Learning object classes from structure

Bai Xiao, Yi-Zhe Song and Peter M. Hall
Media Technology Research Centre
Department of Computer Science
University of Bath

Abstract

The problem of identifying the class of an object from its visual appearance has received significant attention recently. Most of the work to date is premised on photometric measures, often building codebooks made from interest regions. All of it has been tested only on photographs, so far as we know. Our approach differs in two significant ways. First, we do not build a codebook of interest regions but instead make use of a hierarchical description of an image based on a watershed transform. Root nodes in the hierarchy are putative objects to be classified. Second, we classify these putative objects using a vector of fixed length that represents the structure of the hierarchy below the node. This allows us to classify not just photographs, but also paintings and drawings of visual objects.

1 Introduction

The problem of identifying the class of an object from its visual appearance has received significant attention recently. This problem is to be differentiated from that of object segmentation; approximate segmentations of the object in question or even parts of the object are proving sufficient. The problem should also be differentiated from that of specific object identification, there are degrees of freedom open to classes of objects that are unavailable to specific objects.

There are many accounts of the problem of identifying visual classes, including active appearance models [4], pictorial structures [6] and tensor faces [20]. A category of solution of particular interest here acquires the statistics of appearance and relative spatial distribution of local features, in a semi-unsupervised environment (training assumes only that images depict an object known to belong to a given class). Weber *et al.* [21] number amongst the earliest of authors proposing a solution in this category; they learn to recognise letters of the alphabet. Importantly, they showed it is possible to identify object classes without prior explicit segmentation. Leibe and Schiele [2] continued in this vein by demonstrating it is possible to simultaneously segment and recognise a wider range of object classes. Using a Harris detector [13] to locate the centre of interest regions, they cluster features into a codebook. The spatial relation between these parts is determined relative to a centroid determined by a voting system. Fergus *et al.* [7] adopted a similar approach, but included a greater array of invariants and so produced a system that might be expected to be more robust to changes in scale, orientation and so on (but we are not aware of any authoritative comparison). Similar work continues to date, see [19, 11, 18, 1] for example.

BMVC 2007 doi:10.5244/C.21.27

All of this work is premised on photometric features. Yet it is un-controversial to label an object in a drawing or a painting in just the same way one might label a photograph: "Ce n'est pas un pipe" (thanks to Magritte) applies equally to images of all kinds. Accordingly, we would like our visual object classifier to work with images of all kinds. In this paper we propose that image structure should play an important role when classifying objects across image kinds. It is true that graphical structure is included as in many of the models already cited, and is dominant when articulated objects are of interest [6]. Importantly, we can cluster objects of the same class in feature space, regardless image class.

The importance of structure over appearance is evident when we consider our ability to see objects where none exist: in clouds, stains on walls are all too readily interpreted as faces, cars, trains and so on. In all of these cases are difficult to explain without appeal to structure, because the appearance of parts actually observed look nothing like the parts imagined. Interestingly, because the structure of clouds is highly variable it can be argued that seeing clouds exactly as clouds provides a counter example to the importance of structure. We conclude that neither appearance nor structure alone are sufficient to explain the phenomenon of object classification in the round.

The use of structure is not the only novelty in this paper. Our approach is not premised a codebook learned by classifying interest features, as is the greater majority of literature to date, but on watershed regions. Specifically we build a hierarchical description of an image by grouping watershed regions. We chose to group watershed regions because we observed that they tend to isolate features of interest and which are thought to be important for image description: dots, T-junctions, line ends, corners, contrast edges and so on. Some watershed regions enclose no features at all. Consequently watershed regions have the potential to be a useful alphabet that includes whitespace. The fact that watershed regions can vary in shape has proven of little consequence, at least we have not noticed any significant effect.

