Spatial Colour Matching for Content Based Retrieval and Navigation

David Dupplaw, Paul Lewis, Mark Dobie
Multimedia Research Group, University of Southampton,
Southampton, Hampshire, England

Abstract

In this paper we describe two approaches for providing a versatile mechanism for retrieval and navigation using spatially related feature selections. As an example, we build colour histograms from images, and use a quadratic histogram match to achieve a retrieval method based on the spatial colour distribution of the images. Navigation is facilitated by a larger architecture, called MAVIS (Multimedia Architecture for Video, Image and Sound), which is an on-going research project to provide an integrated and networked approach to multimedia content based retrieval and navigation. Demonstrations are provided of the two presented retrieval methods in action.

1 Introduction

With a trend towards digital archiving of textual and visual documents, the management of large databases can be very difficult without an acceptable way of indexing these, in particular to simplify retrieval of relevent information. For many years our group have been working on the development of an innovative multimedia authoring tool and open-hypermedia system, Microcosm [2], which has now been commercialised [1]. The MAVIS (Multimedia Architecture for Video, Image and Sound) [10, 9] project is a programme of research to extend the open hypermedia idea to non-text media. Moving to a portable Java based implementation, MAVIS-2 introduces the use of a multimedia thesaurus [11], intelligent agents and the ability to distribute processes, and perform matches in parallel, by utilising HTTP messaging. The Multimedia Thesaurus (MMT) [11] consists of a network of traditional thesaurus relations between text representations of concepts, augmented by representations of concepts from non-text media. This can extend the range of a query using synonym substitution and enhance cross-media navigation through the concept network.

Each media type supported by MAVIS (images, text, video, etc.) may have multiple feature extraction methods, for different features of the media, and each feature extraction technique is contained in a module, which is responsible for creation and matching of signatures. By preprocessing selections into signatures, and storing and indexing the signatures in a database, time can be saved during the matching process.

This paper describes how the spatial colour distribution module uses a variety of cues in the image to create and match signatures facilitating retrieval of images based on their spatial colour distribution.

2 Related Work

To humans, the colour of an object is the most easily identifiable property - in comparison to texture and shape. In the computer representation is is also easily accessible. It is therefore not suprising that there are many colour-based image retrieval systems in development.

IBM's QBIC System [3, 12, 13] is probably the most well known content-based retrieval engine, and uses colour, shape, and texture. The use of large areas of solid colour, drawn as rectangles in a paint-package, are transposed to a grid and used as a template for matching against images in a database. Queries based on histograms from other images is not supported.

[1] Microcosm is a commercial product of Multicosm, Ltd. See *http://www.multicosm.com*

PicToSeek [6] is a content-based image search system, designed for use on The Web by the Intelligent Sensory Information Systems research group, at the University of Amsterdam. The system uses a colour model that is colour constant - that is, it is independent of the illumination colour, shadows, and shading cues. PicToSeek is, however, only concerned with the whole image histograms, and does not allow spatially oriented queries.

Virage [7] is a system produced by Virage Inc. that performs content-based navigation on video and images, using colour, texture, composition (colour intensity distribution), and structure (shape layout).

VisualSEEk [18] is a content based search engine designed at the Center for Image Technology for New Media, at Columbia University, New York. The system uses colour set back-projection to extract regions of colour from images. Colour back-projection is a way of automatically extracting salient regions, by quantizing the image based on the 'colour sets' - which are thresholded histograms. Because colour sets are binary, the histogram matching functions can be reduced which allows efficient indexing. VisualSEEK allows spatial-colour retrieval based on a query built from areas of solid colour, and semantic relations between those areas.

Color-WISE [14] is an image similarity retrieval system which allows users to search for stored images using matching based on the localized dominant hue and saturation values. It uses a cunning fixed segmentation of overlapping elements to ensure that the matching is slightly fuzzy. The system computes seperate histograms for hue, saturation, and intensity, and reduces their size by finding their area-peak - basically removing noise that is small amounts of isolated colours. Color-WISE uses Microsoft Access to perform the database functions, and uses a similarity metric based on IBM's QBIC system. Querying in Color-WISE is achieved with query-by-image.

