Optimization of Paintbrush Rendering of Images by Dynamic MCMC methods

*Tamás Szirányi [1,3], Zoltán Tóth[2,3]

[1] Analogical Computing Laboratory, Computer. & Automation Research Institue, Hungarian Academy of Sciences, H-1111 Budapest, Kende u. 13-17, Hungary
E-mail: sziranyi@sztaki.hu
[2] Distributed System Department, Computer & Automation Institute, Hungarian Academy of Sciences, E-mail: sac@samson.aszi.sztaki.hu
[3] University of Veszprém, Department of Image Processing and Neurocomputing, H-8200 Veszprém, Egyetem u. 10, Hungary
*This work was partly done during the stay of T. Szirányi at Ariana project, INRIA, 2000

Abstract. We have developed a new stochastic image rendering method for the compression, description and segmentation of images. This paintbrush-like image transformation is based on a random searching to insert brush-strokes into a generated image at decreasing scale of brush-sizes, without predefined models or interaction. We introduced a sequential multiscale 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. The image can be described with the parameters of the consecutive paintbrush strokes, resulting in a parameter-series that can be used for compression. The painting process can be applied for image representation, segmentation and contour detection. Our original method is based on stochastic exhaustive searching which takes a long time of convergence. In this paper we propose a modified algorithm of speed up of about 2x where the faster convergence is supported by a dynamic Metropolis Hastings rule.

1 Introduction

Images can be interpreted in several ways by decomposition into basic functions: strokes [4, 5], fractals [1], etc. Each of these is natural in some sense: strokes are good representations of letters or shapes, Gabor-functions [8] are natural for human sensation, fractals originate from the self-similarity. When we look at some images or image scenes, we usually search for familiar features. This is exploited in fine arts: small details are sometimes neglected while the main features are enhanced.

Nonlinear partial differential equations [6] can be used for enhancing the main image structure. When compressing the image, anisotropic diffusion, based on the scale-

space paradigm, can enhance the basic image features to get visually better quality [10].

Good effects of structured diffusion can be achieved by Gibbs reaction-diffusion method of [14]. A set of about dozen filters is applied to build a potential function for minimization through a stochastic process. Surprisingly good restoring and hiding effects can be demonstrated. This method needs some prior model and it results in partly isotropic morphology.

In our previous paper [11] we introduced a method to follow a painter's process by using simplified artificial strokes. It is not an image-filtering to get a painting-like isotropic transformation. Other image-painting methods of computer graphics deal with model- or edge-controlled methods [e.g. 3,7]. These methods can serve as interesting visual effects (e.g. impressionist style, brush-splines), but they are different from the quality of a 'real-scene' painter (e.g. old-fashion portrait painters or S. Dali). Our first goal is to simulate the painting process to get a picture similar to a 'real-scene' painting where the purpose of the painter is to portray something which looks like to be a real scenery. We deal with a method copying the real visual world into a pleasant form rather than with the artistic interpretation of some special style. Small articles are elaborated with fine brushes, while plain surfaces are painted with greater strokes. On the other hand, the sequential parameters of the consecutive strokes can be applied for image description or some moderate compression as well. We can see that our method enhances the main features, which can easily be followed by eye. The strokes guide our sight.

However, this stochastic stroke-searching and -rendering are exhaustive processes. Any new trial (a new stroke with the color in question) will be accepted if it reduces the distortion. Positioning of a stroke is a random proposal (to avoid ramble-generated "structures"), while the accept/reject decision of the proposed stroke is controlled by the given stage only, namely it is a Markov Chain process.

By selecting an appropriate Monte Carlo Markov Chain (MCMC) process, the exhaustive searching can be replaced with a random decision based on density-approximation of the distribution of the distortion error of strokes. The problem is with the target density: it should be a dynamic definition to avoid the return to the exhaustive search or to fall into the first local minima.

2 The basic concepts of the Paintbrush algorithm [11]

Here we would like to take the human behavior and sensation when looking at and reproducing the visual word into considerations. When looking at an image, there is a very usual way of understanding the visual information:

- Looking at the main outline of the image,
- Looking for the objects,
- Finding small details,
- Relaxed scanning (wandering) on the image.