Watershed regions over-segment an image; we are not the first to group them, but there is necessary novelty in our grouping approach. The arises in part because we seek to acquire objects classes in a semi-unsupervised environment, whereas watershed regions are usually grouped with a view to image segmentation. For example, Haris *et al.* [15] use the watershed of the magnitude of the image derivative. They adopt a hierarchical agglomerative clustering approach which terminates when the cost of grouping any pair rises above a threshold (which varies from image to image). Gauch [10] also segments by clustering watersheds of the derivative image, filtering the image in scale space to do so. Malpica *et al.* [17] build on the work of Haris *et al.* [15] by introducing multidimensional description for regions, rather than the scalar gray level. We also use a multi-dimensional description, but handle it differently from Malpica *et al.* [17], as we will make clear below.

We make two main contributions in this paper:

- We extend watershed clustering to allow multidimensional vector descriptions of regions, and in particular demonstrate the value of decorrelating the vector elements. We also provide a halting criterion that requires no threshold, but is independent of the picture.

- We show that clustering in pattern space places photographs and artwork of the same object into the same category, which provides ground for objection identification.

The first is discussed together in Section 2, the second in Section 3.

2 Grouping watershed regions

This section describes our approach to grouping watershed regions. The aim is to produce a rough segmentation of visual objects. We use hierarchical agglomerative clustering to group; that is we group pairs of regions, thereby creating a new region. The grouping method is described in Subsection 2.1. We could continue until just one region remains but have a principled way to halt the grouper; Subsection 2.2. Having described the structure of our procedure we then demonstrate the value of using a decorrelated feature vector, in which we acquire a decorrelation matrix via supervised learning, see Subsection 2.3 and Subsection 2.4.

2.1 Grouping Method

We use Luc and Soille's watershed algorithm [16] to obtain an initial set of watershed regions. Unlike work aimed to segment images, see [15, 10] for example our watersheds are computed directly from the intensity image, not from it derivative magnitude. Our grouping method is based on the work of Haris *et al.* [15]. They define the cost of grouping a pair of regions as $N_i N_j/(N_i+N_j)(\mu_i-\mu_j)^2$ in which μ is the average gray value over the region of N pixels. They group adjacent regions with the smallest error value, creating a new region from the union of the regions and their boundary. This continues until the cost exceeds a pre-computed threshold which is based on the noise distribution of the image. We note that this threshold is image dependent and that noise estimates are difficult to compute and often unreliable.

As mentioned, Malpica *et al.* [17] have used multi-variate descriptions of watershed regions based on the same grouper. Their extension uses the mean of a set of feature vectors in a region, μ. It redefines the cost as proportional to the sum the root of absolute differences of individual elements between means: $\sum_k |\mu_{ik}-\mu_{jk}|^{1/2}$, the scale factor $N_i N_j/(N_i+N_j)$ is the same. We take a very different stance: we consider the distribution of features vectors across a region, then enquire into the dissimilarity between two distributions.

We begin by supposing that each pixel $\mathbf{x}$ in a region, $\mathscr{R}$, supports a vector of measures $\mathbf{v}(\mathbf{x})$; the vector is typically formed by filtering with a filter bank. Next we approximate the distribution of the vectors in a region with a Gaussian. For the *i*th region we compute the number of pixels, the mean, and the covariance:

$$N_i = |\{\mathbf{x} \in \mathscr{R}_i\}| \tag{1}$$

$$\mu_i = \frac{1}{N_i}\sum_{\mathbf{x}\in\mathscr{R}_i} \mathbf{v}(\mathbf{x}) \tag{2}$$

$$\mathbf{C}_i = \frac{1}{N_i}\sum_{\mathbf{x}\in\mathscr{R}_i} (\mathbf{v}(\mathbf{x})-\mu_i)(\mathbf{v}(\mathbf{x})-\mu_i)^T \tag{3}$$

Together these terms make up an eigenmodel. We define the approximate error for each watershed region $\mathscr{R}_i$ which is $\mathbf{H}(\mathscr{R}_i) = \sum_{x\in\mathscr{R}_i}(\mathbf{v}(\mathbf{x})-\mu_i)^T(\mathbf{v}(\mathbf{x})-\mu_i)$. The ap-

proximation error for the whole picture is, can defined as $\mathbf{W}(\mathbf{P}) = \sum_{\mathscr{R}_i \in \mathbf{P}} \mathbf{H}(\mathscr{R}_i)$ where $\mathbf{P} = \mathscr{R}_1, \mathscr{R}_2, ..., \mathscr{R}_k$ is a k partition for the image.

