Computational Aesthetics in Graphics, Visualization, and Imaging
Paul Rosin (Editor)

EXPRESSIVE 2015

Image Warping for a Painterly Effect

Jiayu Li† and David Mould‡

Carleton University, Ottawa, Canada

Figure 1: *Example results from applying our method. (Hood; cathedral; garlic.)*

Abstract

We propose a two-stage approach to painterly rendering of photographs, where the image plane is first warped to produce a distorted or caricatured effect and then the resulting image is rendered with a painterly effect. We use SLIC superpixels to obtain an oversegmentation, and assign spring parameters uniformly to all pixels within a region; then, the mass-spring simulation distorts the plane in a random but content-sensitive way. With aggressive warping, the subsequent painterly rendering can be done lightly and need not remove much detail. The resulting renderings convey a sense of being painted and leave a sense of being handmade and not overly beholden to the photographic scene.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Bitmap and framebuffer operations

1. Introduction

Painterly rendering is among the oldest styles in non-photorealistic rendering, dating back to Haeberli's seminal work on user-assisted image stylization [Hae90]. Many others have sought to create synthetic painted images since, whether using dedicated simulations of paint [BWL04], image processing [BKTS06, Her98], or particle systems over geometric models [Mei96].

The quality of synthetic painted images has risen steadily over the years. Nonetheless, painterly rendering of geometry and painterly effects based on input photographs usually adhere closely to the structure of the input, and thus capture only a limited range of the possible painterly images; historically, paintings have spanned a wide range from meticulously detailed representations to wholly abstract images.

In this paper, we propose to use image warping to modify an input image before conducting a painterly stylization, thus producing a lively outcome which resembles caricature or other fanciful, exaggerated semi-representational depictions of the subject matter. By manipulating the image before applying the painterly effect, we can convey the impression of a painted image with a painterly filter that is less aggres-

† jiayuli@cmail.carleton.ca
‡ mould@scs.carleton.ca

sive, hence more able to preserve the details in the initial photograph.

Figure 2 helps to illustrate our objective. Above, we see two paintings from Jean-Baptiste Oudry and El Greco that show deliberate distortions and non-photorealistic perspectives. Below, oil paintings from Caravaggio and John William Waterhouse show a high level of detail and fidelity to the original subject matter. Non-photorealistic painterly stylizations of images have concentrated on abstracted images, often using visible strokes in an expressionist style to emphasize the painterly effect. Here, we attempt to create the impression of a painted image without visible brushstrokes.

Figure 2: *Historical paintings illustrating our intention. Above: paintings by Oudry and El Greco demonstrating distortions and problematic perspectives. Below: paintings by Caravaggio and Waterhouse in a photorealistic style.*

Our approach divides the image processing into two phases. First, we distort the image by warping the pixel locations in the image plane. Second, we apply a painterly effect to the warped image; we suggest using a relatively non-intrusive painterly filter so that the details of the input image remain visible. The postprocessing has the benefit of reunifying the warped image, concealing defects that might otherwise be visible after the warping phase. Some example results can be seen in Figure 1. Note that the degree of warping is controllable: we can obtain a delicate effect with little warping, a fairly plausible painterly effect using a medium degree of warping, or a more extreme caricature by warping the image even more heavily. We prefer the results in the middle of this range, but the extremes are available to those who might want to use them to obtain a particular effect.

We make two contributions in this paper:

• We suggest constructing lively painterly images by a two-stage process of first warping the image plane and then applying an image-space painterly filter to the warped image. There is limited precedent for this approach in general painterly rendering.

• We provide a specific mechanism for accomplishing the painterly rendering process just described. In our approach, the warping is accomplished by performing a mass-spring simulation over the image lattice. The painterly effect can be produced by any of several existing image-space painterly rendering methods; for this paper, we primarily rely on a variation of the morphological watercolor effect of Bousseau et al. [BNTS07].

This paper is organized as follows. We discuss related work in the next section. Section 3 describes our approach in detail, with Section 4 giving results of applying the method to sample images and showing comparisons to other methods. The paper's final section concludes and provides suggestions about possible future directions.

