Efficient Indexing and Retrieval of Colour Image Data Using a Vector-Based Approach

by

Dimitrios Androuotos

A thesis submitted in conformity with the requirements for the degree of
Doctor of Philosophy

Graduate Department
Electrical and Computer Engineering
University of Toronto

©Copyright by Dimitrios Androuotos, 1999
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-45683-8
Efficient Indexing and Retrieval of Colour Images Using a Vector-Based Approach

Doctor of Philosophy, 1999

Dimitrios Androutsos
Department of Electrical and Computer Engineering
University of Toronto

Abstract

COLOUR is the most important low-level feature which is used to build image indices, for retrieval of images from a database. Specifically, the colour histogram remains the most popular method for building such indices, due primarily to its simplicity and fast computation. It has many drawbacks and limitations, however, which degrade its effectiveness.

In this thesis, a new framework is proposed for indexing and retrieving images by colour, using representative RGB colour vectors and a perceptually-tuned vector angular-based measure of colour similarity. The feature extraction and indexing stage incorporates a new recursive HSV-based segmentation scheme which identifies pixels based on perceptual prominence and classifies them as bright chromatic, chromatic, black and white. For each of these classes, hue histogram thresholding is performed, while taking into consideration the multi-modal nature of the saturation histogram. Post-processing operations follow the segmentation to arrive at an accurate low-level representation of the colour image. The average colour of all the extracted regions are then stored in an index, along with the added information of colour categorization, number of regions, and colour amount.

For the retrieval process, a new technique is proposed using a multidimensional query distance space, whose dimension is determined by the number of query colours. All images which exhibit similarity to the query colours occupy a position in this space and it is this position, and its relation to the origin and the equidistant line, that determines the overall similarity and ranking of a given image to a colour query. The concept proves to be quite flexible in how queries can be structured, allowing any number of query colours, query-by-example, and also effectively allows the exclusion of a specific colour in a query. An exclusion colour adds an additional component to the multidimensional query distance space which affects the overall position of a given image in the space.

Through extensive testing, it is shown that the proposed colour image retrieval scheme and measure exhibit very high performance in terms of retrieval rate, precision, and recall, especially over common colour histogram techniques and IBM's QBIC system. This was established through Human Query Sets obtained from 25 human volunteers. These sets of images represent the retrieval results containing images judged by humans to best fit a specified query and are used to analyze and compare the retrieval results of the system. Furthermore, the proposed scheme and measure exhibit high resistance to gamma nonlinearity. Retrieval results show that the retrieval rate is the highest over a wider range of gamma nonlinearity values than all other vector measures investigated, and also colour histogram techniques.
Acknowledgements

After so many years at the University of Toronto and especially in the graduate environment, there are many people with which I have interacted and have had the honour of meeting and working with. Without a doubt, my supervisor, friend and mentor Professor Venetsanopoulos, or Tas as we know him, has had the greatest impact on my development as a student, as a researcher and as a person. His character and personality made sure that my daily environment was amusing, intellectual and fruitful, without making me feel like just a student, but rather as an equal. Words cannot convey the respect and gratitude I have towards him.

A very large part of my research career is a result of my close friendship with Professor Kostas Plataniotis. Without him, my publication record would be much shorter and my grasp of advanced image processing would be minimal. I have yet to meet a more dedicated and diligent researcher. Kosta had the insight to focus on new areas of research and ultimately steered me into the direction of image retrieval and multimedia. However, our conversations, daily lunches and innumerable visits to the coffee shop is what helped build a mutual respect and a strong friendship.

The stress and long hours of the last few years of my research had rendered me somewhat unbearable at times. Through all of this, one person endured my frustration and beared my wrath, yet he still survived unscathed. That person is my closest friend Peter who, as chance has it, is also my brother. The torch now passes to him.

I also would like to thank the members of my defense committee. Professor Frank Kschischang, in particular, made an extra effort to proofread my work and spent time to suggest many changes and additions. Also, Professor Wasyl Janischewskyj, Professor Elvino Sousa, and Professor Demetri Terzopoulos provided very encouraging comments and insight to help improve my work. Also, I would like to thank Professor Sethuraman Panchanathan who took time out of his very busy schedule to be at my defense. His interest and enthusiasm in my work was the highest affirmation.

I would also like to thank my lab brothers Nick Herodotou and Nick Ikonomakis who made life in the DSP lab entertaining, to say the least.

Finally, my parents deserve my deepest and most wholehearted thanks. As I get older, I realize more and more what my parents have done for me. I realize what sacrifices they have made and how hard they have worked. They instilled strong family and cultural values and a strong sense of honour and integrity. They were also the ones to make me look upwards.
"Tis the truth of my experiments which is the business in hand. On this my Theory depends, and which is of more consequence, the credit of my being wary, accurate and faithfull in the reports I have made."

Sir Isaac Newton
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Textual Information</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Visual Information</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Content-Based Image Retrieval</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Colour Indexing</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Colour Similarity</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Thesis and Scientific Contribution</td>
<td>6</td>
</tr>
<tr>
<td>1.5 Outline</td>
<td>7</td>
</tr>
<tr>
<td>2 Colour Science</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Colour</td>
<td>10</td>
</tr>
<tr>
<td>2.1.1 Human Visual Pathway</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2 Trichromacy Theory</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Colour Images</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Gamma Nonlinearity</td>
<td>14</td>
</tr>
<tr>
<td>2.4 Colour Spaces</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1 RGB Space</td>
<td>17</td>
</tr>
<tr>
<td>2.4.2 Perceptual Colour Space</td>
<td>18</td>
</tr>
<tr>
<td>2.4.3 Uniform Colour Spaces</td>
<td>21</td>
</tr>
<tr>
<td>CIELAB</td>
<td>21</td>
</tr>
<tr>
<td>CIELUV</td>
<td>22</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Other Colour Spaces</td>
</tr>
<tr>
<td></td>
<td>Munsell Colour Space</td>
</tr>
<tr>
<td></td>
<td>YIQ &amp; YUV colour Spaces</td>
</tr>
<tr>
<td>2.5</td>
<td>Discussion</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Image Indexing and Retrieval</td>
</tr>
<tr>
<td></td>
<td>Feature Extraction</td>
</tr>
<tr>
<td></td>
<td>Colour Similarity</td>
</tr>
<tr>
<td>3</td>
<td>Colour Similarity Measures</td>
</tr>
<tr>
<td>3.1</td>
<td>Dissimilarity Measures</td>
</tr>
<tr>
<td>3.2</td>
<td>Similarity Measures</td>
</tr>
<tr>
<td>3.3</td>
<td>Metricity</td>
</tr>
<tr>
<td>3.4</td>
<td>Gamma Nonlinearity &amp; Similarity</td>
</tr>
<tr>
<td>3.5</td>
<td>Proposed Distance Measure</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Perceptual Tuning</td>
</tr>
<tr>
<td>4</td>
<td>Colour Image Indexing</td>
</tr>
<tr>
<td>4.1</td>
<td>Current Techniques</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Colour Histogram</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Colour Segmentation</td>
</tr>
<tr>
<td></td>
<td>Pixel-Based</td>
</tr>
<tr>
<td></td>
<td>Region-Based</td>
</tr>
<tr>
<td>4.2</td>
<td>Retrieval-Specific Criteria</td>
</tr>
<tr>
<td>4.3</td>
<td>Proposed Feature Extraction Scheme</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Recursive Colour Segmentation</td>
</tr>
<tr>
<td></td>
<td>Bright Chromatic Pixels</td>
</tr>
<tr>
<td></td>
<td>Black and White Pixels</td>
</tr>
<tr>
<td></td>
<td>Chromatic Pixels</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Post-processing</td>
</tr>
<tr>
<td></td>
<td>Median Filtering</td>
</tr>
<tr>
<td></td>
<td>Morphological Processing</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Object Detection and Removal</td>
</tr>
<tr>
<td>4.4</td>
<td>Index Creation</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Representative Colour Vector</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Colour Categorization</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Index Structure</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Computation and Storage Requirements</td>
</tr>
</tbody>
</table>
Quantitative Analysis ......................................................... 105
6.3.2 Multiple Colour Query .............................................. 106
Qualitative Analysis .......................................................... 108
Quantitative Analysis ......................................................... 108
6.3.3 Overall Performance .................................................. 112
6.3.4 Comparison to QBIC .................................................. 114
6.4 Query-By-Example ....................................................... 114
6.5 Colour Exclusion .......................................................... 117
6.6 Retrieval & Gamma ....................................................... 119
6.7 Summary ................................................................ 129

7 Conclusions & Future Work .............................................. 130
7.1 New Measure of Colour Similarity ..................................... 130
7.2 Indexing .................................................................. 131
   Recursive HSV Space Segmentation .................................. 131
   Post Processing Operations ............................................. 131
   Representative Vectors .................................................. 131
   Colour Categorization ................................................... 131
7.3 Retrieval ............................................................... 132
   Multidimensional Query Distance Space ......................... 132
   Flexible Query Structure .............................................. 132
   Results ................................................................ 133
7.4 Future Work .............................................................. 133

References .................................................................. 135

A Image Database .......................................................... 145
List of Figures

2.1 A diagram showing the wavelength range of the electromagnetic spectrum and the corresponding types of radiation which exists, with the visible light spectrum enlarged. 10
2.2 A three-dimensional cut-out view of the human eye. ................................. 11
2.3 A view of the surface of the human retina as it looks through an ophalmoscope (left) and an illustration of the histological layers of the retina, clearly showing RODS and CONES (right). ........................................... 11
2.4 An axial Magnetic Resonance image of the human brain depicting the eye, the optic nerve and the location of the visual cortex. .......................... 12
2.5 Approximate cone absorption curves across the visible spectrum. The S curve peaks in the red, the M in the green and the L curve peaks in the blue. ......................... 13
2.6 Colour CCD array. ........................................................................... 14
2.7 Colour scenes can be acquired by any number of techniques. Traditional techniques such as photographic film and human drawings, once printed can be digitized by a scanner. Digital techniques allow for the images to be processed, displayed, stored, printed, and transmitted easily. ........................................ 15
2.8 Two identical images with different $\gamma$. Clearly, the colour distribution of the two images is not the same. ....................................................... 16
2.9 A three-dimensional view of the surface of the colour cube in the RGB space for 8-bit colour planes. The three components form a colour cube in which the position of an RGB vector determines its colour. ...................................... 18
2.10 A colour with hue = 0° and value = 100, whose saturation is varied. Notice that as the saturation is decreased the colour appears whiter. Since a maximally saturated colour is essentially one specific wavelength, decreasing saturation involves the addition of other wavelengths in the light which is emitted. A completely unsaturated colour contains all wavelengths and thus appears totally white. .................... 19
2.11 HSV cone .................................................................................. 20
2.12 A colour with hue $= 0^\circ$ and saturation $= 100$, whose value is varied. Notice how the colour appears brighter as it moves from minimum, where it appear black, to maximum value.

2.13 Munsell Colour Space. Munsell used red (R), yellow (Y), green (G), blue (B), and purple (P) as his principal hues, placed them at equal intervals around a circle and then inserted five intermediate hues of yellow-red (YR), green-yellow (GY), blue-green (BG), purple-blue (PB), and red-purple (RP).

3.1 Similarity measures map close proximity of vectors to a value of 1, whereas dissimilarity measures map close proximity to 0.

3.2 6 swatches depicting colour change around a central colour vector $\vec{C}$, with RGB values of (187, 83, 78), due to changes in the angular distance with respect to that vector. The angle is calculated for angles of 0.05, 0.1, 0.15, 0.20, 0.25 and 0.5 rad for 8 points around $\vec{C}$. Each swatch shows the effect at a different point along $\vec{C}$, specifically at 125%, 100%, 75%, 50% and 25% of $|\vec{C}|$.

3.3 Two vectors $\vec{x}_i$ and $\vec{x}_j$ and their corresponding projections $h_i$ and $h_j$.

3.4 Colour comparison depicting the non-transitive property of human colour perception. Colour pairs in (a) baby blue & aquamarine exhibit similarity as do colour pairs (b) aquamarine & turquoise. However, baby blue & turquoise exhibit low similarity. Thus, triangular inequality can fail.

3.5 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the vector angular measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

3.6 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_1$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

3.7 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_2$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

3.8 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_\infty$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

3.9 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the Canberra measure. Curves which have a higher position represent higher similarity to the gamma varied colour.
3.10 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the Czekanowski coefficient. Curves which have a higher position represent higher similarity to the gamma varied colour.

3.11 Plot of the similarity function of Eq. 3.20. It depicts the similarity between vectors for different angles and magnitude differences.

3.12 Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the proposed new measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

4.1 Typical HUE histogram.
4.2 HSV cone depicting BLACK, WHITE, BRIGHT CHROMATIC, and CHROMATIC regions.
4.3 Typical SATURATION histogram.
4.4 Flowchart of segmentation procedure.
4.5 Step by step visualization of the segmentation procedure.
4.6 3-D view of segmented colour binary images.
4.7 Morphological EROSION, depicting the original object $X$, the structuring element $B$ and the eroded result.
4.8 Morphological DILATION, depicting the original object $X$, the structuring element $B$ and the dilated result.
4.9 Structuring element implemented in the post-processing operations.
4.10 Flowchart of the object detection and removal algorithm.
4.11 Graphical representation of the contour following algorithm operating on a typical binary object which also contains an inner object.
4.12 Step-by-step view of the post-processing stage. $A$ is the thresholded binary image, $B$ is the median filtered result, $C$ is the morphologically processed result, and $D$ is the final processed binary image after object-removal. The coloured circles draw attention to the typical effects of each post-processing stage.
4.13 Step by step visualization of the segmentation procedure portraying the thresholded regions and their post-processed result.
4.14 Sample image and its feature extraction results. (a) Original image, (b) segmented result and (c) the post-processed result.
4.15 HUE component categorized into 5 regions with their transition regions.
4.16 Index structure. The first entry in the index is the total number of colours present in the corresponding image followed by the colour category tags, which give a "quick look" into the colour content of the index. Next, the representative colour vectors are stored starting with their 8-bit RGB values, the total percentage of the colour present in the image, the number of regions or objects containing the colour in question followed by the area, perimeter and percentage of each object.

4.17 Bit allocation for each entry in the image index.

5.1 Flowchart depicting the general scheme of image retrieval.

5.2 Three histograms arranged so that neighbouring bins are perceptually similar. The calculated similarity of these histograms with the Minkowski metrics do not correspond to their perceived similarity. Specifically, $H_1$ and $H_2$ are determined to be the most dissimilar pairing, for both the $L_1$ and $L_2$ metric, which is clearly not the case. Using these measures we find that $d(H_1, H_2) > d(H_2, H_3) = d(H_1, H_3)$.

5.3 Flowchart depicting the components of the retrieval stage.

5.4 Our system's GUI interface where the query colours are selected.

5.5 Our system's GUI interface showing the tools available for defining a query.

5.6 The intersection of the QuickLook vector and the index header produces the QuickMatch vector, which indicates which colours are common to both the user query and the index.

5.7 A visualization of the MQDS for 3 query colours, showing the equidistant line.

5.8 A visualization of the MQDS for the two dimensional case, i.e., for two query colours. The query colours were $RGB = 26, 153, 33$ (green) and $RGB = 200, 7, 25$ (red). A set of retrieved images are displayed at various points in this 2-D space. Their location represents the point in space where their corresponding $\bar{D}$ exist.

5.9 A graphical view of the MQDS. (a) Two query colours $q_1$ and $q_2$, the MQDV $\bar{D}$, and the equidistant line. (b) When an exclusion colour is specified, the vector $\bar{E}$ pulls $\bar{D}$ away from the equidistant line.

5.10 Plot (a) shows the maximum and minimum number of coloured regions required for two images to be perceived as similar. Plot (b) depicts the tolerance, or spread, of the number of coloured regions between two images which are required for the two images to be considered similar solely on cardinal colour quantity.

5.11 Block diagram depicting the steps which our system performs during query-by-example.

5.12 Dialogue box which appears when search results are to be refined on the AltaVista search site.

5.13 Query Refine Dialog Box.

5.14 Scatter plot of the top ten colour swatches, clearly showing the query clusters.
5.15 Query Refine by Image Dialog Box .......................................................... 95
5.16 The front-end of the image retrieval system showing the results from a sample query. 96

6.1 Colour swatches depicting the five colours which were used to find the human query sets. ............................................................. 98
6.2 Human Query Sets Q determined from data collected from 25 volunteers. (a) >50% seagreen (b) >25% orange (c) >25% yellow (d) >25% red & >25% green (e) >25% red & >25% blue. .................................................. 99
6.3 Typical retrieval effectiveness graphs, which plot recall vs. precision. Curves which are further away from the origin represent systems with better performance. ........ 101
6.4 Retrieval results for Query Q: at least 25% seagreen using (a) new measure, (b) angular distance, (c) L1 norm, (d) L2 norm, (e) \( L_\infty \) norm and the results using histogram techniques (f) \( RGB(8,8,8) \) and (g) \( HSV(13,5,5) \). The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results. .......................................................... 102
6.5 Retrieval results for Query Q: at least 25% orange using (a) new measure, (b) angular distance, (c) L1 norm, (d) L2 norm, (e) \( L_\infty \) norm and the results using histogram techniques (f) \( RGB(8,8,8) \) and (g) \( HSV(13,5,5) \). The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results. .......................................................... 103
6.6 Retrieval results for Query Q: at least 25% yellow using (a) new measure, (b) angular distance, (c) L1 norm, (d) L2 norm, (e) \( L_\infty \) norm and the results using histogram techniques (f) \( RGB(8,8,8) \) and (g) \( HSV(13,5,5) \). The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results. .......................................................... 104
6.7 Precision-recall graphs depicting retrieval performance for the vector-based measures for (a) Query Q: seagreen, Query Q: orange and Query Q: yellow. ......................... 106
6.8 Precision-recall graphs depicting retrieval performance for the histogram-based schemes for (a) Query Q: seagreen, (b) Query Q: orange and (c) Query Q: yellow. .......... 107
6.9 Retrieval results for Query Q: at least 25% red & green using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_\infty$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,5,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.

6.10 Retrieval results for Query S: at least 25% red & blue using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_\infty$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,5,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.

6.11 Precision-recall graphs depicting retrieval performance for two-colour queries. Plots depicted show retrieval effectiveness for vector measures for: (a) Query Q; red & green and (b) Query S; red & blue, and also for colour histogram schemes for (c) Query Q and (d) Query S.

6.12 Average retrieval effectiveness over the 5 test query cases. Graphs depict the average retrieval effectiveness for (a) vector measures and (b) histogram schemes. (c) Shows all precision-recall plots on the same graph.

6.13 Retrieval results produced by QBIC on the 1850 image database for queries: (a) Ω, (b) Θ, (c) Ξ, (d) Ω, and (e) Ω.

6.14 Average retrieval effectiveness over the 5 test query cases. Graphs depict the average retrieval effectiveness for (a) QBIC and the new measure and system. (b) Shows all precision-recall plots for all measures, and QBIC, on the same graph.

6.15 Query-by-example result using the proposed system and new measure. The top left image, beans, is the input to the system. The similarity of the retrieved images is in decreasing order from top left to bottom right.

6.16 Query-by-example result using the proposed system and new measure. The top left image, candy is the input to the system. The similarity of the retrieved images is in decreasing order from top left to bottom right.

6.17 Query result for images with (a) red & green (b) Image in grey-scale are those images containing yellow which should not be retrieved when the query excludes yellow and (c) the actual results obtained from our system when querying for red & green while excluding any amount of yellow.

6.18 bat image and the colour sea green.
6.19 Top 9 retrieval results, using the proposed system and new measure, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 121

6.20 Top 9 retrieval results, using the proposed system and the angular measure, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 122

6.21 Top 9 retrieval results, using the proposed system and the $L_1$ norm, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 123

6.22 Top 9 retrieval results, using the proposed system and the $L_2$ norm, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 124

6.23 Top 9 retrieval results, using the RGB colour histogram with (8,8,8) quantization, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 125

6.24 Top 9 retrieval results, using the RGB colour histogram with (13,5,5) quantization, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query Q. ............................................. 126

6.25 Plot depicting the effect of gamma nonlinearity on retrieval rate. .......................... 127
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Number of crossovers present in the plots of Figures 3.5-3.10.</td>
<td>39</td>
</tr>
<tr>
<td>4.1</td>
<td>HUE ranges which are used for categorizing representative vectors. Note that HUE is a circular parameter and the ranges are given in degrees.</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>Processing time for 4 different CPUs. The database was indexed four times, once using each processor, to calculate the minimum, maximum, and average processing times for each image.</td>
<td>64</td>
</tr>
<tr>
<td>6.1</td>
<td>Number of images in the top 25 retrieval results, that belong to the Human Query Set of their corresponding query, for the 5 vector measures and 2 histogram schemes. Entries in <strong>boldface</strong> denote the highest values.</td>
<td>105</td>
</tr>
<tr>
<td>6.2</td>
<td>Number of images in the top 25 retrieval results, that belong to the Human Query Set of query Q and Q, for the 5 vector measures and 2 histogram schemes. Entries in <strong>boldface</strong> denote the highest values.</td>
<td>108</td>
</tr>
<tr>
<td>6.3</td>
<td>Retrieval rate for 5 different vector distance measures and 2 histogram techniques. Entries in <strong>boldface</strong> denote the highest values. A bar-graph is also included with a graphical view of the retrieval rates.</td>
<td>112</td>
</tr>
<tr>
<td>6.4</td>
<td>Retrieval rate for 5 different vector distance measures, 2 histogram techniques and the QBIC system. Entries in <strong>boldface</strong> denote the highest values. A bar-graph is also included with a graphical view of the retrieval rates.</td>
<td>117</td>
</tr>
<tr>
<td>6.5</td>
<td>Comparison of the number of appearances each image makes in each retrieval result at 10 different gamma levels.</td>
<td>128</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

As we enter the twenty-first century we leave behind the Information Age and enter the Digital Age. With the steady infiltration of digital computers into every aspect of human life, the way in which information is created, stored, processed, and disseminated is rapidly changing. Representing information with binary code proves to be more versatile and flexible than analog methods. The ubiquitous office has been replaced by a workstation, and the filing cabinet by a hard drive. The vinyl record by the compact disc, VHS by Digital Versatile Disc (DVD), and film by charge coupled devices (CCD). In actuality, we are witnessing a conversion of all media and information into machine-processable digital form, and a convergence of service and access to the desktop computer.

The ease and simplicity with which this digital revolution is taking place is not without problems. Primarily, the amount of digital information is growing without bound. If we also take into consideration the explosive growth of the Internet and the World Wide Web, the amount of information available is incomprehensible. Consequently, we find our attention is shifting from amassing information to efficiently accessing it. The focus now is to be able to search and find relevant and important information from within a vast stockpile. Ultimately, it is this access which directly determines the value of information.

1.1 Textual Information

Text retrieval is a well researched area. There are many efficient and effective techniques for searching textual databases and many of them are widely implemented in a multitude of applications. One has to look no further than a public library or click no further than their favourite Internet search engine to see powerful and effective text retrieval at work.

The high success of the area is attributed to the semantic nature of text. Similarity and matching
1.2 VISUAL INFORMATION

is based on the presence or absence of specified words. For example, if we search a university library for text retrieval, we are presented by a finite number of items which contain these two words in their title. We can even search by author or even ISBN number. This inherent low dimensionality of text allows for relatively easy and effective retrieval using a variety of powerful indexing techniques, such as hash tables and B-trees.

1.2 Visual Information

The complexity and challenge of searching the “Digital Haystack” takes on a few orders of magnitude when we deal with visual data such as images and video. The importance of accessing visual data has increased as many industrial and commercial fields, which deal with image capture and analysis, have now opted for digital acquisition and processing:

- Satellite Imaging
- Medical Imaging
- Forensics
- Broadcasting
- Graphic Arts
- Photography
- Textile Industry
- Criminology
- Consumer Imaging
- Photojournalism

The importance of efficient access of visual data to the above-mentioned fields is obvious. The difficulty, however, lies in the fact that visual data has to be processed to extract objects or features that have meaning. Unfortunately, this meaning is not well-defined. The interpretation of visual data is subject to human perception which, at present, cannot be modeled or mimicked by even the most advanced computer vision and image processing techniques.

The ideal scenario is one where a person can ask, or query, a visual database to retrieve a set of images based on their content. For example, the statement:

find all images which contain a red-brick house with a picket fence, a cloudy sky and a yellow sports-car

is typical of the kind of query which a user would naturally be tempted to make. The inherent complexity involved with processing such a query and actually retrieving valid images, however, cannot be overemphasized. The primary factors being:

perceptual interpretation: For each person a query, such as the one above, can have a different interpretation. What constitutes red or yellow to one person is different for another. Similarly, terms such as house, clouds, and sports-car are subject to a person’s preference, experience, mood, and purpose.