When interpreting the image content in a visual form, such as representational painting, the re-creation of the image can proceed a similar way:

- Outlining the main areas,
- Elaborating objects,
- Refining the small details.

In both cases (understanding and reproducing) there is a well-defined scale-space line. First the large details, then the finer details are proceeded. In this aspect the representational painting is similar to the anisotropic diffusion that is based on the scale-space theory [6] based on PDEs. However, there is a very important difference between the two scale-space approaches, namely:

Anisotropic diffusion enhances the main edges but smoothes the others, while the present paintbrush transformation has sharp edges at any stage. We get sharp details even in the case of small or low-contrast areas. In the latter sense representational painting gives better segmentation than the anisotropic diffusion. The effect is similar to that of Markov Random Field segmentation [2]. However, MRF has the drawback that the possible number of "colors" is limited to achieve good segmentation in a finite time.

The other important viewpoint is the question of the finest details of an image. When the image contains too many fine details (e.g. sharp photo), our sight may be disturbed by the unnecessary small details. If the image is compressed previously by a function-set of e.g. Cosine or Wavelet functions, then the image results in annoying artifacts in the range of fine details. We can remove the annoying fine information contents at the cost of smoothing the sharp segment-contours. On the contrary, we see a practical effect when we are looking at an image from a distance:

- In case of artwork painting, the visual effect could be perfect, giving high contrast to represent the objects;
- Getting closer to the painting, we can see sharp edges (paintbrush-strokes), but there is a point where there are no more fine details behind it.
- We can relax when looking at the image, since there are no details, which are quivering when scanning through.

When defining what we expect from a new process that follows the main features of representational painting, we can define the main concepts of the new algorithm:

1. It should have sharp edges at any level of image-construction;
2. There are no fine details below a limit;
3. There are sharp edges at the finest level as well;
4. From a given distance the image must give the same visual scenery as the original.

In the following space we can see that the above constraints can be fulfilled by a method, which follows the generation of painting by using different sizes of paintbrush strokes.

3 Our previous algorithm

Main steps of the original algorithm of [11] to generate paintbrush strokes follow:

1) *Starting with the rawest brush-size;*

2) *or Selecting the next, smaller brush-size (δ);*
- If the finest scale is over then, Goto 14;

3) $\boldsymbol{C}_{\varphi\delta}$*: Convolving the* ***I*** *input image by brush-distributions depending on orientation φ and brush-size δ. At δ brush-size we have at least 8 $C_{\varphi\delta}$ maps due to the φ orientations. This map-series is necessary to estimate the brush-color anywhere in the image without further brush-stroke tuning.*

4) ***D****: Difference image between the original* ***I*** *and the present iterated stage* ***X;***

5) ***A****: Absolute or square values of the* ***D*** *difference image to get a distortion-map;*

6) *If the error-summation over* ***A*** *converges too slowly*
- then Goto 2;

7) ***E****: Convolving the* ***A*** *error image by a smoothing due to the diameter of the δ brush-size;*

8) *Calculating histogram of the distortion-map* ***E****, defining a threshold value where the probability of higher errors is ε;*

9) *Randomly choose a x,y position in the image and a φ brush-orientation;*

10) *If the distortion value of* ***E*** *at the pixel position of x,y is in the upper region of the distortion-histogram with probability ε, then*
- give the color of $\mathbf{C}_{\varphi\delta}$(x,y) to the brush-stroke centered at position (x,y) with the actual φ orientation and δ brush-size;
- or, in case of large strokes, give the color of the majority vote in the brush-area of the original **I** image to the brush-stroke centered at position (x,y);

11) *Cover the painted image by the pattern of this brush-stroke;*

12) *If the error-summation between the original and present stage image over the area of this stroke is decreasing,*
- then accept the new stroke;
- else reject and restore the previous stage in the stroke-area;

13) If the counter of the brush-strokes is smaller than a limit-number,
- then Goto 9 ;
- else Goto 4 ;

14) Ready.

After completing the rendering process, redundant strokes (fully covered by consecutive strokes) are eliminated from the code of the stroke-series. It usually results in a 20-50% decrease of the number of mounted strokes. Presently, the number of the possible φ stroke-orientations is 8.

