Non-Photorealistic Rendering and Content-Based Image Retrieval

Xiaowen Ji
Department of Computer Science
National University of Singapore, Singapore 117543
Fax: +65 6779 4580
jixiaowe@comp.nus.edu.sg

Zoltan Kato
Department of Informatics
University of Szeged, Hungary
P.O.Box 652, H-6701 Szeged, Hungary
Fax:+36 62 420 292
kato@inf.u-szeged.hu

Zhiyong Huang
Department of Computer Science
National University of Singapore, Singapore 117543
Fax: +65 6779 4580
huangzy@comp.nus.edu.sg

Abstract

In this paper, we will show how non-photorealistic rendering (NPR) can take a new role in content-based image retrieval (CBIR). The proposed CBIR method applies a novel image similarity measure: Unlike traditional features like color, texture, or shape, our measure is based on a painted representation of the original image. This is produced by a stochastic paintbrush algorithm which simulates a painting process. We use the stroke parameters (color, size, orientation, and location) as features and similarity is measured by matching strokes of a pair of images. The advantage of our approach is that it provides information not only about the color content but also about the structural properties of an image without the segmentation of the image. Experimental results show that the CBIR method using paintbrush features has higher retrieval rate than traditional methods using color or texture features only.

1. Introduction

Non-photorealistic rendering (NPR) refers to any techniques which can produce a non-photorealistic image. There are three distinct types: direct rendering of 3D scenes ([13]), transformation from a photo ([10, 28, 23, 12, 6, 11, 19]), and interactive drawing ([29, 8]). A common process for the second type methods is understanding the input images where the techniques of image processing and computer vision can be applied. In [10], a method was proposed for creating an image with a hand-painted appearance from a photograph. An image is painted with a series of spline brush strokes. Brush strokes are chosen to match colors in a source image. A painting is built up in a series of layers, starting with a rough sketch drawn with a large brush. The sketch is painted over with progressively smaller brushes, but only in areas where the sketch differs from the blurred source image. Thus, visual emphasis in the painting corresponds roughly to the spatial energy present in the source image. The Stochastic Paintbrush Transformation (SPT) [28] 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 [28] 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 just like a painting can capture the world around us. A more general survey of NPR can be found in [3, 7, 26].

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 [27]. However, a histogram alone cannot provide information about the spatial distribution of colors. An approach

Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG'03)
0-7695-2028-6/03 $17.00 © 2003 IEEE

incorporating spatial correlation of colors in color correlogram was introduced in [14]. Texture [22, 21, 32] and shape [16, 24] are also important features used in CBIR. While the extraction of shape features is not an easy task (one has to segment the image), they are particularly useful for foreground/background type images (containing one well defined object) [16, 2]. Such approaches are also referred to as semantic feature retrieval methods ([4]). Recent research on semantic feature retrieval has tended to concentrate on one of two problems. The first is scene recognition ([25]). The second is focus on object recognition ([5], [9]). However, in case of more complex images (e.g. natural scenes), automatic shape extraction and matching can be a difficult task. One nice work using shape features is Blobworld [2], where a query is formulated in terms of regions (blobs) and region properties (color, texture). By finding image regions which roughly correspond to objects, Blobworld allows querying at the level of objects (semantic information) rather than global image properties. While this is quite powerful when searching for images containing a specific object, other applications may require a search strategy based on the overall visual impression of the whole image (like a stock photographer's database or other artistic image collections). Region-based queries are oriented towards questions like *show me images containing an object like this* while a search based on visual content would be *show me more images looking like this*. Our approach tries to solve the latter problem.

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 [31]. 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 sizes and colors put on the screen one after the other in a sequence by the painter. Small details are elaborated with fine brushes, while plain surfaces are painted with greater strokes. Such painting processes have been extensively studied in NPR. In both NPR and CBIR, one of the key problems is how well one can produce results (either retrieval or painting) which corresponds to human perception. We adapted the Stochastic Paintbrush Transformation (SPT) method [28] in our work because it simulates such a painting process. Note that our CBIR method is quite general so other NPR methods could also be used.