©Dimitrios Androutsos, 1999
1.3. CONTENT-BASED IMAGE RETRIEVAL

A high-level classification: At present, computer vision and image understanding techniques do not come close to achieving extraction and classification of image information which could satisfy such queries.

Early attempts in performing image retrieval implemented keyword annotation, whereby a human would manually enter keywords describing the content of a given image to build database indices. The images would then be retrieved using text-retrieval techniques on these keywords. Major drawbacks, however, exist with this approach, namely:

- annotations are dependent on the interpretation of the human indexer;
- the richness of image information cannot be captured with a finite number of keywords;
- inconsistencies between annotations develop;
- manual annotation task becomes infeasible as database grows;
- query dictionary is limited by keywords, e.g., aquamarine has no relation to blue unless the indexer included both of these words in the annotation.

1.3 Content-Based Image Retrieval

Consequently, there is a great need for new approaches to image retrieval. Techniques and tools are required which can automatically process digital images to extract information which will allow for querying of images based on their visual content.

Content-Based Image Retrieval (CBIR) is a research area which is still in its infancy, dedicated to the image retrieval problem. Numerous research communities have converged in this area and found a common playing field where ideas, research, and methods work together towards this common goal. The primary research areas involved are:

- Database Management
- Image Processing
- Computer Vision

The outcome has been a flurry of research which has resulted in a number of systems and techniques developed to tackle the problem, both in the academic and commercial domain. Most notably, IBM’s Query By Image Content (QBIC) system [1] and Virage Systems’ Visual Information Retrieval (VIR) engine [2, 3] are two popular commercial image retrieval systems available today. Columbia University’s VisualSEEk [4], MIT Media Lab’s PhotoBook [5], CHABOT at the University of California at Berkeley [6], NETRA at the University of Southern California at Santa Barbara [7], Color-WISE at Wayne State University [8] and the PICASSO system at the University of Firenze [9] are well known image retrieval systems which exist in academia. The results from such systems...
1.3. CONTENT-BASED IMAGE RETRIEVAL

are impressive and show much promise, but at the same time they also show that much work has yet to be done.

A key aspect of all CBIR systems is the creation of robust and efficient indices which are used for the actual retrieval. These indices are to CBIR as keywords are to text-annotated image retrieval. The difference, however, is that image indices are features such as colour, texture, and shape, which have been extracted using computational methods. Image retrieval is then performed by similarity matching on the stored indices.

Typical systems follow a basic approach to image retrieval:

1. **POPULATION**: The first stage of most systems deals with the actual population of the database. Images are digitally processed to extract relevant information to build an index. This step, known as **feature extraction**, is mostly an automated task, although some high-level human interaction may also be incorporated.

2. **QUERY**: Users are given an interface in which they can query the image database in a variety of ways. It is common to allow users to search for images based on colour content, texture, and shape. Also, users can query by example, whereby they provide a sketch or another image as input so that similar images are returned.

3. **SEARCHING**: Once a query has been constructed, a **query index** is created which is used to compare against all the database indices. Similarity between the query and the database images is determined through a specific measure.

4. **RETRIEVAL**: The calculated value, or measure of similarity, is used to rank the validity of a given image. Since it is not possible to get an exact match, the similarity ranking is used so that a group of images which are considered as best candidates are presented to the user.

5. **FEEDBACK**: Most systems provide some means of relevance feedback whereby a user's input, as to the validity of retrieved images, is used to refine a given query.

Since images are retrieved based on the similarities calculated from their indices, the most crucial parts of any image retrieval system are 1 and 2 above. Unlike text annotation where the information to be indexed is well-defined, features need to be extracted which capture the contextual information of each image. How they are extracted and how well they capture image information coupled with a valid similarity measure, directly determines the effectiveness of the retrieval results.

1.3.1 Colour Indexing

Colour is the most important low-level feature which is used to build image indices. Specifically, the colour histogram remains the most popular method for capturing low-level colour information which
1.3. CONTENT-BASED IMAGE RETRIEVAL

is used to build an image index. It proves to be very simple and fast to compute, and similarity calculations can be done very easily. Using the colour histogram for indexing, however, has a number of drawbacks:

- histograms require quantization to reduce the dimensionality. A typical 24-bit colour image, of \(512 \times 512\) resolution, generates a histogram with \(2^{24}\) bins, which translates to approximately 50 megabytes of storage space. With quantization, however, comes loss of colour information and there is no set rule as to how much quantization should be done.

- the colour space which is being histogrammed can have a profound effect on the retrieval results and also governs the amount of quantization.

- colour exclusion is difficult using histogram techniques. A CBIR system requires a means to allow certain colours to be *excluded* from a user-defined query right from the start, without requiring an additional level of analysis.

- histograms can provide erroneous retrieval results in the presence of gamma nonlinearity. In general, an image database can contain images acquired from many unknown sources and can pass through a number of stages from the moment it is captured to the moment it is displayed. This poses a problem. For example, a scene can be captured on photographic film, transferred to paper and then scanned to digital format where it can be displayed on any computer monitor. These stages introduce a multiplicative nonlinearity due to the gamma nonlinearity of the various equipment. For image retrieval this can result in very poor performance. It can cause false retrievals and render comparisons and similarity measures between pixel values, and ultimately images, erroneous.

- the histogram captures *global* colour activity; no spatial information is available. To include spatial information requires each image to be partitioned into \(n\) regions and a histogram built for each region, which consequently requires \(n\) times more storage.

1.3.2 Colour Similarity

The importance for valid and robust colour indices is clear. Without a meaningful measure of their similarity, however, they prove ineffective. Unfortunately, what is considered "meaningful" is highly subjective and dependent on human perception. Thus, it is beneficial to include aspects of human colour perception and similarity matching when calculating similarity between indexed colours.

In current systems, indexed colour features, such as a histogram, are treated as high-dimensional vectors. How close these vectors are to a query vector determines how close a query and a database image are. Popular measures which are used for similarity comparison include histogram intersection, Minkowski metrics, Hamming distances, quadratic-form distance measures, and histogram
moments. These measures, however, have no real perceptual basis and do little to mimic human colour similarity. Moreover, the fact that colour features are treated as a group to form a high-dimensional vector can render many perceptually relevant colours insignificant and further hinder perceptual comparison.

1.4 Thesis and Scientific Contribution

In this work, a novel scheme for extracting, indexing and retrieving colour image data is presented. We do away with histogram indexing techniques and instead implement RGB vector techniques. This way we end up with a much smaller index which does not have the over-completeness or granularity of a colour histogram, yet retrieval performance is better and more robust. More specifically,

1) EXTRACTION/INDEXING

- For each database image, recursive HSV-space segmentation is performed to extract regions of prominent and perceptually relevant colour.
- The number of extracted colours proves to be very low, yet very effective.
- The average RGB values of the extracted colours are used as representative image vectors which are stored and indexed, along with information regarding the amount of each colour present in each image, the number of regions which contain each colour, and the approximate colour category to which they belong.

2) RETRIEVAL

- A perceptually-tuned similarity measure, based on the vector angular distance, is introduced and implemented to calculate RGB colour vector similarity. The concept stems from research in vector image filtering, where the angular distance between vectors determines filter output. The measure is a combination of vector-angular distance and magnitude difference, perceptually-tuned through an exponential function.
- A Multidimensional Query Distance Space (MQDS) is introduced. This is a query-dependent space whose dimension is defined by the number of query colours. Specifically, the distances of the closest indexed colours to the query colours form a vector which lies in the MQDS. The location of this distance vector within the space and its relation to the origin and equidistant line, determines the overall ranking of a given image.
- The proposed scheme exhibits great flexibility. Querying can be performed in a number of ways, including query-by-colour and query-by-example. In addition, the proposed method of retrieval allows incorporation of colour exclusion, where certain colours can
be specified in the query to not be present in any of the retrieved images. By virtue of the Multidimensional Query Distance Space, the similarity that indexed colours have to exclusion colours is used to affect the overall ranking of a given image without requiring a separate filtering stage to remove images with unwanted colours.

✓ Retrieval results using the proposed scheme exhibit higher retrieval efficiency than many popular histogram indexing techniques. In addition, the results agree much more closely with results tabulated from 25 human volunteers, who manually searched through a 1850 image database and selected images which were “felt” to match a given query.

✓ Resistance to gamma nonlinearity is investigated and found that the proposed method and similarity measure resist changes to gamma much better than other techniques, including histogram techniques.

5 RELEVANCE FEEDBACK

✓ We also address relevance feedback by implementing a scheme similar to that which is used by Internet search sites. For each query result, a user has the option of being presented with colour swatches. These swatches contain the representative colours, from the highest ranked images, which are the closest match to the initial query colours. The user can then refine retrieval results by selecting which of these colours should be included or excluded. New refined colour query vectors are then determined using the sum-of-angles criterion which are subsequently used to retrieve a new set of results.

1.5 Outline

The outline of this thesis is as follows: Chapter 2 discusses some background and research in colour theory and perception, which is important to colour image retrieval. Chapter 3 discusses common measures of similarity which are used for histogram similarity and some common RGB vector similarity measures. The proposed perceptually-tuned similarity measure is then introduced and explained in detail. Chapter 4 discusses some common feature extraction and indexing techniques and discusses the proposed recursive segmentation scheme and the representative RGB vectors which comprise this schemes’ indices. Chapter 5 deals with the integration of the various components which comprise the retrieval system. The process of searching, similarity calculation, and image ranking, via a proposed new method known as the Multidimensional Query Distance Space, are discussed in this chapter. In addition, our method of relevance feedback or query refinement, using colour swatches from retrieved images, is also presented as a component of the entire proposed retrieval scheme. In addition, colour exclusion is addressed in this chapter and it is shown how it is easily included in the query structure and in the overall similarity measure. Chapter 6 presents some...
retrieval results based on a number of different query types, including query-by-colour and query-by-example, where retrieval efficiency is determined from tabulated human results. Furthermore, we also investigate the effect of gamma nonlinearity on our retrieval system, where it is found that our proposed scheme and measure provide more robust retrieval results in the presence of gamma nonlinearity as compared to popular histogram techniques. Finally, Chapter 7 concludes the thesis and provides directions for future research.
Chapter 2

Colour Science

The role of colour in nature and human culture has always been one of messaging and conveyance of information. Fauna use colour to attract mates or for camouflage. Flora make use of colour to attract attention to their seeds to increase the possibility of being carried away to germinate elsewhere. Humans have used colour in similar ways; from gathering food to identifying membership of clan, organization, or country.

Although in today's society colour is not essential for human survival it is, nonetheless, a very important aspect of everyday life and has prompted much research ranging from the physical to the physiological and psychological aspect. In particular, colour acquisition and reproduction has been an especially active area of study since it applies to such a wide range of important fields. From graphic arts and photography to digital cameras and laser printers, accurate acquisition and reproduction of colour is essential.

In this chapter, we discuss the phenomenon of colour. We begin with a physicist's point of view and explain the electromagnetic nature of light. We then proceed to the physiologist's viewpoint and discuss how the eye and the brain are responsible for the sensation of colour. Next we will discuss how colour images are acquired and reproduced by various methods, all based on the trichromatic nature of the human eye, and how all of these methods and devices have nonlinear response. Finally, we discuss a number of colour spaces which are used for colour representation, along with their advantages and disadvantages. We conclude the chapter with a brief discussion regarding the colour spaces that we chose for our proposed techniques of image indexing and retrieval.
2.1 Colour

Colour is a sensation created in response to excitation of our visual system to electromagnetic radiation known as light. Visible light, which exists in only a small segment of the full electromagnetic spectrum spanning from $10^{-6}$ nm to $10^{12}$ nm, is comprised of electromagnetic radiation with wavelength in the range 380 nm to 780 nm. Figure 2.1 depicts the different types of electromagnetic radiation which exist and the location of visible light within the spectrum. Thus, only a tiny fraction of the entire electromagnetic spectrum is responsible for human colour vision.

The sensation of sight is due to light entering the eyes. Light reflecting from surfaces is what allows us to see objects and colours. Which particular wavelengths are reflected determine the colour of a given object. Hence, roses are seen as red and violets as blue. Unlike mass or length, however, colour is not a physical property which can be measured. All objects are colourless. It is the presence of light and our visual system's response to this light which produces colour; colour blind people, for example, can "see" the world but do not see "correct" colour, due to an anomaly of the photoreceptors of their visual system.

2.1.1 Human Visual Pathway

Colour is a sensation which is comprised of highly complex interactions between several physical, neural, and cognitive phenomena [10, 11], which are an integral part of the visual pathway. Light enters the eye through the pupil and hits the rear part of the eye, known as the retina. It can be thought of as the "film" of our visual system, since the image which is viewed is formed on this surface. The retina is comprised of photo-receptive cells which, among other tasks, are responsible for colour vision. These photo-receptive cells are of two types, known as rods and cones [12, 13].

Rods, which number about 120 million, are extremely sensitive to light and primarily come into use for low-light situations such as night vision or, more formally, scotopic vision.
2.1. COLOUR

Cones, on the other hand, number around 7 million and are less sensitive to light. They are used primarily for photopic vision or day-vision: situations where there are high levels of light. There are 3 types of cones: short (S), medium (M) and long (L), referring to the wavelength of visible light to which they are each most sensitive. Specifically, the sensitivity of S cones peaks at about 400 nm (blue), M cones at 545 nm (green) and L cones peak approximately at 570 nm (red). Consequently, since cones are responsible for colour sensation and only work in high levels of light, humans cannot see colour in low light situations. Outside of the 380 – 780 nm spectral range, cone absorption becomes minimal, and thus governs the spectral range of visible light. Figure 2.3 shows an image of the surface of the retina, as it looks through an ophthalmoscope. Also, an illustration showing the histological layers of the retina is shown, where the shape and arrangement of rods and cones can be seen.

When light hits the photo-receptors (rods and cones), neural impulses are generated that pass through the optic nerve to the visual cortex, a region in the central rear portion of the brain, as confirmed by Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET). It is here that the neural signals are processed and “vision” takes place. Figure 2.4 is an axial MRI image of the human skull, clearly showing the eye, optic nerve, and human brain. The visual cortex, is

© Dimitrios Androutsos, 1999
responsible for all aspects of vision, and certain regions are strictly responsible for colour. In fact, approximately 80 - 90% of the visual pathway is responsible for colour vision [14].

Figure 2.4: An axial Magnetic Resonance image of the human brain depicting the eye, the optic nerve and the location of the visual cortex.

2.1.2 Trichromacy Theory

By virtue of the fact that there are 3 types of spectral absorption cones, all colours which humans can see can be created by combining three primary colours which correspond to the absorption of each cone type [15]. This is the basis of the Trichromacy Theory, first postulated by Young [16] and Helmholtz [17, 18] using the colour experiments of Maxwell [19]. As mentioned above, the peak cone absorptions occur at wavelength locations corresponding to the colours red, green, and blue. Figure 2.5 shows approximate cone absorption curves across the visible spectrum.

Thus, all colours can be created by combining different intensities of 3 light sources which emit light of these three corresponding wavelengths\(^1\). A colour which is obtained by mixing the primaries, is identical to a light source producing the same colour. For this reason, the trichromacy theory is the basis of all imaging devices, in both acquisition and reproduction.

2.2 Colour Images

The task of capturing and reproducing the scenes that we see has been an important aspect of human life; this is evident as far back as neolithic times where early man hand painted scenes on cave walls.

\(^1\)Actually, any 3 colours can be used as primaries, however red, green and blue mix to provide the largest set of reproducible colours.
2.2. COLOUR IMAGES

Figure 2.5: Approximate cone absorption curves across the visible spectrum. The S curve peaks in the red, the M in the green and the L curve peaks in the blue.

Of course, the methods and techniques available today are far more advanced and include 35mm Single Lens Reflex (SLR) Cameras, state-of-the-art scanners, CCDs, and laser printers. Nevertheless, hand drawing still exists, but even it can now be done through computers and drawing/painting software.

Imaging devices exploit the trichromacy theory to regulate how much light, of the three primary wavelengths, are absorbed or reflected to produce a desired colour. Of course, there are a number of ways of acquiring and reproducing colour images, the most popular and common of which are:

- **Human Creation/Reproduction/Graphic Arts**: Traditionally, colour images are created using paints, markers, pastels, etc. Of course, all of the colours which are used can be mixed from the three primary colours. Newer techniques are digital-based and replace conventional media with a computer.

- **Photographic Film**: The film which is used by conventional cameras contains 3 emulsion layers, which are sensitive to red, green, and blue light, which enters through the camera lens.

- **Digital Cameras**: Digital cameras use a CCD array to capture image information. Colour information is captured by placing red, green, and blue filters before the CCDs and storing the response to each channel. More advanced techniques implement a filter array which is overlaid on the CCD array during the fabrication stage, as shown in Figure 2.6.

- **Image Scanners**: The most common method of scanning colour images is by using 3 CCDs, each with a filter to capture red, green and blue light reflectance. These three images are then merged to create a copy of the scanned image.
Cathode-Ray Tubes (CRT): CRTs are the display devices used in televisions and monitors. They are comprised of an extremely fine array of phosphors that emit red, green, and blue light at intensities governed by an electron gun in accordance to an image signal. The close proximity of the phosphors and the spatial filtering characteristic of the eye, effectively mix the three emitted primary colours and produce an overall colour.

Colour Printers: Colour printers are the most popular means of attaining a printed copy of a captured colour image. Again, trichromacy theory is implemented here, however, colour in this domain is subtractive, meaning that the amount of the three primaries which appear on the printed media govern how much light is reflected. Thus, to reflect the colour blue, for instance, all other colours except for blue must be subtracted by the light being reflected. The primaries which are used are cyan, magenta, and yellow, and sometimes black.

The information contained in an image is unlimited, as is the interpretation of each image. Moreover, colour information encodes even more information into an image and can provoke further levels of interpretation. Thus, the importance of "accurate" colour reproduction and representation has been an ongoing challenge.

2.3 Gamma Nonlinearity

The variety of ways available for capturing and reproducing colour images has made these tasks much easier and more feasible. This is especially true with today's digital techniques, with which colour images can be created, captured, and printed with the click of a mouse.

In most cases, however, that which is captured and that which is reproduced does not have a perfect colour reproduction or match; in fact, it hardly ever does. Figure 2.7 shows how a colour image can be created by a number of methods and can pass through various devices from capture, to display, to printing.
Colour scenes can be acquired by any number of techniques. Traditional techniques such as photographic film and human drawings, once printed, can be digitized by a scanner. Digital techniques allow for the images to be processed, displayed, stored, printed, and transmitted easily.

For example, a given scene can be captured by a conventional 35mm camera, printed to paper, after which it is digitally scanned, processed, displayed on a computer monitor, and printed on a colour laser printer. Thus, it can pass through any number of different devices. As the image passes from one device to another, however, its data is transformed to overcome the inherent nonlinear response of the various image detectors and recorders that it encounters. This nonlinear response is known as gamma nonlinearity, and is approximated by a power function:

\[ \bar{y} = \bar{x}^\gamma, \]  

where \( \bar{x} \) and \( \bar{y} \) are input and output image data respectively which has been normalized\(^2\) to values in the range \([0, 1]\), and \( \gamma \) is the gamma nonlinearity of the device in question.

As an example, consider a CRT monitor, which produces colours by exciting phosphors through applied voltage. It turns out that the intensity of the light produced by the phosphors, and ultimately

\(^2\)For example, a typical 8-bit input pixel value would be normalized by dividing it by the maximum 8-bit value, i.e., 255.
the colour which is displayed, is a nonlinear function of its input voltage. Specifically, the phosphor intensity is proportional to the input voltage raised to the $\gamma$ power, as in (2.1). This implies that the displayed colour will not look the same as that in the original image data. Figure 2.8 shows two images with different gamma intensity values. Clearly, the colours present in the two images look quite different.

![Figure 2.8: Two identical images with different $\gamma$.](image)

Many imaging hardware manufacturers try to take into account the gamma nonlinearity of their device by precompensating, either by analog circuits or look-up tables, and applying the inverse power function on the image data. This way the inherent hardware nonlinearity will counteract the precompensation and linearize the output. Unfortunately, precompensation is not always built into the hardware and in an uncontrolled environment can be totally unknown.

Furthermore, different devices have different $\gamma$ as do similar devices from different manufacturers. Thus, when an image passes through various devices, some with precompensation and others without, all the power functions combine to produce an overall $\gamma$ value for a given image:

$$\gamma_f = \gamma_1 \cdot \gamma_2 \cdot \gamma_3 \cdots \gamma_n,$$

where $\gamma_f$ is the overall gamma of the image and $\gamma_n$ are the gamma values for the $n$ different imaging devices through which the image passes.

What is required is for the $\gamma_f$ of a given image to be encoded with the image data itself. This way correct compensation can be made at any stage and on any device to provide the most accurate colour reproduction at the final output. Unfortunately, very few of the accepted image standards and formats in use today attempt to incorporate gamma. Moreover, those that are capable of encoding some gamma information are not widely accepted or users do not take advantage of this functionality. Among these formats are the Tagged Image File Format (TIFF), the Run Length Encoded (RLE) format [20] and the new Portable Network Graphics (PNG) format [21, 22] which
addresses gamma nonlinearity to the greatest degree. Until acceptance of such standards grows, gamma nonlinearity will continue to affect colour reproduction.

In particular, for colour image retrieval systems, gamma nonlinearity can greatly affect the retrieval results. It can cause false retrievals and render comparisons and similarity measures between pixel values, and ultimately images, erroneous [23], [24]. For example, a user may request an image which contains a certain desired colour, but due to the nonlinearity introduced by gamma, a valid image may not appear in the top retrieval results. The gamma differences can make two very similar colours appear quite different and exhibit a low similarity measure. This is especially true when using histogram techniques for colour queries [23].

Thus, for good image retrieval performance, a common image representation space would be ideal. Until such a standard is proposed and accepted, techniques have to be created which deal with gamma.

2.4 Colour Spaces

As discussed above, trichromacy theory allows colours to be defined by three primaries: red, green, and blue. Thus, any colour can be uniquely represented by a three dimensional vector in a colour space defined by the three primaries. Therefore, the components of each vector correspond to the amount of each primary required to produce that given colour.

There exist a number of colour spaces which have been developed to represent colours. All of these have their advantages and disadvantages, which ultimately govern their validity and applicability. Unfortunately, there is no "ideal" space which will suffice for every colour application. Some spaces, for example, were developed specifically for applications such as colour television transmission and others for industrial colour applications.

In this section, we will briefly discuss a few of the most popular colour spaces and their properties.

2.4.1 RGB Space

The RGB colour space is, by far, the most common colour representation space in use today. It is a three-dimensional space whose axes are defined by red, green, and blue components. Thus, it forms a colour cube, as shown in Figure 2.9, in which the location of a point (vector) in this cube determines a colour. The diagonal line from \((R = G = B = 0)\) to \((R = G = B = MAX)\), where all three components are equal, is known as the line of greys; It begins at pure black and ends at white.

The primary reasons for the popularity of the RGB space are:
Figure 2.9: A three-dimensional view of the surface of the colour cube in the RGB space for 8-bit colour planes. The three components form a colour cube in which the position of an RGB vector determines its colour.

- it is directly related to, and a result of, the spectral absorption characteristics of cone receptors in the human retina;

- Colour image acquisition and recording hardware is entirely based on this space. Colour scenes are captured using three sensors corresponding to red, green, and blue sensitivity and are stored directly as three colour planes;

Nevertheless, the RGB space has a number of drawbacks which has made many researchers look towards other colour representation spaces. Firstly, the RGB space exhibits high correlation between its components, a fact which can easily be deduced if we look back at the spectral absorption curves in Figure 2.5. For applications such as image compression and transmission, this causes inefficiency since much redundancy exists and extra coding and bandwidth is ultimately required. Secondly, the RGB space has been found to be perceptually non-uniform. This implies that equal differences between RGB values, using a Euclidean distance, do not correspond to an equally perceived difference in colour. In other words, the Euclidean distance between three colours $C_1$, $C_2$, and $C_3$ may be equal, implying that they should be perceived equally similar however, in reality they may not be.

### 2.4.2 Perceptual Colour Space

According to Trichromacy Theory, the human eye exhibits peak sensitivity to the part of the electromagnetic spectrum corresponding to red, blue, and green light. Unfortunately, humans are not
able, solely by looking at a colour, to determine how much of each spectral component, (i.e., red, green, and blue), comprise the colour: \( R = 128, \ G = 34 \) and \( B = 99 \) is not naturally interpreted by humans.

For this reason, perceptual colour spaces were developed using components which correspond to colour attributes which associate easily with how humans describe and perceive colour. These parameters are hue, saturation, and value, (also referred to as intensity, lightness, or brightness).