2. Previous Work

Painterly rendering has been a major topic in the non-photorealistic rendering literature. Broadly speaking, synthetic painted images can be created in three ways: through digital artists exercising synthetic painting tools [BWL04, CT05]; through paint primitives being distributed according to the details of a geometric scene [Mei96]; or through image-space operations over an input image, usually a photograph, e.g., distributing strokes that match local image properties [Her98]. We are mainly interested in the last of these approaches.

Among the most-studied forms of painterly rendering is portraiture, with specialized techniques to draw human faces. Perhaps the most effective method is that of Zhou and Zhu [ZZ11], who use active shape models to obtain an estimate of facial feature locations, and then learn a mapping from user-drawn strokes to the face geometry. By using a large database of hand-painted portaits, their mapping is made to be fairly reliable; novel portraits can then be made from photos by estimating the face structure and drawing from the stroke placement database. This example-based system produces reasonably realistic drawings, albeit abstracted owing to the small number of strokes. An earlier method that is somewhat closer to our intent is provided by Gooch et al. [GRG04], who automatically estimate facial feature locations in structured photographs, then distort the face shapes to create caricatures. This work is effective but by its nature restricted to portraits with empty backgrounds. Also, the proposed rendering is a monochrome pen-and-ink style, not a painterly rendering.

In general, semantically meaningful image manipulations are difficult to achieve automatically. Two standard workarounds are either to employ user-provided semantics [ZZ10] or to restrict the subject matter and layout of the input images, as in the case of portraiture [GRG04]. In this paper, we hope to create painterly versions of arbitrary photographs, with no limitations on the layout or content. While we do not achieve the quality of results that can be attained by specialized methods applied within their target domains, this is a necessary tradeoff for the robustness we gain instead.

Painterly rendering from photographs suffers from the fixed perspective of the lens. In the case of object-space rendering, the perspective can be altered as the scene is projected onto the image plane, either through interpolating multiple linear perspectives [Sin02] or potentially by performing an entirely nonlinear projection [BSCS07, CS04]. However, such techniques depend on having a full scene description; image-space techniques are more widely applicable.

Abstraction in non-photorealistic rendering falls generally into two categories. First, a basic abstraction can be achieved using a filter to remove details: various linear and nonlinear filters have been proposed for this purpose, and we consider stroke-based rendering a special case of such filters, where an image's pixel colors are replaced b the stroke colors. Second, more deliberate shape abstraction is possible using a higher-level representation of the image; the most common approach here is to create an initial segmentation of the image and then simplify the resulting segments [Mou03, DS02, WLL*06]. We are less interested in purely abstracting the image by removing detail than in reducing its faithfulness to the original photograph; we prefer our final image to be quite detailed in spite of the abstraction process. Of course, quite extreme stylizations are still possible, ranging from somewhat representational forms as in cubism [CH03] to quite abstract forms as seen in the "arty shapes" of Song et al. [SRHC08].

The cubist rendering system proposed by Collomosse and Hall [CH03] has some stages in common with our approach. Taking multiple images as input, this system detects salient image elements, distorts them through transforming a superquadric fit to the element perimeter to a different superquadric, and then fuses multiple elements into a single image to produce a multi-perspective cubist style output. A customized painterly filter helps with the final fusion. Our pipeline uses a single image and attempts to produce a more conventionally representational output image rather than the cubist-style renderings sought by Collomosse and Hall.

Overall, the closest method to ours is "Sisley the abstract painter" [ZZ10], which is a semi-automatic approach for highly stylized painterly filtering. The Sisley method involves segmenting the image and optionally assigning semantic labels and abstraction levels to nodes in a segmentation hierarchy; strokes are then distributed over the image plane so as to convey the image content with the appropriate abstraction level. Stroke colors and stroke geometry are perturbed, potentially dramatically, in order to achieve a high level of abstraction. This method produces quite convincing painterly images in an expressionist style. We seek a more representational style, possibly with exaggerated colors as Sisley used; our underlying process is also quite different. The Sisley method uses shape abstraction to simplify the segments, and then relies on independent perturbations of the stroke geometries to produce further distortion. Even at low levels of abstraction, details of the image are lost. Our approach separates the image warping from the painterly rendering; while we recommend applying a rendering technique such that the resulting image has a painterly appearance, most of the effect is due to the initial automatic warping and not to the painterly filter, which in any case need not be stroke-based.

