CONTENT-BASED IMAGE RETRIEVAL USING STOCHASTIC PAINTBRUSH TRANSFORMATION

Zoltan Kato, Ji Xiaowen, Tamas Sziranyi, Zoltan Toth* and Laszlo Czuni**

National University of Singapore, School of Computing, 3 Science Drive 2, Singapore 117543, Fax: +65 779 4580,
**Image Processing and Neurocomputing Dept., Veszprem University, Veszprem, Hungary*
E-mail: kato@cs.ust.hk, jixiaowe@comp.nus.edu.sg, sziranyi@sztaki.hu

ABSTRACT

Herein, we propose a new content based image retrieval method. The novelty of our approach lies in the applied image similarity measure : Unlike traditional features like color, texture or shape, our measure is based on a painted representation of the original image. We use paintbrush stroke parameters as features. These strokes are produced by a stochastic paintbrush algorithm which simulates a painting process. Stroke parameters include color, orientation and location. Therefore, it provides information not only about the color content but also about the structural properties of an images. Experimental results on a database of more than 500 images show that the CBIR method using paintbrush features has higher retrieval rate than methods using color features only.

1. INTRODUCTION

Content-based image retrieval (CBIR) deals with the indexing and retrieval of images in a large database based on some low level visual features (so called visual content). The main problems in CBIR are as follows: what kind of features describe the best the visual content of an image and how to measure the similarity of two images based on these features. Color is one of the most important aspect of human visual perception which can be easily characterized by a histogram and be compared using histogram intersection [1]. However, a histogram alone cannot provide information about the spatial distribution of colors. An approach incorporating spatial correlation of colors in color correlogram was introduced in [2]. Texture [3, 4, 5] and shape [6, 7] are also useful feature used in CBIR. Shape is especially useful for foreground/background type images (containing one well defined object) [6]. However, in case of more complex images (e.g. natural scenes), shape matching can be a difficult task.

The basic question when dealing with the visual content of an image is how humans can interpret an image. In the biological sense, it is not an easy question. Brain and eye-researchers are looking for the right answer [8]. However, there is an offering answer from an artistic point of view: ask a talented painter and he will give a painted interpretation of the world: the scene as the artist sees it. Such an image is made of brush-strokes of different colors. Stochastic Paintbrush Transformation (SPT) is a new method [9] to simulate such a painting process. It is a completely automatic (no human intervention, no pre-processing) image-painting algorithm which is constructed in such a way that it provides a *multi-scale representation* of an image which is close to the *human sensation* of paintings (see [9] for some image quality measurements). In other words, it provides an *interpretation* of an image. Although this interpretation is quite specific (series of brush-strokes), it can successfully be used to capture the visual content of an image.

The next question is how to measure the similarity of two images using stroke parameters. In fact, the painting process provides a semi-segmented image: important edges are preserved and there are no fine details below a limit (determined by the brush size). However, unlike real segmentation, objects are usually covered by several overlapping strokes and the shape of a stroke does not change. The basic idea here is that similar images have similar painted representations. Hence, our measure is based on matching brush-strokes using their color, orientation and location.

2. THE PAINTBRUSH TRANSFORMATION

SPT is a stochastic process where brush strokes are generated randomly and are either accepted or rejected based on the change in distortion they introduce. Basically, it minimizes the distortion at a given brush size. The optimization algorithm is very similar to a simulated annealing (SA) [10]. In its original formulation [9], SPT is a sequential multi-scale image decomposition method, based on simulated rectangular-shaped paintbrush strokes. The resulting images look like good-quality paintings with well-defined contours, at an acceptable distortion compared to the original image. Herein, we use a slightly modified version of this algorithm: First of all, we use an elliptical-shaped brush because it provides better edges at bigger brush sizes. Second,

Fig. 1. Paintings produced by 60×15 elliptical brush strokes from 600×400 originals.

for indexing purposes, we only use the largest brush size. Hence other scales are not generated (i.e. the algorithm is mono-scale). It is also important that we use the perceptually uniform CIE-L*u*v* color space. The algorithm to produce a painted image is as follows:

Algorithm 1 (SPT)