- **hue**: refers to the “type” of colour, e.g., red, green, yellow, etc. In other words hue corresponds to the dominant wavelength of a given perceived colour stimulus. Hue can be calculated from 8-bit RGB values by:

\[
H_1 = \cos^{-1}\left( \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right),
\]

where \( H = H_1 \) if \( B \leq G \), otherwise \( H = 2\pi - H_1 \);

- **saturation**: refers to the “amount of colour”, “purity”, or “richness” of the perceived colour. A maximally saturated colour contains light only from one certain wavelength. By adding more white light, or effectively additional wavelengths, the perceived colour becomes less saturated. Figure 2.10 depicts a colour whose saturation is varied. Notice that a colour with minimum saturation (zero) is essentially white, since this implies that all other wavelengths are included.

![Figure 2.10: A colour with hue = 0° and value = 100, whose saturation is varied. Notice that as the saturation is decreased the colour appears whiter. Since a maximally saturated colour is essentially one specific wavelength, decreasing saturation involves the addition of other wavelengths in the light which is emitted. A completely unsaturated colour contains all wavelengths and thus appears totally white.](image)

- **intensity** (also referred to as value or lightness): refers to the amount of light which is perceived to be emitted or reflected from a given colour sensation. A colour which has maximum intensity (and minimum saturation) is perceived as white, and that which has minimum intensity is perceived as black. Figure 2.12 shows a colour whose value is varied from minimum to maximum, while hue and saturation are held constant.
\[ V = \frac{\max(R, G, B)}{255}, \] (2.5)

The three components form a colour cone, where hue is a circular component, expressed in degrees, intensity is the vertical axis, and saturation is the horizontal axis. This space is known as the HSV space [25], and is shown in Figure 2.11.

![HSV cone](image)

Figure 2.11: HSV cone

Effectively similar variations of this space are the HSI and HSL space [26, 27]. Essentially, all these spaces are twisted versions of the RGB space, where the line of grey becomes the vertical axis or value, with white at the top. Saturation then, is the distance from the central axis and hue is the position around the axis. All these spaces are really cylindrical, however, they are pictured and treated as cones to accommodate for the perceived change in colour as saturation and value change. At low value, for example, changes in saturation are not perceived and all colours are essentially treated as black.

These perceptual colour space prove to be quite attractive for a number of applications, particularly in colour image analysis. The primary reasons being that the components correspond well with human intuition. Also, it is relatively easy to separate chromatic and achromatic values from each other and colours can be described without reference to intensity [28]. In addition, it has been found that hue can be used for very fast and effective image segmentation. Some hardware manufacturers
2.4. COLOUR SPACES

Figure 2.12: A colour with hue = 0° and saturation = 100, whose value is varied. Notice how the colour appears brighter as it moves from minimum, where it appear black, to maximum value.

have recognized the acceptance of these colour spaces and some modern image acquisition boards are equipped with built-in circuitry for real-time RGB to HSV conversion [29, 30].

Unfortunately, these spaces are also perceptually non-uniform, and simple Euclidean distance measures cannot be used to quantify colour distance. Thus, no effective distance or similarity measure exists to quantify colour distance in these spaces.

2.4.3 Uniform Colour Spaces

The non-uniformity of the RGB space and the perceptual colour spaces, such as the HSV, prompted researchers to find colour spaces where measured differences using a Euclidean distance are perceived as equal by humans. After much research, the Commission International de l’Eclairage (CIE) standardized two approximately uniform colour spaces, the CIELAB (for subtractive colour, i.e., inks, dyes, paints) and the CIELUV (for additive colour, i.e., lights and self-illuminating phosphors). Both these spaces are comprised of 3 components: one achromatic lightness component and two chromatic components.

CIELAB

The CIELAB space is comprised of the three components: Lightness $L^*$, redness-greenness $a^*$ and yellowness-blueness $b^*$, which can be calculated by [31]:

\[
L^* = 116 \left( \frac{Y}{Y_o} \right)^{1/3} - 16, \tag{2.6}
\]

\[
a^* = 500 \left[ f(\frac{X}{X_o}) - f(\frac{Y}{Y_o}) \right], \tag{2.7}
\]

\[
b^* = 200 \left[ f(\frac{Y}{Y_o}) - f(\frac{Z}{Z_o}) \right], \tag{2.8}
\]

where

\[
f(x) = \begin{cases} 
  x^{1/3}, & \text{if } x > 0.008856 \\
  7.787x + \frac{16}{116}, & \text{otherwise}
\end{cases} \tag{2.9}
\]
2.4. COLOUR SPACES

$X$, $Y$, and $Z$ are the CIE tristimulus values [32] and $X_o$, $Y_o$ and $Z_o$ are the values for the reference white (illuminant) selected. Specifically, the $XYZ$ space can be calculated by:

\begin{align}
X &= 0.490R + 0.310G + 0.200B \\
Y &= 0.177R + 0.812G + 0.011B \\
Z &= 0.000R + 0.010G + 0.990B.
\end{align}

This relationship, however, is dependent on the illuminant and the light sources used [32], and can only be used reliably in a controlled environment. For example, European CRTs and North American CRTs use different illuminants and have drastically different reference whites, resulting with CIE tristimulus values which are quite different [28]. In addition, illuminants used in scanners and lighting equipment for photographic equipment vary widely. Unless precise knowledge is known of the lighting, the CIELAB space cannot be effectively used.

CIELUV

The CIELUV space is also comprised of three components. Lightness $L$, which is identically defined as in (2.7), and $u$, and $v$, which are defined as:

\begin{align}
 u^* &= 13L \cdot (u' - u'_o) \\
v^* &= 13L \cdot (v' - v'_o)
\end{align}

where

\begin{align}
 u' &= \frac{4X}{X + 15Y + 3Z} \\
v' &= \frac{9Y}{X + 15Y + 3Z}
\end{align}

Both of these spaces have been applied to colour analysis, particularly segmentation. Due to the fact that a simple Euclidean distance can be used, clustering algorithms which are often used for colour image segmentation, can work much faster.

Unfortunately, these spaces are computationally intensive, and both require an intermediate conversion to CIE $XYZ$ tristimulus components to calculate their final values. This implies dependence on a reference white, illuminants and the viewing conditions as defined by the CIE Standard Observer [33], which, among other factors, takes into consideration viewing angle and visual field. Thus, they are limited to being used under fixed and controlled viewing conditions. Lastly, these spaces are unintuitive, making it difficult for humans to specify or relate to colours.
2.4.4 Other Colour Spaces

Munsell Colour Space

The Munsell colour space is the oldest space dealing with human colour perception. It is the result of subjective observation of human colour matching experiments, instead of measured experimentation [34]. As in the perceptual space discussed above in Section 2.4.2, the Munsell colour space attempts to describe colours by human recognizable attributes of hue (H), chroma (C), corresponding to saturation and value (V). The H component is arranged in a circle around a vertical central axis formed by V, which ranges from black to white. The C component is quantified by the radius of the H circle. The greater the radius, the greater the chroma. At each level of V there is a set of H circles of varying C.

The Munsell space is comprised of a set of 1200 colour chips, each assigned unique H, V, and C, which form a colour atlas [35, 32, 36]. These chips are grouped in such a way that they form a three dimensional solid, somewhat resembling a warped sphere. Figure 2.13(a) shows a slice at a given V showing a number of colour chips and their arrangement. Figure 2.13 shows a slice of the space parallel to the V axis, showing how the chips are arranged along this axis. The importance of the

Figure 2.13: Munsell Colour Space. Munsell used red (R), yellow (Y), green (G), blue (B), and purple (P) as his principal hues, placed them at equal intervals around a circle and then inserted five intermediate hues of yellow-red (YR), green-yellow (GY), blue-green (BG), purple-blue (PB), and red-purple (RP).

Munsell colour space lies in the fact that the colour chips are arranged in such a way that the distances between colours are perceptually equal as determined through experimental data. The space has numerical scales with perceptually uniform steps for each of the three attributes. The Munsell space
allows colour to be referred to in a standard way, and allows for effective communication of colours, rather than relying on vague terms such as “cherry red”. Thus, the Munsell space has become widely accepted in many industrial fields which deal with colour.

However, it has been found that the Munsell space is not as perceptually uniform as originally claimed [37, 38] and thus colour difference cannot be effectively measured using a Euclidean distance, therefore other more complex measures are implemented [28]. Furthermore, there are no closed-form transformations to convert from the RGB space to the Munsell colour space. Conversion from RGB (and other spaces) is achieved through look-up tables and published charts [39, 32].

Another drawback of the Munsell space is the fact that it is based on subtractive colour theory and, since the differences with the two schemes can yield incorrect colour quantification and differences, it cannot be effectively used with additive colour schemes.

**YIQ & YUV colour Spaces**

Two colour spaces which are a direct result of colour television development are the YIQ and YUV colour spaces [40, 41]. YUV was developed and is still being used in PAL and SECAM colour television transmission in Europe and Asia. YIQ is used in the NTSC standard in use primarily in North America. The common component between the two is the luminance component $Y$, defined as:

$$ Y = 0.299R + 0.587G + 0.114B. \quad (2.16) $$

The luminance component is essentially the achromatic signal, which allows for backward compatibility with black & white television; an important factor when colour television was introduced. The other two components are both chrominance quantities, which can be calculated using:

$$ U = -0.147R - 0.289G + 0.437B \quad (2.17) $$

$$ V = 0.615R - 0.515G - 0.100B \quad (2.18) $$

---

3The Munsell colour notation has been standardized by the American National Standards Institute ANSI Z138.2, the Japanese Industrial Standard for Colour JIS Z8721 and the German Standard Colour System DIN 6164.

4Phase Alternating Line

5Sequential Couleur Avec Memoire

6National Television System Committee or “Never The Same Colour”
and

\[ I = 0.596R - 0.274G - 0.322B \]  
(2.20)  
\[ Q = 0.211R - 0.523G + 0.312B. \]  
(2.21)

Studies have shown that the spatial sensitivity of the human eye to chrominance is less than that to luminance. Thus, researchers have exploited this fact when dealing with YIQ and YUV colour spaces. For example, in transmission of television signals, chrominance components are subsampled to save bandwidth. The loss of information resulting from subsampling is perceptually unnoticeable when the signal is decoded.

2.5 Discussion

Without a doubt, colour is a well-researched topic. A complete understanding of human colour vision and perception, however, remains elusive and, to date, is based on mostly experimental and empirical data. Furthermore, as colour imagery moves to the digital domain and image acquisition and reproduction hardware become increasingly varied and widespread, additional challenges emerge. Consequently, there is no accepted means of colour representation. A wide variety of colour spaces have been developed, each with inherent advantages and disadvantages. None of them, however, can be considered "ideal" for all applications. Hence, careful examination of the intended application is required to make a valid selection of a colour representation space.

2.5.1 Image Indexing and Retrieval

Colour is a mechanism that conveys and contains information. Thus, as discussed in Chapter 1, it only seems natural that image retrieval systems would make colour an integral component, if not the most important component.

Unfortunately, extracting prominent and important information from colour images (indexing) can be done in various ways, and each method is dependant on the colour space selected. Moreover, what information is extracted from the images can greatly affect which measures of similarity can be implemented to compare colours between images.

Feature Extraction

Most image retrieval systems base their feature extraction on a colour histogram. The dimensionality of a colour histogram, however, can be very high. For this reason, colour spaces other than the RGB
space are used to build a colour histogram for images. Often, researchers implement spaces such as the YIQ and CIELAB so that the chrominance components can be subsampled, effectively reducing the histogram size.

As will be seen in Chapter 4, our method of colour image retrieval does not implement histograms. In our stage of feature extraction we use the HSV perceptual space to perform colour image segmentation to extract prominent and important colour features. We chose this space because it allows for fast and automated image segmentation with extremely good results. Furthermore, as discussed in Section 2.4.2, the HSV space is essentially a rotated variation of the RGB space, and this provides a better link to the angular distance measure that we implement for colour similarity.

**Colour Similarity**

Comparison of colour histograms becomes difficult due to their inherent high dimensionality. Thus, many researchers compare colour histograms with Euclidean distance metrics, which can be calculated very quickly. Furthermore, the RGB space is rarely used to build colour histograms since it is considered perceptually non-uniform and its components are correlated. Perceptually, however, non-uniformity is a weakness of the RGB space, which is used to give support to the use of Euclidean distances. Similarly, the correlation which exists between RGB components, reduces the efficiency and effectiveness of colour histograms.

Our technique, as mentioned, does not use colour histograms. Instead it uses three dimensional RGB colour vectors and, as will be seen in Chapter 5, our measure of similarity is a vector measure which is based on the angle between two RGB colour vectors. The foundation and theory behind angular-based RGB vector measures lies in colour image filtering, where colour vector comparison is used to determine inclusion or exclusion from the filter output.
Chapter 3

Colour Similarity Measures

RETRIEVAL systems are based on similarity measurements between a given query and indexed data. When dealing with textual retrieval systems, similarity has a well-defined solution; either a word exists in the data or it does not. When we deal with a qualitative image attribute such as colour, however, the solution is not so simple. For image retrieval, extracted colour information is normally stored and indexed in vector form and similarity is determined by direct comparison of these vectors. Some measure of closeness or proximity between colours is thus required, and this proves to be an ill-posed problem. The difficulty lies in the perceptual interpretation of colour and whether the similarity between two colour vectors mimics human similarity response. How close is aquamarine to baby blue? sea green to pine green? Clearly, there is no simple way to quantify perceptual attributes. In addition, since colour is usually expressed as a vector quantity, the number of measures that can be applied greatly increases. While we will be dealing with retrieving images based on their colour similarity in future chapters, knowing how to measure similarity/dissimilarity is of fundamental importance and will be addressed in this chapter.

Colour similarity, or more generally vector similarity, has been an issue of great importance and significance to a number of areas. How close vectors are in relation to each other or to a specific group of vectors, has been at the crux of areas ranging from image filtering to psychology to neural networks. Thus, many different measures have been developed and implemented to quantify similarity between vectors.

Features, in a given space, can be considered similar if they are in close proximity to each other and dissimilar if they are further apart. In order to analyze proximities between features in a given space, a measure is required. These measures can be classified under two categories, namely dissimilarity measures, $d$, and similarity measures, $s$, depending on how they map vector proximity. Normalized measures produce values in the range $[0, 1]$. The interpretation of a value in
this range depends on the measure itself. Some may map vectors which are in close proximity to values near 0, thus quantifying dissimilarity, (i.e., low dissimilarity or high similarity). Alternatively, some measures map vectors which are far apart to values near 0, thus quantifying similarity, (i.e., low similarity or high dissimilarity). Figure 3.1 graphically compares the two types.

We discuss dissimilarity first since the focus of interest and research has usually concentrated on it, primarily because of the inherently easier visualization and representation as points in a given space. Yet, a similarity measure can always be converted to a dissimilarity measure, and vice versa, by applying an appropriate transformation. If \( d \) is the dissimilarity between two vectors, \( s \) is the direct opposite and may be obtained by using any monotonically decreasing function to transform \( d \):

\[
s = T(d).
\]

(3.1)

As an example, the simplest transformation is:

\[
s = 1 - d.
\]

(3.2)

In applications where the measures are used for rank ordering, as in image retrieval, such a simple transformation, such as Equation (3.2), does not violate the data [42, 43]. Of course, these transformations apply in the reverse also, so that similarity can be transformed into dissimilarity.

More complicated transformations, however, can be selected to stress certain ranges of proximities. These transformations can affect the similarity or ranking of two vectors and ultimately influence the interpretation. The choice of a valid measure, along with a proper transformation, is usually closely tied to the intended use. Special attention must be given to the type of data available.

Figure 3.1: Similarity measures map close proximity of vectors to a value of 1, whereas dissimilarity measures map close proximity to 0.

In this chapter, we discuss some well-established and common multidimensional vector measures. Many of these have been applied to image retrieval, while others have not. In addition, we present some new content-based measures, one of which is implemented in our retrieval scheme, in a later chapter. We also discuss how valid transformations can be used to tune a given measure for more valid results. Specifically, we show how we apply a transformation to our proposed distance measure.
3.1 Dissimilarity Measures

Dissimilarity measures are more commonly known as distance measures, since increasing the distance between two given vectors implies increasing dissimilarity, as noted above. All the following discussed measures are generalized for m-D vectors and we assume that two m-D vectors $\tilde{x}_i$ and $\tilde{x}_j$ are available.

* Minkowski Metrics. The most commonly used measure to quantify distance between two vectors is the generalized Minkowski metric ($L_p$ norm), defined as [44]:

$$d_p(i, j) = \left( \sum_{k=1}^{m} |(x_{ik} - x_{jk})|^p \right)^{\frac{1}{p}},$$  \hspace{1cm} (3.3)

where $m$ is the dimension of the vector $\tilde{x}_i$ and $x_{ik}$ is the $k^{th}$ element of $\tilde{x}_i$. An important consideration for deciding on an appropriate value of $p$ is the degree of emphasis to be placed on $|(x_{ik} - x_{jk})|$; higher values of $p$ emphasize larger values of the absolute difference to a greater degree, (i.e., dissimilar pairs are exaggerated more).

Three special cases of the $L_p$ metric are of particular interest, namely:

- The City-Block distance or Manhattan Distance ($L_1$ norm) corresponding to $p = 1$. In this case, the distance between the two m-D vectors is considered to be the summation of the absolute values between their components:

$$d_1(\tilde{x}_i, \tilde{x}_j) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|.$$  \hspace{1cm} (3.4)

- The Euclidean distance ($L_2$ norm) corresponding to $p = 2$. In this case, the distance between the two m-D signals is set to be the square root of the summation of the square distances among their components:

$$d_2(\tilde{x}_i, \tilde{x}_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}.$$  \hspace{1cm} (3.5)

- The Chessboard distance ($L_\infty$ norm) corresponding to $p = \infty$. In this case, the distance between the two m-D vectors is considered equal to the maximum distance among their components:

$$d_\infty(\tilde{x}_i, \tilde{x}_j) = \max\{|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, ..., |x_{ik} - x_{jk}|\}.$$  \hspace{1cm} (3.6)
The Minkowski metric discussed above is only one of many possible measures [45]. Other measures can be devised in order to quantify distances among multichannel signals. One example of such a measure is:

* The Canberra Metric defined as follows:

\[
d_c(\bar{x}_i, \bar{x}_j) = \sum_{k=1}^{m} \frac{|x_{ik} - x_{jk}|}{(x_{ik} + x_{jk})},
\]

(3.7)

where \(m\) is the dimension of the vector \(\bar{x}_i\) and \(x_{ik}\) is the \(k^{th}\) element of \(\bar{x}_i\). The summand in (3.7) is defined to be zero if both \(x_{ik}\) and \(x_{jk}\) are zero. This measure applies to \(m\)-D vectors that do not take on negative values, such as colour. The attractiveness of this measure lies in the fact that it incorporates scaling by weighting each difference by \((x_{ik} + x_{jk})\), rendering it very sensitive to small values of \(x_{ik}\) and \(x_{jk}\).

Of course, there are many other measures by which a distance function can be constructed. Depending on the nature of the problem and the constraints imposed by the design, one method may be more appropriate than the other.

### 3.2 Similarity Measures

The problem of defining similarity between two multidimensional signals can also be accomplished by utilizing a symmetric function, whose value is large when two given vectors are similar (as depicted in Figure 3.1). Any non-parametric function, \(s(\bar{x}_i, \bar{x}_j)\), can be used to compare two multichannel signals, \(\bar{x}_i\) and \(\bar{x}_j\), and as discussed in the previous section, a dissimilarity measure can become a similarity measure with an appropriate transformation.

Examples of similarity functions are:

* the normalized inner product defined as:

\[
s_1(\bar{x}_i, \bar{x}_j) = \frac{\bar{x}_i^T \bar{x}_j}{||\bar{x}_i|| ||\bar{x}_j||} = \cos(\theta),
\]

(3.8)

which corresponds to the cosine of the angle between the two vectors \(\bar{x}_i\) and \(\bar{x}_j\). Alternatively, the angle between the two vectors can be used as a measure of their dissimilarity [46], [47]. The vector angular measure, is defined as:

\[
\theta = \cos^{-1}\left(\frac{\bar{x}_i^T \bar{x}_j}{||\bar{x}_i|| ||\bar{x}_j||}\right),
\]

(3.9)
It can be argued that similar vectors have almost parallel orientations and that significantly
different vectors point in different overall directions in a given space. Thus, the angular distance
which quantifies the orientation difference between two colour signals is a meaningful measure
of their similarity. It has been shown that for colour, i.e., in a 3-dimensional vector space, this
argument is upheld and valid [46]. Figure 3.2 depicts the colour variation of a given colour
vector in the RGB space as the angle is varied at 8 points around the central vector. From the
figure we can see that as the angle increases from the central vector, the perceived colour also
changes. For the small angles of 0.05 rad, and 0.10 rad however, the colour is perceptually
the same as the central colour, thus a small neighborhood around a given vector in the RGB
space contains colours that can be considered equivalent.

![Figure 3.2: 6 swatches depicting colour change around a central colour vector Ĉ, with RGB values of (187, 83, 78), due to changes in the angular distance with respect to that vector. The angle is calculated for angles of 0.05, 0.1, 0.15, 0.20, 0.25 and 0.5 rad for 8 points around Ĉ. Each swatch shows the effect at a different point along Ĉ, specifically at 125%, 100%, 75%, 50% and 25% of Ĉ.]

* The Czekanowski coefficient is another measure applicable only to vectors with non-negative
components, defined as follows:

\[
s_2(\vec{x}_i, \vec{x}_j) = \frac{2 \sum_{k=1}^{m} \min(x_{ik}, x_{jk})}{\sum_{k=1}^{p}(x_{ik} + x_{jk})},
\]

(3.10)

Many times, in the literature, the Czekanowski coefficient is defined as a dissimilarity measure, through the transformation of (3.2), as \(1 - s_2(i, j)\).

A generalized similarity measure model which effectively quantifies differences among multichannel signals is the family of *content model* measures. These measures take into consideration both the magnitude and the orientation of each vector signal. The measures discussed thus far utilize only part of the information carried by the vector signal. These generalized measures based on both the magnitude and the orientation of the vectors provide a robust solution to the problem of similarity between two vectors.

The main idea behind the *content model* family of measures is that similarity between two vectors is regarded as the degree of common content in relation to the total content [48, 49, 50, 51]. Therefore, given the common quantity, *commonality* \(C_{ij}\), and the total quantity, *totality* \(T_{ij}\), the similarity between \(\vec{x}_i\) and \(\vec{x}_j\) is defined as:

\[
s(\vec{x}_i, \vec{x}_j) = \frac{C_{ij}}{T_{ij}}.
\]

(3.11)

Based on the general framework of Equation (3.11), different similarity measures can be obtained by utilizing different interpretations of commonality and totality. Assume that given the two input signals \(\vec{x}_i\) and \(\vec{x}_j\), the angle between them is \(\theta\) and their magnitudes are \(|\vec{x}_i|\) and \(|\vec{x}_j|\) respectively.

Based on these elements,

* commonality can be defined as the sum of the projections of one vector on the other, and totality as the sum of their magnitudes [52]. The similarity measure can then be written as:

\[
s_3(\vec{x}_i, \vec{x}_j) = \frac{h_i + h_j}{|\vec{x}_i| + |\vec{x}_j|} = \frac{|\vec{x}_i| \cos(\theta) + |\vec{x}_j| \cos(\theta)}{|\vec{x}_i| + |\vec{x}_j|} = \cos(\theta),
\]

(3.12)

where \(h_i\) is the projection of \(\vec{x}_i\) on \(\vec{x}_j\), defined as \(h_i = |\vec{x}_i| \cos(\theta)\), as depicted in Figure 3.3. We see that (3.12) is equivalent to (3.8). Thus the inner product can be considered a special case of the *content based* family.

* Another measure results when totality \(T_{ij}\) is defined as the vector sum of the two vectors under consideration. In such a case the similarity measure is defined as:

\[
s_4(\vec{x}_i, \vec{x}_j) = \frac{h_i + h_j}{\sqrt{(|\vec{x}_i|^2 + |\vec{x}_j|^2 + 2|\vec{x}_i||\vec{x}_j| \cos(\theta))}}.
\]

(3.13)
It is also possible to define commonality between two vectors as a vector algebraic sum, rather than a simple sum of their projections. That gives a mathematically lower value of commonality than the one used in the models reported earlier. Using the two totality measures we can comprise a new similarity measure, defined as:

\[
 s_5(\vec{x}_i, \vec{x}_j) = \frac{\sqrt{(h_i^2 + h_j^2 + 2h_i h_j \cos(\theta))}}{|\vec{x}_i| + |\vec{x}_j|},
\]  

or

\[
 s_5(\vec{x}_i, \vec{x}_j) = \frac{\cos(\theta) \sqrt{(|\vec{x}_i|^2 + |\vec{x}_j|^2 + 2|\vec{x}_i||\vec{x}_j| \cos(\theta))}}{|\vec{x}_i| + |\vec{x}_j|}.
\]

Of course, as with dissimilarity measures, a large number of similarity measures can be developed. Their overall behaviour and intended application, however, governs their validity. In general, when measuring similarity of colour, certain properties should be taken into consideration. Two such properties are metricity and gamma nonlinearity resistance, which can have a profound effect on colour similarity calculations.

