COLORBRUSH: ANIMATED DIFFUSION FOR INTUITIVE COLORIZATION SIMULATING WATER PAINTING

Nicolas Märki[2]*, Oliver Wang*[1]*, Markus Gross,*[1,2]*, and Aljoša Smolić*[1]

[1]Disney Research Zurich [2]ETH Zurich

ABSTRACT

Water painting is an art, where the result and experience strongly depend on the process of creating it. Color is injected by the artist in a controlled way and diffuses into a final state. We present a system to simulate such a process, which builds on existing colorization approaches. The geodesic distance builds the mathematical foundation for our animated diffusion. We add a time-dependent weight function to generate a diffusion-like spreading effect. Tiled approximation is used to achieve interactive rates on modern mobile devices. The result is a colorization framework simulating water painting that allows giving real-time feedback on touch events with the limited hardware resources of a tablet computer.

Index Terms— Colorization, diffusion, edge-aware filtering, domain transform, interactive imaging

1. INTRODUCTION

Humans have created images for artistic expression since the very beginning of history. Painting forms part of our basic culture, history and heritage from cave paintings to digital art and photography. Technology is evolving; allowing us to create images using mice, touch screens, etc. as input devices.

Besides the invention of completely new tools, technology also tries to copy or simulate traditional ways of painting and imaging. Often those traditional techniques give a certain feel and experience of the process of creating the paintings, which form an integral part of the art. One example of such a creative process is water painting [1],[2]. As the artist injects paint onto the canvas, it diffuses in a very characteristic way into a final image. The diffusion process as well as the final result are both important for the look and experience of water painting.

Relevant parameters include, among other things, the amount and type of paint and water, the structure/texture of the canvas, and lines/structures/edges, which the artist uses on canvas to control/guide the diffusion.

In this paper, we present a system for intuitive simulation of water painting as illustrated in Fig. 1. The user injects a color selected from a palette as a stroke via touch screen or mouse. The way the color spreads into the steady state is controlled by edge-aware filtering. Also the dynamics of the diffusion process are integrated into the filter. These technical details of our approach are discussed in Section 3. Then we present results and examples in Section 4, before concluding and discussing future work in Section 5. Example videos illustrating our algorithm can be found online [3].

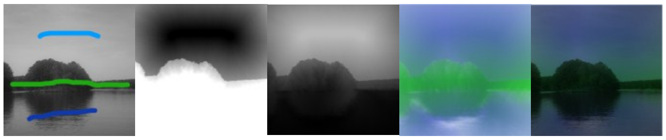


Fig. 1: Pipeline of our approach for animated diffusion simulating water painting.

2. RELATED WORK

The first computer-aided colorization process was introduced by Wilson Markle in 1970 and used to add color to video footage of the moon from the Apollo mission [4]. It required a color mask which was drawn manually for one frame of each scene and then transformed to match the other frames by motion detection [5].

Manual masking an image is not a reasonable interaction model for a touchscreen based system. One method that reduced the required accuracy of input was presented by Levin et al. [6], formalizing colorization as a quadratic optimization problem in YUV space. They introduce the assumption that neighboring pixels with similar intensities should have similar colors and include the user input as additional constraints. Their results prove that only a few color scribbles are usually enough to produce surprisingly good results. Nie et al. [7] improve the algorithm by introducing a new local connectivity factor that requires pixels to be similar in connected regions based on a modified Canny edge detector.

While such optimization based solutions can yield high quality solutions, solving for the minimal error is computational intensive. Horiuchi and Kotera [8] introduce a lightweight method that calculates the weighted average of the chroma channels in CIALAB color space. Weights decrease both with Euclidean distance and intensity difference to seed pixels. Yatziv and Sapiro [9] also calculate the weighted average of chroma channels in CIELAB, but use the geodesic distance - integrals over the image gradient along paths from seeds to each pixel - to determine the weight. Compared to the results of Horiuchi and Kotera [8], results are much better around hard edges in the image.

The calculation of the exact geodesic distance is rather expensive, but a good approximation can be calculated in linear time for the complete image following Criminisi et al. [10], who present a general image editing framework that propagates sparse user input based on the geodesic distance. Sagiv et al. [11] use the geodesic distance in Gabor space to improve results with textured images. Lang et al. [12] extend the principle into the temporal dimension for efficient processing of video.

