Colour Adjacency Histograms for Image Matching

Allan Hanbury*[1] and Beatriz Marcotegui[2]

[1] PRIP, Institute of Computer-Aided Automation, Vienna University of Technology, Favoritenstraße 9/1832, A-1040 Vienna, Austria. hanbury@prip.tuwien.ac.at
[2] Centre de Morphologie Mathématique, Ecole des Mines de Paris, 35, rue Saint-Honoré, 77305 Fontainebleau, France. marcoteg@cmm.ensmp.fr

Abstract. The use of 2D colour adjacency histograms for image matching in image retrieval scenarios is investigated. We present an algorithm for extracting representative colours from an image and a new method for matching 1D colour histograms and 2D colour adjacency histograms obtained from images quantised using different colour palettes. An experimental evaluation of the matching performance is done.

1 Introduction

The comparison of image colour distributions is an important tool in Content-Based Image Retrieval. It is often done by comparing colour histograms of images [1], which eliminates information on the spatial distribution of colours.

In this paper, we investigate characterising images for image retrieval by a 2-dimensional histogram which includes information on colour adjacency. Such information in the form of a colour adjacency graph has been used successfully for object recognition [2]. The work closest to the approach presented in this paper is that of Lee et al. [3]. They first quantise the hue into seven hue components. After each pixel in an image has been assigned to one of these components, a matrix summarises the adjacency of the hue components at the pixel level.

We propose to use a similar colour adjacency histogram, but having the following differences: the histogram uses a palette of colours determined for each image, and the histogram does not summarise the information at the pixel level, but at the region level. These regions are created by a combined colour quantisation and segmentation algorithm.

The contributions of this paper are: a new algorithm for extracting representative colours from an image using hierarchical clustering and segmenting the image based on the results of the clustering, the construction of 2D colour adjacency histograms based on the colour quantisation and segmentation and a new method for matching the 1D and 2D histograms obtained from images quantised using different colour palettes.

* This work was supported by the European Union Network of Excellence MUSCLE (FP6-507752), and the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04). The authors wish to thank Fernand Meyer for helpful discussions.

The paper is structured as follows. Section 2 discusses our combined colour quantisation and segmentation approach. The histograms used in image matching are described in Section 3. The results of image retrieval experiments are presented in Section 4. Section 5 concludes.

2 Colour quantisation

There exist many algorithms for reducing or quantising the number of colours in an image, including many based on some form of clustering in colour space [4]. Our colour quantisation is done in two steps: (1) pre-segmentation using the watershed with volume extinction values and (2) a hierarchical clustering, fusing the closest colours of the pre-segmented regions in the CIELAB space. It results in a segmentation having large, regular regions. Deng et al. [5] also use colour clustering to obtain the dominant colours in an image. They segment an image, apply vector quantisation in each region and then cluster the resulting colours from all the regions by agglomerative clustering such that the minimum distance between two centroids exceeds a preset threshold. To obtain a segmentation, each pixel is assigned to the centroid closest to it in colour space. The resulting regions will however be more fragmented and less regular than those obtained using the method presented in this paper. While this presents no problem for classic one-dimensional histogram calculation, the small regions and irregular boundaries make the result unsuitable for determing colour adjacency and the boundary length between adjacent colours. The steps of the colour quantisation and segmentation algorithm are presented here and discussed further in the subsections referred to:

1. Pre-segmentation using the watershed with volume extinction values (Section 2.1). The result is a *mosaic image*, in which each region is replaced by the mean of the RGB values it contains.
2. Conversion of the mosaic image into the CIELAB space.
3. Extraction of one set of CIELAB coordinates for each region.
4. Hierarchical clustering of the extracted colour coordinates to obtain the required number of colour clusters (Section 2.2).