Like many other techniques that perform image-space stylization, we depend on an initial segmentation of the image in order to preserve image features. Mean-shift segmentation is the most commonly employed technique. We use instead the oversegmentation provided by SLIC, the *simple linear iterative clustering* method proposed by Achanta et al. [ASS*12]. SLIC produces superpixels by k-means clustering in a space combining color distances and Euclidean distances; while the resulting segments have no semantic meaning, they tend to respect image edges and they tend to have similar sizes to nearby segments, a property that helps us distribute our distortion throughout the image, as we will see in the next sections. We accomplish image warping using a mass-spring system, akin to the "pelting" procedure of Piponi and Borshukov [PB00], who proposed mass-spring systems for texture coordinate assignment. While Piponi and Borshukov sought a smooth stretching of the textured surface, and hence used constant spring lengths, we deliberately stretch some portions more than others to obtain warping, using spatially varying spring lengths and spring constants to do so. The classic technique for image warping involves thin-plate splines [Bro89] but the smoothness of the splines creates a coherent distortion. We want our distortion to be coherent only insofar as it modifies a single object; the warp can be discontinuous across image edges. The mass-spring system, with spring parameters associated with SLIC regions, accomplishes this aim.

3. Algorithm

Figure 3 shows the pipeline of our method. First, we segment the input image into clusters using SLIC. The resulting clusters tend to be compact, uniform in size, and edge-sensitive. Second, we construct the mass-spring system by connecting springs between each pixel and its four-connected neighbours. All springs within a SLIC cluster receive a rest length according to a single randomly chosen value assigned to that cluster; the rest lengths range from approximately 0.1 to 2. Since the default spacing between adjacent pixels is 1, the clusters with x_0 smaller than 1 tend to shrink; and the clusters with x_0 greater than 1 tend to expand.

Our mass-spring simulation then iteratively computes the forces and updates the pixel locations. We halt the simulation

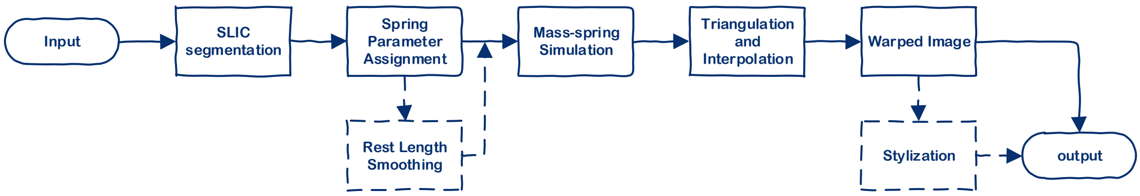

Figure 3: *Schematic of our processing pipeline. Dashed boxes represent optional steps.*

after a fixed number of iterations, and create a warped image by triangulating the new pixel locations and interpolating the color. Finally, we apply a painterly filter to the warped image; we used a morphological filter to achieve a watercolor appearance [BNTS07], but any desired filter could be used instead. The process is summarized in Figure 5.

Given an input image and the approximate superpixel diameter S, we measure the distance D between pixels and superpixel centers using equation 1:

$$D = d_{rgb} + (m/S)d_{xy} \tag{1}$$

where d_{rgb} denotes the Euclidean distance the RGB color cube, and d_{xy} is the Euclidean distance in the image plane. The parameter m lets us control the compactness of the segmentation. In this paper, we empirically use $m = 150$; note that RGB values lie in the range 0 to 255. Using this distance computation, we iteratively move all superpixel centers to the centroids of their regions until convergence.

