STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 1, 2010

METADATA FOR CONTENT-BASED IMAGE RETRIEVAL

ADRIAN STERCA AND DANIELA MIRON

Abstract. This paper presents an image retrieval technique that combines content based image retrieval with pre-computed metadata-based image retrieval. The resulting system will have the advantages of both approaches: the speed/efficiency of metadata-based image retrieval and the accuracy/power of content-based image retrieval.

1. Introduction

Traditionally, there are two approaches for image searching and retrieval. The first one uses pre-computed metadata and involves a textual search in this metadata and the other one uses computer vision for extracting various features from images and comparing them and so, it is a content-based search.

The first approach, also known as *metadata-based image retrieval* (i.e. MBIR), implies that images are annotated with keywords that are then stored in traditional databases for later access to the image. While metadata-based searching is the most prominent search technique used in the Intrenet (e.g. for example it is used by the Google search engine) due to its low computational cost, its many limitations led to the development of alternative methods of image searching and retrieval that would solve these limitations. The efficiency of metadata-based image retrieval equals the search efficiency in the underlying metadata database.

Therefore, the second method for searching and retrieving images is built starting from the disadvantages of metadata-based search techniques: as databases become increasingly large, a few words describing the image are not sufficient to capture the entire contents of the image . Moreover, performing annotations involves (partially or totally) the human work, which is subjective in terms of image descriptions, but also time consuming. This technique is

Received by the editors: ...

2010 *Mathematics Subject Classification.* 94A08, 68P20.

1998 *CR Categories and Descriptors.* I.4.0 [**Image Processing and Computer Vision**]: General – *Image processing software*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Search process*.

Key words and phrases. image retrieval, content-based image retrieval, metadata-based image retrieval, image feature extraction.

known as *content-based image retrieval* (i.e. CBIR). This search technique applies computer vision methods for extracting features from digital images and using these features in searching and retrieval of images. In this context, the term *content* refers to any of the characteristics of images: color, shape, texture, edge or other information that is encoded in the image itself and not in metadata.

In content-based search, the response time is large (i.e. low efficiency), but the power of the search (i.e. expressiveness of search) is high. In contrast, for metadata-based search the efficiency is high (i.e. lower response time), but the search power is rather small, since we can not fully capture the contents of an image in just a few keywords.

This paper tries to combine the two aforementioned techniques, namely content-based image retrieval and metadata-based image retrieval. More precisely, we are building metadata to help the process of image searching and retrieval, so in essence the retrieval process is metadata-based, but this metadata is extracted/built automatically and is non-textual as it contains simplified content-data from the image like color, shape, edge etc., so this makes the retrieval process a content-based one.

The paper is structured as follows. Section 2 describes the idea of our image search technique, i.e. combining metadata-based with content-based image retrieval in order to obtain an image searching technique more powerful than metadata-based image retrieval, but at the same time, less computationally expensive than content-based image retrieval. Then Section 3 presents our search algorithm, followed by Section 4 which presents the methods used for image feature extractions. Section 5 presents metrics used for comparing image features and the paper ends with evaluation tests in Section 6 and conclusions.

2. Combining Metadata-based and Content-based Image Retrieval

As we already mentioned, we developed in this paper an image searching technique which combines content-based image retrieval and metadata-based image retrieval. The main idea of our image searching technique is to use a metadata database in the search process to speed up the process, so it is primarily a metadata-based search, but this metadata is non-textual, it refers to the color content of the image (e.g. color histogram, shape, edge), so in this sense the searching technique has the power of a content-based searching technique. Also, in order to reduce the computing time of the search process, this metadata is automatically extracted when an image is added to the search set and stored in a database, so it is done separately from the searching process itself and does not add to the search time.

By combining these two approaches to image searching and retrieval, the search process will gain efficiency (i.e. lower response time) because it uses metadata for searching and does not compare images pixel by pixel, and will also gain power (due to the fact that metadata is not textual, but in the form of features extracted from the image content itself).

