Quantization of color histograms using GLA

Christopher C. Yang* and Milo K. Yip
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong

ABSTRACT
Color histogram has been used as one of the most important image descriptor in a wide range of content-based image retrieval (CBIR) projects for color image indexing. It captures the global chromatic distribution of an image. Traditionally, there are two major approaches to quantize the color space: (1) quantize each dimension of a color coordinate system uniformly to generate a fixed number of bins; and (2) quantize a color coordinate system arbitrarily. The first approach works best on cubical color coordinate systems, such as RGB. For other non-cubical color coordinate system, such as CIELAB and CIELUV, some bins may fall out of the gamut (transformed from the RGB cube) of the color space. As a result, it reduces the effectiveness of the color histogram and hence reduces the retrieval performance. The second approach uses arbitrarily quantization. The volume of the bins is not necessary uniform. As a result, it affects the effectiveness of the histogram significantly. In this paper, we propose to develop the color histogram by tessellating the non-cubical color gamut transformed from RGB cube using a vector quantization (VQ) method, the General Loyld Algorithm (GLA) [6]. Using such approach, the problem of empty bins due to the gamut of the color coordinate system can be avoided. Besides, all bins quantized by GLA will occupy the same volume. It guarantees that uniformity of each quantized bins in the histogram. An experiment has been conducted to evaluate the quantitative performance of our approach. The image collection from UC Berkeley's digital library project is used as the test bed. The indexing effectiveness of a histogram space [3] is used as the measurement of the performance. The experimental result shows that using the GLA quantization approach significantly increase the indexing effectiveness.

1. INTRODUCTION
Identifying effective features of color images is a fundamental problem for color image indexing and retrieval. Recent researches have focused on low-level visual features, including color, texture, shape, etc. These features can be extracted automatically and efficiently. Among the variety features, color histogram has been used as one of the most important image descriptor in a wide range of content-based image retrieval (CBIR) projects [5,7,8,9]. Color histogram captures the global chromatic distribution of an image. It has the advantage of being invariant under translation and rotation about the viewing axis. Despite changes of view, the color histogram only changes slightly in changes of scale and occlusion. Swain et al. [13] presented a general approach of using color histogram in image retrieval. Given a discrete color space, the color histogram is obtained by discrete the image colors and counting the frequency of each discrete color occurs in the image array.

In many recent projects of color image retrieval, several color coordinate systems, such as HSV [15] and CIELUV [1], have been employed for the color histograms. There are two major approaches to quantize the color space: (1) quantize each dimension of a color coordinate system uniformly to generate a fixed number of bins; and (2) quantize a color coordinate system arbitrarily. The first approach has been adopted in many CBIR systems. Zhong et. al [15] quantized HSV space to 256 bins, with 8 divisions in both H and S, and 4 divisions in V. ImageRover [10] quantized CIELUV space to 64 bins, with 4 units per dimension. Yang et. al [14] uses $10 \times 10$ chromatic bins in the H and S dimension of the LHS coordinate system. The first approach works best on cubical color coordinate

* yang@se.cuhk.edu.hk; phone (+852) 2609 8239; fax (+852) 2603 5505; http://www.se.cuhk.edu.hk/~yang/;
Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China
systems, such as RGB. For other non-cubical color coordinate system, some bins may fall out of the gamut (transformed from RGB cube) of the color coordinate systems. Thus, the frequency of these bins will always be zero. Removing these empty bins from the histogram reduces the dimensions; furthermore, the bins on the boundary of the gamut are still not fully utilized. As a result, it reduces the effectiveness of the color histogram and hence reduces the retrieval performance. Indeed, most of the color coordinate systems, including the CIE uniform color spaces (CIELAB and CIELUV), are non-cubical.

Alternatively, some researches applied the second approach. Albuz et. al [1] quantized the CIELAB color space to 5 groups by arbitrarily planes. Smith et. al [11] proposed a color sets as approximation to color histogram, where the HSV space is quantized to 18 values of hue (20 degree each), 3 levels of saturations, and 3 values of lightness, plus 4 levels of grays (166 color bins totally). Since this approach uses arbitrarily quantization, the volume of the bins is not necessary uniform, which affect the effectiveness of the histogram significantly.