We can see that our algorithm is stochastic, error-controlled and multiscale. In our present experiments brush-strokes are simple rectangles (see Figure 1). Since strokes need relatively great convolutions, the $C_{\varphi\delta}$ maps are generated in the Fourier domain.

If the process is not random enough (forcing any prior structural constraints when placing a stroke, as considering and modeling ‘edgy’ places), the result may suffer from structural side-effects, like strong and disturbing contours.

The method has been tested for several parameter sets and test images. Some of the resulted images can be found in Figure 2. The “painted” images seemed to have high quality at the last phase with the finest brush. However, at every stage the edges are sharp and the patterns and textures are appropriate. It is interesting to note that the sharp edges and patterns can be generated without any a priori definitions of contours or textured places.

Fig. 1. Painting strokes with border (left) and without border (right) to demonstrate the non-structured random searching of the process (detail of test image “Barbara”)

This method demonstrates that a fully random searching process generating brush-strokes can result in a high quality and pleasant-to-see image. This method takes about 10-30 minutes on a Pentium III PC when generating a 512x512 image. However, it can be easily implemented in parallel processor-arrays at high speed, like MRF-based algorithms in [12]. Since the above method is a ‘brute-force’ algorithm with an exhaustive stochastic searching, any development in the convergence speed and compactness of the generated code of the stroke-series may help in the applications.

Original image Intermediate, coarser brush-size Finished image with small brush-size

Fig. 2. Painting stages at different brush-sizes of image “Lena” , “Leopard”, and “Goldhill” (Number of the scales of brush-strokes was up to 10)

4. Generating Series of Strokes constituting a Markov Chain with respect to the Metropolis-Hastings [9] rule

When generating strokes to cover a patch on the image, the process is random: there is a target-density to constrain for the bounding errors and the proposal density depends on the characteristics of the painting process. First, we overview some Metropolis Hastings algorithms that we may use in the sequel.

The paintbrush rendering as originally described in [11] is a brute-force algorithm when considering its goal-function, error-bound and convergence. Applying MH methods, we hope to reach better-defined algorithms, with explicable effects and distributions and well-proven convergence properties.

The goal of the painting process is to get a stroke-patched image which differs from the original one by less than a minimum mean-error, the difference being smaller than some minimum for any stroke-position. It can be best described by using an appropriate target density for the similarity-error over the individual strokes.

The individual event $X^{(t)}$ generates a paintbrush stroke at a position (coordinate of its reference point and the tilting angle) $S^{(t)}$ and with color $C^{(t)}$. The distortion error between $X^{(t)}$ and the original image in the area of $X^{(t)}$ is $\mathrm{E}^{(t)} = E\left(X^{(t)}\right)$. The position $S^{(t)}$ is randomly generated and may or may not depend on the previous stage and the overall image-parameters, while color $C^{(t)}$ is defined by the mean or majority color behind the proposed stroke in the original (reference) image. The target distribution f is the goal-function of $E\left(X^{(t)}\right)$: $f = f\left(E^{(t)}\right)$.

With the above notation, we can generate the series of $X^{(t)}$ s using the following algorithms:

PB-MH Algorithm:

Given $x^{(t)}$, the stroke of the *t*-th iteration,

1. Generate $Y_t \sim q\left(y \mid x^{(t)}\right)$
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \rho\left(x^{(t)}, Y_t\right) \\ x^{(t)} & \text{with probability} \quad 1 - \rho\left(x^{(t)}, Y_t\right) \end{cases} \tag{1}$$

Where $\rho(x, y) = min\left\{\frac{f(y)}{f(x)}\frac{q(x \mid y)}{q(y \mid x)}, 1\right\}$

The algorithm starts with the error-bound of the proposed stroke-approximations as target density f. The conditional density $q\left(y \mid x\right)$ is a random position-generator of S^{Y_t}, related to the overall controlling error/edge map and the previous $X^{(t)}$ stroke.

Independent PB-MH

Given $x^{(t)}$

1. Generate $Y_t \sim g(y)$ (generated from the error-map and/or edge-map of the input image)
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \min\left\{\dfrac{f(Y_t)g(x^{(t)})}{f(x^{(t)})g(Y_t)},\ 1\right\} \\ x^{(t)} & \text{otherwise} \end{cases} \tag{2}$$

Here the proposal density $g(y)$ is a random position-generator of $S^{(t+1)}$, independently of the previous $S^{(t)}$ position. This $g(y)$ might be uniform on the whole image.