3. The Search Algorithm

Our image searching technique uses a relational database for storing the binary metadata of each image from the search set. The schema of this database (in fact a table from a database) should contain for each image the following type of information:

- *Average colors* - average red, average blue and average green across the image matrix pixels;
- *Color Histogram* - the vector of the color histogram of the image;
- *Edge points* - the vector of coordinates of edge points from the image; We have used the following first-order edge operators: Roberts, Prewitt, Canny.

We have considered these three types of image content features because we want the search process to be flexible. If we want the search to be efficient (i.e. low computing time), then comparing average colors of images would be the best choice. But the search would not be so accurate. In order to increase the accuracy of the search color histogram comparisons can be used. Note that this would also increase the computing time. In order to be even more accurate comparing the edge profile of images is the best choice. In section 5 several metrics for comparing image features are presented.

The search algorithm is formed by two functions: the search itself and the function for building metadata of a newly added image.

The search algorithm is no different than a common image search algorithm:

Input: $queryIMG$ - the image (or a part of an image) which is searched for in the image search set
$searchSet$ - the set of available images where $queryIMG$ is searched for

BEGIN
$meta = metadata(queryIMG)$;
for img in $searchSet$ do

$meta1 = select\_metadata(img)$;
if $compare(meta, meta1) < threshold$
echo "image found!", img;
endif;
endfor;
END

For the search operation, first the metadata is extracted from the query image (i.e. using the $metadata()$ function and then this metadata is compared against the metadata (retrieved from the database using function $select\_metadata()$) of each image from the search set. The metadata retrieved from the database was pre-computed when the respective image was added to the search set. The $compare()$ function compares various features from the metadata (average colors or histograms or edge profile). When a new image is added to the search set, the following is executed:

BEGIN
$meta = metadata(newIMG)$;
$add\_metadata(meta)$;
$searchSet = searchSet + newIMG$;
END

More precisely, the metadata of the new image is automatically extracted from the image (using function $metadata()$) and saved into the metadata database (using function $add\_metadata()$) and the image is then added to the search set.

4. Feature Extraction Operators

We have considered three types of image features: average colors, color histograms and edges.

The average red, green and blue are computed across the image's matrix pixel.

Then, the color histogram is computed as follows. First the luminance of each pixel is computed from the red (R), green (G) and blue (B) components of the pixel using the following formula:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

and then for each level of luminosity (from 0 -black to 255 -white) the frequency of occurrence of that luminosity is computed across the picture. The vector

+1	0
0	-1

(a) M^-

0	+ 1
-1	0

(b) M^+

FIGURE 1. The Roberts operator mask

of 256 levels of luminance which contains the frequency of occurrence of each luminance level in the picture is the histogram vector for the picture.

For computing edges, we considered the following first-order edge detection operators: Roberts, Prewitt, Canny. The Roberts edge operator convolves the mask from Fig.1 with the image's pixels. The value of the edge in the pixel with coordinates (x, y) is computed as follows [3]:

$$E_{x,y} = max\{|M^+ * P_{x,y}|, |M^- * P_{x,y}|\},$$

where $x, y \in 1..n$ and $P_{x,y}$ is the luminance value of the pixel with coordinates (x, y).

Prewitt operator uses the mask from Fig. 2 in order to detect edge points. After this mask is convolved with the image's pixels, the Prewitt operator computes the edge magnitude, M, and the edge direction, θ, using the following formula [2]:

$$M(x, y) = \sqrt{Mx(x, y)^2 + My(x, y)^2}$$
$$\theta(x, y) = tan^{-1}\left(\frac{My(x,y)}{Mx(x,y)}\right)$$

The Canny edge detection operator is an improved multi-stage algorithm which has the following steps [1, 6]:

- *noise reduction*: the image is convolved with the first derivative of a Gaussian; results a blurred version of the original image;
- *compute the edge gradient and direction using the Sobel operator* [4]; the direction angle is rounded to 0, 45, 90 or 135;
- *non-maximum suppression*: determine if the gradient magnitude assumes a local maximum in the gradient direction; "thin edges" are obtained;
- *threshold with hysteresis to connect edge points*;

We can see in Fig. 3 that the Canny edge operator is more accurate than Prewitt and Roberts and produces less edge points.

