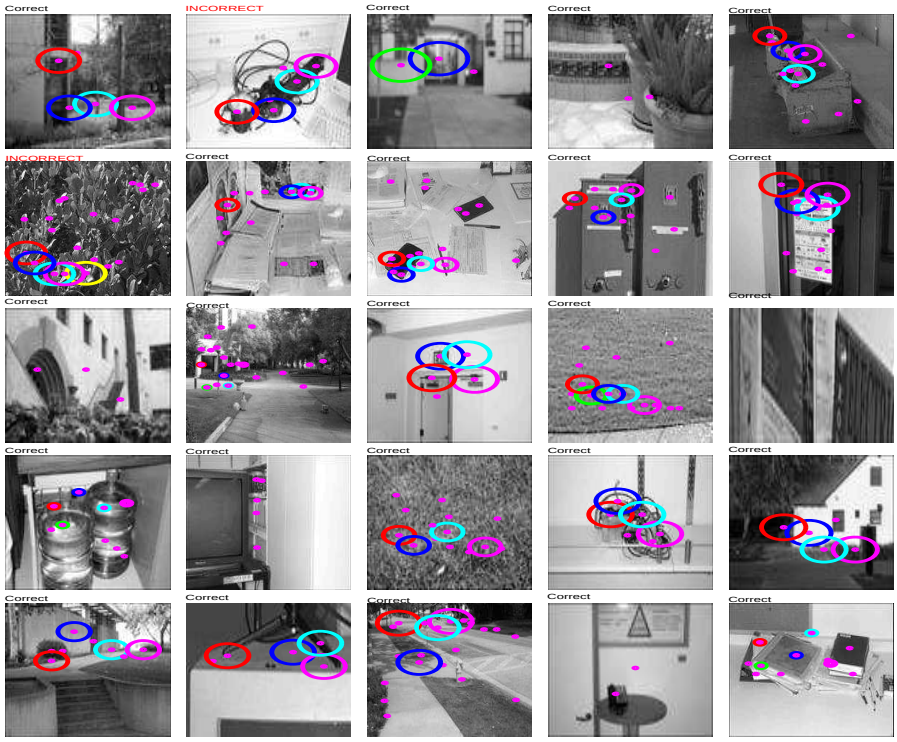
A variety of experiments were carried out on a number of different pre-prepared datasets, each one being a different category. The first series of experiments were performed on large datasets which were split randomly into two separate sets of equal size. The model was then trained on the first and tested on the second. In recognition, the decision was a simple object present/absent one, except for the cars (side) dataset where multiple instances of object and their location within the image were to be found. To avoid confusion, we call these the *closed world* experiments.

In the second series of experiments, models were trained on a limited number of hand-selected images obtained from Google's image search. They were then tested on data from a different source (i.e. movie footage) - quite different in nature to the training images. These are called the *open world* experiments.

5.1 Methodology and datasets for closed world experiments

The performance was evaluated by plotting receiver-operating characteristic (ROC) curves. To ease comparisons we use a single point on the curve to summarises its performance, namely the point of equal error (i.e. p(True positive)=1-p(False positive)) when testing against one of two background (negative) datasets. For example a figure of 91% means that 91% of the foreground images were correctly classified but 9% of the background images were incorrectly classified (i.e.

(a)



(b)

Fig. 6. **(a)** The motorbike model from Fig. 5 evaluating a set of query images. The pink dots are features found on each image and the coloured circles indicate the features of the best hypothesis in the image. The size of the circles indicates the scale of feature. The outcome of the classification is marked above each image, incorrect classifications being highlighted in red. **(b)** The model evaluating query images consisting of scenes around Caltech – the negative test set.

thought to be foreground). While the number of foreground test images varied between experiments, depending on the number of images available for each class, the foreground and background sets were always of equal size.

There were two phases of experiments. In the first, the fixed-scale experiments, those datasets with scale variability, were normalized so that the objects were of uniform size. The algorithm was then evaluated on the datasets and compared to other approaches. In the second phase the scale-invariant algorithm was run on the datasets containing scale variation and the performance compared to the scale-normalized case.

In all these experiments, the only feature type used were Kadir & Brady interest regions and the fully connected configuration model was used in preference to the star model. The following parameters were adopted: $k = 15$, $P = 6$ and on average $N = 25$. The only parameter that was adjusted at all was the scale over which features were found. The standard setting was $4 - 60$ pixels but for the scale-invariant experiments, this was changed to account for the wider scale variation in features.