The remaining part of the paper is organized as follows: first, we describe an SPT method that we have implemented in our CBIR method (Section 2). It is based on the work proposed in [28] with some improvements. Then, we propose the CBIR method in Section 3. The painting representation derived from the SPT method is directly used to compute the similarity value between two images. In Section 4, we brief the implementation and show the experiment results. For experiment study, we have compared our method with Global Histogram Intersection (GHI) and Oracle's CBIR functions on a database of 1,017 images. The results show that the method has a better performance. Finally, we conclude this paper in Section 5 with discussion of the future work.

2. A Stochastic Paintbrush Transformation Method

SPT [28] is a stochastic process where brush strokes are generated randomly at decreasing scales of brush-sizes. The strokes are then either accepted or rejected based on the change in distortion they introduce. Basically, it can be regarded as a multi-scale image decomposition method, based on simulated, arbitrarily shaped paintbrush strokes. A stroke is determined by five parameters: shape, size, location, orientation and color. Herein, we use rectangular-shaped brushes. Brush size is decreased at every painting stage. Color is determined by the majority color in the stroke area of the original image while location and orientation are randomly generated. It is also important that we use the perceptually uniform CIE-$L^*u^*v^*$ color space [15]. A new stroke is accepted according to a probabilistic rule which prefers strokes decreasing the distortion of the painting. The resulting images look like good-quality paintings with well-defined contours, at an acceptable distortion compared to the original image (See Fig. 6).

The original algorithm [28] has been modified and improved in several places for CBIR purposes:

First, the stroke color policy at large brush size, which sets stroke color to the majority-vote color in the stroke-area of the original image, is extended to all brush sizes. This modification makes the edge information in the original image better preserved in the painting.

Second, Simulated Annealing is used in SPT to control the production and acceptance of strokes. Basically, our painting algorithm is formulated as an optimization problem which is solved by simulated annealing (SA) [17, 18]. Note that SA is proven to find the global optimum, hence our painting should be the one with the lowest distortion given the set of brushes used during the process: The painting process in a stage (i.e. at a brush size) is a SA process and can be separated into iterations, with each iteration a Metropolis Monte Carlo simulation [17]. A randomly produced stroke in an iteration circle is a Monte Carlo step. The energy of the system, i.e. the painting process, relates to the distortion value between the current painting and the original image. Thus we define the energy difference on the acceptance of the stroke is $log(D') - log(D)$. D is the current distortion value over the stroke area between the original image and the current painting; D' is the distortion value over the stroke area with the introduction of the stroke. A stroke is accepted with the probability:

$$\min\{1, (D/D')^{1/T_n}\}, \tag{1}$$

where T_n is the Simulated Annealing temperature which decreases when iteration goes on. It is easy to verify that Eq. (1) accords with the acceptance rule in Metropolis Monte Carlo simulation.

Third, distortion-map Δ_n is refreshed after every acceptance of new stroke instead of after an iteration in the original algorithm. This makes the painting process use information brought by new strokes as soon as possible.

Last, the average error-summation of the distortion-map Δ_n in the last 10 iterations is used to control quality of the painting process, instead of the error-summation of the distortion-map Δ_n in the last iteration. This makes the quality of the painting more guaranteed. The modified SPT rendering algorithm is as follows:

Algorithm 1 (SPT)

① *Set brush size d to the next smallest brush 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.*

⑨ *Go to Step ① until the smallest brush size is over. Note that now $\mathcal{P}_n$ in Step ② will be the final painting obtained at the current level. The painting simply continues with a smaller brush!*

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. Δ_n controls the quality of the painting: Each pixel in Δ_n is the color distance between the corresponding pixel in $\mathcal{I}$ and $\mathcal{P}_n$. We use the sum of Δ_n to trace the quality of $\mathcal{P}_n$. $\mathcal{E}_n$, ϵ, and F are used to control the painted area at each iteration: For a more efficient stroke generation, we want to modify the image area having higher distortions in the current painting. Each pixel in $\mathcal{E}_n$ tells the mean distortion value in a circle of radius d (current brush size) around it. With the histogram of $\mathcal{E}_n$ and F, we can compute the threshold ϵ such that new strokes are only produced in the upper F percent of highly distorted area.

