Texture vs contours

Pápari, Giuseppe

Chapter 7

Contonous Glass Patterns for Painterly Rendering

Based on the content of the paper:
G. Papari and N. Petkov, "Contonous Glass Patterns for Painterly Rendering", *IEEE Transactions on Image Processing* **18**(3) : 652-664, 2009

Abstract

Glass patterns have been exhaustively studied both in the vision literature and from a purely mathematical point of view. We extend the related formalism to the continuous case and we show that continuous Glass patterns can be used for artistic imaging applications. The general idea is to replace natural texture present in an input image with synthetic painterly texture that is generated by means of a continuous Glass pattern, whose geometrical structure is controlled by the gradient orientation of the input image. The behavior of the proposed algorithm is analytically interpreted in terms of the theory of dynamical systems. Experimental results on a broad range of input images validate the effectiveness of the proposed method in terms of lack of undesired artifacts, which are present with other existing methods, and easy interpretability of the input parameters.

7.1 Introduction

Glass patterns (GP) [1–3] have drawn considerable attention in the vision literature. Fig. 7.1 shows examples of such patterns, obtained by superposing a random point set with a rotated version of it. If the rotation angle is sufficiently small, a circular structure is clearly visible [1]. As the rotation angle increases, the circular structure becomes less salient and finally disappears. Similar effects can be achieved if other geometric transformations are considered instead of rotation (Fig. 7.2).

A large amount of research has been carried out in order to understand the perception of geometrical structure in GP. Direct measurements of the neural activity in the areas V1 and V2 of the brain of primates indicate that, when GP are presented, neurons of the visual cortex strongly respond to the local orientation of pairs of dots [4, 5]. The responses to such dipole patterns are then processed by means of association fields, thus extracting long chains

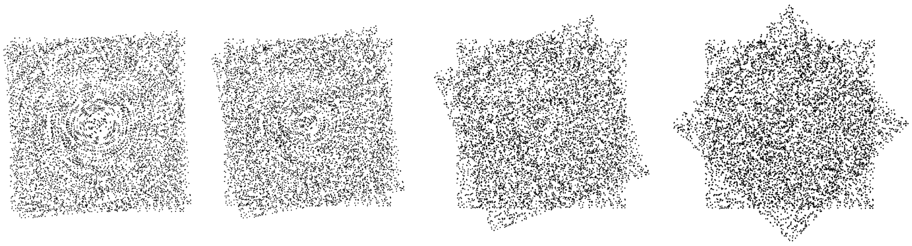


Figure 7.1: *Glass patterns obtained by superposing a random point set on a rotated copy of it, with rotation angles of 5, 10, 20, and 45 degrees.*

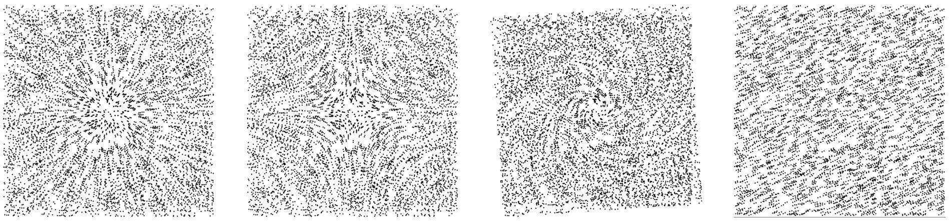


Figure 7.2: *Glass patterns obtained by several geometric transformations. From left to right: isotropic scaling, expansion and compression in the horizontal and vertical directions, respectively, combination of rotation and isotropic scaling, and translation. Note that translational GP are the least salient.*

of collinear segments [6, 7]; these chains determine the geometrical structure perceived in GP. Circular and spiral structures in GP are more salient and more robust to noise than radial, hyperbolic and translational geometries [8–10]. This indicates that the contour integration process that is performed by the visual system is more sensitive to closed geometric structures [11] and this might be the basis of the gestalt principles of closure and praegnanz [11, 12]. Several computational models of the perception of GP, taking into account both local and global mechanisms, have been proposed [11, 13–15].

GP have also been studied from a purely mathematical point of view in terms of their macro and microstructure. The former concerns local dot density, which turns out to be lower in the centre of the pattern. The related phenomenology has been exhaustively studied [16, 17] and it can be explained by the general theory of moiré effects on both periodic and non-periodic patterns [18, 19]. On the other hand, microstructure concerns the orientation field that is induced by the strong correlation between the two superposed point sets that compose a GP. Microstructural properties of GP can be studied naturally in the framework of the theory of dynamical systems [20, 21] and algorithms able to synthesize any microstructure have been provided [21].

Despite their theoretical importance, and the large amount of research conducted on them, GP have not been used in image processing yet. This is probably due to the fact that continuous structures are more suitable for image processing tasks than point sets.

In this chapter, we introduce a continuous version of GP and we show that it can be used to produce a nice artistic effect in photographic images. The idea is to replace the natural texture of the input image with a synthetic painterly texture generated by means of a continuous Glass pattern. The rest of this chapter is organized as follows: In Section 7.2, the mathematical formalism related to GP is reviewed and extended to continuous signals. In Section 7.3, the proposed painterly rendering algorithm is described. Results and comparison with other techniques are presented in Section 7.4 and conclusions are drawn in Section 7.5.

7.2 Discrete and Continuous Glass Patterns

In this section we review the mathematical formalism related to the classical discrete GP (Section 7.2.1) and we extend it to the continuous case (Section 7.2.2). We will use the notation $g_\sigma(\mathbf{r})$, $\mathbf{r} = (x, y)$ for a 2D Gaussian function with standard deviation σ:

$$g_\sigma(\mathbf{r}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\mathbf{r}|^2}{2\sigma^2}\right) \tag{7.1}$$

7.2.1 Discrete Glass patterns

Let $\mathbf{v}(\mathbf{r})$ be a vector field defined on $\mathbb{R}^2$ and let us consider the following differential equation:

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v}(\mathbf{r}) \tag{7.2}$$

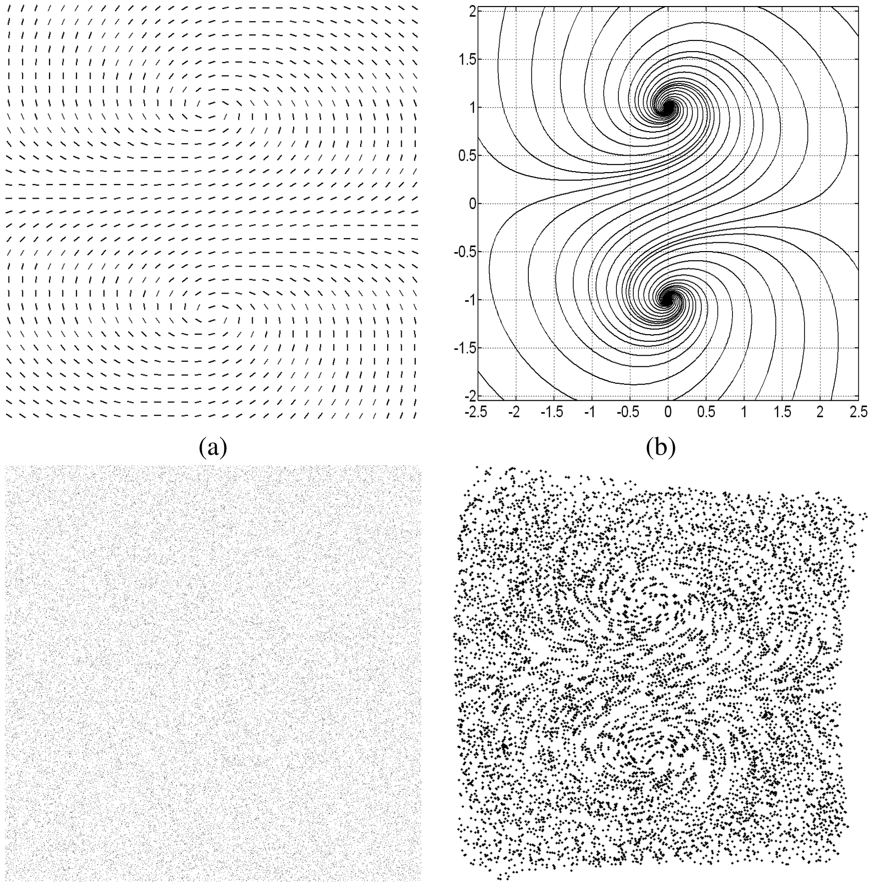
We indicate the solution of (7.2), with the initial condition $\mathbf{r}(0) = \mathbf{r}_0$, by $\mathbf{r}(t) = \mathbf{\Phi}_\mathbf{v}(\mathbf{r}_0, t)$. For a fixed value of t, $\mathbf{\Phi}_\mathbf{v}$ is a map from $\mathbb{R}^2$ to $\mathbb{R}^2$, which satisfies the condition $\mathbf{\Phi}_\mathbf{v}(\mathbf{r}, 0) = \mathbf{r}$.