1	0	-1
1	0	-1
1	0	-1

(a) Mx

1	1	1
0	0	0
-1	-1	-1

(b) My

FIGURE 2. The Prewitt operator mask

5. METRICS FOR COMPARING IMAGE FEATURES

We introduce in this section the metrics used to compare average colors, histogram and edge profiles (i.e. the $compare()$ function from the search algorithm. When comparing two average colors (red, green or blue) their absolute difference is used as metric. When comparing two luminance histogram, we use the following metric, where $H1$ is the first histogram vector and $H2$ is the second histogram vector:

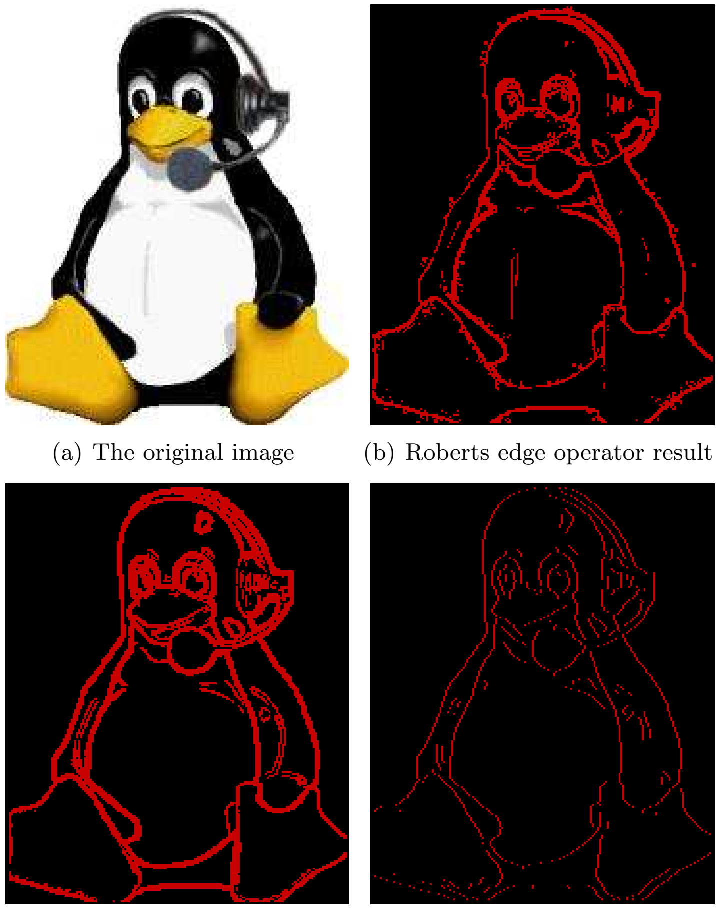
$$distance = \frac{\sum_{i=0}^{255} min(H1[i], H2[i])}{\sum_{i=0}^{255} H2[i]}$$

When we compare two edge profile (where an edge profile of an image is a vector of components (x, y, θ) where x and y are the coordinates of an edge point and θ is the direction of the edge) we use the following two metrics:

$$distance1 = \frac{1}{n}\sum_{i=1}^{n}(X_1[i] + Y_1[i]) - \frac{1}{m}\sum_{j=1}^{m}(X_2[j] + Y_2[j])$$

$$distance2 = \frac{1}{n}\sum_{i=1}^{n}\theta_1[i] - \frac{1}{m}\sum_{j=1}^{m}\theta_2[j]$$

The first metric compares positions of edge points ($(X_1[i], Y_1[i])$ is the position of an edge point in the first edge profile and $(X_1[j], Y_1[j])$ is the position of an edge point in the second edge profile) and the second one compares directions of the edge ($\theta_1[i]$ is the direction of the i-th edge point in the first edge profile). n and m are the number of edge points of the first edge profile and the second edge profile respectively. For the second metric each $\theta[i]$ is scaled



(a) The original image (b) Roberts edge operator result

(c) Prewitt edge operator result (d) Canny edge operator result

FIGURE 3. Edge detection operators

(i.e. divided) to the maximum value of direction across the image, in order to be in the interval $[0, 1]$.