We attach each pixel with springs to its four-connected neighbours. Assuming the spacing between adjacent pixels is 1 unit, we randomize the springs' rest lengths roughly in the range of 0.1 to 2 units. In order to exaggerate the warping effect, i.e., to exaggerate the difference between rest lengths, we want a distribution biased towards the minimum and maximum, with less likelihood of values between. All springs within the same superpixel get the same rest lengths, so that the whole region can expand or shrink. With all these considerations, we randomize the rest lengths x_0 as follows:

For all the springs s in one SLIC segment,

$$x_0(s) = x_{\min} + x_d + x_d \times (|r|^{\alpha})\,\mathrm{sgn}(r) \tag{2}$$

where x_{min} denotes the minimum rest length. Parameter x_d is the difference between the minimum rest length and the average rest length: i.e., the average length is $x_{\min} + x_d$. The maximum permitted rest length is $x_{\min} + 2 \times x_d$. Parameter r is a random number drawn from a uniform distribution from -1 to 1, and α is a real number in the range 0 to 1, controlling the distribution of spring lengths. Lower values of α make the spring lengths more likely to be close to minimum or maximum, with the extreme case of a binary distribution at $\alpha = 0$. A choice of $\alpha = 1$ gives a uniform distribution.

We experimented with different settings for α, but the difference is quite subtle, especially compared to other influences such as superpixel size and spring-strength constant. The results shown in this paper were obtained using $\alpha = 0.5$. We also explored different combinations for the minimum and maximum rest length; we settled on a range of $(0.1, 1.9)$ for the results that we show.

Each spring's spring constant k is scaled according to equation 3:

$$k(s) = \gamma \times |(x_0(s) - (x_{min} + x_d))| \tag{3}$$

where $|(x_0 - (x_{\min} + x_d))|$ deviation of the rest length of this spring from the average. Very long springs and very short springs have greater deviation. Thus, they are stronger: higher $k(s)$. The γ value scales the overall strength of all springs. Greater γ values produce a stronger warping effect. A γ of around 3 or 4 gives a reasonable amount of distortion; we show the effect of varying γ in the next section.

Optionally, we can smooth the spring parameters across regions: for applications where quality is paramount and extra processing time is less of a concern, we suggest smoothing the spring parameters by applying a cross-bilateral filter against the original image. This process causes similar regions to blend spring parameters together, yielding a less noticeable transition. Dissimilar neighboring regions do not change much. The effect is minor, but in our judgement produces a slight improvement. We applied the smoothing process to all the results shown in this paper.

Once spring parameters have been finalized, our mass-spring simulator iteratively calculates the forces exerted by all springs and moves the pixels accordingly. We halt the simulation at 50 iterations, a figure found to produce adequate convergence. Having completed the mass-spring simulation, we triangulate the new pixel layout and apply barycentric interpolation to determine pixel color, which gives us the warped image.

Even standing alone, many warped images are already quite interesting. Nonetheless, we suggest post-processing the warped images with painterly filters to accentuate the artistic appearance. Users can apply different filters, if desired; with our objective of maintaining some detail, we relied on a morphological watercolor style inspired by Bousseau et al. [BNTS07]. A visual summary of the approach is presented in Figure 5.

Figure 4: *Effect of the full process. Upper left: original image. Upper right: warping, no painterly effect. Lower left: painterly effect, no warping. Lower right: both warping and painterly effect.*

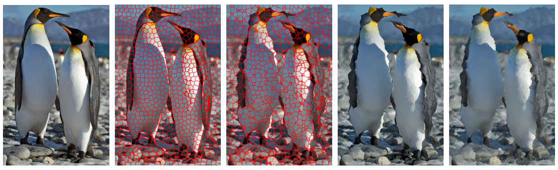

Figure 5: *Progression of an image through the pipeline. Left to right: original image; SLIC segments; warped SLIC segments; warped image; warping plus painterly effect.*

4. Results and Discussion

Figure 6 shows a few more sample results. Additional results appear on the first page. We demonstrate the effect on images with varied subject matter and backgrounds, including portraits, still lifes, animals, landscapes, and cityscapes. We spent some, but minimal, effort selecting good parameters for specific images; a wide range of parameters give substantially similar overall results for a particular photograph. Figure 6 also shows the source images.