The Digital Library Project [5] taking place at the University of Berkeley, California, uses low-level grouping techniques to create "blobs of stuff", which can be texture, colour, or symmetry. The blobs can be matched against their content, and their position, and it is possible to use high-level techniques to analyse the semantics of the blobs (such as where they are in relation to other blobs), and conclude what they might represent.

Image-MINER [1] is an image and video retrieval system developed by the AI group at the University of Bremen. Their colour indexing system for images uses local histograms in a fixed grid geometry. Further grouping of the fixed elements occurs to get 'color-rectangles', which are signatures for their input images. The colour based segmentation module, is part of the larger Image-MINER system which includes video retrieval methods, including shot detection and subsequent 'mosaicing'.

Haung et al. at Cornell University, has developed the idea of a 'correlogram' [8] which is a single feature incorporating both spatial-correlation and colour information. The spatial relationships index on the probability that a pixel will be of a particular colour, when a given pixel is of another colour, and a certain distance from that pixel.

3 Spatial Colour Matching using a Grid

Using a simple histogram based on the whole image content is not sufficient for accurate colour-based retrieval of images, because it has no spatial dimension in the match. This inability to distinguish where the colours are located has led to many variations which take into account the spatial distribution of colours in the image. Taking the location of colour into account, is more likely to retrieve similar images — in effect, adding the extra dimension of matching decreases the possible number of matches and increases the precision of retrieval.

For the spatial colour matching, the starting point was a grid mechanism similar in concept to the Image-MINER and, to some extent, the QBIC approach.

To create a signature, the image is divided into a grid, each element of which contains part of the query image. The grid is overlaid on an image, meaning that a grid element will be of a different size for different images of different sizes. Each grid element has a histogram calculated for it based on the currently selected colour space. The histogram is normalised to ensure that histograms from different sized images are comparable. By using a simple grid mechanism the user interface is kept simple to use and learn.

To save processing at match time, images are usually converted before hand and stored into a database. The module can then retrieve signatures of previous images and match them against the signature of the query image. This requires only the query image to be converted at match time.

4 Colour Histogram Matching

Another desicion which is crucial when creating histograms, is which colour representation to use. The common colour models to use are RGB, HSV (and similar), YUV, CIE L*u*v* (and derivations, like CIE L*a*b*, CMC, BFD and M&S), and Munsell. As a compromise between computational effort, and relevance to the human visual system, the HSV colour model has been chosen for use in the work so far.

To convert RGB pixels into the HSV colour space we use the fast implementation as described in [4] which approximates the HSV cone to a hex-cone, with the corners on the colours red, yellow, green, cyan, blue, and magenta.

Although we have chosen to use the HSV colour model, we have ensured that this is not a final design decision. To this end, the colour model is selectable, but the colour space in to which the query is quantised, must be the same as the space which was used to create the database, so that the histogram match has some meaning.

Matching of histograms uses the histogram quadratic distance technique (weighted Euclidean distance), developed by IBM for their QBIC system [12] for histogram matching on full images, and objects in images. It takes into account the similarity of colours in different bins of the histogram, i.e. how similar the red bin is to the orange bin, which simple histogram distance measures based on the Minkowski metric do not. Each n-dimensional histogram becomes a one-dimensional row vector, the query histogram, H_q, and the histogram to match against, H_m. The histograms are normalised so that the area beneath them is 1. The differences between the frequency stored in each bin, i, is taken, and becomes another one-dimensional vector, Z, as shown in equation 1.

$$Z_i = H_{q_i} - H_{m_i} \tag{1}$$

The match between a bin in histogram H_q and histogram H_m is weighted by the distance between the bins in the colour space. If the distance between the furthest two bins in the colour space is d_{max}, and the distance between bin i and bin j (which is dependent on the colour space) is given by $d(i,j)$, then the weighting matrix, A, is given by equation 2.

