CZECH TECHNICAL UNIVERSITY IN PRAGUE

Doctoral Thesis Statement

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Ing. Michal Lukáč

Example-based Creation of Digital Imagery

A doctoral thesis statement submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy.

Ph.D. programme: Electrical Engineering and Information Technology
Branch of study: Information Science and Computer Engineering

Prague, November 2015

The doctoral thesis was produced in combined manner Ph.D. study at the Department of Computer Graphics and Interaction of the Faculty of Electrical Engineering of the CTU in Prague.

Candidate:

Ing. Michal Lukáč
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering of Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, CZ

Thesis Supervisor:

prof. Ing. Jiří Žára, CSc.
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering of Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, CZ

Opponents:

....

....

....

The doctoral thesis statement was distributed on: ....

The defence of the doctoral thesis will be held on .... at .... a.m./p.m. before the Board for the Defence of the Doctoral Thesis in the branch of study (to be specified) in the meeting room No. .... of the Faculty of Electrical Engineering of the CTU in Prague.

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

....
Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study Information Science and Computer Engineering
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6.

Contents

1 Introduction

Digital image creation, or less formally digital art, is a computerized counterpart of the physical art of painting. From the research side, it is characterised by a continuous effort to develop new tools which let users create images in ever-higher fidelity while "intelligently" considering the image content.

With the increases of computational power and the volume of image data being processed over the past 30 years, the requirements on fidelity and quality of the result only grew. For instance, selecting objects for editing in images with a resolution of a megapixel or beyond using simple geometric selection was no longer feasible, and so automated selection techniques were developed [Rother et al., 2004]. Similarly, flat-colour, fixed-footprint brushes were no longer acceptable as tools for high-quality artwork, necessitating the development of more sophisticated tools.

Such tools generally fall under the heading of *procedural tools*; that is, brushes, filters, etc., which have been hand-crafted to achieve a certain artistic effect, be it an imitation of a physical art style or an entirely novel one (see Figure 1.1 for an example). Commercial programs such as *Corel Painter* take advantage of these tools for the purposes of digital painting. Similarly, algorithms have been devised to generate textures, usually by the expedient of applying post-processing to a noise function [Ebert et al., 2002]. Simultaneously, simulation-based approaches are being developed. In these, physical behaviour of various painting media is simulated to achieve maximum veracity, as shown in Figure 1.2.

Both the procedural and the simulation-based approaches do, however, suffer from an important drawback – each of the tools can only generate one particular style, and new ones have to be carefully designed by programmers who simultaneously have to have a degree of domain expertise. This limits the palette of available tools and styles, and thus the breadth of use of these approaches.

In response to this, *example-based* approaches (Figure 1.3) were conceived. These approaches aim to use a single algorithm to mimic an unlimited number of artistic styles by the expedient of analysing an example of an arbitrary style and then attempting to replicate it. It is exactly these methods which are the focus of the thesis.

1.1 Example-based Methods

The label "example-based" has been applied to a multitude of methods intended for various applications. What they have in common is that the parameters used for the underlying model for generation of visual content are neither provided by the user, nor determined by simulation, but instead inferred by the analysis of a user-provided exemplar. This makes them significantly more versatile with respect to the visual characteristics of the output, but also presents a challenging research problem; the methods need to be sufficiently flexible to account for various visual styles but not have so many free parameters that learning them from available input data is infeasible. At the same time, the computational expense of analysing the input exemplar

Figure 1.1: *An example of artwork generated by various brushes of a procedural system [Měch and Miller, 2012]*

