Classification of Raster Maps for Automatic Feature Extraction

Yao-Yi Chiang and Craig A. Knoblock
University of Southern California
Department of Computer Science and Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292
yaoyichi, knoblock@isi.edu

ABSTRACT

Raster maps are widely available and contain useful geographic features such as labels and road lines. To extract the geographic features, most research work relies on a manual step to first extract the foreground pixels from the maps using the distinctive colors or grayscale intensities of the pixels. This strategy requires user interaction for each map to select a set of thresholds. In this paper, we present a map classification technique that uses an image comparison feature called the luminance-boundary histogram and a nearest-neighbor classifier to identify raster maps with similar grayscale intensity usage. We can then apply previously learned thresholds to separate the foreground pixels from the raster maps that are classified in the same group instead of manually examining each map. We show that the luminance-boundary histogram achieves 95% accuracy in our map classification experiment compared to 13.33%, 86.67%, and 88.33% using three traditional image comparison features. The accurate map classification results make it possible to extract geographic features from previously unseen raster maps.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial Databases and GIS*

General Terms

Algorithms, Design

Keywords

Raster Map Classification, Content-Based Image Retrieval, Image Similarity, Luminance-Boundary Histogram, Color-Coherence Vectors, Color Histogram, Color Moments

1. INTRODUCTION

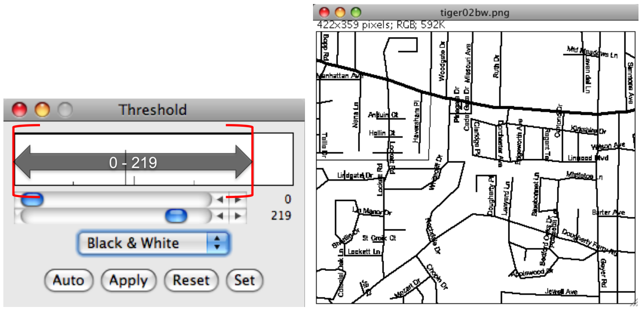
Due to the popularity of Geographic Information System (GIS) and high quality scanners, we can now obtain more and more raster maps from various sources on the Internet. For example, the United States Geological Survey (USGS) distributes various types of scanned maps, such as topographic maps and thematic maps displaying water resources through their own website[1] and the TerraServer-USA website.[2] Map repositories such as the University of Texas Map Library[3] contain information rich scanned maps for many areas around the globe, including historical scanned maps. Moreover, in our previous work, we developed an automatic approach to mining collections of maps from the Web [13]. In that work, we harvest images from image search engines (e.g., Yahoo Image Search) and then identify raster maps among the images.

Raster maps are an important source of geospatial information. First, raster maps provide geographic features that are difficult to find elsewhere, such as the landmarks in historical maps. Moreover, for certain types of geographic features, raster maps contain the most complete set of data, such as the USGS topographic maps that have the contour lines of the entire United States in various scales. However, because of the variety of image quality (e.g., poor image quality from scanning and/or image compression processes), the complexity of maps, and the typical lack of metadata (e.g., map geocoordinates, map source, original vector data, etc.), it is difficult for computers to automatically extract geographic features from the maps and utilize the information locked in the raster format.

To automatically process the raster maps, much of the research work relies on user input to extract the foreground pixels (e.g., pixels representing road lines and text labels) from the maps as a preprocessing step of their feature extraction algorithms. Salvatore and Guitton [15] use a color classification technique as their first step to extract contour lines from topographic maps. Cao et al. [1] utilize a preset grayscale threshold to remove the background pixels from raster maps and then detect text labels from the maps. Li et al. [10] utilize an image filter to first extract the "black layer" from the USGS topographic maps and then work on the black layers to extract and rebuild the text labels and lines. Khotanzad et al. [9] utilize a color segmentation method with user annotations to extract the contour lines from the USGS topographic maps. Chen et al. [2] later extend the color segmentation method from Khotanzad et al. [9] to handle common topographic maps (i.e., not limited to the USGS topographic maps) using local segmentation techniques.

ACM GIS '09 Seattle, WA USA

[1] http://nmviewogc.cr.usgs.gov/viewer.htm
[2] http://terraserver-usa.com/
[3] http://www.lib.utexas.edu/maps/



(a) Adjusting thresholds (b) Foreground pixels

Figure 1: Adjusting thresholds in the grayscale histogram to extract foreground pixels from a TIGER/Line map

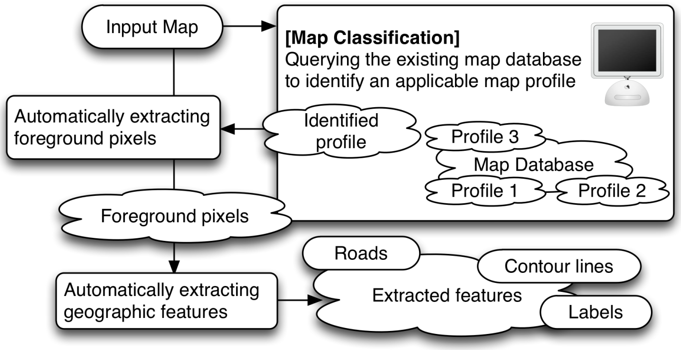
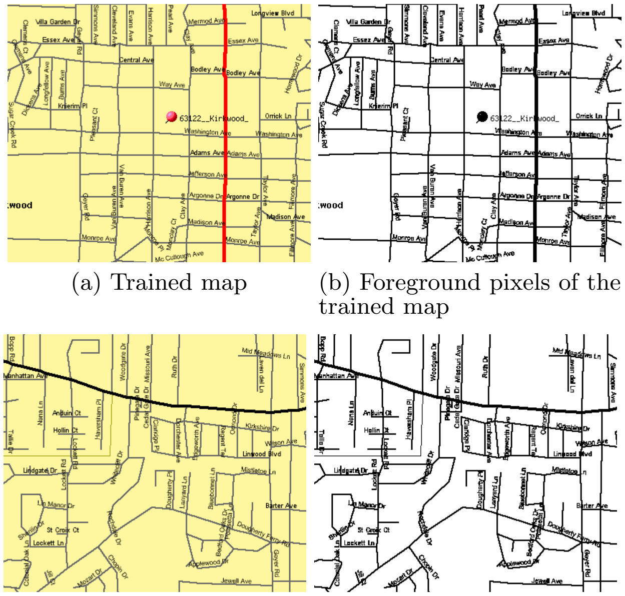