Microsoft recently released a showcase app for their new Windows 8 tablets which has an even more realistic painting simulation. Fresh Paint [13] uses an oil paint simulation by Chu et al. [14]. By simulating the color transfer process between canvas and brush, they achieve a very realistic oil painting experience

3. METHOD

Our method defines an intuitive simulation of water painting, which includes the final result of the color diffusion as well as the look and feel of the color diffusion process over time. As such our colorization approach is not fully automatic, but controllable by the user without noticeable limitation of the creative freedom. Inspired by real world water painting and the colorization approach of Levin et al. [6], interaction should be based on simple color strokes, painted on the gray-scale image. Rough painting over the desired region should be enough to fill out detailed structures with little effort.

The geodesic distance (3.1) is an appropriate mathematical model for such a diffusion process. Animated diffusion is controlled by a weight function (3.2) with steady state and temporal components and parameters. Real-time performance is achieved by tiled approximation (3.3). Finally, the output image is computed by blending and overlay (3.4).

3.1 Geodesic distance

The geodesic distance is used as the basic function to control edge-aware diffusion of injected color through the image. Besides the geometric distance also image properties like edges and textures contribute to its value.

Let $I(x) : \Omega \rightarrow \mathbb{R}$ be a representation of our input image (grayscale or line art), and $C_{\boldsymbol{u},\boldsymbol{v}}(s) : [0,1] \rightarrow \Omega$ an arbitrary curve with $C_{\boldsymbol{u},\boldsymbol{v}}(0) = \boldsymbol{u}$ and $C_{\boldsymbol{u},\boldsymbol{v}}(1) = \boldsymbol{v}$. The geodesic distance between two points is then defined as the length of the shortest such curve along the gradient of the image,

$$d(\boldsymbol{u},\boldsymbol{v}) := \min_{C_{\boldsymbol{u},\boldsymbol{v}}} \int_0^1 \left|\nabla I \cdot C(\dot{s})\right| ds.$$

These properties of the geodesic distance are illustrated in Fig. 2. Image features like edges and textures create gradients that make the corresponding paths through the image more costly. A geometrically longer part through an unstructured region may well be less costly, i.e. shorter in the geodesic sense. This property is well suited to simulate natural diffusion around "obstacles" in the image.

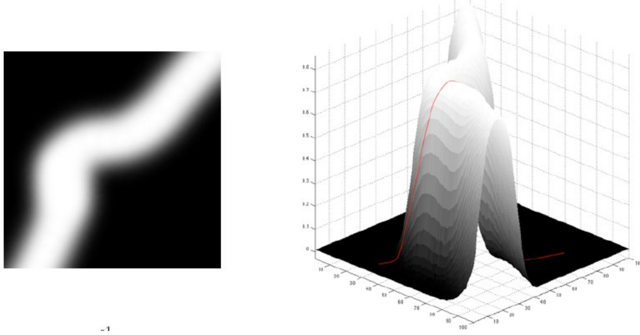


Fig. 2: Geodesic distance (right) is defined by the distance along a manifold described by the image (left).

Given an input stroke from the user represented as a set of pixels $S \subset \Omega$, the geodesic distance transformation defines the shortest distance from a certain pixel in the image to any pixel of the stroke,

$$D(\boldsymbol{x}) := \min_{\boldsymbol{v} \in S} d(\boldsymbol{x},\boldsymbol{v}).$$

Intuitively, the further away a pixel is in the geodesic sense, the less color from the stroke should reach it. This is realized by our formulation. In practice, the geodesic distance transform could be calculated exactly with algorithms like Dijkstra. But since we need real-time updates of the geodesic distance transform for the complete image, we use a raster scan approximation, following Criminisi et al. [10].