$$A_{i,j} = 1 - \frac{d(i,j)}{d_{max}} \tag{2}$$

Matching between the two histograms is then a matter of matrix algebra, given by equation 3.

$$\|Z\| = Z^T A Z \tag{3}$$

The single matrix cell results in a similarity measure, z, where $0 \leq z \leq 2$, and where 0 is an identical match. Conversion to a percentage message of similarity, s, is given by equation 4.

$$s = \frac{2-z}{2} \times 100 \tag{4}$$

5 Matching using a Grid

For any single query, a particular match can be made upon a multiple or single grid element of the query image. Also, a match can take place with or without regard to the location of the grid element. Performing a spatial match, a single grid element in the query image, I_{q_i}, will be matched against the grid elements in the database images at that particular position, I_{m_i}. If n multiple elements are selected, each is compared to its equivalent in the database image and an average taken across the matches. This is expressed in equation 5.

$$\frac{1}{n}\sum_{i=1}^{n} match(I_{m_i}, I_{q_i}) \tag{5}$$

A non-spatial match with only one selected grid element, will match all the grid elements in the database images against that single element in the query image. This returns multiple matches which will give rise to the problem of how to aggregate the matches. The most sensible way, in this case, would be to choose the highest match and give it as the final measure of similarity - so, if we are looking for a very red square, if there is a red square in the database image somewhere, it will be a good match. In a non-spatial match with multiple grid elements selected, the selected elements are each matched against all elements in the database image. For any particular selected query element, the maximum from all the similarities for that element is chosen as the best match. The average is taken over them all to be the match for the whole query.

$$\frac{1}{n}\sum_{i=1}^{n} max\left(\forall j \in I_m, match(I_{m_j}, I_{q_i})\right) \tag{6}$$

6 Spatial Colour Matching using Quad-Trees

A problem that was brought to light in testing was that sometimes deciding whether a particular grid element should be selected for a match was difficult, particularly if it contained a strong colour gradient (for example, it had an edge running through it).

To attempt to overcome this problem the use of quad-trees was introduced to localise changes in the features in an image during segmentation.

A quad-tree is a simplification of the idea of the split and merge algorithm and the T-pyramid [19]. Quad-trees involve recursively dividing the image into quadrants until all elements are homogenous, or until a predefined, "grain", size is reached. Again, a histogram is built for each element in the quad-tree. This means that, in general, the quad-tree method requires more storage than the grid method (and hence longer match times). Unlike the split and merge algorithm which is used for object delineation, the divided areas are not merged again, even if they are adjacent and their total area would fit the homogeneity criteron. This ensures no loss of spatial data.

To match quad-trees with regard to spatial location of elements, requires some cases which did not need to be considered with the grid method. In particular, it is possible that the query selection is smaller than the nearest available element in the quad-tree retrieved from the database. It is also quite possible, that the opposite occurs, and the user's selection in the query is larger than that in the retrieved quad-tree. To recognise these cases, we use the leaf-code, as proposed by Sonka et al. in [19]. Each quadrant is numbered from one to four, left-to-right, top-to-bottom — that is, the top-left quadrant is quadrant-one, and the bottom-right quadrant is quadrant-four. This allows every node in the quad-tree to be represented by a unique code, its length representing the depth at which the node resides. It is then possible to find the nearest node in the quad-tree in the database to the node which is currently being matched from the query image's quad-tree representation. This method also increases the speed of spatial-matching.

If the nearest node in the database quad-tree is the same depth as that in the query node, then they can be directly matched. If the nearest node in the database quad-tree is not a leaf node, then the histograms below that node are summed, so that, effecively, the area in the query image and the area from the retrieved database image are equivalent and can be directly matched. The most awkward case is that when the query quad-tree node is at a greater depth than the nearest match from the database image. Rather than trying to reduce the size of the retrieved selection, the pragmatic approach, suggests that the smaller query area can be directly matched with the larger area from the database. This would seem resaonable because for the images to be segmented in this way, the features must be close to homogenous, and further division would give no better results.