Figure 2: A feature extraction system with a map classification component



(a) Trained map (b) Foreground pixels of the trained map

(c) New input map (d) Foreground pixels of the new input map

Figure 3: Example maps from TIGER/Line and their foreground pixels

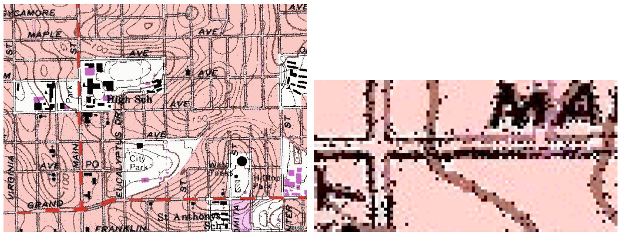
These feature extraction techniques all require prior knowledge of the raster maps and experiments to generate a proper set of color or grayscale thresholds to first extract foreground pixels from the maps. In particular, because the raster maps need to be readable when printed with non-color printers, the luminance (i.e., the grayscale intensity) is the most representative component among the color components by design, such as the red, green, and blue (RGB) or hue, saturation, and luminance (HSL) components. Therefore, to separate the foreground pixels from a raster map, a common approach is to manually examine the grayscale histogram of the raster map and select the luminance intervals that separate the foreground pixels from the map background. For example, Figure 1 shows that we can adjust the thresholds in the grayscale histogram using an image processing software[4] to identify a luminance interval for separating the foreground pixels from a TIGER/Line map.[5]

Because of the varieties of image quality and complexity of raster maps, it is tedious to manually examine every input map for extracting the foreground pixels from the map. To minimize the manual work and to enable automatic feature extraction processes, we need to be able to automatically reuse the trained map profiles (i.e., the luminance intervals) on applicable maps. Figure 2 shows a feature extraction system. The example system includes a map classification component to eliminate the repetitive manual examination step and hence the system can automatically process previously unseen maps if the map is classified as similar to one of the trained maps. For example, if we have a trained map profile to extract the foreground pixels from the map shown in Figure 3(a), the map classification component can automatically select the trained map profile for a new input map shown in Figure 3(c) to extract the foreground pixels from the map as shown in Figure 3(d). The solution might seem to be straightforward with this example of a TIGER/Line map, where we can simply apply the same set of thresholds on maps from the same source to extract their foreground pixels. However, it is difficult to determine the source of a raster map automatically. To make this problem worse, even maps from a single map source may need different sets of thresholds due to the noise from scanning or compression processes. For example, as shown in Figure 4 and Figure 5, both topographic maps are from USGS and the colors of roads are very different from one area to another (the noise in Figure 5 comes with the original topographic maps downloaded from TerraServer-USA). In this example, the topographic map covering El Segundo, CA requires the luminance interval of 0 to 184 and the one covering St. Louis requires the luminance interval of 0 to 36 to extract their foreground pixels.

In this paper, we developed an image comparison feature, called the luminance-boundary histogram. We built a map classification technique based on a nearest-neighbor classifier using the luminance-boundary histogram to compare an input map with previously trained maps for identifying an applicable map profile to extract the foreground pixels from the input map. The luminance-boundary histogram is based on *the spatial relationships between the luminance levels used in the raster maps (i.e., the luminance usage of an image)*. To demonstrate the spatial relationships between luminance levels, Figure 6 shows two maps from Google Maps, from

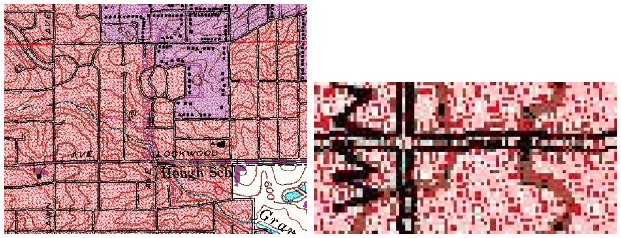
[4] We use ImageJ (http://rsbweb.nih.gov/ij/) to demonstrate the manual approach.

[5] http://tiger.census.gov/



(a) A topographic map (b) The detail view

Figure 4: An example USGS topographic map covering El Segundo, CA



(a) A topographic map (b) The detail view

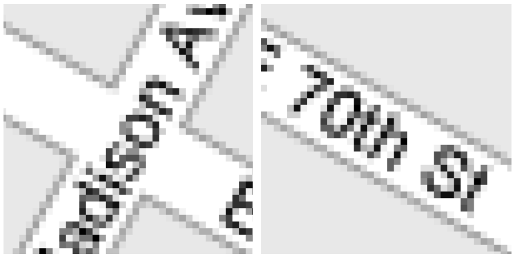
Figure 5: An example USGS topographic map covering St. Louis, MO

which we can extract their foreground pixels by applying the same luminance intervals on the maps' grayscale histogram. The pixels of various luminance levels are used in the two maps to constitute lines, characters, and background. Although the number of pixels of each luminance level is different between Figure 6(a) and Figure 6(b), the luminance usage in both images are similar. For example, the black pixels that make up the skeletons of the characters are surrounded by the same set of gray pixels used as shadows to make the characters stand out against the white pixels in the two maps. Likewise, the set of gray pixels used to draw road boundaries are always in between the white road pixels and light gray background pixels. Therefore, we can compare the luminance-boundary histograms of raster maps to identify the maps that have similar luminance usage and then apply the same map profile to extract their foreground pixels.

The remainder of this paper is organized as follows. Section 2 discusses related work on map classification features. Section 3 describes our approach to generate and compare the luminance-boundary histogram. Section 4 reports on our experimental results, and Section 5 presents the conclusion and future work.

2. RELATED WORK ON MAP CLASSIFICATION FEATURES

One approach for classifying raster maps is based on the metadata of the maps (e.g., map names, production time, location, geocoordinates, themes, etc.). The metadata can be manually specified or generated automatically from the surrounding text of the raster maps in the document where the map was obtained (e.g., a web page or an article)[8]. Map classification using metadata helps to answer queries such as finding the historical raster maps of a specific region for a specific year, but does not enable our goal to identify maps sharing the same luminance usage. On the other hand, our map classification technique compares the image content of raster maps using the luminance-boundary histogram, which identifies the raster maps that share similar spatial relationships between their luminance levels. In this section, we focus on reviewing image features used to find similar images and compare these image features with our luminance-boundary histogram.