2.1. Properties of a Painted Image

Paintings produced by SPT have several nice properties. Herein, we will look at closely those that are particularly interesting for CBIR.

- A painting is equivalent to a series of brush strokes. Hence SPT can be seen as a *transformation* of the original image into a sequence of brush parameters. There is a well-defined scale-space line here: First large details, then finer details are proceeded (just like a real painter would do). We also note that many aspects of human perception also works according to the scale-space paradigm [20].
- By looking at the paintings (see Fig. 6), one can immediately see that paintings have sharp edges and important boundaries are kept at any brush size. Furthermore, there are no fine details below a limit determined by the actual brush size. It is also clear that one can get arbitrarily close to the original image by using smaller brushes. Hence the applied brush sizes determine the *level of abstraction*.
- The following observations are particularly important for CBIR: 1. Every part of the image is painted with the largest possible brush (Fig. 1). 2. Independently of the brush size, density of strokes around edges is much higher than in other parts of the image (Fig. 2). 3. Stroke orientation is in correlation with the structural properties of the image (Fig. 3).

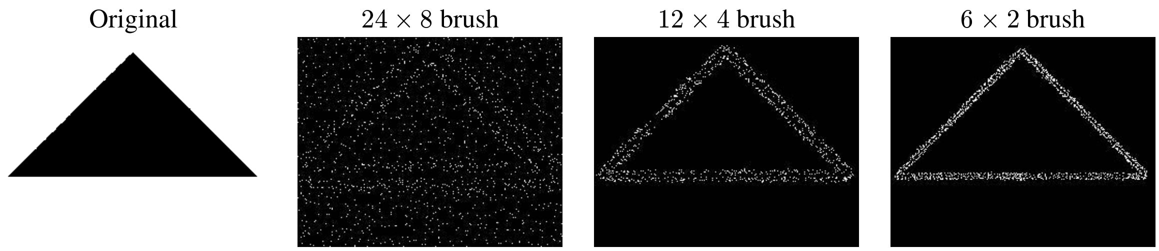

Figure 1. In subsequent stages, accurately covered areas are not repainted with smaller brushes, i.e. every part of the image is painted with the largest possible brush. White dots denote the center of brush strokes.

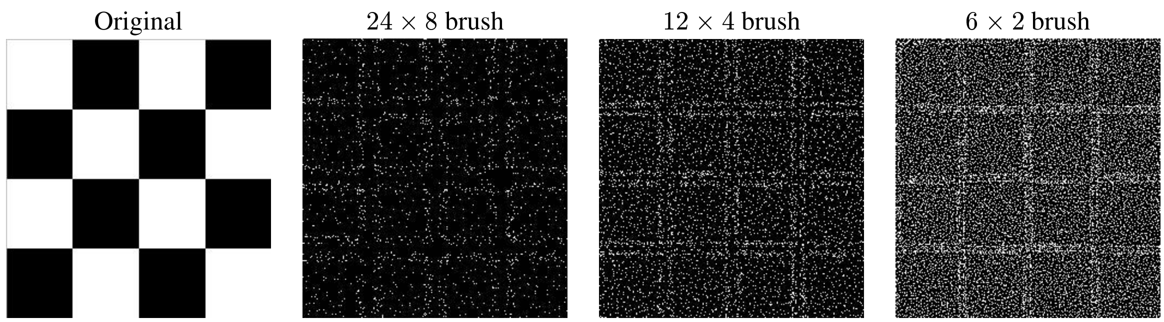

Figure 2. Stroke density is much higher around boundaries independently of the brush size. Note that for clarity of presentation, each of the above paintings were produced by a separate SPT run with a single brush size.

