A Sparse Object Category Model for Efficient Learning and Complete Recognition

Rob Fergus[1], Pietro Perona[2], and Andrew Zisserman[1]

[1] Dept. of Engineering Science
University of Oxford
Parks Road, Oxford
OX1 3PJ, U.K.
{fergus,az}@robots.ox.ac.uk

[2] Dept. of Electrical Engineering
California Institute of Technology
MC 136–93, Pasadena
CA 91125, U.S.A.
perona@vision.caltech.edu

Abstract

We present a "parts and structure" model for object category recognition that can be learnt efficiently and in a weakly-supervised manner: the model is learnt from example images containing category instances, without requiring segmentation from background clutter.

The model is a sparse representation of the object, and consists of a star topology configuration of parts modeling the output of a variety of feature detectors. The optimal choice of feature types (whose repertoire includes interest points, curves and regions) is made automatically.

In recognition, the model may be applied efficiently in a complete manner, bypassing the need for feature detectors, to give the globally optimal match within a query image. The approach is demonstrated on a wide variety of categories, and delivers both successful classification and localization of the object within the image.

1 Introduction

A variety of models and methods exist for representing, learning and recognizing object categories in images. Many of these are variations on the "Parts and Structure" model introduced by Fischler and Elschlager [10], though the modern instantiations use scale-invariant image fragments [1–3, 12, 15, 20, 21]. The constellation model [3, 8, 21] was the first to convincingly demonstrate that models could be learnt from weakly-supervised unsegmented training images (i.e. the only supervision information was that the image contained an instance of the object category, but not the location of the instance in the image). Various types of categories could be modeled, including those specified by tight spatial configurations (such as cars) and those specified by tight appearance exemplars (such as spotted cats). The model was translation and scale invariant both in learning and in recognition.

However, the Constellation model of [8] has some serious short-comings, namely: (i) The joint nature of the shape model results in an exponential explosion in computational cost, limiting the number of parts and regions per image that can be handled. For N feature detections, and P model parts the complexity for both learning and recognition is $O(N^P)$; (ii) Since only 20-30 regions per image and 6 parts are permitted by this complexity, the model can only learn from an incredibly sparse representation of the image. Good performance is therefore highly dependent on the consistent firing of the feature detector; (iii) Only one type of feature detector (a region operator) was used, making the model very sensitive to the nature of the class. If the distinctive features of the category happen, say, to be edge-based then relying on a region-based detector is likely to give poor results (though this limitation was overcome in later work [9]); (iv) The model has many parameters resulting in over-fitting unless a large number of training images (typically 200+) are used.

Other models and methods have since been developed which have achieved superior performance to the constellation model on at least a subset of the object categories modeled in [8]. These models range from bag-of-word models (where the words are vector quantized invariant descriptors) with no spatial organization [5, 18], through to fragment based models [2, 15] with particular spatial configurations. The methods utilize a range of machine learning approaches EM, SVMs and Adaboost.

In this paper we propose a heterogeneous star model (HSM) which maintains the simple training requirements of the constellation model, and also, like the constellation model, gives a localization for the recognized object. The model is translation and scale invariant both in learning and in recognition. There are three main areas of innovation: (i) both in learning and recognition it has a lower complexity than the constellation model. This enables both the number of parts and the number of detected features to be increased substantially; (ii) it is heterogeneous and is able to make the optimum selection of feature types (here from a pool of three, including curves). This enables it to better model objects with significant intra-class variation in appearance, but less variation in outline (for example a guitar), or vice-versa; (iii) The recognition stage can use feature detectors or can be complete in the manner of Felzenswalb and Huttenlocher [6]. In the latter case there is no actual detection stage. Rather the model itself defines the areas of most relevance using a matched filter. This complete search overcomes many false negatives due to feature drop out, and also poor localizations due to small feature displacement and scale errors.

2 Approach

We describe here the structure of the heterogeneous star model, how it is learnt from training data, and how it is applied to test data for recognition.

2.1 Star model

As in the constellation model of [8], our model has P parts and parameters θ. From each image i, we extract N features with locations $\mathbf{X}^i$; scales $\mathbf{S}^i$ and descriptors $\mathbf{D}^i$. In learning, the aim is to find the value of θ that maximizes the log-likelihood over all images:

$$\sum_i \log\, p(\mathbf{X}^i, \mathbf{D}^i, \mathbf{S}^i|\theta) \tag{1}$$

Since $N >> P$, we introduce an assignment variable, $\mathbf{h}$, to assign features to parts in the model. The log-likelihood is obtained by marginalizing over $\mathbf{h}$.

$$\sum_i \log \sum_{\mathbf{h}} p(\mathbf{X}^i, \mathbf{D}^i, \mathbf{S}^i, \mathbf{h}|\theta) \tag{2}$$

In the constellation model, the joint density is factored as:

$$p(\mathbf{X}^i, \mathbf{D}^i, \mathbf{S}^i, \mathbf{h}|\theta) = \underbrace{p(\mathbf{D}^i|\mathbf{h}, \theta)}_{Appearance}\, \underbrace{p(\mathbf{X}^i|\mathbf{S}^i, \mathbf{h}, \theta)}_{Rel.\ Locations}\, \underbrace{p(\mathbf{S}^i|\mathbf{h}, \theta)}_{Rel.\ Scale}\, \underbrace{p(\mathbf{h}|\theta)}_{Occlusion} \tag{3}$$

In [8], the appearance model for each part is assumed independent but the relative location of the model parts is represented by a joint Gaussian density. While this provides the most thorough description, it makes the location of all parts dependent on one another. Consequently, the EM-based learning scheme, which entails marginalizing over $p(\mathbf{h}|\mathbf{X}^i, \mathbf{D}^i, \mathbf{S}^i, \theta)$, becomes an $O(N^P)$ operation. We propose here a simplified configuration model in which the location of the model part is conditioned on the location of a *landmark* part. 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. A similar model, where the reference frame acts as a landmark is used by Lowe [16] and was studied in a probabilistic framework by Moreels *et al.* [17]. Figure 1 illustrates the differences

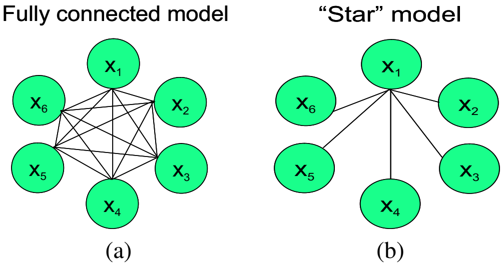


Fig. 1. (a) Fully-connected six part shape model. Each node is a model part while the edges represent the dependencies between parts. **(b)** A six part Star model. The former has complexity $O(N^P)$ while the latter has complexity $O(N^2P)$ which may be further improved in recognition by the use of distance-transforms [6] to $O(NP)$.

between the full and star models. In the star model the joint probability of the configuration aspect of the model may be factored as:

$$p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta) = p(\mathbf{x}_L|h_L) \prod_{j \neq L} p(\mathbf{x}_j|\mathbf{x}_L, s_L, h_j, \theta_j) \tag{4}$$

where $\mathbf{x}_j$ is the position of part j and L is the landmark part. We adopt a Gaussian model for $p(\mathbf{x}_j|\mathbf{x}_L, s_L, h_j, \theta_j)$ which depends only on the relative position and scale between each part and the landmark. The reduced dependencies of this model mean that the marginalization in Eqn. 2 is $O(N^2P)$, in theory allowing us to cope with a larger N and P in learning and recognition.

