Using Graph Search Techniques for Contextual Colour Retrieval*

Lee Gregory and Josef Kittler

Centre for Vision Speech and Signal Processing, University Of Surrey
Guildford, Surrey. United Kingdom
L.Gregory@eim.surrey.ac.uk
http://www.ee.surrey.ac.uk/Personal/L.Gregory

Abstract. We present a system for colour image retrieval which draws on higher level contextual information as well as low level colour descriptors. The system utilises matching through graph edit operations and optimal search methods. Examples are presented which show how the system can be used to label or retrieve images containing flags. The method is shown to improve on our previous research, in which probabilistic relaxation labelling was used.

1 Introduction

The increasing popularity of digital imaging technology has highlighted some important problems for the computer vision community. As the volume of the digitally archived multimedia increases, the problems associated with organising and retrieving this data become ever more acute.

Content based retrieval systems such as ImageMiner [1], Blobworld [3], VideoQ [4], QBIC [13], Photobook [14] and others [11] were conceived to attempt to alleviate the problems associated with manual annotation of databases.

In this paper we present a system for colour image retrieval which draws on higher level contextual information as well as low level colour descriptors. To demonstrate the method we provide examples of labelling and retrieval of images containing flags. Flags provide a good illustration of why contextual information may be important for colour image retrieval. Also, flags offer a challenging test environment, because often they contain structural errors due to non rigid deformation, variations in scale and rotation. Imperfect segmentation may introduce additional structural errors.

In previous work [9], the problem was addressed using probabilistic relaxation labelling techniques. The shortcomings with the previous method in the presence of many structural errors motivated the current research which is based on optimal search and graph edit operations [12]. The method still retains the invariance to scale and rotation, since only colour and colour context are used

* This work was supported by an EPSRC grant GR/L61095 and the EPSRC PhD Studentship Program.

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 186–194, 2002.

in the matching process. In addition the examples show how this method performs in the presence of structural errors and ambiguous local regions in the images/models.

Graph representations are well suited to many computer vision problems, however matching such graphs is often very computationally expensive and may even be intractable. Non optimal graph matching methods may be much less expensive than optimal search methods, but often perform badly under conditions where structural errors prevail. Such non optimal methods include probabilistic and fuzzy relaxation labelling [5], genetic search, and eigendecomposition [10] based approaches.

Other work has focused on making optimal graph search methods more suitable for database environments. Messmer and Bunke [12], presented a decomposition approach also based on A* search, which removes the linear time dependency when matching many graphs within a database. More recently the work of Berretti et al [2] formalised metric indexing within the graph matching framework.

2 Methodology

In this section we present the details of the adopted method. First, the notation for the graph matching problem is defined and the system implementation is then described in detail. Consider an attributed relational graph (ARG) $G = \{\Omega, E, \mathbf{X}\}$, where $\Omega = \{\omega_1, \omega_2, \cdots, \omega_N\}$ denotes the set of nodes. E represents the set of edges between nodes, where $E \subseteq \Omega \times \Omega$, and $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$ defines a set of attributes associated to the nodes in Ω, where x_i denotes the attributes (features) for node ω_i.

2.1 Matching

The matching problem is often formulated by defining a model graph, representing a query, which is matched to at least one scene graph. Let $G = \{\Omega, E, \mathbf{X}\}$ and $G' = \{\Omega', E', \mathbf{X}'\}$ denote the model and scene graphs respectively.

Now consider an injective function $f : \Omega \mapsto \Omega'$ which specifies mappings from the nodes Ω in the model graph G to the nodes $X \subseteq \Omega'$ contained in some subgraph of the scene G'. Such a function represents an error correcting subgraph isomorphism, since any mapped subgraph of the scene, can be isomorphic with the model graph, subject to an appropriate set of graph edit operations. The edit operations required to achieve such an isomorphism, represent the errors of an error correcting subgraph isomorphism. These errors are quantified, and are used to guide the graph search process. Error correcting subgraph isomorphism, is well suited for computer vision tasks, where noise and clutter may distort the scene graphs.