6. Evaluation

We have implemented our searching technique as a PHP application [5] and evaluated the performance of the search on a search set of 10 images each having a resolution ranging from 179x218 to 376x394 pixels. Depending on the threshold we set, we were able to find in the search set the image we were looking for or similar images (when the threshold was higher). However, we must say that setting the threshold to the right value was critical for our experiments and the setting we have used might not work on other image search sets. The images from the search set contain computer-generated graphical data (like the one from Fig. 3) and real-world data (e.g. animal pictures). We have performed several search operations and the average results are displayed in the following table.

Feature used	Percent of correct results	Percent of incorrect results	No. of average comparisons per image	No. of average pixel-by-pixel comparisons	Threshold used
Average Red, Green, Blue	10%	90%	3	63240	0.04
Histogram	14%	86%	256	63240	0.4
Roberts edge profile	10%	90%	9106	63240	0.04
Prewitt edge profile	66%	33%	9851	63240	0.04
Canny edge profile	66%	33%	1836	63240	0.04

Column 1 presents the image feature used for searching (i.e. for comparing two images). Column 2 presents the percent of the result images which are correct results (i.e. similar image to the query image). Column 3 presents the percent of the result images which are incorrect results (i.e. images that are not similar to the query image). Column 4 presents the average number of metadata component comparisons (i.e. the number of components from the metadata of an image; for example, when the Average Red, Green and Blue feature is used, the number of components from the metadata of an image is 3) between two images during the search operation, while column 5 presents the same data if pixel-by-pixel comparisons would have been used for comparing two images (i.e. average width*height of all images). The last column presents the threshold value used for detecting similarity between 2 images. During the edge profile search the second metric for comparing edge profiles was used.

Regarding the efficiency of the search operation, for the image displayed in Fig. 3 if pixel-by-pixel comparison is done, 179*218 = 39022 comparisons are performed. But if the Canny edge profile of this picture is used in comparisons, only 1452 comparisons are used. This is of course an increase in efficiency.

From the values of columns 4 and 5 from the table we can see that the most efficient search is done using the Average color feature, followed by Histogram, then Canny edge profile, Roberts edge profile and Prewitt edge profile. We can see from column 4 and 5 that all these feature comparisons based search are more efficient (regarding the number of individual comparisons) that if content-based searching was used and each pixel from an image is compared with each pixel from the other image. We can see from column 2 and 3 that Canny edge profile and Prewitt edge profile comparisons produce the best results.

7. Conclusions

We have presented an image searching technique which combines metadata-based with content-based image searching and retrieval. This technique has the advantage of both approaches, namely the speed/efficiency of metadata-based search and the power of content-based search. The main idea of our technique is to construct a metadata database for an efficient search and retrieval process, but this metadata is not textual, but extracted from the content of images. This technique is more efficient than content-based searching of images with pixel-by-pixel comparisons. We have seen from the test performed that our technique produces reasonable good results during the image search operation. As future work, new metrics for comparing edge profiles can be considered and also second-order edge detection operators can be considered, for better search results.

Acknowledgment

This work was supported by the national grant CNCSIS No. PD-444, 2010.

References

[1] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679698, 1986.

[2] Prewitt, J. M. S. and Mendelsohn, M. L., *The Analysis of Cell Images*, Ann. N. Y. Acad. Sci., 128, pp. 10351053, 1966

[3] Roberts, L. G., *Machine Perception of Three-Dimensional Solids*, In: Optical and Electro-Optical Information Processing, MIT Press, Cambridge, MA, pp. 159197, 1965

[4] Sobel, I. E., *Camera Models and Machine Perception*, PhD Thesis, Stanford University, 1970

[5] Miron D., *Metadata pentru Cautarea Bazata pe Continut a Imaginilor*, Master thesis, Babes-Bolyai University, 2010.

[6] M. Nixon, A. Aguado: *Feature Extraction and Image Processing*, Second Edition, Academic Press, 2008.

Babes-Bolyai University, Faculty of Mathematics and Computer Science, 1, M. Kogalniceanu St., 400084-Cluj-Napoca, Romania
E-mail address: forest@cs.ubbcluj.ro

Fortech, 19, Meteor St., Cluj-Napoca, Romania
E-mail address: daniela2003_ro@yahoo.com