The geodesic distance can be computed from image features, but also other types of input may be used to influence the color diffusion process. As illustrated in Fig. 3 an artist may provide additional edges (middle, the outline) via another painting tool, which then influence the way the color spreads (see also video [3]). These additional edges/outlines may be made visible or invisible in the final image. Also canvas properties like roughness and granularity can easily be simulated by defining corresponding textures, which will create gradients that influence diffusion accordingly. Fig. 3 shows on the right the result of the geodesic distance per pixel for the input stroke and image on the left, with the outline in the middle. Distribution of color is controlled by all features. It may also bleed over weak edges as in natural water painting.

Fig. 3: Geodesic distance given a prescribed artist outline.

3.2 Weight function

Instead of directly using the geodesic distance, we introduce a weight function to control the color mixing process. It provides useful parameters to adjust the steady state of color diffusion as well as the temporal behavior of the diffusion process. This allows us to carefully design animated diffusion. Our choice of the weight function is the product of two sigmoid functions:

$$w(\boldsymbol{x},t)=\frac{1}{1+e^{\alpha(D(x)-\beta)}}\cdot\frac{1}{1+e^{\gamma(D(x)-\delta t)}}.$$

As illustrated in Fig. 4, weight decreases with the geodesic distance towards zero. This behavior is controlled by the first term in the equation and defines the steady state of the diffusion process for large time *t*. Fig. 5 visualizes the steady state weight function and geodesic distance for the image from Fig. 1.

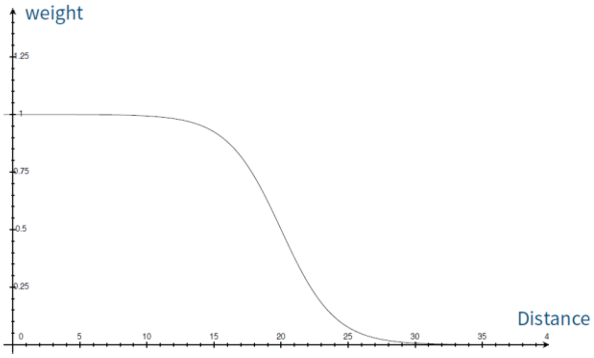


Fig. 4: Weight function as distance decreases.

The temporal behavior of the diffusion process is controlled by the second term of the weight function, as illustrated in Fig. 6. With increasing time this term converges to 1, i.e. it can be used to create an animated diffusion over time, of how more and more color reaches a pixel up to the steady state.

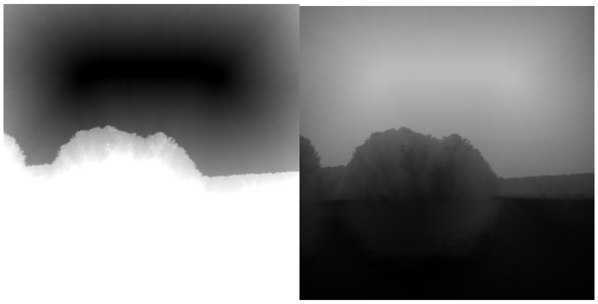


Fig. 5: Geodesic distance to weight visualization.

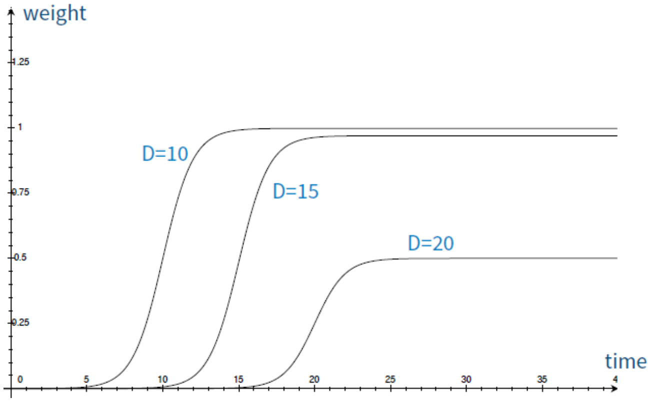
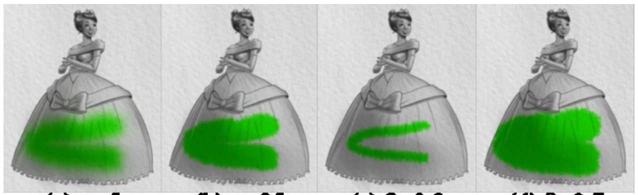


Fig. 6: Temporal behavior of weight function for animated diffusion.

