Rotation, Scale and Translation invariant image retrieval method based on Circular Segmentation and Color Density

P. Ayyalasomayajula, S. Grassi and P.-A. Farine

Ecole Polytechnique Fédérale de Lausanne, Institute of Microengineering, Electronics and Signal Processing Laboratory, EPFL IMT-NE ESPLAB, Rue A.-L. Breguet 2, 2000, Neuchâtel, Switzerland
Tel: + (41) 32 718 34 25, email: pradyumna.ayyalasomayajula@epfl.ch

Abstract— **We propose a fast and efficient method for Content Based Image Retrieval (CBIR) which uses color densities within concentric circular zones of the image, encompassing edge-pixels. This method is invariant to Rotation, Scale and Translation (RST). Small-sized feature vectors are used to store and effectively characterize the color content of the image. Consequently the memory and time required for data querying are reduced. This computationally inexpensive method is suited for portable applications. We briefly present an example of application in a handheld pictogram recognition device, used for rehabilitation and education, in which the proposed method is used as pre-selection stage of a heavier method for reducing complexity while keeping recognition accuracy.**

Index Terms— Feature extraction, Image classification, Image color analysis, Image retrieval, Content based image retrieval.

I. Introduction

The color content of an image is a representative and convenient information, which is widely exploited in Content Based Image Retrieval (CBIR) systems. Initial work in CBIR using color was done by Swain and Ballard [1] who proposed image indexing based on color histograms. This was followed by two-decades of work by the research community [2] proposing a variety of CBIR methods based on color histograms and, more recently, on different forms of region-based color characterization.

Color histograms provide useful information for CBIR, are invariant to translation and rotation, and vary very slowly to change in scale and occlusion. A color histogram is a representation of the combined distribution of colors in an image, obtained by counting the number of pixels of each of given sets of color ranges in two-dimensional (2D) or three-dimensional (3D) color spaces. 2D color spaces with no intensity information, such as rg-chromaticity or hue-saturation, are typically used. If the number of bins per dimension is Q, the length of the feature vector is Q^2. For good retrieval accuracy, the color space needs to be finely represented, with a large enough Q, yielding to large feature vector lengths, in the order of few hundreds to one hundred, the latter with drastic quantization of the color space and subsequent reduction in retrieval accuracy. Large feature vectors are impractical both in terms of storage space requirements and computational complexity.

In this paper we present an efficient image retrieval method, in which the image color characteristics are modeled using small-sized feature vectors, yielding to fast retrieval and low storage requirements. The extracted image features are the color densities of concentric circular zones encompassing the detected edge-pixels. The proposed method is invariant to Rotation, Scale and Translation (RST). As the computational load is low, the proposed method is suited for implementation as a pre-selection stage in portable image recognition devices such as “PictoBar”, as explained in Section IV.

The paper is organized as follows. The proposed image retrieval method is explained in Section II. Experimental evaluation and results are presented in Section III. Practical implementation in a handheld device is briefly presented in Section IV. Conclusions are drawn in Section V.

II. Proposed Image Retrieval Method

The proposed method is divided into Feature Extraction, and Similarity Calculation and Image Retrieval, as explained in the next subsections.

A. Feature Extraction

Feature extraction process consists of circular segmentation, concentric cropping, calculation of the color density and normalization. The resulting features are called the Color Density Circular Crop (CDCC) features.

1) Circular segmentation

We use Canny edge detector [3] on the query image I, to obtain the Edge-Set (E) containing the K edge-pixels representing all the edges in the image, as follow:

$$E \equiv \bigcup_{i=1}^{K} P_i \tag{1}$$

Where P_i is the i$^{\text{th}}$ pixel for which an edge was found, represented by its coordinates $\{x_i, y_i\}$.

Figure 1: An example of circular segmentation and concentric cropping of an image into 4 concentric regions.

We encompass all the edge-pixels with a circle of center $\{x_c, y_c\}$ and radius r_c. In order to compute the center we first calculate the extrema points of the Edge-Set as follows:

$$\begin{aligned} x_{Max} &= sup\{x_i | P_i \in E\} \\ x_{Min} &= inf\{x_i | P_i \in E\} \\ y_{Max} &= sup\{y_i | P_i \in E\} \\ y_{Min} &= inf\{y_i | P_i \in E\} \end{aligned} \qquad (2)$$

Then we calculate the center using these extrema points:

$$\{x_c, y_c\} = \left\{\frac{x_{Max} + x_{Min}}{2}, \frac{y_{Max} + y_{Min}}{2}\right\} \qquad (3)$$