2.1 Pre-segmentation

The *watershed using volume extinction values* [6] creates a hierarchy based on the fusion of lakes in watershed catchment basins. During the flooding process, a record is kept of the merging in the form of a graph, where each node represents a lake and each edge represents the merging of two adjacent lakes [6]. The weight on each edge is the volume of the smallest lake involved in the merging when two adjacent lakes merge. It can be shown that the resulting graph is a minimal spanning tree (MST). To obtain a segmentation with k regions, one simply cuts the $(k-1)$ highest valued edges of the MST. The only parameter of this segmentation algorithm is the number of regions required.

The flooding process is carried out on the gradient of the colour image. We use the gradient found to give the best results in a morphological waterfall segmentation in [7]. This is the *saturation weighing-based colour gradient* applied in the L1 norm 3D polar coordinate colour space [8]. This gradient gives a larger weight to the differences in hue when the saturation is high, and a larger weight to differences in luminance when the saturation is low.

In order to simplify the image before segmenting it, thereby eliminating small regions, we make use of the morphological *leveling* [9]. The filter used to produce the marker for the leveling operator is the morphological alternating sequential filter [10], where the size of the filter refers to the number of subsequent opening and closing operations. In order to apply these filters to colour images, we apply the filter separately to each colour component.

2.2 Hierarchical Clustering

A hierarchical clustering method is a procedure for transforming a proximity matrix into a sequence of nested partitions [11]. We use an agglomerative clustering method working on the Euclidean distances between the coordinates of the points in the CIELAB space. At the start, the colour of each region in the mosaic image is taken to be a separate cluster in CIELAB space. The two clusters which have the smallest distance between them are fused into a single cluster, and this process is iterated until a single cluster remains. Distances between clusters are measured as the average Euclidean distance between all pairs of points in cluster r and cluster s (average linkage). The result of this clustering is a tree representing the order in which clusters have been fused with one another. The leaves of the tree are the points corresponding to the original CIELAB coordinates. The distance between two clusters which are fused is also stored.

Once the tree has been built, a specified number of clusters k in the CIELAB space is obtained from the corresponding level of the tree. The representative colour of each of the k clusters is then calculated. The representative colour $\mathbf{c}_i$ of cluster i is the mean of all the colours belonging to cluster i. One then has a palette of k colours $F = \{\mathbf{c}_i, i = 1, \ldots, k\}$ for an image. Each region in the mosaic image is then given a label indicating the colour cluster to which it belongs. This results in an image labelled by colour cluster membership. The segmentation of the image in Figure 1(a) into 16 colours is shown in Figure 1(b).

3 1D and 2D Histograms

We discuss the construction and similarity measures for 1D and 2D histograms calculated from the segmentation results.

3.1 Histogram construction

Two types of histogram are tested, the standard 1D colour histogram and the proposed 2D colour adjacency histogram. The 1D colour histogram of an image

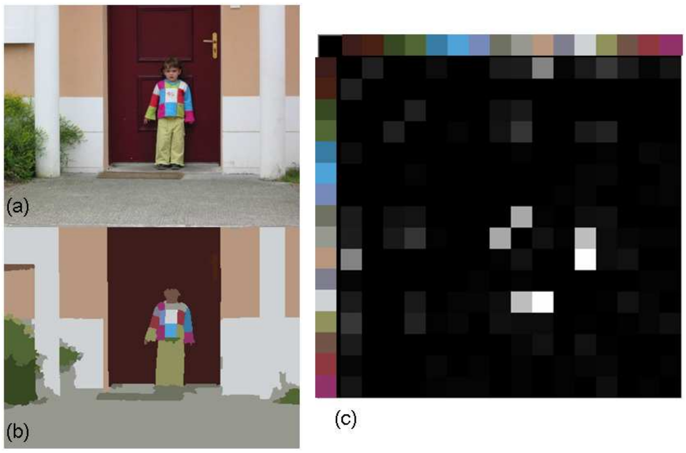

Fig. 1. (a) Original image. (b) Segmentation into 16 colours. (c) 2D histogram. The colours corresponding to each row and column are shown to the left and above. A higher value in a histogram bin is indicated by a lighter grey.

with colour palette F containing k colours is $P = \{p_i, i = 1, \ldots, k\}$, where p_i is percentage of the image area occupied by colour $\mathbf{c}_i$.