Let $S = \{\mathbf{r}_1, ..., \mathbf{r}_N\}$ be a random point set and let be $\mathbf{\Phi}_\mathbf{v}(S, t) \triangleq \{\mathbf{\Phi}_\mathbf{v}(\mathbf{r}, t) | \mathbf{r} \in S\}$. Using this notation, we define the Glass pattern $G_{\mathbf{v},t}(S)$ associated with S, $\mathbf{v}$, and t as follows:

$$G_{\mathbf{v},t}(S) \triangleq S \bigcup \mathbf{\Phi}_\mathbf{v}(S, t) \tag{7.3}$$

Examples of GP generated by linear differential equations are shown in Figs. 7.1 and 7.2, where the elements of $G_{\mathbf{v},t}(S)$ are rendered as black dots on a white background[1]. In Fig. 7.1, Glass patterns related to the vector field $\mathbf{v}(x, y) = (-y, x)$ are shown for different values of the parameter t ($t = \frac{\pi}{36}, \frac{\pi}{18}, \frac{\pi}{9}, \frac{\pi}{4}$). More general vector fields give rise to more sophisticated geometries (Fig. 7.3). In general, the geometrical structure exhibited by the GP defined in (7.3) is related to the trajectories which solve (7.2) [21].

[1] Other renderings and superposition rules may result in different microstructures, as shown, for instance in [19, 22].



(a) (b)

(c) (d)

Figure 7.3: *Glass pattern associated to a nonlinear vector field. (a) vector field* $\mathbf{v}(x,y) = [y^2 - 1 + \frac{1}{3}xy, \frac{1}{3}(y^2-1) - xy]^\intercal$*, (b) the trajectories which solve the corresponding differential equation* $\dot{\mathbf{r}} = \mathbf{v}(\mathbf{r})$*, (c) the random point set* S*, and (d) the corresponding GP.*

7.2.2 Continuous Glass patterns

In order to generalize this formalism to the continuous case, we define a binary field $b_S(\mathbf{r})$ associated with a point set S as follows:

$$b_S(\mathbf{r}) \triangleq \begin{cases} 1, & \mathbf{r} \in S \\ 0, & \mathbf{r} \notin S \end{cases} \tag{7.4}$$

It is straightforward to see that the binary field associated with the superposition of two point sets S_1 and S_2 is equal to:

$$b_{S_1 \bigcup S_2}(\mathbf{r}) = \max[b_{S_1}(\mathbf{r}), b_{S_2}(\mathbf{r})] \tag{7.5}$$

Therefore, from (7.3) and (7.5), we see that the binary field associated with a GP is equal to:

$$b_{G_{\mathbf{v},t}(S)}(\mathbf{r}) = \max\{b_S(\mathbf{r}), b_S[\mathbf{\Phi_v}(\mathbf{r}, t)]\} \tag{7.6}$$

The generalization of (7.6) to the continuous case is straightforward: first, a continuous set of patterns $b_S[\mathbf{\Phi}(\mathbf{r}, \tau)]$, with $\tau \in [0, 1]$ is considered, instead of only two as in (7.6); second, any real valued random image $z(\mathbf{r})$ can be used instead of a Poisson process $b_S(\mathbf{r})$. Specifically, a *continuous Glass pattern* (CGP) $\mathfrak{G}_\mathbf{v}(\mathbf{r})$ is defined as follows:

$$\mathfrak{G}_\mathbf{v}(\mathbf{r}) \triangleq \max_{\tau \in [0,1]} \{z[\mathbf{\Phi_v}(\mathbf{r}, \tau)]\} \tag{7.7}$$

Given a vector field $\mathbf{v}(\mathbf{r})$ and a random image $z(\mathbf{r})$, CGP can be computed in a straightforward way by integrating (7.2) and by taking the maximum of $z(\mathbf{r})$ over an arc of the solving trajectory $\mathbf{\Phi_v}(\mathbf{r}, t)$, as indicated in (7.7). For reasons of simplicity, in our implementation we integrated (7.2) numerically by means of the Euler algorithm [23] and we generated $z(\mathbf{r})$ by convolving a white Gaussian noise $n(\mathbf{r})$ with a 2D Gaussian function g_{σ_z} with standard deviation σ_z:

$$z(\mathbf{r}) = n(\mathbf{r}) \star g_{\sigma_z}(\mathbf{r}) \tag{7.8}$$

Examples of CGP are shown in Fig. 7.4, in which the histograms of all images have been equalized for visualization purposes. These patterns are related to the same vector fields $\mathbf{v}(\mathbf{r})$ as for Figs. 7.1, 7.2, and 7.3 and, as we see, the CGP exhibit similar geometric structures to the corresponding discrete GP. Unlike discrete GP, CGP do not loose their miscrostructure when $|\mathbf{v}(\mathbf{r})|$ increases. On the other hand, both for the discrete and the continuous case, the geometric structure disappears as $|\mathbf{v}(\mathbf{r})| \to 0$.

From a mathematical point of view, a CGP can be considered as the output of adaptive morphological dilation of a random noise, where the position-dependent structuring element is an arc of the curve $\mathbf{\Phi_v}(\mathbf{r}, t)$ that solves the differential equation (7.2) (see [24] for a theoretical treatment and a survey on adaptive morphology). As we see in Fig. 7.4, dilating noise as in (7.7) gives rise to a random pattern which resembles curved brush strokes oriented along $\mathbf{v}(\mathbf{r})$, whose length is proportional to $|\mathbf{v}(\mathbf{r})|$.

7.3 Proposed Algorithm

In this section, we show how CGP can be used to add an artistic effect to a photographical image. The proposed algorithm is depicted in Figs. 7.5 and 7.6. We first describe the algorithm for a graylevel input image $I(\mathbf{r})$ (Section 7.3.1) and then we extend the method to the color case (Section 7.3.2).

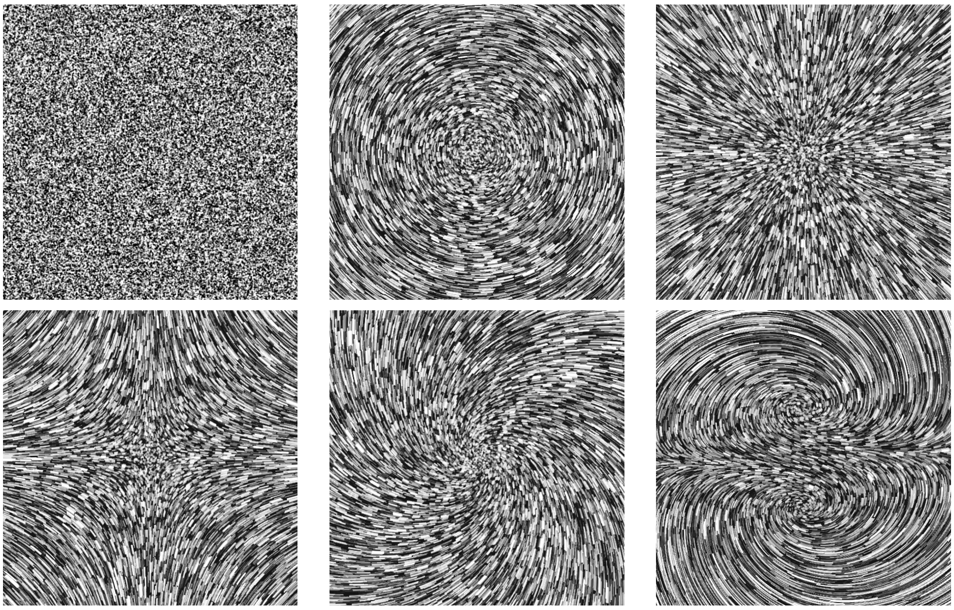


Figure 7.4: *Random image $z(\mathbf{r})$ and examples of CGP obtained from it, using the same vector fields as in Figs. 7.1-7.3. Their geometrical structure is analogous to the discrete case, but it remains well visible also when $|\mathbf{v}(\mathbf{r})|$ increases.*

7.3.1 Graylevel images

Referring to Figs.7.5, the first step is edge preserving smoothing (EPS), which removes texture from the input image whil preserving object contours (Fig. 7.5b). Several algorithms for this task have been proposed in the literature (see, for instance, [25–32]); we use a modification of the operator presented in [33] (see AppendixA for details). Compared with other approaches for EPS, the operator deployed here has the advantage of sharpening edges instead of only preserving them. This is a desirable property because paintings usually have sharper edges than photographic images [33]. We denote the output of EPS $I_{EPS}(\mathbf{r})$.