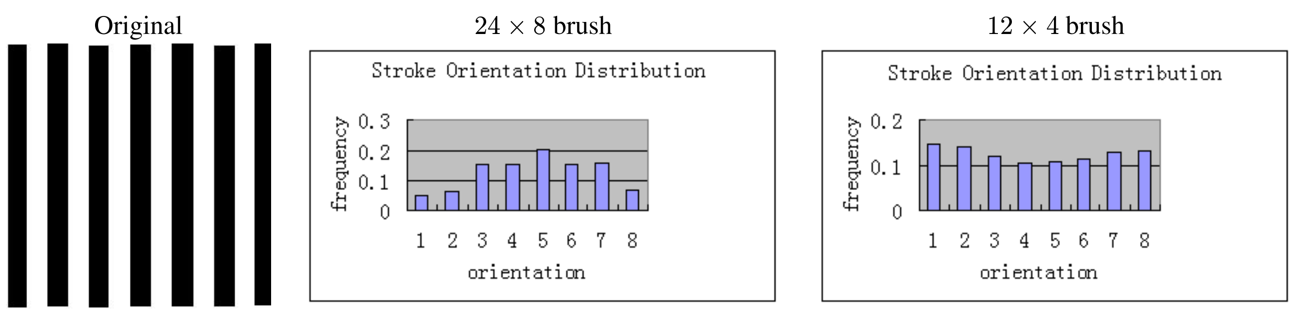

Figure 3. The distribution of stroke orientations is determined by the structural properties of the image: orientation of longer strokes is in correlation with the direction of strips of the image (vertical directions preferred) but orientation of shorter strokes are uniform because they can be placed equally well in any orientation.

In a painting produced by SPT, there might be many strokes that are completely covered by succeeding strokes. Such invisible strokes are redundant because they do not carry any additional information. Therefore they are removed before similarity is computed.

3. The CBIR Method

In this section, we present our content-based image retrieval (CBIR) method. The novelty is that it is based on the painted representation of image, obtained by Stochastic Paintbrush Transformation (SPT), which automatically simulates painting process. We will show how the color and structure information can be used. Then, we will show how semantic meanings can also be combined.

For two images I_1 and I_2, we have their stroke sequences S_1 and S_2 derived from SPT. The information associated with each stroke is size, orientation, position, and color. Note that each sequence is already sorted by its size due to our SPT algorithm. For our purpose of retrieval, we prioritize the information in a decreasing order: We start with matching 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. Thus, among the same size strokes of one painting, we first sort them by their color, then the orientation, and finally the position.

Now the similarity value $sim(I_1, I_2)$ between images I_1 and I_2 is derived by the comparison of the stroke sequences S_1 and S_2. The process can be described as follows:

Algorithm 2 (Compute the Similarity Value)

① *Pick two same size strokes s_1 and s_2 respectively from stroke sequences S_1 and S_2.*

② *Compute the similarity values $sim_{col}(s_1, s_2)$, $sim_{ori}(s_1, s_2)$, and $sim_{pos}(s_1, s_2)$ of color, orientation, and position between the strokes s_1 and s_2.*

③ *Add the 3 similarity values $sim_{col}(s_1, s_2)$, $sim_{ori}(s_1, s_2)$, and $sim_{pos}(s_1, s_2)$ with the weights w_{col}, w_{ori} and w_{pos} together, the similarity value contributions of the stroke s_1 and s_2, to the similarity value $sim(I_1, I_2)$ of images I_1 and I_2.*

④ *Repeat steps Step ① to Step ③ until running out of the same size strokes from S_1 or S_2.*

⑤ *Count the number n of the remaining same size strokes of S_1 or S_2 and remove them. This number n is used to adjust the similarity value $sim(I_1, I_2)$. The smaller the number n, the higher the similarity value of these I_1 and I_2.*

⑥ *Repeat steps Step ① to Step ⑤ using the smaller size strokes until running out of the strokes in S_1 or S_2.*

⑦ *Adjust the similarity value $sim(I_1, I_2)$ by the number n, the number of the remaining strokes in S_1 or S_2.*

⑧ *End.*

The similarity values $sim_{ori}(s_1, s_2)$ and $sim_{pos}(s_1, s_2)$ of orientation and position are computed in joint space and Cartesian space respectively. When the two orientation angle difference $\Delta\theta$ (distance d) of strokes s_1 and s_2 is 0^o (0 unit), its orientation (position) similarity value $sim_{ori}(s_1, s_2)$ ($sim_{pos}(s_1, s_2)$) equals to 1.0 (1.0). The orientation angle difference (distance) is bounded by 180^o (the diagonal length of the image). This will lead to the orientation and position similarity values in a range of $[0.0, 1.0]$. The similarity value $sim_{col}(s_1, s_2)$ of two colors can be computed in CIE-L*u*v* color space [15]. It is a 3-D space, so the computing is in a similar way as computing the position similarity value in Cartesian space. Their weights w_{col}, w_{ori} and w_{pos} to the overall similarity value $sim(I_1, I_2)$ are determined in the experiment tuning.