These result images contain many elements of interest. Textures in the image are largely retained (e.g., books, secretarybird feathers), albeit somewhat abstracted by the painterly filter; in some cases, such as the concrete in the street image, the texture has been augmented by the warping. The distortion of facial features gives new expressions and provides a new interpretation of the image: we find the processed old man image particularly intriguing. The secretarybird now looks monstrous, with distortions in the beak now resembling teeth, and the distorted eye shape adding a further sense of menace. More subtle effects are also possible: the eyes of the boxer are slightly enlarged, giving the dog an even more plaintive expression. In general, there is a feeling of a hand-drawn style, with exaggerated shapes (e.g., lighthouse) and wavering lines (e.g., books), combined with a high degree of detail.

Other features are visible in the teaser image. The abstracted and distorted face of the man in the hood gives the image a wild look. The warped, fanciful architecture of the cathedral might serve to illustrate a book of fairy tales. The "garlic" still life has a scratchy look that looks somewhat careless and handmade.

Our non-optimized CPU implementation takes about 50 seconds for a half-megapixel image. This time is broken down into approximately 85% to compute the SLIC segmentation, 15% to do the mass-spring simulation, and negligible time to do the painterly filter (less than one second). We are confident that this time could be improved considerably with some effort; the SLIC calculations, in particular, can be sped up a great deal. These timing figures scale linearly with the number of pixels.

Figures 7 and 8 show the results of changing the algorithm's parameters. In Figure 7, the images with higher γ have stronger springs with more polarized rest lengths, producing more exaggerated distortion. The weak springs with $\gamma = 1$ produce an image with scarcely noticeable distortion. As γ increases, the distortion becomes more apparent. We prefer the milder distortions of approximately $\gamma = 2$, but even more extreme distortions are possible. Note that the exact nature of the distortion depends on the specific parameters assigned to specific regions, and because we assign parameters randomly, different outcomes are available by redoing an image.

In the examples in Figure 8, the SLIC region size ranges from 20 to 80. The smaller regions produce more high-frequency structure; with larger, fewer regions, the distortion occurs at a larger scale and larger objects can be coherently distorted. The "street" input image has structure on multiple scales, so which SLIC size is appropriate depends on the user's intent: all of these results are plausible. In images with a single important scale, such as portraits, it is more crucial to obtain the right size – see our failure examples below.

Figure 9 shows a comparison between our results with Sisley [ZZ10] under a low-abstraction configuration. As befitting its efforts to make a fairly abstract image, Sisley removes both shape and color detail; small-scale details are blurred out. With these low-abstraction settings, the colors are fairly close to those of the original photograph. There is a strong paint texture and visible paint strokes. In our result, the details are somewhat preserved, but are still somewhat modified, most notably on the face of the square building

Figure 6: *Above: several results of applying our method (old man, secretarybird, street, books, lighthouse, boxer). Below: original images (hood, cathedral, garlic, old man, boxer, lighthouse, street, books, secretarybird).*