The 2D colour adjacency histogram $H(i, j)$ of an image containing k colours is a $k \times k$ matrix. The row index i and column index j ($i, j \in 1, 2, \ldots, k$) index the colours in the colour palette. The matrix entry $H(i, j)$, where $i \neq j$, contains the total length of the common boundary between all regions having colour indices i and j. Using a count of the number of times that regions occur next to each other proved to be less effective. This histogram is obviously symmetric, as the adjacency relation is symmetric. The entries on the diagonal $H(i, i) = 0$, as two adjacent regions with the same colour index cannot be distinguished. An example of such a histogram is shown in Figure 1(c).

3.2 Histogram similarity

Once we have constructed the histograms for all images in a database, we wish to match the images based on the similarity of their histograms. To do this, we require a similarity measure between two histograms. We describe two approaches here, the first is our proposed approach based on colour matching and reorganisation of one of the histograms based on the results of the colour matching. The second is the quadratic colour histogram distance [5].

Similarity by Colour Matching and Histogram Reorganisation This approach requires that each image in the database is quantised so as to have the same number of colours — we use $k = 16$. As each of the images in the database has a different colour palette, the first task is to match the closest colours in two colour palettes, where each palette contains k sets of CIELAB coordinates.

We represent the palettes by $F_1 = \{\mathbf{c}_i, i = 1, \ldots, k\}$ and $F_2 = \{\mathbf{b}_j, j = 1, \ldots, k\}$, where F_1 is assumed to be the palette of the query image, and F_2 the palette of an image to be compared to it. The histograms corresponding to these images and their palettes are 1D histograms P_1 and P_2, and 2D histograms H_1 and H_2, corresponding to palettes F_1 and F_2 respectively. The aim is to match each of the colours in F_2 to a colour in F_1. We begin by calculating a $k \times k$ distance matrix $D(i, j)$, where the value in row i and column j contains the Euclidean distance between colour coordinates $\mathbf{c}_i$ and $\mathbf{b}_j$. Two types of colour matching are tested, 1-to-1 matching and 1-to-many matching

In *1-to-1 matching*, each colour in F_1 may only be matched to a single colour in F_2. Colours are matched in the order of increasing Euclidean distance to create a list of matching colour indices. Each index to a colour in F_1 and each index to a colour in F_2 appears only once in the list. The first items in the list represent the better colour matches (lower Euclidean distance). An alternative to this greedy algorithm could be a global optimisation algorithm such as one of the algorithms for bipartite graph matching. However, such algorithms attempt to minimise the sum of the distances between the matched colours. This is however often better done by making a large number of "average matches" instead of a few extremely bad matches. This is not optimal, because if an image contains a completely different colour a bad match for this colour is better than average matches for all of them. In *1-to-many matching*, each colour in F_1 may be matched to more than one colour in F_2. In the resulting list of matches, indices to colours in F_1 may occur more than once.

After the colours have been matched, the entries in the 1D histogram P_2 or the rows and columns in the 2D histogram H_2 are reordered based on the colour match list so as to correspond to P_1 or H_1 as closely as possible. These reordered histograms are labelled P_{2r} and H_{2r}.