Finally, to show how semantic meanings can be combined, we applied the CBIR method by regions in images. The image segmentation method used in BLOBWORLD ([1, 2]) was applied to obtain regions. The strokes of an image were separated into groups such that each group corresponds to a region and centroids of the strokes in a group are all located in the corresponded region. The similarity between two images, $\mathcal{Q}$ and $\mathcal{I}$, is measured by Algorithm 3 listed as follows:

Algorithm 3 (Semantic Measurement)

① *While not all regions in $\mathcal{Q}$ have been selected as foreground.*

  ⓘ *While not all regions in $\mathcal{I}$ have been selected as foreground.*

    ⓐ *Select the next region in $\mathcal{Q}$, regard it as the foreground and the remaining regions as the background.*

    ⓑ *Select the next region in $\mathcal{I}$, regard it as the foreground and the remaining regions as the background.*

ⓒ *Compute the similarity $\mathcal{S}_f$ between the foregrounds and the similarity $\mathcal{S}_b$ between the backgrounds. The similarity between $\mathcal{Q}$ and $\mathcal{I}$ at this configuration is computed as $\mathcal{S} = \frac{2}{3}\mathcal{S}_f + \frac{1}{3}\mathcal{S}_b$ and $\mathcal{S}$ is stored.*

② *Choose the maximum S stored at Step ⓒ as the similarity between $\mathcal{Q}$ and $\mathcal{I}$.*

③ *End.*

4. Implementation and Experiment Study

In our experiments, we have used an image database containing 1,017 images including human portraits, natural scene, city scene, rural scene, synthetic images, etc. The original size of these images is around 600×400 or 400×300. They were resized into a bounding box of 256×256 while keeping the original ratio for SPT transformation. SPT went through 3 stages for every image. The brush size d used in each stage were respectively 24×8, 12×4 and 6×2. We have used the following weights for computing similarity: $w_{col} = 0.40$, $w_{ori} = 0.35$ and $w_{pos} = 0.25$. 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 web site `www.cs.ust.hk/~kato/reserach/spt/cbir/`

For comparison, we have also implemented global histogram intersection (GHI) [27]. 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. This partition makes a cube with the same length in each channel. The histogram of an image is then computed in this quantized space. Finally, the proposed method has also been compared to Oracle 9i's CBIR function [30] which indexes images on four attributes: global color, local color, texture and shape. The similarity between two images is a weighted sum of similarity scores on these four attributes. Users determine the weights of these four attributes. In our experiments the weights were set as $globalcolor = 0.4$, $localcolor = 0.1$, $texture = 0.4$ and $shape = 0.1$, as this setting gave the best performance on the database.

For each query image, we retrieved the 20 most similar images, ordered by similarity, from the database. To measure the performance of the algorithm, the commonly used precision rate and recall rate were computed. The resulting precision-recall graph on Fig. 4 shows that similarity based on stroke parameters has higher retrieval rate than traditional color-based similarity or Oracle 9i's CBIR function. We also found that the SPT-based method returns relevant images at the beginning of the list (see Fig. 5). This is an important issue when less than 20 images are retrieved.

Concerning the computing time on a Pentium III 930MHz PC, our method needs less than 500 msec to decide whether two images are similar. Therefore retrieval time is quite fast. However, indexing time is higher as the SPT algorithm is more CPU intensive. Fortunately, indexing is done offline and only once, when the image is entered into the database. It takes about 300 sec on 256×256 images. However, 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 10 sec for images of 64×64, or 30 sec for 128×128. The retrieval quality of the thumbnail images is the same as for the original 256×256 images.