The second step is the generation of *synthetic painterly texture* (SPT) which simulates oriented brush strokes (Fig. 7.5c). Let $\nabla_\sigma I_{EPS}$ be the scale-dependent gradient of I_{EPS}, defined as the convolution of I_{EPS} with the gradient of a Gaussian function g_σ:

$$\nabla_\sigma I_{EPS} = I_{EPS} \star \nabla g_\sigma \tag{7.9}$$

From it, we construct a vector field $\mathbf{v}(\mathbf{r})$ whose norm we set equal to a constant a and which forms a constant angle θ_0 with the direction $\theta_\sigma(\mathbf{r})$ of $\nabla_\sigma I_{EPS}$, where a and θ_0 are input parameters:

$$\mathbf{v}(\mathbf{r}) = a[\cos(\theta_\sigma(\mathbf{r}) + \theta_0)\,,\ \sin(\theta_\sigma(\mathbf{r}) + \theta_0)]^\mathsf{T} \tag{7.10}$$

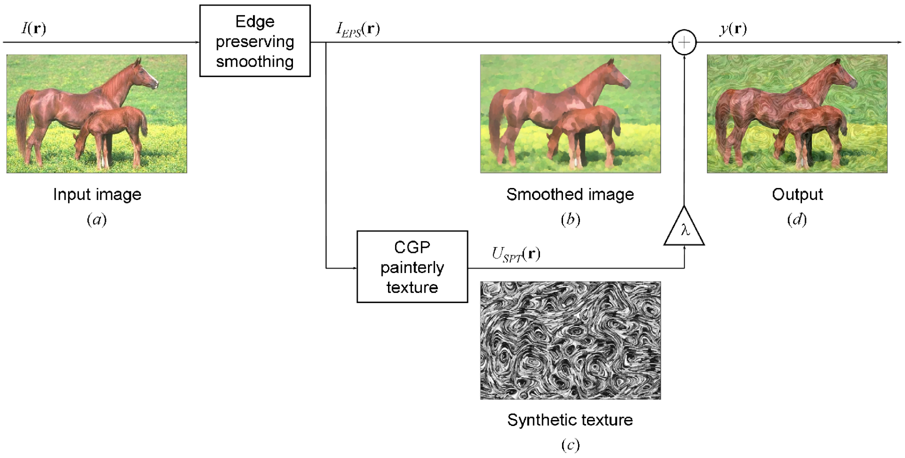


Figure 7.5: *Proposed approach for generation of artistic images.*

The function $\theta_\sigma(\mathbf{r})$ is undefined on points $\mathbf{r}$ for which $|\nabla_\sigma I_{EPS}(\mathbf{r})| = 0$. For such points, we take by definition $\mathbf{v}(\mathbf{r}) = \mathbf{0}$.

In this notation, we define the SPT $U(\mathbf{r})$ as a function η of the CGP $\mathfrak{G}_{\mathbf{v}}(\mathbf{r})$ associated with the vector field defined in (7.10):

$$U(\mathbf{r}) = \eta[\mathfrak{G}_{\mathbf{v}}(\mathbf{r})] \tag{7.11}$$

where η is histogram equalization which results in a flat histogram of $U(\mathbf{r})$ in $\left[-\frac{1}{2}, \frac{1}{2}\right]$.

An example of such SPT is shown in Fig. 7.5c for $\theta_0 = \pi/2$ and $a = 18$, for an image of size (320×480). We see that the geometric structure of SPT is similar to the elongated brush strokes that artists use in paintings. For $\theta_0 = \pi/2$, such strokes are oriented orthogonally to $\nabla_\sigma I_{EPS}(\mathbf{r})$. This mimics the fact that artists usually tend to draw brush strokes along object contours. Moreover, it is easy to prove that for $\theta_0 = \pi/2$ the trajectories $\mathbf{\Phi}_{\mathbf{v}}(\mathbf{r}, t)$ that solve the differential equation (7.2) are closed curves (see Appendix B for a formal proof). Thus, the brush strokes tend to form whirls which are typical of some impressionist paintings.

Finally, our artistic effect is achieved by adding SPT to $I_{EPS}(\mathbf{r})$, thus obtaining the final output $y(\mathbf{r})$ (Fig. 7.5d):

$$y(\mathbf{r}) = I_{EPS}(\mathbf{r}) + \lambda U(\mathbf{r}) \tag{7.12}$$

where the parameter λ controls the strength of the SPT. By comparing Figs. 7.5a and 7.5d, we see that natural texture of the input image has been replaced by SPT. Such a texture manipulation produces images which look like paintings.

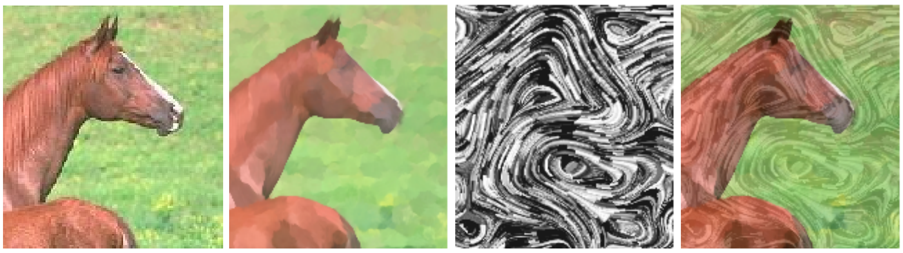


Figure 7.6: *From left to right: crops of an input image $I(\mathbf{r})$, the result $I_{EPS}(\mathbf{r})$ of edge preserving smoothing, the associated synthetic painterly texture $U(\mathbf{r})$, and the final output $y(\mathbf{r})$ for the example of Fig. 7.5.*

7.3.2 Color images

For color images, the straightforward application of the above described algorithm to each color component of the input image would produce undesired color artifacts, even if the same random image $z(\mathbf{r})$ is used for all color components. Therefore, we generate a monochromatic SPT that is added to each color component $I^{(i)}_{EPS}(\mathbf{r})$ of the color image $\mathbf{I}_{EPS}(\mathbf{r})$:

$$y^{(i)}(\mathbf{r}) = I^{(i)}_{EPS}(\mathbf{r}) + \lambda U(\mathbf{r}), \qquad i = 1, 2, 3. \tag{7.13}$$

The CGP $\mathfrak{G}_{\mathbf{v}}(\mathbf{r})$ is still computed by means of the structuring vector field defined in (7.10), where $\theta_\sigma(\mathbf{r})$ is now the orientation of a color gradient [34]. Specifically, $\theta_\sigma(\mathbf{r})$ is given by the direction of the eigenvector associated with the maximum eigenvalue of the following matrix:

$$K_\sigma(\mathbf{r}) = \sum_i \left[\nabla_\sigma I^{(i)}_{EPS}(\mathbf{r})\right] \left[\nabla_\sigma I^{(i)}_{EPS}(\mathbf{r})\right]^{\mathsf{T}} \tag{7.14}$$

The value of $\theta_\sigma(\mathbf{r})$ is undefined on points $\mathbf{r}$ for which the eigenvalues of $K_\sigma(\mathbf{r})$ are equal. Similarly to the graylevel case, for such points we take by definition $\mathbf{v}(\mathbf{r}) = \mathbf{0}$.

7.4 Results and Comparison

In this section some experimental results are presented and commented in order to illustrate the ability of the proposed algorithm to add artistic effects to photographic images, and to study the influence of the input parameters. Our algorithm is compared with two of the most popular existing artistic operators, namely the impressionist rendering (IR) proposed in [35] and the non-photorealistic rendering technique called *artistic vision* (AV) presented in [36]. IR consists in rendering overlapping rectangular brush strokes of a given size and orientation; intersection between strokes and object contours is avoided by means of a stroke clipping procedure based on the Sobel edge detector [37]. In AV, curved brush strokes

Table 7.1: *Values of the parameters used for the studied approaches, with each color component of the input image ranging between 0 and 1.*

Algorithm	**Parameter values**
Proposed approach	$a = 18$, $\sigma = 6$, $\lambda = 0.3$, $\sigma_z = 0.7$, $\theta_0 = \pi/2$
IR [35]	$l = 5$, $w = 1$, $\Delta r = \Delta g = \Delta b = 0.06$, $\Delta I = 0.15$
AV [36]	Segm. levs. = 48, Enh. levs. = 96,
	Brush. Art. lev. = 40, $\alpha = 50\%$

of different sizes are rendered by means of a more sophisticated segmentation approach. We show some results in Figs. 7.7-7.12; a larger set of examples is available at the URL http://www.cs.rug.nl/~imaging/glassart. Unless differently specified, we use the parameter values according to Table 7.1.

The output of the proposed operator, for the input images of Figs. (7.7-7.9)a, is shown in Figs. (7.7-7.9)b. As we see, our operator effectively mimics curved brush strokes oriented along object contours, such as the contour of the bird in Fig. 7.8b. The whirls that are present in contourless areas resemble some impressionist paintings. In Figs (7.7-7.9)c, the outputs of AV are shown for the same input images; though simulation of curved brush strokes is attempted, several artifacts are clearly visible, especially on flat areas. IR (Figs. (7.7-7.9)d) does not produce artifacts, but it tends to render blurry contours, such as the contour of the zebras' heads in Fig. 7.10d, and small object details are lost, such as the legs of the sheep in Fig. 7.7 or the small branches under the bird in Fig. 7.8. Moreover, IR is less effective in rendering impressionist whirls.

