Inexact Graph Retrieval

Benoit Huet and Edwin R. Hancock
Department of Computer Science
University of York, York, Y010 5DD, UK
huetb@cs.york.ac.uk erh@cs.york.ac.uk

Abstract

This paper describes a graph-matching technique for recognising line-pattern shapes in large image data-bases. We use a Bayesian matching algorithm that draws on edge-consistency and node attribute similarity. This information is used to determine the a posteriori probability of a query graph for each of the candidate matches in the data-base. The node feature-vectors are constructed by computing normalised histograms of pairwise geometric attributes. Attribute similarity is assessed by computing the Bhattacharyya distance between the histograms. Recognition is realised by selecting the candidate from the data-base which has the largest a posteriori probability.

1 Introduction

Broadly speaking there are two sources of information that can be tapped in the content based retrieval of images from large data-bases. The first of these is to use a compact summary of the image attributes. One of the best known examples here is the attribute histogram originally popularised by Swain and Ballard [1] for retrieving colour images from data-bases. The idea has been extended to texture [2], local feature orientation representations [3, 4] and the geometry of line-patterns [5]. The alternative to the use of attributes is to retrieve images based on their structural information content [6, 7]. Here the basic idea is to recognise relational object descriptions by comparing graph-structure.

Although both structural indexing and attribute histograms have been separately used to recall image data from data-bases, they have not been used in conjunction. The aim in this paper is to fill this gap in the literature by developing an efficient graph-matching algorithm which draws on attribute histograms to perform structural recognition. In fact, since Barrow and Popplestone [8] first suggested that relational structures could be used to represent and interpret 2D scenes, there has been considerable interest in the machine vision literature in developing practical graph-matching algorithms [9, 10, 11, 12]. The main computational issues are how to compare relational descriptions when there is significant structural corruption and how to search for the best match [11]. Despite resulting in significant improvements in the available methodology for graph-matching, there has been little progress in applying the resulting algorithms to large-scale object recognition problems. Most of the algorithms developed in the literature are evaluated for the relatively simple problem of matching a model-graph against a scene known to contain the relevant structure. A more realistic problem is that of taking a large number (maybe thousands) of scenes and retrieving the ones that best match the model. Because of the perceived fragility of the graph matching process, there has been rather less effort directed at attempting to retrieve shapes using relational information [7, 13] than in the use of low-level atttibute histograms.

Here we aim to fill this gap in the literature by using graph-matching as a means of retrieving the shape from a large data-base that most closely resembles a query shape. Although the indexation images in large data-bases is a problem of current topicality in the computer vision literature [14, 15, 1], the work presented in this paper is more ambitious. Firstly, we adopt a structural abstraction of the shape recognition problem and match using attributed relational graphs. Each shape in our data-base is a pattern of line-segments. The structural abstraction is a nearest neighbour graph for the centre-points of the line-segments. In addition, we exploit attribute information for the line patterns. Here the geometric arrangement of the line-segments is encapsulated using a histogram of Euclidean invariant pairwise (binary) attributes. For each line-segment in turn we construct a normalised histogram of relative angle and length with the remaining line-segments in the pattern. These histograms capture the global geometric context of each line-segment. Moreover, we interpret the pairwise geometric histograms as measurement densities for the line-segments which we compare

using the Bhattacharyya distance.

Once we have established the pattern representation, we realise object recognition using a Bayesian graph-matching algorithm. This is a two-step process. Firstly, we establish correspondence matches between the individual tokens in the query pattern and each of the patterns in the data-base. The correspondence matches are sought so as to maximise the *a posteriori* measurement probability. Here we use an extension of the graph-matching technique recently reported by Wilson and Hancock [12] in which we use the correspondence matches residing on the edges rather than the nodes of our ARG's to asses consistency. Once the MAP correspondence matches have been established, then the second step in our recognition architecture involves selecting the line-pattern from the data-base which has maximum matching probability.