In practical terms, we can achieve translation invariance by subtracting the location of the landmark part from the non-landmark ones. Scale invariance is achieved by dividing the location of the non-landmark parts by the locally measured scale of the landmark part.

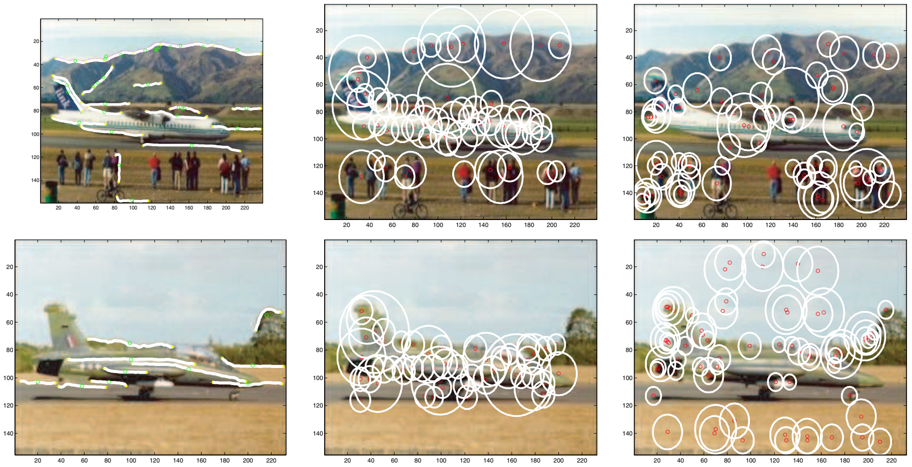
It is useful to examine what has been lost in the star compared to the constellation model of [8]. In the star model any of the leaf (i.e. non-landmark) parts can be occluded, but (as discussed below) we impose the condition that the landmark part must always be present. With small N this can lead to a model with artificially high variance, but as N increases this ceases to be a problem (since the landmark is increasingly likely to actually be detected). In the constellation model any or several parts can be occluded. This is a powerful feature: not only does it make the model robust to the inadequacies of the region detector but it also assists the convergence properties of the model by enabling a subset of the parts to be fitted rather than all simultaneously.

The star model does have other benefits though, in that it has less parameters so that the model can be trained on fewer images without over-fitting occurring.

2.2 Heterogeneous features

By constraining the model to operate in both learning and recognition from the sparse outputs of a feature detector, good performance is highly dependent on the detector finding parts of the object that are characteristic and distinctive of the class. The majority of approaches using feature-based methods rely on region detectors such as Kadir and Brady or multi-scale Harris [11, 13] which favour interest points or circular regions. However, for certain classes such as bottles or mugs, the outline of the object is more informative than the textured regions on the interior. Curves have been used to a limited extent in previous models for object categories, for example both Fergus *et al.* [9] and Jurie & Schmid [12] introduce curves as a feature type. However, in both cases the model was constrained to being homogeneous, i.e. consisting only of curves. Here the models can utilize a combination of different features detectors, the optimal selection being made automatically. This makes the scheme far more tolerant to the type of category to be learnt.

In our scheme, we have a choice of three feature types: Kadir & Brady; multi-scale Harris and Curves. Figure 2 shows examples of these 3 operators on two sample airplane images. The detectors were chosen since they are somewhat complementary in their properties: Kadir & Brady favours circular regions; multi-scale Harris prefers interest points, and curves locate the outline of the object.



(a) (b) (c)

Fig. 2. Output of three different feature detectors on two airplane images. **(a)** Curves. **(b)** Kadir & Brady. **(c)** Multi-scale Harris.

To be able to learn different combinations of features we use the same representation for all types. Inspired by the performance of PCA-SIFT in region matching [14], we utilize a gradient-based PCA approach in contrast to the intensity-based PCA approach of [8]. Both the region operators give a location and scale for each feature. Each feature is cropped from the image (using a square mask); rescaled to a $k \times k$ patch; has its gradient computed and then normalized to remove intensity differences. Note that we do not perform any orientation normalization as in [14]. The outcome is a vector of length $2k^2$, with the first k elements representing the x derivative, and the second k the y derivatives. The derivatives are computed by symmetric finite difference (cropping to avoid edge effects).

The normalized gradient-patch is then projected into a fixed PCA basis[3] of d dimensions. Two additional measurements are made for each gradient-patch: its unnormalized energy and the reconstruction error between the point in the PCA basis and the original gradient-patch. Each region is thus represented by a vector of length $d + 2$.

Curve features are extracted in the same manner as [9]: a Canny edge detector is run over the image; the edgels are grouped into chains; each chain is then broken at its bitangent points to give a curve. Since the chain may have multiple bitangent points, each chain may result in multiple curves (which may overlap in portions). Curves which are very straight tend to be uninformative and are discarded.

The curves are then represented in the same way as the regions. Each curve's location is taken as its centroid with the scale being its length. The region around the curve is then cropped from the image and processed in the manner described above. We use the curve as an feature detector, modeling the textured region around the curve, rather

[3] The fixed basis was computed from patches extracted using all Kadir and Brady regions found on all the training images of Motorbikes; Faces; Airplanes; Cars (Rear); Leopards and Caltech background.

than the curve itself. Modeling the actual shape of the curve, as was done in [9], proved to be uninformative, in part due to the difficulty of extracting the contours consistently enough.

