Histogram Refinement for Content-Based Image Retrieval

Students:

- 120050038 Harish Koilada
- 120050070 Nikhil Sri Ram
- 120070034 Sai Akhil Suggu

Abstract:

We used a technique for comparing images called histogram refinement, which imposes additional constraints on histogram based matching. Histogram refinement splits the pixels in a given bucket into several classes, based upon some local property. Within a given bucket, only pixels in the same class are compared. We then implemented a better technique called split histogram using color coherence vectors (CCV), which partitions each histogram bucket based on spatial coherence. A database with 12 images has been queried using CCV's. We demonstrate that Color coherence vectors can be used to distinguish images whose colour histograms are indistinguishable. Further this method can be improved by using successive refinement (similar to histogram refinement using CCV).

Introduction

Our implemention is the process of retrieving images from a database or library of digital images according to the visual content of the images. Images have always been an inevitable part of human communication and its roots millennia ago. Images make the communication process more interesting, illustrative, elaborate, understandable and transparent.

Colour histogram is one of the most important technique for content based image retrieval because of its efficiency, effectiveness and are generally insensitive to small changes in camera position. However due to statistical nature, colour histogram can only index the contents of the image in a limited way. Images with similar histograms can have dramatically different appearances i.e. A colour histogram provides no spatial information; it merely describes which colours are present in the image, and in what quantities. To make colour histogram more effective for image indexing, spatial information should be considered. Our project uses a split histogram called a color coherence vector (CCV), which partitions pixels based upon their spatial coherence.

Color coherence vector is a split histogram which consist of coherent vector and incoherent vector. We define a color's coherence as the degree to which pixels of that color are members of large similarly-colored regions. We refer to these significant regions as coherent regions, and observe that they are of significant importance in characterizing images. Our coherence classifies pixels as either coherent or incoherent. Coherent pixels are a part of some sizable contiguous region, while incoherent pixels are not. A color coherence vector represents this classification for each color in the image. This notion of coherence allows us to make fine distinctions that cannot be made with simple color histograms. Two different images with identical color histograms can have different split histograms.

Implementation:

In histogram refinement, the pixels within a given bucket are split into classes based upon some local property. Split histograms are compared on a bucket by bucket basis, similar to standard histogram matching. Within a given bucket, only pixels with the same property are compared.

The initial stage in computing a CCV is similar to the computation of a color histogram. First blur the image slightly by replacing pixel values with the average value in a small local neighbourhood including the 8 adjacent pixels. Define the color space with only n distinct colors in the image. The next step is to classify the pixels within a given color bucket as either coherent or incoherent. After words determine the pixel groups by computing connected components.

When this is complete, each pixel will belong to exactly one connected component. Classify pixels as either coherent or incoherent depending on the size of its connected component. A pixel is coherent if the size of its connected component exceeds a fixed value τ; otherwise, the pixel is incoherent.

For a given discredited color, some of the pixels with that color will be coherent and some will be incoherent. Let us call the number of coherent pixels of the jth discrete color αj and the number of incoherent pixels β_j. Clearly, the total number of pixels with that color is $\alpha_j + \beta_j$, and so a color histogram would summarize an image as $(\alpha_1 + \beta_1,\ldots, \alpha_n + \beta_n)$

Instead, for each color we compute the pair (α_j, β_j) which we will call the coherence pair for the jth color. The color coherence pair's vector for the image consists of $\{(\alpha_1, \beta_1),\ldots, (\alpha_n,\beta_n)\}$. Classification of coherence is determined by a fixed value τ. Each pixel is checked whether coherent or not. Two images I and I_1 can be compared using their CCV's, by using the L distance. Let the coherence pairs for the jth color bucket is (α_j, β_j) in I and (α'_j, β'_j) in I'. Using the L distance to compare CCV's, the jth bucket's contribution to the distance between I' and I is

$$\Delta CCV = |\alpha_j - \alpha'_j| + |\beta_j - \beta'_j|$$

Note that when using color histograms to compare I and I', the j'th bucket's contribution is

$$\begin{aligned}\Delta CH &= |(\alpha_j + \beta_j)-(\alpha'_j+ \beta'_j)| \\ &= |(\alpha_j - \alpha'_j + \beta_j - \beta'_j)| \\ &\le |\alpha_j - \alpha'_j| + |\beta_j - \beta'_j| \ (= \Delta CCV)\end{aligned}$$

Thus formally, $\Delta CH \le \Delta CCV$. It follows that CCV's create a finer distinction than color histograms. A given color bucketj can contain the same number of pixels in *I and I'*, but these pixelsmay be entirely incoherent in *I* and entirely coherent in *I'*.

Centering refinement is defined as spliting the buckets of a histogram based on whether the pixel is located in the 75% of the image.