Error correcting subgraph isomorphism matches any graph to any other given graph, since an appropriate set of graph edit operations is able to transform

any graph arbitrarily. It is therefore essential to define costs for the graph edit operations. Defining such costs allows state space search methods to seek the lowest cost (best matching) mapping between any pair of graphs, given the costs for permissible edit operations. In this implementation, the following traditional graph edit operations are used. Each edit operation λ has an associated cost $C(\lambda)$.

$$\lambda : \omega_i \mapsto \omega'_j : \text{ map the model node } \omega_i \text{ to scene node } \omega'_j \quad (1)$$
$$\lambda : \omega_i \mapsto \emptyset : \text{ map the model node } \omega_i \text{ to the null attractor } \emptyset \quad (2)$$
$$\lambda : e_i \mapsto e'_j : \text{ map the model edge } e_i \text{ to the scene edge } e'_j \quad (3)$$
$$\lambda : e_i \mapsto \emptyset : \text{ map the model edge } e_i \text{ to the null attractor } \emptyset \quad (4)$$
$$\lambda : e'_j \mapsto \emptyset : \text{ map the scene edge } e_j \text{ to the null attractor } \emptyset \quad (5)$$

Note that the symbol $\emptyset$ represents a null attractor which is used to express missing edges and vertices.

Graph matching algorithms which employ state space search strategies, recursively expand partial mappings to grow error correcting subgraph isomorphisms in the state space. Our implementation uses the A* algorithm for optimal search. For any given partial mapping $f : \Omega \mapsto \Omega'$ there exists a set of graph edit operations $\Delta_f = \{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ which transform the mapped scene nodes into a subgraph isomorphism with the partial model. Hence the search through the state space can be guided by the costs of the graph edit operations required for each partial mapping.

The state space search starts from the root state which is the top node in the search tree. From this node, child nodes are generated by allowing the first model node ω_1 to be mapped to each available input node in turn $\{\omega'_1, \omega'_2, \cdots, \omega'_N, \emptyset\}$. Also a child state for a missing vertex is added by mapping the model node to the null attractor.

Each leaf of the tree now represents an error correcting subgraph isomorphism $f_k : \Omega \mapsto \Omega'$ from a partial model graph to the scene graph. The cost of these graph mappings are computed as $C(\Delta_{f_k})$, and the leaf with the lowest cost is expanded. This process continues until the model is fully mapped and the isomorphism with the least cost is found. For the sake of efficiency, the graph edit distance for a given leaf node in the search tree, is computed incrementally from its parent node.

The complexity of the described state space search, is in the worst case exponential, although in practice the actual complexity is data dependent and the optimal search often becomes tractable. To further prune the search space and reduce the complexity, lookahead terms are often used when computing the costs for a given state. The lookahead term, computes an estimate of the future cost of any proceeding mappings based on the current partial interpretation. The exact computation of a minimal future mapping is itself an error correcting subgraph isomorphism problem, and therefore has a worst case exponential complexity.

Hence an estimate is used instead. To prevent false dismissals, such an estimate must provide a lower bound on future cost for any proceeding mappings. To provide such a lower bound for future mapping cost, we consider each unmapped node independently, therefore breaking the exponential complexity of the lookahead. Tests show that a lower bound which ignores edge constraints is faster than a more refined lookahead scheme which considers the edge costs. The lookahead function $L(f : \Omega \mapsto \Omega')$ is defined as

$$L(f : \Omega \mapsto \Omega') = \sum_{\omega_i \in \emptyset_M} \min_{\omega_j \in \emptyset_I} (C(\lambda : \omega_i \mapsto \omega'_j)) \tag{6}$$

where $\emptyset_M$ and $\emptyset_I$ denotes the set of model and input nodes which are not mapped in the current partial interpretation. This result is in agreement with Berretti et al[2] where a faster less accurate lookahead was shown to outperform a more complex scheme. This does not affect the optimality, since any lower bound estimate will not allow false dismissals.

2.2 Pre-processing

We now explain how the images are initialised for the graph matching. During the pre-processing stage, images are segmented so that a region adjacency graph can be built. Each pixel in the image is represented as a 5D vector, the first three dimensions are the RGB colour values for the pixel and the last two dimensions are the pixel co-ordinates. The feature space is then clustered using the mean shift algorithm [6][7]. The mean shift algorithm is an iterative procedure which seeks the modes of the feature distribution. The algorithm is non-parametric and does not assume any prior information about the underlying distributions, or the number of clusters. This is an important implication because it allows the algorithm to operate unsupervised. In practice only a window size and co-ordinate scale are needed by the algorithm. Every pixel is given a label corresponding to the cluster which it has been classified to. The region labels correspond to homogeneous colour regions within the image. A connected component analysis stage ensures that only connected pixels may be assigned the same label.