5. Conclusion and Future Work

In this paper, we have explored a new role of non-photorealistic rendering (NPR) in content-based image retrieval (CBIR). We have proposed 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 without segmentation. The algorithm has considerably higher retrieval rate compared to traditional color or texture based features.

For future work, we will do more experiment study. We can try other NPR methods, compare with other CBIR methods in particular, the methods based on multiresolution images, and test on more image databases. Another improvement is to make our method to be orientation invariant. One straight forward way is by rotating the statistic data of stroke orientation, e.g., when using histogram intersection to compare orientation histograms, we can shift the orientation histograms of index image 8 times (we have only 8 choices a stroke) and choose the maximum value of the intersection operation in the 8 shifts to be the similarity between two histograms.

6. Acknowledgment

The work was partially supported by NUS research grant R252-000-085-112. The first author was supported by the NUS scholarship for graduate study. The second author has

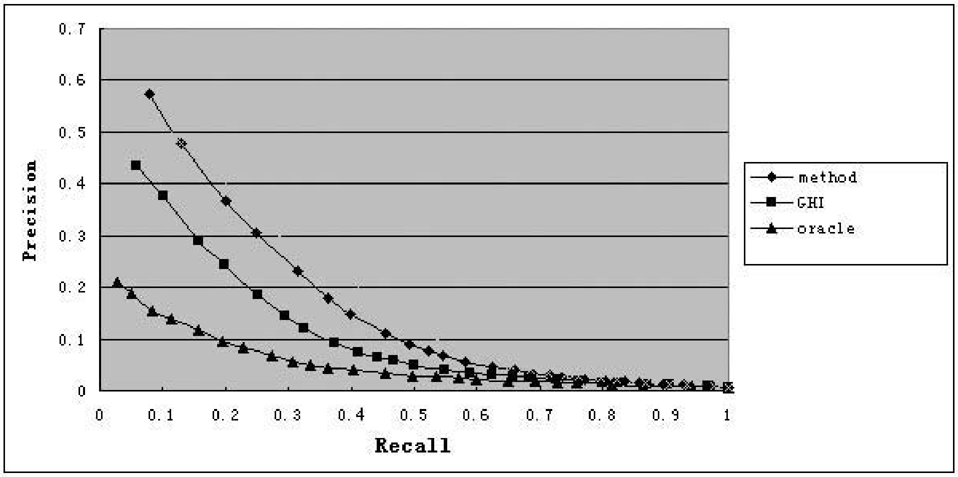

Figure 4. Precision-recall curves of the proposed method, GHI, and Oracle CBIR function.

been partially supported by the Janos Bolyai fellowship of the Hungarian Academy of Sciences. We also thank the comments of the PG 2003 anonymous reviewers.

References

[1] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[3] C. Curtis, A. Gooch, B. Gooch, S. Green, A. Herzmann, P. Litwinowicz, D. Salesin, and S. Schofield. Non-photorealistic rendering. *Course Notes SIGGRAPH'1999)*, 1999.

[4] J. P. Eakins and M. E. Graham. Content-based image retrieval – a report to the jisc technology applications programme. Technical report, Institute for Image Data Research, University of Northumbria at Newcastle, UK, January 1999. http://www.unn.ac.uk/iidr/CBIR/report.html.

[5] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. K. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. In *Object Representation in Computer Vision*, pages 335–360, 1996.

[6] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: Painterly rendering using computer vision techniques. In *Non-Photorealistic Animation and Rendering 2002 (NPAR '02)*, pages ?–?, Annecy, France, June 2002.

[7] B. Gooch and A. Gooch. *Non-Photorealistic Rendering*. Peters, A K, Limited, 2001.

[8] P. E. Haeberli. Paint by numbers: Abstract image representations. *Computer Graphics, (Proc.SIGGRAPH'1990)*, pages 207–214, 1990.

[9] N. Haering, Z. Myles, and N. Vitoria. Locating deciduous trees. In *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 18–25, 1997.

[10] A. Hertzmann. Painterly rendering with curved brush strokes of multiple size. *Computer Graphics (Proc.SIGGRAPH'1998)*, pages 453–460, 1998.

[11] A. Hertzmann. Fast paint texture. In *Non-Photorealistic Animation and Rendering 2002 (NPAR '02)*, pages 91–96, Annecy, France, June 2002.