Six diverse datasets were used in the experiments, examples of which can be seen in figure 1. These include motorbikes, airplanes, cars (rear), faces and cluttered scenes around Caltech (used as the negative test set). Two additional background (negative) datasets are used. The first was collected from Google's image search using the keyword "things", resulting in a highly diverse collection of images. The second was a set of empty road scenes for use as a realistic background test set for the cars (rear) dataset. The motorbike and airplanes datasets contain images that were manually flipped to ensure the object was facing the same way. The spotted cat dataset, obtained from the Corel base, was only 100 images originally, so another 100 were added by reflecting the original images, making 200 in total. Amongst the datasets, only the motorbikes, airplanes and cars (rear) contained any meaningful scale variation. All images from the datasets were converted to grayscale as colour was not used in our experiments. Table 1 gives the size of training set used for each dataset in the experiments.

Fixed scale experiments: Figures 7-9 show models and test images for four of the datasets. Notice how each model captures the essence, be it in appearance or configuration or both, of the object. The face and motorbike datasets have tight configuration models, but some of the parts have a highly variable appearance. For these parts any feature in that location will do regardless of what it looks like (hence the probability of detection is 1). Conversely, the spotted cat dataset has a loose configuration model, but a highly distinctive appearance for each region. In this instance, the model is just looking for regions of spotty fur, regardless of their location. The differing nature of these examples illustrate the flexible nature of the model.

The majority of errors are a result of the object receiving insufficient coverage from the feature detector. This happens for a number of reasons. One possibility is that, when a threshold is imposed on N (for the sake of speed), many

features on the object are removed. Alternatively, the feature detector seems to perform badly when the object is much darker than the background (see examples in figure 9). Finally, the clustering within the feature detector is somewhat temperamental and can result in parts of the object being missed.

Dataset	Total size of dataset	Object width (pixels)	(a)	(b)
Motorbikes	800	200	91.0	90.5
Faces	435	300	91.7	88.5
Airplanes	800	300	85.5	86.5
Spotted Cats	200	80	85.0	92.0
Cars (Rear)	800	100	88.8	89.3

Table 1. Recognition results on five scale-normalized datasets. (a) is the detection rate (%) at the point of equal-error on an ROC curve, testing against the Caltech background (negative) dataset as the negative set (with the exception of Cars (rear) which uses empty road scenes as the negative test set). The algorithm's parameters were kept *exactly* the same. (b) is the same as (a), except that the Google background dataset was used in testing.

Table 1 shows the performance of the algorithm across the five datasets, with some of the learnt models illustrated in figures 7-9. In experiments (a) and (b), exactly the same algorithm settings are used for all models. Note that the performance is above 90% for all four datasets, regardless of the choice of background dataset.

		Recognized category		
Query image	Motorbike	Spotted cat	Airplane	Face
Motorbike	94.1	4.3	1.2	0.4
Spotted Cats	1.0	97.0	2.0	0
Airplane	8.0	1.1	90.5	0.4
Face	1.4	0.9	7.8	89.9

Table 2. Confusion table between the four classes for fixed scale learning and recognition of the fully connected model using regions. Each row gives a breakdown of how a query image of a given category is classified (in %). No negative (background) dataset was used, rather images belonging to each class acted as negative examples for models trained for the other classes. The optimum would be 100% on the diagonal with zero elsewhere.

Table 2 shows a confusion table between the different classes. For each query image all four models are applied and the one with the highest likelihood determines the category. Despite being inherently generative, the models can distinguish between the classes well. In order to construct this table, it was necessary to be able to directly compare likelihoods between different models and so the

same PCA basis was used for all classes, in contrast to all other experiments, where it was computed for each dataset separately. In changing the way the PCA basis was computed, the object present/absent performance is not significantly altered - in the case of motorbikes, the performance increased from 91.0% (PCA basis computed for motorbike dataset alone) to 91.3% (fixed PCA basis across all classes).

Comparison with other methods: Figure 10 shows a recall-precision curve (RPC) and a table comparing the algorithm to previous approaches to object class recognition [1,26,28]. In the majority of cases, the performance of the algorithm is superior to the earlier methods, despite not being tuned for a particular dataset. In the Cars (Side) dataset from Agarwal *et al.* [1] the performance also includes localizing the object instance(s) within the image.