The segmented image can now be expressed as an ARG. The attributes X are defined as follows.

$$x_{i,1} = n_i \tag{7}$$

$$x_{i,2} = \bar{\mathbf{R}}_i = \frac{1}{n_i} \sum_{p \in P_i} R_p \tag{8}$$

$$x_{i,3} = \bar{\mathbf{G}}_i = \frac{1}{n_i} \sum_{p \in P_i} G_p \tag{9}$$

$$x_{i,4} = \bar{\mathbf{B}}_i = \frac{1}{n_i} \sum_{p \in P_i} B_p \tag{10}$$

where n_i is the number of pixels within region (node) ω_i and P_i denotes the set of pixels in region ω_i. R_p, G_p, B_p denote the red, green and blue pixel values respectively for pixel p. The segmentation is further improved by merging adjacent nodes which have a small number of pixels, or 'similar' feature space representation. Consider a node ω_i which has a set of neighbouring nodes N_i. The best possible candidate $\omega_{j_{best}}$, for merging with node ω_i, is given by the following equation:-

$$j_{best} = \arg \min_{j \in N_i} \left\{ \sqrt{\left(\bar{\mathbf{R}}_i - \bar{\mathbf{R}}_j\right)^2 + \left(\bar{\mathbf{G}}_i - \bar{\mathbf{G}}_j\right)^2 + \left(\bar{\mathbf{B}}_i - \bar{\mathbf{B}}_j\right)^2} \right\} \tag{11}$$

Node ω_i is only merged with node $\omega_{j_{best}}$ if the following criterion is satisfied:-

$$\sqrt{\left(\bar{\mathbf{R}}_i - \bar{\mathbf{R}}_{j_{best}}\right)^2 + \left(\bar{\mathbf{G}}_i - \bar{\mathbf{G}}_{j_{best}}\right)^2 + \left(\bar{\mathbf{B}}_i - \bar{\mathbf{B}}_{j_{best}}\right)^2} \leq \tau_c \tag{12}$$

where τ_c is some pre-specified threshold which controls the degree of merging for similarly coloured homogeneous regions. In a second merging stage, each node ω_i is merged with node $\omega_{j_{best}}$ if the following criterion is satisfied:-

$$\tau_s \geq \frac{n_i}{\sum\limits_{\forall j} n_j} \tag{13}$$

In practice τ_s controls how large, relative to the size of the image, the smallest region is allowed to be. It is expressed as a fraction (typically 1%) of the total number of image pixels. The resulting graph provides an efficient representation for the images within the system.

2.3 Contextual Colour Retrieval

In order to match a model image with a set of given scene images, the attributed graphs are created from the segmented images generated by the pre-processing. Edges in the attributed graph correspond to adjacent regions within the image. In contrast to the pre-processing stage, the double hexicone HLS colour space [8] is used for attribute measurements. The attributes of a vertex ω_j, are: mean hue H_j, mean lightness L_j and mean saturation S_j.

The conical bounds of the space limit the saturation according to lightness. This is intuitively better than some other polar colour spaces, which allow extremes in lightness (black and white) to be as saturated as pure hues, which is obviously not a desired trait. We define a colour distance measure $d_{i,j}$ between two vertices's ω_i and ω_j as :-

$$d_{ij} = \begin{cases} \Delta H_{i,j} & : \quad s_i > \tau_{sat}, s_j > \tau_{sat} \\ \sqrt{\frac{1}{4}\Delta_x^2 + \frac{1}{4}\Delta_y^2 + \Delta L_{i,j}^2} & : \quad \text{otherwise} \end{cases} \tag{14}$$

where

$$\Delta_x = S_i \cos(H_i) - S_j \cos(H_j) \tag{15}$$
$$\Delta_y = S_i \sin(H_i) - S_j \sin(H_j) \tag{16}$$
$$\Delta L_{i,j} = L_i - L_j \tag{17}$$
$$\Delta H_{i,j} = H_i - H_j \tag{18}$$

where τ_{sat} is a threshold which determines a boundary between chromatic and achromatic colours. Colour comparisons are often hindered by varying illumination and intensity. For this reason the difference in hue $\Delta H_{i,j}$ is chosen as the measurement criterion for chromatic colours. However, difference in hue is not an appropriate measurement for achromatic colours since hue is meaningless for colours with low saturation. In these cases, the more conventional euclidean distance type measurements are used.