### 3.3 Metricity

When dealing with distance measures and similarity, the issue of metricity needs to be addressed. Whether a given measure is a metric or not can have profound effects on certain applications. Distance measures, in general, are considered "valid" if they satisfy certain conditions to form a metric space [53]. Yet, when we deal with qualitative or perceptually dependent data, such as colour, the metric requirement is not necessarily an advantage.
3.3. MERTICITY

Recall that a distance, $D$, is a metric if it satisfies the axioms of:

\begin{align*}
\text{minimality : } & \quad D(a, b) \geq D(a, a) = 0, \\
\text{symmetry : } & \quad D(a, b) = D(b, a), \\
\text{triangular inequality : } & \quad D(a, c) \leq D(a, b) + D(b, c).
\end{align*}

(3.16)

It has been found that \textit{minimality} and \textit{symmetry} are appropriate for stimuli such as colour. Naturally, how close \textit{aquamarine} is to \textit{baby blue} should be the same value as \textit{baby blue} to \textit{aquamarine}, and each colour should have a distance value of zero when compared to itself. The axiom of \textit{triangular inequality}, however, is not valid in a perceptual scenario such as colour. To realize this fact, we only need to look as far as the non-transitive nature of the perceived distance of one colour to another. For example, \textit{baby blue} is similar to \textit{aquamarine} and \textit{aquamarine} is similar to \textit{turquoise}, but \textit{baby blue} may be perceptually dissimilar to \textit{turquoise}, as Figure 3.4 can attest. Thus, if $D(a, c)$ and $D(a, b)$ in (3.16) are very small (i.e., very similar), and $D(b, c)$ is very large (i.e., highly dissimilar), triangular inequality can fail.

![Figure 3.4: Colour comparison depicting the non-transitive property of human colour perception. Colour pairs in (a) \textit{baby blue} & \textit{aquamarine} exhibit similarity as do colour pairs (b) \textit{aquamarine} & \textit{turquoise}. However, \textit{baby blue} & \textit{turquoise} exhibit low similarity. Thus, triangular inequality can fail.]

Psychological research in human similarity judgments have demonstrated that human vision is non-metric [54]. In addition, it has been found that distance measures that are robust to outliers, usually do not satisfy the axiom of triangular inequality, and are thus non-metric [55, 56, 57]. Such distance measures are typical in comparison methods which operate on subsets of data, such as those in colour image filtering.
3.4 Gamma Nonlinearity & Similarity

As discussed in Chapter 2, gamma nonlinearity can wreak havoc on colour similarity calculations. Thus, it would be beneficial to find a measure of colour similarity that would be unaffected or resistant to unknown gamma. In other words, we would like to find a measure that retains the same retrieval ranking of candidate images, based on colour, for a wide range of gamma changes.

To this end, we analyzed the discussed measures under varying gamma levels. Specifically, we compared a set of 16 randomly selected colours $C$ and calculated their similarity to a colour $q$, with 8-bit RGB values of (130,164,53). To test the effects of gamma nonlinearity on similarity measurement, we varied the gamma value of $q$ from 0.5 to 3.0, using the gamma power function:

$$\bar{y} = \bar{x}^\gamma,$$  \hspace{1cm} (3.17)

where $\bar{x}$ is the normalized input value and $\bar{y}$ the normalized output value. Equation 3.17 was applied to each of the 8-bit RGB planes uniformly for each gamma value. Similarity was then calculated for each colour in $C$ at each gamma level using the different measures.

Figures 3.5-3.10 show plots of the similarity calculations for each of the discussed measures. In each plot, the abscissa depicts the change in colour of $q$ as $\gamma$ is varied, and the ordinate quantifies similarity. Each of the lines in each plot shows how the similarity of each $C$ to $q$ changes as the gamma of $q$ is varied. Interestingly, the relation of each line with respect to each other is the ranking that each colour would receive. In other words, at each value of $\gamma$, the colour whose graph has the highest position, is the one which is most similar to $q$ at that given $\gamma$ level. Thus, the ensemble of lines and their fluctuations depict the overall stability of a given measure to varying $\gamma$; ultimately depicting the stability of retrieval ranking. Clearly, a measure whose plot exhibits less fluctuations and crossovers, retains a more stable similarity ranking across the $\gamma$ range.

Clearly, the least erratic behavior of all the discussed measures is exhibited by the angular measure in Figure 3.5, which has substantially less crossover points. In Table 3.1 are tabulated the number of crossovers present in each of the graphs in figures 3.5-3.10, where it can be seen that the least number are obtained with the angular measure. These points correspond to a change in the similarity ranking of a given colour, or the retrieval ranking of an image in a database. The angular measure has a much smoother behavior and low variation, resulting in a retrieval ranking order that remains much more stable over the varying gamma levels. This is primarily due to the fact that gamma nonlinearity primarily affects the intensity of a colour, (i.e., its magnitude). Since the angular measure is only concerned with orientation, it proves more resilient to $\gamma$ changes. However, using orientation alone for colour similarity can cause some perceptually erroneous matches, especially in cases where two colour vectors are collinear.
3.4. GAMMA NONLINEARITY & SIMILARITY

Figure 3.5: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the vector angular measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

Figure 3.6: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_1$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.
Figure 3.7: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_2$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

Figure 3.8: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the $L_\infty$ measure. Curves which have a higher position represent higher similarity to the gamma varied colour.
Figure 3.9: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the Canberra measure. Curves which have a higher position represent higher similarity to the gamma varied colour.

Figure 3.10: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the Czekanowski coefficient. Curves which have a higher position represent higher similarity to the gamma varied colour.
3.5. PROPOSED DISTANCE MEASURE

Less erratic behaviour in the presence of gamma nonlinearity is a very desireable property, especially in image retrieval systems. Thus, a measure of colour similarity, to be used in image retrieval, should strive for such gamma insensitivity. From the above investigation, initial indications are that a measure which is based on the angle between vectors to determine colour similarity, is the best choice.

<table>
<thead>
<tr>
<th>crossovers</th>
<th>angle</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>Czekanowski</th>
<th>Canberra</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>45</td>
<td>35</td>
<td>28</td>
<td>65</td>
<td>51</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Number of crossovers present in the plots of Figures 3.5-3.10.

3.5 Proposed Distance Measure

From the discussion of the various measures in the preceding sections, we can summarize certain properties which a distance measure should have when it is to be used for colour similarity:

1. values must be constrained to the range [0, 1],
2. exploit directional information between two vectors, (i.e., vector angle),
3. incorporate some degree of the magnitude of the vector difference,
4. not required to be a metric; can fail triangular inequality axiom,
5. exhibit stability under $\gamma$ nonlinearity variation.

In our system we implement a vector distance measure based on the vector angular measure between two vectors in (3.9). In the RGB space, similar colours have nearly parallel orientations and a small neighbourhood around a given RGB vector can be considered to contain perceptually equivalent colours, as discussed in Section 3.2. Specifically, our distance measure is a combination of the angle between two vectors and the magnitude of the vector difference, defined as [58]:

$$
\delta(\vec{x}_i, \vec{x}_j) = 1 - \left[ 1 - \frac{2}{\pi} \cos^{-1} \left( \frac{\vec{x}_i \cdot \vec{x}_j}{|\vec{x}_i||\vec{x}_j|} \right) \right] \left[ 1 - \frac{|\vec{x}_i - \vec{x}_j|}{\sqrt{3 \cdot 255^2}} \right], \quad (3.18)
$$

where $\vec{x}_i$ and $\vec{x}_j$ are 3-dimensional colour vectors. Since we deal with RGB vectors, we are constrained to one quadrant of the Cartesian space. Thus, the normalization factor of $\frac{2}{\pi}$ in the angle portion is attributed to the fact that the maximum angle which can possibly be attained is $\frac{\pi}{2}$. Also, the
3.5. PROPOSED DISTANCE MEASURE

\( \sqrt{3 \cdot 255^2} \) normalization factor, in the magnitude part of (3.18), is due to the fact that the maximum magnitude difference which can exist is \(|(255, 255, 255)| = \sqrt{3 \cdot 255^2} \). Both normalization factors contribute so that \( \delta \) takes on possible values in the range \([0, 1]\).

Furthermore, 3.18 can be considered as the product of two content model measures. Specifically, the two components of (3.18), angle and magnitude both exhibit the commonality/totality structure of (3.11). As further evidence to this, consider the special case of (3.18):

\[
s_6(\mathbf{x}_i, \mathbf{x}_j) = \left( \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right) \left( 1 - \frac{\|\mathbf{x}_i\| - \|\mathbf{x}_j\|}{\max(\|\mathbf{x}_i\|, \|\mathbf{x}_j\|)} \right),
\]

which is clearly a member of the content model family [59]. This measure uses the inner product instead of the vector angle. The relationship between the two, however, is a simple transformation.

As mentioned, the new distance measure of Equation (3.18), takes into consideration both the angle between two vectors and the magnitude of the vector difference. Both portions contribute to the overall measure, but, when two vectors under consideration are collinear, only the vector difference magnitude is used. This is important since a zero angle between two colour vectors does not necessarily imply that the two colours are identical, (i.e., they may point in the same direction but have quite different lengths).

Also, it is important to note that this new measure is non-metric. Specifically, it does not satisfy the condition of triangular inequality, which as discussed in section 3.3, is an invalid constraint when dealing with a perceptual quantity such as colour.

Upon close observation of (3.18), one can see that it has the form of a distance measure, in that it maps similar vectors to a value of zero and dissimilar colour vectors to 1. This is a design and application consideration, which allows for a proper function to be applied to transform (3.18) into a measure of similarity. The transformation which we apply takes into account human similarity response and in essence "perceptually tunes" (3.18) for colour similarity measurement.

3.5.1 Perceptual Tuning

To perceptually tune our distance measure to mimic human similarity interpretation, we pass the distance values in (3.18) through an exponential membership function:

\[
\beta(\mathbf{x}_i, \mathbf{x}_j) = e^{-\alpha d(\mathbf{x}_i, \mathbf{x}_j)},
\]

where \( \alpha \) is a design parameter. We found through extensive simulation that a value of \( \alpha = 5 \) provided the most effective similarity results. This is due to the fact that after applying (3.20), vector angular differences below a value of 0.1 retained a high weighting in the overall similarity measure. This
can be seen in the similarity function plot in Figure 3.11. This is of great significance since angular differences within the range [0, 0.1] have been found to be perceptually identical [60], as can also be visually attested from Figure 3.2. In addition, with $\alpha = 5$, only "small" magnitude differences have an effect on (3.20). Thus, vectors with relatively large magnitude difference receive a low similarity value. Notice also, in Figure 3.11, that similarity increases as both the angle between two given vectors and the magnitude difference approach zero, which is clearly the ideal case.

![Figure 3.11](image)

Figure 3.11: Plot of the similarity function of Eq. 3.20. It depicts the similarity between vectors for different angles and magnitude differences.

The motivation for (3.20) draws from psychological research in similarity [61, 49]. According to Shephard [62], human response to visual stimuli, separated by a special interval, (e.g., a distance measure), exhibits monotonic behavior approximating a simple exponential decay function. This membership function has also been used in fuzzy vector filter design [63] to determine filter weights, with high success.

Performance under varying gamma nonlinearity shows that our new distance measure is resilient in this respect. Figure 3.12 is a plot, similar to Figures 3.5-3.10, depicting the performance of similarity matching as $\gamma$ varies. Clearly, the overall behaviour is relatively smooth as compared to the other measures discussed and there is relatively low number of crossovers, 18 to be exact, implying that the similarity ranking remains robust as the $\gamma$ varies. This high insensitivity to gamma is to be expected since the new measure is based on the vector angular measure, which exhibited high insensitivity to gamma variation. In fact, the vector angular measure was the most insensitive to gamma variation. As we will see in 6.6, however, the new measure exhibits the highest retrieval rate over varying gamma and is thus a much more robust measure.
Figure 3.12: Plot of the similarity of 16 colours against a colour whose gamma is varied from 0.5 to 3.0, shown on the abscissa. Similarity was calculated using the proposed new measure. Curves which have a higher position represent higher similarity to the gamma varied colour.
Chapter 4

Colour Image Indexing

In this chapter, we turn our attention to colour indexing (often referred to as feature extraction). A colour similarity measure is of limited use unless colour information is available to perform similarity calculations. The task of identifying and storing valid colour information constitutes the task of indexing. Actually, indexing and a valid similarity measure are the main components of any image retrieval system and directly govern its performance. Thus, their importance cannot be stressed enough. Having discussed the properties of colour and measures of similarity, we now discuss indexing.

Consider a typical 24-bit colour RGB image, where 8 bits are allocated for each colour plane. Such an image is capable of containing colours from a palette of 16,777,216, \(2^{24} \times 256 \times 256\), possible colours. Needless to say, it is not necessary to index all of these possible colours. From a practical standpoint, the storage requirement to index the entire 24-bit palette in most cases would be greater than the image itself and would ultimately defeat the purpose of indexing. Perceptually, such precision is superfluous in a similarity-based scheme such as retrieval.

Thus, to make colour image retrieval viable, some a-priori information is needed; important and valid features need to be extracted. Various image analysis techniques can be applied to gather, or extract, this information which is later used for similarity calculation. How the extraction is performed, however, and what exactly is extracted, directly governs what kind of similarity measure should be used, and how flexible and robust the system will be.

If certain criteria are met, the resulting information which will be indexed can provide the required flexibility and robustness. For example, it is very difficult, if not impossible, for humans to visually discern the difference between two very close colour values. Also, humans describe the colour content of an image using terms such as red, dark yellow, or bright green, not RGB values. In addition, they also tend to focus on and remember regions of colour extremum, (i.e., black, white,
4.1. CURRENT TECHNIQUES

bright regions and large regions of colour) in an image. Furthermore, the spatial placement of colour in a given image is of importance, such as if the image contains large colour regions or if the colour is spread across an image.

We begin, in section 4.1, by investigating current techniques of feature extraction which are used to build image indices for retrieval purposes. Namely, the colour histogram and colour segmentation techniques are examined. In section 4.2 we present certain criteria which should be addressed when performing feature extraction to allow for subsequent robust similarity calculations. In section 4.3, we describe our proposed scheme using colour segmentation, which is implemented to extract perceptually relevant colour information that is stored and indexed for our retrieval system. Finally, we discuss the structure of our stored features which comprise each image index in 4.4 and also discuss computational and storage requirements.

4.1 Current Techniques

4.1.1 Colour Histogram

Colour remains the most important low-level feature that is used to build indices for database images. Specifically, the colour histogram remains the most popular index. Consider an RGB colour as a 3-dimensional random variable \( C = \{R, G, B\} \). The colour histogram of a given image can then be defined as being the joint probability of \( R, G, \) and \( B \), scaled by the number of pixels, \( N \), in the image:

\[
H(r, g, b) = N \cdot P[R = r, G = g, B = b].
\]

Of course, instead of the RGB space, other colour space representations can be used to build a histogram such as LUV, HSV, and others.

The colour histogram's attractiveness is due primarily to its simplicity [64, 65, 66, 67]. It can be calculated very quickly as can the similarity between two histograms. In addition, it is invariant to rotation and relatively unaffected by scale change and translation. However, using the colour histogram for indexing has a number of drawbacks:

- Histograms require quantization to reduce their dimensionality. A typical 24-bit colour image generates a histogram with \( 2^{24} \) bins, which can translate to over 16 megabytes of storage space, depending on the size of the given image. However, with quantization comes loss of colour information and there is no set rule as to how much quantization should be done.

- The colour space which is being histogrammed can have a profound effect on the retrieval results and also governs the amount of quantization.
The colour granularity provided by histogram indexing is, in most cases, not necessary. This is especially true when the final observer is a human.

There is no correlation between neighbouring bins; colours which are perceptually similar and are in adjacent bins may end up being treated as dissimilar. Attempts have been made to overcome this by smoothing histograms to spread colour content across neighbouring bins [1], while others have investigated using cumulative histograms and higher order moments of the colour distribution [68].

Colour exclusion is difficult using histogram techniques. A CBIR system requires a means to allow certain colours to be excluded from a user-defined query right from the start, without requiring an additional level of analysis. For example, with colour histograms a simple way to exclude a colour would be to first query the database to find images which contain colour most similar to the exclusion colour and then rank the images in reverse order to find the images that contain colours which are least similar to the exclusion colour. If a more complex type of query is desired, however, a subsequent similarity stage would have to exist to accommodate for any other specified colours.

Histograms can provide erroneous retrieval results in the presence of gamma nonlinearity. In general, an image database can contain images acquired from many unknown sources and can pass through a number of stages from the moment it is captured to the moment it is displayed. This poses a problem. For example, a scene can be captured on photographic film, transferred to paper, and then scanned to digital format where it can be displayed on any computer monitor. These stages introduce a multiplicative nonlinearity due to the gamma nonlinearity of the various equipment [69]. For image retrieval this can cause very poor performance. It can cause false retrievals and render comparisons and similarity measures between pixel values, and ultimately images, erroneous [23, 24].

The histogram captures global colour activity; no spatial information is available. To include spatial information requires each image to be partitioned into n regions and a histogram built for each region [70, 71, 72], which consequently requires n times more storage.

### 4.1.2 Colour Segmentation

Recently, some image retrieval systems have started to move away from histogram techniques and have begun to make use of segmentation to extract and index features [73, 9, 74, 75]. Colour image segmentation is an area which has received a lot of attention and research. Its popularity and effectiveness lies in the fact that the task of colour segmentation is an inherent component of human visual processing [15]. At the earliest stages of human vision, low-level processing naturally and
automatically partitions a perceived scene, without any recourse to information regarding content or context. These extracted colour features are then used in later stages of human vision to build objects which are then identified and classified by the brain.

Thus, colour image segmentation is essentially a low-level task which tries to mimic this synaptic behaviour by partitioning an image into regions of homogeneous colour, or more technically, by classifying pixels into clusters of uniform colour or texture. Segmented colour regions can then be used in a multitude of ways by higher-level processes dealing with recognition and classification in a variety of fields ranging from robot vision to image retrieval [76, 77, 78].

Colour image segmentation techniques can be generally split into two types: pixel-based and region-based, each with their own pros and cons.

Pixel-Based

Pixel-based segmentation is achieved by observing the colour characteristics of each individual pixel. Popular techniques are:

- **histogram thresholding**: A colour histogram is built and peaks are identified, which correspond to pixels of uniform colour [79, 80, 81]. Thresholding of these peaks is performed to actually segment regions. The advantage of this technique is that it can be performed relatively quickly and is not dependent on any variables or optimization criteria. Segmentation is based solely on the number and size of peaks present in the histogram. Consequently, what thresholds are selected can affect the accuracy of the segmentation results. When coupled with some high-level knowledge and post-processing operations, however, these techniques can provide extremely good segmentation results. Yet, results can vary depending on what colour space is chosen to build the histogram, since any space such as RGB, LUV, or HSV can be used [65].

- **colour clustering**: This technique groups, or clusters, colour pixels around seed colours. Candidate pixels are compared to the seed pixels and are assigned, or clustered, to their closest seed. These seed colours are usually predefined or can be recursively updated so that the colour clusters are optimized [82, 45]. Segmentation using clustering has gained popularity due to the number of effective clustering algorithms available, such as the k-means and c-means [83, 84]. The problem lies in the fact that results are highly dependent on proper selection of the seed colours. Furthermore, computational requirements increase greatly as the number of pixels and seeds increase.

Region-Based

Region-based techniques perform segmentation by utilizing both the colour information of each pixel and their spatial relation to other pixels in a given neighbourhood. Popular techniques are:
Region Growing: This technique begins by identifying seed pixels at certain locations in an image. Neighbouring pixels are then analyzed to determine if their colour characteristics match those of the seed [45]. If so, the neighbouring pixel is grouped with the seed and a region begins to grow. Once again, results depend heavily on the proper selection of initial seed pixels, both in their number and location. Furthermore, if a proper similarity criterion is not implemented, regions of different colour may accidentally be merged. Results with this type of technique can be extremely good. This is especially true if a higher-level post-processing stage is included where neighbouring segmented regions may be merged together [85].

Split-and-Merge: Segmentation follows a similar notion as in region growing, except that in this case the image is partitioned into a number of non-overlapping regions. If neighbouring regions satisfy a criterion based on colour information and proximity, then these two regions are merged together. Otherwise the regions are recursively split and the subregions are tested for similarity [86].

Other image segmentation techniques which have been recently developed, include model-based segmentation techniques, which implement Gibbs [87, 88] or Markov random fields [89, 90, 91, 87] to model the statistical and spectral nature of colour in a given area. Finally, many new techniques which have emerged fall under more than one category. These tend to be hybrid techniques which incorporate one or more of the above mentioned schemes [92, 93, 94, 76].

4.2 Retrieval-Specific Criteria

New methods of colour feature extraction which address some of the inherent problems mentioned above in section 4.1.1, have been investigated. Among these new techniques, back-projection and colour-sets [4, 95], correlograms [96, 97], blob-based [74], and colour segmentation (as discussed in Section 4.1.2) are techniques which provide more meaningful feature extraction, thus strengthening the argument against using the colour histogram. Unfortunately, these techniques, and the systems which implement them, only improve on a few of the aforementioned limitations namely, quantization and spatiality. In addition, many retrieval systems fuse histogram and non-histogram techniques together [4, 98, 8].

To arrive at a proper colour indexing scheme, certain criteria should be met by the feature extraction stage to provide for flexible and robust colour image retrieval:

- make use of human perception of colour
- complement the similarity measure implemented
- low storage requirement
allow flexibility in how colours can be queried

allow for spatial colour information to be easily indexed

With this motivation, the next section discusses our proposed colour feature extraction stage which is based on colour segmentation.

4.3 Proposed Feature Extraction Scheme

To extract colour features and build indices into our image database we take into consideration factors such as human colour perception and recall. Humans tend to focus primarily on certain relevant features [9]. More specifically, we tend to focus on and remember bright, saturated colour regions present in an image, and regions of high colour content. Thus, when we wish to describe or to find a desired image, we essentially build a low-level model of the image in question in our mind and compare candidate images to this model.

Thus, a natural scheme to colour indexing would be to segment an image into regions of perceptually prominent colour to allow retrieval of candidate images based on the similarity to the colour of each of these regions. It is important to note that we do not attempt to achieve a scheme which obtains an exact segmentation of colour regions. Rather, we are interested in finding approximate representations of colour regions. Similar reasoning has been used in the Blobworld representation [74, 75], and in the Percentile Blob Similarity algorithm [99], where colour regions are represented by approximate blob-like shapes depicting the overall general shape of a colour region and its location in an image. We follow the same train of thought, while at the same time, taking human colour perception traits into account and identifying relevant colours. Our entire indexing and feature extraction phase is comprised of three stages:

- segmentation
  - identify prominent regions
  - recursive HUE thresholding

- post-processing
  - median filtering
  - morphological processing
  - object detection and removal

- index creation and storage
4.3.1 Recursive Colour Segmentation

For our feature extraction stage we developed a recursive HSV-space, pixel-based, segmentation scheme to extract regions within the image which contain perceptually similar colour. We chose the HSV space for segmentation, due to its proven performance [100, 101, 80, 92]. We specifically opted for a histogram thresholding scheme, which allows for fast and efficient segmentation [102], without dependence on training, seed pixels/regions or predetermined thresholds, such as those required in clustering and growing techniques [82, 45] as discussed previously in section 4.1.2. This proves to be of great significance if database population is to be independent of human intervention so as to be effectively automated.

As discussed in Chapter 2, the HSV-space classifies similar colours under similar hue orientations. Hue is particularly important since it represents colour in a manner which is proven to imitate human colour recognition and recall. Figure 4.1 depicts a typical hue histogram.

![Figure 4.1: Typical HUE histogram.](image)

While it is true that the hue is very important, it cannot be analyzed independently without saturation and value information. Specifically, in our method, we threshold the hue histogram, which is known to contain most of the colour information, while also taking into account saturation and value information.

As value (intensity) and saturation (purity) of a colour changes, so does its perceived stimulus. For example, if value is very low, irrespective of what hue or saturation a colour has, it will be perceived as black. Similarly, a very unsaturated colour with relatively high value, in most subjective cases will be perceived as white. In addition, certain colour characteristics are very noticeable in
any given image. In particular, very bright and saturated colours, tend to stand out and are the
colours which are most easily identified and remembered. This fact has even been exploited by film
companies, such as Kodak and Fuji, who increase the saturation of their colour film to make colours
appear more vivid to better mimic our recall of a photographed scene [10].

Thus, it is quite logical to segment these perceptually relevant regions accordingly, to better
represent what colour regions a human would tend to focus on in an image, and be more in line with
what is remembered.