More recently several methods [7,11,14,18,24] have exceeded the performance of the fully connected constellation model given in the table of Figure 10 for some classes. However, in the case of [7,11,18,24] the methods do not determine the localization of the object instance in the image. The method of Leibe and Schiele [14] does determine the location, but also includes a verification stage. This stage is responsible for a 6.5% improvement in performance over the original hypothesis. The constellation model could also benefit from a stage similar to this.

Scale-invariant experiments: Table 3 shows the performance of scale-invariant models on unscaled images. Comparing these results to those of the scale-variant models tested on the pre-scaled data (as in table 1 above), it can be seen that the performance is roughly the same as the scale-normalised case. In the case of airplanes, the improved performance may be explained by the imprecise normalisation of the data when using the fixed-scale model. Figure 11 shows the scale-invariant model for this cars (read) dataset. This model was tested against a negative set consisting of background road scenes (rather than the background images, examples of which are in Fig. 1) to make a more realistic experiment.

Dataset	Total size of dataset	Object size range (pixels)	Scale-invariant performance
Motorbikes	800	200-480	90.5
Airplanes	800	200-500	90.8
Cars (Rear)	800	100-550	90.3

Table 3. Results for scale-invariant learning/recognition.

5.2 Methodology and datasets for open world experiments

The aim of the open world experiments is to test the algorithm in a more realistic environment, where large collections of training data, similar in nature to the test

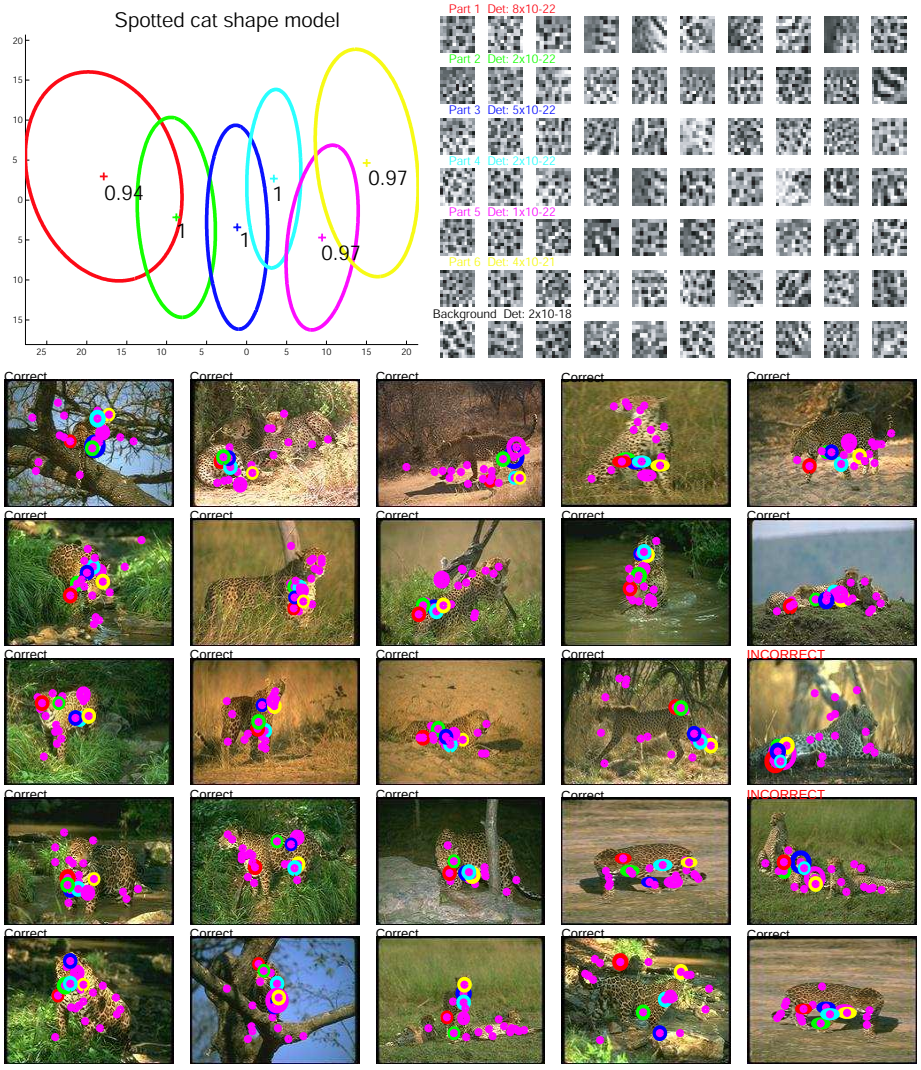


Fig. 7. A typical spotted cat fully connected model with 6 region parts. Note the loose configuration model but distinctive "spotted fur" appearance.

