Non-Photorealistic Rendering and Content-Based Image Retrieval

Xiaowen Ji
Department of Computer Science
National University of Singapore, Singapore 117543
jixiaowe@comp.nus.edu.sg

Zoltan Kato
Department of Informatics
University of Szeged, Hungary
kato@inf.u-szeged.hu

Zhiyong Huang
Department of Computer Science
National University of Singapore, Singapore 117543
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). We propose a content-based image retrieval method. The novelty is that it is based on the painting representation of images, obtained by Stochastic Paintbrush Transformation (SPT), which automatically simulates a painting process. The painting representation is a stroke sequence. Stroke parameters include color and structure (size, orientation, and position) information. 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 our method has a better performance.

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 ([9]), transformation from a photo ([8, 16, 13, 12]), and interactive drawing ([17, 6]). A more general survey of NPR can be found in [3].

Content-based image retrieval (CBIR) extracts features to index images. The features extracted from images can be semantic or primitive. The most common primitive features are color, texture, and edge/shape. In CBIR, users search images by submitting query image or as alternatives some texture pattern or color from a palette or sketch drawn by users. Although the vast majority of current CBIR techniques are designed for primitive level retrieval, some researchers have attempted to tackle 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 ([15]). The second is focus on object recognition ([5], [7]). However it is hard to obtain semantic features without human assistance.

NPR and CBIR seem to be the two separated areas. The current CBIR work is only limited the use of a hand drawing sketch as one way of the input to retrieve image databases. One major problem of CBIR is that there is a gap between human perception and the commonly used primitive features. However, the human perception in a painting is a key problem in NPR and has been well addressed. It will be promising to explore the use of NPR in CBIR. In this paper, we propose a CBIR method. The novelty is that it is based on the painting representation of image (NPR), obtained by Stochastic Paintbrush Transformation (SPT) method, which automatically simulates painting process. The painting representation is a stroke sequence. Stroke parameters include color and structure (size, orientation, and position) information that is contained in the painting representation. We use the information to measure similarity between images. The intuition is that the similar images have similar painting representations (the stroke sequences) and the representation is closely related to the human perception. 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.

The remaining part of 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 [16] with some improvements. Note that our CBIR method is general and other NPR methods also can be used. 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. Finally, we conclude this paper in Section 5.

2. A Stochastic Paintbrush Transformation Method

A stochastic paintbrush transformation (SPT) is an NPR method that simulates the painting process with sequential strokes at decreasing scales of brush-sizes. While the edge information in the original image is preserved in the painting, fine details are discarded. The rendering process of SPT is a random searching process to generate sequential strokes. A stroke is determined by 5 parameters: shape, size, orientation, position and color, where the shape is constant in the whole painting process; the size is decreased at every painting stage; the orientation and position are randomly generated but accepted according to some criteria; the color is determined by the color distribution in the stroke area of the original image. Acceptance of a stroke depends on the change of the error-summation between the original image and the painting over the stroke area with the introduction of the stroke.

One SPT algorithm was proposed by Sziranyi and Toth in [16]. To adopt it for our purpose, we modified and improved the algorithms in several places:

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, since the painting process of SPT can be seen as a sampling procedure. Kirkpatrick et al. proposed Simulated Annealing, a method using Metropolis Monte Carlo simulation to find the lowest energy (most stable) orientation of a system ([11]). Metropolis Monte Carlo simulation is a sampling procedure that incorporates a temperature of the system ([14]).

The painting process in a stage (i.e. at a brush size) is a Simulated Annealing process and can be separated into iterations, with each iteration a Metropolis Monte Carlo simulation. 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. As Simulated Annealing is to find the lowest energy (most stable) orientation of a system, the introduction of it into SPT makes the final status of SPT with the lowest energy, i.e. the painting with least distortion to the original image.

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 algorithm is listed as follows.

Algorithm 1 (Modified SPT)

① *While not all brushes have been used.*

- ⓘ *Pickup the brush of the next biggest brush-size d.*
- ⓘⓘ *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).*
- ⓘⓘⓘ *While the average error-summation of the distortion-map Δ_n in the last 10 iterations is bigger than a limit.*
  - ⓐ *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.*
  - ⓑ *Compute the histogram of $\mathcal{E}_n$ and set threshold ϵ such that the frequency of higher values equals to a predefined F.*
  - ⓒ *While the number of strokes generated at this status is less than a threshold.*
    - ◯ *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 colors of the original pixels falling inside the new brush stroke $\mathcal{B}(d, s, \phi, C)$.

◯ *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 $\min\{1, (D/D')^{1/T_n}\}$.*

◯ *Update distortion map Δ_n.*

ⓓ $T_{n+1} = 0.8T_n$, $n = n + 1$.

② *Done.*

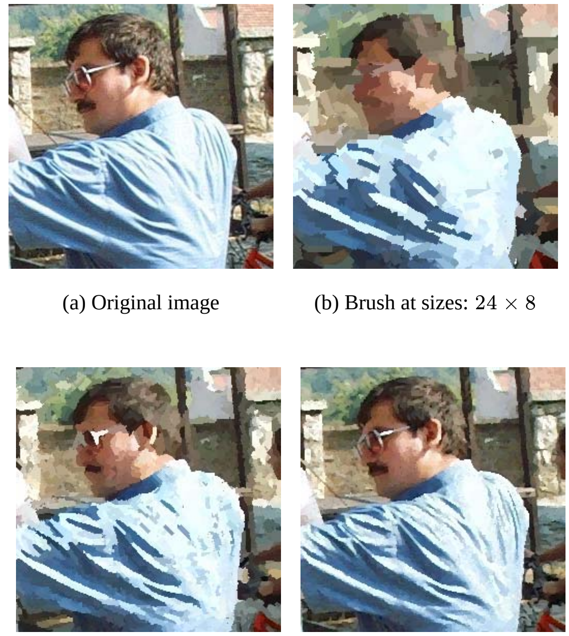
Paintings drawn by our adapted SPT method is shown in Fig. 1 with a rectangle shape brushes. At every stage, SPT produces a stroke sequence and a painting, which are equivalent. In a stroke sequence produced by SPT, there are a lot of strokes wholly covered by succeeding strokes in the sequence. These strokes are redundant, as the information carried by them is replaced by the information carried by succeeding strokes, i.e. they do not contribute to the painting. Therefore, we delete these redundant strokes from the stroke sequences produced by SPT.