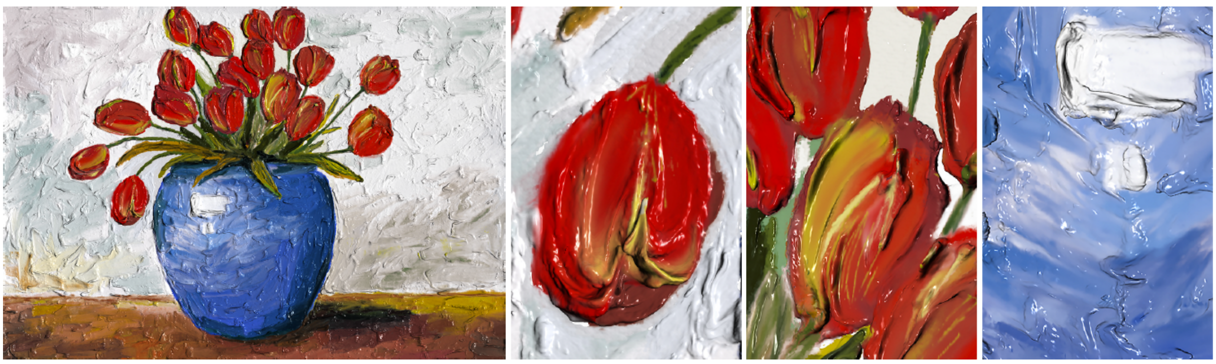


Figure 1.2: *Artwork generated by a simulation-based painting system. In this system, behaviour of brush bristles is simulated in order to realistically determine pigment deposition patterns. [Chen et al., 2015]*

Figure 1.3: *Realbrush by Lu et al. [2013] is an example-based approach for digital painting. The brush stroke examples on the left, painted on paper and captured with a digital camera, are analysed and then used in stroke-wise synthesis to let artist create the image on the right.*

needs to be considered. Nonetheless, the attractiveness of their versatility is sufficient to make them an active research topic in many areas of computer graphics.

There are methods for example-based super-resolution [Freeman et al., 2002], colourisation [Bonneel et al., 2013], image filtering [Liu et al., 2014], portrait stylization [Shih et al., 2014], or paint compositing [Lu et al., 2014b]. Of particular interest for us are methods for *Example-based Texture Synthesis*, due to the wide variety of applications they are the basis for.

These methods were initially designed to generate infinite textures [Paget and Longstaff, 1998; Lefebvre and Hoppe, 2005]. Later, new methods sharing their theoretical basis have been developed to fill in holes in images [Efros and Leung, 1999; Criminisi et al., 2004; Wexler et al., 2007], create new structured images [Risser et al., 2010], or alter the artistic style of existing images [Hertzmann et al., 2001]. In the remainder of this section, we describe the two basic classes of example-based texture synthesis methods – *Region-growing Approaches* and *Optimization-based Approaches* – grouped according to their basic functional principles. The following section describes how the basic texture synthesis methods can be put to use in more complex applications.

1.1.1 Region-growing Approaches

Region-growing approaches operate by first dividing the image into a known region with assigned values and an undefined region. Gradually, new values are assigned to pixels in the undefined region (as shown by Figure 1.4), either pixel-by-pixel [Efros and Leung, 1999] or in larger coherent parts [Efros and Freeman, 2001]. When selecting an exemplar fragment to copy into the undefined region, values of adjacent pixels are compared to corresponding values in the assigned region adjacent to the area to be filled in order to make the output visually similar to the exemplar.

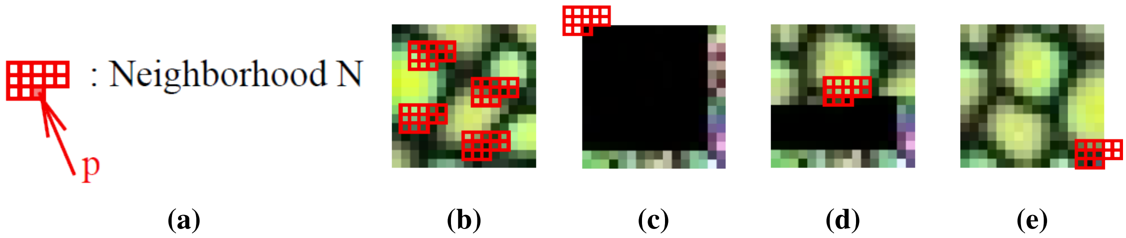


Figure 1.4: *An illustration of a region-growing synthesis process according to Wei and Levoy [2000].* (a) *A pixel neighbourhood considered in filling the pixel p* (b) *Various such neighbourhoods in the source are considered according to their similarity with the neigbhourhood of the output pixel being filled* (c) *The top left pixel is being filled. Note that the neighbourhood is considered under toroidal geometry, and the final two lines and columns of the output have been randomly initialized as a seed* (d) *Partially synthesized output. Note the shrinking of the black undefined region* (e) *Fully synthesized output.*