Fig. 8. A typical fully connected airplane model with 6 region parts. The long horizontal structure of the fuselage is captured by the configuration model.

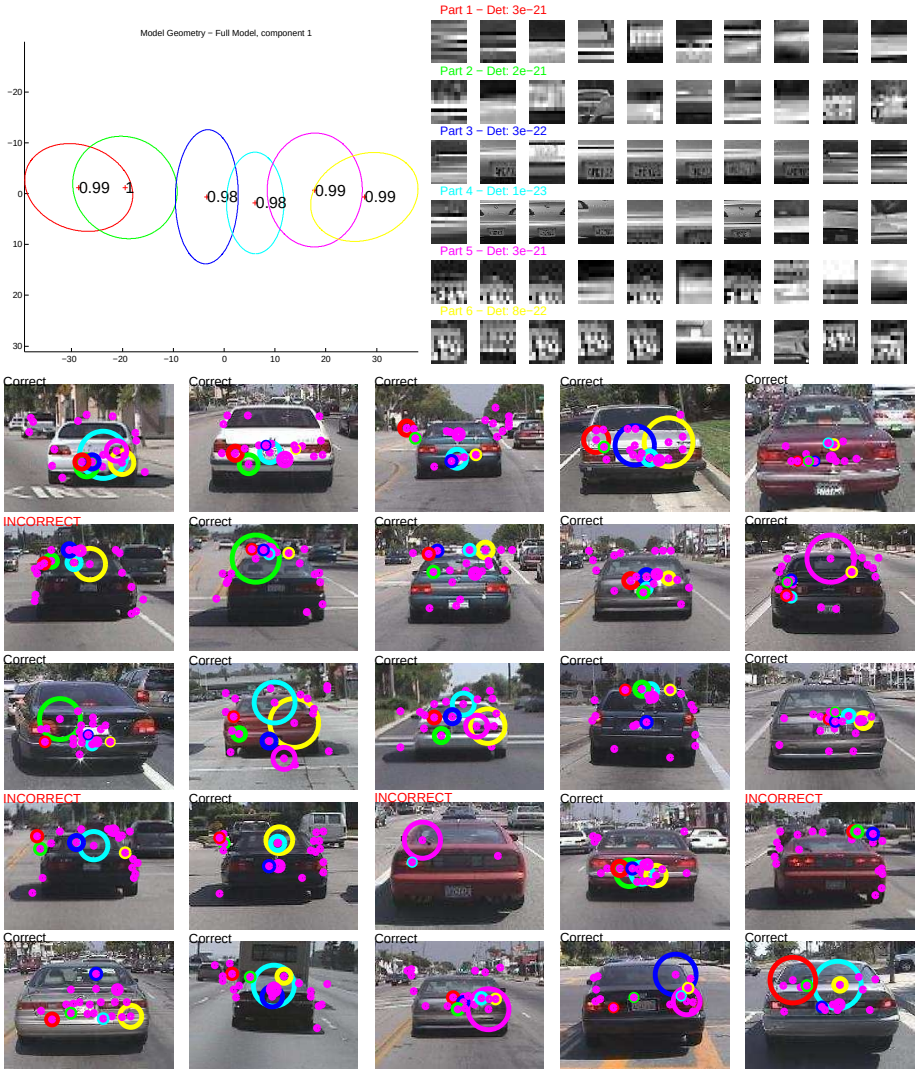
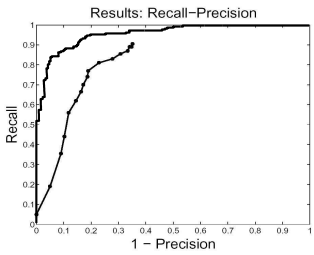


Fig. 9. A fixed scale cars (Rear) fully connected model with 6 region parts. The feature detector fires somewhat erratically, thus the model resorts to finding low-level horizontal line type structures on the back of the car.