2 MAP Framework

Formally our recognition problem is posed as follows. We abstract the shapes to be recognised as attributed relational graphs (ARG's). Each ARG in the database is a triple, $G = (V_G, E_G, A_G)$, where V_G is the set of vertices (nodes), E_G is the edge set ($E_G \subset V_G \times V_G$), and A_G is the set of node attributes. In our experimental example, the nodes represent line-structures segmented from 2D images. The edges are established by computing the N-nearest neighbour graph for the line-centres. Each node $j \in V$ is characterised by a vector of attributes, $\underline{x}_j$ and hence $A_G = \{\underline{x}_j | j \in V\}$. In the work reported here the attribute-vector is represents the contents of a normalised pairwise attribute histogram.

The data-base of line-patterns is represented by the set of ARG's $\mathcal{D} = \{G\}$. The goal is to retrieve from the data-base $\mathcal{D}$, the individual ARG that most closely resembles a query pattern $Q = (V_Q, E_Q, A_Q)$. We pose the retrieval process as one of associating with the query the graph from the data-base that has the largest *a posteriori* probability of match. In other words, the class identity of the graph which most closely corresponds to the query is

$$\omega_Q = \arg\max_{G' \in \mathcal{D}} P(G'|Q)$$

However, since we wish to make a detailed structural comparison of the graphs, rather than comparing their overall statistical properties, we must first establish a set of best-match correspondences between each ARG in the data-base and the query Q. The set of correspondences between the query Q and the ARG G at iteration n is a relation $f_G^{(n)} : V_G \mapsto V_Q$ over the vertex sets of the two graphs. The mapping function consists of a set of Cartesian pairings between the nodes of the two graphs, i.e. $f_G^{(n)} = \{(a, \alpha); a \in V_G, \alpha \in V_Q\} \subseteq V_G \times V_Q$. Although this may appear to be a brute force method, it must be stressed that we view this process of correspondence matching as the final step in the filtering of the line-patterns. We provide more details of practical implementation in the experimental section of this paper.

With the correspondences to hand we can re-state our maximum *a posteriori* probability recognition objective as a two step process. For each graph G in turn, we locate the maximum *a posteriori* probability mapping function $f_G^{(n)}$ onto the query Q. The second step is to perform recognition by selecting the graph whose mapping function results in the largest matching probability. These two steps are succinctly captured by the following statement of the recognition condition

$$\omega_Q = \arg\max_{G' \in \mathcal{D}} \max_{f_{G'}^{(n)}} P(f_{G'}^{(n)}|G', Q)$$

The rule for updating the correspondence match between the query graph Q and the graph indexed G in the data-base is

$$f_G^{(n+1)}(a) = \arg\max_{\alpha \in V_Q} \left[\ln p(f_G^{(n+1)}(a) = \alpha | \underline{x}_a, \underline{x}_\alpha) + \sum_{\substack{(a,b) \in E_G \\ (\alpha,\beta) \in E_Q}} \left\{ \ln(1 - P_e) s_{a,\alpha}^{(n)} s_{b,\beta}^{(n)} + \ln P_e (1 - s_{a,\alpha}^{(n)} s_{b,\beta}^{(n)}) \right\} \right]$$

where the quantity $p(f_G^{(n+1)}(a) = \alpha | \underline{x}_a, \underline{x}_\alpha)$ is the *a posteriori* probability of the data graph node a matching the model graph node α. The $s_{a,\alpha}^{(n)}$ are assignment variables which are used to represent the current state of match and convey the following meaning

$$s_{a,\alpha}^{(n)} = \begin{cases} 1 & \text{if } f_G^{(n)}(a) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

This process is applied in parallel to the nodes of the graph G for a fixed number of iterations [16].