With that, the weight function gives us 4 parameters to adjust the animated diffusion process properly. The softness parameter α controls the leakage of the color, which can be made softer (Fig. 7 (a)) or more edgy (Fig. 7 (b)). A higher value of the spread parameter β increases the spread of a stroke (Fig. 7 (c) vs. (d)). The speed parameter δ controls the velocity of the diffusion. High values result in a faster animation. Finally, the softness of the spreading color front can be adapted with parameter γ.



(a) α=5 **(b) α=25** **(c) β=0.2** **(d) β=0.7**

Fig. 7: Effects of softness parameter *α* spread parameter *β*.

3.3 Tiled geodesic distance transform

In order to create a truly intuitive simulation of the water painting experience, the algorithm must achieve real-time performance. To handle (almost) arbitrary sized images in real-time, we use the following observation: For the first few frames after the creation of a new stroke, the color can only reach pixels with a small geodesic distance due to the animated diffusion effect. Since the geodesic distance can never be smaller than the Euclidean distance, we know that the affected pixels are always in a bounded region around the stroke. Therefore, we only need to update the geodesic distance map for a small region around the stroke and can postpone the rest of the calculations to later frames.

This is realized as tiled approximation as illustrated in Fig. 8. For the first frame only a small portion of the image is evaluated, i.e. a single tile. For frame 10 a larger portion of tiles is affected. At frame 25 already the whole image is processed. Note that tiles are only processed until the weight function approximately reaches the steady state, i.e. inner tiles terminate processing earlier than outer tiles.

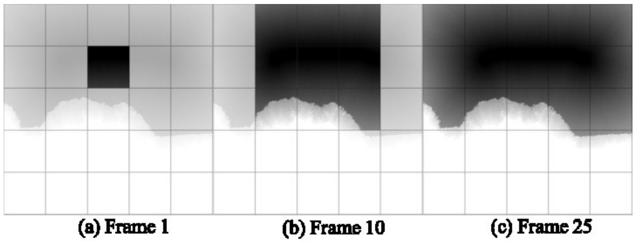

(a) Frame 1 (b) Frame 10 (c) Frame 25

Fig. 8: Tiled approximation for real-time performance.

3.4 Blending and overlay

Animated diffusion as discussed so far defines the color contribution from a certain user stroke to any pixel of the image controlled by image structures (visible/invisible edges, textures). If more than one color stroke is used, appropriate color mixing has to be performed by pixel. The resulting color has to be combined with the underlying greyscale background. We use a standard subtractive/additive color mixing model and overlay in YUV space to retain perceived brightness.

4. RESULTS

Our system allows for intuitive colorization of greyscale or line art images simulating the look and feel of water painting. Fig. 9 shows some results, all generated with only a couple of strokes. High resolution versions of these images can be found online [3]. We also provide demo videos that illustrate the intuitive look and feel of our water painting application [3].

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented an intuitive colorization framework simulating water painting. Animated diffusion is used to approximate the natural look and feel of the experience. The geodesic distance calculated over image gradients forms the basis of the approach. Edges and textures of the input image as well as artist input like visible and invisible lines control diffusion of the injected color. A weight function is introduced which allows proper design of the steady state as well as the temporal behavior of the diffusion process. Computation is performed over tiles to achieve interactive processing rates. Implemented as iPad app our system provides a creative tool for colorization of grey scale images and line art as shown in our results. Most testers were excited by the smooth drawing experience which is much more enjoyable than other iPad drawing apps.