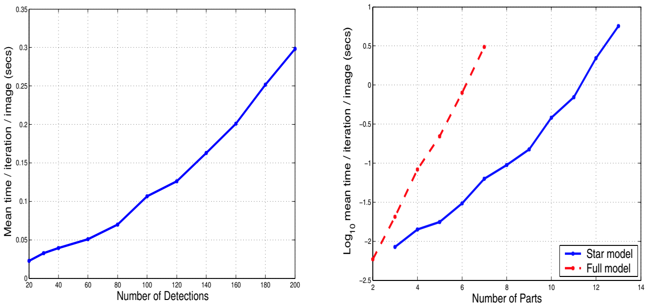
2.3 Learning the model

Learning a heterogeneous star model (HSM) can be approached in several ways. One method is to learn a fully connected constellation model using EM [8] and then reduce the learnt spatial model to a star by completely trying out each of the parts as a landmark, and picking the one which gives the highest likelihood on the training data. The limitation of this approach is that the fully connected model can only handle a small number of parts and detections in learning. The second method, which we adopt, is to learn the HSM directly using EM as in [8, 21], starting from randomly-chosen initial conditions, enabling the learning of many more parts and with more detections/image.

Due to the more flexible nature of the HSM, successful learning depends on a number of factors: To avoid combinatorics inherent in parameter space and to ensure the good convergence properties of the model, an ordering constraint is imposed on the locations of the model parts (e.g. the x-coordinates must be increasing). However, to enable the landmark part to select the most stable feature on the object (recall that we force it to always be present), the landmark is not subject to this constraint. Additionally, each part is only allowed to pick features of a pre-defined type and the ordering constraint only applies within parts of the same type. This avoids over-constraining the shape model. Imposing these constraints prevents exact marginalization in $O(N^2P)$, however by using efficient search methods, an approximation can be computed using all hypotheses within a threshold δ of the best hypothesis that obeys the constraint ($\delta = e^{-10}$ in our experiments). In Figure 3, the mean time per iteration per frame in learning is shown as N and P are varied. In Figure 3(a) P is fixed at 6 and N varied from 20 up to 200 while recording the mean time per image over all EM iterations in learning. The curve has a quadratic shape with the time per image still respectable even for $N = 200$. It should be noted that a full model cannot be learnt with $N > 25$ due to memory requirements. In Figure 3(b) N is fixed at 20 and P varied from 2 to 13 with the mean time per image plotted on a log-scale y-axis. The curve for the full model is a straight line as expected from the $O(N^P)$ complexity, stopping at $P = 7$ owing to the memory overhead. The star model's curve, while also roughly linear, has a much flatter gradient: a 12 part star model taking the same time to learn as a 6 part full model.

The optimal choice of feature types is made using a validation set. For each dataset, given a pre-defined number of parts, seven models each with different combinations of types are learnt: Kadir & Brady (KB); multi-Scale Harris (MSH); Curves (C); KB + MSH; KB + C; MSH + C; KB + MSH + C. In each case, the parts are split evenly between types. In cases where the dataset is small and the validation set would be too small to give an accurate estimate of the error, the performance on the training set was used to select the best combination.

Learning is fairly robust, except when a completely inappropriate choice of feature type was made in which case the model occasionally failed to converge, despite multiple re-runs. A major advantage of the HSM is the speed of learning. For a 6 part model with 20 detections/feature-type/image the HSM typically takes 10 minutes to converge, as



(a) (b)

Fig. 3. Plots showing the learning time for a star model with different numbers of parts (P) and detections per image (N). (a) P fixed to 6 and N varied from 20 to 200. The curve has a quadratic shape, with a reasonable time even for $N = 200$. (b) N fixed to 20 and P varied from 2 to 13, with a logarithmic time axis. The full model is shown with a dashed line and the star model with a solid line. While both show roughly exponential behavior (i.e. linear in the log-domain), the star model's curve is much flatter than the full model.

opposed to the 24 hours of the fully connected model – roughly the same time as a 12 part, 100 detections/feature-type/image would with the HSM. Timings are for a 2Ghz Pentium 4.

2.4 Recognition using features

For the HSM, as with the fully connected Constellation Model of [8], recognition proceeds in a similar manner to the learning process. For a query image, regions/curves are first found using an feature detector. The learnt model is then applied to the regions/curves and the likelihood of the best hypothesis computed using the learnt model. This likelihood is then compared to a threshold to determine if the object is present or not in the image. Note that as no ordering constraint is needed (since no parameters are updated), this is an $O(N^2P)$ operation.

Good performance is dependent on the features firing consistently across different object instances and varying transformations. To ensure this, one approach is to use a very large number of regions, however the problem remains that each feature will still be perturbed slightly in location and scale from its optimal position so degrading the quality of the match obtainable by the model. We address these issues in the next section.

2.5 Complete recognition without features

Relying on unbiased, crude feature detectors in learning is a necessary evil if we wish to learn without supervision: we have no prior knowledge of what may or may not be informative in the image but we need a sparse representation to reduce the complexity of the image sufficiently for the model learning to pick out consistent structure. However in recognition, the situation is different. Having learnt a model, the appearance densities model the regions of the image we wish to find. Our complete approach relies on these densities having distinctive mean and a sufficiently tight variance so that they can be used for soft template matching.

The scheme, based on Feltzenswalb and Huttenlocher [6], proceeds in two phases: first, the appearance densities are run completely over the entire image (and at different scales). At each location and scale, we compute the likelihood ratio for each part. Second, we take advantage of the Star model for location and employ the efficient matching scheme proposed by [6], which enables the global maximum of both appearance and location to be found within the image. The global match found is clearly superior to the maximum over a sparse set of regions. Additionally, it allows us to precisely localize the object (and its parts) within the image. See figure 4 for an example.

In more detail, each PCA basis vector is convolved with the image (employing appropriate normalizations), so projecting every patch in the image into the PCA basis. While this is expensive ($O(k^2N)$, where N is now the number of pixels in the image and k is the patch size) this only needs to be performed once regardless of the number of category models that will evaluate the image. For a given model, the likelihood ratio of each part's appearance density to the background density is then computed at every location, giving a likelihood-ratio map over the entire image for that part. The cost is $O(dN)$, where the dimension of the PCA space, d is much less than k^2.