① *Set brush size to d and initialize T_0 (SA temperature).*

② *Produce the distortion map Δ_n which is the difference image between the original image $\mathcal{I}$ and the current painting $\mathcal{P}_n$: $\Delta_n = \mid \mathcal{I} - \mathcal{P}_n \mid$ ($\mathcal{P}_0$ is an empty painting).*

③ *Compute the error image $\mathcal{E}_n$ which is a smoothed version of Δ_n, where the smoothing at each pixel is performed in a circle of diameter d.*

④ *Compute the histogram of $\mathcal{E}_n$ and set threshold ϵ such that the frequency of higher values equals to a predefined F.*

⑤ *Randomly choose a position s and brush orientation $\phi \in \{0^o, 22.5^o, 45^o, 67.5^o, 90^o, 117.5^o, 135^o, 157.5^o\}$ such that $\mathcal{E}_n(s) \geq \epsilon$ (i.e. the distortion at s is high enough). Set the brush color C to the majority vote of the original pixel colors falling inside the new brush stroke $\mathcal{B}(d, s, \phi, C)$.*

⑥ *Compute the new distortion D' and the previous (i.e. without $\mathcal{B}(d, s, \phi, C)$) distortion D over the stroke area. The new stroke is then accepted with a probability* $\min\{1, (D/D')^{1/T_n}\}$.

⑦ *Update distortion map Δ_n. If the number of strokes generated at the current temperature T_n is less than a threshold then go to Step ⑤.*

⑧ *$T_{n+1} = 0.8T_n$, $n = n+1$. Go to Step ③ until the average Δ_n in the last 10 iterations is less than a predefined value δ.*

The above algorithm produces a series of brush strokes, each of them is determined by five parameters: shape, size d, position s, orientation ϕ, and color C. The first two parameters (shape and size) do not change in our implementation. Strokes completely covered by successive strokes (i.e. invisible ones) are removed before similarity is computed. Fig. 1 shows some examples of painted images.

3. SIMILARITY MEASURE

We define similarity by matching strokes in the query image Q and a candidate image R from the database. The matching is based on the color, orientation and location of a pair of strokes. The algorithm is as follows:

Algorithm 2 (Stroke Matching)

① *Pick the next stroke from the list of Q and select those strokes from R whose color distance from the query stroke is less than a threshold $\mathcal{C}$. Note that color distance in CIE-L***u***v** *color space is obtained as the Euclidean distance of the two color vectors.*

② *Out of the candidates selected in the previous step, choose the ones with the* same orientation *as the query stroke.*

③ *Out of the strokes selected in the previous step, select the closest one to the query stroke. If a matching stroke is found then remove it from the list of R.*

④ *Remove the stroke from the list of Q and go to Step ① until list of Q is empty.*

We start the matching with the color parameter because it is the most important feature. Then we match the orientation which characterizes the structure of the image. Finally the location of the strokes are matched which tells us about the spatial location of a stroke. The algorithm produces n matching pairs of brush strokes: $\{\langle \mathcal{B}^1_Q, \mathcal{B}^1_R \rangle, \ldots, \{\langle \mathcal{B}^n_Q, \mathcal{B}^n_R \rangle\}$. We expect that each pair have similar color, same orientation and they are close to each other. Note that we do not require an exact matching and not all strokes have a matching pair. The similarity $\mathcal{S}$ of two images is defined as

$$\mathcal{S} = \frac{1}{N} \sum_{i=1}^{n} 1 - \frac{\delta(\mathcal{B}^i_Q, \mathcal{B}^i_R)}{D} \tag{1}$$

where $\delta()$ denotes the distance between the location of two strokes, N is the number of strokes in the query image Q, and D is the largest distance in the query image. Therefore the value $\mathcal{S} \in [0, 1]$ depends on the number of matching pairs (controlled by Step ① and Step ②) *and* the distance between the pairs.

4. EXPERIMENTS

The proposed method has been tested on a database of more than 500 images containing natural scenes, city scenes, buildings, human portraits, synthetic images, etc... The size of an image is around 600×400.

Fig. 2. Retrieval results using stroke matching and histogram intersection.

Method	Best	Worst	Found
Histogram intersection	221	211	67%
Proposed algorithm	339	93	74%

Table 1. Performance of each experiments. *Found* gives the average percentage of similar images retrieved by each method in the first 20 images. *Best* gives the number of images for which the given method gave the best retrieval in terms of the number of similar images found *and* the order of these images.