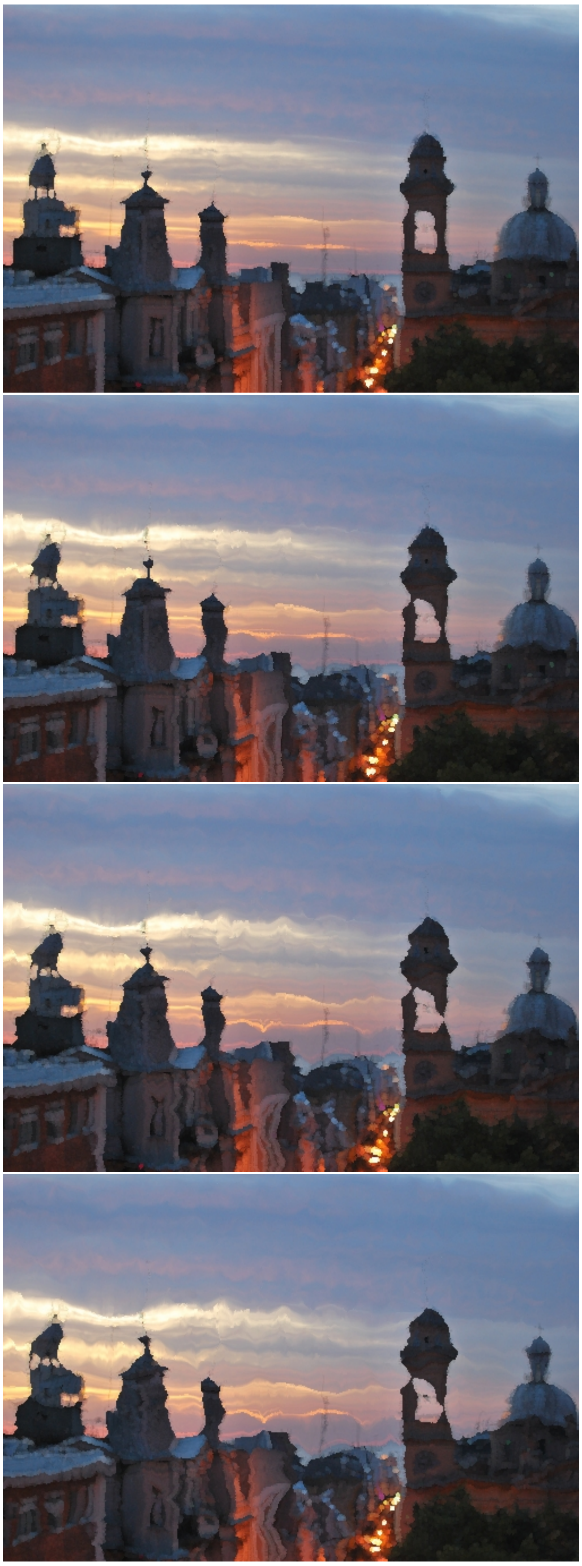

Figure 7: *Spring strength increases from top to bottom: spring parameters are* $\gamma = 1$, $\gamma = 1.5$, $\gamma = 2$, $\gamma = 3$.

Figure 8: *From top to bottom: SLIC regions of diameter 20, 40, 60, 80 pixels.*

just above the sailboat. There is some large-scale distortion: the shoreline is altered, and the right-hand sail of the sailboat is pushed out as if billowing in the wind. The aims of the two methods are quite different, but somewhat complementary, with Sisley aiming at greater degrees of abstraction and ours aiming for an impression of painterliness but retaining significant amounts of detail.

Figure 9: *Comparison of our approach with abstract rendering by Sisley. Above: Sisley. Below: ours.*

Figure 10 shows two failure cases. These images show the effects of the warping only, without painterly postprocessing. In the "crying" image, there is a mismatch between the size of the facial features and the relatively small SLIC regions. This mismatch, plus the use of very strong springs, induces a high-frequency texture over the face; the facial features become jagged as neighboring regions pull them in different directions. The "flamingo" image has quite strong springs as well, but a better match of SLIC region size to feature size. However, the blurred but varied background now has a cobblestone-like texture from the SLIC regions. Also, the flamingos' legs are particularly distorted: in general, when the image contains extended linear features, the distortion is particularly noticeable, and the effect may or may not be acceptable depending on the user's intention.

Figure 10: *Two failure cases. Above: crying. Below: flamingo.*

We have used this image to illustrate specific problems, but it has some visual interest anyway, and the birds' feathers are nicely conveyed. Overall, though, our process is not well suited to this type of image.

5. Conclusion

In this paper, we presented a two-stage methodology for creating painterly images from photographs: first, the image is distorted using a mass-spring system, and then a painterly filter is applied to the warped image. We recommend using a painterly filter that does not change the image very much. The warping process only changes the size and relative position of details, but does not by itself remove detail; by using a painterly filter only lightly, we can unify the image into a painterly style but still produce a detailed image. We hope that this paper spurs others to consider more representational automatic painterly rendering, as opposed to the more abstract and expressionist examples of the medium that have been popular in NPR.