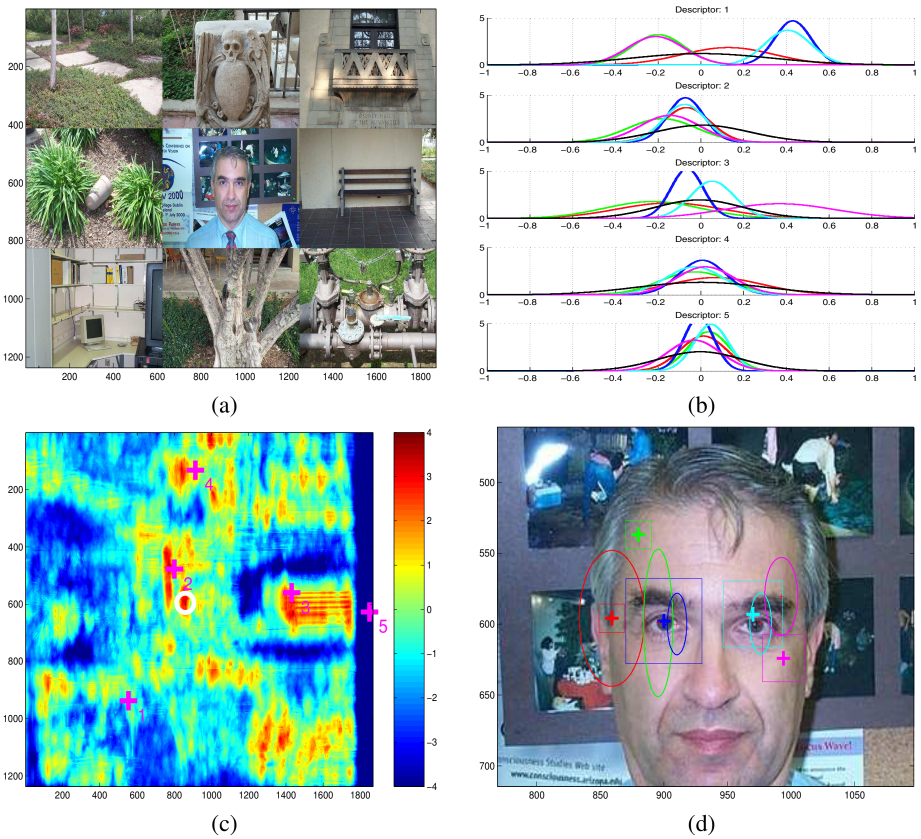
We then introduce the shape model, which by the use of distance transforms [6], reduces the cost of finding the optimal match from $O(N^2P)$ to $O(NP)$. Note that we cannot use this trick in learning since we need to marginalize out over all possible matches, not just the optimal. Additionally, the efficient matching scheme requires that the location model be a tree. No ordering constraint is applied to the part locations hence the approximations necessary in learning are not needed.

3 Experiments

We investigate the performance of the HSM in a number of ways: (i) we compare to the fully connected model; (ii) the effect of increasing the number of parts and detections/image; (iii) the difference between feature-based and complete recognition.

3.1 Datasets

Our experiments use a variety of datasets. Evaluation of the HSM using feature-based detection is done using nine widely varying, unnormalized, datasets summarized in Table 1. While some are relatively consistent in nature (Motorbikes, Faces) others were collected from Google's image search and are not normalized in any way so are highly variable (Camels, Bottles). Guitars and Houses are two diverse datasets, the latter of which is highly variable in nature. The negative test set consists of a variety of scenes around Caltech. All datasets are available from our website [7]. In recognition, the test is a simple object present/absent with the performance evaluated by comparing the equal error rates (p(Detection)=1-p(False Alarm)). To test the difference between feature-based and complete recognition where localization performance is important,



(a) (b) (c) (d)

Fig. 4. An example of the complete recognition operation on a query image. **(a)** A mosaic query image. **(b)** First five descriptor densities of a 5 part face model (black is background density). **(c)** Overall matching probability (red is higher). The global optimum indicated by the white circle, while the magenta $+$'s show the maximum of each part's response. Note they are not in the same location, illustrating the effect of the shape term. **(d)** Close-up of optimal fit with shape model superimposed. Crosses indicates matched location of each part, with the squares showing their scale. The ellipses show the variance of the shape model at 1 standard deviation.

the UIUC Cars (Side) dataset [1] is used. In this case the evaluation in recognition involves localizing multiple instances of the object.

3.2 Comparison of HSM and full model

We compare the HSM directly with the fully connected model [8], seeing how the recognition performance drops when the configuration representation is simplified. The results are shown in Table 1. It is pleasing to see that the drop in performance is relatively small, only a few percent at most. The performance even increases slightly in cases where the shape model is unimportant. Figures 6-9 show star models for guitars, bottles and houses.

Dataset	Total size of dataset	Full model test error (%)	Star model test error (%)
Airplanes	800	6.4	6.8
Bottles	247	23.6	27.5
Camels	350	23.0	25.7
Cars (Rear)	900	15.8	12.3
Faces	435	9.7	11.9
Guitars	800	7.6	8.3
Houses	800	19.0	21.1
Leopards	200	12.0	15.0
Motorbikes	900	2.7	4.0

Table 1. A comparison between the star model and the fully connected model across 9 datasets, comparing test equal error rate. All models used 6 parts, 20 Kadir & Brady detections/image. In general, the drop in performance is a few percent when using the simpler star model. The high error rate for some classes is due to the inappropriate choice of feature type.

3.3 Heterogeneous part experiments

Here we fixed all models to use 6 parts and have 40 detections/feature-type/frame. Table 2 shows the different combinations of features which were tried, along with the best one picked by means of the training/validation set. We see a dramatic difference in performance between different feature types. It is interesting to note that several of the classes perform best with all three feature types present. Figure 6 shows a heterogenous star model for Cars (Rear).

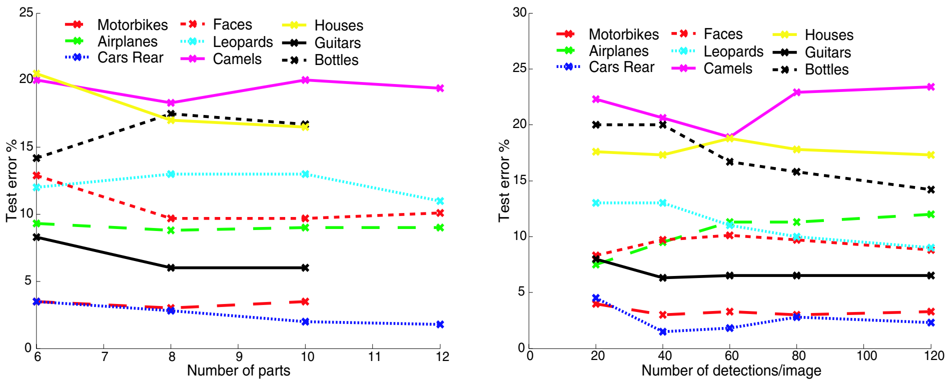
Dataset	KB	MSH	C	KB,MSH	KB,C	MSH,C	KB,MSH,C
Airplanes	**6.3**	22.5	27.5	11.3	13.5	18.3	12.5
Bottles	24.2	23.3	17.5	24.2	20.8	**15.0**	17.5
Camel	25.7	**20.6**	26.9	24.6	24.0	22.9	24.6
Cars (Rear)	11.8	6.0	5.0	2.8	4.0	5.3	**2.3**
Faces	10.6	16.6	17.1	12.0	13.8	12.9	**10.6**
Guitars	**6.3**	12.8	26.0	8.5	9.3	18.8	12.0
Houses	**17.0**	22.5	36.5	20.8	23.8	26.3	20.5
Leopards	14.0	18.0	45.0	**13.0**	23.0	23.0	18.0
Motorbikes	**3.3**	3.8	8.8	3.0	3.3	3.8	3.5