To calculate histogram similarity, the histograms P_1 and P_{2r} in the 1D case, or H_2 and H_{2r} in the 2D case, are compared using histogram intersection modified to use the distances between the matched colours as weights to reduce the effect of poorly matched colours on the similarity calculation. We begin by normalising the histograms (both 1D and 2D) so that the sum of their elements is 1. For weighting, the function D_ℓ returns the distance between the colour $\mathbf{c}_\ell$ in palette F_1 and the colour in palette F_2 that was matched to it (in practice, this can be read from the colour match list). For 1D histograms, the similarity $S\left(P_1, P_{2r}\right)$ is calculated as

$$\sum_{\ell=1}^{k} \min\left[P_1\left(\ell\right), P_{2r}\left(\ell\right)\right] w\left(D_\ell\right) \tag{1}$$

where the weight w is a colour similarity measure (between 0 and 1, with 1 indicating identical colours) based on those in [5, 12]:

$$w\left(d\right) = \begin{cases} 1 - \frac{d}{d_{\max}} & \text{if } d \leq T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where T is a threshold on colour similarity, which we set to $d_{\max}$. In [12], $d_{\max}$ is set to the largest value in the distance matrix, thereby eliminating the need for a threshold. We set $d_{\max}$ to a constant value of 50, half the distance between black and white in the CIELAB space. The threshold $T = d_{\max}$ then simply avoids that the colour similarity value becomes negative.

For 2D histograms, the similarity $S(H_1, H_{2r})$ is

$$2 \sum_{\ell=1}^{k} \sum_{n=\ell+1}^{k} \min\left[H_1(\ell, n), H_{2r}(\ell, n)\right] w(D_\ell)\, w(D_n) \tag{3}$$

As we are comparing neighbouring colours, two weights are present, indicating how well each of the colours considered is matched. The sum is taken over only half of the histogram as it is symmetric.

Quadratic Colour Histogram Distance The quadratic 1D colour histogram distance used is presented by Deng et al. [5]. We do the colour quantisation as described in Section 2. This technique can compare images having a different number of colours in their palettes. However it requires that the distance between the closest pair of quantised colours for an image exceeds a preset threshold T_d. Therefore, instead of choosing a preset number of colours, we cut the tree resulting from the hierarchical clustering using a distance threshold of T_d. Given two images having a possibly different number of quantised colours $F_1 = \{\mathbf{c}_i, i = 1, \ldots, k_1\}$ and $F_2 = \{\mathbf{b}_j, j = 1, \ldots, k_2\}$, with corresponding histograms $P_1 = \{p_i, i = 1, \ldots, k_1\}$ and $P_2 = \{q_j, j = 1, \ldots, k_2\}$, the distance between the histograms is given by

$$D^2(P_1, P_2) = \sum_{i=1}^{k_1} p_i^2 + \sum_{j=1}^{k_2} q_j^2 - \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} 2 a_{ij} p_i q_j \tag{4}$$

where a_{ij} is the similarity coefficient between colours $\mathbf{c}_i$ and $\mathbf{b}_j$ given by $w(d_{ij})$ in Equation 2, where d_{ij} is the Euclidean distance between the colours and the threshold T in Equation 2 is set to T_d. The value of $d_{\max}$ is taken to be αT_d, where we take $\alpha = 1.2$ as done in [5].

4 Experiments

We perform retrieval experiments on a dataset of 108 personal photographs taken in four locations: a sofa, the area in front of a house, the forest and the beach. In many sets of personal photos, some common locations occur very often and should be characterised by specific colour adjacencies. As it would be useful to find all the photos taken in a specific location, we evaluate the capability of the algorithms to retrieve images of the same location as the query image.

To evaluate the matches, we use the R-precision measure, which is the precision at R, where R is the number of relevant images in the dataset. The value

	R	1-to-1_2D	1-to-many_2D	1-to-1_1D	quaddist_1D
Overall		74.2	74.2	76.5	73.9
Sofa	32	66.9	74.0	70.3	68.9
Forest	5	66.7	75.0	69.4	61.1
Beach	9	59.0	54.0	87.0	59.0
Residence	58	81.7	77.7	78.9	80.5

Table 1. The per class and overall R-precision values for retrieval using 2D histograms with 1-to-1 and 1-to-many matching (columns 3 and 4), and 1D histograms with 1-to-1 matching and the quadratic colour histogram distance (columns 5 and 6).

of R for each class is shown in the second column of Table 1. We tested both 1D and 2D histograms, using 1-to-1 and 1-to-many matching for the 2D histograms, and 1-to-1 matching and the quadratic colour histogram distance for the 1D histograms. The R-precision values for each of these methods are shown in Table 1, where the overall R-precision and the R-precision per class are shown.