We identify four regions of relevance within the HSV-cone:

* BRIGHT CHROMATIC
* BLACK
* WHITE
* CHROMATIC

Bright Chromatic Pixels

Our feature extraction begins by first identifying the bright chromatic pixels in an image and seg-
menting their regions. We have found experimentally [103] that these tend to be pixels that have:

\[
\text{value} > 85 \quad \text{and} \quad \text{saturation} \geq 20.
\]

Once the pixels which satisfy this criterion are identified, the hue histogram is built and thresholded
into \( m \) bright colours. From the remaining image pixels, \( \text{saturation} \) and \( \text{value} \) are used to determine
which regions of the image are achromatic, namely \( \text{black} \) and \( \text{white} \).

Black and White Pixels

The complete sensation of perceived white and black, for more exact prediction, should take into
account surrounding colours. In addition, the level of illumination in a scene can also affect the
sensitivity level of human colour vision, a phenomenon known as brightness adaptation [104, 105].
At each level of sensitivity there is a lower brightness level, (known as the black point), below which
all colour stimuli are perceived as black. If, however, the brightness level is held constant, as with
images displayed on a monitor, this lower level remains essentially constant. If we account for colour
constancy, where a colour viewed under different illumination is usually perceived as similar, and
colour memory, where we associate a colour stimuli with our memory of a colour, it can be concluded
that there exist sets of colour stimuli that are perceived as black and white, regardless of brightness
levels and surrounding stimuli.

©Dimitrios Androutsos, 1999
We have found, in the literature [80, 106] and experimentally [107], that black can be classified as pixels with:

\[ \text{value} < 25, \]

and white, which is usually the brightest region in a given scene or image [104], are pixels with:

\[ \text{saturation} < 20 \quad \text{and} \quad \text{value} > 85. \]

Chromatic Pixels

All remaining pixels fall in the chromatic region of the HSV cone. Figure 4.2 depicts the HSV-cone and pictorially identifies the discussed regions.

Figure 4.2: HSV cone depicting BLACK, WHITE, BRIGHT CHROMATIC, and CHROMATIC regions.
When treating the chromatic regions, special attention must be given to the possibility of a wide range of saturation values. Many segmentation researchers have assumed the saturation histogram to be bimodal and others do not include saturation values below a certain threshold [80]. This, however, is not entirely valid. It is generally multi-modal and, by taking this fact into consideration, more accurate colour segmentation can be obtained. Figure 4.3 depicts a typical saturation histogram.

We incorporate this into our segmentation scheme by building the saturation histogram and thresholding it accordingly. If we assume that the saturation histogram exhibits \( p \) peaks, we threshold each of these peaks and calculate the hue histogram for the pixels contained under each peak. The resulting hue histogram is then thresholded to obtain \( h \) colours and the process is repeated again for each of the \( p \) saturation peaks. The entire segmentation process is shown in the flowchart in Figure 4.4 and an incremental segmentation result is depicted in figure 4.3.1.

### 4.3.2 Post-processing

For each thresholded peak, (i.e., segmented colour region), a resulting binary image is produced, which is comprised of all pixel locations containing the extracted colour. Thus, if there are, in total, \( 10 \) extracted colours in a given image, then there are \( 10 \) such binary images, (one for each extracted colour), which when “flattened” or merged together, form the overall segmented image, as shown in Figure 4.6. Within each of these binary images, there can be pixels which resemble impulsive-like noise. In addition, there may be small holes or gaps within regions that should be filled, or small regions which are unnecessary, and can be removed. In general, these binary images need to be
Figure 4.4: Flowchart of segmentation procedure.
4.3. PROPOSED FEATURE EXTRACTION SCHEME

Figure 4.5: Step by step visualization of the segmentation procedure

©Dimitrios Androutsos, 1999
Figure 4.6: 3-D view of segmented colour binary images.
post-processed so that the colour regions they contain exhibit smooth contours with homogeneous appearance without being of relatively tiny size.

To achieve this, we implement a post-processing stage consisting of:

* Median Filtering
* Morphological Processing
* Object Removal

**Median Filtering**

The first step in post-processing is the removal of visually insignificant impulsive noise-like pixels and thin lines by application of a *Median Filter*. The median filter is a nonlinear filter which has found wide use in the filtering of image data which is corrupted by binary noise [78, 108]. Its operation is quite simple: a window \( W \) of size \( N \times N \) is slid across an image and the pixel values within the window are numerically ordered. The pixel located at the center of the filtering window then has its value replaced with the median value of the ordered values. Mathematically, the operation can be expressed as:

\[
I_M(m,n) = \text{median}\{I(m-i,n-j), \text{ where } (i,j) \in W_{N \times N}\},
\]

where \( I \) is the original image, \( I_M \) is the median filtered output image and \( N \) is an odd integer.

For the case of binary images, the median filtering operation can be done extremely quickly since ordering can be replaced with simple counting:

1) slide \( N \times N \) window, \( W \), across unfiltered image \( I \)

2) at location \((m,n)\), count number of nonzero pixels, \( p \), within \( W \)

   IF \( p > \frac{(N^2-1)}{2} \)

   THEN \( I_M(m,n) = 1 \)

   ELSE \( I_M(m,n) = 0 \)

3) repeat for every \((m,n)\) location

We implement a \( 7 \times 7 \) window for our median filter operation, which effectively removes stray pixel points, pixel clusters, and thin lines. Application of the Median Filter as a first stage removes tiny objects and pixels very quickly. Thus, the additional processing stages which follow will be slightly quicker since they will not need to operate on these tiny regions.
Morphological Processing

After median filtering, small holes and gaps within other regions can remain along with tiny regions which are too large to be removed by the median filter. To effectively treat these artifacts, we implement morphological operations. Mathematical morphology uses a *structuring element* to operate directly on the shape of objects in an image [109, 110].

The basic operations, or building blocks, of mathematical morphology are *erosion* and *dilation* [78]. Let \( X \) be an object in a binary image and \( B \) be the structuring element, then erosion can be defined as:

\[
X \ominus B \triangleq \{ z : B \subset X \},
\]

(4.3)

where \( z \) are the number of pixel points in \( X \) that the structuring element is included in \( X \), i.e., \( z \) are all the pixels in the image which, when the origin of \( B \) is positioned there, all the pixels of \( B \) overlap with nonzero pixels of \( X \).

![Figure 4.7: Morphological EROSION, depicting the original object X, the structuring element B and the eroded result.](image)

An example erosion is depicted in Figure 4.7, where it can be easily deduced that erosion is a *reducing* operation; as the name suggests.

Similarly, *dilation* is defined as:

\[
X \oplus B \triangleq \{ z : B \cap X \neq \emptyset \},
\]

(4.4)

which can be expressed as the pixels \( z \) in \( X \) which have nonzero intersection with \( B \), or alternatively, all the points \( z \), where the overlaid structuring element coincides with at least one nonzero pixel. Figure 4.8 depicts a simple dilation operation clearly showing that dilation is an *expansion* operation.

These basic morphological operations can be combined in a number of ways to perform some very powerful and effective transforms. In particular, in our post-processing stage, we implement an *OPENING* followed by a *CLOSING*, using a circular structuring element with a diameter of 7...
4.3. PROPOSED FEATURE EXTRACTION SCHEME

pixels (pictured in Figure 4.9).

Opening, is essentially erosion followed by dilation:

\[ X_O = (X \ominus B) \oplus B. \] (4.5)

It is especially effective for smoothing the contours of all the objects in the images and it also removes small objects and spurs. Closing is the opposite of opening; it is a dilation followed by an erosion:

\[ X_C = (X \circ B) \ominus B. \] (4.6)

It proves to be very effective in filling small gaps or narrow channels. The added attractiveness of these morphological operations is that they do not substantially alter the size of the image objects.

4.3.3 Object Detection and Removal

Following median filtering and morphological processing, there still may remain small objects that are irrelevant. For instance, small regions of relatively minimal area or size are perceptually irrelevant and can be discarded. However, these small objects may have been too large for the first two stages of post-processing. For instance, a small square area of size 10 x 10 pixels, represents less than 0.5% of a 512 x 512 image, and can effectively be considered irrelevant. Such an object, however, would not be removed by our median filter and morphological processing, unless these operations were applied repeatedly and/or their operating window (or structuring element) was considerably increased. Consequently, the resulting shapes would change substantially and their size would also increase.

To effectively deal with such objects requires individual segmented image objects to be detected

![Figure 4.8: Morphological DILATION, depicting the original object X, the structuring element B and the dilated result.](image)

\[ X \circ B \]
4.3. PROPOSED FEATURE EXTRACTION SCHEME

Figure 4.9: Structuring element implemented in the post-processing operations.

and for the irrelevant ones, based on a size parameter, to be removed (or filled) without affecting the others.

To this end, we implement a contour-following [78, 111] algorithm, to trace the boundaries of the objects in the image. The algorithm begins at the top left pixel in the binary image and scans horizontally across until it hits a non-zero transitive pixel (i.e., at an edge point). If this pixel has not been classified, (i.e., found to be on the contour of an object), it is tagged and the contour of the object it belongs to is followed. The nearest neighbouring pixel, in a clock-wise direction, is then found and tagged until the starting pixel is met. Once the contour has been found and tagged, the algorithm continues to scan the image until another non-zero pixel is found. In addition, while the algorithm is scanning across, it keeps track of whether it has entered and exited an object; this allows the algorithm to mark an object as outer or inner, (i.e., existing inside another object). The process continues until the entire image of size \( WIDTH \times HEIGHT \) has been scanned and all objects have been found, traced and, tagged.

Figure 4.10 is a flowchart which depicts the entire algorithm. Figure 4.11 shows a graphical detailed view of the contour following process, depicting the starting pixel and the direction of search. As can be seen, it is important for the algorithm to keep track of whether it is inside an object or not, otherwise inner regions may go undetected.

Once the objects have been identified, the following is performed:

- small objects of area < 100 pixels are removed;
- small inner objects of area < 100, are filled;
- any objects which are nested within an object tagged to be removed or filled, are automatically included in the parent region's operation;
- area and size of remaining objects are calculated and stored;
- shape parameters can also be calculated and stored for each remaining object.
Storage of area and size (i.e., perimeter) information for each object is not necessary for the colour queries of the present system, since spatial colour information is not used. These two parameters, however, can provide a quick indication as to the shape of a given object, such as circularity [24], which can be easily incorporated into the system in the future.

Figure 4.14 depicts a sample image, the thresholded result and the post-processed result. Figure 4.13 shows each thresholded stage and the corresponding post-processed binary image, clearly showing the effects of the post-processing operations.

4.4 Index Creation

The final step, in the feature extraction stage, is to actually build and store the index, to be later used for retrieval purposes. We divide this stage into 3 steps:

- Calculate Representative Vectors
- Categorize Colours
- Build & store Index

4.4.1 Representative Colour Vector

For each image, the segmentation stage extracts $c$ colours, as discussed in Section 4.3.1:

$$c = \sum_{p} h_p + m, \quad (4.7)$$

which is clearly an image dependent quantity. Finally, the average RGB colour of each of the $c$ segmented colours is calculated over the number of pixels which comprise each one. This value is then used as each segmented colour's representative vector:

$$V_i = (R_{av}, G_{av}, B_{av}), \quad \text{where} \ i = 1, \ldots, c. \quad (4.8)$$

Thus, if an image has 10 segmented colours, it can be represented by 10 colour vectors, plus 2 more vectors for black and white content.
4.4.2 Colour Categorization

To aid in the retrieval process, we perform some categorization of the representative vectors. Since we do not implement histograms for comparison and will not be implementing a high-dimensional vector (of dimension higher than 3) for similarity calculation, the placement of the representative vectors within the index (as will be shown in the next section) can allow for more efficient retrieval.

Our motivation for categorization stems from research in colour naming. It has been found, through anthropological, linguistic, and neurophysiological research, that there exist a set of basic colour categories that are common for speakers of any language [112, 113, 114]. These eleven colours are: white, gray, black, red, green, yellow, blue, pink, brown, orange, and purple. These basic colour terms have some important traits [115, 116, 117]:

✓ are single words;
✓ are colours which first come to mind;
✓ are easily learned and recognized (even by children [118]);
✓ translate easily between languages;
✓ do not refer to objects (except for orange).

We utilize the above colour names to coarsely categorize the extracted representative vectors \( V \), to aid in the later stages of querying and similarity calculation. By examining the hue orientation of the representative vectors, we can use some a-priori knowledge to provide a clue as to the vector’s colour name. For example, if the representative vector has a hue of 15°, it can be categorized as red. If it was also found that it belonged to the bright chromatic region, then it can be further tagged as bright chromatic red.

We have partitioned the hue into 5 “slices”, as listed in the table 4.1 and graphically depicted in 4.15. The basic colour terms listed above are 11, but pink, brown, and orange are essentially subsets of the 5 slices we have chosen or can be made by combination of two of the colours in table 4.1. Thus, orange and brown are contained in either yellow or red, and pink within purple or red. Also, we take into account a transition region of 30° at the border of two neighbouring slices.

Taking into consideration the saturation and value ranges for bright chromatic, chromatic, and black and white, we can form 12 solid regions within the HSV cone, namely:

- white
- bright red
- green
- bright blue
- black
- yellow
- bright green
- purple
- red
- bright yellow
- blue
- bright purple
Each representative vector that is calculated, is checked to see which category it falls under. For example, a representative vector with \((R, G, B) = (199, 75, 67)\) has corresponding \((H, S, V) = (3, 66, 78)\). This would categorize it as red and we would add a tag into our index, indicating that this image contains a representative vector in the red range; any query looking for red or similar, would then test the representative vectors within this index with a similarity measure. Similarly, if a representative vector has \((R, G, B) = (231, 142, 84)\) and accordingly, \((H, S, V) = (23, 63, 90)\), it falls within the transition region between the yellow and red slices. Thus, a tag for both these regions is included in the index, signifying that the colour exhibits similarity to both regions.

As can be concluded, we are not attempting to precisely label colours; such a task is an ill-posed problem. Rather, we are attempting to provide an over-complete categorization for a given colour to provide some indication as to what it may be and to what it may be most similar. With this coarse categorization scheme, we can provide a header for each index which provides a good indication as to whether the indexed colours should be used for finer comparison with any query colours. Obviously comparing a green colour with red colour is unnecessary. Hence, if a user was looking for images which contained a certain red, the headers of the index files would be checked to see which images contain representative vectors in the red solid. Those that do not would not be processed any further, saving unnecessary computation. Those indices, however, which do indicate red content would be further processed. Specifically, the red representative vectors and the query colour vector would be accurately compared using a similarity measure.

<table>
<thead>
<tr>
<th>SLICE</th>
<th>HUE Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>([350^\circ, 25^\circ))</td>
</tr>
<tr>
<td>red-yellow</td>
<td>([25^\circ, 45^\circ))</td>
</tr>
<tr>
<td>yellow</td>
<td>([45^\circ, 65^\circ))</td>
</tr>
<tr>
<td>yellow-green</td>
<td>([65^\circ, 85^\circ))</td>
</tr>
<tr>
<td>green</td>
<td>([85^\circ, 160^\circ))</td>
</tr>
<tr>
<td>green-blue</td>
<td>([160^\circ, 180^\circ))</td>
</tr>
<tr>
<td>blue</td>
<td>([180^\circ, 270^\circ))</td>
</tr>
<tr>
<td>blue-purple</td>
<td>([270^\circ, 290^\circ))</td>
</tr>
<tr>
<td>purple</td>
<td>([290^\circ, 330^\circ))</td>
</tr>
<tr>
<td>purple-red</td>
<td>([330^\circ, 350^\circ))</td>
</tr>
</tbody>
</table>

Table 4.1: HUE ranges which are used for categorizing representative vectors. Note that HUE is a circular parameter and the ranges are given in degrees.

### 4.4.3 Index Structure

To make use of the extracted colour features we need to store all information in an index. All operations of searching and similarity will be performed on these indices. The actual database images are only accessed at the final stage of the retrieval system when the retrieval results are to
be displayed.

Our indices are structured as shown in Figure 4.16. As can be seen, the colour category tags act as an information header which gives a quick overview of what colour ranges are contained in a given image. Also, it points to how many representative vectors, \( V \), fall in a given range, and their location within the index itself. This allows for a "quick look" into the colour content and for fast access to the candidate representative vectors, whose similarity to the query vector will be calculated using the similarity measure. Thus, computation time can be reduced and the overall search time is reduced since:

★ indices whose tags indicate no \( V \) in the same category as the query colour are not tested for similarity;

★ sequential searching within an index is eliminated since the tags point directly to the location, within the index, of the candidate \( V \).

### 4.4.4 Computation and Storage Requirements

The presented feature extraction technique was automatically performed without any human interaction on our test database of 1850 24-bit images of \( 512 \times 512 \) resolution (see Appendix A). The database consists of general content images including natural scenes, people, architecture, animals, and plants. Each image index was built using the extracted representative colour vectors, \( V \), as discussed in section 4.4.3.

It is also important to note that the indexing scheme is not dependant on the size of the image. Our image database contains only \( 512 \times 512 \), however, the indexing algorithm works with any size images of any aspect ratio.

#### Computation Time

The feature extraction and indexing algorithm was written entirely in the C programming language. The entire database was indexed on 4 different machines to test computation times on some common computer architectures and processors. The systems which were used were three Intel processor based machines running the Linux operating system. Specifically, the three processors were an Intel Pentium Pro 200 MHz CPU, an Intel Pentium II 350 MHz CPU, and an Intel Pentium II 400 MHz CPU. The fourth machine was a SUN Microsystems ULTRA 10 Creator workstation with a 266 MHz UltraSparc CPU, running on the Solaris 2.51 operating system. Table 4.2 lists the average, maximum and minimum processing times required to index a database image with a specific processor. The values were determined upon indexing the entire image database.
4.4. INDEX CREATION

<table>
<thead>
<tr>
<th>CPU</th>
<th>min</th>
<th>max</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium Pro 200 MHz</td>
<td>0m:41s</td>
<td>1m:37s</td>
<td>1m:15s</td>
</tr>
<tr>
<td>Intel Pentium II 350 MHz</td>
<td>0m:19s</td>
<td>0m:51s</td>
<td>0m:31s</td>
</tr>
<tr>
<td>Intel Pentium II 400 MHz</td>
<td>0m:18s</td>
<td>0m:47s</td>
<td>0m:29s</td>
</tr>
<tr>
<td>SUN UltraSparc 266 MHz</td>
<td>0m:34s</td>
<td>3m:17s</td>
<td>1m:49s</td>
</tr>
</tbody>
</table>

Table 4.2: Processing time for 4 different CPUs. The database was indexed four times, once using each processor, to calculate the minimum, maximum, and average processing times for each image.

As can be seen, the indexing scheme is relatively fast using common processors and systems presently available, especially if we consider the fact that substantial post-processing is performed.

Of course, extremely fast indexing is not a priority for an image database system since indexing is usually done off-line. There is no need for real-time indexing. Typically, images are entered into the system and indices are built to create the database which will ultimately be queried to retrieve images. When new images are added to an existing database, they undergo feature extraction and their indices are added to the existing set.

Quick indexing is still attractive since it will cut down on processing time when a very large set of images is to be indexed, but is not essential for the actual retrieval process. What is important here is a fast and effective retrieval engine that takes the posed query and retrieves candidate images in an acceptable time frame.

Statistical Analysis

A statistical analysis of our entire 1850 image database revealed that the average number of colours extracted was 4.68, (including black and white), and the maximum and minimum number of extracted colours were 16 and 1, respectively. This is a surprisingly small number of colours but, as we will see, retrieval proves to be very effective.

It is also important to note, that by virtue of using colour segmentation, spatial colour information can easily be incorporated into our indices quite easily and efficiently, without having to dramatically increase the index size.

Storage Requirements

Since the number of colour features extracted, using our scheme, varies from image to image, so does the storage requirement of each index. Even so, the resulting indices are relatively small and require at most 1 kilobyte of storage space each. Figure 4.17 depicts the number of bits allocated for each element of the index.

As an example, consider an image that had 16 extracted colours, each comprised of 10 regions. Such an image would require 7,633 bits or 955 bytes. By comparison, an RGB histogram quantized
to 8 bins per plane, (i.e., $8,8,8$), would require 1,152 bytes. An HSV histogram with $(12,4,4)$ quantization would require 675 bytes. Consequently, the histogram indices would require additional storage space for any spatial information, whereas our method already has some spatial information incorporated. Also, as discovered from statistical analysis, the maximum number of colours extracted in our 1850 image database was 16, and 10 colour regions per extracted colour is a very generous estimate. Thus, we can comfortably conclude that our indices are relatively quite small and efficient.
Figure 4.10: Flowchart of the object detection and removal algorithm.

Figure 4.11: Graphical representation of the contour following algorithm operating on a typical binary object which also contains an inner object.
Figure 4.12: Step-by-step view of the post-processing stage. A is the thresholded binary image, B is the median filtered result, C is the morphologically processed result, and D is the final processed binary image after object-removal. The coloured circles draw attention to the typical effects of each post-processing stage.
Figure 4.13: Step by step visualization of the segmentation procedure portraying the thresholded regions and their post-processed result.
Figure 4.14: Sample image and its feature extraction results. (a) Original image, (b) segmented result and (c) the post-processed result.

Figure 4.15: HUE component categorized into 5 regions with their transition regions.
4.4. INDEX CREATION

Figure 4.16: Index structure. The first entry in the index is the total number of colours present in the corresponding image followed by the colour category tags, which give a “quick look” into the colour content of the index. Next, the representative colour vectors are stored starting with their 8-bit RGB values, the total percentage of the colour present in the image, the number of regions or objects containing the colour in question followed by the area, perimeter and percentage of each object.

Figure 4.17: Bit allocation for each entry in the image index.
Chapter 5

Image Retrieval by Colour

Colour image retrieval can be broken into two broadly defined components:

1. identifying important colour features from an image
2. comparing these features to determine their similarity

In chapter 3, we discussed similarity and our proposed measure for colour similarity calculations. Then, in Chapter 4, we discussed feature extraction and image indexing, which extracts the colour information from the database images. The integration of the two stages completes the system and allows for image retrieval to take place. Figure 5.1 depicts the general framework of most image retrieval systems and how the various stages are merged together into an image retrieval system. All systems are comprised of these basic stages, however, they can not be designed independently. How

![Flowchart depicting the general scheme of image retrieval.](image-url)
the feature extraction is performed, along with what is extracted, governs what kind of similarity measure should be implemented. Moreover, the query structure, flexibility and robustness, (i.e., the design of the retrieval algorithm), are directly dependent on both feature extraction and similarity calculation.

In this chapter, we address the issue of “integration” and show how our similarity measure and extracted features are merged together to perform image retrieval. Our system proves to be quite flexible, allowing a user to make various types of colour queries to retrieve images from our test database.

In section 5.1 we briefly discuss some current techniques developed for colour image retrieval, and the inherent limitations and obstacles which exist. In section 5.2 we introduce our retrieval system, at the core of which is the Multidimensional Query Distance Space which we use to determine image similarity and ranking. Our system allows various types of image queries, such as single colour, multiple colour, and query-by-example. In addition, section 5.2.5 shows how easily this space allows for certain colour to be excluded from a query.

5.1 Current Techniques

The most widely used approach to database retrieval has been modeling data and queries as vectors [119, 120]. The primary reason being that some simple measures of similarity can be implemented easily and quickly, such as the \( L_1 \) metric. Typically, for a given query, a query vector is created and compared to all stored data vectors, (i.e., indices), to determine a measure of similarity (see Figure 5.1). These similarity values are then sorted and all those which exhibit similarity below a predefined threshold are retrieved as the most similar. Alternatively, the first \( k \) most similar can be retrieved from the sorted set.

Such techniques developed and evolved initially for text retrieval [121], but have made their way into image retrieval. This is because early colour image indices were nothing more than colour histograms. Consequently, most systems today still subscribe to this type of scheme and are based on high-dimensional vector indexing and similarity.

For example, most systems still implement colour histograms, which are essentially vectors whose dimensionality is determined by the number of histogram bins. The colours of an image are mapped onto a discrete colour space of \( n \) colours. The histogram of these colours has \( n \) bins and forms an \( n \)-dimensional vector:

\[
\mathcal{H}(M) = (h_1, h_2, h_3, \ldots, h_n),
\]

where each component \( h_j \) represents the number of pixels of colour \( j \) in the image \( M \). The total
number of pixels in image $M$ are:

$$N = h_1 + h_2 + h_3 + \ldots + H_n.$$  \hfill (5.2)

### 5.1.1 Similarity Calculation

As mentioned, this representation of colour images allows for relatively fast similarity calculations using simple measures. Specifically, the most popular measures are the Minkowski Metric and the Histogram Intersection.