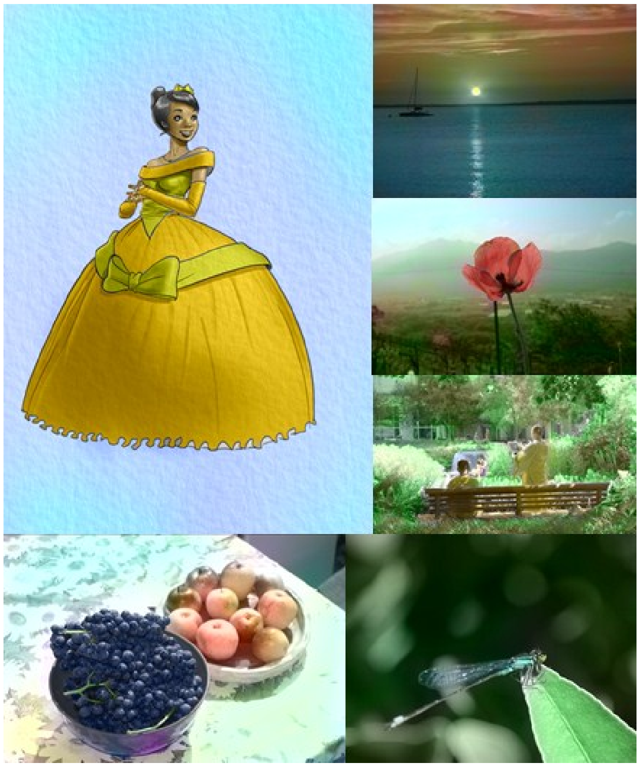


Fig. 9: A few sample images created using our method. Each example here was composed with only a couple of strokes.

Our future work will include improvement and optimization of any of the components. A more sophisticated and natural modeling of the animated diffusion process could be one direction. For instance, Sagiv et al. [11] describe the use of the geodesic distance in Gabor space. Another direction could be incorporation of a more sophisticated color mixing model [15]. Finally, we plan to extend our system to colorization of video following ideas on spatio-temporal edge-aware filtering from [12].

6. REFERENCES

[1] Wikipedia, Watercolor painting, last checked: 14.12.2014 URL http://en.wikipedia.org/wiki/Watercolor_painting.

[2] Beautiful Water Painting, last checked: 14.12.2014 URL http://www.youtube.com/watch?v=LNFdqCVi_Gc.

[3] ColorBrush, last checked: 14.12.2014 URL http://zurich.disneyresearch.com/~smolica/ColorBrush.htm.

[4] A. Reifsnyder, “Coloring Old Movies: Foes See Red, Backers See Green,” *Chicago Tribune*, 1986.

[5] W. Markle, and B. Hunt, “Coloring a black and white signal using motion detection,” *US Patent Office*, December 1986.

[6] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[7] D. Nie, L. Ma, S. Xiao, and X.Z. Xiao, “Grey-scale image colorization by local correlation based optimization algorithm,” *VISUAL’05: 8th International Conference on Visual Information and Information Systems, Springer-Verlag*, July 2005.

[8] T. Horiuchi and H. Kotera, “Colorization for Monochrome Image with Texture,” *Color Imaging Conference,* pp. 245–250, 2005

[9] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *Image Processing, IEEE Transactions on*, 15(5):1120–1129, 2006.

[10] A. Criminisi, T. Sharp, C. Rother, and P. Perez, “Geodesic image and video editing,” *ACM Transactions on Graphics (TOG)*, 29(5), October 2010.

[11] C. Sagiv, N.A. Sochen, and Y.Y. Zeevi, “Gabor-space geodesic active contours,” *In Algebraic Frames for the Perception-Action Cycle, G. Sommer and Y.Y. Zeevi (Eds.), Springer-Verlag,* pages 309–318, 2000.

[12] M. Lang, O. Wang, T. Aydin, A. Smolic and M. Gross, “Practical temporal consistency for image-based graphics applications,” *ACM Transactions on Graphics (SIGGRAPH)*, July 2012.

[13] Microsoft, Fresh Paint, last checked: 14.12.2014. URL http://apps.microsoft.com/windows/en-us/app/fresh-paint/1926e0a0-5e41-48e1-ba68-be35f2266a03.

[14] N. Chu, W. Baxter, L.-Y. Wei, and N. Govindaraju, “Detail-preserving paint modeling for 3D brushes,” *NPAR’10, 8th International Symposium on Non-Photorealistic Animation and Rendering,* June 2010.

[15] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, and D.H. Salesin, “Computer-Generated Watercolor”, *ACM Transactions on Graphics (SIGGRAPH)*, August 1997.