Efficient Coding of Stroke-Rendered Paintings

Levente Kovács
University of Veszprém, Hungary
kla@vision.vein.hu

Tamás Szirányi
Hungarian Academy of Sciences
sziranyi@sztaki.hu

Abstract

There are more and more applications of non-photorealistic rendered images, sketches and drawings. Several techniques for generating such imagery are widely known. The stochastic painting-based painterly image (and video) generation presented herein is a multi-purpose image rendering and representation method, suitable for many purposes: painterly rendering, storing, compression or indexing. It incorporates many new features like multiscale edge following, stroke-set optimizations, templates, color morphology, etc. We will demonstrate that the presented technique (called enhanced Stochastic Paintbrush Transformation or eSPT) is suitable for fast high quality painterly rendering, providing good lossless painted compression ratios and features that make it suitable for many applications.One of these we wish to emphasize is the suitability to code painted images in a way that does not introduce any coding artifacts (blockiness, ringings, etc.) but provides a compact form of representation that still retains the main property of a painting: that it is *a painting after all.*

1. Introduction

The goals we wanted to achieve when we started working on the presented stroke-based rendering technique were to achieve painting-like image generation in an automatic way which would mimic human painting styles and produce painting/cartoon-like images. Before we start presenting our Painting technique, let us overview the major contributions in the field of stroke-based rendering.

Haeberli [3] has introduced painting with an ordered collection of strokes, controlling shape, size, color, orientation. Painting is mainly done by following the cursor and randomly sampling the color data. Another possibility to control brush direction is to use arbitrary images and gradient data to control the brush direction across the canvas. In Litwinowitz's work [4] strokes with given center, length, radius and orientation with color bilinearly interpolated are generated, adding random variation and perturbation, gradient-based orientation. In Hertzmann's work [6] painting is done with a series of antialiased cubic B-spline brush-strokes aligned to a grid , with constant color and thickness, with a series of layers, in a coarse-to-fine way, presenting a framework for describing a wide range of styles. The system presented in Gooch et al.'s work [7] lets the designer directly annotate a 3D model with strokes, imparting a personal aesthetic to the non-photorealistic rendering of the object. The artist chooses a brush-style then draws strokes over the model from one or more viewpoints. In [8] a promising extension of "traditional" automatic painterly rendering methods is presented, namely guiding the painting process by eye movement, focusing on areas of the image where the user is looking as being more important as others. This way they can generate an abstraction of the input image which has those areas emphasized which the user thinks it's important. In Kaplan et al.'s work [9] the author presents an algorithm for rendering subdivision surface models of complex scenes using particle systems and graftals introducing geograftals.

Our eSPT (enhanced Stochastic Paintbrush Transformation) painting method:

- is a fully automatic multi-layer coarse-to-fine painterly image transformation technique,
- is a stroke-template based image representation which is invariant of size and scale and suitable for storing, compression, indexing and retrieval,
- uses multiscale gradient-based stroke orientation,
- uses stroke-based painterly image compression giving good scale-independent compression ratios (more than 5 times better painted compression ratios than usual lossless image coders),
- is suitable for cartoon generations,
- uses multi-layer buildup, texture and color morphology.
- output is losslessly compressed with high ratios; compressing painterly rendered images this way provides higher compression ratios than DCT/wavelet coders could do still retaining lossless quality.

2. eSPT

Our Enhanced Stochastic Paintbrush Transformation

0-7695-2128-2/04 $20.00 (C) 2004 IEEE

(eSPT from now on) is an extension of the works in the field presented in [1,2]. The basis of this painting technique consists of a very simple template-based stroke-placing algorithm as shown on Figure 1.