3 Histogram-based consistency

We now furnish some details of the shape retrieval task used in our experimental evaluation of the recognition method. In particular, we focus on the problem of recognising 2D line patterns in a manner which is invariant to rotation, translation and scale. The raw information available for each line segment are its orientation (angle with respect to the horizontal axis) and its length (see figure 1). To illustrate how the Euclidean invariant pairwise feature attributes are

computed, suppose that we denote the line segments associated with the nodes indexed a and b by the vectors $\underline{\mathrm{v}}_a$ and $\underline{\mathrm{v}}_b$ respectively. We use two pairwise attributes. The first these is the relative angle given by $\theta_{\underline{\mathrm{X}}_{ab},\underline{\mathrm{X}}_{cd}} = \arccos[\frac{\underline{\mathrm{X}}_{ab}\cdot\underline{\mathrm{X}}_{cd}}{|\underline{\mathrm{X}}_{ab}||\underline{\mathrm{X}}_{cd}|}]$. The second is the normalised length ratio between the oriented baseline vector $\underline{\mathrm{v}}_a$ and the vector $\underline{\mathrm{v}}\prime$ joining the end (b) of the baseline segment (ab) to the intersection of the segment pair (cd) defined as $\vartheta_{\underline{\mathrm{X}}_{ab},\underline{\mathrm{X}}_{cd}} = [\frac{1}{2} + \frac{D_{ib}}{D_{ab}}]^{-1}$. The two attributes are used as a feature-vector $\underline{\mathrm{z}}_{a,b} = (\theta_{a,b}, \vartheta_{a,b})^T$ for the line-segment pair. Each node in

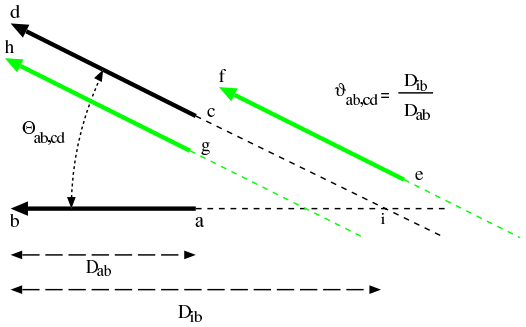

Figure 1: Geometry for shape representation

the shape graph, i.e. each line in the pattern, is represented by the histogram of its pairwise geometric attributes to the remaining lines in the pattern. This histogram can be thought of as a local estimate of the probability distribution for the pairwise attributes. Accordingly, the angle and position attributes $\theta_{a,b}$ and $\vartheta_{a,b}$ are binned in a histogram. Suppose that $S_a(\mu,\nu) = \{(a,b)|\theta_{a,b} \in A_\mu \wedge \vartheta_{a,b} \in R_\nu \wedge b \in V_D\}$ is the set of nodes whose pairwise geometric attributes with the node a are spanned by the range of directed relative angles A_μ and the relative position attribute range R_ν. The contents of the histogram bin spanning the two attribute ranges is given by $H_a(\mu,\nu) = |S_a(\mu,\nu)|$. Each histogram contains n_A relative angle bins and n_R length ratio bins. The probability of match between the normalised histograms $h_a(\mu,\nu)$ is computed using the Bhattacharyya coefficient $B_{a,\alpha}$. The relationship between the coefficient and the *a posteriori* probability is as follows

$$P(f_G^{(n)}(a) = \alpha|\underline{\mathrm{x}}_a, \underline{\mathrm{x}}_\alpha) = \exp[-B_{a,\alpha}] = \frac{\sum_{\mu=1}^{n_A}\sum_{\nu=1}^{n_R}\sqrt{h_a(\mu,\nu)h_\alpha(\mu,\nu)}}{\sum_{\alpha\in V_Q}\sum_{\mu'=1}^{n_A}\sum_{\nu'=1}^{n_R}\sqrt{h_a(\mu,\nu)h_\alpha(\mu,\nu)}}$$

4 Alternative Retrieval Algorithms

The aim in this paper is to compare the graph-based recognition algorithm with sone alternatives. The two algorithms use less complex graph representations. The most straightforward uses a global histogram of the attributes on the edges of a nearest neighbour graph. The second algorithm uses the set of attributes on the edges and realises comparison on an element-by-element basis using a robust error kernel. The two algorithms have been described in detail elsewhere [13, 17], but hitherto there has been no attempt at comparative sensitivity analysis.