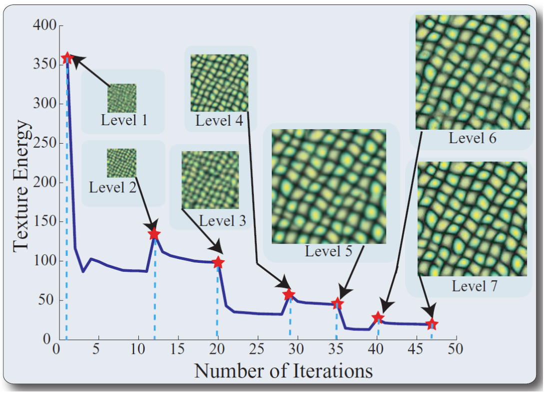
Approaches vary by the exact mechanics of the selection [Wei and Levoy, 2003], the order in which new fragments are added [Criminisi et al., 2004] as well as the size of the fragments and the mechanics used to splice them to existing content [Kwatra et al., 2003].

The advantage of these approaches is that they may be seamlessly adapted for *de novo* texture synthesis by starting with a random seed fragment, or used for hole-filling directly. They have also been adapted for example-based image stylization [Hertzmann et al., 2001], as well as interactive texture painting [Ritter et al., 2006].

The disadvantages are their sensitivity to filling order (as discussed by Criminisi et al. [2004]), and a related issue with synthesizing large-scale structures. The limited size of the area considered before splicing and the fact that all decisions are made locally lead to artefacts where pieces of texture grown from different directions meet. Some approaches would run additional passes of synthesis after the initial one to correct for this (yielding the results shown in Figure 1.6). Later variants have attempted to use coarse-to-fine synthesis to alleviate this problem [Wei and Levoy, 2000], but doing so also introduces additional implementation issues as the information from the less-reliable previous synthesis steps needs to be considered differently from the comparatively more reliable information in the constraints.

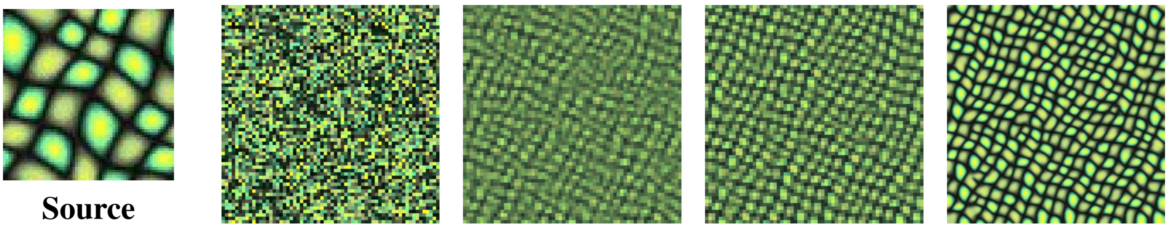
1.1.2 Optimization-based Approaches

In optimization-based approaches, an optimization criterion is first defined, with the assumption that optimizing for this criterion yields an output more similar to the exemplar (as in Figure 1.5d). Usually the entire output is initialized to random values. The optimization criterion is then evaluated on a single pixel and its neighbourhood, or on the entire image (depending on whether the optimization step is pixel-wise or global), and an optimization or sampling technique is applied to improve its value. The specifics in which these approaches vary are principally the formulation of the optimization criterion, and the optimization process used to improve it.



$$E = (x; \{z_p\}) = \sum_{p \in X^{\dagger}} ||x_p - z_p||^2$$

(a) **(b)**



(c)

(d) *Energy optimization texture synthesis by Kwatra et al. [2005].* (a) *shows the progress of the optimization process over iterations, with the stars marking points at which the optimization advanced to the next level of the image pyramid.* (b) *The pixel-wise energy being optimized. It is defined for a pixel x and its most similar counterpart in the source* z_p*, and measures the sum of squared differences over their respective neighbourhoods. Total energy is expressed as the sum over all pixels x. In alternating steps, values of x are altered and new nearest counterparts are found.* (c) *shows the exemplar next to the output of the algorithm; shown are the initialization, two intermediate results and the final result (not to scale).*

The original formulation was based on Markov Random Fields and represented the texture as such [Paget and Longstaff, 1998]. The fundamental assumption was that if the output is drawn from the same distribution as is determined by the exemplar, the textures will be similar. The texture was initialised by randomly sampling pixels in the input image and then pixel-wise Gibbs sampling was performed to get the final result. The local conditional probability distribution function was assumed to be homogeneous and was derived from an analysis of all neighbourhoods in the exemplar, effectively using pixel neighbourhood patches to retrieve similar pixels.