The matching of quad-trees without regard to the spatial location of the elements being matched, is as simple as doing it with grid elements, however it is likely there will be more elements to match, meaning the time taken to match the same database could be longer.

7 User Interface

To ensure that this functionality is all readily available and able to be used, the user interface has been designed in such a way to make selection and querying of images as simple as possible. We have made sure that all the functions that will be required in constructing a query have all been placed in view, to avoid use of menus. The main query viewer for the grid method can be seen in Figure 1, and for the quad-tree method in Figure 3.

Helper tools have been built around the main interface which facilitate creation of a colour, and also the copying of a selection in another image, into an element in the query image. This allows the user to build a completely new

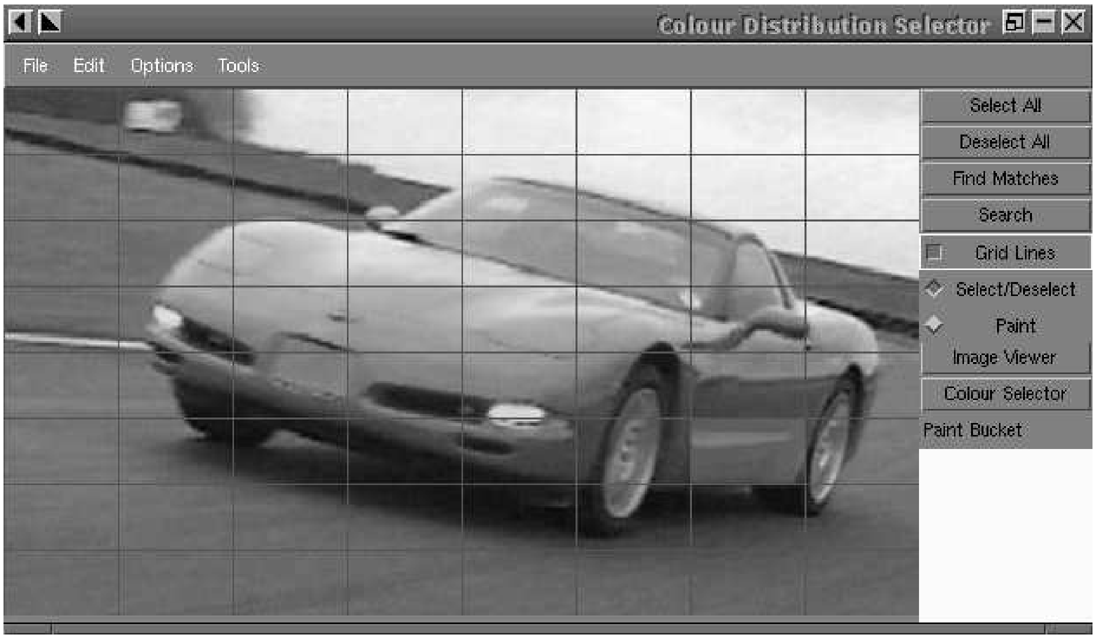

Figure 1: The main viewer for the grid method, with an area of the car selected for matching upon, in a non-spatial match. With the grid lines on, it is possible to see how the images are segmented with this method.

query, not based on a single example image. A simple results viewer was built so that the matches could be viewed in similarity order, and that the best matching grid element be visible if appropriate.

Copying data from colour selectors, or from other areas of images has been achieved using a method which we have called the "paint bucket". This allows users to place into it anything from a helper tool (the colour selector, or image viewer). The user can then change to "paint" mode on the interface which will allow them to copy the contents of paint bucket into elements in the query image. Changing back to "select/deselect" mode lets them change which grid elements are selected for querying. All the windows in the system have been made non-modal so that at any time the user can perform any task, which includes multiple results windows, and image viewers.