A painting obtained by SPT is expressed as a stroke sequence, characterized by its parameters including color, size, orientation and position. Thus we use the stroke features to measure similarity between images. The intuition is that the similar images have similar painting representations (the stroke sequences) and the representation is closely related to the human perception.

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 painting 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: color, orientation, and position. This setting is based on the observation



(a) Original image (b) Brush at sizes: 24×8

(c) Brush at sizes: 12×4 (d) Brush at sizes: 6×2

Figure 1. Sample paintings drawn with SPT Algorithm 1 using rectangle brushes

and experiment study. Thus, among the same size strokes of one painting, we first sort them by their color, then the orientation, and finally the position. The setting my be different for other applications.

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 is 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 [10]. 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

One image database was used in experiments. It contains 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.

The next step is to tune the parameter setting for weights of the global color, local color, texture and shape. We take images randomly from the database. We found that the weight setting: $w_{col} = 0.40$, $w_{ori} = 0.35$ and $w_{pos} = 0.25$, gives the best performance on this database. Thus, we will use them in the following experiments.

To evaluate the performance of the CBIR methods, we manually chose similar images for every image in the database. Although this is subjective, it can determine whether the CBIR performance is close to human perception. In each query, all images in database are ordered on the similarity measurement.

Precision rate and recall rate are mostly used criteria to determine the performance of a CBIR method. We used precision-recall graph to illustrate the performance of CBIR methods.

The proposed CBIR method was compared with GHI and oracle 9i's CBIR function. In implementing GHI, we partitioned the CIE-L*u*v* color space such that the L channel was divided into 6 intervals, the U channel was divided into 14 intervals, and the V channel was divided into 10 intervals. This partition makes a cube with the same length in each channel. The oracle's CBIR function indexes images on four attributes, i.e. global color, local color, texture and shape. The similarity between two images is a weighted summation of similarity scores on these four attributes. Users determine the weights of these four attributes. The comparison result is depicted in Fig. 2. Two retrieval samples are showed in Fig. 3 and Fig. 4. The database together with the ground truth are available on our web site: www.cs.ust.hk/k̃ato/research/spt/cbir/. The experiment shows that the proposed CBIR method has better performance than GHI and oracle 9i's CBIR function.

5. Conclusion

In this paper, we have explored a new role of non-photorealistic rendering (NPR) in content-based image retrieval (CBIR). We have proposed and implemented a CBIR method using the painting representation (the strokes) derived from a stochastic paintbrush transformation (SPT) method. We have conducted the experiment study and the preliminary results are encouraging.

6. Acknowledgment

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

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

[7] 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.

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

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

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

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

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

[13] 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.

[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[15] 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.

[16] T. Sziranyi and Z. Toth. Random paintbrush transformation. In *Proc. 15th International Conference on Pattern Recognition*, volume 3, pages 155–158, Barcelona, Spain, 2000.

[17] 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.

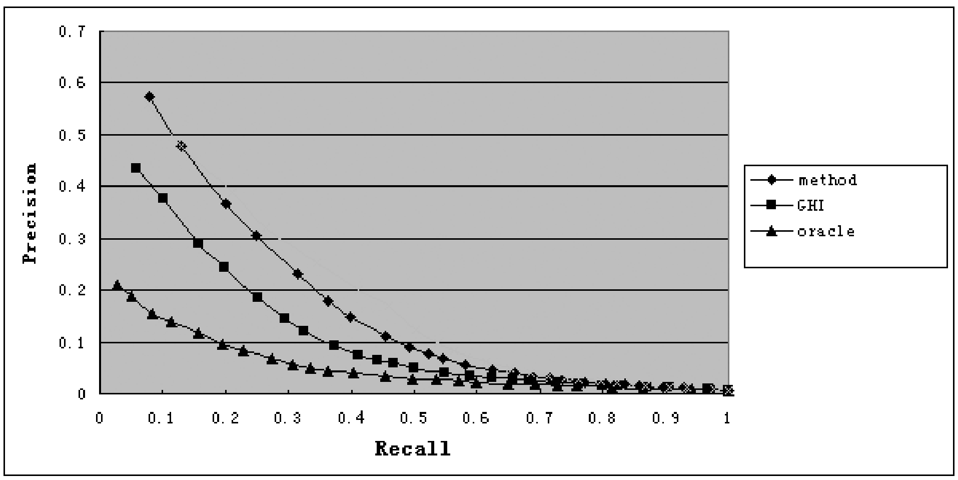


Figure 2. Precision-recall curves of the proposed method, GHI, and Oracle CBIR function

Figure 3. Sample retrieval result 1

Query image

Ground truth (3 images)

First 5 images retrieved by selected method

First 5 images retrieved by GHI

First 5 images retrieved by Oracle

Figure 4. Sample retrieval result 2