Further, when using centering refinement in both the cases,

$\alpha_j = m(\alpha_j) + o(\alpha_j)$ - $m(\alpha_j)$ gives the number of centered pixels

- $o(\alpha_j)$ gives the remaining pixels in the bucket

$$\Delta CHRef = |(m(\alpha_j) + m(\beta_j)) - (m(\alpha'_j) + m(\beta'_j))| + |(o(\alpha_j) + o(\beta_j)) - (o(\alpha'_j) + o(\beta'_j))|$$

$$\Delta CH = |(m(\alpha_j) + m(\beta_j)) - (m(\alpha'_j) + m(\beta'_j)) + (o(\alpha_j) + o(\beta_j)) - (o(\alpha'_j) + o(\beta'_j))|$$

$\Delta CH \leq \Delta CHRef$

$\Delta CCV = |\alpha_j - \alpha'_j| + |\beta_j - \beta'_j| = |(m(\alpha_j) + o(\alpha_j)) - (m(\alpha'_j) + o(\alpha'_j))| +$

$|(m(\beta_j) + o(\beta_j)) - (m(\beta'_j) + o(\beta'_j))|$

$\Delta CCVRef = |m(\alpha_j) - m(\alpha'_j)| + |o(\alpha_j) - o(\alpha'_j)| + |m(\beta_j) - m(\beta'_j)| + |o(\beta_j) - o(\beta'_j)|$

$\Delta CHRef$ cannot be compared with ΔCCV but $\Delta CHRef \leq \Delta CCVRef$

$\Delta CCV \leq \Delta CCVRef$

Inference:

$\Delta CH \leq \Delta CHRef \leq \Delta CCVRef$

$\Delta CH \leq \Delta CCV \leq \Delta CCVRef$

So Successive Refinement produces better results due to better distinction as evident from the above equations

Algorithm:

- The image needs to be discretized to the required number of colors(512)
- Calculate the CCV of the image
- For the database images the CCV needs to calculated
- For each database image CCV calculate the L1 distance and for the image which the error is least is given as the output

If I is the database image and I' is the query image, then the similarity measure is computed as follows,

- Calculate histogram vector vI = [vI1, vI2, ….vIn] and ccv vector cI = [cI1, cI2, ….cIn] of the database images
- Calculate the vectors vI' and cI' for the query image also
- The Euclidean distance between two feature vectors can then be used as the similarity measurement

Results:

L1_TestimagesAgainstDatabase_hist =

1	2	3	4	5	1	7	8	9	10	11	12
12732	11040	18144	13766	9996	16436	10944	18758	13294	1143	17874	16738

L1_TestimagesAgainstDatabase_histRef =

1	2	3	4	5	6	7	8	9	10	11	12
14996	13092	23622	15964	12298	18432	14396	20340	14228	15076	19426	20036

L1_TestimagesAgainstDatabase_ccv =

1	2	3	4	5	6	7	8	9	10	11	12
14272	17094	18962	15056	11472	20844	12262	19348	14786	15150	19590	20372

L1_TestimagesAgainstDatabase_sucRef =

1	2	3	4	5	6	7	8	9	10	11	12
15838	18022	24318	18060	13624	22622	15436	20930	17360	17804	22014	23888

L1_DatabaseAgainstTestimages_hist =

1	2	3	4	5	5	7	8	9	10	11	12
12732	11040	18144	13766	9996	15434	10944	18758	13294	11438	17874	16738

L1_DatabaseAgainstTestimages_histRef =

1	2	3	4	5	5	7	8	9	10	11	12
14996	13092	23622	15964	12298	18218	14396	20340	14228	15076	19426	20036

L1_DatabaseAgainstTestimages_ccv =

1	2	3	4	5	5	7	8	9	10	11	12
14272	17094	18962	15056	11472	17416	12262	19348	14786	15150	19590	20372

L1_DatabaseAgainstTestimages_sucRef =

1	2	3	4	5	5	7	8	9	10	11	12
15838	18022	24318	18060	13624	19868	15436	20930	17360	17804	22014	23888

The first row indicates the most similar image among the database images for that particular test image. The second row indicates the error between the two of them. Both the test and database images are numbered from 1 to 12, and the i^{th} column in the results corresponds to the i^{th} test image

Conclusion:

The color histograms of two different images can be same as they do not contain any spatial information in them. Where as histogram refinement introduces some amount of spatial information(centering refinement). But this may not give better results all the time. If we use the spatial coherence by calculating the CCV of the image, then we are guaranteed to make better distinction between two images as $\Delta CH \leq \Delta CCV$. Now applying the centering refinement(successive refinement) on the CCV further gives better distinction as $\Delta CCV \leq \Delta CCVRef$. Hence the use of successive refinement can improve the Content Based Image Retrieval. CCV is insensitive to small changes in rotations and translations.

Pitfalls:

- In some cases where two different images with similar content can cause erroneous results
- The CCV is sensitive intensity changes between images, this can lead to wrong results

Work distribution:

120050038 – Color Coherence vector, Successive refinement codes

120050070- Color space discretization, testing functions

120070034- Analyzing the results, color coherence vector, testing functions