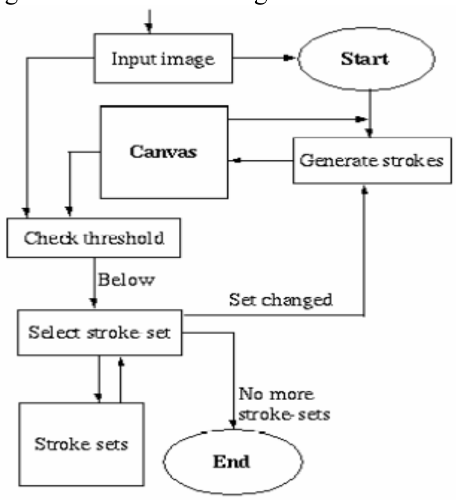

Figure 1. Stroke-placing algorithm.

An ordinary input image is taken as a model and a blank canvas as a starting point. A stroke-type (template) is also chosen with all its stroke-sets (which contain the variants in size of the chosen template). Then chosen strokes are placed at stochastically picked positions using Monte Carlo Markov Chain optimizations at the stroke acceptance step as described in [11]. A set change (changing the actual stroke to a smaller one) occurs when no more reasonable improvement can be made with the present stroke. This is checked at times and if the improvement of the canvas was not up of a threshold, we choose a smaller stroke to continue with. After each set-change there is a post-processing step, which eliminates fully covered strokes. The painting is over, when there are no more smaller stroke-sets.

This base-algorithm was the starting point for the presented eSPT painting technique. Strokes now can be any 60x60 pixel grayscale images, arranged in a stroke-set. When painting, a stroke can be placed at 8 different orientations (at each *pi*/8). Samples of usable templates are on Figure 2.

On Figure 4 there is a sample image painted with default rectangular strokes. A placed stroke is represented by its type, position, orientation and color data, which is very suitable for statistical lossless compression as will be presented in Section 2.5. At the same time, this method of image representation proved to be usable for image indexing and retrieval as presented in [12].

Besides the mentioned properties, our painting method generates painted representations which turn out to be scale independent. The decoder than starts to place the coded stroke series scaling up the respective strokes and aligning them to the right coordinates. This way we get a painterly coded image that can be decoded to almost any magnification. In the following subsections we present further properties and features of our eSPT algorithm.

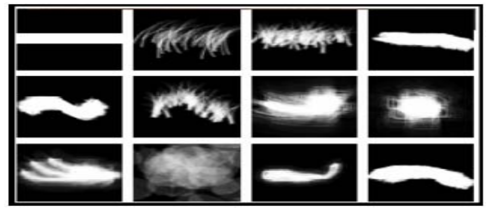
Figure 2. Sample stroke templates.

2.1 Edge following

A new extension to the Stochastic Painting method is the use of a [10]-based multiscale edge detection method, which we use to get orientations of placed strokes. This way we avoid choosing random directions of strokes. We choose this multiscale edge detection variant of Lindeberg's work because it produces more smooth and

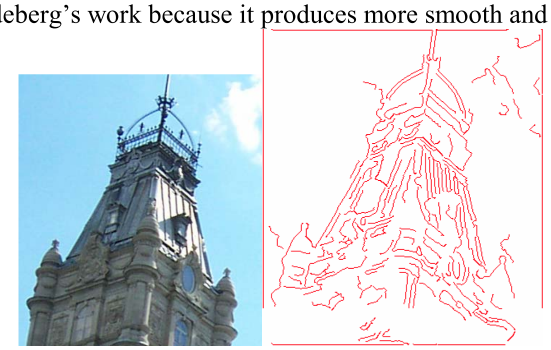
Figure 3. Multiscale edges used for painting.

connected edges than e.g. Canny-based edge detectors [5]. At each location edge orientation is determined from the orientation of the closest laying multiscale edges. For a sample see Figure 3.

2.2 Color morphology

This feature is one that does not cause higher painted quality (nor does it decrease quality), but causes considerable speedup of the transformation process (see Figure 6). We start with a stroke and make the algorithm stop painting the actual layer when it gets covered beyond a given percent (relative to the whole canvas area). When this stage is achieved we use color morphology to cover the remaining areas. This extension also reduces painted image sizes. Example on Figure 4.

2.3 Region of interest

Region of interest selection in our case means that we can select areas which will be covered only at the last step, with the finest stroke-set. This will result in smoother coverage over the selected areas.