Dataset	Ours	Others	Ref.
Motorbikes	91.0	84	[26]
Faces	91.7	94	[28]
Airplanes	85.5	68	[26]
Cars(Side)	88.5	79	[1]

Fig. 10. Comparison to other methods [1,26,28]. The diagram on the left shows the RPC for [1] and our algorithm on the cars (side) dataset. On the right the table gives ROC equal error rates (except for the car (side) dataset where it is a recall-precision equal error) on a number of datasets. The errors for our algorithm are at least half those of the other methods, except for the face dataset.

data are not typically available. The particular application was to find objects within a video sequence, the BBC situation comedy Fawlty Towers. Every 30th frame from a 40 minute episode was extracted, forming a test set of 1463 images. The dataset is challenging because images are of poor quality due to the effects of de-interlacing; motion blur and age of the original footage (the series was made in the 1970's). A couple of example frames from the test set is shown in Figure 12.

Two objects present in a reasonable portion of the Fawlty Towers sequence were chosen as classes to search for: antique barometer (present in 12.9% of frames) and car (front) (present in 1.5%). 15 images for each category were then hand-selected from Google's image search[4] to form a limited training set. No pre-processing of any kind was done on the images.

In this application, since it is not clear what would constitute a negative test set, the evaluation criterion used is a Recall-Precision curve. Recall and Precision are defined as:

$$\text{Recall} = \frac{\text{Portion of positives returned}}{\text{Total positives in dataset}} \qquad \text{Precision} = \frac{\text{Portion of positives returned}}{\text{Total returned}}$$

Positives and negatives in this instances are frames with the object present/absent respectively.

Models are learnt using a combination of both Kadir & Brady interest regions and Curves. Given the small size of the target object within the frames, a large N was needed to ensure good converge of the image. Hence the star model was used for both learning and recognition. A variety of combinations were evaluated for each category. The following parameters were adopted: $k = 15$, $P = 5$ and on average $N = 200$ in recognition.