(a) Area one (b) Area two

Figure 6: The spatial relationships between luminance levels are similar in the two images from Google Maps

The research on comparing image content to find similar images is called content-based image retrieval (CBIR), which has been a very active research topic [16]. Generally, the image features used to compare two images for identifying the similarity between them fall into one of the three categories: shape features, texture features, and color features. Shape features are used for recognizing objects of similar shapes in images, such as using the histogram of oriented gradient (HOG) descriptors for pedestrian detection from photos [7]. Since objects of the same shape in two maps can be drawn with different colors, the fact that two raster maps both have objects of the same shape does not indicate the similarity between their luminance usage. For the texture features, a commonly used set is the Tamura texture features [19], which are based on psychological experiments on human interpretation of images. The Tamura features describe the overall image texture, such as the coarseness and the contrast of the images. Another type of texture feature first transforms the image into another domain (e.g., the frequency domain) and then generates descriptions of the image textures using the transformed image, such as the Gabor wavelet transform features [11]. Other texture features are based on the edge detectors such as the edge histogram descriptor [12], which splits the image into small regions and generates the edge histogram for each region. For our problem of classifying the raster maps based on their luminance usage, using the texture features that are generated from the whole image to represent the overall texture of an image does not help much since two images with similar textures (e.g., contrast) do not necessarily share similar luminance usage. Instead, we utilize the luminance differences locally around each luminance level in the image to build the luminance-boundary histogram.

For the color features, the three major ones are the color histogram, color moments, and color-coherence vectors. These color features can work on any single or combined color components, such as using only the H or V component from the HSV color domain or a weighted combination of the R, G, and B component from the RGB color domain. If only the V component or a specific RGB combination is used, the color features then represent the luminous intensities in the image; otherwise, the H component only can be quantized and then used to represent the colors in the image. The color histogram records the number of pixels used for every color in an image [18]. For the color histograms of two images to be similar, the corresponding colors in the two histograms should have similar numbers of pixels in the images, which is often not the case with two raster maps sharing similar spatial relationships between their luminance levels. The color moments [18] are based on statistical analysis and use the average, standard deviation, and skewness of the color histogram as features to describe an image. Since the color moments are built on the color histogram, the color moments rely on the assumption that two similar images use similar numbers of pixels for the colors in the color histogram. Moreover, the color moments require manual adjustments and experiments to tune the weights of each of the statistic components. The color-coherence vectors [14] incorporate the sizes of color regions into the color histogram and usually produce better and more robust results among the three color features. However, the region size needs to be tuned to achieve the best result and the size parameter cannot be intuitively applied to compare raster maps.

These color features capture some concepts of the color or luminance usage (depends on the choice of color components) in the raster map, such as the type of colors or luminous intensities used in the images or in a region smaller or larger than a predefined size. However, these color features do not take into account the spatial relationships between colors or luminance levels used in the image, and hence do not help much on finding raster maps with similar luminance usage. The luminance-boundary histogram requires no threshold tuning and represents the luminance usage of a raster map by capturing the spatial relationships between the map's luminance levels. We compare the luminance-boundary histogram with the color histogram, color moments, and color-coherence vectors in our experiments. The experiments show that the luminance-boundary histogram is efficient and produces the best and most robust results compared to the other color features for classifying raster maps.

3. RASTER MAP CLASSIFICATION BASED ON IMAGE CONTENT

In this section, we present our map classification technique that uses the luminance-boundary histogram and a nearest-neighbor classifier for identifying raster maps sharing the same luminance interval for extracting their foreground pixels. We describe the technical details to generate the luminance-boundary histogram and the metric that the nearest-neighbor classifier uses to compare the luminance-boundary histograms of two images.

3.1 Generating the Luminance-Boundary Histogram

We generate two luminance-boundary histograms (LBH) for an image. The first luminance-boundary histogram is called the high luminance-boundary histogram (HLBH), in which the X-axis represents the luminance spectrum and the Y-axis represents the normalized high luminance-boundary value of each luminance level in the luminance spectrum. We compute the high luminance-boundary value of a luminance level, L, using the luminous intensities that have *higher* luminous values than L and are adjacent to L. Similarly, we generate a second luminance-boundary histogram called the low luminance-boundary histogram (LLBH), which contains the normalized low luminance-boundary values. We compute the low luminance-boundary value of a luminance level, L, using the luminous intensities that have *lower* luminous values than L and are adjacent to L.

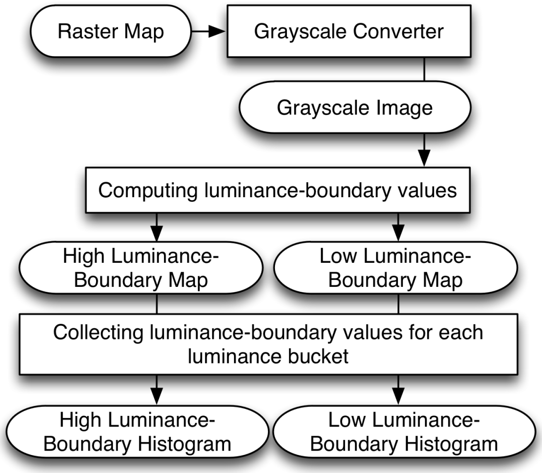

Raster Map
Grayscale Converter
Grayscale Image
Computing luminance-boundary values
High Luminance-Boundary Map
Low Luminance-Boundary Map
Collecting luminance-boundary values for each luminance bucket
High Luminance-Boundary Histogram
Low Luminance-Boundary Histogram

Figure 7: The approach to generate the high and low luminance-boundary histograms

The HLBH and LLBH are designed to capture the luminance usage of a raster map (i.e., the spatial relationships between luminance levels) by exploiting the luminous differences between adjacent luminance levels in the map. The HLBH value of a luminance level indicates a higher boundary in the grayscale histogram that separates the luminance level from its adjacent luminance levels in the raster map. Similarly, the LLBH value indicates a lower boundary. For example, a luminance level of 128 with its HLBH value as 10 implies that 138 is the average luminance intensity of the pixels that have their intensity level higher than 128 and are around pixels of luminance level 128 in the raster map. For two maps that have similar luminance usage and share the same luminance intervals to extract their foreground pixels, there are similar boundaries in the grayscale histogram to separate the two maps' foreground and background luminous intensities. Therefore, we can compare the LBH of the new input map and trained maps to determine if any of the trained map profile can be applied on the new map for extracting its foreground pixels.