Figure 4. Painting using morphology in a middle-step (left:1st,right:2nd,bottom:last step).

2.4 Coding optimizations

We also discovered more optimization possibilities when using Stochastic painting. One of these is that experiments show that using more stroke sets does not always mean more improvement, or faster transformation. An optimal choice is almost always the usage of 3 sets (big, middle, small). See Figure 5. With 3 sets, edge following and morphology the same quality can be achieved in less time (see Figure 6).

2.5 Rendering paintings

A useful area of application for painterly transformation could be representations of real life paintings, as shown on Figure 7. Features like sharp edges, fine textures or high contrast transitions can be easily described by the correct templates, while we would need a reasonably high number of DCT/wavelet coefficients to code such regions, and we would still loose some important image characteristics. By transforming these images with eSPT and losslessly compressing the stroke series we get a representation that can reproduce the painting, with no induced compression artifacts.

3 Compression

If we store the painted images as a losslessly coded stroke-stream, we can achieve as much as 5 times better

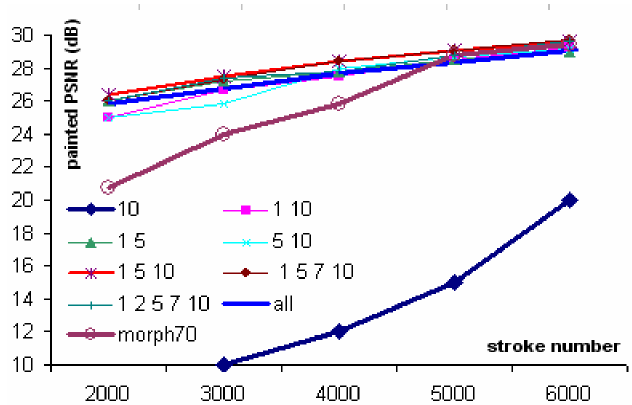

Figure 5. Using around 3 stroke-sets gives the same quality (in about 10x less time).

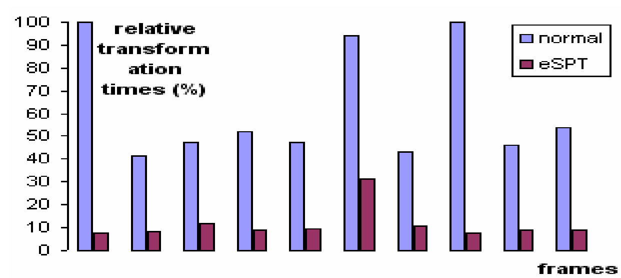

Figure 6. Relative transformation times of frames from the Mom and Daughter qcif sequence ("normal" means stochastic painting without the presented enhancements).

Figure 7. Rendered oil painting (top:original) (Profile of a Woman / Zorka by József Rippl-Rónai, 1916).

lossless compression ratios of painted imagery than DCT and wavelet-based lossless raster image coders can do. See Figure 8. Currently we use Huffman and run-length encoding.

Also, when coding the painted images treated as stroke sequences, it turns out (as further shown on Figure 9. for image sequences) that it is more efficient to code painted images by coding the stroke series that describe the painting. We mention again, that the goal is not to introduce compression artifacts over the painterly rendered images, and to keep their features.

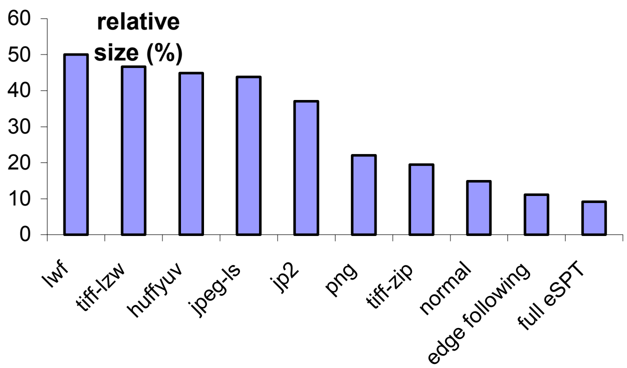

Figure 8. Coding painted image (320x272, 24bit color) with lossless codecs (DCT and wavelet-based) and eSPT. eSPT produces higher lossless compression ratios (("normal" is 10 layer full covering painting, "edge following" is the same but with added gradient following, "full eSPT" is 3 layer coverage to 70% and gradient following).