[12] A. Hertzmann, C. E. Jacobs, N. Oliver, and B. Curless. Image analogies. In *Computer Graphics, (Proc.SIGGRAPH'2001)*, pages 327–341, Annecy, France, Aug. 2001.

[13] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. *Computer Graphics (Proc.SIGGRAPH'2000)*, pages 23–28, 2000.

[14] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *IEEE Conference Computer Vision Pattern Recognition*, pages 762–768, 1997.

[15] A. K. Jain. *Fundamentals Of Digital Image Processing*, chapter 3 Image Perception, pages 49–79. Prentice Hall, 1989.

[16] A. K. Jain and A. Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, Sept. 1997.

[17] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[18] P. V. Laarhoven and E. Aarts. *Simulated annealing : Theory and applications*. Reidel Pub., Dordrecht, Holland, 1987.

[19] N. Li and Z. Huang. A feature-based pencil drawing method. *Proc. Graphite'2003*, pages 135–140, 2003.

[20] T. Lindeberg and B. M. H. Romeny. *Linear Scale-Space I-II*. Kluwer Academic Publishers, 1992.

0-7695-2028-6/03 $17.00 © 2003 **IEEE**

Query image

Ground truth	SPT	GHI	Oracle

Figure 5. Example 1: retrieval results of the proposed method, histogram intersection, and Oracle's CBIR function.

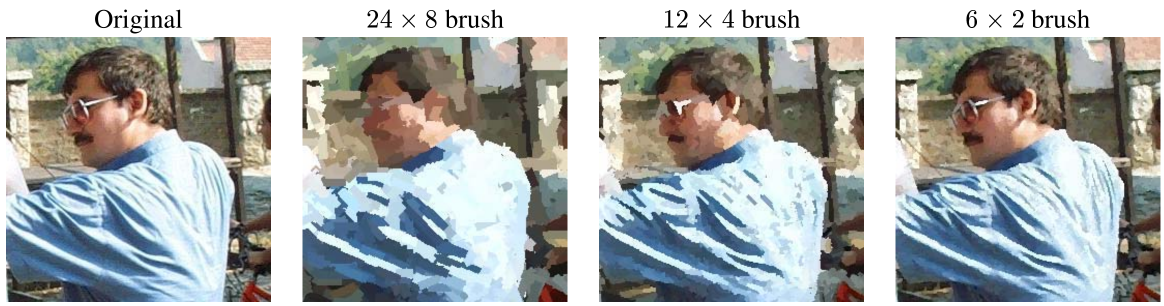

Figure 6. Typical paintings produced by SPT from a real color image with different brush sizes.