The overall results obtained by all the methods tested are similar. However it can be seen that for different query images, different matching methods have the best retrieval results. This is particularly evident for beach class, where the R-precision for the 1D histogram with 1-to-1 matching is notably higher than for the other methods. For the sofa and forest classes, the 2D histograms with 1-to-many matching have the highest R-precision, while for the residence class there is little variation between the methods. These differences can also be seen for two individual queries in Figure 2. For the query image from the sofa class, the top four retrieved images for the 2D histograms are correct, whereas the 1D histogram results both include an image from the residence class. This situation is reversed for the beach class, where the 1D histogram with 1-to-1 matching retrieves four correct images, while all other methods include images from the residence class in the top four. Examining the 2D histograms more closely reveals why the girl in red was retrieved as a good match to the beach. By chance a few of the adjacent colours having a long common boundary are matched resulting in 2D histograms having high peaks at the same place. The 2D histograms do not consider the percentage of each colour present in the image. For the very similar beach images, this happens to be the best comparison measure.

5 Conclusion

We have investigated the use of 2D colour adjacency histograms for matching images in image retrieval scenarios. An interesting outcome is the class-dependent retrieval performance of the 1D and 2D histograms. However, for each class, at least one of the proposed methods outperforms the quadratic colour histogram distance. Further research will involve investigating the conditions under which 1D or 2D histograms perform better and hence designing an efficient combination of the 1D and 2D distance measures.

(a) 1-to-1_2D

(b) 1-to-many_2D

(c) 1-to-1_1D

(d) quaddist_1D

(e) 1-to-1_2D

(f) 1-to-many_2D

(g) 1-to-1_1D

(h) quaddist_1D

Fig. 2. The first four images retrieved for a query image (shown left) in the (a)–(d) sofa and (e)–(h) beach class. The matching methods used are given below the images.

References

1. Schettini, R., Ciocca, G., Zuffi, S.: A survey of methods for colour image indexing and retrieval in image databases. In Luo, R., MacDonald, L., eds.: Color Imaging Science: Exploiting Digital Media. John Wiley (2001)
2. Matas, J., Marik, R., Kittler, J.: The color adjacency graph representation of multi-coloured objects. Technical Report VSSP-TR-1/95, University of Surrey (1995)
3. Lee, H.Y., Lee, H.K., Ha, Y.H.: Spatial color descriptor for image retrieval and video segmentation. IEEE Trans. on Multimedia **5**(3) (2003) 358–367
4. Scheunders, P.: A comparison of clustering algorithms applied to colour image quantization. Pattern Recognition Letters **18** (1997) 1379–1384
5. Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., Shin, H.: An efficient color representation for image retrieval. IEEE Trans. on Image Processing **10**(1) (2001) 140–147
6. Meyer, F.: An overview of morphological segmentation. International Journal of Pattern Recognition and Articial Intelligence **15**(7) (2001) 1089–1118
7. Angulo, J., Serra, J.: Color segmentation by ordered mergings. In: Proc. of the Int. Conf. on Image Processing. Volume II. (2003) 125–128
8. Hanbury, A., Serra, J.: Colour image analysis in 3D-polar coordinates. In: Proc. of the DAGM'03 conference. (2003) 124–131
9. Meyer, F.: Levelings, image simplification filters for segmentation. Journal of Mathematical Imaging and Vision **20** (2004) 59–72
10. Soille, P.: Morphological Image Analysis. second edn. Springer (2002)
11. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
12. Hafner, J., Sawhney, H.S., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. IEEE Trans. on Pattern Analysis and Machine Intelligence **17**(7) (1995) 729–736