4.1 Relational Histograms

The idea here is to conglomerate the node histograms into a global histogram. This histogram provides a statistical summary for the pairwise attributes residing on the edges of a nearest neighbour graph [13].

The best-matching pattern is retrieved from the data-base on the basis of similarity with the query pattern histogram. Our similarity measure is the histogram correlation. The measure of pattern correlation is the Bhattacharyya distance. The class identity of the retrieved pattern is

$$\omega_Q = arg\max_{G\in\mathcal{D}} \ln \sum_{\mu=1}^{n_A}\sum_{\nu=1}^{n_R}\sqrt{h_T^Q(\mu,\nu)\times h_T^G(\mu,\nu)}$$

4.2 Feature-sets

Here our aim is to effect retrieval on the basis of the similarity of the set of attributes residing on the edges of the nearest neighbour graph [17]. One of the most popular way of comparing a set of unordered observations whose correspondences are unknown is to use the Hausdorff distance. However, this measure is notoriously susceptible to measurement outliers. For this reason, we choose instead to gauge similarity using a robust error kernel. The class identity of the retrieved pattern is

$$\omega_Q = \arg\max_{G\in\mathcal{D}} \sum_{(i,j)\in E_G} \max_{(I,J)\in E_Q} \left(\Gamma_\sigma(||\underline{\mathrm{z}}_{I,J}^Q - \underline{\mathrm{z}}_{i,j}^G||)\right)$$

where $\Gamma_\sigma(\rho) = \exp\left(-\frac{\rho^2}{\sigma}\right)$ is a robust weighting kernel, $\underline{\mathrm{z}}_{I,J}^Q$ is a feature feature-vector from the query graph and $\underline{\mathrm{z}}_{i,j}^G$ is a feature-vector from a target graph.

5 Sensitivity Analysis

The aim in this section is to investigate the sensitivity of the three graph-based retrieval strategies to the systematics of the line-segmentation process. To this end we have simulated the segmentation errors that can occur when line-segments are extracted from realistic image data. Specifically, the different processes that we have investigated are listed below:

- **Extra lines:** Here we have added additional lines at random locations. The lengths and angles of the added lines have generated by randomly sampling distribution for the existing image-segments.

- **Missing lines:** Here we have deleted a known fraction of line-segments at random locations.

- **Split lines:** Here a predefined fraction of lines have been split into two segments. The splitting process is effected by deleting an internal fraction of each line-segment. The deleted segment is randomly positioned along the line. The fraction of the line deleted is uniformly sampled from the range $(0, 1)$.

- **Segment end-point errors:** Here we have introduced random displacements in the end-point positions for a predefined fraction of lines. The distribution of end-point errors is Gaussian. The degree of error is controlled by the variance of the Gaussian distribution.

- **Combined errors:** Here we have introduced the four different segment errors described above in equal proportion.

The performance measure used in our studies is computed as follows. We query the data-base with a sample of line patterns. For each pattern in turn we determine whether or not the correct retrieval occurs in the top-ranked position. By computing the fraction of queries that return a correctly recognised recall, we determine the avarage retrieval accuracy.