#### Minkowski Metric

As discussed in Chapter 3, the Minkowski measures are based on (3.3), with the most widely-used cases being when $L = 1, 2$. The distance between two histograms $H$ and $Q$ using an $L_1$-norm is defined as:

$$d(H, Q) = \sum_{i=1}^{n} |q_i - h_i| \leq 2N,$$  \hfill (5.3)

where $2N$ is the maximum attainable value. Similarly, distance with an $L_2$ norm is defined as:

$$d(H, Q) = \sqrt{\sum_{i=1}^{n} |q_i - h_i|^2} \leq \sqrt{2N},$$  \hfill (5.4)

where $\sqrt{2N}$ is the maximum attainable value.

These distance measures can be very fast to compute providing a quick result. However, they do not take into consideration the similarity between histogram bins. To illustrate this shortcoming consider a colour histogram whose bins are ordered in such a way that neighbouring bins contain perceptually similar colour. Application of an $L_1$ or $L_2$ metric can rank similar histograms lower than less similar pairs. Figure 5.2 shows 3 different colour histograms whose bins are assumed to be ordered by colour similarity. Clearly, the most similar pair should be $H_1$ and $H_2$. However, by application of $L_1$ and $L_2$, $d(H_1, H_2) > d(H_2, H_3) = d(H_1, H_3)$, which does not correspond to the perceived similarity.

Cumulative histograms have also been proposed as a more robust index to provide better colour image retrieval and some inter-bin similarity [72]. Since they are completely dense vectors, as opposed to regular colour histograms which are quite sparse, index size and computation time increase considerably.
Figure 5.2: Three histograms arranged so that neighbouring bins are perceptually similar. The calculated similarity of these histograms with the Minkowski metrics do not correspond to their perceived similarity. Specifically, $H_1$ and $H_2$ are determined to be the most dissimilar pairing, for both the $L_1$ and $L_2$ metric, which is clearly not the case. Using these measures we find that $d(H_1, H_2) > d(H_2, H_3) = d(H_1, H_3)$.

### Weighted $L_2$ Metric (QBIC$^{TM}$)

A metric which was developed to somewhat overcome the deficient inter-bin similarity of Minkowski-based measures, is the metric used by the QBIC team [1, 122]. This distance measure implements a symmetric matrix $A$, whose entries correspond to predetermined colour similarity (based on the Munsell colour Space), between bins:

$$d(H, Q) = \sqrt{(H - Q) \cdot A \cdot (H - Q)^T}. \tag{5.5}$$

If $A$ is diagonalized, 5.5 can be reduced to a weighted $L_2$ mean [72]:

$$d(H, Q) = \sqrt{\sum_{i=1}^{n} w_i (q_i - h_{it})^2}. \tag{5.6}$$

where $w_i$ are the eigenvalues of $A$, and $h_{it}$ and $q_{it}$ are the transformed histograms. By setting $w_i = 1$, 5.6 reduces to Equation 5.4.
5.1. CURRENT TECHNIQUES

This measure is a step in the right direction, however, it still suffers from dimensionality problems and also has greater computational requirements than simple histogram techniques since matrix multiplication is required.

Histogram Intersection

Another popular similarity measure for two histograms is the histogram intersection method [64]:

\[ (H, Q) = \frac{\sum_{t=1}^{n} \min(H_t, Q_t)}{\sum_{t=1}^{n} Q_t}, \]  

(5.7)

where \( H \) is the histogram of the database image, \( Q \) is the query histogram, and \( n \) is the number of histogram bins. The numerator of (5.7) is the actual "intersection" operation, which essentially calculates the number of pixels which are common to both histograms. The denominator performs normalization so that (5.7) has a fractional value in the range \([0, 1]\), where a value of 1 is total similarity. One advantage which the histogram intersection has over Minkowski-based measures, is that it only takes into account non-zero bins. Thus, empty bins do not influence the overall similarity of two images; only common colours are compared. Unfortunately, histogram intersection also does not take inter-bin similarity into account. Consequently, the histograms in Figure 5.2, would be erroneously ranked as they were for the Minkowski metrics.

5.1.2 Excluding Colours

All image retrieval systems are designed to retrieve images which contain certain desired colours. No system, however, addresses the issue of excluding certain colours from a query. It may be desired, or required, that the retrieved images do not contain a specified colour, or more generally, do not contain any colours which are perceptually similar to a specified colour.

To achieve such a task with histogram techniques is rather awkward. There are two possible ways that it can be attempted:

1. Provide a secondary stage to filter out retrieved images which contain the exclusion colour(s)
   - Such a method requires additional searching and comparison, resulting in increased computation time. Furthermore, the above-mentioned problems with inter-bin similarity can misclassify an image containing an exclusion colour and ultimately not omit the image from the retrieval results.

2. Omit those histograms whose bins corresponding to the exclusion colour(s), are non-empty
   - The inherent problem with such an exclusion scheme lies in determining which bins should be considered perceptually similar to the exclusion colour. It is not sufficient to just
5.2. SYSTEM INTEGRATION

specify one particular bin. In most cases, it would be required to specify a number of bins. Moreover, this number can not be considered constant for different colours since some colours exhibit perceptual similarity over more bins than other colours. Furthermore, this scheme would not allow the strictness of similarity to be tuned for a given query. For example, some queries might require the exclusion of an exact colour whereas other times a more relaxed constraint will suffice.

5.1.3 Obstacles

Clearly, there are certain aspects of vector similarity techniques using colour histograms that are undesirable. In addition, there are limitations in using colour histograms that should be overcome to allow for a more efficient retrieval algorithm and system. These obstacles can be summarized as follows:

- are extension of text-retrieval schemes;
- vector dimensionality is high;
- quantization limits the flexibility of the system;
- inter-bin similarity is not taken into account without more complex means;
- colour exclusion is not easily and effectively incorporated.

In the following Section, we discuss the integration of our feature extraction stage and our proposed similarity measure to form the retrieval stage. We will show how our system exhibits flexibility in the query structure and similarity ranking, and how it deals with the shortcomings of current high-dimensional vector techniques.

5.2 System Integration

Our retrieval stage is comprised of a number of individual components, namely:

- Query Definition & Input
- QuickLook Vector Creation & Comparison
- Index Parsing
- Similarity Calculation
5.2. SYSTEM INTEGRATION

Image Ranking via the Multidimensional Query Distance Space

Figure 5.3 depicts a high-level flow-chart of the entire retrieval stage. As can be seen, our system begins by specifying a certain colour query, followed by a searching and similarity operation. Then, the Multidimensional Query Distance Space (MQDS) is responsible for using the similarity values to rank the database images to be finally presented to the user or system.

![Flowchart of the retrieval stage.](image)

Figure 5.3: Flowchart depicting the components of the retrieval stage.

5.2.1 Query Definition & Input

The input component or “front-end” to the retrieval stage is where a user specifies what the images desired from the database should contain or “look like”. A user typically interacts with the image retrieval system through a Graphical User Interface (GUI), designed to provide a means to easily and intuitively specify desired image attributes, or more technically, input a query. This query data is then sent into the retrieval system to search the database for candidate images.

To specify a query based on colour, which is our focus, a number of tools can be incorporated into the GUI which will aid the user. Figure 5.4 shows the colour query component of the GUI of our system and Figure 5.5 shows the colour selection tools. The GUI allows for the selection of any number of colours, through one of the following ways:

- direct RGB or HSV component specification;
- colour selection using a Colour Wheel;
quick colour selection from a set of 11 predetermined colours.

specifying a sample or example image to which retrieved images should be similar

![Figure 5.4: Our system’s GUI interface where the query colours are selected.](image)

Each colour selected can also have an approximate desired amount specified. Also, each colour must be tagged to inform the system how each colour should be treated. One of the following tags must be assigned to each query colour:

- **REQUIRED**: Colour **must** exist in the retrieved images.
- **EXCLUDE**: Specified colour **must not** exist in the retrieved images.

By default, the system assumes that the minimum number of colours that must exist in a candidate image are the number of query colours tagged as **REQUIRED**. Any number of colour combinations can be specified with various tags. Furthermore, additional information can be included in each query:

- **NUMBER OF TOTAL COLOURS**: The **total** number of colours that each retrieved image must contain; may be greater or equal to the number of specified colours in a given query, since the minimum number of colours that a retrieved image can have is equal to the number of **REQUIRED** colours. If this is left unspecified, the system assumes that all retrieved images should have at least the minimum number of colours, and no maximum number of colours.

- **AMOUNT OF COLOUR**: An approximate colour amount can be given for each query colour, specified in percentage over the entire image. If this value is left unspecified, the system assumes that the retrieved images can contain any amount of the query colours, regardless of what percentage of the image they cover.

- **NUMBER OF RETRIEVED IMAGES**: The user can also set an upper bound as to how many images should be retrieved. By default, the system assumes that all candidate retrieved images are to be accessible at the output stage, but displays the $\chi$ most similar images by ranking.
Figure 5.5: Our system’s GUI interface showing the tools available for defining a query.
Thus, the system allows for queries to be built from a description such as:

"all images with at least 20% red, between 30% and 40% of green, possibly the presence of blue and no presence of yellow"

### 5.2.2 QuickLook Vector

In Section 4.4.2, we segmented the HSV-cone into colour categories which were used for classification of the extracted representative vectors. This classification was then used to construct a header field for each image index. The header provides a "quick look" into the index, indicating whether the stored representative vectors are "close" to those colours which are queried. Ultimately, the index header determines whether further processing of the index should be performed to calculate a similarity measure and ranking.

For this decision to be made, each index header must be compared to a vector, of equivalent dimension, which contains the colour classification information of the user-defined query. We refer to this vector as the *QuickLook* vector, which is constructed by determining which colour category each query colour belongs to and setting the corresponding bits to 1.

Through binary operations, the *QuickLook* vector can then be quickly compared to the index headers, through binary operations, to determine which indices can be omitted and which should be further processed.

#### Bit Comparison

Since the index header and the *QuickLook* vector are of equal dimension, a simple binary intersection, or *AND* operation, produces an equivalent sized vector, which we call the *QuickMatch Vector*. This resulting binary vector represents the common non-zero bins, (i.e., colours), between the *QuickLook* vector and the index header, as shown in Figure 5.6.

![QuickMatch Vector Diagram](image)

**Figure 5.6**: The intersection of the *QuickLook* vector and the index header produces the *QuickMatch* vector, which indicates which colours are common to both the user query and the index.
If the *QuickMatch* vector is identical to the *QuickLook*, then the image index in question contains representative vectors which belong to the same categories as those colours specified in the query (including exclusion colours). For example, in Figure 5.6, the *QuickLook* vector corresponds to a user query consisting of *black*, *bright red*, a *reddish-yellow* colour, and a *reddish-purple* colour. When binary intersection (bitwise AND) of the *QuickLook* vector with a typical index header is performed, the resulting *QuickMatch* vector indicates that the index in question contains representative colour vectors that may be similar to the query colours; this implies that the image represented by this index may exhibit high colour similarity. The retrieval algorithm can then proceed to parse the index in question, and perform more precise similarity calculations on the contained representative vectors.

It is important to note that the *QuickLook* vector does not encode other information about the query structure, such as if a colour is to be excluded. If the index proves to contain candidate representative vectors, then the next level of similarity calculation will take into consideration such information.

### 5.2.3 Index Parsing

If the system determines that the index in question contains representative vectors which may be similar to the query colours, the index must be parsed to extract the desired representative vectors, to perform the actual similarity calculation.

The information to begin the parsing of the index is contained in the index header. As seen in Figure 4.17, each colour category tag consists of 23 bits:

- **3 bits** for the *number* of representative vectors of this image belonging to this category. This allows for 8 possible representative vectors, however, we place a limit of 4 per colour category.

- **20 bits** allocated to encode each representative vector’s position in the index. Since we limit 4 representative vectors for each colour category, **5 bits** are allocated for position information, (i.e., 32 total representative vectors in an image implying a possible 32 extracted colours); a generous allocation, as concluded from statistical analysis in Section 4.4.4.

To better visualize this process, we once again refer to Figure 5.6. The result for the bit comparison concluded that the image index in question contained representative vectors that may be similar to the query. The bits of the index header are then used to find the location of each representative vector. For example, in Figure 5.6, from the index header we can see that the image contains:

- **black,**

- **1 bright red** representative vector stored in *POSITION 1* of the index,
2 representative vectors from the red category, located at POSITION 7 & POSITION 8,

1 representative vector from the yellow/red transition category, located at POSITION 3,

1 representative vector from the red/purple transition category, located at POSITION 10.

As seen in Figure 4.16, the representative vectors with their accompanying information are stored in sequential order in the index, following the header information, and the white and black information. Thus, to access a certain representative vector, an offset must be calculated. Specifically,

- white information begins at bit location \( b_W = 353 \),
- black information begins at bit location \( b_B = 389 + 42m_W \), where \( m_W \) is the number of white regions; a number stored in the 5 bits starting at the 32\(^{nd} \) bit of the white information,
- representative vectors begin at POSITION 1 starting at bit location \( b_{p1} = 425 + 42(m_W + m_B) \), where \( m_B \) is the number of black regions. All representative vectors stored in higher positions can be accessed at their corresponding bit locations, calculated using the following formula:

\[
b_{p_i} = 389 + 36i + 42(m_W + m_B + m_{i-1} + m_{i-2} + \ldots + m_1), \quad \text{for } i > 1.
\]

### 5.2.4 Colour Similarity Calculation

Having extracted the candidate representative vectors from a given image, the system proceeds to perform a more accurate assessment of similarity via the perceptually-tuned colour similarity measure of (3.20), developed in Section 3.5.

For each of the specified \( n \) RGB query colours \( \vec{q}_1, \vec{q}_2 \ldots \vec{q}_n \), a similarity value \( \beta \) is calculated, using (3.20). Actually, a similarity value is calculated only when a query colour and a representative RGB vector \( \vec{t} \) of an image index, are found to be members of the same colour category. There may exist, however, \( r \) representative vectors in the same colour category as the query colour, i.e., \( \{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_r\} \in \Omega \). For such cases, similarity is calculated for each representative vector and the highest value of the resulting set of \( r \) similarity values, (i.e., most similar), is retained and assigned to the query colour. In general, this is expressed analytically by:

\[
s_j = \max(\beta(\vec{q}_j, \vec{t}_1), \ldots, \beta(\vec{q}_j, \vec{t}_r)), \quad \text{where } j = 1 \ldots n; \quad \{\vec{t}\} \in \Omega_j,
\]

where \( \Omega_j \) is the colour category of the \( j^{th} \) query colour. This process is repeated for each colour specified in the query. The end result is a set of \( n \) similarity values, \( s_1, s_2, \ldots, s_n \), quantifying the
similarity of each of the \( n \) query colours to the indexed representative vectors of a given image. The "fusion" of these similarity values finally determines the overall colour similarity and ultimately the ranking of the image.

Our similarity calculation using RGB colour vectors is very similar to the theory of vector colour image filters. At first glance, the two areas would seem completely unrelated to each other. Surprisingly, however, colour image retrieval and vector colour image filtering are very similar operations. Vector filters calculate a distance between a given vector and RGB colour vectors in a processing window. This distance is then used to order the RGB vectors. Those vectors that are found to be outliers, (i.e., furthest away from the group) are discarded or "filtered out". This is precisely the overall operation of image retrieval, where a measure of similarity is used to order indices, which in turn determine the candidacy of images at the output.

Among the group of vector filters are the Vector Median Filter (VMF) [123], which uses an \( L_1 \) norm to calculate vector distance, and the Generalized Vector Directional Filter [124, 125, 46] which uses the angle between vectors, Equation (3.9), as the measure of similarity. More recent work has expanded on the VMF and GVDF by implementing content-based measures discussed in Section 3.2 and also by implementing adaptive and fuzzy techniques to weight, or tune, each RGB vector's candidacy to the filter output [44, 126, 57, 127].

5.2.5 Image Ranking

Following the similarity calculation, the system has \( n \) similarity values (one per query colour) for each analyzed image index. Each value quantifies the highest similarity that each query colour has to the representative vectors found to exist in the same category. If there is only one query colour then this calculated similarity value can be treated as the colour similarity measure of the overall image. Consequently, when there are multiple query colours, the similarity measures need to be combined in some fashion. A simple way would be to perform a scalar addition of the \( n \) similarity values:

\[
S = s_1 + s_2 + \ldots + s_n, \tag{5.10}
\]

where \( S \) is the overall similarity of the image and \( s_1, s_2, \ldots, s_n \) are the calculated similarity values for each of the query vectors. Unfortunately, such a scalar sum does not provide any indication as to how similar each query colour is to the representative vectors. For example, a scalar sum may show that the overall image similarity is high, while the individual similarity values may indicate otherwise. Consider 2 sets of 3 calculated similarity values \( \{ s_A \} = [1.0, 0.05, 0.05] \), with a scalar sum of \( S_A = 1.1 \), and \( \{ s_B \} = [0.37, 0.36, 0.37] \), with a scalar sum of \( S_B = 1.0 \). If we judge \( \{ s_A \} \) and \( \{ s_B \} \) strictly by their scalar sums, we would conclude that \( \{ s_A \} \) exhibits higher similarity to the query colours than \( \{ s_B \} \), which is clearly false. Without a doubt, \( \{ s_B \} \) exhibits greater overall
similarity to all three query colours, whereas \( \{s_A\} \) exhibits very high similarity to only one of the query colours.

What is required is a means to combine these similarity values together to accurately determine an overall colour similarity for the image in question, which is based on all the specified query colours.

**Multidimensional Query Distance Space**

To this end, we propose a vector representation of the similarity values, which we refer to as the *multidimensional query distance vector* (MQDV) \( \vec{D} \), defined as:

\[
\vec{D}(q_1, \ldots, q_n) = \vec{r} - (s_1, s_2, \ldots, s_n),
\]

where \( \vec{q}_m \) are the \( n \) query colours and \( \{s_1, s_2, \ldots, s_n\} \) are the similarity values of each query colour to the representative vectors, as defined by (5.9), and \( \vec{r} \) is a vector of size \( n \) with all entries set to 1. \( \vec{r} \) essentially converts all the similarity values to distance values, without affecting their ranking\(^1\).

As an example, assume that a query consists of 2 query colours, \( \vec{q}_1 \) and \( \vec{q}_2 \), and a given index contains representative colour vectors, \( \vec{r}_1 \) and \( \vec{r}_2 \), which belong to the same colour category \( \Omega_1 \) as \( \vec{q}_1 \) and \( \vec{r}_3 \), which belong to the same colour category \( \Omega_2 \) as \( \vec{q}_2 \). The maximum similarity \( s_1 \) between \( \vec{q}_1 \) and \( \vec{r}_1 \) and \( \vec{r}_2 \) is taken, along with the similarity \( s_2 \) between \( \vec{q}_2 \) and \( \vec{r}_3 \), to build \( \vec{V}(q_1, q_2) \). The process is repeated for each candidate database index, resulting with \( M \) distance vectors \( \vec{S} \), where \( M \) is the total number of indices upon which similarity was calculated. The maximum value of \( M \) is, of course, the total number of images in the database, \( N \).

This set of \( \vec{D} \) vectors span what we call the *multidimensional query distance space* (MQDS) [103, 128, 129]. For the case of 2 query colours, the space is two-dimensional, for 3 colours it is three-dimensional, etc. Each database image exists at a point in this space and its location can be used to calculate a retrieval ranking for the corresponding image based on the given query. More specifically, the magnitude and orientation of \( \vec{D} \) determines which of the query colours the image index is most similar to, and also the degree of similarity to each of the query colours.

For the special case of single colour query, where only one colour is specified in the system query, the MQDV is a one-dimensional vector and the MQDS reverts to a scalar comparison of this similarity value.

\(^1\)This simple similarity—distance conversion is required because the origin plays an important role in how we determine image ranking, as will be seen shortly.
5.2. SYSTEM INTEGRATION

Equidistant Line

The key to calculating the image ranking lies in the origin of the MQDS and the *equidistant line*. The *equidistant line* is the set of all points in the MQDS where all component values of $\vec{D}$ are equal. Figure 5.7 depicts the MQDS. For example, in a 3-D MQDS, the *equidistant line* passes through $(0,0,0)$ and $(1,1,1)$. In other words, all MQDVs along this line represent image indices whose representative vectors exhibit an equal amount of similarity to their corresponding query colours.

![Equidistant Line Diagram](image)

Figure 5.7: A visualization of the MQDS for 3 query colours, showing the equidistant line.

Clearly then, the database image that is the closest match to *all* the given query colours $q_1, q_2, \ldots, q_n$ is the one whose index produces an MQDV that is collinear with the *equidistant line* and at the same time is closest to the origin, (i.e., has the smallest magnitude). Figure 5.8 provides an intuitive representation to this concept. The location of each tiny image displayed in the two dimensional MQDS, formed by the two query colours $RGB = 26,153,33$ (*green*) and $RGB = 200,7,25$ (*red*), corresponds to each image's calculated $\vec{D}$. How close an image is to each of the axes quantifies how similar that particular image's colour content is to the query colour which is represented by that axis; i.e., images that are closer to the *green* axis contain a colour that is closer in similarity to the *green* query colour, as compared to the similarity that one of its colours exhibits to the *red* query colour.

Thus, to finally rank the images, we need to take into account both the magnitude of $\vec{V}$ and the angle, $\angle \vec{D}$, that it makes with the *equidistant line*. To do this we combine the two values using
Figure 5.8: A visualization of the MQDS for the two dimensional case, i.e., for two query colours. The query colours were $RGB = 26, 153, 33$ (green) and $RGB = 200, 7, 25$ (red). A set of retrieved images are displayed at various points in this 2-D space. Their location represents the point in space where their corresponding $\vec{D}$ exist.

a weighted sum:

\[ R = w_1 |\vec{D}| + w_2 \angle \vec{D}, \]

(5.12)

where lower rank values $R$ imply images with a closer match to all the query colours. The weights $w_1$ and $w_2$ can be adjusted to control which of the two parameters, i.e., magnitude or angle, are to dominate. We have found that values of $w_1 = 0.8$ and $w_2 = 0.2$ give the most robust results. This is to be expected since collinearity with the equidistant line does not necessarily imply high similarity with any query colour. It implies that each query colour is equally close to the indexed colours. As $|\vec{D}| \to 0$, however, closer matches to one or more colours is implied. Thus, a greater emphasis must be placed on the magnitude component.

Colour Exclusion

Our proposed vector approach provides a framework which easily accepts exclusion in the query process. It allows for image queries containing any number of colours to be excluded in addition to including colours in the retrieval results. As discussed above, those images which exhibit high similarity to all the query colours, are those whose corresponding MQDV $\vec{D}$ are collinear with the equidistant line and which have small magnitude. Hence, if an image contains a colour which exhibits
high similarity to a query colour tagged as exclude, this value should affect the overall ranking by decreasing the overall similarity of the image in question; this effectively reduces the image's ranking and places it further away from the top results.

Thus, the exclusion colour's similarity value should affect $\tilde{D}$ accordingly by changing its relation to the equidistant line and the origin. For example, if it is found that an image contains an indexed colour which is close to an exclusion colour, the distance between the two can be used to either pull or push $\tilde{D}$ closer or further to the ideal and accordingly affect the retrieval ranking of the given image.

To this end, we determine the similarity of each exclusion colour with the indexed representative vectors, which exist in the same colour category using (5.9). We call this the exclusion distance vector (EDV):

$$x_j = \max(\beta(\xi_j, \tilde{i}_1), \ldots, \beta(\xi_j, \tilde{i}_r)), \quad \text{where} \quad j = 1 \ldots m; \quad \{\tilde{i}\} \in \Omega_j^X,$$

(5.13)

where $\xi_1, \xi_2, \ldots, \xi_m$ are the $m$ exclusion colours and $\tilde{i}_1, \tilde{i}_2, \ldots, \tilde{i}_r$ are the $r$ indexed representative colours of each database image, which belong to the same colour category $\Omega^X$ as the exclusion colours. As with (5.11) we build a vector comprised of the similarity values of all the exclusion colours:

$$\vec{\Xi}(\xi_1, \xi_2, \ldots, \xi_m) = (x_1, x_2, \ldots, x_m),$$

(5.14)

where $\vec{x}_m$ are the $m$ query colours and $\{x_1, x_2, \ldots, x_n\}$ are the similarity values of each exclusion colour to the representative vectors, as defined by (5.13). The EDV is then merged with the MQDV to form the total query distance vector (TQDV) $\vec{\Delta}$ in a new higher-dimensional space:

$$\vec{\Delta} = [\tilde{D} \quad \vec{\Xi}].$$

(5.15)

The dimensionality of $\vec{\Delta}$ is equal to the number of query colours + number of exclusion colours, as is the new space.