Table 2. The effect of using a combination of feature types on test equal error rate. Key: KB = Kadir & Brady; MSH = Multi-scale Harris; C = Curves. All models had 6 parts and 40 detection/feature-type/image. Figure in bold is combination automatically chosen by training/validation set.

3.4 Number of parts and detections

Taking advantage of the efficient nature of the star model, we now investigate how the performance alters as the number of parts and the number of detections/feature-type/frame is varied. The choice of features-types for each dataset is fixed for these experiments, using the optimal combination, as chosen in Table 2.

As the number of parts in the model is increased (for a fixed number of detections/frame) some of the categories show a slight change in performance but many remain constant. Examination of the models reveals that many of the additional parts do not find stable features on the object, suggesting that more features on the image are required. Increasing the number of detections/feature-type/image increases the error rate slightly in some cases such as camels, since many of the additional detections lie in the background of the image, so increasing the chances of a false positive. With a suitable combination of feature-types however, the increased number of parts and detections can give a more complete coverage of the object, improving performance (e.g. Cars (Rear) where the error drops from 4.5% at 8 parts to 1.8% with 12 parts, using 40 detections/image of all 3 feature types).



(a)

(b)

Fig. 5. (a) Test equal error rate versus number of parts, P, in the star model for 40 detections/feature-type/image. **(b)** Test equal error rate versus the number of detections/feature-type/image, N, for 8 part star models. In both cases the combinations of feature-types used was picked for each dataset from the results in Table 2 and fixed.

3.5 Complete search experiments

We now investigate the performance of feature-based recognition versus the complete approach. Taking the 8-part Cars (Rear) model shown in Figure 6, we apply it completely to the same test set resulting in the equal error rate dropping from 4.5% to 1.8%. Detection examples for the complete approach are shown in Figure 11, with the ROC curves for the two approaches shown in Figure 11(b).

The localization ability of the complete approach is tested on the Cars (Side) dataset, shown in Figure 10. A fully connected model (Figures 10 (a) & (b)) was learnt and then decomposed into a star model and run completely over the test set. An error rate of 7.8% was achieved – a decrease from the 11.5% obtained with a fully connected model using feature-based detection in [8]. The performance gain shows the benefits of using the complete approach despite the use of a weaker shape model. Examples of the complete star model localizing multiple object instances can be seen in Figure 10(c).

4 Summary and Conclusions

We have presented a heterogeneous star model. This model retains the important capabilities of the constellation model [8, 21], namely that it is able to learn from unsegmented and unnormalized training data; and in recognition on unseen images it is able to localize the detected model instance. The HSM outperforms the constellation model on almost all of the six datasets presented in [8]. It is also faster to learn, and faster to recognize (having $O(NP)$ complexity in recognition rather than the $O(N^P)$ of the constellation model). We have also demonstrated the model on many other object categories varying over compactness and shape. Note that while other models and methods have achieved superior performance to [8], for example [5, 15, 18, 19], they are unable to both learn in a weakly-supervised manner and localize in recognition.

There are several aspects of the model that we wish to improve and investigate. Although we have restricted the model to a star topology, the approach is applicable to a trees and k-fans [4], and it will be interesting to determine which topologies are best suited to which type of object category.

Acknowledgments

We are very grateful for suggestions from and discussions with Michael Isard, Dan Huttenlocher and Alex Holub. Financial support was provided by: EC Project CogViSys; EC PASCAL Network of Excellence, IST-2002-506778; UK EPSRC; Caltech CNSE and the NSF.

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. E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 109–124, 2002.
3. M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *Int. Workshop on Automatic Face and Gesture Recognition*, 1995.
4. D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, volume 1, pages 10–17, 2005.
5. 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.
6. P. Feltzenswalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, January 2005.

7. R. Fergus and P. Perona. Caltech Object Category datasets. `http://www.vision.caltech.edu/html-files/archive.html`, 2003.
8. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
9. R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. Springer-Verlag, May 2004.
10. M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92, Jan. 1973.
11. C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.
12. 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.
13. T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
14. Y. Ke and R. Sukthankar. PCA–SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, June 2004.
15. 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.
16. D. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii*, pages 682–688. Springer, December 2001.
17. P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, pages 55–68, 2004.
18. A. Opelt, A. Fussenegger, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, 2004.
19. 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.
20. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, pages 762–769, 2004.
21. 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.