Random Walk PB-MH

Given $x^{(t)}$

1. Generate $Y_t \sim g(y - x^{(t)})$
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \min\left\{\dfrac{f(Y_t)}{f(x^{(t)})},\ 1\right\} \\ x^{(t)} & \text{otherwise} \end{cases} \tag{3}$$

Here the proposal density $g(y - x^{(t)})$ is a random position-generator of $S^{(t+1)}$, related to the previous $S^{(t)}$ position.

The original exhaustive searching algorithm in [11] operates with the 'Random walk' or 'Independent' proposal positioning, while in the accept/reject method f is a simple threshold on E , ($f(X) = f(\mathrm{E}(X))$), where $f(x^{(t)})$ and $f(Y_t)$ are considered on the proposed position, and $X^{(t)}$ and $Y^{(t)}$ indicate the whole image at different painted stages. When trying probabilistic accept/reject strategies in the same framework, we cannot get better convergence. As examples, we tested proposals as

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \min\left\{\dfrac{E(x^{(t)})}{E(Y_t)},\ 1\right\} \\ x^{(t)} & \text{otherwise} \end{cases} \quad \text{or}$$

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \max\left\{\min\left\{E(x^{(t)}) - E(Y_t)\ ,\ 1\right\},\ 0\right\} \\ x^{(t)} & \text{otherwise} \end{cases},$$

but the resulted quality and/or the convergence speed were much poorer than the simple exhaustive searching. We stepped toward a fully stochastic, MCMC-based algorithm.

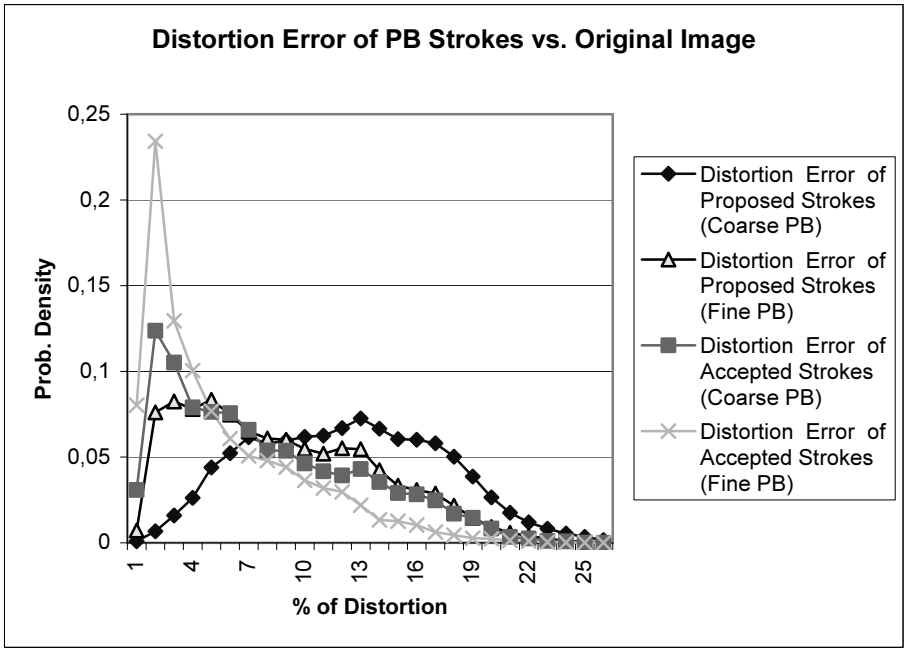

Fig. 3. Probability densities of distortion error of the proposed (Y_t) and accepted ($X^{t+1} = Y_t$) strokes versus the original image when generating the strokes for 'Barbara' image. First, coarse (20x5), finally, fine (7x2) strokes are generated.