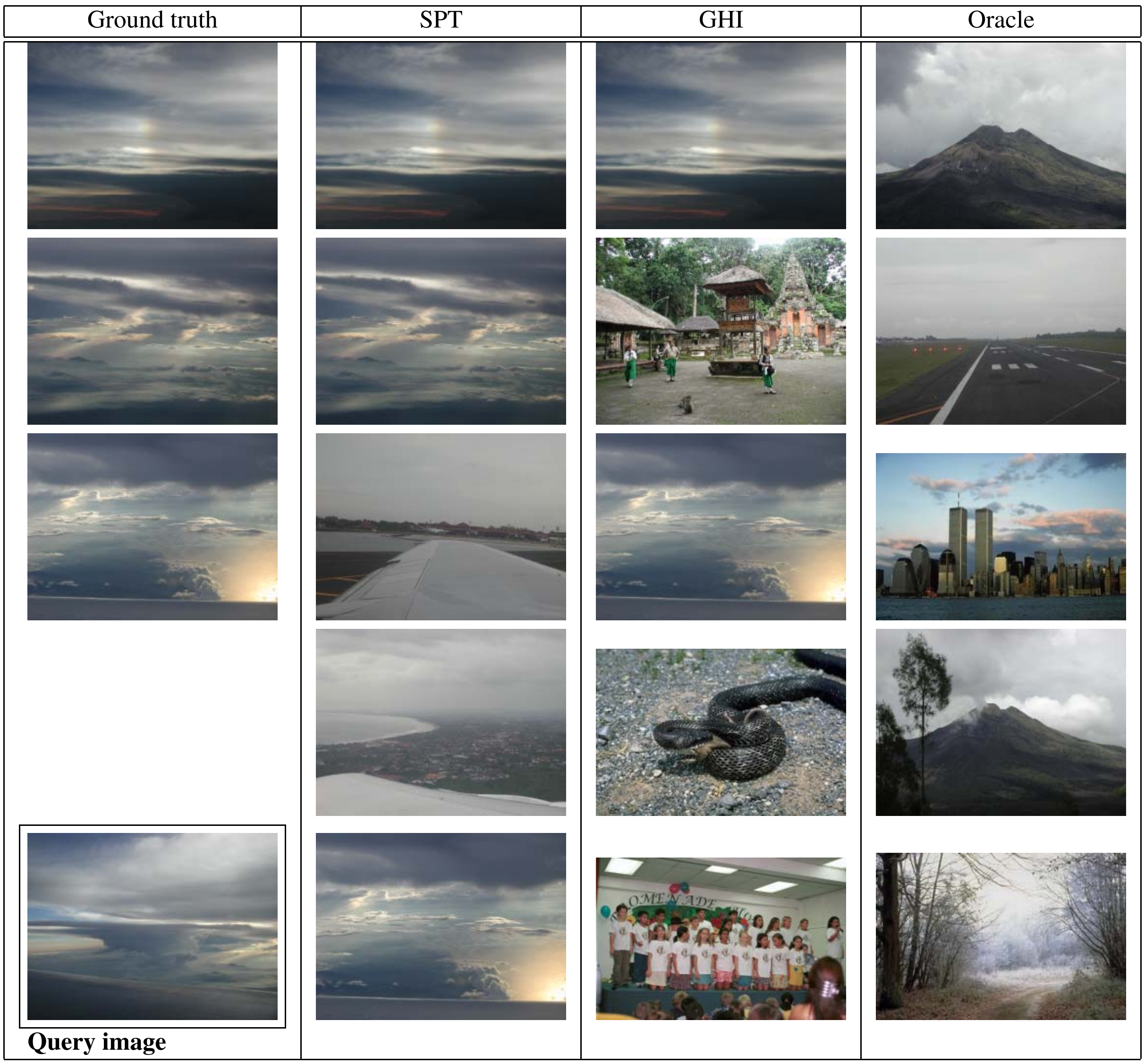

Figure 7. Example 2: retrieval results of the proposed method, histogram intersection, and Oracle's CBIR function.

[21] 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*, 18(7):722–733, July 1996.

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

[23] X. Mao, Y. Nagasaka, and A. Imamiya. Automatic generation of pencil drawing from 2d images using line integral convolution. *Proc. CAD/Graphics'2001*, pages 240–248, 2001.

[24] P. Pala and S. Santini. Image retrieval by shape and texture. *Pattern Recognition*, 32(3):517–527, Mar. 1999.

[25] A. L. Rantan and W. E. L. Grimson. Training templates for scene classification using a few examples. In *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 90–97, 1997.

[26] T. Strothotte and S. Schlechtweg. *Non-Photorealistic Computer Graphics. Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers, 2002.

[27] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[28] T. Sziranyi and Z. Toth. Random paintbrush transformation. In *Proc. 15th International Conference on Pattern Recognition*, volume 3, pages 151–154, Barcelona, Spain, 2000.

[29] A. H. Vermeulen and P. P. Tanner. Observational model of blenders and erasers in computer-generated pencil rendering. *Proc. Graphics Interface'1989*, pages 138–143, 1989.

[30] R. Ward and M. Reinhardt. Oracle visual information retrieval user's guide and reference, release 8.1.7, September 2000. Copyright (c) 1997, 2000, Oracle Corporation. All rights reserved.

[31] F. Werblin, A. Jacobs, and J. Teeters. The computational eye. *IEEE Spectrum*, pages 30–37, 1996.

[32] Y. Zhong and A. K. Jain. Object localization using color, texture and shape. *Pattern Recognition*, 33(4):671–684, Apr. 2000.