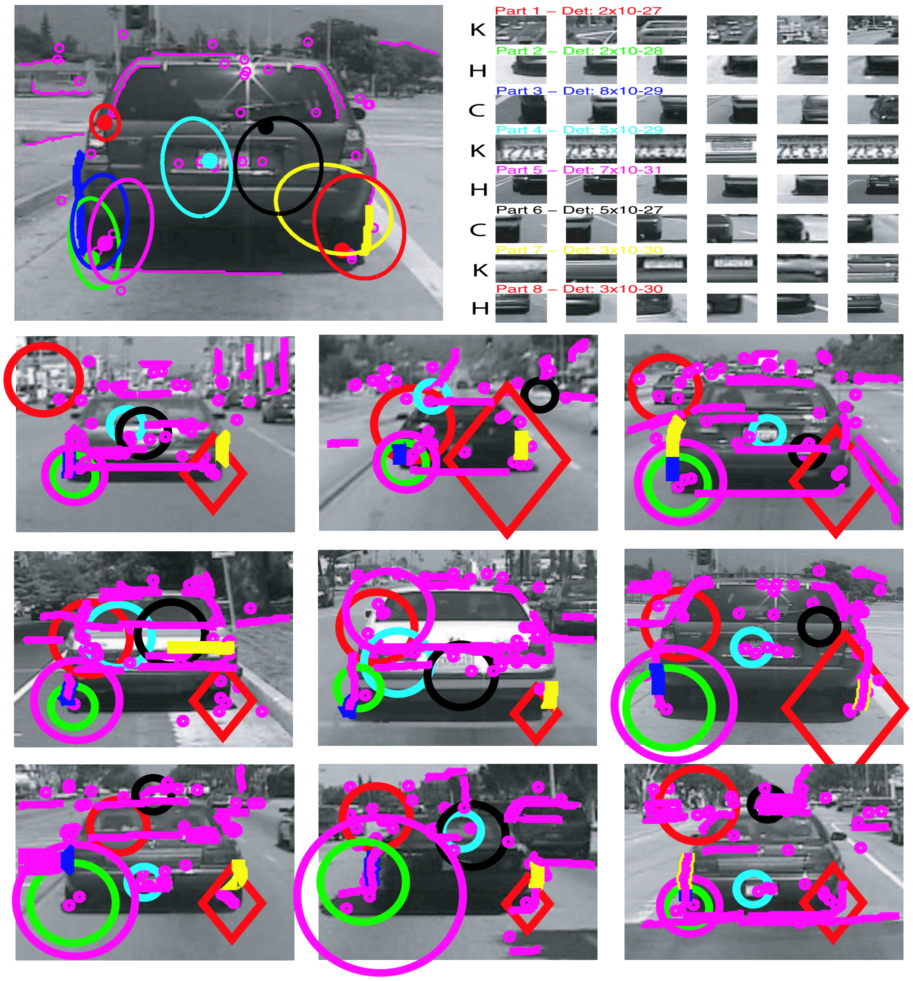


Fig. 6. An 8 part heterogeneous star model for Cars (Rear), using all three feature types (Kadir & Brady (K); multi-Scale Harris (H); Curves (C)) **Top left**: Detection in a test image with the spatial configuration model overlaid. The coloured dots indicate the centers of regions (K or H) chosen by the hypothesis with the highest likelihood. The thick curve in red is the curve selected by one of the curve parts – the other curve part being unassigned in this example. The magenta dots and thin magenta curves are the centers of regions and curves assigned to the background model. The ellipses of the spatial model show the variance in location of each part. The landmark detection is the top left red one. **Top right**: 7 patches closest to the mean of the appearance density for each part, along with the determinant of the variance matrix, so as to give an idea of the relative tightness of each distribution. The colour of the text corresponds to the colour of the dots in the other panels. The letter by each row indicates the type of each part. **Bottom panel**: More detection examples. Same as top left, but without the spatial model overlaid. The size of the coloured circles and diamonds indicate the scale of regions in the best hypothesis. The test error for this model is 4.5%.

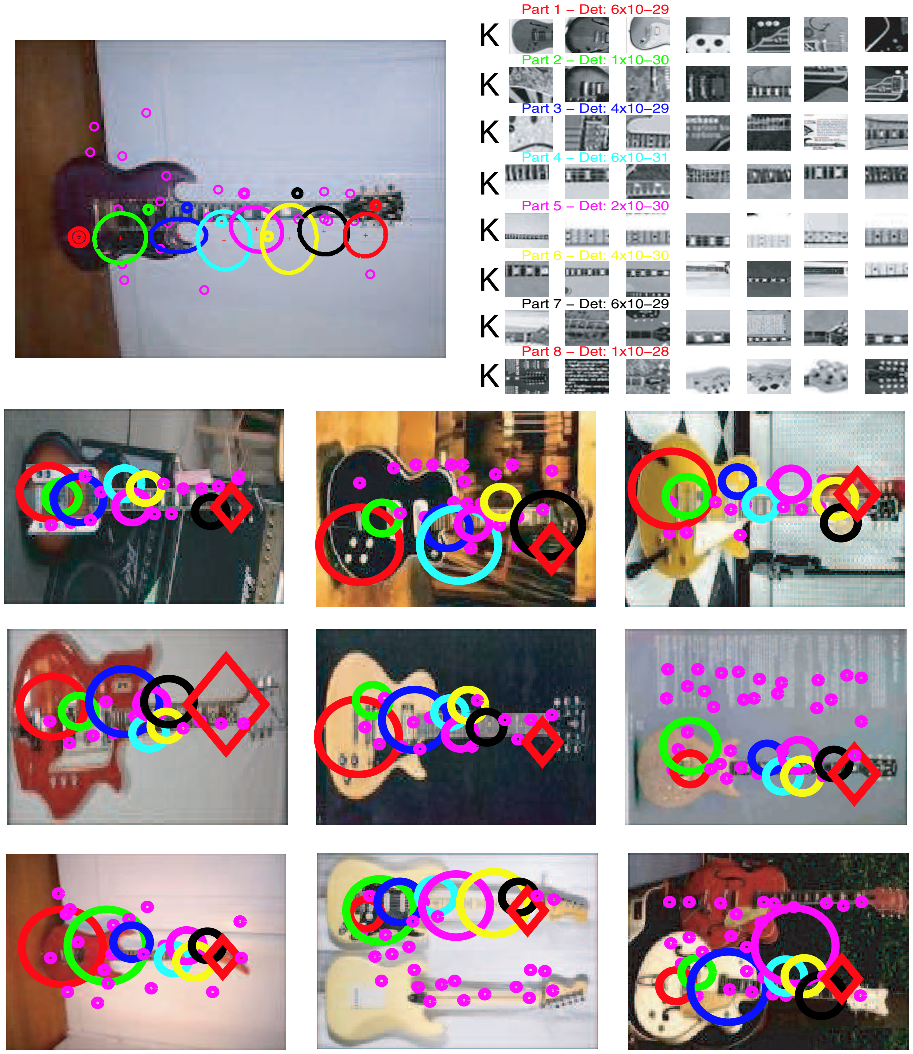


Fig. 7. An 8 part model for Guitars, using 40 Kadir & Brady features per image. 6.3% test error.