5 Considering Statistical Tendencies of Densities of Distortion

The above algorithm [11] generates nice painting-like images, but the process needs optimization to get better speed. We can simply apply some quality constraints for $E(x^{(t)}) - E(Y_t) \geq \varepsilon > 0$ to force the acceptance of greater changes, but ε could depend on the image and the current situation, causing a loss of most of the proposals as in Fig 5(left). Practically, choosing ε=0.01 may improve the compression ratio or convergence speed for several images. However, we should find some more self-calibrated manner for acceptance.

In the following space, $X^{(t)}$ (previous stage) and Y_t (proposed stage) denote state images with events of new strokes at position S^t and $S^{(t+1)}$ on the state image (painted by previous strokes as well).

We define distortion errors against the original image I:

- $E(x^{(t)}) = D\left(X^{(t)}, I\right)$ is the distortion error between the original and the previously painted stage image,
- $E(Y_t) = D\left(Y_t, I\right)$ is the distortion error between the original and the proposed painted stage Y_t,
- $E(x^{(t-1)}) = D\left(X^{(t-1)}, I\right)$ is the distortion error between the original and the last before painted stages.

$D(.,.)$ is defined as the mean square error, normalized by the actual area of the stroke into the [0.0, 1.0] interval.

Figure 3 shows the plot of statistics of the proposed and accepted $D\left(Y_t, I\right)$ values. First coarser, later finer strokes are generated, in the meantime the average error rate is decreasing. We can see that the distribution $f\left(D\left(Y_t, I\right)\right)$ converges to a Dirac-delta at zero for the accepted ($X^{t+1} = Y_t$) proposals. If we use any of the density functions of Figure 3 for a Metropolis-Hastings approach of eq. (1), the process does not work:

- If we use a long-term statistics of $f\left(Y_t\right) = f\left(D\left(Y_t, I\right)\right)_{\text{proposed}}$, we get the same or much slower convergence than the exhaustive searching, since it does not fit to the acceptation rule;
- If we use a short-term statistics or the final $f\left(Y_t\right) = f\left(D\left(Y_t, I\right)\right)_{\text{accepted}}$ (cc. Dirac-delta), the process stops at the beginning or stops at a low quality.

Applying the above statistics anyhow, we have the problem that Metropolis-Hastings MCMC converges relatively slowly, while $f\left(D\left(Y_t, I\right)\right)$ and $f\left(D\left(x^{(t)}, I\right)\right)$ will change considerably. So, the convergence and the basic idea of the process fail during the iterations. The other problem is that we must accept only proposals which decreases the overall error, otherwise the image will be unstructured, as in Fig 5(right).

In our new development of process, the following constraints should be kept:

- Accept error reduction only;
- The previous event and the present proposal of different proposed stroke-position are compared instead of distortion errors in the same place of the proposed stroke;

- The MCMC density-approximation is dynamic, following the changing target density.

Testing several experiments and on the above constraints it is clear that not the $D(.\ ,\ .)$ terms, but their difference is important in the optimization process. Now we define two error differences:

$$Diff(Y_t) = E(x^{(t)}) - E(Y_t)$$
$$Diff(x^{(t)}) = E(x^{(t-1)}) - E(x^{(t)})$$

Figure 4 shows the statistics of the proposed (different cases are nearly the same) and the accepted (two cases) $Diff(Y_t)$ events, as the function of the relative error-rate.