8 Experimental Results

As explained in [6], the hue component of a colour is invarient to highlights and shadows, and it is the colour which matters mostly in a search of this kind. After testing various quantisations of the histogram, it was decided to use a segmentation of $12 \times 3 \times 3$, with 12 hue bins, giving 108 bins in a histogram. This division provided enough accuracy to be able to distinguish well between objects, as well as keeping the size of a histogram to a minimum.

Figure 1 shows the main viewer for the grid method. Figure 3 shows the main viewer for the quad-tree method. Both have equivalent areas of the image selected. We can perform a non-spatial match on these areas using both methods. This will look for the small selection in all of the other images, and in any position within these images. Figure 2 shows the top three images of the grid-method. Figure 4 shows the equivalent results for the quad-tree method. It can be seen that the quad-tree method has found the correct images more accurately, due to the ability to have small elements, which are specific to a feature in the image.

We can perform a spatial match using the same tools. In Figure 5 we have selected part of the sky using the grid method. In Figure 7 we have selected the equivalent part of the sky using the quad-tree method. The results are shown in Figure 6 and Figure 8 for the grid method and the quad-tree method, respectively. It can be seen, both methods have

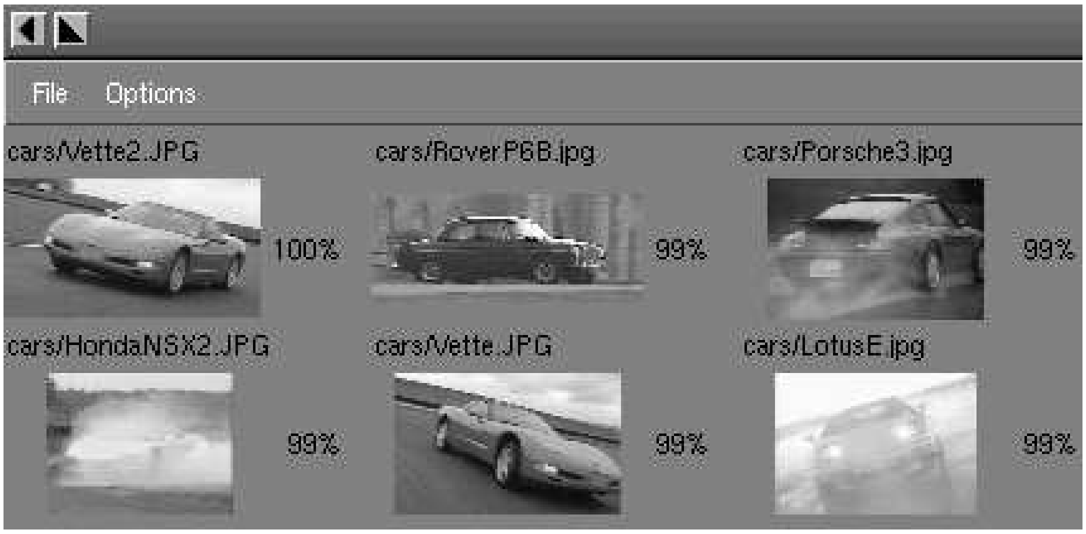

Figure 2: The results from the non-spatial match of the area in Figure 1, using the grid-method.

retrieved relevant images. This suggests that for spatial matching, the grid method is as good as the quad-tree method, and, in practice, the match times were almost identical. The quad-tree method has advantages when used to perform non-spatial matching and also to make the user-interface friendlier, but the grid method is much faster at non-spatial matching due to the smaller amount of data to process.

Further functionality can be achieved by using combinations of the functionality in these spatial-colour matchers. For example, by selecting all the grid elements in a query image, and performing a non-spatial match, is the equivalent of using the global histogram technique for image retrieval.

9 Conclusion and Future Work

A novel approach to image matching and retrieval based on spatially related colour cues using a grid, and quad-trees has been presented. Initial experiments with the approach suggest that it will enhance the precision of image retrieval over other colour retrieval methods, and improve the reliability of content based navigation.