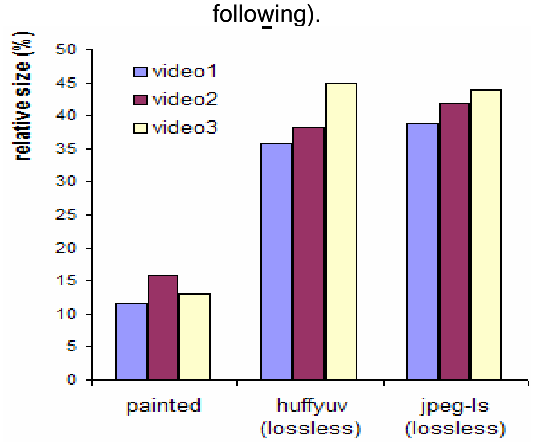

Figure 9. Lossless coding of stroke series is more efficient than lossless pixel-based coding of painted sequences (HuffYUV uses Huffman coding, JPEG-LS uses wavelets).

4. Conclusions

We presented a way of generating non-photorealistic painterly rendered images and representing the painted outputs using stroke-series compression that is suitable for high ratio lossless compression and retrieval and at the same time provides a method for scale-independent image representation. It is important to note that the lossless compression scheme of the resulting stroke series data stream provides an easy, compact and artifact-free way of high ratio coding of painterly rendered imagery. We also explored possibilities of reducing stroke-numbers for achieving similar quality by following main edge curves, by using a kind of morphology for missing area inpainting and by reducing the number of used layers. The method is also suitable for usage on motion picture for generating cartoon-like sequences.

5. References

[1] Szirányi, T., and Tóth, Z. 2000. Random Paintbrush Transformation. In Proc. of 15th ICPR, Barcelona, IAPR&IEEE, V.3, 155-158.

[2] Kovács, L., and Szirányi, T. 2002. Creating Animations Combining Stochastic Paintbrush Transformation and Motion Detection, In Proceeedings of 16th ICPR, IAPR&IEEE, Quebec, Canada, vol. 2, 1090-1093.

[3] Haeberli, P. 1990. Paint by numbers: Abstract image representations. In Proc. of ACM SIGGRAPH 90, 207–214.

[4] Litwinowicz, P. 1997. Processing images and video for an impressionist effect. In Proc. of ACM SIGGRAPH 97, 407–414.

[5] Kovács, L., Szirányi, T.: Painterly Rendering Controlled by Multiscale Image Features, Proceedings of SCCG 2004, 183-190, also Published by ACM.

[6] Hertzmann, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In Proc. of ACM SIGGRAPH 98, 453–460.

[7] Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F., and Finkelstein, A. 2002. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. ACM Transactions on Graphics, 21.

[8] Santella,A., and DeCarlo,D. 2002. Abstracted Painterly Renderings Using Eye-Tracking Data. In Proc. of NPAR 2002, 75-82.

[9] Kaplan, M., Gooch, B., and Cohen, F. 2000. Interactive Artistic Rendering. In Proceeedings of NPAR 2000, 67-74.

[10] Lindeberg, T.: Edge Detection and Ridge Detection with Automatic Scale Selection. International Journal of Computer Vision 2, 30 (1998), 117–154.

[11] Szirányi, T., and Tóth, Z. 2001. Optimization of Paintbrush Rendering of Images by Dynamic MCMC Methods. In Lecture Notes on Computer Science, Springer Verlag, Vol.LNCS 2134, 201-215.

[12] Szirányi,T., Kató,Z., Ji,X., Tóth,Z., and Czúni,L. 2002. Content-Based Image Retrieval Using Stochastic Paintbrush Transformation. In Proc. of ICIP 2002, Multimedia Retrieval and Applications, IEEE, Rochester.