Finding the optimal partition $\mathbf{W}_{min}$ can be transformed into a stepwise merging process(a greedy algorithm) which merges two watershed regions at each time which has minimum value of cost for merging them. The cost of grouped a pair of regions is

$$e_{ij} = \frac{N_i N_j}{N_i + N_j} (\mu_i - \mu_j)^T (\mu_i - \mu_j) \tag{4}$$

when two regions are merged we can efficiently compute a new Gaussian:

$$\mathbf{C}_{\text{new}} = \frac{N_i}{N_i + N_j} \mathbf{C}_i + \frac{N_j}{N_i + N_j} \mathbf{C}_j + \frac{N_i N_j}{(N_i + N_j)^2} (\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{5}$$

there is need for us to return to the image data. The final term allows for the difference between means. It can be thought of as measuring the volume of the symmetric difference of two classes. Its trace is proportional to the error as we define it, and is also related to the errors defined by Haris *et al.* [15] and Malpica *et al.* [17].

As the regions merge the feasibility of the Gaussian assumption declines. This is acceptable because so we are willing to accept broader approximations of larger region, which is in line with scale-space descriptions. Alternatively, the size of the approximation is appropriate to the size of the region.

2.2 Stopping Criterion

Now we turn our attention to the halting criterion. Rather than use a threshold we appeal to spectral graph theory [3] to halt the merging process. Each step of the grouping process produces a new partition of the original watershed regions. The elements in the partition delimit regions of the image which, we hope, correspond to visual objects. At least, we desire the partition to *roughly* segment visual objects — recall that our aim in object class identification, not image segmentation.

The watershed regions in any element in a partition form a graph of nodes and arcs: watershed regions are nodes, arcs connect adjacent regions. Hence each partition has an associated structure inherited from the watershed transform. We make use if the structure within partition elements to develop a halting criterion; we assume there is no connectivity between sections of the partition (because the image comprises the union of independent objects).

Our halting criterion uses the Laplacian energy of a graph [12]. For a graph $G = (V, E)$, where V is the node sets and E is the arc relationship among the nodes V. We can construct the adjacency matrix A for the graph, the Laplacian matrix L is defined as $L = D - A$, where D is the degree matrix. The Laplacian energy for graph G is defined as

$$E(G) = \sum_{i=1}^{|V|} \left| \lambda_i - 2\frac{m}{|V|} \right| \tag{6}$$

In which: the λ_i are eigenvalues of the Laplacian matrix L; m is the sum of the arc weights over the whole graph, which is just the sum of the elements in matrix A divided by 2; $|V|$ is the number of nodes in graph. The arc weights are defined here as a function

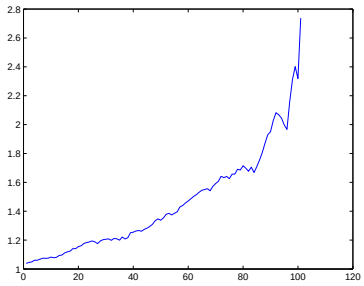

Figure 1: An example of a partition energy plot as a function of iteration number. We halt the grouping process at the right-most local minimum.

of the the merging error in Equation 4 $w_{ij} = \exp(-e_{ij}/e_{\max})$. The normalising factor, $e_{\max}$, is the maximum error over the all the original watershed regions, not just those in a particular element (region, a putative object) of a particular partition (regions collected into an image). It is therefore a constant for a given picture.