The colour measurement defined above forms the basis of the vertex assignment graph edit operation. The assignment of a model vertex to the null attractor is defined to have a constant cost, as is the assignment of model or scene edges to the null attractor. In this implementation, edges are not attributed and therefore edge substitutions have zero cost (since all edges have the same attributes). More formally:

$$C(\lambda : \omega_i \mapsto \omega'_j) = 1 - N_\sigma(d_{i,j}) \tag{19}$$
$$C(\lambda : \omega_i \mapsto \emptyset) = \zeta_m \tag{20}$$
$$C(\lambda : e_i \mapsto e'_j) = 0 \tag{21}$$
$$C(\lambda : e_i \mapsto \emptyset) = \eta_m \tag{22}$$
$$C(\lambda : e_j \mapsto \emptyset) = \eta_i \tag{23}$$

where ζ_m is the cost for a missing node (0.5 typical), η_m is the cost for a missing edge (0.5 typical) and η_i is the cost for an inserted edge (0.1 typical). $N_\sigma()$ represents a Gaussian probability distribution

$$N_\sigma(x) = e^{-(\frac{1}{2})(\frac{x}{\sigma})^2} \tag{24}$$

where sigma has a typical value of 0.5. The shape of the assumed distribution does affect the efficiency of the search process. The distribution helps to discriminate between well and poorly matched attributes better. This allows the graph matching algorithm to expand deeper into the search tree before backtracking is necessary.

3 Experimental Results

The experimental results contained within this section were obtained using a C++ implementation running on an Athlon 1400XP with 512 Mb of RAM.

Fig. 1. Examples of synthetic models

Synthetic flags shown in figure 1 were considered as models. In contrast to the relaxation labelling approach in the previous work [9], the method is able to correctly self-label any of the given synthetic models, although in some cases (UK model) the symmetric regions were labelled arbitrarily. This is expected since the optimal search should always find the zero cost solution. Examples of such synthetic models are shown below.

Examples are presented which show how the system is able to label real images from synthetic models. The images in figure 2 show the interpretations for models of the Canadian and German flags, being matched to real images containing targets for these models. The first image in each example shows the target(scene) image. The second image shows the segmentation and hence graph structure, and the third image shows the interpretation results when matched to the corresponding synthetic model.

Note in each of these images the labelling is in complete agreement with ground truth data by manual annotation. The other example in figure 2 shows how the system is able to label quite complex model and scene images. The example of the USA flag shows how the system again is able to label with 100% accuracy the complex USA image in the presence of over segmentation errors.

In the previous example the model graph contains 15 nodes, and the scene graph contains 76 nodes. Even with this complexity, the system completed the match in 6.4 seconds of CPU time. The simple examples shown in figure 2 were typically matched in 0.8 seconds of CPU time including feature extraction and graph creation.

Retrieval performance can be evaluated by matching a given model to every scene image in the database. For each match, the sum of the graph edit operation costs is used as a similarity measure, from which the database can be ordered. A diverse database containing approximately 4000 images from mixed sources (including Internet, television and landscapes) was used as the experimental testbed. Ground truth data was created by manually identifying a set of target images $\mathbf{T}_m$ for a given synthetic model m.

In order to calculate the system performance Q_m, for a given model m, effective rank is introduced. Effective rank $\mathbf{R}(I_{j(i)})$ for a target image $I_{j(i)} \in \mathbf{T}_m$, is defined as the ranking of the target image $I_{j(i)}$ relative only to images which are not themselves target images ($I_k \notin \mathbf{T}_m$). This scheme is intuitive since the rank of a target should not be penalised by other targets with higher rank. The Effective rank is only penalised by false retrievals which have a higher database