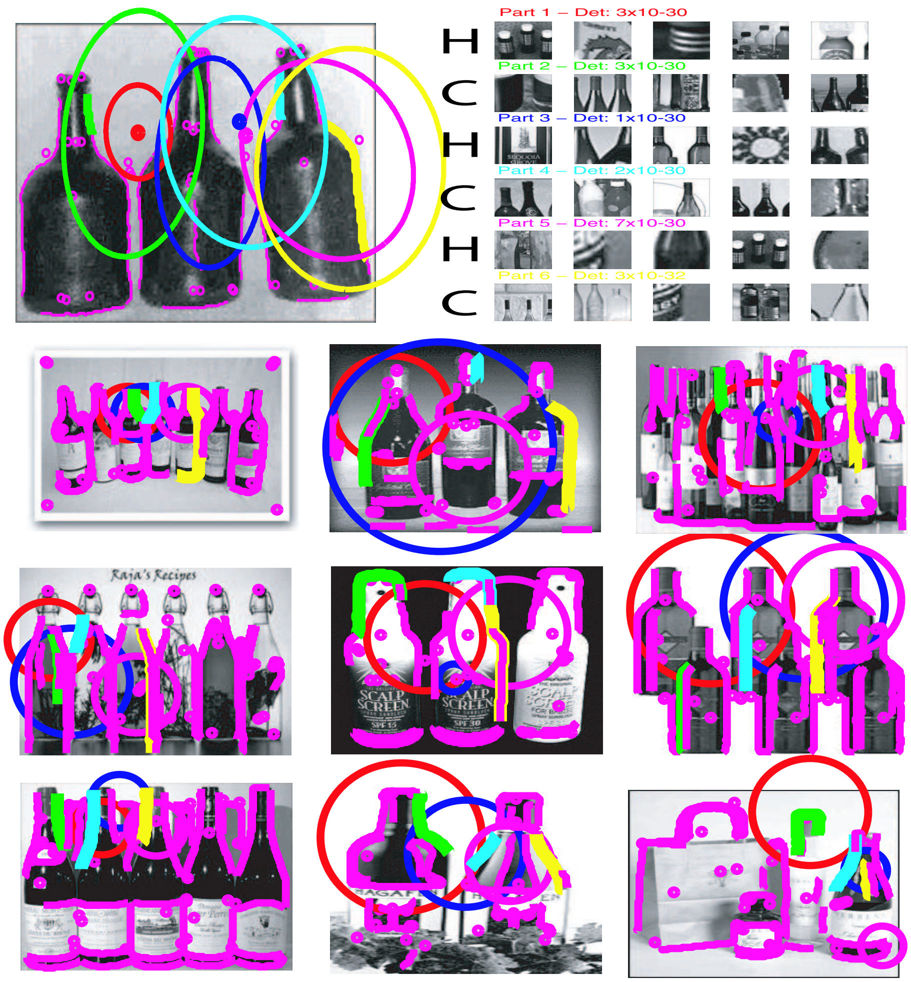


Fig. 8. A 6 part model for Bottles, using a maximum of 20 Harris regions and 20 Curves per image. 14.2% test error.

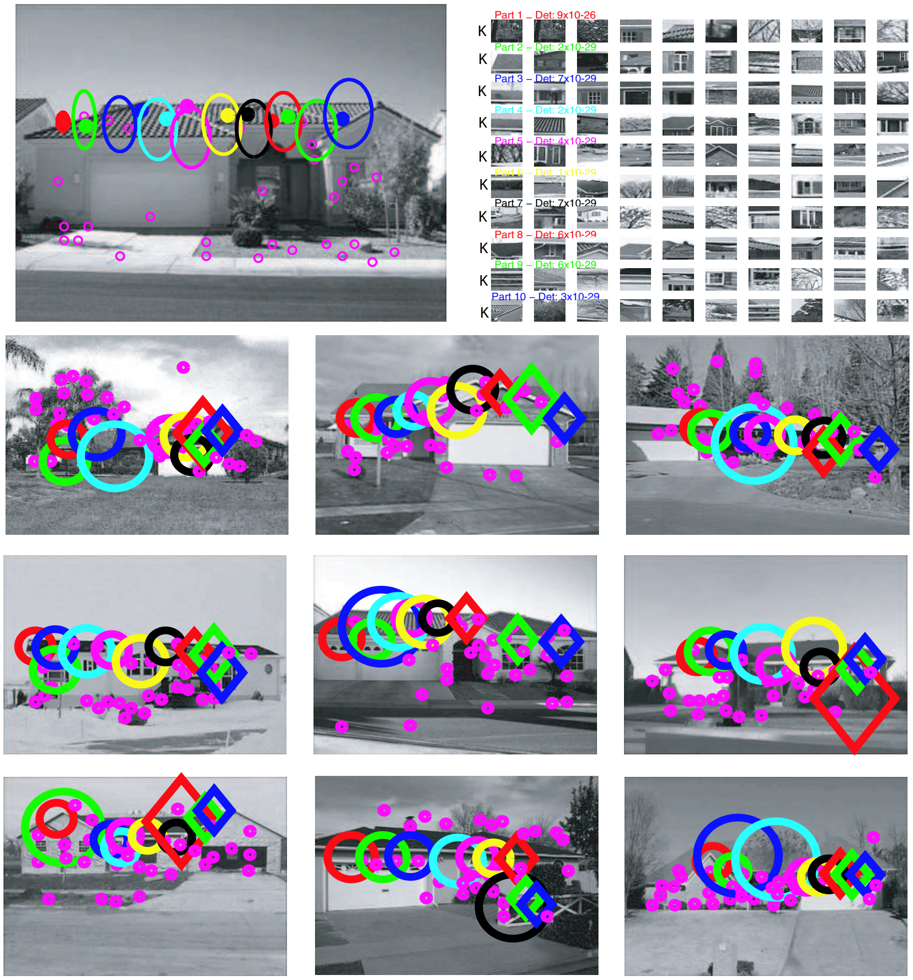
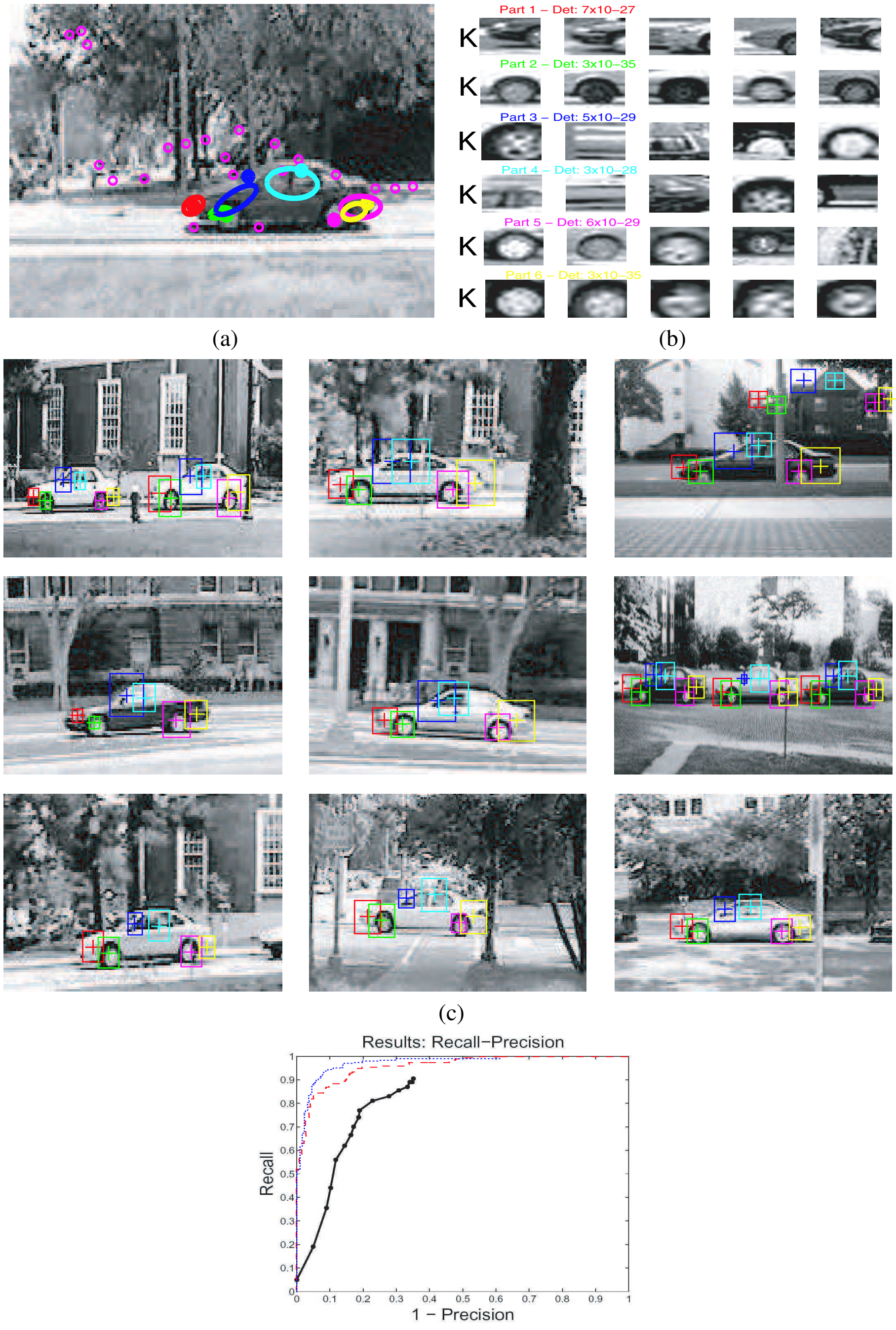