The $E(G)$ gives the energy of an element in a partition. We define the energy of the partition to be the average *normalised* Laplacian energy over all N elements of the partition:

$$\xi = \frac{1}{N}\sum_{i=1}^{N}\frac{E(G_i)}{|V_i|} \tag{7}$$

We use normalised Laplacian energy $E(G)/|V|$ to estimate the average connection energy per node; this is rather like normalising spatial measures (such as image moments) by area.

Now, the partition energy ξ is a function of iteration number — simply because each step of the grouping process yields a new partition; therefore we should write $\xi(k)$. Halting conditions occur when $\xi(k)$ is a local minimum, that is $d\xi(k)/dk = 0$ and $d^2\xi(k)/dk^2 > 0$. Figure 1 shows that there is more than one such local minimum. We currently select the last of those, this gives a partition with the fewest number of regions.

2.3 Decorrelating feature vectors

This grouping method as described so works reasonably well, but performance is improved by decorrelating terms in the vector $\mathbf{x}$. We use a decorrleating linear transform, a square matrix $\mathbf{K}$, so that

$$\mathbf{x} \mapsto \mathbf{K}\mathbf{x} \tag{8}$$

Now the error is proportional to $(\mu_i - \mu_j)^T\mathbf{K}^T\mathbf{K}(\mu_i - \mu_j)$. Now, since $\mathbf{x}^T\mathbf{A}\mathbf{x}$ can be written as an inner product $\mathbf{A}_{ij}(\mathbf{x}\mathbf{x}^T)_{ij}$ (using tensor notation) we see that the error is just a linear combination of the all terms in the difference between the means, not just the diagonal.

The rational underpinning decorrelation arises from the desire for a small inner product in the cost term of Equation 4. Clearly, orthogonal vectors give a zero inner product, therefore it is sensible used decorrelate feature vectors when compute the cost.

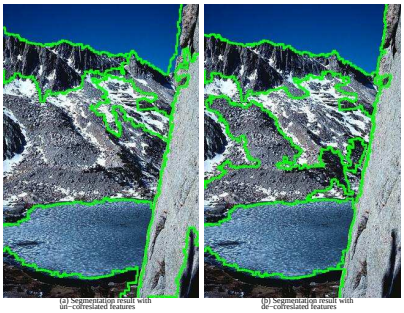

Figure 2: Left Image: Segmentation result with uncorrelated features; Right Image: Segmentation result with decorrelated features.

The decorrelation matrix **K** comes via supervised learning: users pick pairs of adjacent regions they wish to be grouped in 10 different pictures. During training, a user may group in hierarchical fashion. We recorded all pairings (about 1500 in total) and thereby obtain a training collection of $\mathbf{u}_{ij} = (\mu_i - \mu_j)$ vectors. Supervised training helps ensure that decorrelation has at least some grounding in human perception – and using a trained **K** does improve grouping — but we are clear in making no stronger claim about **K** than this.

The feature de-correlation matrix **K** is computed using a fixed-point based variant of independent component analysis (ICA) due to Hyvärinen [14]; which is computationally very efficient yet statistically robust. ICA differs from PCA in that, ICA yields statistically independent components rather than simple decorrelation. Given the set of input training vectors $\mathbf{u}_{ij} \in \Re^n$, we determine a $n \times n$ matrix **K**, such that the components of the mapped vector **Kx** are mutually independent.

2.4 Grouping Results

In all of the results shown, a 4-dimensional feature vector $f = (r,g,b,v)$ for each pixel was used, where r,g,b are just the colour channels of the image and v being the gradient magnitude. Results of our grouper are shown in Figure 2 and 3. Within each Figure, we compare differences in the final segmentations using uncorrelated and decorrelated features.

In Figure 2, the left image corresponds to the final segmentation result using uncorrelated features, while the right image used decorrelated features. It is obvious that result on the right segments out the mountain on the right and the lake well, plus a clearer separation of the mountain at the back. Again, in Figure 3, the segmentation on the right (using decorrelated features) has a better segmentation on the house, car, grass and mountain.