Further work needs to be directed towards a faster method of database retrieval and indexing. A way to achieve this may be to use a pyramid of histograms by using the quad-tree idea, and performing initial esimates of colour matches between images, to remove those that are sure not to yield good results. Rather than using colour, a generalised spatial-feature matcher could be designed using the ideas shown here; rather than extracting colour from a query selection, extracting texture would allow for a spatial-texture matcher. An extension to this system might be to use fast object delineation techniques to select general shapes for matching upon which would require a more complex algorithm for object location matching.

However, it can be seen that there is a limit to the information that can be presented by roughly segmenting an image based on generalised feature methods. Work is taking place on using scale-space methods to produce an object tree based on the features in the image. A matching algorithm for these object-trees could provide the basis for a more generalised feature matcher.

Extensions to MAVIS are also planned that will increase the efficiency of the database retrieval, by using a commercial database to store all the data.

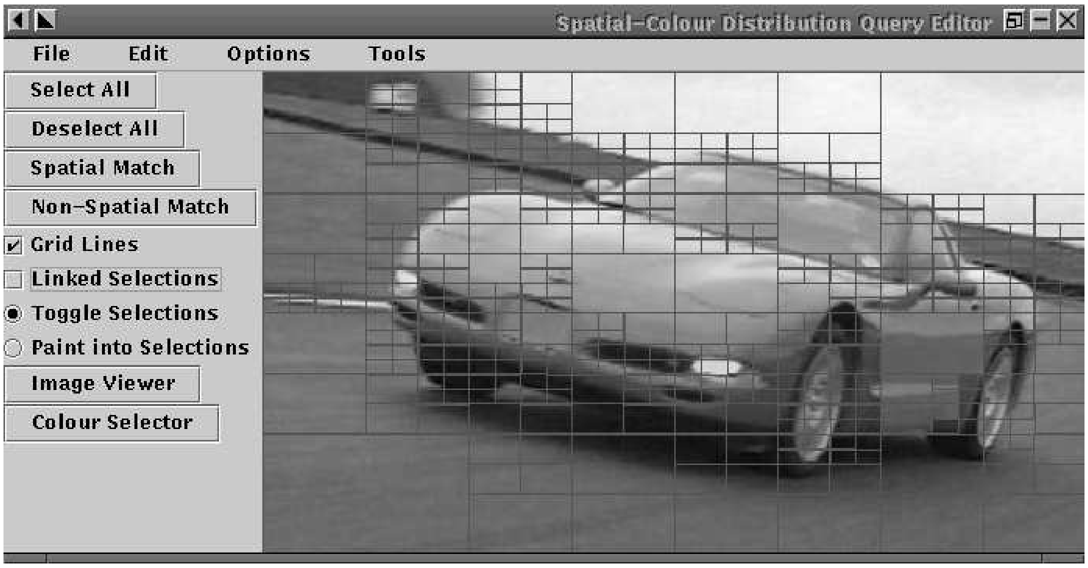

Figure 3: The main viewer for the quad-tree method, with an equivalent area of the car selected for matching upon in a non-spatial match. With the grid lines on, it is possible to see how the images are segmented with this method.

10 Acknowledgements

The first author is grateful to the Engineering and Physical Sciences Research Council (EPSRC) for the support of a research studentship.

References

[1] P. Alshuth, Th. Hermes, J. Kreyß, et al. Image-MINER - intelligent retrieval for images and videos. In Smeulders and Jain [17], pages 241–251.

[2] Hugh C. Davis, Wendy Hall, Ian Heath, Gary J. Hill, and Robert J. Wilkins. Microcosm: An open hypermedia environment for information integration. Technical Report CSTR 92-15, University of Southampton, 1992.

[3] M. Flickner, H. Sawhney, W. Niblack, et al. Query by image and video content: The QBIC system. *Computer*, 28(9):23–32, September 1995.

[4] J.D. Foley, A. Van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 1996. Second Edition in C.