In this paper, we propose to develop the color histogram by tessellating the non-cubical color spaces using a vector quantization (VQ) method, the General Loyld Algorithm (GLA) [6]. Using such approach, the problem of empty bins due to the gamut of the color space can be avoided. Besides, all bins quantized by GLA will occupy the same volume. It guarantees that uniformity of each quantized bins in the histogram. 

2. SYSTEM ARCHITECTURE

An overview of our approach is depicted in Figure 1. In our approach, it is composed of two phases, codebook generation and histogram generation. In codebook generation phrase, a color space is tessellated using GLA, given dimension n (number of bins). In the histogram generation phrase, color histogram of each image is generated using the generated codebook.

Figure 1: Overview of the algorithm
3. CODEBOOK GENERATION

In the codebook generation phrase, a lookup table, which maps from color space to histogram space, is generated. The procedure includes samples generation, color coordinates transformation, training sample generation, GLA training and lookup table generation. Figure 2 illustrates these procedures by visualizing them on a two-dimension space.

Figure 2: Codebook Generation (illustration in two-dimensional space)

Figure 3: CIELUV color space visualized by the 6 surfaces.
First, the boundary of the original coordinate system (the square in figure 2) is generated. In our algorithm, RGB is used as the original color coordinate system, thus its boundary is composite of six surfaces of the RGB cube. The next step is to transform the boundary samples to a target color coordinate system using a color coordinate transformation function, f. As shown in the Figure 2, the shape of the transformed boundary is irregular. Figure 3 shows an example of the transformed boundary of the CIELUV color coordinate system.

The uniform training samples, T, are then generated for the target color coordinate system based on the uniform quantization of each dimension of RGB color cube. Given that $C_{rgb}$ is the set of uniformly quantized color bins of the RGB color coordinate system, $S = \{ c_1, c_2, \ldots, c_m \ | \ c_i \in C_{rgb} \}$ and $S' (= \{ f(c_i) \ | \ c_i \in S \})$ is transformed from $S$ using the formulation of color coordinate transformation to the target color coordinate system ($C'$), $f: C_{rgb} \rightarrow C'$. Using the modified scan-line polygon-filling algorithm, uniform training samples, $T$, are generated with a given resolution. Figure 4 illustrates the training samples of CIELUV color coordinate system.

![Figure 4: Uniform training samples generated by ScanLine](image)

The training samples, $T$, are then trained by GLA, with a threshold, $\Delta t$, to develop a codebook with $n$ codebook vectors evenly distributed in the target color coordinate system, $C'$. Each codebook vector occupies a volume in 3D, called cell. The distance between any points in a cell to its codebook vector is the minimum among all codebook vectors. The Voronoi diagram shown in Figure 2 illustrates this property. Figure 5 shows the 64 codebook vectors generated by GLA for CIELUV color coordinate system. In order to accelerate the process of generating histogram, a lookup table $g$ for mapping each discrete color in RGB to the generated codebook is constructed. Given the generated codebook, $K=\{k_1, k_2, \ldots, k_n\}$, $g(c) = \arg \min || f(c) - k_i ||$ is the mapping from RGB to index of cell.

The GLA algorithm is as followed:
function GLA(T, n, Δt)
begin
    Initialize an n-dimensional codebook C_1 with random values;
    t_1 := 0;
    m := 1;
    loop:
        C_{m+1} := Lylod_Iteration(T, C_m, n)
        t_{m+1} := AverageDistortion(T, C_{m+1}, n)
        if |t_{m+1} - t_m| > Δt then
            begin
                m = m + 1;
                goto loop;
            end;
        return C_{m+1}
end;

function Lylod_Iteration(T, C={y_1, y_2, ..., y_n}, n)
begin
    for i := 1 to n
    begin
        // Step1: Partition the training set into cluster sets R_i
        R_i = { x ∈ T : d(x, y_i) ≤ d(x, y_j); all j≠i }
        // Step2: Compute the centroid for the cluster sets to obtain new codebook
        cent_i = cent(R_i)
    end;
    return { cent_1, cent_2, ..., cent_n }
end;

function AverageDistortion(T, C={y_1, y_2, ..., y_n}, n)
begin
    for j := 1 to n
    begin
        d_j := \min_{x \in T} d(x_i, y_j)
    end;
    return mean(d_1, d_2, ..., d_n);
end;