[4] http://www.google.com/imghp

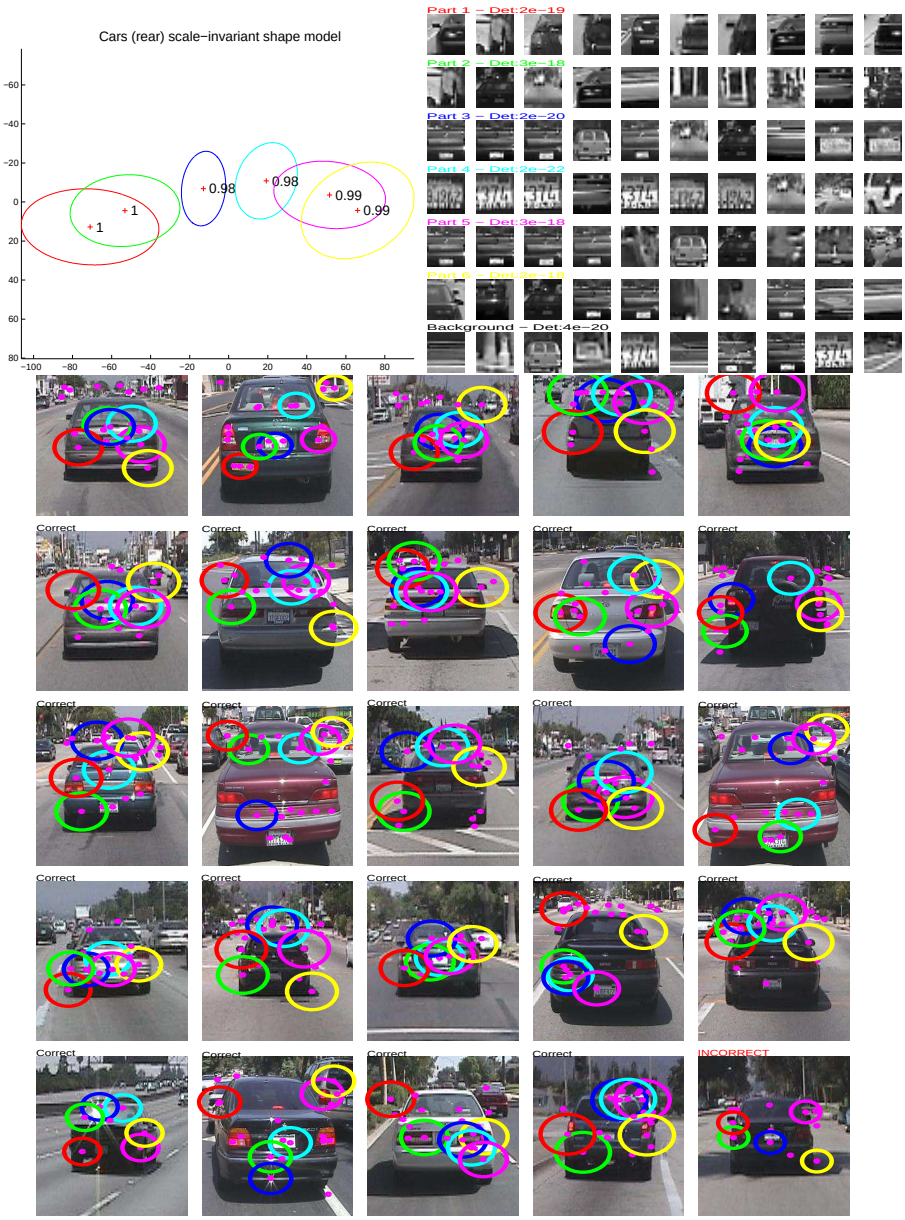


Fig. 11. A scale-invariant cars (rear) fully connected model with 6 region parts.

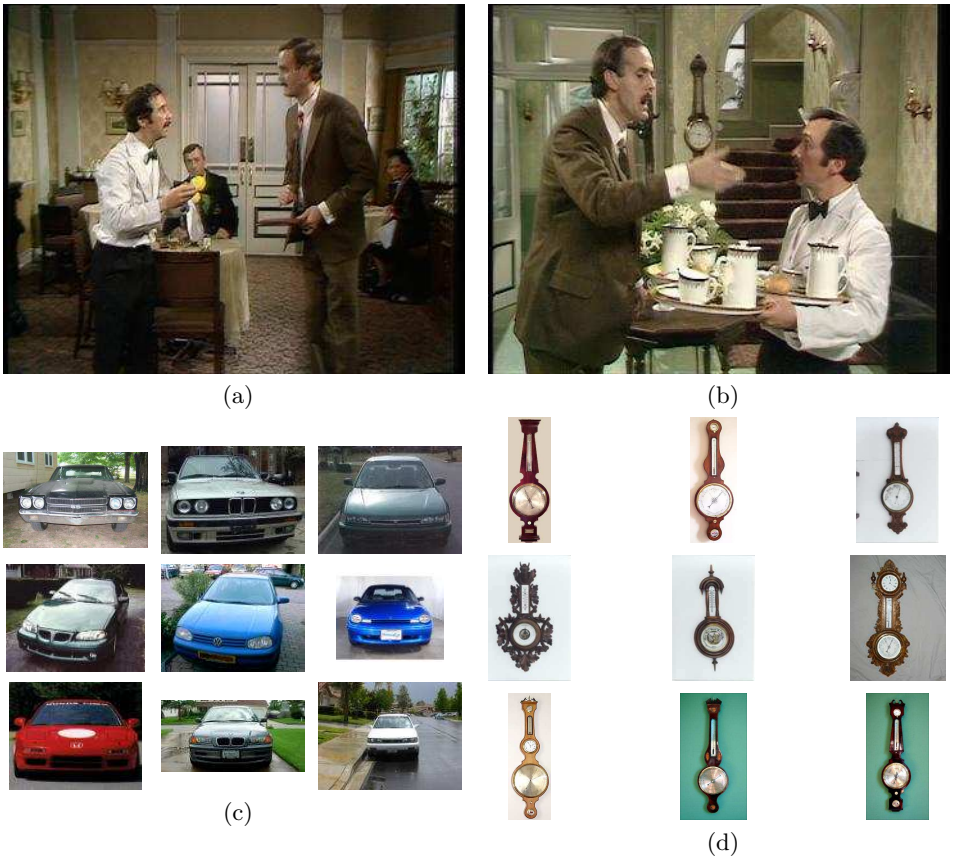


Fig. 12. (a) & **(b)** Two typical scenes from the sit-com "Fawlty Towers". **(c)** Sample training images for Cars (Front). **(d)** Sample training images for Antique Barometers.

Fawlty Towers experiments: Two object classes were learnt from Google images and evaluated on the Fawlty Towers video sequence: antique barometer and car (front). The antique barometer model with the best performance is shown in Figure 13. Note that it uses a combination of curves and regions. As Figure 15(a) shows, using either all regions or all curves gives inferior performance. The first pages of frames returned by the model (ranked by likelihood ratio) is shown in Figure 14.