We have used the following parameters in Algo. 1: The brush size d was 60×15, initial temperature $T_0 = 0.8$, frequency threshold $F = 0.6$, the number of strokes generated at a given temperature was given by $10D/d$ (D is the size of the image), and δ was 20.0. The color distance threshold $\mathcal{C}$ in Algo. 2 has been set to 5.0. We tuned these values on a small set of training images and then used them during our tests.

For the evaluation, we have used a ground truth produced by manual classification of images. For each image, the ground truth data contains all of the similar images found in the database. Similarity is judged by a user. Although this is inherently subjective, the ground truth data shows a rather strict similarity requirement. The whole database together with the ground truth are available on our website (`www.cs.ust.hk/~kato/reserach/spt/cbir/`).

For comparison, we have also implemented global histogram intersection [1]. For this purpose, the CIE-L*u*v* color space is quantized such that the L channel is divided into 6 intervals, the U channel is divided into 14 intervals, and the V channel is divided into 10 intervals. These values have been manually tuned based on the quality of retrieval. The histogram of an image is then computed in this quantized space.

For each query image, we retrieved the 20 most similar images, ordered by similarity, from the database. To measure the performance of the algorithm, two criteria are considered: 1) How many similar images are retrieved; 2) the order of the retrieved similar images (i.e. whether the similar images can be found at the beginning of the list). For all methods, the evaluation is based on the same ground truth data. Table 1 shows that similarity based on stroke parameters has higher retrieval rate than traditional color-based similarity. It is also clear that the SPT-based method returns relevant images at the beginning of the list (it has been outperformed in only 93 cases, and it performed better in more than 200 cases). This is an important issue when less than 20 images are retrieved.

Concerning the computing time on a Pentium III 930MHz PC, both methods need less 1 sec to decide whether two images are similar. Therefore retrieval time is quite fast. However, indexing time is higher as the SPT algorithm is quite CPU intensive. It takes about 20-30 minutes on 600×400 images. We also note that indexing can be done on a small thumbnail (with proportionally smaller brushes) instead of the original image which requires considerably less CPU time: about 4.6 sec for images of 64×64, 15.6 sec for 128×128 and 145.6 sec for 256×256. The retrieval quality of the thumbnail images is the same as for the original 600×400 images.

Fig. 3. Retrieval results using stroke matching and histogram intersection.

5. CONCLUSION

We have presented a new image similarity measure based on a painted representation of the original image. Unlike traditional CBIR methods, we use brush-stroke parameters as features and our measure is computed by matching strokes in a pair of images. The advantage of our method is that it provides information about both color and structural properties of an image. The algorithm has considerably higher retrieval rate compared to traditional color-based features.

6. REFERENCES

[1] M. J. Swain and D. H. Ballard, “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[2] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, “Image indexing using color correlograms,” in *IEEE Conference Computer Vision Pattern Recognition*, 1997, pp. 762–768.

[3] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[4] F. Liu and R. W. Picard, “Periodicity, directionality, and randomness: wold features for image modeling and retrieval,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 722–733, July 1996.

[5] Yu Zhong and Anil K. Jain, “Object localization using color, texture and shape,” *Pattern Recognition*, vol. 33, no. 4, pp. 671–684, Apr. 2000.

[6] Anil K. Jain and Aditya Vailaya, “Shape-based retrieval: A case study with trademark image databases,” *Pattern Recognition*, vol. 31, no. 9, pp. 1369–1390, Sept. 1997.

[7] P. Pala and S. Santini, “Image retrieval by shape and texture,” *Pattern Recognition*, vol. 32, no. 3, pp. 517–527, Mar. 1999.

[8] F. Werblin, A. Jacobs, and J. Teeters, “The computational eye,” *IEEE Spectrum*, pp. 30–37, 1996.

[9] T. Sziranyi and Z. Toth, “Random paintbrush transformation,” in *15th ICPR*, Barcelona, Spain, 2000.

[10] P.J. Van Laarhoven and E.H. Aarts, “Simulated annealing : Theory and applications,” *Reidel Pub., Dordrecht, Holland*, 1987.