We now illustrate the influence of the input parameters on the output of the proposed operator. The parameters σ_z and a, defined in (7.8) and (7.10) respectively, determine the spatial correlation of the SPT along and orthogonally, respectively, to the local brush stroke direction. As shown in Fig. 7.11, larger values of a correspond to longer brush strokes, whereas larger values of σ_z give rise to wider brush strokes. As to the parameter σ defined in (7.9), it controls the degree of smoothness of $\mathbf{v}(\mathbf{r})$ and, consequently, the degree of smoothness of the lines that are traced out by the brush strokes (Fig. 7.12). Larger values of σ also imply larger impressionist whirls. Concerning the parameter θ_0, it controls the angle between the brush strokes and the nearest object contours. In Fig. 7.13 it is shown how different artistic effects can be achieved by varying the value of θ_0. As we can see, for $\theta = \pi/2$, the strokes follow the object contours and form whirls in flat areas, while for $\theta = 0$ the strokes are orthogonal to the contours and build star-like formations in flat regions. Finally, the parameter λ defined in (7.12) controls the strength of SPT.

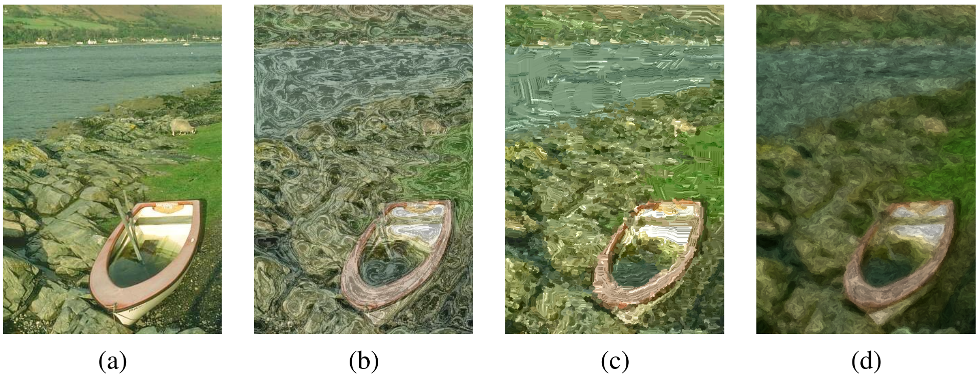


Figure 7.7: *(a) An input image and outputs of (b) the proposed algorithm, (c) AV [36], and (d) IR [35].*

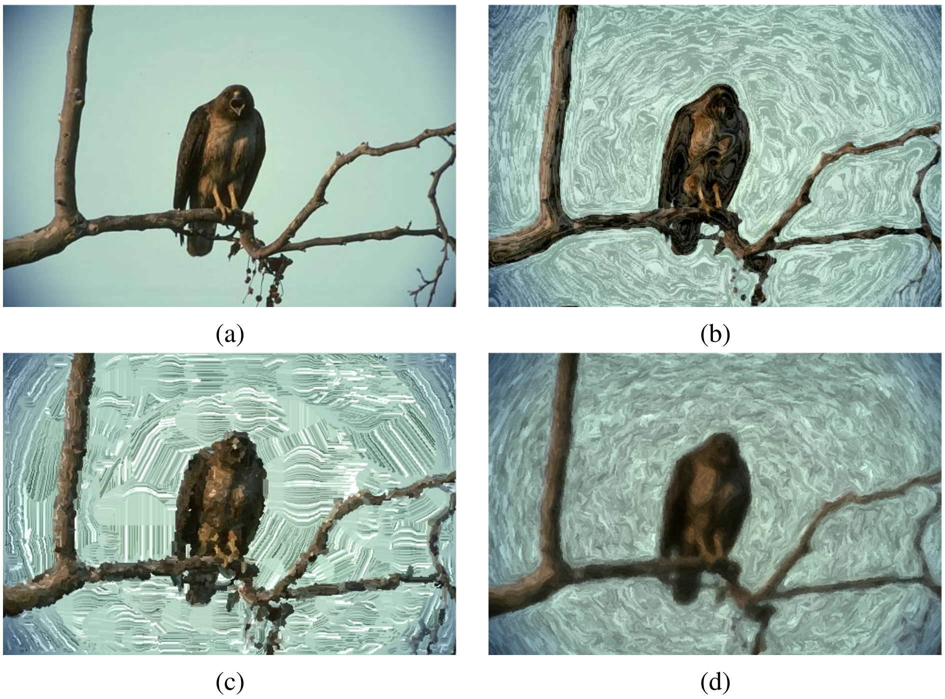


Figure 7.8: *(a) An input image and outputs of (b) the proposed algorithm, (c) AV [36], and (d) IR [35].*

7.5 Discussion and Conclusions

The classic approach to produce painting-like images with the aid of a computer consists in generating a set of possibly overlapping brush strokes, which are rendered in a certain order

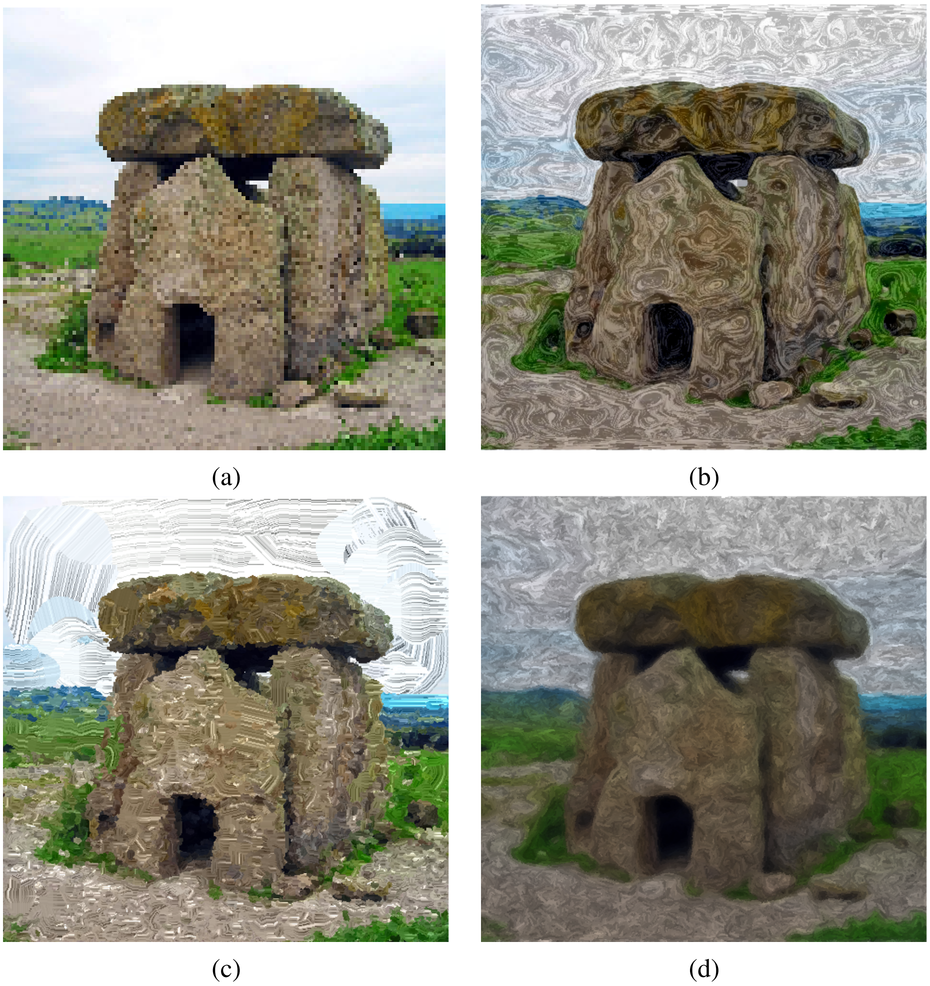
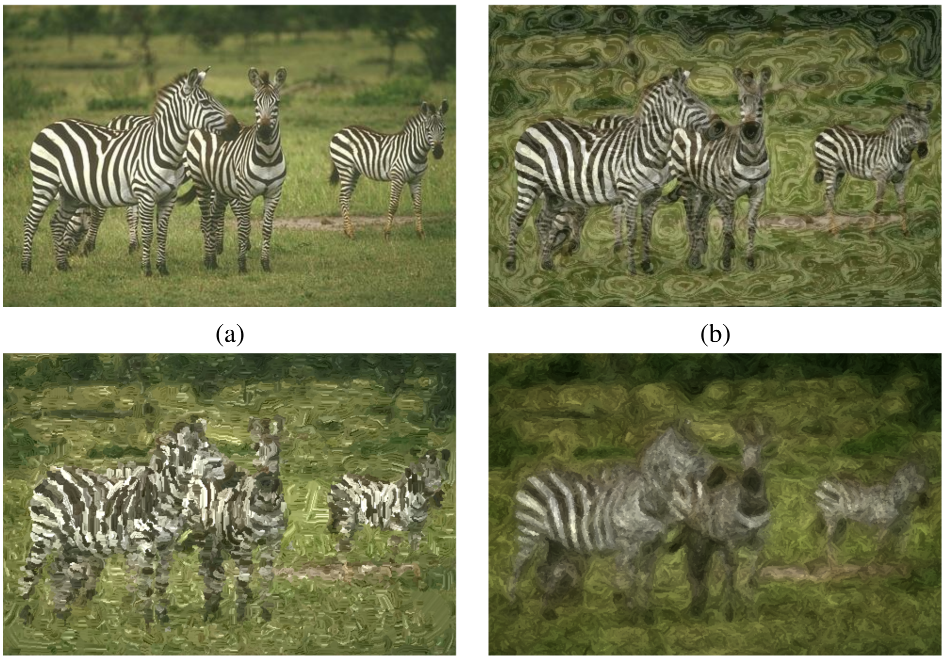