4. HISTOGRAM GENERATION

In the histogram generation phase, a color histogram H(M) of image M is a 1-D discrete vector representing the frequencies of color bins in image M, which is typically defined as:

\[
H(M) = [h_1, h_2, ..., h_n]
\]

\[
h_k = \frac{n_k}{N} \quad k = 1, 2, ..., n
\]

where N is number of pixels in image M and n_k is the number of pixels with color bin h_k. The division normalizes the histogram such that:
\[
\sum_{k=1}^{n} h_k = 1.0
\]

As a lookup table \( g \) has been defined to transform the color space to the histogram space, the equation of the histogram can be modified as:

\[
h_i = \frac{1}{N} \sum_{j=1}^{N} \begin{cases} 1 & i = g(c_j) \\ 0 & \text{otherwise} \end{cases}
\]

Since the lookup table \( g \) pre-calculated the color space transformation function \( f \), the run-time complexity of histogram generation phrase for any color space is the same as the traditional approach.

Figure 5: 64 codebook vectors generated by GLA
(Brightness visualizes the distance from the view-point)

5. EXPERIMENT

We have conducted an experiment to evaluate the quantitative performance of our approach. The image collection from UC Berkeley's digital library project [4] is used as the corpus. The indexing effectiveness of a histogram space [3] is used as the measurement of the performance. Given a \( n \)-dimensional histogram space \( H \), a metric \( d \) on \( H \) and
a distance threshold \( t \), the capacity \( C \) of \( H \) is given by the maximal number of \( t \)-different histograms that fit into \( H \) [12]. An empirical capacity curve \( C(t) \) is computed by all image couples in an image database [2]. The indexing effectiveness, \( \varepsilon \), of a histogram space is defined as [3]:

\[
\varepsilon = \int y C(y) dy
\]

In our experiment, \( L_1 \)-norm is utilized as the metric to measure the similarity between two histograms. The histogram spaces using the combination of CIELAB or CIELUV, and GLA quantization approach or uniform quantization on each dimension of the corresponding color spaces are compared. The results are shown in Table 1.

The experimental result shows that using the GLA quantization approach significantly increase the indexing effectiveness. However, using the GLA quantization approach on CIELAB or CIELUV has not significant difference.

<table>
<thead>
<tr>
<th>Histogram Space</th>
<th>Indexing Effectiveness, ( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIELAB and GLA quantization approach</td>
<td>66.728</td>
</tr>
<tr>
<td>CIELUV and GLA quantization approach</td>
<td>66.382</td>
</tr>
<tr>
<td>CIELAB and uniform quantization on each dimension</td>
<td>53.805</td>
</tr>
<tr>
<td>CIELUV and uniform quantization on each dimension</td>
<td>50.755</td>
</tr>
</tbody>
</table>

6. CONCLUSION

Color histogram is a popular technique to represent the chromatic feature of color images for content-based image retrieval. However, in generating color histogram, the traditional approaches of quantization, uniform quantization on each dimension of the color coordinate system separately and arbitrary quantization of the color coordinate system, cannot effectively utilize the color histogram for indexing. Uniform quantization does not fully utilize the color histogram. Many empty bins may be wasted for indexing. Arbitrary quantization does not produce a uniform volume for all quantized bins. In this paper, we have proposed to generate the color histogram by GLA. Using such approach, the problems in uniform quantization and arbitrary quantization can be avoided. The indexing effectiveness has been used in our experiment to compare the performance of different quantization techniques applying on CIELAB and CIELUV. The experimental result shows that quantization using GAL significantly improve the indexing effectiveness.

ACKNOWLEDGEMENT

This project is supported by the Earmarked Grant for Research of the Hong Kong Research Grants Council, CUHK 7034-98E.

REFERENCES

4 Digital Library Project, University of California, Berkeley, http://elib.cs.berkeley.edu/