We calculate the radius as the minimum distance from the center which is required to circumscribe all the K edge-pixels P_i in the Edge-Set (E):

$$r_c = sup\left\{\bigcup_{\{i|P_i \in E\}} \left\{\sqrt{((x_i - x_c)^2 + (y_i - y_c)^2)}\right\}\right\} \qquad (4)$$

2) Concentric Circular Crop

The segmented circular region, with center $\{x_c, y_c\}$ and radius r_c, is cropped into q concentric zones, Z_h, using q concentric circles described by:

$$(x - x_c)^2 + (y - y_c)^2 = \left(\frac{h\, r_c}{q}\right)^2 \qquad (5)$$

where $h = q,\ q-1,\ \ldots,\ 1$. In *Figure 1* we show an example of circular segmentation and concentric cropping of an image when $q = 4$. The four concentric circular zones Z_1 to Z_4 are shown at the right of the image.

3) Color Density Calculation

The color density of the red, green and blue channels of each of the q concentric zones is calculated, by adding the contribution of each of the pixels inside the zone:

$$R_h = \sum_{\{i|P_i \in Z_h\}} r_i$$
$$G_h = \sum_{\{i|P_i \in Z_h\}} g_i\,; \qquad B_h = \sum_{\{i|P_i \in Z_h\}} b_i \qquad (6)$$

where R_h, G_h and B_h are the color densities of the red, green and blue channels for the concentric zone Z_h, and r_i, g_i and b_i are the RGB color intensities of the pixel P_i.

4) Color Density Normalization

Finally the calculated color densities are normalized by ${r_c}^2$ and assembled together in a feature vector of length $L = 3q$:

$$\boldsymbol{f} = \bigcup_{h=1}^{q} \left\{\frac{R_h}{{r_c}^2}\,;\, \frac{G_h}{{r_c}^2}\,;\, \frac{B_h}{{r_c}^2}\right\} \qquad (7)$$

The vector $\boldsymbol{f}$ constitutes the CDCC ("Color Density Circular Crop") extracted features. These features are RST invariant: Rotation invariance is due to the use of circular zones, Scale invariance is achieved with normalization of the color densities by ${r_c}^2$, whereas translation invariance is due to the calculation of the center using the edge-pixel locations.

For a typical value of $q = 4$ we have a feature vector of length $L = 12$. As the feature vector length is very small, so are the retrieval time and the memory used for storing the pre-calculated feature vectors. Besides, the complexity of the feature calculation is low, compared with existing methods.

B. Similarity Calculation and Image Retrieval

For every image stored in the database, the feature vector $\boldsymbol{f} = \{f_1, f_2, \ldots, f_L\}$ is pre-calculated and stored. When a query image is presented, the feature vector of the query is calculated and compared with the feature vector of each of the images stored in the database, using the L^1 distance:

$$d(I_n, I_{query}) = \sum_{i=1}^{L} \left|[f_i]_{I_n} - [f_i]_{I_{query}}\right| \qquad (8)$$

where I_n is the n-th image of the database of size N. The N distances are calculated and the M smallest distances are selected and ordered:

$$d(I_{n1}, I_{query}) \le d(I_{n2}, I_{query}) \le \cdots \le d(I_{nM}, I_{query}) \qquad (9)$$

The M retrieved images are the M images $\{I_{n1}, I_{n2}, \ldots, I_{nM}\}$ corresponding to the smallest distances.

III. EXPERIMENTAL EVALUATION

In the next sub-sections we explain the databases used for experimental evaluation, testing procedure and results.