Figure 7.9: *(a) An input image and outputs of (b) the proposed algorithm, (c) AV [36], and (d) IR [35].*

on a white or canvas-textured background. Early works [38] proposed supervised methods in which the user had to specify position, shape and orientation of each brush stroke. Since this requires a considerable effort from the user, much research has been carried out in order to develop algorithms with higher degree of automatization [35, 39–47]. Particular attention has been paid to the generation of curved brush strokes, brush strokes of variable size, and textured brush strokes. As for the latter point, several algorithms for brush strokes rendering have been proposed, based on physical models of the phenomenology related to the diffusion of pigments on paper [48]. Other development concern the possibility to render strokes on



(a) (b)

(c) (d)

Figure 7.10: *(a) An input image and outputs of (b) the proposed algorithm, (c) AV [36], and (d) IR [35].*

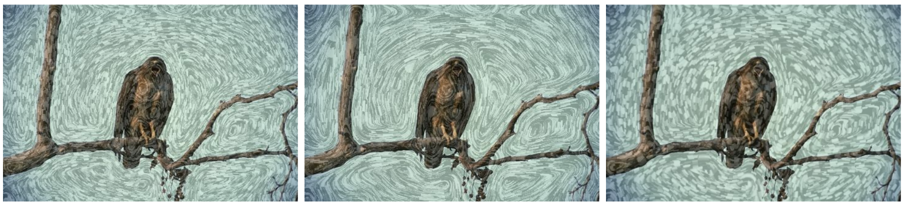


Figure 7.11: *Illustration of the influence of the parameters σ_z and a, which control width and length of the brush strokes. From left to right: output of the proposed operator for the input image of Fig. 7.8, with (left) $\sigma_z = 0.6$, $a = 8$, (centre) $\sigma_z = 0.6$, $a = 16$, and (right) $\sigma_z = 1.2$, $a = 8$.*

top of an image instead of a white background (underpainting) [36], the use of global saliency maps to determine the order in which brush strokes must be rendered [45], or methods based on some properties of the human visual system [41, 46]. In [44], it is shown that representing a synthetic painterly image in terms of a list of brush strokes attributes results in a much higher lossless compression ratio with respect to well established general purpose image compressors. For a survey of these techniques we refer to [49].

The methods referred above can achieve interesting artistic effects. However, most of

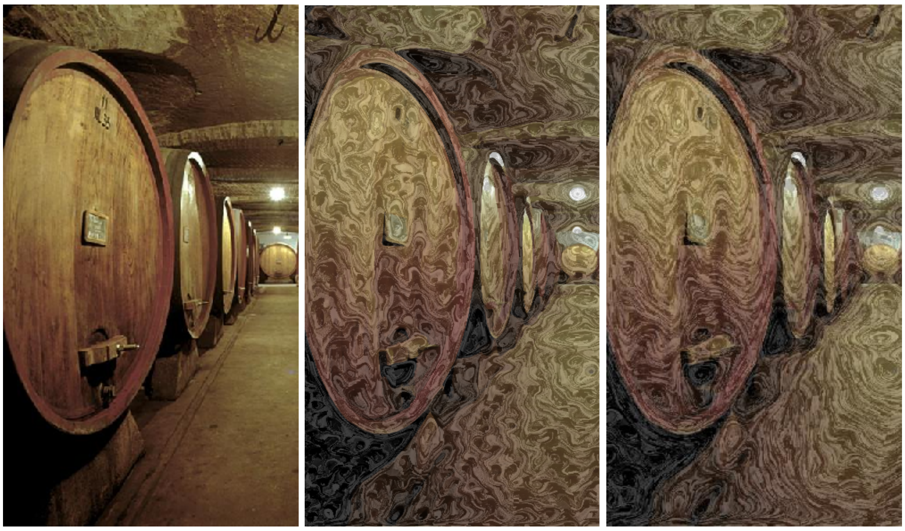
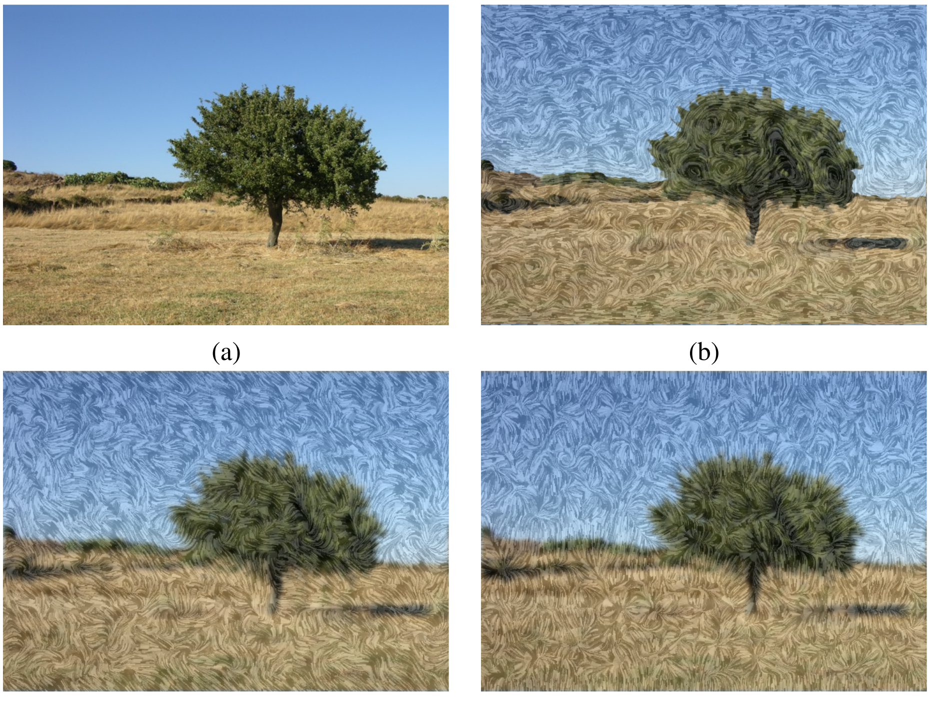


Figure 7.12: *Illustration of the influence of the parameter σ. From left to right: Input image and outputs of the proposed operator for $\sigma = 4$ and $\sigma = 8$. In the second case the lines traced out by the brush strokes are smoother and the average size of the impressionist whirls is larger.*

them strongly rely on heuristics and require a large number of intermediate computational steps, such as edge detection or image segmentation, which are sensitive to parameter settings. As admitted in [36], these algorithms may fail as soon as one of the intermediate operations does not perform well for some particular input image. Moreover, some of the input parameters do not have a clear interpretation in the context of painting simulation (such as, for instance, the threshold of the Sobel edge detector deployed in [35]). In contrast, we achieve artistic effects by replacing the natural texture of the input image by synthetic texture which simulates brush strokes, rather than generating a list of strokes and specifying the attributes of each one of them (position, shape, orientation, color, texture). This results in a much simpler approach, which comprises only two elementary steps - EPS and generation of SPT - which are robust to variations of the input image. The input parameters are easily interpreted in terms of size of the brush strokes, size of the impressionist whirls, and angle between brush strokes and the nearest object contours (Figs. 7.11 - 7.13).