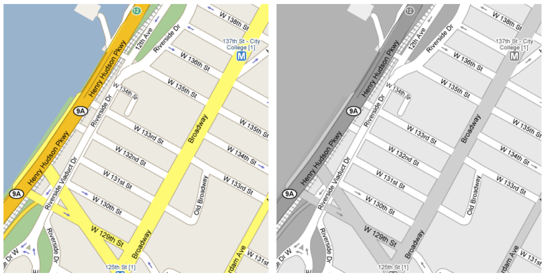
The overall approach to generate the HLBH and LLBH is shown in Figure 7. We first convert the input raster map to a grayscale image with 256 luminance buckets (i.e., the luminance spectrum is quantized to 256 levels). Then, we scan every pixel on the grayscale image to collect the high and low luminance-boundary values for each luminance bucket to generate the HLBH and LLBH. The algorithm to compute and normalize the HLBH and LLBH values from a grayscale image is shown in Table 1. The following subsections describe the algorithm in detail.

3.1.1 Extracting the Luminance Component

We extract the luminance component from the raster map by converting the map to a grayscale image. The luminance is chosen instead of using one or all of the R, G, and B components or H, S, and L components for two reasons.

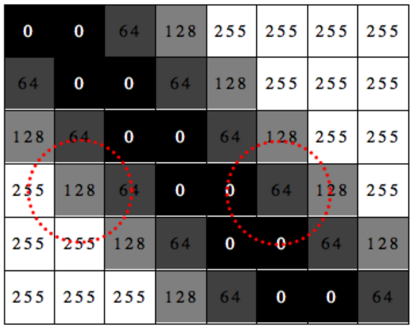
Table 1: Pseudo code for generating luminance-boundary histograms

```
/* Global variables */
Histogram $HLBH$, $LLBH$
Histogram $HLBPixelCount$, $LLBPixelCount$
GrayscaleImage $GImage$
Function GenerateHistogram()
    For each row $Y$ in the $GImage$
        For each colum $X$ in the $GImage$
            Luminance $L = GImage[X, Y]$
            $HLBH[L]$=$HLBH[L]$ + GetHLBValue(X,Y)
            $LLBH[L]$=$LLBH[L]$ + GetLLBValue(X,Y)
        End For
    End For
    /* Average the histograms*/
    Value $totalHLB$, $totalLLB$
    Foreach Luminance $L$ in the $GImage$
        $HLBH[L] = HLBH[L] \ / \ HLBPixelCount[L]$
        $totalHLB = totalHLB + HLBH[L]$
        $LLBH[L] = LLBH[L] \ / \ HLBPixelCount[L]$
        $totalLLB = totalLLB + LLBH[L]$
    End Foreach
    /* Normalize the histograms*/
    Foreach Luminance $L$ in the $GImage$
        $HLBH[L] = HLBH[L] \ / \ totalHLB$
        $LLBH[L] = LLBH[L] \ / \ totalLLB$
    End Foreach
End Function
/* Return the lowest luminance higher than the
luminance of the center pixel in a 3-by-3 area */
Function GetHLBValue(X, Y)
    Luminance $L_{HLB} = 256$
    Luminance $L_c = GImage[X, Y]$
    For Integer $I = -1$; $I < 2$; $I{+}{+}$
        For Integer $J = -1$; $J < 2$; $J{+}{+}$
            Luminance $L_n = GImage[X + I, Y + J]$
            IF($L_n < L_{HLB}$ AND $L_n > L_c$)
                $L_{HLB} = L_n$
            End IF
        End For
    End For
    IF $L_{HLB} \neq 256$
        $HLBPixelCount[L_c]{+}{+}$
        Return $L_{HLB}$ - $L_c$
    Else
        Return 0
    End IF
End Function
/* Return the highest luminance lower than the
luminance of the center pixel in a 3-by-3 area */
Function GetLLBValue(X, Y)
    Luminance $L_{LLB}$ = -1
    ...
            IF($L_n > L_{LLB}$ AND $L_n < L_c$)
    ...
    IF $L_{LLB} \neq$ -1
        $HLBPixelCount[L_c]{+}{+}$
        Return $L_c$ - $L_{LLB}$
    Else
        Return 0
    End IF
End Function
```



(a) The color map (b) The grayscale map

Figure 8: An example map from Google Maps and the map in grayscale



0	0	64	128	255	255	255	255
64	0	0	64	128	255	255	255
128	64	0	0	64	128	255	255
255	128	64	0	0	64	128	255
255	255	128	64	0	0	64	128
255	255	255	128	64	0	0	64

Figure 9: Luminance levels of sample line pixels (background pixels are shown in white)

First, using the one-dimensional features (i.e., the L component) is more computational efficient than using the three dimensional features (i.e., the R, G, and B or H, S, and L components). Second, the luminance component is the most representative component by design since most of the maps need to be readable when printed with non-color printers. For a color in the RGB domain, the luminous intensity is calculated as follows using the RMY filter:[6]

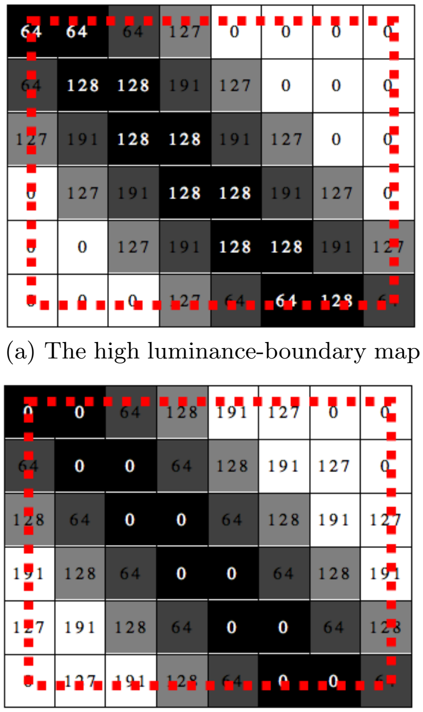
$$LuminousIntensity = R * 0.5 + G * 0.419 + B * 0.081 \quad (1)$$

Figure 8 shows a color map from Google Maps and the converted grayscale image of the map. In spite of the grayscale conversion, the objects on the grayscale maps are still recognizable, as they are in the original color map.