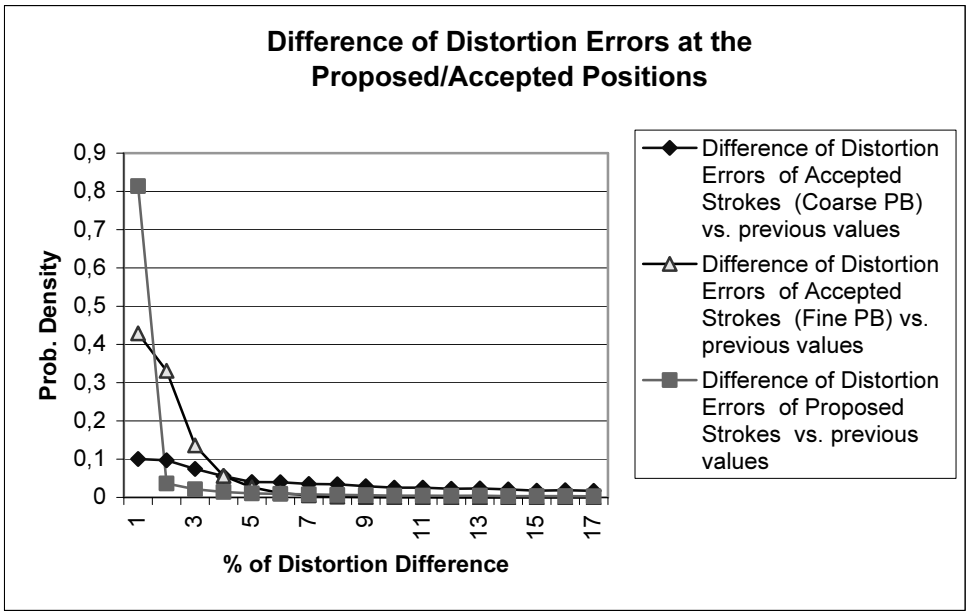

Fig. 4. Probability densities of difference btw distortion errors of the proposed and accepted strokes and the distortion on previous area of the stroke ($Diff(Y_t)$), when generating the strokes for 'Barbara' image. First, coarse (20x5), finally, fine (7x2) strokes are generated.

Fig. 5 *Erroneous results of rendering when*
(left) Only strokes of high quality increase are accepted, $Diff(Y_t) \geq 0.1$;
(right) The acceptance of a stroke may increase the overall error-rate. It is nearly the same as a process of fully *random* positioning, proposal and *acceptance.*

It is clearly demonstrated that density $f\left(Diff(Y_t)_{accepted}\right)$ is changing considerably through the iterations, while probability of the higher values of the proposed $Diff(Y_t)$ should decrease in time to get better convergence. It means that the stochastic rendering should consider the higher distortion values with a higher probability. Since density $f\left(Diff(Y_t)_{proposed}\right)$ is nearly a Dirac-delta, containing mostly low $Diff(Y_t)$ values, it can be supposed that the proposed dynamic target density is

- $f(Y_t) = 0$ at $Diff(Y_t) \leq 0$, so it is rejected;
- $f(Y_t)$ is progressive for higher $Diff(Y_t)$ values;
- $f(Y_t)$ should be normalized by the variance of the current distribution of $Diff(Y_t)$ values to follow the narrowing of the distribution.

On the above considerations, we suppose the next accept/reject rule:

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability} \quad \rho\left(x^{(t)}, Y_t\right) \\ x^{(t)} & \text{with probability} \quad 1 - \rho\left(x^{(t)}, Y_t\right) \end{cases}$$

$$\text{Where} \quad \rho(x, y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x \mid y)}{q(y \mid x)},\ 1\right\} \tag{4}$$

$$f(y) \sim \begin{cases} 0, & \text{if} \quad Diff(Y_t) < 0 \\ \left(Diff(Y_t) / Var_t \right)^q, & \text{else} \end{cases}$$

$$f(x) \sim \left(Diff(x^{(t)}) / Var_t \right)^q$$

Here Var_t is the variance of $Diff(Y_t)$ in a neighborhood time interval around t. Since Var_t is changing slowly by time, it can be eliminated (and neglected) from $\rho(x, y)$ in eq. 4. Although $f(.)$ is the target density, it cannot be measured as a result of long-term statistics, since Var_t causes dynamic windowing effect on the output histogram, as Figure 6 demonstrates it.