The final retrieval rankings are then determined from the magnitude of $|\vec{\Delta}|$ and the angle which $\vec{\Delta}$ in (5.15) makes with the equidistant line of the query colour space, (i.e., the space spanned by $\vec{\Delta}$), without the exclusion distance vector $\vec{\Xi}$. Figure 5.9 depicts our concept of colour exclusion. Figure 5.9(a) depicts a typical $D$ for a 2 colour query, in its corresponding two dimensional MQDS. Figure 5.9(b) depicts the EDV $\vec{\Xi}$ and the new MQDS formed by including the similarity value of one exclusion colour. As can be seen from this figure, the inclusion of the exclusion vector $\vec{\Xi}$ effectively pulls $\tilde{D}$ away from the equidistant line and at the same time increases the magnitude of the vector, forming $\vec{\Delta}$. Thus, the ranking of an image which contains a representative colour with high similarity.
to the exclusion colour, will substantially increase, as compared to those images that do not contain the exclusion colour.

![Equidistant line](image)

Figure 5.9: A graphical view of the MQDS. (a) Two query colours $q_1$ and $q_2$, the MQDV $\vec{D}$, and the equidistant line. (b) When an exclusion colour is specified, the vector $\Xi$ pulls $\vec{D}$ away from the equidistant line.

### 5.2.6 Query-By-Example

The previous section dealt with user queries based on selecting certain query colours and specifying an amount. Another popular way to query an image database is by selecting an example image and inputting that as a query. All retrieved images are then expected to be similar to the example image. This type of query is known as *query-by-example* and is implemented by most current image retrieval systems. The attractiveness lies in the fact that example images are not necessarily required to be real or natural images; they can also be synthetic or user drawn images. For example, a user can sketch a scene in a designated drawing area and input that image as an example to query the database. Of course, an ideal query-by-example scenario would take into consideration additional image information such as texture, shape, and the spatial relation between objects.

**Colour Cardinality**

Including other features, as mentioned in Chapter 4, can be easily incorporated into our indexing scheme. The main focus of this work, however, is on colour content similarity, thus we approach query-by-example solely from the colour similarity side. We do consider some spatial information about colour in our system when dealing with query-by-example, but we focus on *colour cardinality*. 

©Dimitrios Androutsos, 1999
By this term, we refer to the number of regions of distinct colour within an image and not the amount of each colour.

In histogram retrieval techniques, query-by-example is achieved by determining the similarity of the histogram distribution at a global level. Psychological research, however, has shown that colour, size, and cardinal quantity are the features which are essential to the early stages of human vision[130]. Interestingly, the reaction time to recognition monotonically increases with the number of items present in an image. Motivated by this, we investigated how overall similarity between images is affected as the number of coloured items or regions between two images changes.

We generated a set of test images, each with a varying number of coloured regions, and asked 25 volunteers to compare each image to every other test image and make a decision as to whether the two were similar in cardinal colour quantity. Figure 5.10 shows plots of the tabulated data.

![Figure 5.10: Plot (a) shows the maximum and minimum number of coloured regions required for two images to be perceived as similar. Plot (b) depicts the tolerance, or spread, of the number of coloured regions between two images which are required for the two images to be considered similar solely on cardinal colour quantity.](image)

It was found that as the number of different coloured objects in a given image increased, so did the tolerance for the number of coloured objects required in another image, for the two to be considered similar in colour content. In other words, an image that is comprised of two colours cannot be effectively compared to an image with fifteen different colours. Primarily, only images which contain between 2 and 4 colours in total need to be compared with such an image.

**Similarity Pre-Processing**

As an initial similarity stage, we take this characteristic of human perception into consideration when performing query-by-example. The process starts by performing feature extraction on the example image input into the system, generating the query-by-example index. This index, like all
5.3. QUERY REFINEMENT

database indices, contains all the extracted colours, (or representative vectors), of the image, along with the number of colours present in the example image. Knowing this amount, we make use of the plots in Figure 5.10 to estimate the maximum and minimum number of colours required to make further colour similarity comparisons between the example image and the database images. By extrapolating the tabulated data, the best-fit lines for the maximum and minimum number of colours, \( C_{\text{max}} \) and \( C_{\text{min}} \), respectively, can be calculated by:

\[
\begin{align*}
C_{\text{max}} &= 1.3125 \cdot C_{\text{exact}} \\
C_{\text{min}} &= 0.6875 \cdot C_{\text{exact}}
\end{align*}
\]

given the number of colours, \( C_{\text{exact}} \), in an example image. If the number of colours in a candidate database image falls within the limits of \( C_{\text{max}} \) and \( C_{\text{min}} \), then overall colour similarity can be calculated using the similarity and ranking procedure of Section 5.2.5 discussed above. The block diagram in Figure 5.11 gives a brief overview of the query-by-example process.

![Block diagram](image)

Figure 5.11: Block diagram depicting the steps which our system performs during query-by-example.

5.3 Query Refinement

The primary goal in an image retrieval system is to automate the task of searching for images. Ideally, both the indexing and searching components of CBIR systems should be free of user interaction, except at the query stage, where it is necessary for input to build an image query. Once the retrieval results are presented, however, interaction from the user can once again be used, this time to refine the query results.

Upon viewing candidate retrieval results, or data generated from these results, a user may want to change the query parameters or modify them to possibly improve the output. This interactivity or feedback, has been in use in text retrieval systems for years and is known as relevance feedback [131, 119]. In image retrieval systems, the concept is still relatively new but has received ample attention and some systems have even had it incorporated [132, 133, 134, 135, 136].

©Dimitrios Androutsos, 1999
The importance of refining query results based on user feedback, especially for image retrieval systems, is due to the fact that human subjectivity is, as of yet, an ill-posed problem that cannot be modeled. By using some human feedback, a link is created between high-level human subjectivity and preference, and low-level image features. Furthermore, feedback which is obtained from users can be used by the system to modify future queries, since data collected can lead to a better understanding of a given user's perception or human perception in general. Such important data can eventually lead to the design of a "customization agent" which may work with the retrieval engine to more effectively tailor the retrieval results, based on who the user is.

Overall, the aim is to "move" a given query vector closer towards relevant items and away from irrelevant ones. In our system, we provide for such fine tuning of the retrieval results through refinement of the query vectors, using a technique which we call visual colour grouping.

5.3.1 Visual Colour Grouping

If we take a look at any one of the many available text retrieval systems, such as the powerful search engines of the World Wide Web, (e.g., Lycos\(^2\), Infoseek\(^3\) and AltaVista\(^4\)), we find that all of them have some means of refining the retrieved results. In particular, consider the "refine your search" link which AltaVista provides after an initial query. Figure 5.12 depicts the dialogue box which appears prompting the user to tag whether a word should be excluded, required, or don't care. Of course, the words which are presented are words which were found within the top retrieval results which had high occurrence.

We follow a similar approach to refinement, in that we provide the option for a user to view a set of colours collected from the top retrieval results, tag each colour, and search the results using a new "refined" query vector built from the tagged colours and the initial query colour. Figure 5.13 shows an example dialog window which appears when a user requests to refine the retrieval results. The initial query which was posed specified three colours to search for, depicted at the top of the dialog box in Figure 5.13. Under each query colour, there is a list of 10 colour swatches. Each of these swatches depicts the representative vectors which were found to be the closest match to the corresponding query vector, via the retrieval process of Section 5.2. In addition, next to each colour swatch is listed the measure of similarity calculated by (5.9). The user has the option of selecting which colour swatches to include in the refined search by selecting the corresponding check-boxes.

\(^2\)http://www.lycos.com
\(^3\)http://www.infoseek.com
\(^4\)http://www.altavista.com
Refine your search by requiring a few relevant topics, excluding irrelevant ones, and ignoring the others.

**Figure 5.12:** Dialogue box which appears when search results are to be refined on the AltaVista search site

**Query Vector Refinement**

Once the user has selected which colour swatches to use in the refined search, the system creates a new set of *refined query vectors*, one for each initial query colour. For a given query colour, the corresponding swatches all exist in a small localized neighbourhood of the RGB cube, resulting from the initial similarity calculation performed by the system. In essence, these colours can be thought of as forming a *query cluster*, as shown in Figure 5.14. The system refines the query colours by finding the most centrally located colour vector within this cluster and assigning it as the new refined query vector (colour). This task is achieved by performing a *sum-of-angles* (SOA) operation [125, 47] on the group of colours and retaining the colour with the minimum value. The operation is defined as:

\[
\alpha_i = \sum_{j=1}^{n} A(v_i, v_j), \quad \text{where } i = 1, 2, \ldots n \text{ and } i \neq j, \quad (5.17)
\]

where \(A\) is the angle between two given vectors \(v_i\) and \(v_j\), defined as in Equation (3.9). The variable \(n\) is determined by the number of colour swatches which are selected by the user. For example, in Figure 5.13, for **QUERY COLOUR 1**, the user has checked 4 colour swatches; thus \(n = 5\) in this case, since the initial query colour is also taken into consideration.

For a given colour vector \(v_i\), the *sum-of-angles* operation calculates and sums the angles which
Figure 5.13: Query Refine Dialog Box

$v_i$ makes with all the other vectors $v_j$. Thus, for a set of 5 colour vectors, 4 angles are calculated and added together for each $v_i$. The result is a set of 5 SOA measurements $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$. The most centrally located colour vector is then the one whose corresponding $\alpha$ is the minimum, since this is the colour vector which is closest to all of the others in the cluster. This central vector is then assigned as the new refined query vector $q_R$:

$$q_R = \min(\alpha_1, \alpha_2, \ldots, \alpha_n).$$

(5.18)

The system takes the new refined query vectors and uses them to build a new query. It is not necessary, however, to search the entire database at this point. The system searches and performs similarity calculations and ranking only within the top $X$ retrieved images, since it is known that these images are already similar to the initial query. The value of $X$ is a design parameter which
5.3. QUERY REFINEMENT

Figure 5.14: Scatter plot of the top ten colour swatches, clearly showing the *query clusters*.

can either be set to a predetermined value, or can even be dynamically changed by the system, via a threshold, depending on the results of each query and the rank value calculated by (5.12). Alternatively, the value could even be set by the user to constrain how many images are considered.

5.3.2 Image Relevance

Another way in which the query vectors can be refined, without as much feedback from the user, is by allowing each of the retrieved images to be tagged, without being presented with a set of colour swatches. Upon viewing the retrieved set of candidate images, the user can tag, via a check-box, those images that contain colour content closest to that which the user feels matches their initial query. This is shown in Figure 5.15.

The representative vectors of the selected images, which correspond to the closest match from the initial query, are then used to refine the original query vectors. This operation can be thought of as being the same as the above (via colour swatches), where the user is limited to selecting an *entire row* of swatches instead of being free to tag any swatch, since each row depicts the representative vectors of a retrieved image. Thus, this method can be thought of as a more restricted version of the above. The disadvantage of this method of feedback, however, is that it is very difficult for a human to judge a retrieved image solely on colour, without regard to subjective parameters such as mood and appeal. Of course, if other features are also incorporated into the system, user feedback at this level can provide important information.
5.4 Retrieval Summary

This chapter has shown how the components of feature extraction and colour similarity have been merged together to form the overall retrieval system. In addition, a simple, yet practical, graphical user interface has been developed, which presents users with the means to input various colour queries and retrieve candidate images from the underlying database. The interface, shown in Figure 5.16, allows for any number of query colours to be input into the system. It also allows for any number of colours to be excluded from a query, and can easily accept input images for query-by-example. In addition, user feedback regarding retrieval results is also used to refine the initial system query to tune the retrieval results more closely with the user's preference.

The actual retrieval process, once a query is input, is carried out through the *multidimensional query distance space*, which is the core of the retrieval system. It has been shown that this concept has great flexibility in how a query can be structured and how the final ranking of the database images is determined.

In the following chapter, actual results from the proposed system will be obtained and presented. A thorough analysis and comparison will be made to determine and substantiate the performance and viability of the system.
Figure 5.16: The front-end of the image retrieval system showing the results from a sample query.
Chapter 6

Results

In this chapter, sample results are presented from querying the proposed image retrieval system. The results are analyzed, both qualitatively and quantitatively, to determine both the perceptual accuracy of the system and the retrieval rate and effectiveness. For this, perceptual data was collected from 25 human volunteers and was used to determine Query Sets which contain the images which typical users found to best match a given set of queries. Furthermore, we also test the retrieval system under different values of gamma nonlinearity to examine how the retrieval results are affected, and how the overall system performs in the presence of unknown gamma nonlinearity.

In section 6.1 we present the "ideal" human Query Sets and the colour queries to which they correspond. In section 6.2 we discuss the measures of retrieval performance which are commonly used to study retrieval systems. In section 6.3 we present the retrieval results from our system to single-colour and two-colour queries, along with the results using other vector distance measures and also histogram based retrieval schemes. Query-by-example results are then presented in section 6.4 and colour exclusion results in 6.5. Finally, in section 6.6 we present retrieval results under varying levels of gamma nonlinearity and discuss the performance of our system as gamma varies.

6.1 Human Query Sets

Due to the great amount of subjectivity involved in image retrieval, it is impossible to arrive at a standard or ideal way of judging the effectiveness of retrieval results generated by a given colour query. The simplest, and most natural, technique to rate the performance of a given system, is by viewing the results to a given query and judging their validity.

For such a technique to be decisive, however, data from a large sample set of different users and queries and various type of images must be collected and processed. The desired result would be to
generate a set of "standard", or "ideal", retrieval results which contain the images from the database which humans consider the best matches for specific queries. The magnitude and complexity of obtaining such a set is obvious, especially in a non-empirical manner (i.e., computationally).

An effective human query set, however, can be obtained experimentally by using a set of observers. Instead of actually entering queries into the retrieval system and having users determine the validity of the results, we chose an alternative procedure. We asked a set of volunteers to manually search our image database of 1850 images, and select those images which best matched a set of specific queries. Specifics of the procedure important to note are:

- 25 untrained human volunteers participated:
- 5 queries were given: 3 single colour queries and 2 multiple colour queries, comprised of the colours depicted in Figure 6.1:
  1. >25% seagreen
  2. >25% orange
  3. >25% yellow
  4. >25% red & >25% green
  5. >25% red & >25% blue
- volunteers were given an unlimited amount of time to test for similarity and complete the task
- volunteers were instructed to judge images on colour content only.

<table>
<thead>
<tr>
<th>seagreen</th>
<th>orange</th>
<th>yellow</th>
<th>red</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
</table>

Figure 6.1: Colour swatches depicting the five colours which were used to find the human query sets.

Every image which was considered to fit the given query was given a "point" by each volunteer. The top 25 images with the most “points”, (i.e., most selected by the volunteers), were used to comprise the Human Query Sets, which are shown in Figures 6.2(a)-(e).
Figure 6.2: Human Query Sets $Q$ determined from data collected from 25 volunteers. (a) >50% seagreen (b) >25% orange (c) >25% yellow (d) >25% red & >25% green (e) >25% red & >25% blue.
6.2 Retrieval Performance Measures

Having determined the human Query Sets, some established measures can be used to assess the performance of our retrieval system. The first of these measures is retrieval rate, defined as [70]:

\[ R_{i,j} = \frac{N_j}{N_i} \times 100, \]  

(6.1)

where \( N_i \) is the total images in a given query set \( Q \), (i.e., the Human Query Sets), and \( N_j \) are the number of images which appear in the top \( N_i \) retrieval positions which are part of \( Q \). Two other important measures of performance are [137]: recall, which measures the ability of retrieving relevant images, defined as:

\[ R = \frac{\#I_{\text{retrieved}} \in Q}{Q}, \]  

(6.2)

and precision, which measures retrieval accuracy, defined as:

\[ P = \frac{\#I_{\text{retrieved}} \in Q}{N_r}, \]  

(6.3)

where \( N_r \) are the total number of images retrieved. A good system strives for high precision and high recall. In reality, however, as recall increases, precision decreases and vice versa. This inverse relationship is due to the fact that to increase recall, (i.e., retrieve all the images which are part of \( Q \)), a greater number of total images must be retrieved. Consequently, this increase in the number of total retrieved images also means that more images which are not members of \( Q \) will appear in the retrieval results, and thus decrease precision. Similarly, when precision is very high, recall tends to be low since usually a small number of total images are retrieved, of which only a few are members of \( Q \).

Therefore, both precision and recall are usually considered in tandem and are plotted against each other in precision-recall graphs which depict the overall retrieval effectiveness of a given system [24]. Figure 6.3 depicts the typical shape and trend of precision-recall graphs. Curves which are further away represent retrieval systems that have higher performance.

6.3 Specific Colour Queries

6.3.1 Single Colour Query

Our first test of the system deals with single colour query, (i.e., where only one colour is specified). Specifically, we tested the three queries \( \Theta \), \( \Phi \), and \( \Theta \) listed above in Section 6.1. The system was
queried to find the top 25 images which contained:

1. >20% seagreen
2. >25% orange
3. >25% yellow

Retrieval proceeded as described in Section 5.2. Our retrieval system was tested using our proposed new measure of similarity (3.20) and also using a number of other vector measures discussed in Chapter 3.

In addition, we have also included results using histogram indexing and retrieval techniques to compare against our retrieval scheme. Specifically, we tested RGB and HSV-space histogram techniques. For this, colour histograms were constructed for all the images in the database. Due to the relatively uniform distribution of the RGB bands, we chose 8 uniform quantization bins for each of the RGB bands. For the HSV colour space histogram, we took the statistical nature of the HSV space into consideration, along with human sensitivity to hue and saturation, and uniformly quantized the hue, saturation, and value histograms into 13,5,5 bins respectively [65]. The similarity metric that we implemented for the actual retrieval was the histogram intersection [64].

Figures 6.4-6.5 show the retrieval result using our new proposed measure along with the retrieval result of the other vector distance measures and the results obtained using the two histogram schemes.

©Dimitrios Androutsos, 1999
Figure 6.4: Retrieval results for Query Q: at least 25% seagreen using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_\infty$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,5,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.
Figure 6.5: Retrieval results for Query Q: at least 25% orange using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_{\infty}$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,5,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.
Figure 6.6: Retrieval results for Query Q: at least 25% yellow using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_\infty$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,5,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.
Qualitative Analysis

All methods returned images that contained colours similar to the query colour however the new measure returned more images which were perceptually more accurate. This was established by comparing the retrieved results with the Human Query Sets, \( Q \), which contain the images which most humans would consider to fit the given query. In our case, comparing Figures 6.4, 6.6 & 6.5 with Figures 6.2(a)-(c), we see that our retrieval scheme implementing our proposed new measure retrieves more images that belong to the Human Query Sets, \( Q \), than the other investigated measures and techniques, for all three queries. Specifically, our measure returned 16 images from the seagreen query set \( Q \) for Query ①, 15 images from the yellow query set for Query ② and 10 images from the orange query set for Query ③. Table 6.1 lists the number of images in the top 25 retrieval results that belong to the Human Query Set for each of the three queries for the 5 vector measures and 2 histogram techniques. The entries in boldface denote the highest number. Overall, we can see that the proposed scheme using the new measure retrieved the most images belonging to the Human Query Sets. Even the use of the other vector measures using the proposed system, in place of the new measure, resulted in more images from the Human Query Sets than the histogram schemes.

Table 6.1: Number of images in the top 25 retrieval results, that belong to the Human Query Set of their corresponding query, for the 5 vector measures and 2 histogram schemes. Entries in boldface denote the highest values.

<table>
<thead>
<tr>
<th>query</th>
<th>( # ) of retrieved images ( \in Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>①</td>
<td>16</td>
</tr>
<tr>
<td>②</td>
<td>15</td>
</tr>
<tr>
<td>③</td>
<td>10</td>
</tr>
</tbody>
</table>

One reason why histogram techniques do not agree as well with human perceived results is due to the global nature of colour histograms. Pixels which may exhibit similarity to the query colour may exist in an image but may be scattered throughout the image, and ultimately go unperceived. With our indexing scheme however, colour is extracted via regions and not pixel-by-pixel. Consequently, homogeneous regions are what a human observer would ultimately identify as containing a certain colour.

Quantitative Analysis

For a less empirical interpretation of the retrieval results, precision and recall were calculated and plotted for the three queries and the discussed measures and schemes. Figures 6.7 depict the precision-recall graphs for our system using the new measure and also for the \( L_1 \), \( L_2 \), and \( L_\infty \) norms. Figure 6.8 depicts the precision-recall graphs for the two discussed histogram-based retrieval
schemes. As can clearly be seen, our retrieval scheme using the new proposed measure exhibits the highest performance, in terms of retrieval effectiveness, over all the other vector-based measures. In addition, it provides higher performance than the colour histogram retrieval schemes.

![Figure 6.7: Precision-recall graphs depicting retrieval performance for the vector-based measures for (a) Query ⃣ seagreen, Query ⃣ orange and Query ⃣ yellow.]

**6.3.2 Multiple Colour Query**

Next we tested the system with multiple colour queries. Specifically, the system was tested with two-colour queries corresponding to the two queries ⃣ and ⃣ listed above in Section 6.1; the system was queried to find the top 25 images which contained:

- ⃣ >25% red & >25% green
- ⃣ >25% red & >25% blue

Retrieval proceeded as described in Section 5.2. Our retrieval system was tested using our proposed new measure of similarity (3.20) and also using a number of other vector measures discussed in

©Dimitrios Androutsos, 1999
Figure 6.8: Precision-recall graphs depicting retrieval performance for the histogram-based schemes for (a) Query ⊙ seagreen, (b) Query ⊙ orange and (c) Query ⊙ yellow.
Chapter 3.

Once again, results using histogram indexing and retrieval techniques are also included to compare against our retrieval scheme, using the same colour histograms as those generated in section 6.3.1.

Figures 6.9 and 6.10 show the retrieval result using our new proposed measure along with the retrieval results of the other vector distance measures and the results obtained using the two histogram schemes.

Qualitative Analysis

As in the single-colour results, all methods returned images that contained colours similar to the two query colours. The new measure, however, returned more images which were more perceptually accurate. Once again, this was established by comparing the retrieved results with the Human Query Sets, Q, in Figure 6.2. In our case, comparing Figures 6.9 and 6.10 with Figures 6.2(d), (e), we find that for both queries, our retrieval scheme, implementing our proposed new measure, retrieved more images that belong to the Human Query Sets than the other investigated measures and techniques. Specifically, our measure returned 15 (of 25) images from the red & green query set Q (Figure 6.2(d)) and 14 from the red & blue query set Q (Figure 6.2(e)). Table 6.2 lists the number of images from the Human Query Sets that each investigated measure and scheme retrieved in the top 25 positions, for query Q and Q. The entries in boldface denote the highest number.

Table 6.2: Number of images in the top 25 retrieval results, that belong to the Human Query Set of query Q and Q, for the 5 vector measures and 2 histogram schemes. Entries in boldface denote the highest values.

<table>
<thead>
<tr>
<th>query</th>
<th>new</th>
<th>angle</th>
<th>L_1</th>
<th>L_2</th>
<th>L_∞</th>
<th>RGB_{(8,8)}</th>
<th>RGB_{(13,5,5)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>15</td>
<td>12</td>
<td>11</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Q</td>
<td>14</td>
<td>12</td>
<td>11</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Quantitative Analysis

The precision-recall graphs for these two queries are shown in Figure 6.7, both for the vector measures and for the colour histogram techniques. Once again, it can be seen that our retrieval scheme using the new proposed measure, has a curve which is furthest from the origin. Thus, it exhibits the highest performance over all the other vector-based measures, and over the colour histogram retrieval schemes, in terms of retrieval effectiveness.
Figure 6.9: Retrieval results for Query Q: at least 25% red & green using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_{\infty}$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,3,3)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.
Figure 6.10: Retrieval results for Query Ø: at least 25% red & blue using (a) new measure, (b) angular distance, (c) $L_1$ norm, (d) $L_2$ norm, (e) $L_\infty$ norm and the results using histogram techniques (f) $RGB_{(8,8,8)}$ and (g) $HSV_{(13,3,5)}$. The top 25 images are displayed along with a colour swatch depicting the colour which was used for the query, in the top left position. Similarity is in decreasing order from top left to bottom right, for each set of results.
6.3. SPECIFIC COLOUR QUERIES

Figure 6.11: Precision-recall graphs depicting retrieval performance for two-colour queries. Plots depicted show retrieval effectiveness for vector measures for: (a) Query $\Diamond$; red & green and (b) Query $\heartsuit$; red & blue, and also for colour histogram schemes for (c) Query $\spadesuit$ and (d) Query $\clubsuit$. 
6.3.3 Overall Performance

Table 6.3 lists the retrieval rates $R_{ij}$ for the above mentioned distance measures, as defined by (6.1). Clearly, it can be seen that the angular-based measures exhibit higher retrieval rates, and in particular, our proposed new measure provided the highest retrieval rate over all others, including the histogram techniques. Furthermore, in Figure 6.12 are plotted the retrieval effectiveness graphs of the averaged results of the 5 test queries. Figure 6.12(a) depicts the average retrieval effectiveness for the vector measures, and Figure 6.12(b) shows the plots for the histogram schemes. Figure 6.12(c) plots both the averaged vector and histogram precision-recall graphs together.