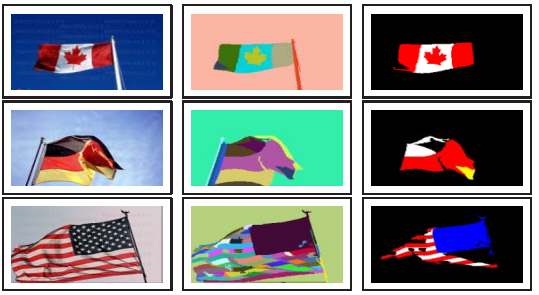

Fig. 2. Labelling Examples

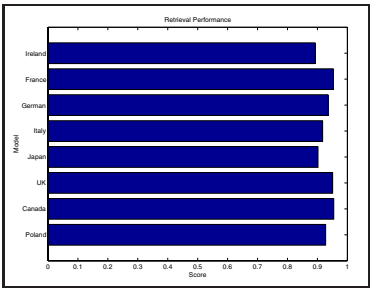

Fig. 3. Retrieval Performance

rank than the target image. Based upon the effective rank $\mathbf{R}(I_{j(i)})$, a model score $0 \leq Q_m \leq 1$ is defined as

$$Q_m = \frac{q_m}{q_{max}} \tag{25}$$

$$q_m = \sum_{I_{j(i)} \in \mathbf{T}_m} (N - \mathbf{R}(I_{j(i)}) + 1) \tag{26}$$

$$q_{max} = \sum_{i=1}^{N_{T_m}} (N - i + 1) \tag{27}$$

where N is the number of total images in the database and N_{T_m} is the total number of target images for model m. This performance evaluation criterion would yield a score of unity if all target images were ranked at the top of the database.

The system has performed well for each synthetic model. On all synthetic models, the average effective rank for the corresponding target images was always within the top 10% (approximately) of the image database.

4 Conclusion

We have presented a system for contextual colour retrieval based on graph edit operations and optimal graph search. Examples have demonstrated the performance of this system when applied to image labelling and image retrieval. Since the system uses only colour and adjacency information, it remains invariant to scale and rotation.

The results show that the adopted methods performs well in both labelling and retrieval domains. The method clearly outperforms our previous work [9]. The method is still exponential in the worst case, however the results show that for small models, the problem is quite tractable.

Future work on this system may include the incorporation of other measurements into the graph matching framework. This should improve the accuracy of

labellings and the precision of retrieval. More measurement information would also push back the computational boundary since the search process would be better informed.

References

1. P. Alshuth, T. Hermes, L. Voigt, and O. Herzog. On video retrieval: content analysis by imageminer. In *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering*, volume 3312, pages 236–47, 1997. 186
2. S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, October 2001. 187, 189
3. C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries (Cat. No.97TB100175). IEEE Comput.Soc.*, pages 42–9, June 1997. 186
4. S-F. Chang, W. Chen, HJ. Meng, H. Sundaram, and D. Zhong. Videoq: an automated content-based video search system using visual cues. In *Proceedings ACM Multimedia 97. ACM.*, pages 313–24, USA, 1997. 186
5. W. J. Christmas, J. V. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:749–764, 8 1995. 187
6. Dorin Comaniciu and Peter Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, San Juan, Puerto Rico, June 1997. 189
7. Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *IEEE Int'l Conf. Computer Vision (ICCV'99)*, pages 1197–1203, Greece, 1999. 189
8. James Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics.* Addison Wesley Longman Publishing Co, 2nd edition, 1995. 190
9. L. Gregory and J. Kittler. Using contextual information for image retrieval. In *11th International Conference on Image Analysis and Processing ICIAP01*, pages 230–235, Palermo, Italy, September 2001. 186, 192, 193
10. B. Lou and E. Hancock. A robust eigendecomposition framework for inexact graph-matching. In *11th International Conference on Image Analysis and Processing ICIAP01*, pages 465–470, Palermo, Italy, September 2001. 187
11. K Messer and J Kittler. A region-based image database system using colour and texture. *Pattern Recognition Letters*, pages 1323–1330, November 1999. 186
12. B. Messmer and H. Bunke. A new algorithm for error tolerant subgraph isomorphism detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 493–504,, May 1998. 186, 187
13. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: querying images by content using color, texture, and shape. In *Proceedings of Spie - the International Society for Optical Engineering*, volume 1908, pages 173–87, Feb 1993. 186
14. A. Pentland, RW. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–54, June 1996. 186