3.1.2 *Computing Luminance-Boundary Values*

With the grayscale image, we generate the high and low luminance-boundary maps (i.e., images contain the luminance-boundary values) by scanning each pixel in the image. For each pixel, we search on a 3-by-3 pixel neighborhood to compute the high and low luminance-boundary values for the luminance level of the center pixel. Figure 9 shows an example of a line segment in pixel-view (i.e., every cell is an image pixel), and the number on each pixel represents the luminance level of the pixel, such as 0 for black (foreground) and 255 for white (background). To compute the luminance-boundary values for the pixels identified by the dashed circles, we search their 8 neighboring pixels (i.e., in a 3-by-3 pixel neighborhood) that intersect with the circles.

[6] The RMY filter is one of the existing techniques to compute the luminance from the RGB color space. More discussions about the color space conversions can be found in [17]



(a) The high luminance-boundary map

(b) The low luminance-boundary map

Figure 10: High and low luminance-boundary maps

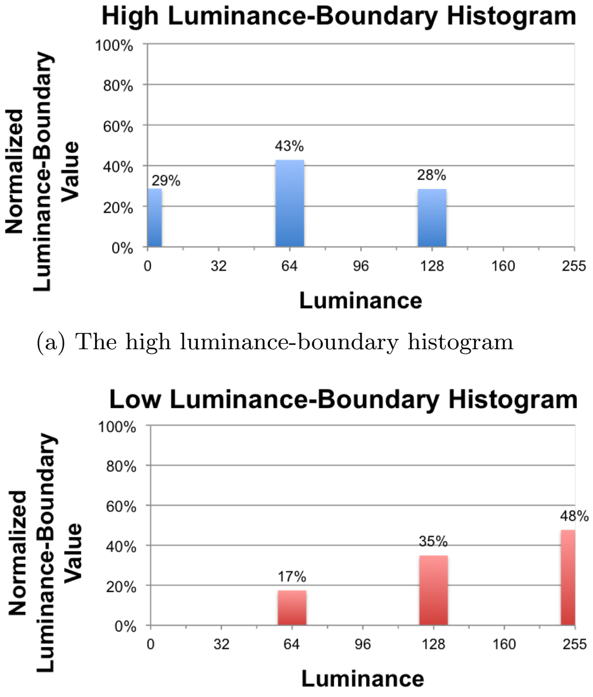
For the high luminance-boundary value, we search in the 3-by-3 pixel area to find the neighboring luminance that is higher than the luminance of the center pixel, $L(X,Y)$, and the difference between the neighboring luminance and $L(X,Y)$ is minimum (i.e., the least upper bound). For the low luminance-boundary value, we search in the same 3-by-3 pixel area to find the neighboring luminance that is lower than the luminance of the center pixel, $L(X,Y)$, and the difference between the neighboring luminance and $L(X,Y)$ is minimum (i.e., the greatest lower bound). Formally, the luminance of a pixel in image M at (X,Y) is $L(X,Y)$, and $D_{u,v\in\{-1,0,1\}}$ represent the eight luminance differences between the pixel and its neighboring pixels in the 3-by-3 area. The high luminance-boundary value of the pixel, $M(X,Y)$, is the smallest positive number in $D_{u,v\in\{-1,0,1\}}$, where

$$D_{u,v\in\{-1,0,1\}} = \{L(X+u,Y+v) - L(X,Y)\} \qquad (2)$$

If every number in $D_{u,v\in\{-1,0,1\}}$ is negative or equal to 0, the high luminance-boundary value is 0 (i.e., no neighboring pixel has a higher luminance level than the pixel $M(X,Y)$). Similarly, the low luminance-boundary value of the pixel $M(X,Y)$ is the largest negative number in $D_{u,v\in\{-1,0,1\}}$. If every number in $D_{u,v\in\{-1,0,1\}}$ is positive or equals to 0, the low luminance-boundary value is 0 (i.e., no neighboring pixel has a lower luminance level than the pixel $M(X,Y)$). Figure 10 shows the high and low luminance-boundary maps of Figure 9. The pixels that are crossed out by the dashed rectangle are the boarder pixels, which are generally not considered in any convolution type image processing because of their lack of eight neighboring pixels.

3.1.3 Normalizing Luminance-Boundary Histograms

We sum-up the luminance-boundary values for each luminance level in each of the luminance-boundary maps to generate the normalized high and low luminance-boundary histograms. The normalized luminance-boundary values of a luminance level represent the comparative importance of the luminance level in a raster map since the luminance level of a highlighted feature have comparatively high contrast against the luminance levels of the feature's neighboring pixels than other features or the background. The normalized luminance-boundary values are computed as follows:



(a) The high luminance-boundary histogram

(b) The low luminance-boundary histogram

Figure 11: The luminance-boundary histograms

Normalized high luminance-boundary value:

$$nHLBValue_i = HLBValue_i / \sum_{i=0}^{255} HLBValue_i \qquad (3)$$

Normalized low luminance-boundary value:

$$nLLBValue_i = LLBValue_i / \sum_{i=0}^{255} LLBValue_i \qquad (4)$$

We normalize the luminance-boundary value of each luminance level by dividing the each of the luminance-boundary value with the summation of the luminance-boundary value of every luminance level in the raster map. The normalization of the high and low luminance-boundary values are done separately. Figure 11 shows the two luminance-boundary histograms after we normalize the high and low luminance-boundary values of each luminance level in Figure 9.

3.2 Comparing Luminance-Boundary Histograms

To measure the similarity of two sets of luminance-boundary histograms from two raster maps, we utilize a nearest-neighbor classifier that employs a common histogram comparison metric, the L_1-distance. Given two maps and their luminance-boundary histograms *HLBH1*, *LLBH1* and *HLBH2*, *LLBH2* the L_1 distance between the two sets of luminance-boundary histograms is defined as:

$$L_1 = \sum_{i=0}^{255} |HLBH1_i - HLBH2_i| + |LLBH1_i - LLBH2_i| \quad (5)$$

A smaller distance indicates that the spatial relationships between luminance levels in one map are similar to the ones in the other map.

4. EXPERIMENTS

We compared our luminance-boundary histogram with three traditional color-based features: the color histogram, the color moments, and the color-coherence vectors. We followed the steps in [18] to implement the color histogram and color moments. Color-coherence vectors were implemented as described in [14] with the region threshold tuned to 5%. The raster maps were all quantized in the RGB color space using 16 color buckets for R, 16 color buckets for G, and 16 color buckets for B (i.e., a total of 4,096 buckets) before we generated the color histogram, color moments and color-coherence vectors.