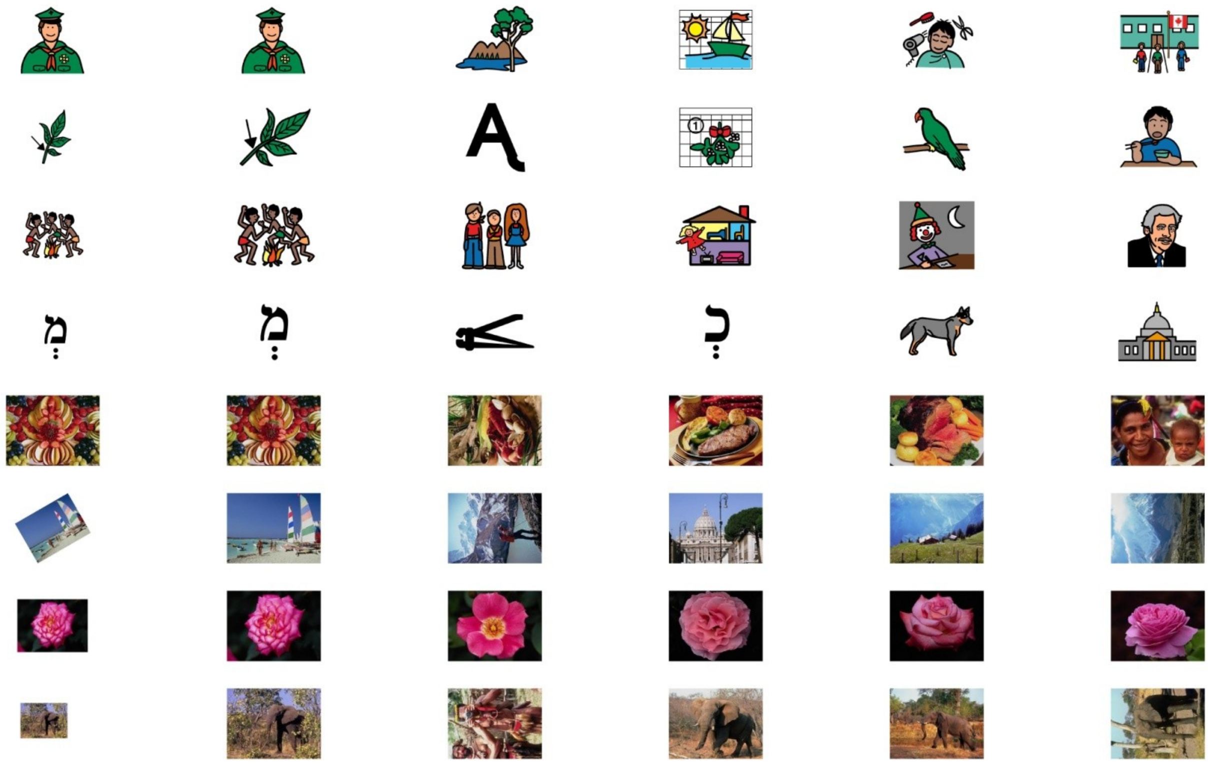

Figure 2: Examples of retrieved images with first four rows showing pictograms and the next four rows showing pictures. The first column contains the query image. Second to sixth columns contain top 5 retrieved images. The rows correspond to no RST, 30 degree rotation, 0.5 scale and (100,100) translation.

A. Experimental Databases

Two databases, one for pictograms and one for pictures were used during the tests. The pictogram database contains 4859 pictograms from the Picture Communication Symbols set [4]. The pictograms were stored in JPEG format with 85% quality, and VGA resolution (640 x 480). The picture database contains the 1000 pictures of the COREL photograph data set used in [5] which are JPEG compressed color images with QVGA (320 x 240) resolution. We refer to these two databases as the "clean pictogram database" and the "clean picture database".

Additionally, a set of 50 representative pictograms and 50 representative pictures were chosen from the clean databases. These images were printed and acquired using a fixed focus OV7675 VGA camera from Omnivision. The images were manually aligned and cropped using a printed visual reference [6]. These two sets of 50 images constitute what we call the "real condition pictogram database" and the "real condition picture database".

B. Testing Procedure

The proposed retrieval method was implemented in Matlab. For testing, we have used the two clean databases presented in Subsection III.A, and $q = 4$, which gives a feature vector length of $L = 12$. Every image within the database is successively used as query, running the search for this image on the entire database. With each query image, we run eight searches, each with different conditions: no RST, rotation by an angle θ of 30, 60 and 90 degrees, scaling by a factor s of 0.5, 0.75 and 1.25, and translation by $(\Delta x, \Delta y) = (100{,}100)$. After running each search we recorded the 30 best matches. We then calculated the image retrieval precision p as:

$$p = \frac{\sum_{k=1}^{N} V_k}{N} \tag{10}$$

where $V_k = 1$ when the query is contained in the M retrieved images, and $V_k = 0$ otherwise. N is the total number of images in the database on which the search is run. We have computed the precision for $M = 1$, 10 and 30, i.e. for the case of retrieval of the top 1, top 10 and top 30 best matches.
Similar test is done with the two real condition databases, with no RST.