In the proposed method, a key role is played by the vector field $\mathbf{v}(\mathbf{r})$, which determines the geometric structure of the SPT and, consequently, the local orientation of the brush strokes. We have chosen to relate $\mathbf{v}(\mathbf{r})$ to $\nabla_\sigma I_{EPS}$ as defined in (7.10) for reasons of conceptual simplicity and low computational demand. However, different vector fields $\mathbf{v}(\mathbf{r})$ may be used as well. For instance, deploying anisotropic diffusion (AD) for the computation of the gradient of I_{EPS}, instead of Gaussian blurring, would bring improvement in presence of sharp corners (Fig. 7.14). As we can see, with Gaussian blurring the lines traced


(a)

(b)

(c)

(d)

Figure 7.13: *Illustration of the influence of the parameter θ_0. (a) Input image and outputs of the proposed operator for (b) $\theta_0 = \pi/2$, (c) $\theta_0 = \pi/4$, and (d) $\theta_0 = 0$. For the last two images, EPS has been performed with $q = 0$.*

out by the brush strokes may be too rotund and, therefore, they may cross object contours (Fig. 7.14, left), whereas AD overcomes this problem. On the other hand, with respect to Gaussian blurring, AD is computationally more demanding, less robust to noise, and it may give rise to spurious whirls, such as the one on the front side of the horse's head shown in Fig. 7.14b. In general, there is no universal algorithm for the computation of $\mathbf{v}(\mathbf{r})$ from an input image and its choice is application dependent.

CGP share some similarities with the line integral convolution (LIC), which has been proposed in [50] with the purpose of representing a vector field by means of oriented texture. However, CGP result in a better brush strokes simulation than LIC, especially for large values of a and σ_z, thus they are more suitable for artistic imaging applications (compare Figs. 7.15a-b). In particular, since the intensity profile of a CGP is more piece-wise constant with respect to LIC, brush strokes are more visible and their edges are sharper. Moreover, if a CGP with large brush strokes is linearly combined with a LIC associated with the same vector field, for small values of a and σ_z (Fig. 7.15c), it is possible to simulate textured brush

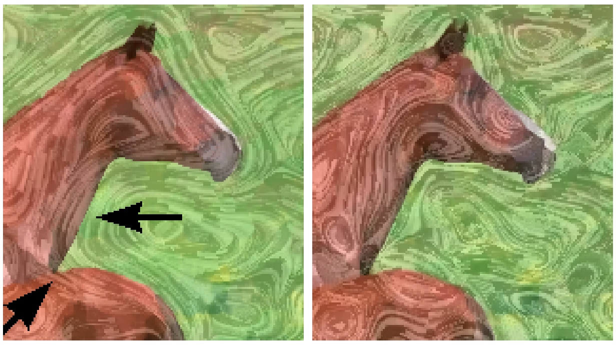


Figure 7.14: *Possible improvement obtained by replacing Gaussian smoothing by anisotropic diffusion. (a) Output of the proposed operator for the input image of Fig.7.6a; the arrows mark rotund brush strokes that cross the contour of the horse. (b) Improvement brought by using anisotropic diffusion.*

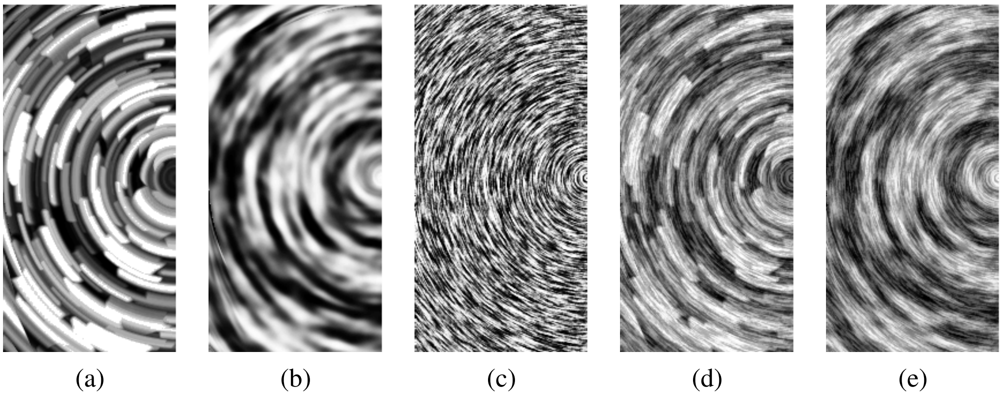


Figure 7.15: *Comparison between CGP and LIC. (a) CGP and (b) LIC associated with the vector fields* $\mathbf{v}(x,y) = \frac{(-x,y)}{\sqrt{x^2+y^2}}$, *with* $a = 24$ *and* $\sigma_z = 3$. *Compared to CGP, LIC results in a noticeably worse brush strokes simulation. (c) LIC associated with the same vector field, with* $a = 6$ *and* $\sigma_z = 0.6$. *(d) Linear combination of patterns (a) and (c), which simulates textures brush strokes. (e) Linear combination of patterns (b) and (c), which fails in simulating textured brush strokes.*

strokes (Fig. 7.15d). Such an interesting possibility cannot be achieved by only using LICs, as shown in Fig. 7.15e. In Fig. 7.16, the result of such a technique is shown for a natural image (1176 × 990 pixel).

To summarize, we have shown that the mathematical formalism related to GP, which are originally defined as dot patterns, can be easily extended to the continuous case, while maintaining the same geometric structure as in the discrete case. CGP offer several advantages over the discrete GP: first, their visibility does not decrease when $|\mathbf{v}(\mathbf{r})|$ increases and, second, they are more suitable for image processing tasks. As an application, we show that CGP

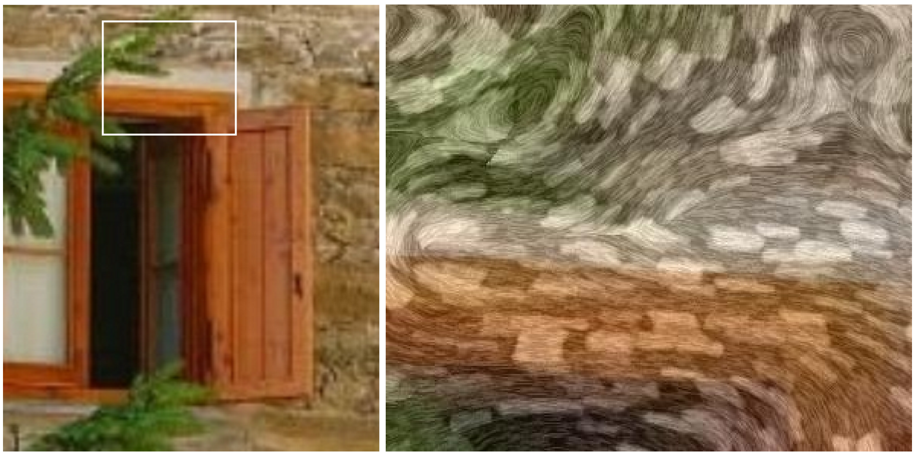


Figure 7.16: *(Left) Input image (*1176×990 *pixel), and (Right) the result of our painterly rendering with textured brush strokes (crop).*

can be used to produce nice artistic effects. Experimental results on a broad range of input images validate the effectiveness of the proposed method. Due to its simplicity, input parameters are easily interpretable, and undesired artifacts that affect some existing algorithms are avoided. Moreover, the possibility to achieve considerably different artistic effects by simply changing the values of certain input parameters (such as θ_0, see Fig. 7.13) makes the method versatile and promising.

Appendix A. Edge Preserving Smoothing

In this appendix we briefly describe the operator we deploy for EPS. Let $I(\mathbf{r})$ be a graylevel image and let $m(\mathbf{r})$ and $s(\mathbf{r})$ be the local average and the local standard deviation of $I(\mathbf{r})$, computed as the following two convolutions:

$$m = I \star g_\sigma, \qquad s^2 = (I^2 \star g_\sigma) - m^2 \tag{7.15}$$

The output $\Psi_q(\mathbf{r})$ of the EPS deployed here is a weighted local average of the values of $m(\mathbf{r})$, with weights proportional to $[s(\mathbf{r})]^{-q}$, computed as follows:

$$I_{EPS} = \Psi_q = \frac{(ms^{-q}) \star g_\sigma}{s^{-q} \star g_\sigma} = \frac{\int m(\boldsymbol{\rho})[s(\boldsymbol{\rho})]^{-q} g_\sigma(\mathbf{r} - \boldsymbol{\rho}) \mathrm{d}\boldsymbol{\rho}}{\int [s(\boldsymbol{\rho})]^{-q} g_\sigma(\mathbf{r} - \boldsymbol{\rho}) \mathrm{d}\boldsymbol{\rho}} \tag{7.16}$$

where $q \geq 0$ is an input parameter.