In one experiment, we evaluated the robustness of the four features using image retrieval queries. In a second experiment, we simulated a map classification task (i.e., the map classification component shown in Figure 2) to classify raster maps based on their luminance usage by comparing the maps' image content using the four features. In both experiment settings, we used 60 test maps and an image repository with raster maps collected from various sources. We collected the 60 test maps from 11 map sources on the Internet, namely Google Maps, Live Maps, Yahoo Maps,[7] MapQuest Maps,[8] USGS topographic maps, Rand McNally,[9] Map24,[10] TIGER/Line map, OpenStreetMap,[11] Streetmap.co.uk,[12] and ViaMichelin.[13] Table 2 shows the map sources, map types, and map counts of each source. We combined two existing image repositories to create a large map repository for the two experiments. One image repository contained 1,112 raster maps identified in [13]. The other image repository has 383 images wrapped from Yahoo Image Search and Google Image Search using search keywords such as "street map" or "Los Angeles map."

For the ground truth of our experiments, we manually separated the test maps into 12 classes based on manually trained map profiles to extract the maps' foreground pixels. Initially we created a class for every map source and assigned every map to a class according to the map's source since the raster maps from the same source are most likely to share a set of thresholds to extract their foreground pixels. Next, within each class, we manually identified a luminance interval to separate the foreground pixels from a map and tested the luminance interval on every map in the class. For example, Figure 14 shows two maps from Rand McNally and their foreground pixels after we applied the same threshold setting of 0 to 190. The fact that the two maps share the same threshold setting indicates that the two maps belong to the same class in our ground truth. If one set of thresholds cannot be applied to every map in one class, we further divided the class into smaller classes and manually identified a set of thresholds for each class. Eventually the maps in one class all shared one set of thresholds to extract their foreground pixels. For example, we initially assigned all ten topographic maps from USGS to one class. Then, we further separated the ten maps into two classes since the ten maps did not share the same threshold setting to extract their foreground pixels. Figure 13 shows the foreground pixels of the topographic maps after we applied the luminance interval of 0 to 184 on Figure 4(a) and the luminance interval 0 to 36 on Figure 5(a). The manually trained map profiles for the test maps are shown in the luminance interval column in Table 2. For the map repository, we manually examined each map image. We removed the non-map images and map images that have similar color usage as the test maps or generated from the same map sources as the 11 test map sources. Therefore, besides the test maps, the map images in the repository should not be classified into any of our test classes using the four test features.

Table 2: Test maps and the ground truth

Map Source	Map Type	Map Counts	Intensity Interval
Google Maps	Digital	5	0–230
Live Maps	Digital	5	0–225
Yahoo Maps	Digital	5	0–200
MapQuest Maps	Digital	5	0–220
USGS topographic maps	Scanned	5	0–36
USGS topographic maps	Scanned	5	0–184
Rand McNally	Digital	5	0–190
Map24	Digital	5	0–215
TIGER/Line	Digital	5	0–110
OpenStreetMap	Digital	5	0–238
Streetmap.co.uk	Digital	5	0–175
ViaMichelin	Digital	5	0–234

4.1 Experiments on Image Retrieval

We first tested the robustness of the luminance-boundary histogram and the three traditional color-based features as follows: First, after every test map was inserted into the image repository, we removed one class of test maps from the repository and reinserted one test map from the removed class into the image repository as the target map. Then, we used the remaining test maps from the removed class to query the image repository in turn. The repository returned the query results by ranking the images in the repository based on their similarity to the query image. The similarity was computed by comparing the four test features in turn. Finally, we recorded the rank of the target map in the query results. For example, as shown in Tabel 2, there were five test maps from Google Maps, namely G1, G2, G3, G4, and G5. After we inserted G1 into the image repository, we first used G2 as the query image. We then recorded the rank of G1 in the returned query results. Next, we used G3 as the query image and recorded the rank of G1 in the returned query results. After G4 and G5 were both used as the query image, we removed G1 from the repository and inserted G2 into the repository. Then, we used the other four images as the query images to record the rank of G2. This process

[7] http://map.yahoo.com/

[8] http://www.mapquest.com/

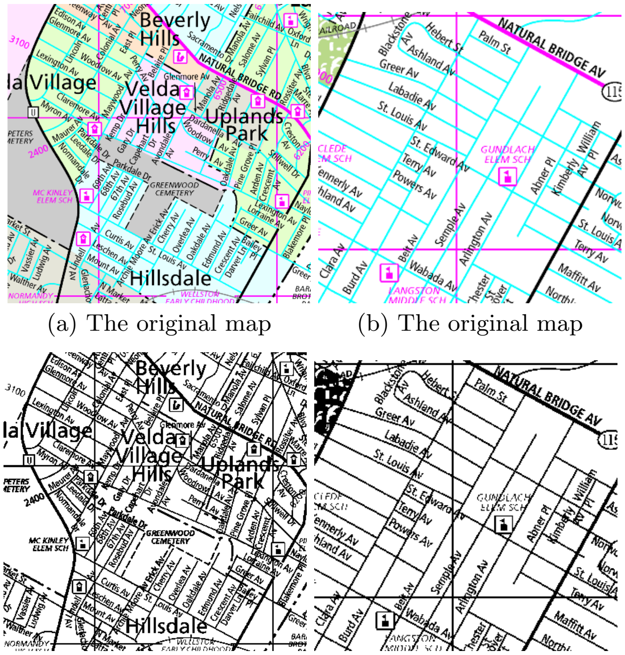
[9] http://www.randmcnally.com/

[10] http://www.map24.com/

[11] http://www.openstreetmap.org/

[12] http://streetmap.co.uk/

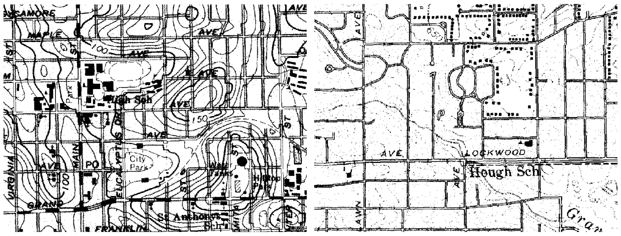
[13] http://www.viamichelin.com/



(a) The original map (b) The original map

(c) The foreground pixels (d) The foreground pixels

Figure 12: Example Rand McNally test maps and their foreground pixels



(a) El Segundo, CA (b) St. Louis, MO

Figure 13: The foreground pixels of the USGS topographic maps shown in Figure 5(a) and Figure 4(a)

was repeated until every test map had been inserted into the image repository. As a result, for the 60 images (i.e., 5 maps in a class and there are 12 classes), we conducted 240 queries (i.e., 20 queries for each of the 12 classes) for the four test features in the image retrieval experiments.