Our warping was done using a mass-spring system, with spring coefficients and rest lengths chosen randomly for each region of a SLIC oversegmentation of the input photograph. The mass-spring distortion is fast and flexible, properties noted by Piponi and Borshukov [PB00]. Using SLIC gives us a general and robust segmentation, where the segments are all approximately the same size. Because the segment boundaries tend to lie on image edges, we preserve some structure of the input image through this process.

After warping, the image can be processed by any conventional image-space painterly rendering system. We used a morphological filtering method to produce a watercolor effect, following Bousseau et al. [BNTS07].

Our process is effective at generating painted-looking images. The two elements of the process cooperate to produce the illusion: the warping ensures that the image is not excessively faithful to the underlying photograph, while the painterly post-processing makes the image look painted and allows the naive viewer to attribute the distortion to the painting process itself. Sisley [ZZ10] is the process most similar to ours, doing both distortion and painterly rendering in a unified stroke-based rendering environment; its aim was to create highly abstracted images, whereas ours is to produce representational images that nonetheless are painterly and not excessively photographic. Collomosse and Hall [CH03] used separate warping and painterly filtering stages in their pipeline, with the aim of creating cubist paintings rather than more conventionally representational images.

Our method has some limitations. Its effectiveness depends somewhat on matching the scale of the segmentation to the scale of the objects within the image: the spatially-varying distortion of textures and very small objects may look incoherent. Because of the random assignment of spring parameters, executing the process multiple times over the same image produces different results, not all of which are equally appealing.

In future work, we are interested in further exploring a detailed painterly style that lacks visible brushstrokes. Replacing our content-agnostic distortions with shape simplifications and using more spatially coherent parameters might be helpful. Although in this work we concentrated on fully automatic processing, the system could benefit from a lightweight user-assisted labeling interface.

We want to continue the pursuit of realism in painting, as practiced by 19th-century Romantic painters, for example. One aspect of this is to include paint texture without necessarily indicating individual brush strokes. Alternatively, less-realistic paintings could be produced by fusing the results of multiple different distortions of a single input image. Lastly, adapting the technique to video could be interesting: in this case, we would want to attach the spring parameters to persistent objects in the video, possibly accomplished by performing a segmentation over the video cube.

Acknowledgements

Thanks to other GIGL members for productive comments and discussions. Funding for this work was provided by NSERC and by Carleton University.

We used many images from Flickr under a Creative Commons license. Thanks to the numerous photographers who provided material: spablab (hood), Doug Kerr (cathedral), Hafiz Issadeen (garlic), Pedro Ribeiro Simões (old lady), **penguin, Anne Worner (old man), Wayne Dumbleton (secretarybird), Yiannis Chatzitheodorou (street), Stefan Schmidt (books), InAweofGod'sCreation (lighthouse), MythicSeabass (boxer), Ana Raquel S. Hernandes (towers), sean dreilinger (crying), **flamingo.

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (Nov. 2012), 2274–2282. URL: http://dx.doi.org/10.1109/TPAMI.2012.120, doi:10.1109/TPAMI.2012.120. 3

[BKTS06] BOUSSEAU A., KAPLAN M., THOLLOT J., SILLION F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2006), NPAR '06, ACM, pp. 141–149. URL: http://doi.acm.org/10.1145/1124728.1124751, doi:10.1145/1124728.1124751. 1

[BNTS07] BOUSSEAU A., NEYRET F., THOLLOT J., SALESIN D.: Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics 26*, 3 (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276507, doi:10.1145/1276377.1276507. 2, 4, 5, 9

[Bro89] BROOKSTEIN F.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 6 (June 1989), 567–585. 3

[BSCS07] BROSZ J., SAMAVATI F. F., CARPENDALE M. S. T., SOUSA M. C.: Single camera flexible projection. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2007), NPAR '07, ACM, pp. 33–42. URL: http://doi.acm.org/10.1145/1274871.1274876, doi:10.1145/1274871.1274876. 3