Overall, the performance of the proposed new measure using our indexing and retrieval scheme is much higher than conventional colour histogram image retrieval schemes. This is also true even when simple Minkowski metrics are used in place of the new measure. This can be concluded from Figure 6.12(c), where the histogram techniques exhibit curves which are closer to the origin, and consequently of lower performance. As determined from the Human Query Sets, retrieval rate, precision, and recall are highest for the proposed scheme, while at the same time retrieval results are more perceptually accurate.

Table 6.3: Retrieval rate for 5 different vector distance measures and 2 histogram techniques. Entries in boldface denote the highest values. A bar-graph is also included with a graphical view of the retrieval rates.

<table>
<thead>
<tr>
<th>measure</th>
<th>retrieval rate $R_{ij}$ (%)</th>
<th>average $R_{ij}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$new$</td>
<td>64 60 40 60 56</td>
<td>56.00</td>
</tr>
<tr>
<td>$L_1$</td>
<td>44 56 36 44 44</td>
<td>44.80</td>
</tr>
<tr>
<td>$L_2$</td>
<td>48 48 36 48 48</td>
<td>44.80</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>36 40 32 40 36</td>
<td>36.80</td>
</tr>
<tr>
<td>angle</td>
<td>60 40 36 48 48</td>
<td>46.40</td>
</tr>
<tr>
<td>$RGB(8,8,8)$</td>
<td>48 28 28 36 28</td>
<td>33.60</td>
</tr>
<tr>
<td>$HSV(13,5,5)$</td>
<td>28 36 32 48 32</td>
<td>35.20</td>
</tr>
</tbody>
</table>

©Dimitrios Androutsos, 1999
Figure 6.12: Average retrieval effectiveness over the 5 test query cases. Graphs depict the average retrieval effectiveness for (a) vector measures and (b) histogram schemes. (c) Shows all precision-recall plots on the same graph.
6.3.4 Comparison to QBIC

Sections 6.3.1 and 6.3.2 dealt with specific colour queries and compared results using different measures. For a more practical comparison of retrieval efficiency, we compare retrieval results using our system and measure with IBM's QBIC (Query By Image Content) System. We obtained the QBIC engine from IBM's web page\textsuperscript{1} and installed it on a SUN Microsystems ULTRA 10 Creator Workstation. Our entire image database was indexed by QBIC and retrieval results were obtained for the five queries in sections 6.3.1 and 6.3.2. Figure 6.13 shows the retrieval results that QBIC produced for queries 1-5 and Table 6.4 lists the retrieval rate for the QBIC system on our image database, and also includes the data from Table 6.3 in Section 6.3.3 for complete comparison. As can be seen, the new measure and system exhibits better retrieval rate overall and even surpasses QBIC in 4 of the 5 queries. QBIC exhibited a slightly better retrieval rate only with query 3. Figure 6.14 depicts the retrieval effectiveness of QBIC on our image database and compares it with our new measure and system in Figure 6.14(a) and then with all the investigated measures and histogram techniques, in Figure 6.14(b).

Clearly, the QBIC system does not provide retrieval effectiveness as high as the new measure and system, however, it does provide better results than strict colour histogram indexing and retrieval schemes. The reason for this is that QBIC's retrieval engine is not based solely on colour histograms but rather on a two stage retrieval scheme which uses colour histogram similarity calculations as a final step [1].

6.4 Query-By-Example

Another important type of query is Query-By-Example, where a sample image is input as the query and the system retrieves images which exhibit similar colour content. We tested our system with two sample input images. The retrieval results from the proposed scheme and the new measure are depicted in Figures 6.15 and 4.14. Specifically, Figure 6.15 shows the query results when the beans image of Figure 4.14(a) was used as the input to the system. All retrieved images exhibit colour similarity and also exhibit overall similarity in the number of colours, with the query image. Figure 6.4 depicts another query-by-example result when the example image was the top left image, referred to as candy. Again, we can see that there is overall colour similarity to all the images and specifically, all retrieved images exhibit a perceptually equivalent number of colours.

Query-by-example proves to be an important type of query since it allows not only for a sample image to be input to the system, but also for the user to sketch or draw a scene and use that as the

\textsuperscript{1}http://wwwqbic.almaden.ibm.com/. The QBIC engine is available for download with a free 90 day trial license and is available for AIX, Linux, Solaris, Windows NT/Windows95/98, and Macintosh PPC operating systems.
Figure 6.13: Retrieval results produced by QBIC on the 1850 image database for queries: (a) □, (b) ⊙, (c) ⊘, (d) ⊙, and (e) ⊙.
Figure 6.14: Average retrieval effectiveness over the 5 test query cases. Graphs depict the average retrieval effectiveness for (a) QBIC and the new measure and system. (b) Shows all precision-recall plots for all measures, and QBIC, on the same graph.
Table 6.4: Retrieval rate for 5 different vector distance measures, 2 histogram techniques and the QBIC system. Entries in **boldface** denote the highest values. A bar-graph is also included with a graphical view of the retrieval rates.

<table>
<thead>
<tr>
<th>measure</th>
<th>retrieval rate $R_{1,1}$ (%)</th>
<th>average $R_{1,1}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>64  60  40  60  56</td>
<td>56.00</td>
</tr>
<tr>
<td>$L_1$</td>
<td>44  56  36  44  44</td>
<td>44.80</td>
</tr>
<tr>
<td>$L_2$</td>
<td>48  48  36  48  44</td>
<td>44.80</td>
</tr>
<tr>
<td>$L_{\infty}$</td>
<td>36  40  32  40  36</td>
<td>36.80</td>
</tr>
<tr>
<td>angle</td>
<td>60  40  36  48  48</td>
<td>46.40</td>
</tr>
<tr>
<td>$RGB_{(8,8,8)}$</td>
<td>48  28  28  36  28</td>
<td>33.60</td>
</tr>
<tr>
<td>$HSV_{(13,5,5)}$</td>
<td>28  36  32  48  32</td>
<td>35.20</td>
</tr>
<tr>
<td>QBIC</td>
<td>28  32  44  40  44</td>
<td>37.60</td>
</tr>
</tbody>
</table>

input. Thus, retrievals can be made which are more likely to be more “in tune” with what the user is specifically expecting to retrieve.

### 6.5 Colour Exclusion

We also tested our system for colour exclusion. Specifically, we queried the system for images which contained:

- $c_r > 25\%$ red
- $c_r > 25\%$ green
- exclude yellow

where the colours red, green, and yellow are as defined in Figure 6.1. Selecting to use the same colours as tested in the queries in section 6.3.2, allows us to see how exclusion affects the retrieval results.
Figure 6.15: Query-by-example result using the proposed system and new measure. The top left image, beans, is the input to the system. The similarity of the retrieved images is in decreasing order from top left to bottom right.

Figure 6.16: Query-by-example result using the proposed system and new measure. The top left image, candy is the input to the system. The similarity of the retrieved images is in decreasing order from top left to bottom right.

Figure 6.17(a) depicts the query result when red and green are queried, as defined in Query Q. This is the identical result depicted in Figure 6.9(a). Figure 6.17(b) is identical to (a) except that those images which contain yellow are depicted in grey-scale. These images should be removed from the top retrieval results when yellow is excluded in the query. Figure 6.17(c) shows the query results when the exclusion of yellow is specified. Notice how the grey-scaled images in (b) are completely removed from the top retrieval results, and how all the retrieved images exhibit colours very similar to red and green.

It is important to note that new images exhibiting similarity to red and green, that did not appear in Figure 6.17(a), entered into the top retrieval results. This is due to the fact that the similarity that these new images have to red and green is slightly lower than that which the removed images exhibited. The fact that the removed images contain the exclusion colour yellow, lowers their overall similarity and moves them out of the top retrieval results, while moving other images closer to the top.

In fact, it was found that none of the removed images remained among the top 50 retrieval results: their ranking decreased significantly and all of them ranked among the bottom 27% of the total group of retrieved images. The flexibility of this technique allows any number of colours to be excluded in a given colour query. Furthermore, the amount by which the exclusion vector, $\vec{X}$ of
6.6 Retrieval & Gamma

We have shown in Section 3.4 that the proposed new measure performs robustly to gamma changes, at least for a small 16 colour set. This section tests our scheme and the new measure on our 1850 image database.

For this, each image in the database was indexed, with our technique described in Section 4.3, using gamma values of 0.8 to 2.6 at steps of 0.2. Specifically, for each gamma level, each image had its pixel values altered using the gamma power function of Equation (2.1). The power function was applied to each of the three RGB values of each pixel:

\[ y(i, j, p) = x^\gamma(i, j, p), \quad (6.4) \]

where \( x \) and \( y \) are input and output pixel data at image position \((i, j)\), and \( p \) is the colour plane. This step created new images with gamma value \( \gamma \). Thus, 9 new image databases were created of 1850 images each, resulting in a total of 10 databases, since \( \gamma = 1.0 \) is the original database.
Our indexing technique was then performed on all the new databases, to generate image indices at all the gamma levels. To test the retrieval performance, we posed Query \( Q \) to the system, (i.e., images with \( > 25\% \) seagreen), which, incidentally, was extracted from the bat image reproduced in Figure 6.18.

![Figure 6.18: bat image and the colour sea green.](image)

The query results for our new measure are shown in Figure 6.19, along with the results using the angular measure in Figure 6.20 and the \( L_1 \) and \( L_2 \) norms in 6.21 and 6.22. In addition, we also tested the effects of gamma on the retrieval results when using colour histogram techniques, by building an RGB and HSV histogram for each database image, using the same quantization schemes as in Section 6.3. The retrieval results, at each of the gamma levels, for these two schemes are shown in Figures 6.23 and 6.24.

The figures depict the relative ranking of the top 9 retrieval results in decreasing similarity (left to right) at increasing levels of gamma (top to bottom), using the query colour.

Upon first observation, we can see that our scheme using the proposed new measure in Figure 6.19. retrieves the bat image, from which the query colour seagreen was extracted, in 7 of the 10 gamma-varied queries. Also, at all gamma levels, almost all of the retrieved images exhibited colour distribution that perceptually resembled the query colour. Furthermore, the majority of retrieved images remain among the top 9 matches and their final ranking varies slightly at different gamma levels. This is of great benefit in a database scheme since the goal is to retrieve a set of best matches and not necessarily an exact match. The results when using the angular measure, shown in Figure 6.20, also exhibit high similarity to the query colour, however, more images were returned that did not resemble the query colour. Specifically, many images appeared that had a yellow content. The
Figure 6.19: Top 9 retrieval results, using the proposed system and new measure, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query ∅.
Figure 6.20: Top 9 retrieval results, using the proposed system and the *angular* measure, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% *seagreen*, as in Query Q.
Figure 6.21: Top 9 retrieval results, using the proposed system and the $L_1$ norm, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query ⊙.
Figure 6.22: Top 9 retrieval results, using the proposed system and the $L_2$ norm, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% *seagreen*, as in Query ⊙.
Figure 6.23: Top 9 retrieval results, using the RGB colour histogram with (8, 8, 8) quantization, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query ①.
Figure 6.24: Top 9 retrieval results, using the RGB colour histogram with (13, 5, 5) quantization, at 10 levels of gamma nonlinearity. System was queried to find images containing at least 25% seagreen, as in Query ⊘.
6.6. RETRIEVAL & GAMMA

Bat image, however, was retrieved in the top 9 positions for 9 gamma levels.

The results obtained by using the $L_1$ and $L_2$ measures, in Figure 6.21 and Figure 6.22, are not as robust to changes in gamma. Specifically, the bat image was retrieved very few times in the top positions and there was a high degree of erratic ranking for all images. In addition, there was a distinct increase in false retrievals, (i.e., images which clearly did not contain the query colour or any colour that was perceptually close).

The results obtained by using the histogram techniques are shown in Figure 6.23, for $RGB_{(8,8,8)}$, and Figure 6.24 for $HSV_{(13,5,5)}$. Both methods retrieved images which exhibited similarity to the query colour, however, the $HSV_{(13,5,5)}$ scheme retrieved the bat image only once among the top 9 spots. The $RGB_{(8,8,8)}$ method performed better, retrieving the bat image 9 times in the top 9 spots.

For a more quantitative analysis, the top 25 positions at each gamma level were analyzed to study the effect on retrieval rate with gamma variation, and also to analyze the stability of the retrieval rankings across the gamma levels.

For each retrieval scheme and measure, the retrieval rate was calculated at each gamma level, as defined by Equation 6.1 and using the seagreen human query set in Figure 6.2(a). Figure 6.25 plots the results, clearly showing how retrieval rate for each measure changes as gamma varies. All curves exhibit a similar trend, however, it is important to note that the widest peak is exhibited by the new measure using the proposed scheme. This implies that this measure should provide better retrieval performance over a wider range of gamma values.

![Figure 6.25: Plot depicting the effect of gamma nonlinearity on retrieval rate.](image)

For further analysis of the results, we studied the stability of the retrieved images across the gamma levels, by tracking how many times each image appeared across the gamma levels.

For each gamma level, the top 25 retrieval results were displayed simultaneously. Each image
was tracked to see if it remained among the top 25 retrieved images, as the gamma value was varied, and the number of appearances that each image made was tabulated.

The results are shown in Table 6.5. The entries with an asterisk indicate the number of appearances that the bat image made, (i.e., the image from which the query colour was extracted). For the new measure and angular distance measure there were 21 images which appeared 6 or more times. Three images appeared 10 times when using the new measure, and 7 images appeared 10 times for the angular measure. It was found, however, that many of the images which the angular measure retrieved 10 times were not accurate retrievals, and only 3 of them were members of the Human Query Set Q for seagreen (Figure 6.2(a)). On the other hand, with the new measure, all 4 images were members of the seagreen Q.

In addition, a low number of unique images factored into the retrieval results when using the new measure. By unique images, we refer to images which appeared only once across the varying gamma level retrieval results. Thus, the ideal retrieval result over all the gamma values, would be 25 unique images, (i.e., the same 25 images retrieved consistently amongst the top 25 retrieval results), at each of the 10 gamma levels. The worst case is 25 different images retrieved at each gamma level, (i.e., 250 unique images). The angular measure had the least unique images, however, as mentioned above, many retrieved images were inaccurate retrievals.

Table 6.5: Comparison of the number of appearances each image makes in each retrieval result at 10 different gamma levels.

<table>
<thead>
<tr>
<th>appearances</th>
<th>combo</th>
<th>angle</th>
<th>L2</th>
<th>L1</th>
<th>RGB (8,8,8)</th>
<th>HSV (13,5,5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>12</td>
<td>27</td>
<td>27</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>14</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>14</td>
<td>14*</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8</td>
<td>12*</td>
<td>11</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>2</td>
<td>3*</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>4*</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2*</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>7*</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

The other investigated measures and techniques did not provide results which were nearly as robust to gamma changes as our proposed method and measure. They resulted with a very high number of unique images and few images made more than 6 appearances across the gamma levels. The only exception is the HSV(13,5,5) histogram scheme, which seems to perform well in the presence of gamma nonlinearity. However, if we look at Figure 6.25, where retrieval rate is plotted against gamma, we see that the HSV-histogram technique exhibits the lowest retrieval rates across the
gamma values and thus the worst performance.

Overall, the proposed new measure and retrieval scheme provides the highest retrieval rates over a wider range of gamma values, above all the other techniques, while retaining a very low number of unique images.

This robustness and stability to gamma can be attributed to the fact that the new measure takes both angle and magnitude difference into consideration. Since gamma nonlinearity primarily affects intensity, angular differences remain relatively unaffected. This explains why the angular measure results with a low number of unique images but contains false retrievals. Intensity difference, however, is still important for optimal colour similarity, which explains why the proposed new measure also has a low number of unique images, but at the same time exhibits much better retrieval rates.

6.7 Summary

Our results indicate that the proposed scheme for indexing and retrieval, along with the proposed new measure of similarity, exhibits high performance, both from a qualitative and quantitative view. The new measure exhibits higher retrieval rate, as determined through the Human Query Sets and the precision-recall graphs further attest to the high performance. Furthermore, the proposed scheme has been found to outperform common colour histogram indexing and retrieval schemes, which are widely used in current image retrieval systems and also provides much better retrieval efficiency than the popular QBIC system developed by IBM for the five colour queries investigated. Moreover, it has been demonstrated that the proposed scheme and measure exhibit relatively high insensitivity to unknown gamma nonlinearity. The retrieval results tend to be less affected by gamma variations and the retrieval rate of the system remained relatively high over a wide range of gamma values; this is a desirable trait since, in general, the gamma value of a database image is unknown.

All in all, the proposed system and measure provide a colour image retrieval scheme that allows for flexible colour query structure with high retrieval performance and high insensitivity to gamma nonlinearity.
Chapter 7

Conclusions & Future Work

In this thesis we presented a novel scheme for retrieval of colour images from a large database of natural colour images, based solely on colour information. It addresses feature extraction, image indexing, retrieval, and also incorporates some user feedback to refine retrieved results. Our system does not use the conventional means of indexing and retrieval which implement colour histograms. We implement colour vector techniques on three-dimensional RGB colour vectors. In this way, we end up with a small image index which contains all the perceptually relevant and prominent colour features of a given image, along with some spatial information. A perceptually-tuned measure of similarity based on the angle between two vectors is then implemented to calculate colour similarity between query colours and indexed colours, and a final image retrieval ranking is determined through the proposed multidimensional query distance space.

In Chapter 2 the importance of colour was discussed and how important its representation is for accurate and effective acquisition and reproduction. Unfortunately, there is no “ideal” colour representation space which can be used for all applications. Thus, selection of a colour space is highly dependent on the intended application and hardware which will be used. In addition, the number of acquisition and reproduction devices which exist impose a nonlinear transformation of image data, known as gamma nonlinearity, which can have profound effects on colour similarity and matching.

7.1 New Measure of Colour Similarity

In Chapter 3 we discussed some popular vector measures of similarity which are used for colour similarity calculations. We then introduced a new measure, a member of the content-based family, which is based on the vector angular measure between two vectors, perceptually tuned through an
7.2 Indexing

Recursive HSV Space Segmentation

In Chapter 4 we presented our feature extraction scheme. Specifically, for each database image, recursive HSV-space segmentation is performed to extract regions of prominent and perceptually relevant colour. This is accomplished by identifying regions of the HSV cone which are BRIGHT CHROMATIC, CHROMATIC, BLACK, and WHITE. Those which are BRIGHT CHROMATIC and CHROMATIC are segmented using a recursive HUE histogram thresholding technique, which also takes into consideration the multi-modal nature of the SATURATION histogram.

Post Processing Operations

Following the segmentation of the colour regions, we perform post-processing operations to the regions. Specifically, we perform median filtering followed by morphological operations of opening and closing, to essentially remove small irrelevant pixel regions to provide homogeneous and smooth colour regions. Then, a contour following algorithm is applied to trace the contour of all the segmented regions. This allows the system to remove any regions which were unfiltered, and at the same time to capture some spatial information such as the number of objects, their perimeter, and their area.

Representative Vectors

The average RGB value of each segmented region is then assigned as the representative vector of those regions and is stored in the image index, along with the number of regions which contain the extracted colour. It was found that by using this feature extraction technique, a low number of colours are extracted, while still maintaining a good low-level representation of the image colour content.

Colour Categorization

Upon storing the extracted representative vectors and building the index, we also perform some coarse colour categorization of the representative vectors to allow a quick look as to the types of colours contained in a given image. This way, during the retrieval stage, only indices which may contain representative vectors (colours) belonging to the same category as the query colours are visited. This reduces search time by limiting similarity calculations on a smaller subset of indices.
In summary, our indexing method builds relatively small image indices which require less storage space than conventional colour histogram techniques, while at the same time incorporating some spatial information and allowing for easy addition of future colour information.

7.3 Retrieval

In Chapter 5 we addressed the issue of retrieval and showed how we integrate our measure of similarity with the database indices to retrieve images based on colour information.

**Multidimensional Query Distance Space**

We introduced the Multidimensional Query Distance Space, the core of the retrieval scheme, which is a query-dependent space whose dimension is defined by the number of query colours. Specifically, the distances of the closest indexed colours to the query colours form a vector which lies in this space. The location of this distance vector within the space, and its relation to the origin and *equidistant line*, determines the overall ranking of a given image.

**Flexible Query Structure**

It was shown in Section 5.2.5 that the proposed scheme exhibits great flexibility. Querying can be performed in a number of ways, including query-by-colour, where any number of query colours can be specified, and query-by-example, where a sample image is fed in and images which are similar to it, in terms of colour content, are retrieved. For query-by-example, we also take into account *Colour Cardinality* and the tolerance which human perception has with the number of different colours present in a given image. It was found that as the number of different coloured objects in a given image increased, so did the tolerance for the number of coloured objects required in another image, for the two to be considered similar in colour content.

In addition, the proposed method of retrieval allows easy and effective incorporation of *colour exclusion*, where certain colours can be specified in the query to *not* be present in any of the retrieved images. By virtue of the Multidimensional Query Space, the similarity that indexed colours have to exclusion colours is used to affect the overall ranking of a given image *without* requiring a separate filtering stage to remove images with unwanted colours, as would be required with other retrieval methods such as colour histograms.

Furthermore, we also show how easy it is to provide *query refinement* via relevance feedback when using our colour vector indexing and retrieval scheme. If a user wishes to refine a retrieval result, they are able to do so either by tagging the relevance of colour swatches and depicting the representative vectors of the top 10 retrieved images, or by tagging the relevance of entire images.
In both cases, the system refines the initial query vectors by determining the most centrally located colour vector among the tagged colours, using the *sum-of-angles* criterion. These new query vectors are then used to refine the search by performing similarity calculations on the top set of retrieved images.

Results

In Chapter 6 we present various image retrieval results which our system generated based on a variety of query definitions. In addition, results from some common colour histogram indexing and retrieval techniques are included along with results obtained using other vector distance measures, instead of those proposed.

In summary, the retrieval results using the proposed scheme exhibit higher retrieval efficiency than popular histogram indexing techniques. In addition, the results agree much more closely with results tabulated from 25 human volunteers, who manually searched through our image database and selected images which were "felt" to match a given query. Also, it was found that the system and new measure exhibited higher retrieval efficiency than IBM's QBIC retrieval system, which was tested on our image database using five specific colour queries.

Furthermore, resistance to gamma nonlinearity was investigated and it was found that the proposed scheme and similarity measure resist changes to gamma much better than other techniques, including histogram techniques. Specifically, our scheme exhibited the highest retrieval rate over a wider range of gamma nonlinearity than all other techniques and measures investigated. This proves to be of utmost importance since general unconstrained image databases contain images from unknown sources.

7.4 Future Work

- Since the presented work concentrates on colour features and similarity for retrieval, future work should concentrate on integrating other types of features, both low-level and high-level. In particular, information regarding spatial location of colour regions within an image and the spatial relation between colour regions in an image, should be incorporated. By virtue of the relatively small size of our proposed indices, incorporating such spatial information should not increase the total index size considerably.

- The collection of image indices should be organized in such a way as to make the searching and retrieval process more efficient, in terms of speed. The coarse colour categorization which is performed at the indexing stage is a first step. This information can be used to build a
7.4. FUTURE WORK

relational association of the indices, similar to a tree structure, that will limit searching to specific nodes or clusters.

As the database grows, an in-depth statistical analysis of the indices can be performed to analyze the indexed colour distribution. By collecting such data, the system can be fine-tuned since better insight as to how much of a certain type of colour exists. For instance, if it is found that an image database is biased towards images with high blue colour content, the retrieval stage can be adjusted to perform more strict similarity calculation on query colours which fall in the blue category, and more relaxed similarity calculation for all other query colours. In addition, such statistical feedback can provide important information on what the dominant colours, or clusters, are in the indexed data, to allow for a more efficient categorization.

A more thorough comparison with other image retrieval systems, both commercial and educational, is ultimately needed to see how the new system and measure compares. The initial test with QBIC shows promise and that higher level techniques, such as the proposed technique, are required for efficient and effective colour image retrieval.

Feedback from the user regarding retrieved results can potentially be used to modify the measure of similarity so that it behaves differently for different colours. In essence, the retrieval stage can use the collected information and slowly adapt the similarity measure accordingly. As more data is continually collected over long periods of use and many queries, important perceptual information can be gathered both for individual users and humans in general, to strive for even more effective colour retrieval.
References


[34] A. H. Munsell, A Color Notation; an illustrated system defining all colors and their relations by measured scales of hue, value and chroma, Munsell Color Company, Baltimore, MD, 1905.


Appendix A

Image Database

The images contained in this database are all royalty-free high resolution 24-bit colour images. Some images have been scanned from the author's private photographic collection while the rest were obtained from COREL Corporation's Super Ten Royalty-Free High-Resolution Photo Paks, purchased from COREL.
©Dimitrios Androutsos, 1999