The model and detection examples for Cars (Front) are shown in figure 16, with the recall-precision curve shown in figure 15(b). The poor quality of the model for this class – it just appears to prefer indistinct horizontal edge features on the car's radiator grille – is attributable to the tiny quantity of training data. Since 200-300 images are typically required to eliminate over-fitting, it is clear that using only 15 images places these models at a severe handicap. Nevertheless, the precision of the model is very good.

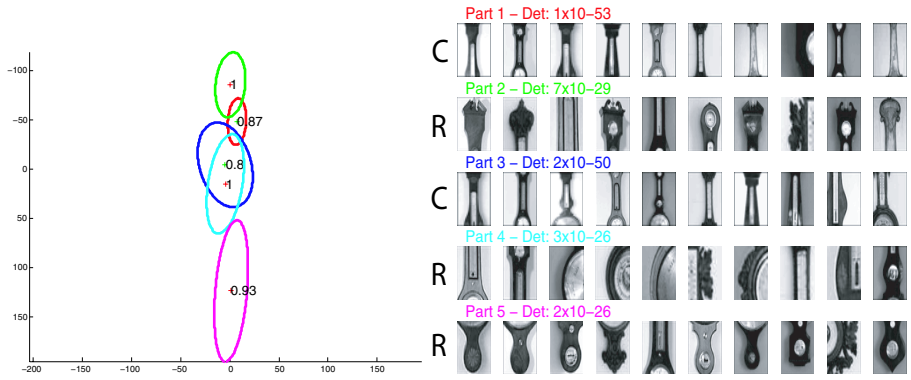


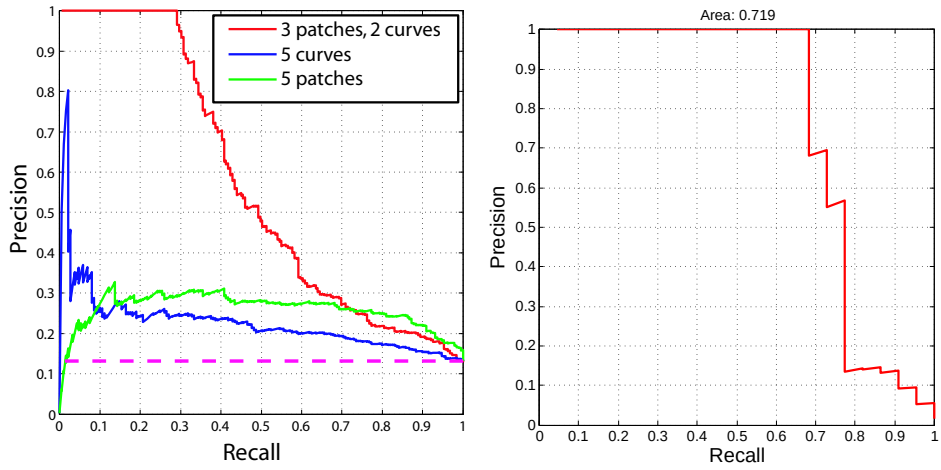
Fig. 13. A barometer star model consisting of 3 regions (R) and 2 curves (C).

6 Conclusions and Further work

We have proposed an improved learning scheme for the 'constellation model' which is scale-invariant and where configuration and appearance are learnt simultaneously. We tested an implementation of such scheme on six diverse and challenging datasets. We find that learning is robust with respect to clutter, scale variation and inter-object variability. It does not require human intervention to segment, normalize or otherwise pre-process the training data.

There are two other areas where improvements will be very beneficial. The first is in a further generalization of the model structure to have a multi-modal appearance density with a single configuration distribution. This will allow more complex appearances to be represented, for example faces with eyes (wide) shut or open. Second, we have built in scale-invariance, but full affine-invariance

Fig. 14. The first page of images returned by the barometer model, ordered by likelihood ratio.



(a) (b)

Fig. 15. (a) Recall-precision curves for barometer models with different combinations of feature types. **(b)** Recall-precision curves for Cars (Front).