In Figure 2 we show the accuracy of retrieval for each of the different error processes in turn as a function of the fraction of added noise. In each case the sold curve is for the attribute histogram, the dashed curve is for the feature-sets and the dotted curve is for the graph matching algorithm. The plots summarise the results for an inexact query. In other words we query with a line-pattern that is similar but not identical to one of of the patterns in the data-base. For extra lines (Figure 2a), split lines (Figure 2c), end-point errors (Figure 2d) and combined errors (Figure 2e), the graph-matching algorithm is significantly better than the attribute histogram and also offers an improvement over the use of feature-sets. In the case of missing lines, however, the graph-matching algorithm only performs as well as the attribute histogram and is significantly poorer than the use of feature-sets. In other words, the method copes well with the addition of clutter (through either noise or line-fragmentation) and measurement error, but performs relatively badly when part of the line-pattern is removed.

6 Conclusions

We have presented a practical graph-matching algorithm for data-mining in large structural libraries. The main conclusion to be drawn from this study is that the combined use of structural and histogram information improves both recognition performance and recall speed. There are a number of ways in which the ideas presented in this paper can be extended. Firstly, we intend to explore more a perceptually meaningful representation of the line patterns, using grouping principal derived from Gestalt psychology. Secondly, we are exploring the possibility of formulating the filtering of line-patterns prior to graph matching using Bayes decision trees.

References

[1] M. J. Swain and D. H. Ballard, "Indexing via colour histograms," *ICCV*, pp. 390–393, 1990.

[2] G. Gimelfarb and A. K. Jain, "On retrieving textured images from an image database," *Pattern Recognition*, vol. 29, no. 9, pp. 1461–1483, 1996.

[3] C. Dorai and A. K. Jain, "COSMOS: A representation scheme for 3D free-form objects," *IEEE PAMI*, vol. 19, pp. 1115–1130, October 1997.

[4] I. Rigoutsos and R. Hummel, "A bayesian approach to model matching with geometric hashing," *CVIU*, vol. 62, p. 11, 26 1995.

[5] N. A. Thacker, P. A. Riocreux, and R. B. Yates, "Assessing the completeness properties of pairwise geometric histograms," *Image and Vision Computing*, vol. 13, pp. 423–429, June 1995.

[6] M. S. Costa and L. Shapiro, "Scene analysis using appearance-based models and relational indexing," *Int. Symp. on CV*, pp. 103–108, 1995.

[7] K. Sengupta and K. L. Boyer, "Organising large structural databases," *IEEE PAMI*, vol. 17, no. 4, pp. 321–332, 1995.

[8] H. G. Barrow and R. J. Popplestone, "Relational descriptions in picture processing," *Machine Intelligence*, vol. 5, pp. 377–396, 1971.

[9] A. Sanfeliu and K. S. Fu, "A distance measure between attributed relational graph," *IEEE SMC*, vol. 13, pp. 353–362, 1983.