The behavior of this operator is illustrated in Fig. 7.17. If point $\boldsymbol{\rho}$ is sufficiently close to an edge (Fig. 7.17a), the local standard deviation $s(\boldsymbol{\rho})$ is high, thus the term $m(\boldsymbol{\rho})[s(\boldsymbol{\rho})]^{-q}$

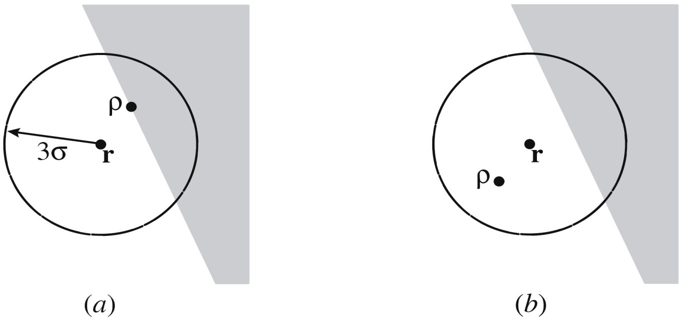


Figure 7.17: *Behavior of the deployed EPS, when point $\boldsymbol{\rho}$ is (a) close and (b) far from an edge*

gives a negligible contribution to the numerator of (7.16). On the contrary, if $\boldsymbol{\rho}$ is far from edges (Fig. 7.17b), $s(\boldsymbol{\rho})$ is small and the contribution of $m(\boldsymbol{\rho})[s(\boldsymbol{\rho})]^{-q}$ is high. In this way, smoothing across edges is avoided. In (7.16), the Gaussian term $g_\sigma(\mathbf{r}-\boldsymbol{\rho})$ limits the local averaging of $m(\boldsymbol{\rho})$ to a neighborhood of $\mathbf{r}$, giving more weight to points $\boldsymbol{\rho}$ that are closer to $\mathbf{r}$, and the denominator plays the role of a normalizing factor.

The parameter q controls the sharpness of the edges in the output image: for $q = 0$, (7.16) reduces to the convolution of the input image $I(\mathbf{r})$ with a Gaussian function with standard deviation equal to $\sigma\sqrt{2}$; this corresponds to the minimum edge sharpness. On the other hand, for $q \to \infty$, we have $\Psi_\infty(\mathbf{r}) = m[\boldsymbol{\rho}_{min}(\mathbf{r})]$, where $\boldsymbol{\rho}_{min}(\mathbf{r})$ is the value of $\boldsymbol{\rho}$, within a neighborhood of $\mathbf{r}$ of radius, say, 3σ for which $s(\mathbf{r})$ is minimum. This corresponds to the Gaussian-Kuwahara EPS [51, 52], which gives rise to the maximum edge sharpness.

For other values of q, the EPS deployed here is an intermediate case between Gaussian smoothing and the Gaussian-Kuwahara EPS. In Fig. 7.18, the performance of the proposed operator in terms of texture and noise rejection and edge degradation are plotted versus each other for different values of q. Texture and noise rejection is measured in terms of PSNR (dB), while edge degradation ξ is defined as follows:

$$\xi = \frac{1}{N_E}\int [I(\mathbf{r}) - I_{EPS}(\mathbf{r})]^2 \,\mathrm{d}\mathbf{r} \tag{7.17}$$

where $I(\mathbf{r})$ is a synthetic image for which the number N_E of edge pixels is known. As we see in Fig. 7.18, when the optimum is reached ($q = 4$), we achieve almost the same noise rejection as with Gaussian smoothing and almost the same edge preservation as with Gaussian Kuwahara operator.

This EPS is an evolution of the operator presented in [33]. Though the outputs of the two algorithms are similar, the approach proposed here has the following advantages: (i) lower computational complexity, since here only four separable convolutions are required, whereas in [33] a larger number of non-separable convolutions is involved. (ii) Higher texture and

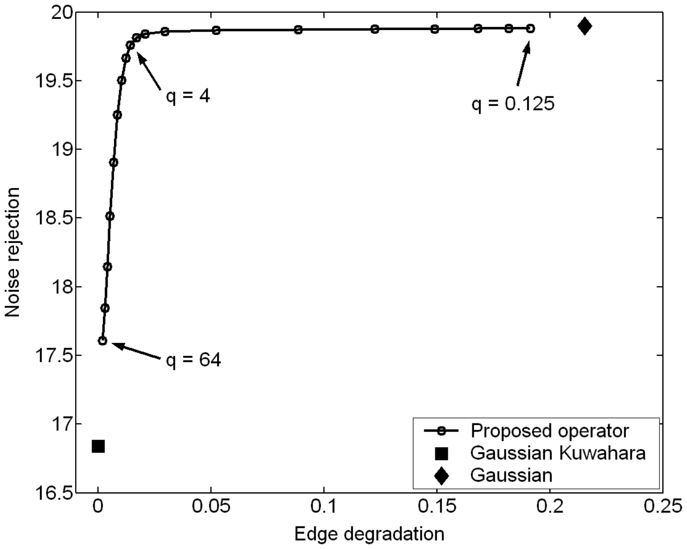


Figure 7.18: *Performance of the proposed EPS operator, for different values of q, in terms of noise rejection and edge degradation. When the optimum is reached, we achieve almost the same noise rejection as a Gaussian smoothing and almost the same edge preservation as the Gaussian Kuwahara operator.*

noise rejection: Here, almost the same amount of texture is removed as by Gaussian filter with standard deviation equal to $\sigma\sqrt{2}$, while the operator proposed in [33], for the same edge preservation, removes less texture than a Gaussian filter with standard deviation equal to σ. (iii) Smaller number of input parameters, since in [33] an additional parameter N is introduced.

Appendix B. Analysis of the algorithm

In this appendix we prove some analytical properties of the streamlines $\mathbf{r}(t) = \mathbf{\Phi}_{\mathbf{v}}(\mathbf{r}_0, t)$ of the vector field $\mathbf{v}(\mathbf{r})$ defined in (7.10), which solve the differential equation (7.2). The concerned vector field can be written as:

$$\mathbf{v}(\mathbf{r}) = \frac{(\nabla_\sigma I_{EPS})\cos\theta_0 + (\nabla_\sigma^\perp I_{EPS})\sin\theta_0}{|\nabla_\sigma I_{EPS}|} \tag{7.18}$$

with $\nabla_\sigma = g_\sigma \star [\partial/\partial x, \partial/\partial y]^\intercal$, $\nabla_\sigma^\perp = g_\sigma \star [-\partial/\partial y, \partial/\partial x]^\intercal$. Since the multiplication of $\mathbf{v}(\mathbf{r})$ for any nonzero scalar function does not change the streamlines of $\mathbf{v}(\mathbf{r})$, but only the local speed at which they are traced out, we can simplify our study by considering the vector field $\mathbf{w}(\mathbf{r}) = |\nabla_\sigma I_{EPS}(\mathbf{r})|\mathbf{v}(\mathbf{r})$ instead of $\mathbf{v}(\mathbf{r})$:

$$\mathbf{w}(\mathbf{r}) = (\nabla_\sigma I_{EPS}) \cos\theta_0 + (\nabla_\sigma^\perp I_{EPS}) \sin\theta_0. \tag{7.19}$$

We formally prove that, for $\theta_0 = \pi/2$, the following two facts hold: (T_1) streamlines of $\mathbf{v}(\mathbf{r})$ are closed curves. (T_2) such curves are traced around points $\mathbf{r}_k$ such that $\mathbf{w}(\mathbf{r}_k) = \mathbf{0}$ and the determinant $\det H$ of the Hessian matrix of $I_{EPS} \star g_\sigma$ is positive.

In order to prove (T_1) we observe that, from basic algebra, the divergence of $\mathbf{w}(\mathbf{r})$ is equal to:

$$\nabla\cdot \mathbf{w}(\mathbf{r}) = \nabla_s igma^2 I_{EPS} \cos\theta_0 \tag{7.20}$$

Therefore, for $\theta_0 = \pi/2$, $\nabla\cdot \mathbf{w}(\mathbf{r})$ is zero everywhere in $\mathbb{R}^2$ and the vector field $\mathbf{w}(\mathbf{r})$ is solenoidal. Thus, in force of a well-known theorem in topology, the streamlines of $\mathbf{w}(\mathbf{r})$ must necessarily be closed curves[2].

Concerning (T_2), a Taylor expansion yields:

$$\mathbf{w}(\mathbf{r}) = H\cdot (\mathbf{r} - \mathbf{r}_k) + O(|\mathbf{r} - \mathbf{r}_k|) \tag{7.21}$$

As well known, a singular point $\mathbf{r}_k$ is a centre of a closed curve iff the eigenvalues of H have zero real part. Basic algebra shows that, for the vector field $\mathbf{w}(\mathbf{r})$ defined in (7.19), the eigenvalues $\lambda_{1,2}$ of H are equal to:

$$\lambda_{1,2} = \frac{cos\theta_0 \mathrm{tr} H \pm \sqrt{cos^2\theta_0 \mathrm{tr}^2 H - 4\det H}}{2} \tag{7.22}$$

where $\mathrm{tr}H$ is the trace of H. By means of the inequality $\mathrm{tr}^2 H > 4\det H$, which holds for any symmetric matrix, it is straightforward to prove that $\lambda_{1,2}$ are purely imaginary iff $\theta_0 = \pi/2$. This completes the proof.

The consequences of these analytical facts to the behavior of the proposed artistic operator are illustrated in Fig.7.19. It is shown that the centers of the impressionist whirls are located at the zeros of the gradient magnitude of I_{EPS}, for which $\mathbf{w}(\mathbf{r}) = \mathbf{0}$. Since the density of such points decreases quadratically with σ, the above discussion also provides an analytical explanations of why the size of the impressionist whirls grows with σ. In contrast, for $\theta_0 \neq \pi/2$, the streamlines of $\mathbf{w}(\mathbf{r})$ and $\mathbf{v}(\mathbf{r})$ are no longer closed lines, thus the artistic effect shown in Figs. 7.13c-d is achieved.