[5] D. Forsyth, J. Malik, and R. Wilensky. Searching for digital pictures. *Scientific American*, pages 72–77, July 1997.

[6] T. Gevers and A.W.M. Smeulders. PicToSeek: A color invariant retrieval system. In Smeulders and Jain [17], pages 25–37.

[7] A. Hampapur, A. Gupta, B. Horowitz, et al. Virage video engine. In Sethi and Jain [15], pages 188–197.

[8] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR97*, pages 762–768, 1997.

Figure 4: The results from the non-spatial match of the area in the Figure 3, using the quad-tree method. The retrieved results are more precise than those using the grid method.

[9] P. Lewis, J. Kuan, S. Perry, et al. Navigating from images using generic links based on image content. In Sethi and Jain [15], pages 238–248.

[10] Paul H. Lewis, Hugh C. Davis, S. Griffiths, Wendy Hall, and Robert J. Wilkins. Media-based navigation with generic links. In *In Hypertext '96. The Seventh ACM conference on Hypertext, Washington*, pages 215–223. ACM Press, 1996.

[11] P.H. Lewis, H.C. Davis, M.R. Dobie, and W. Hall. Towards multimedia thesaurus support for media-based navigation. In Smeulders and Jain [17], pages 111–118.

[12] W. Niblack, R. Barber, W. Equitz, et al. The QBIC project: Querying images by content using color, texture, and shape. *Computer Science*, (RJ9203), February 1993.

[13] W. Niblack, Z. Xiaoming, J.L. Hafner, et al. Updates to the QBIC system. In Sethi and Jain [16], pages 150–161.

[14] I.K. Sethi, I. Coman, B. Day, et al. Color-WISE: A system for image similarity retrieval using color. In Sethi and Jain [16], pages 140–149.

[15] I.K. Sethi and R.C. Jain, editors. *Storage and Retrieval for Image and Video Databases V*, 1997.

[16] I.K. Sethi and R.C. Jain, editors. *Storage and Retrieval for Image and Video Databases VI*, 1998.

[17] A.W.M. Smeulders and R. Jain, editors. *Image Databases and Multimedia Search*, 1997.

[18] J. Smith and S. Chang. VisualSEEk: A fully automated content-based image query system. *ACM Multimedia 1996*, pages 87–98, November 1996.

[19] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thompson Computer Press, 1996.

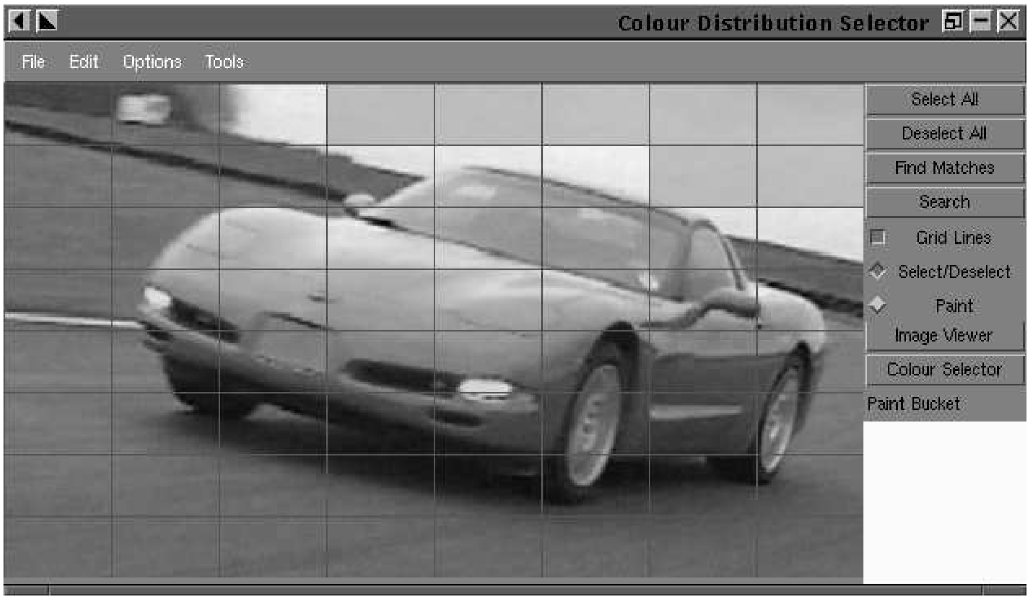

Figure 5: The main viewer for the grid method, with an area of the sky selected for performing a spatial match upon.