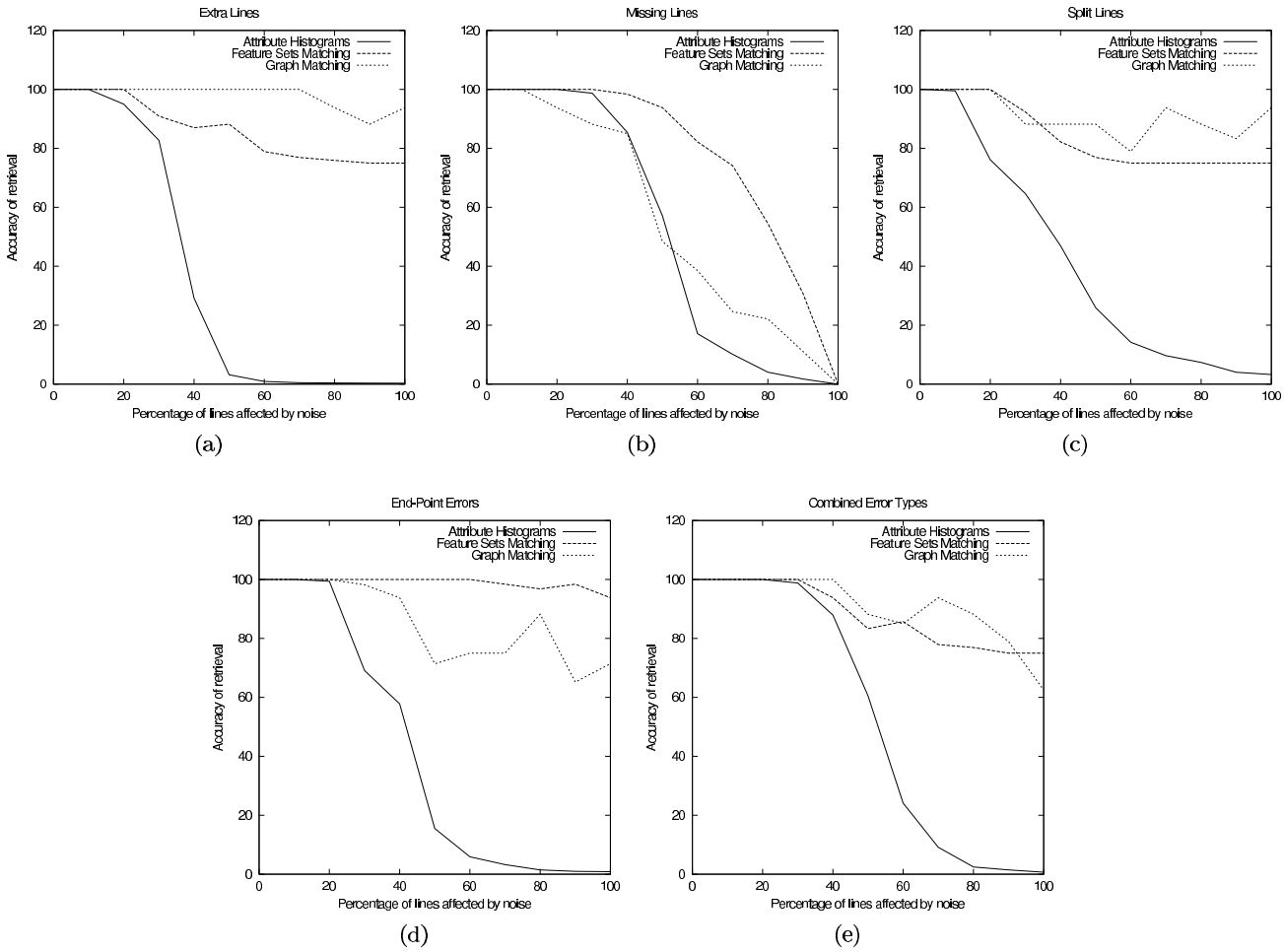

Figure 2: Effect of various kinds of noise for similarity queries.

[10] L. G. Shapiro and R. M. Haralick, "A metric for comparing relational descriptions," *IEEE PAMI*, vol. 7, no. 1, pp. 90–94, 1985.

[11] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE PAMI*, vol. 18, pp. 377–388, 1996.

[12] R. Wilson and E. R. Hancock, "Structural matching by discrete relaxation," *IEEE PAMI*, vol. 19, pp. 634–648, June 1997.

[13] B. Huet and E. R. Hancock, "Relational histograms for shape indexing," *ICCV*, pp. 563–569, Jan 1998.

[14] W. Niblack, *et al.*, "The QBIC project: Querying images by content using color, texture and shape.," *SPIE Storage and Retrieval of Image and Video Databases*, vol. 1908, pp. 173–187, February 1993.

[15] A. P. Pentland, *et al.*, "Photobook: Content-based manipulation of image databases," *Int. Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[16] B. Huet, A. D. J. Cross, and E. R. Hancock, "Graph matching for shape retrieval," *NIPS, MIT Press*, 1998.

[17] B. Huet and E. R. Hancock, "Fuzzy relational distance for large-scale object recognition," *CVPR*, pp. 138–143, June 1998.