3 Clustering in Pattern Space

In this section, we explain how objects classes can be automatically learned from the hierarchical descriptions output by our grouper. The significant novelty is the use of structure as a feature in its own right, which means we form object classes using images of different kinds: photographs, paintings, and drawings.

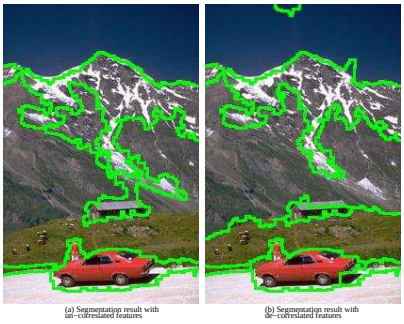

Figure 3: Left Image: Segmentation result with uncorrelated features; Right Image: Segmentation result with decorrelated features.

The idea here is the structure of an object — the structure output by the grouping process — is more or less invariant to changes of image type: the structure of a face in a photograph is the same as the structure of a face in a drawing. We make use of methods from graph spectral theory that encode structures, even noisy structures, into a pattern space of fixed dimension. This furnishes us with the opportunity to cluster objects on the basis of their structure. As the experiment below shows, we are able to cluster objects in a single class regardless of the kind of image they come from.

Given a root node output by the grouper (that is, an element in the final partition) we form a feature vector for it. This feature vector encodes information regarding both content and structure. Content information is held in the eigenmodels associated with every region in the merging process. Recall that this describes decorrelated feature distributions rather than photometric data. The eigenvalues of the covariance matrix for the node are taken to represent its content. Structural information comes from the hierarchy generated by the merging process. Each root note expands into a binary tree; we compute the eigenvalues of the Laplacian matrix of this tree. Note that this Laplacian differs from that used for the halting condition: this one uses a tree output by a process rather than a spatially-dependent graph.

We form a vector of fixed size by choosing a fixed number of dominant eigenvalues both content and structure; we use the largest four from each. Now we have a vector of fixed size to describe root nodes. The structural component of this compound vector is particularly interesting: it is robust in the sense that graphs extraneous or ”noisy” arcs and nodes all project to about the same point. This is a very useful property because it means the result of this projection is robust to grouping error and also to watershed segmentations. Finally, the feature vector for each root node is subject to a multidimensional scaling transform [5] to project it into three dimensions. This reduction of dimension mitigates the ”curse of dimensionality” problem given sparse data, as we have. We use a Gaussian Mixture Model [9] to cluster feature vectors for root nodes, which is a fully automatic method — including choosing the number of clusters.

We now describe our object class identification experiments. We selected two different objects, faces and leaves. For each object we collected real world photos from the Oxford-Caltech database [8], and also paintings and drawings. Figure 4 shows the examples images. The grouper fully automatically partitioned each image. Then we extracted the feature vectors for each root node of the partition, and clustered them. Figure 6 shows

Figure 4: Original faces and leaves (Pictures, Painting & Drawing)