By conducting the image query for every pair of images in each class, a robust image comparison feature should have a low average rank and low variation between the ranks from all queries. Table 3 shows the average ranks and standard deviations of the results for the four test features. Figure 14 to Figure 16 show sample queries, where the ranks in the captions are presented in the order of using the luminance-boundary histogram (LBH), color-coherence vectors (CCV), the color histogram (CH), and color moments (CM).

Overall, the luminance-boundary histogram did well compared with other color-based features on both the average rank and the standard deviation. In particular, the luminance-boundary histogram and color-coherence vectors are much better and robust (both have low variances) than the color histogram and color moments since the first two features both utilize more than just color or luminance. Between the luminance-boundary histogram and color-coherence vectors, the average rank using the luminance-boundary histogram was better, and the difference is statistically significant.[14] In addition, the luminance-boundary histogram had lower variance and no tuning was required.

Table 3: Experimental results using the four image comparison features in 240 queries

Feature	Average Ranks	σ
Luminance-Boundary Histogram	5.95	24.15
Color-Coherence Vectors	15	52.14
Color Histogram	28.17	116.85
Color Moments	232.87	239.52

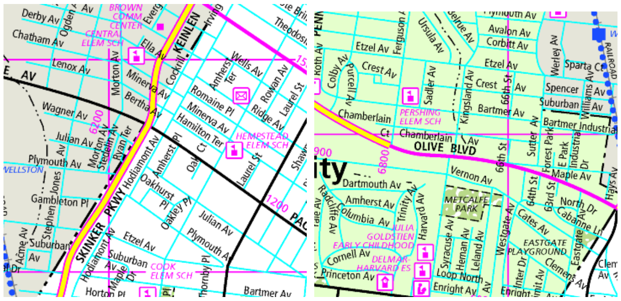
As shown in Figure 14, the luminance-boundary histogram handled cases where color-coherence vectors did not work well. Because the luminance levels of the background used in Figure 14(a) and Figure 14(b) are very different, color-coherence vectors that rely on the consistency of region sizes of the same color showed poor result on classifying these two images (other color features show even worse results). On the other hand, since the foreground pixels in the Rand McNally maps have a strong luminance level against the background in both maps, the luminance-boundary histogram worked well on finding the target map. For a more complex set of maps shown in Figure 15, the two maps share similar size of color regions but different number of colors, so color-coherence vectors did better than the color histogram and color moments. Luminance-boundary histograms had the best rank since the high and low luminance-boundary values are designed to capture the spatial relationships between adjacent luminance levels, which are similar between the two maps.

Figure 16 shows a case where the luminance-boundary histogram did not have the best result. This is because there are only a few luminance levels used in the maps, and the extra luminance levels used to draw the highways and major roads in the query image have strong high and low luminance-boundary values. The strong luminance-boundary value from the extra luminance levels lowered the normalized luminance-boundary values for the luminance levels shared by the two maps such as the ones used to draw the road boarders and the characters. This issue could be resolved if we further apply a threshold on the luminance-boundary histogram to compare only the luminance levels that have pixel counts larger than a preset percentage. However, the performance will then depends on the threshold tuning, which is similar to color-coherence vectors. Color-coherence vectors and the color histogram both did well on this query since the colors that exist in only one of the images have a small number of pixels compared to the colors shared by the two images.

4.2 Experiments on Map Classification

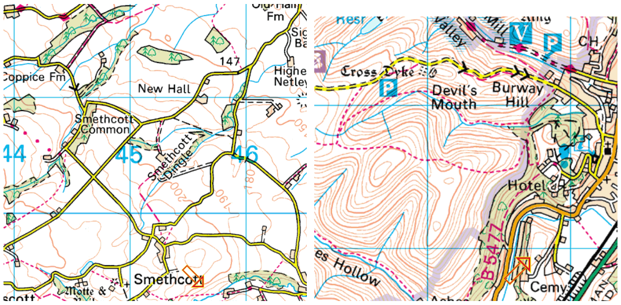
In the second experiment, we simulated a map classification task to test the classification of raster maps using the luminance-boundary histogram and the three color-based features with a nearest neighbor classifier. For an input map, the map classification component searched the image repository to find a target map that shared a trained map profile with the input map to extract their foreground pixels. The experiments worked as follows: First, after we inserted every

[14] Using a one-tailed distribution paired t-test (p=0.006)



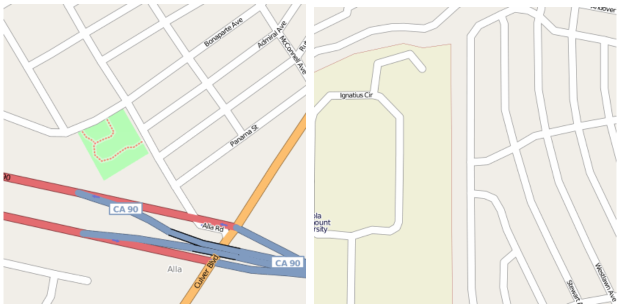
(a) The query map (b) The target map with ranks: 1/289/713/275

Figure 14: A query with Rand McNally maps (ranks are listed as LBH/CCV/CH/CM)



(a) The query map (b) The target map with ranks: 1/15/269/724

Figure 15: A query with Streetmap.co.uk maps (ranks are listed as LBH/CCV/CH/CM)



(a) The query map (b) The target map with ranks: 3/1/1/231

Figure 16: A query with OpenSteetMap maps (ranks are listed as LBH/CCV/CH/CM)

Table 4: Classification results using the four image comparison features

Feature	Accuracy
Luminance-Boundary Histogram	95%
Color-Coherence Vectors	86.67%
Color Histogram	88.33%
Color Moments	13.33%

test map into the image repository, we removed only one test map from the repository and used the removed test map as the query image. If the first returned map belonged to the same class as the query image, the classification was successful. Then, we reinserted the query image into the repository and removed another test map to test the map classification until every test map had served as the query image. For example, we first removed G1 to query the repository (i.e., G1 represents a new input map). If the first returned map was G2, G3, G4, or G5, then we had a correct classification (i.e., we successfully identified an applicable trained map profile for G1); otherwise, the classification failed. We report the classification accuracy for each feature. The accuracy is defined as the number of successful classifications divided by the total number of tested classifications.