C. Test results

Table 1 summarizes the test results. We can see that the query image is always found within the first 10 retrieved images, when using clean databases and within the first 30 retrieved images, when using real condition databases. Practical application of this latter result is discussed in Section IV. *Figure 2* illustrates an example of the first 5 retrieved

			Pictograms			Pictures		
			M=1	M=10	M=30	M=1	M=10	M=30
Clean databases	No RST		1	1	1	1	1	1
	Rotation	$\theta = 30^{\circ}$	0.849	1	1	0.832	1	1
		$\theta = 60^{\circ}$	0.867	1	1	0.828	1	1
		$\theta = 90^{\circ}$	1	1	1	1	1	1
	Scale	$s = 0.5$	0.979	1	1	0.729	1	1
		$s = 0.75$	0.978	1	1	0.849	1	1
		$s = 1.25$	0.991	1	1	0.965	1	1
	Translation $(\Delta x, \Delta y) = (100{,}100)$		1	1	1	0.987	1	1
	Real condition databases		0.62	0.92	1	0.68	0.86	1

Table 1: Summary of results for different tests performed.

images, for a pictogram with no RST, a pictogram with 30 degrees rotation, a pictogram scaled by 0.5 and a pictogram translated by (100,100). It also shows an example of a picture with no RST, with 30 degrees rotation, scaled by 0.5 and translated by (100,100).

1) *Computational time*

The tests were performed on MATLAB running on an Intel Core 2 Duo processor with 2.4 GHz. When the feature vector length is $L = 12$, the average time to retrieve $M = 30$ images from the $N = 4859$ pictogram database is 303.5ms. This retrieval time is divided into the time to extract the CDCC features of the query, and the time to find the best M matches from the database. The computational complexity of the CDCC feature extraction also affects the time needed for pre-calculating and storing the features for each database image. The computational time for extracting the CDCC features vary based on the image content (amount and location of the edge-pixels). The average time for extracting the CDCC features of the 4859 pictograms database is 291.5 ms. The time needed to find the M best matches is the addition of the distance calculation time and the sorting time. Distance calculation time is proportional to the feature vector length $L = 12$, and to the database size $N = 4859$. The average time taken for Distance calculation is 2.168 ms.

IV. PRACTICAL IMPLEMENTATION

The proposed CDCC retrieval method was implemented as part of a pictogram/picture recognition system called PictoBar [6], a handheld Alternative and Augmentative Communication (AAC) device used in speech rehabilitation and education. Once an image is recognized, PictoBar plays a pre-recorded sound message, associated to the recognized image. PictoBar features a DM6446 DaVinci Processor and an OmniVision OV7675 camera. The implementation of CDCC was done using Codec Engine (CE) framework from Texas Instruments. To provide uniform lighting to the image, a LED crown was mounted on the camera and the system was calibrated for any variation in color gamut, by using a calibration image.

The CDCC retrieval method is used as pre-selection stage for a heavier method, based on correlations over the DCT phase of 8x8 blocks, for reducing complexity while keeping good recognition accuracy. PictoBar was tested with the databases presented in Section III.A. Every image from this database is successively used as query, running the search for this image on the entire database. These tests resulted in 100% accuracy, validating the DSP implementation. The CDCC retrieval method performed well, always retrieving the query image in the first 30 images being sent to the next stage. The average retrieval time for CDCC was about 0.21 seconds.

V. CONCLUSIONS

We have proposed an efficient scheme for Content Based Image Retrieval based on Color Density Circular Crop (CDCC) features. These features are the normalized color densities within circular concentric zones in the image, encompassing the edge pixels. They are invariant to Rotation, Scale and Translation.

The typical size of feature vector is small ($L = 12$) and hence the memory used and retrieval time are reduced. Further, the computational load of the feature extraction is low. Accuracy of the system is good as the query image is always found in the top 10 best matches for clean database and the top 30 best matches for real condition database.

The proposed method was successfully implemented in a handheld image recognition device as pre-selection stage in conjunction with a heavier method, to decrease computational complexity while keeping good recognition accuracy.

Future work is in the direction of making the method insensitive to lighting variations, e.g. by using colors spaces with no intensity information.

ACKNOWLEDGMENTS

This work was partly supported by the Swiss Federal Office for Professional Education and Technology (OPET) through the Innovation Promotion Agency (CTI) under the Grant CTI 8811.2 PFNM-NM (“PictoBar”project).

REFERENCES

[1] M. J. Swain and D. H. Ballard, “Color Indexing”, International Journal of Computer Vision, 7, (1), pp. 11-32, 1991.

[2] R. Datta, D. Joshi, J. Li, and James Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age”, ACM Computing Surveys, vol. 40, no. 2, article 5, 2008.

[3] J. Canny, “A computational approach to edge detection”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679–698, 1986.

[4] http://www.mayer-johnson.com

[5] James Z. Wang, Li Jia and G. Wiederhold, “SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 9, pp. 947-963, 2001.

[6] P. Ayyalasomayajula, S. Grassi, P. -A. Farine, “Low complexity image recognition algorithm for handheld applications”, Proceedings of the 7th Conference on Ph.D. Research in Microelectronics and Electronics, PRIME, pp. 165-168, 2011.