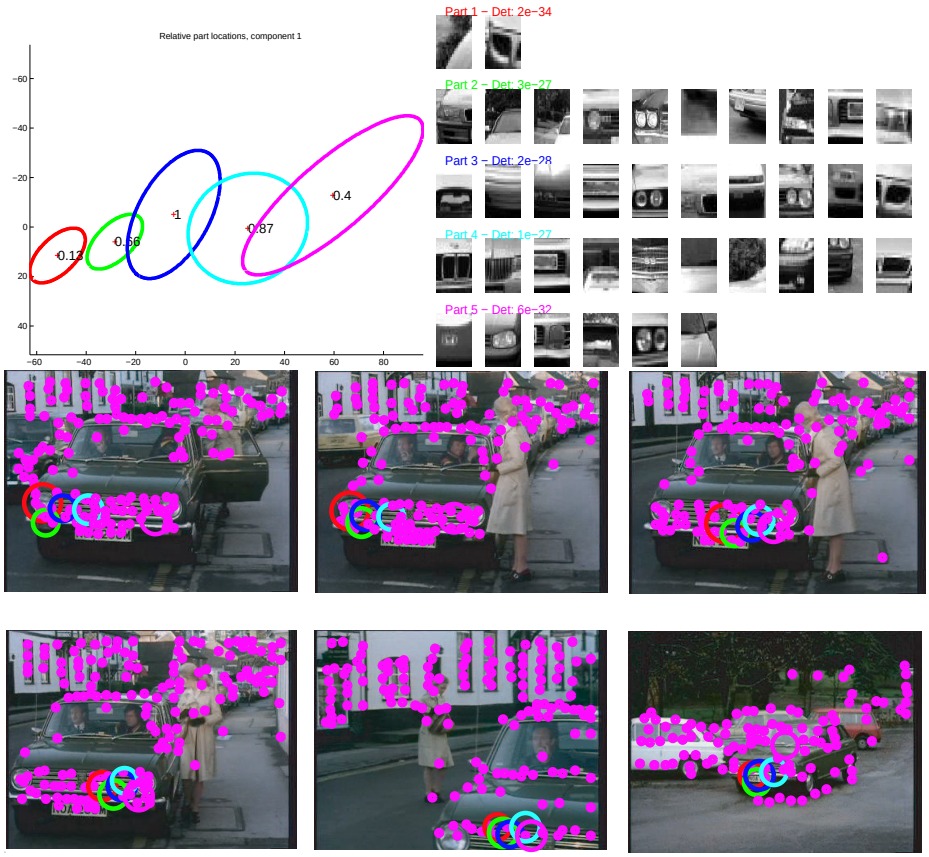


Fig. 16. A Cars (Front) star model consisting of 5 parts using regions

should also be possible. This would enable learning and recognition from images with much larger viewpoint variation.

Acknowledgment

We are grateful for suggestions from Michael Isard, to Timor Kadir for advice on the feature detector, and to D. Roth for providing the Cars (Side) dataset. Funding was provided by National Science Foundation Engineering Research Center for Neuromorphic Systems Engineering, the UK EPSRC, and European Union (FP5-project 'CogViSys', IST-2000-29404).

References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the European Conference on Computer Vision*, pages 113–130, 2002.
2. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
3. S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Neural Information Processing Systems*, pages 831–837, 2000.
4. E. Borenstein. and S. Ullman. Class-specific, top-down segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 109–124, 2002.
5. M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of the European Conference on Computer Vision*, pages 628–641, 1998.
6. J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
7. G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.
8. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JRSS* **B**, 39:1–38, 1976.
9. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2000.
10. W. E. L. Grimson. *Object Recognition by Computer, The Role of Geometric Constraints*. MIT Press, 1990.
11. F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, pages 90–96, 2004.
12. T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
13. Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Howard, and L. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems II (Denver 1989)*, pages 396–404. Morgan Kaufmann, San Mateo, CA, 1990.

14. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.
15. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
16. D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
17. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001.
18. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In T. Pajdla and J. Matas, editors, *ECCV'04*, volume 3022 of *LNCS*, pages 71–84. Springer, 2004.
19. C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(2), 1995.
20. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, Jan 1998.
21. C. Schmid. Constructing models for content-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45, 2001.
22. H. Schneiderman and T. Kanade. A statistical approach to 3D object detection applied to faces and cars. In *Proc. Computer Vision and Pattern Recognition*, pages 746–751, 2000.
23. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, Jan 1998.
24. J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, pages 518–529, 2004.
25. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.
26. M. Weber. *Unsupervised Learning of Models for Object Recognition*. PhD thesis, California Institute of Technology, Pasadena, CA, 2000.
27. M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 101–109, June 2000.
28. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the European Conference on Computer Vision*, pages 18–32, 2000.