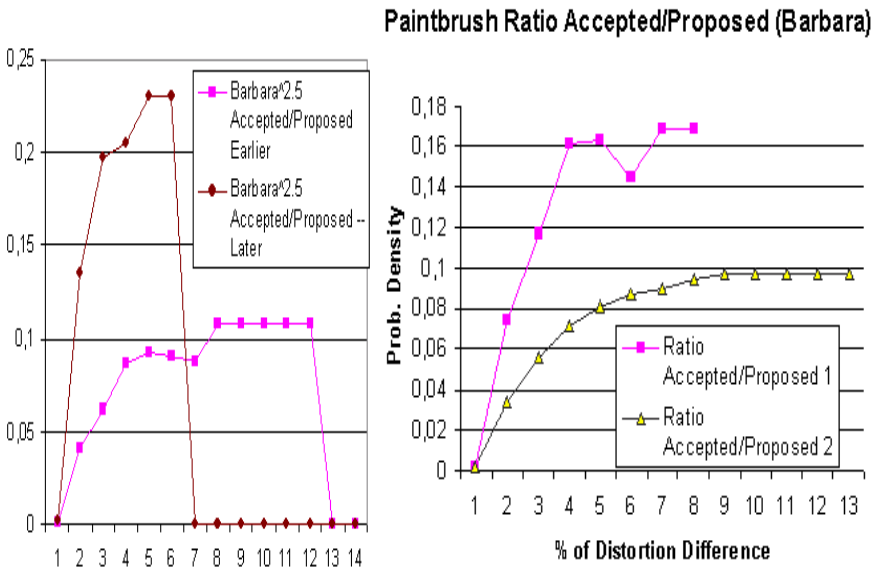

Fig. 6 *Narrowing effect of the distribution of* $Diff(Y_t)$ *and the measured statistics of the conditional probability*

(left) Plot of ratio of measured histograms of the accepted and proposed distributions $P(Diff(Y_t)_{accepted}) / P(Diff(Y_t)_{proposed})$ in the same size of stroke-series, showing the change in the conditional variance;

(right) The above ratio of histograms for two different brush-sizes in whole-runs;
For less optimized cases the monotonic function becomes a simple thresholding.

Applying eq. 4, we tested it with different exponents q to reach the same image quality in PSNR at the same constraints (starting and stopping conditions,

brush-sizes etc.). The results for image Barbara (using 3 different sizes of strokes) can be found in Table 1. We have tested many other images and parameter settings, with very similar conclusions. Tables 2 and 3 demonstrate some results for two 512x512 color images. Tables 4 shows a test of 'Lena' by using 10 different sizes of strokes. Exhaustive search of ε=0.01 also may give good compression result, but at the cost of twice effort, while MCMC at *q=2.5* shows a good compromise.

We can see that eq. 4 gives an effective algorithm for image rendering at *q=2 … 3.* Convergence speed is much better because of the smaller number of proposing and drawing events, at a considerable gain in the resulted number of paintbrush strokes.

Table (1) Painting results for 'Barbara 512x512' at a standard $D(Y_t, I)$=25.5dB PSNR final distortion error rate. Counts are in thousand events.

Method	Number of non-redundant PB strokes # thousand	Number of all drawn strokes $\#((Y_t)_{accepted})$	Number of all proposed strokes $\#\left((Y_t)_{proposed}\right)$
$Diff(Y_t) > 0$ (exhaustive search)	35	61	2645
q=1	29	40	2380
q=2	26	31	1555
q=3	28	34	1200

Table (2) Painting results for a 512x512 color image (detail of http://www.sztaki.hu/~sziranyi/Sz-Barbara.jpg) with room, flowers and a girl at a standard $D(Y_t, I)$=26.5dB PSNR final distortion error rate. Counts are in thousand events.

Method	Number of non-redundant PB strokes # thousand	Number of all drawn strokes $\#((Y_t)_{accepted})$	Number of all proposed strokes $\#\left((Y_t)_{proposed}\right)$
$Diff(Y_t) > 0$ (exhaustive search)	32	52	6490
q=1	26	34	2975
q=2	24	27	1630
q=3	19	22	3445

We have also tested the role of the proposal density $q(y \mid x)$. Usually, it is symmetrical, or it is very close to be symmetrical. Step 10 in the algorithm of Section 3

may be asymmetrical, but its outcome is not too important. So, usually, we can consider $q(y \mid x) = q(x \mid y)$.

The proposal method can also be important in some cases. Applying 'Random walk' with small neighborhood instead of 'Independent' method with any sizes of steps, textured images can better converge.

In a recent research we develop the above method to consider MRF-conform neighborhood calculus into the energy term to run a structured MRF-like painting algorithm in a MCMC estimation series.

Table (3) Painting results for a 512x512 color image with a boy and color books, at a standard $D(Y_t, I)$=26.6dB PSNR final distortion error rate. Counts are in thousand events.