Fig. 9. A 10 part model for Houses, using 40 Kadir & Brady features per image. 16.5% test error.



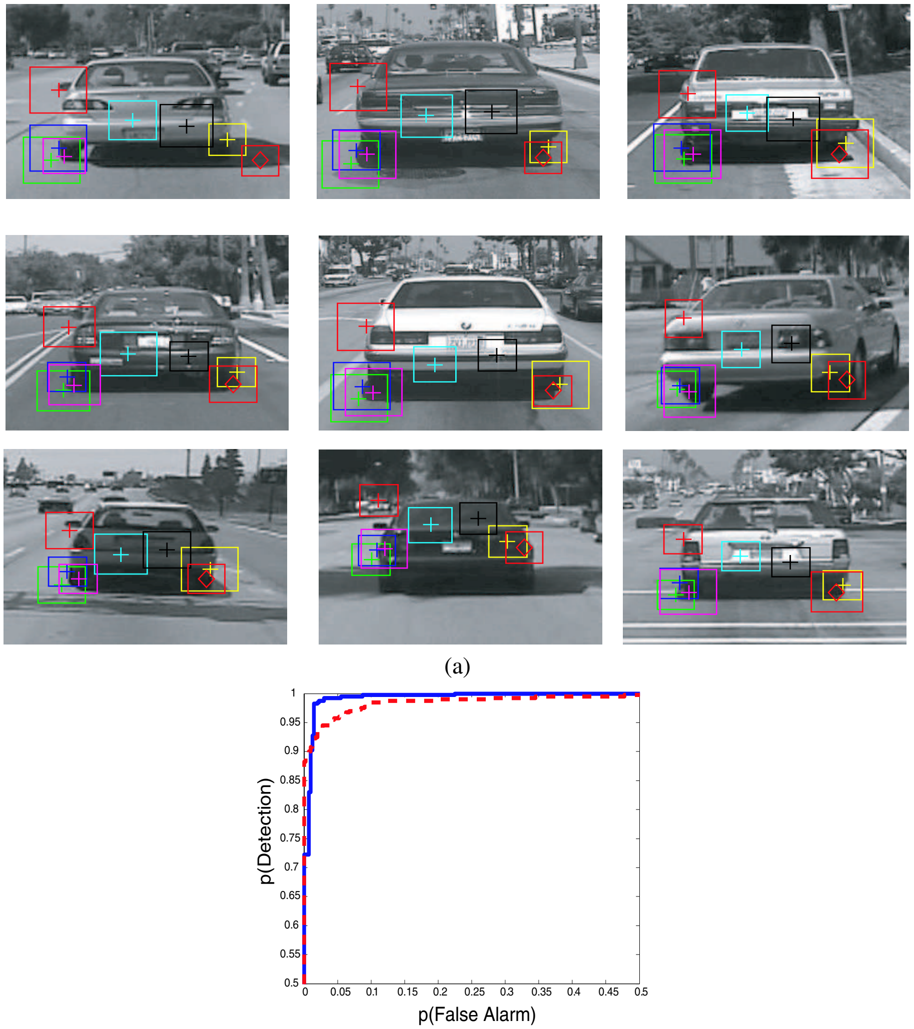
(a)

(b)

(c)

(d)

Fig. 10. **(a)** & **(b)** A 6 part model Cars (Side), learnt using Kadir & Brady features. **(c)** Examples of the model localizing multiple object instances by complete search. **(d)** Comparison between feature-based and complete localization for Cars (Side). The solid recall-precision curve is [1]; the dashed line is the fully connected shape model with feature-based detection [8] and the dotted line is the complete-search approach with star model, using the model shown in (a) & (b). The equal error rate of 11.5% from [8] drops to 7.8% when using the complete search with the star model.



(a)

(b)

Fig. 11. (a) Detection examples of the 8 part Cars (Rear) model from Figure 6 being used completely. **(b)** ROC curves comparing feature-based (dashed) and complete detection (solid) for the 8 part Cars (Rear) model in Figure 6. Equal error improves from 4.5% for feature-based to 1.8% for complete.