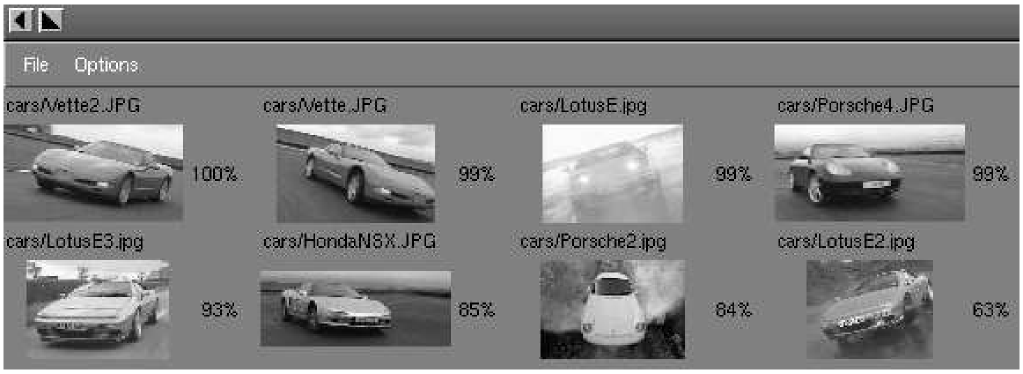

Figure 6: The results of a spatial match of the query presented in Figure 5 using the grid method. The results are very good for the simpler method.

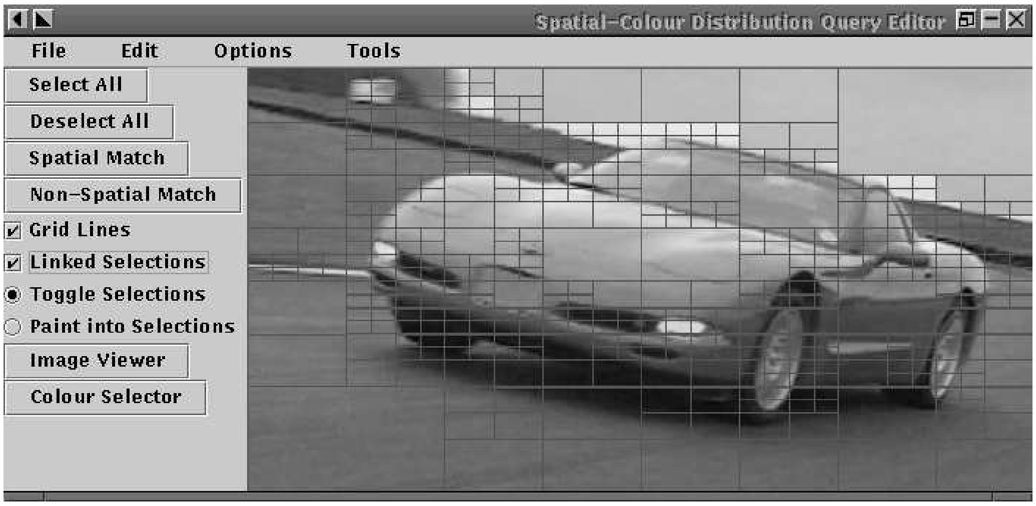

Figure 7: The main viewer for the quad-tree method, with an area of the sky selected for performing a spatial match upon.

Figure 8: The results of the spatial match of the query presented in Figure 7 using the quad-tree method. The results are good, with as good recall and similar precision to those using the grid method.