Method	Number of non-redundant PB strokes # thousand	Number of all drawn strokes $\#((Y_t)_{accepted})$	Number of all proposed strokes $\#\left((Y_t)_{proposed}\right)$
$Diff(Y_t) > 0$ (exhaustive search)	9	16	760
q=1	7.4	12	860
q=2	7	9.6	560
q=3	7.4	8.7	495

Table (4) Painting results for 'Lena 512x512' at a standard $D(Y_t, I)$=30.45dB PSNR final distortion error rate. Counts are in thousand events.

Method	Number of non-redundant PB strokes # thousand	Number of all drawn strokes $\#((Y_t)_{accepted})$	Number of all proposed strokes $\#\left((Y_t)_{proposed}\right)$
$Diff(Y_t) > 0$ (exhaustive search)	28	41	1369
$Diff(Y_t) > 0.01$ (exhaustive search)	11	11	2463
q=1.5	17	20	2441
q=2.5	14	14	1504

Conclusions

We have developed a dynamic approximation of target density for image rendering. A Metropolis-Hastings rule is proposed where the target density dynamically fits the process when the characterizing densities are changing through the iterations. This method results in better convergence (lower number of proposed and accepted

strokes) and better compression (lower number of non-redundant strokes). The above MCMC-based stochastic search algorithm is quite different from other searching & matching methods, like matching pursuit [8] or effect-oriented paintings [7].

Acknowledgements

Many thanks are due to Josiane Zerubia. This work was supported by the Hungarian Research Fund (OTKA T032307) and by the French Ministry of Education and Research.

References

1. Y. Fisher. (ed.), *Fractal Image Compression*, Springer Verlag (1994)
2. S. Geman, D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Tr. PAMI*, Vol. 6, pp. 721-741 (1984)
3. P.E. Haeberli, "Paint by numbers: Abstract image representations", *Computer Graphics*, (Proc. SIGGRAPH' 1990), V.24, pp.207-214, (1990)
4. H. H.S. Ip, H. T. F. Wong, "Generation of Brush Written Characters with fractal Characteristics from True-type Fonts", ICCPOL, *17th Int. Conf. on Computer Processing of Oriental Languages*, pp. 156-161, Hong Kong (1997)
5. H. H S Ip, H. T F Wong, "Calligraphic Character Synthesis using Brush Model", CGI'97, *Computer Graphics International conference*, pp. 13-21, Hasselt-Diepenbeek, Belgium, June 23-27 (1997)
6. T. Lindeberg, B. M. Haar Romeny, "Linear scale-space I-II", Geometry - Driven Diff. In Computer Vision, *Kluwer Academic Publishers*, pp.1-72 (1992)
7. P. Litwinowicz, 'Processing Images and Video for An Impressionist Effect', *Computer Graphics*, (Proc.SIGGRAPH'1997), pp.407-414 (1997)
8. R. Neff, A. Zakhor, "Very low bit-rate video coding based on matching pursuits", *IEEE Tr. CAS VT*, Vol. 7, pp.158-171 (1997)
9. Ch. P. Robert, G. Casella, "*Monte Carlo Statistical Methods*", *Springer* Text in Statistics (1999)
10. T. Szirányi, I. Kopilovic, B. P. Tóth, "Anisotropic Diffusion as a Preprocessing Step for Efficient Image Compression", *Proc. of the 14th ICPR*, Brisbane, *IAPR&IEEE*, Australia, pp. 1565-1567, August 16-20 (1998)
11. T. Szirányi, Z. Toth, "Random Paintbrush Transformation", *15th ICPR*, Barcelona, IAPR&IEEE, V.3, pp.155-158 (2000)
12. T. Szirányi, J. Zerubia, "Markov Random Field Image Segmentation using Cellular Neural Network", *IEEE CAS I.*, Vol. 44, pp. 86-89 (1997)
13. P. Teo, D. Heeger, "Perceptual Image Distortion", First *IEEE Int.Conf. Image Proc*., Vol.2, pp.982-986 (1994)
14. S.C. Zhu, D. Mumford, "Prior Learning and Gibbs Reaction-Diffusion",*IEEE Tr. PAMI*, Vol. 19, pp.1236-1250 (1997)