In the previous image retrieval experiments, there was only one target map in the repository. In the map classification experiments, there were four target maps in the repository, and if the test feature ranked any of the target maps as the first map in the returned query results, the classification was successful. The enlarged pool was a simulation of a real map classification application since in practice, the size of any map class grows after we have seen more maps. In the classification results shown in Table 4, the luminance-boundary histogram had the highest accuracy of 95% among the features on classifying the raster maps in our experiments. We missed three maps in the map classification experiments using the luminance-boundary histogram. Although the enlarged pool contained more diverse maps, the three maps still suffered from the same issue as the query shown in Figure 16. We expect this problem to be resolved when there are more maps in each class as the training set becomes larger. The color histogram also benefited from the enlarged pool. Despite the fact that the color histogram had a lower average rank in the previous test, it had similar accuracy as color-coherence vectors in the map classification experiments. However, the accuracy of color-coherence vectors and the color histogram were both lower than the luminance-boundary histogram.

4.3 Efficiency

We implemented our experiments using Microsoft .Net running on a Microsoft Windows 2003 Server powered by a 3.2 GHz Intel Pentium 4 CPU with 4GB RAM. We recorded the computation time for generating the two luminance-boundary histograms and color-coherence vectors as the comparison for efficiency. Both features can be generated using a single-pass on the image. The smallest test image in pixels is 130-by-350 and the largest image is 3000-by-2422. With 1,949 images, it took 428 seconds to generate the luminance-boundary histograms and 805 seconds to generate color-coherence vectors. One of the dominant factors for the time differences was that our luminance-boundary histograms used 256 buckets while color-coherence vectors used 4,096 buckets (16 R, 16 G, 16 B). The implementations of these two features were not optimized and improvements can still be done to speed up the processes.

5. CONCLUSION AND FUTURE WORK

Identifying the foreground color or luminance in the raster map is labor intensive since the process requires user input on specifying foreground pixels in the raster maps, especially for scanned maps that have numerous colors. Instead, we present an approach to classify raster maps based on their luminance usage for reusing existing trained map profiles to extract foreground pixels from new input maps. The classification task is achieved by using the luminance-boundary histogram and a nearest-neighbor classifier to compare the image content of two raster maps. In the image retrieval experiments, the luminance-boundary histogram produced robust results compared with other traditional color features on finding maps with similar luminance usage. In the map classification experiments, the luminance-boundary histogram achieved 95% accuracy compared with the traditional color features with average ranks from 13.33% to 88.33%. In other words, we can identify an applicable trained map profile for 95% of the test maps by using the luminance-boundary histogram for comparing the input map with existing maps in the repository and hence make automatic map processing work practical. In addition, the generation of the luminance-boundary histogram is efficient without parameter tunings.

In the future, we plan to incorporate modern classifiers or off-the-shelf CBIR systems with the luminance-boundary histogram to explore the possibility of enhancing the map classification. Moreover, we intend to integrate the map classification component with our current map processing system that extracts geographic feature from raster maps. In our previous work, we presented a map processing system with an automatic technique for extracting the road pixels from simpler maps (e.g., digitally generated maps) [6] and a supervised technique for more complex maps or maps with poor image quality (e.g., scanned maps, a metro map contains various types of lines, etc.) [5]. The map processing system can process the road pixels to extract road-intersection templates [3] and then utilize the road intersection templates to extract road vectors from the raster map [4]. By integrating the map classification component described in this paper with the map processing system, we will be able to automate the processes for extracting geographic features from more diversified raster maps and fuse the maps with imagery and other geospatial data.

6. ACKNOWLEDGMENTS

This research is based upon work supported in part by the University of Southern California under the Viterbi School Doctoral Fellowship, in part by the United States Air Force under contract number FA9550-08-C-0010.

The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

7. REFERENCES

[1] R. Cao and C. L. Tan. Text/graphics separation in maps. In *Proceedings of the Fourth International Workshop on Graphics Recognition Algorithms and Applications*, pages 167–177, 2002.

[2] Y. Chen, R. Wang, and J. Qian. Extracting contour lines from common-conditioned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing*, 44(4):1048–1057, 2006.

[3] Y.-Y. Chiang and C. A. Knoblock. Automatic extraction of road intersection position, connectivity, and orientations from raster maps. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2008.

[4] Y.-Y. Chiang and C. A. Knoblock. Automatic road vectorization of raster maps. In *Proceedings of the Eighth International Workshop on Graphics Recognition*, 2009.

[5] Y.-Y. Chiang and C. A. Knoblock. A method for automatically extracting road layers from raster maps. In *Proceedings of the Tenth International Conference on Document Analysis and Recognition*, 2009.

[6] Y.-Y. Chiang, C. A. Knoblock, C. Shahabi, and C.-C. Chen. Automatic and accurate extraction of road intersections from raster maps. *GeoInformatica*, 13(2):121–157, 2008.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, 2005.

[8] J. Gelernter. Data mining of maps and their automatic region-time-theme classification. *SIGSPATIAL Special*, 1(1):39–44, 2009.

[9] A. Khotanzad and E. Zink. Contour line and geographic feature extraction from USGS color topographical paper maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):18–31, 2003.

[10] L. Li, G. Nagy, A. Samal, S. C. Seth, and Y. Xu. Integrated text and line-art extraction from a topographic map. *International Journal of Document Analysis and Recognition*, 2(4):177–185, 2000.

[11] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[12] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[13] M. Michelson, A. Goel, and C. A. Knoblock. Identifyinig maps on the world wide web. In *Proceedings of the Fifth International Conference on Geographic Information Science*, 2008.

[14] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, pages 65–73, 1996.

[15] S. Salvatore and P. Guitton. Contour line recognition from scanned topographic maps. In *Proceedings of the Winter School of Computer Graphics*, 2004.

[16] N. Sebe, Q. Tian, M. S. Lew, and T. S. Huang. Similarity matching in computer vision and multimedia. *Computer Vision and Image Understanding*, 110(3):309–311, 2008.

[17] G. Sharma and H. J. Trussell. Digital color imaging. *IEEE Transactions on Image Processing*, 6:901–932, 1997.

[18] M. Stricker and M. Orengo. Similarity of color images. In *In Storage and Retrieval of Image and Video Databases III*, volume 2420, pages 381–392, 1995.

[19] H. Tamura, S. Mori, and T. Yamawaki. Textual features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–472, 1978.