7.6 Bibliography

[1] L. Glass. Moiré effect from random dots. *Nature*, 223:578–580, 1969.

[2] Unbounded curves are considered closed since they close themselves at infinity.

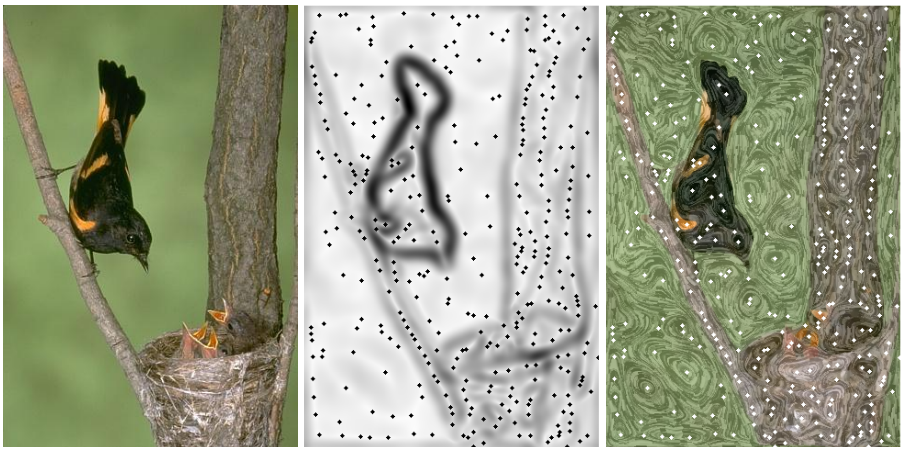


Figure 7.19: *From left to right: input image, gradient magnitude of $I_{EPS}(\mathbf{r})$ (black dots mark the zeros of the gradient magnitude), and output of the proposed operator. The centers of the impressionist whirls correspond to the zeros of the gradient magnitude.*

[2] L. Glass and R. Perez. Perception of random dot interference patterns. *Nature*, 246:360–362, 1973.

[3] L. Glass and E. Switkes. Pattern recognition in humans: Correlations which cannot be perceived. *Perception*, 5(1):67–72, 1976.

[4] M.A. Smith, W. Bair, and J.A. Movshon. Signals in macaque striate cortical neurons that support the perception of Glass patterns. *Journal of Neuroscience*, 22(18):8334–8345, 2002.

[5] M.A. Smith, A. Kohn, and J.A. Movshon. Glass pattern responses in macaque V2 neurons. *Journal of Vision*, 7(3):5, 2007.

[6] S.C. Yen and L.H. Finkel. Extraction of perceptually salient contours by striate cortical networks. *Vis. Res.*, 38(5):719–741, 1998.

[7] Z. Li. A neural model of contour integration in the primary visual cortex. *Neur. Comp.*, 10(4):903–940, 1998.

[8] R.K. Maloney, G.J. Mitchison, and H.B. Barlow. Limit to the detection of Glass patterns in the presence of noise. *J. Opt. Soc. Am. A*, 4(12):2336–2341, 1987.

[9] H.R. Wilson and F. Wilkinson. Detection of global structure in Glass patterns: implications for form vision. *Vision Research*, 38(19):2933–2947, 1998.

[10] L. Seu and V.P. Ferrera. Detection thresholds for spiral Glass patterns. *Vision Research*, 41(28):3785–3790, 2001.

[11] S.C. Dakin. The detection of structure in Glass patterns: Psychophysics and computational models. *Vision Research*, 37(16):2227–2246, 1997.

[12] K. Prazdny. Psychophysical and computational studies of random-dot moire patterns. *Spatial Vision*, 1(3):231–242, 1986.

[13] T.H. Phillips and A. Rosenfeld. A simplified method of detecting structure in Glass patterns. *Pattern Recognition Letters*, 4(3):213–217, 1986.

[14] H. Greenspan, M. Porat, and Y.Y. Zeevi. Projection-based approach to image analysis: pattern recognition and representation in the position-orientation space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1105–1110, 1992.

[15] J.A. Wilson, E. Switkes, and R.L. De Valois. Glass pattern studies of local and global processing of contrast variations. *Vision Research*, 44(22):2629–2641, 2004.

[16] I. Amidror. Unified approach for the explanation of stochastic and periodic moirés. *Journal of Electronic Imaging*, 12(4):669–681, 2003.

[17] I. Amidror. Glass patterns as moiré effects: new surprising results. *Optics Letters*, 28(1):7–9, 2003.

[18] I. Amidror. *The theory of the Moiré phenomenon, volume 1: periodic layers*. Springer, 2000.

[19] I. Amidror. *The theory of the Moiré phenomenon, Volume 2: Aperiodic layers*. springer, 2007.

[20] L. Glass. Looking at dots. *Math. Intell.*, 24(4):37–43, 2002.

[21] I. Amidror. Dot trajectories in the superposition of random screens: analysis and synthesis. *Journal of the Optical Society of America A*, 21(8):1472–1487, 2004.

[22] S.M. Anstis. Phi movement as a subtraction process. *Vision Research*, 10(12):1411–143030, 1970.

[23] G. Hall and J.M. Watt. *Modern Numerical Methods for Ordinary Differential Equations*. Clarendon press, Oxford, UK, 1976.

[24] N. Bouaynaya, M. Charif-Chefchaouni, and D. Schonfeld. Theoretical foundations of spatially-variant mathematical morphology part I: Binary images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):823–850, 2008.

[25] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[26] L.I. Rudin, S.J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[27] P. Saint-Marc, J.S. Chen, and G. Medioni. Adaptive smoothing: a general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529, 1991.

[28] X. Yang and P.S. Toh. Adaptive fuzzy multilevel median filter. *IEEE Transactions on Image Processing*, 4(5):680–682, 1995.

[29] S. Teboul, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Variational approach for edge-preserving regularization using coupled PDEs. *IEEE Transactions on Image Processing*, 7(3):387–397, 1998.

[30] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, page 839. Bombay: Narosa Publishing House, 1998.

[31] D. Barash. Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, 2002.

[32] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.*, 21(3):257–266, 2002.

[33] G. Papari, N. Petkov, and P Campisi. Artistic edge and corner enhancing smoothing. *IEEE Transactions on Image Processing*, 29(10):2449–2462, 2007.

[34] A. Cumani. Edge detection in multispectral images. *CVGIP*, 53(1):40–51, 1991.

[35] P. Litwinowicz. Processing images and video for an impressionist effect. In *Siggraph*, pages 407–414, 1997.

[36] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: painterly rendering using computer vision techniques. In *NPAR*, pages 83–90, 2002.

[37] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989.

[38] P. Haeberli. Paint by numbers: Abstract image representations. *Computer Graphics*, 24(4):207–214, 1990.

[39] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Siggraph*, pages 453–460, 1998.

[40] M. Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *NPAR*, pages 53–58, 2000.

[41] D. De Carlo and A. Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics (TOG)*, 21(3):769–776, 2002.

[42] N. Li and Zhiyong Huang. Feature-guided painterly image rendering. In *ICIP*, pages 653–656, 2002.

[43] L. Kovacs and T. Sziranyi. Efficient coding of stroke-rendered paintings. In *ICPR*, pages II: 835–838, 2004.

[44] L. Kovács and T. Szirányi. Painterly rendering controlled by multiscale image features. In *SCCG*, pages 177–184, 2004.

[45] J.P. Collomosse and P.M. Hall. Salience-adaptive painterly rendering using genetic search. *International Journal on Artificial Intelligence Tools*, 15(4):551–575, 2006.

[46] S. Nunes, D. Almeida, V. Brito, J. Carvalho, J. Rodrigues, and J.M.H. du Buf. Perception-based painterly rendering: functionality and interface design. *Ibero-American Symp. Comp. Graphics*, pages 53–60, 2006.

[47] A. Kasao and K. Miyata. Algorithmic painter: a NPR method to generate various styles of painting. *The Visual Computer*, 22(1):14–27, 2006.

[48] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, and D. Salesin. Computer-generated watercolor. In *Siggraph*, pages 421–430, 1997.

[49] A. Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, 2003.

[50] B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In Lynn Valastyan and Laura Walsh, editors, *Proceedings of the Annual Conference on Computer Graphics*, pages 263–270, New York, NY, USA, 1993. ACM Press.

[51] Mark A. Schulze and John A. Pearce. A morphology-based filter structure for edge-enhancing smoothing. In *ICIP*, pages 530–534, 1994.

[52] R. van den Boomgaard. Decomposition of the Kuwahara-Nagao operator in terms of a linear smoothing and a morphological sharpening. In *ISMM*, pages 283–292. CSIRO Publishing, 2002.