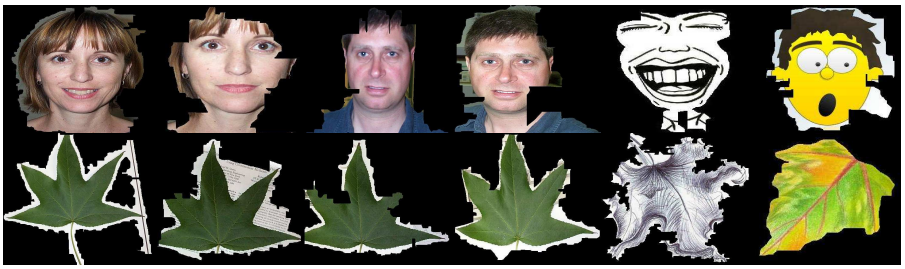

Figure 5: Segmented faces and leaves

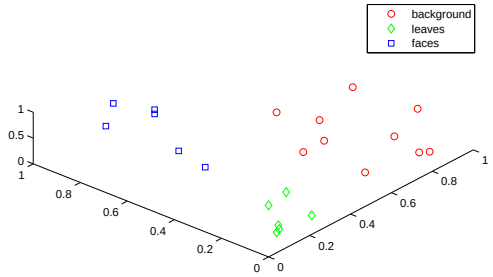

Figure 6: faces and leaves recognition results by using the Gaussian mixture model

the results; the leaves, faces and general background are well separated. To separate foreground from background, a user named(by pointing) each class; the nodes in the leaf and face class were then rendered to give Figure 5. As can be seen the faces and leaves have not been segmented from the background correctly but also clustered regardless of their image class.

4 Conclusions

This paper introduces a novel way to group watershed regions; it demonstrates the value of decorrelating feature vectors by training, and provides a robust halting condition that is picture independent. The output of the grouper is a set of regions, each of which expands

into a structure. We have encoded that structure in a vector of fixed length. This allowed us to cluster objects of the same class but from different image types. Except for the training stage of the grouper, the fact we use a semi-unsupervised approach for learning object class clusters, and the need to identify background explicitly our method is fully automatic. Wide variance in background is expected, whereas objects are expected to show much lower variance. This may be exploited in future work to remove the need for an user to differentiate between them.

The use of structure clearly enables objects from different image classes to be classified as one. It is not at all clear that alternative approaches to learning object classes can do likewise, because they are photometrically based. It is true that some use curves as the basis of class identification [1], but these curves are fitted to contrast edges in a photograph — this is not a suitable approach for line drawings. Watershed regions, on the other hand, tend to isolate regions of interest is all classes of image; as we observed at the top of this paper watershed regions provide a useful alphabet that include whitespace (so whitespace appears in our binary tree).

We believe this paper supports a considerable body of future work, an empirical comparison with contemporary methods for object class identification being just one direction. The ability to classify objects across image classes has implications for content based retrieval as well. In the future, we aim to improve the grouping process by investigating other means of feature vectors, which we hope will lead to better segmentations. Furthermore, we will test on more objects to evaluate the robustness of our algorihm.

5 Acknowledgements

This work was supported by EPSRC grant EP/D05429X/1

References

[1] A. Pinz A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. *ECCV*, 2:575–588, 2006.

[2] B. Leibe asnd B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, volume 2, pages 759–768, 2003.

[3] F. R. K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.

[4] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–, Freiburg, Germany, 1998. Springer-Verlag.

[5] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 1(61):55–79, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[8] R. Fergus, P. Perona, and A. Zisserman. Objects class recognition by unsupervised scale-invariant learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:264–271, 2003.

[9] M. A. F. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE PAMI*, 24(3):381–396, 2002.

[10] J.M. Gauch. Image segmentation and analysis via multiscale gradient watersheds. *IEEE Transactions on Image Processing*, 8(1):69–79, 1999.

[11] K. Grauman and T. Darrell. Unsupervised learning of catagories from sets of partially matching image features. In *CVPR*, 2006.

[12] I. Gutman and B. Zhou. Laplacian energy of a graph. *Linear Algebra and its Applications*, 44:29–37, 2006.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 189–192, Manchester, UK, 1998.

[14] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[15] K.Haris, S.Efstratiadis, N.Maglaveras, and A.Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. Image Processing*, 7(12):1684–1699, 1998.

[16] V. Luc and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[17] N. Malpica, J.E. Ortuño, and A. Santos. A multi-channel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24:1545–1554, 2003.

[18] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[19] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006.

[20] M. Alex O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, pages 447–460, 2002.

[21] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for visual object class recognition. In *ECCV, volume 1842 of Lecture Notes in Computer Science*, pages I:18–32, 2000.