[BWL04] BAXTER W., WENDT J., LIN M. C.: IMPaSTo: A realistic, interactive model for paint. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2004), NPAR '04, ACM, pp. 45–148. URL: http://doi.acm.org/10.1145/987657.987665, doi:10.1145/987657.987665. 1, 2

[CH03] COLLOMOSSE J. P., HALL P. M.: Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics 9*, 4 (Oct. 2003), 443–453. URL: http://dx.doi.org/10.1109/TVCG.2003.1260739, doi:10.1109/TVCG.2003.1260739. 3, 9

[CS04] COLEMAN P., SINGH K.: Ryan: Rendering your animation nonlinearly projected. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2004), NPAR '04, ACM, pp. 129–156. URL: http://doi.acm.org/10.1145/987657.987678, doi:10.1145/987657.987678. 3

[CT05] Chu N. S.-H., Tai C.-L.: MoXi: Real-time ink dispersion in absorbent paper. *ACM Transactions on Graphics 24*, 3 (July 2005), 504–511. URL: http://doi.acm.org/10.1145/1073204.1073221, doi:10.1145/1073204.1073221. 2

[DS02] DeCarlo D., Santella A.: Stylization and abstraction of photographs. *ACM Transactions on Graphics 21*, 3 (July 2002), 769–776. URL: http://doi.acm.org/10.1145/566654.566650, doi:10.1145/566654.566650. 3

[GRG04] Gooch B., Reinhard E., Gooch A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics 23*, 1 (Jan. 2004), 27–44. URL: http://doi.acm.org/10.1145/966131.966133, doi:10.1145/966131.966133. 2

[Hae90] Haeberli P.: Paint by numbers: Abstract image representations. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1990), SIGGRAPH '90, ACM, pp. 207–214. URL: http://doi.acm.org/10.1145/97879.97902, doi:10.1145/97879.97902. 1

[Her98] Hertzmann A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 453–460. URL: http://doi.acm.org/10.1145/280814.280951, doi:10.1145/280814.280951. 1, 2

[Mei96] Meier B. J.: Painterly rendering for animation. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 477–484. URL: http://doi.acm.org/10.1145/237170.237288, doi:10.1145/237170.237288. 1, 2

[Mou03] Mould D.: A stained glass image filter. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), EGRW '03, Eurographics Association, pp. 20–25. URL: http://dl.acm.org/citation.cfm?id=882404.882407. 3

[PB00] Piponi D., Borshukov G.: Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 471–478. URL: http://dx.doi.org/10.1145/344779.344990, doi:10.1145/344779.344990. 3, 9

[Sin02] Singh K.: A fresh perspective. In *Proceedings of Graphics Interface 2002* (May 2002), CHCCS, pp. 17–24. 3

[SRHC08] Song Y.-Z., Rosin P. L., Hall P. M., Collomosse J.: Arty shapes. In *Proceedings of the Fourth Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging* (Aire-la-Ville, Switzerland, Switzerland, 2008), Computational Aesthetics'08, Eurographics Association, pp. 65–72. URL: http://dx.doi.org/10.2312/COMPAESTH/COMPAESTH08/065-072, doi:10.2312/COMPAESTH/COMPAESTH08/065-072. 3

[WLL*06] Wen F., Luan Q., Liang L., Xu Y.-Q., Shum H.-Y.: Color sketch generation. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2006), NPAR '06, ACM, pp. 47–54. URL: http://doi.acm.org/10.1145/1124728.1124737, doi:10.1145/1124728.1124737. 3

[ZZ10] Zhao M., Zhu S.-C.: Sisley the abstract painter. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (New York, NY, USA, 2010), NPAR '10, ACM, pp. 99–107. URL: http://doi.acm.org/10.1145/1809939.1809951, doi:10.1145/1809939.1809951. 2, 3, 5, 9

[ZZ11] Zhao M., Zhu S.-C.: Portrait painting using active templates. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering* (New York, NY, USA, 2011), NPAR '11, ACM, pp. 117–124. URL: http://doi.acm.org/10.1145/2024676.2024696, doi:10.1145/2024676.2024696. 2