This yields a randomized approach that synthesizes textures satisfying this Markovian similarity criterion. As with the region-growing approaches, the size of the considered patch was shown to limit the size of the features that the method is able to consider, but in this case, a pyramid-based coarse-to-fine approach can be designed to be principled; starting with a coarse version of the output, the MRF is sampled until the burn-in period is considered to have finished, whereupon the result is upsampled and used as the initialisation of the next level.

Because the previously described method was computationally intensive, requiring a similarity search for each pixel update, and it was difficult to judge when the sampling converged, Lefebvre and Hoppe [2005] designed an alternate, coordinate-space method based on locally maximising the similarity in each pixel. The advantage of operating on coordinate space was that similarity could be pre-computed, and after each upsampling operation, a new output value would be selected from a set of precomputed similar alternatives to maximize local coherence. In this approach, random perturbation was used to introduce variation. This method, further developed to operate on appearance space [Lefebvre and Hoppe, 2006], remains the basis for the state of art in infinite stationary texture synthesis.

An explicit formulation based on global energy optimization was first developed in the context of video inpainting [Wexler et al., 2007] and later independently applied to infinite texture synthesis [Kwatra et al., 2005]. Both of these approaches define a global energy based on the sum of appearance-based dissimilarities over the set of all patches in the output region. This energy is then minimized using Expectation-Maximization, which has been shown to lead to a reasonable global minimum.

Like the previous group, these algorithms are generally run on a pyramid in a course-to-fine fashion, advancing to the next level on convergence. This requires no situational adaptations to the algorithm, as the objective function can be defined in exactly the same way for all levels.

Initially, these approaches were impractical because of the need to compute a dense nearest-neighbour field. However, after the development of PatchMatch [Barnes et al., 2009] global optimization at interactive rates became feasible. In addition, the generality of an energy-based approach allowed for adaptation to a variety of situations. Energy formulation has been extended to be bi-directional and used for image and video summarisation [Simakov et al., 2008], generalized to arbitrary dimension and used for synthesis of volume textures [Kopf et al., 2007] or adapted for multiple exemplars to facilitate a type of texture interpolation [Darabi et al., 2012]. An example of the latter application is shown in Figure 1.7.

Figure 1.6: *Selected results from GraphCut Textures [Kwatra et al., 2003], a region-growing synthesis approach.*

Figure 1.7: *A result of Image Melding Darabi et al. [2012], an optimization-based approach, being used to interpolate between two different textures.*

Figure 1.8: *An example of structure-preserving variation synthesis, as realised in* Synthesizing Structured Image Hybrids *[Risser et al., 2010].*

1.2 Applications in Image Creation

Image creation is a more difficult problem than texture synthesis, as images, unlike textures, have semantically meaningful high-level structure that needs to be imposed or preserved. Automatically synthesising this high-level structure is out of reach of current technology, nor is it ultimately desirable, as the users typically want to be able to define the semantics of the image.

Therefore, there are multiple possible approaches to creating a digital image while preserving visual style of an example. Generally, an example-based texture synthesis approach will be used to provide visual detail, with some sort of guidance integrated into it to ensure that the right visual elements are transferred to the appropriate portions of the image. According to how the guidance interacts with the synthesis, we may divide the available approaches into several categories:

Structure-preserving Variation Synthesis creates new images from a set of examples, with a general underlying assumption that the high-level structure is identical and images only vary in details (see Figure 1.8). The intended purpose is to expand a set of images with freshly synthesized ones in such a fashion that no detectable repetition occurs. The high-level structure may be inferred from the images being mutually spatially aligned [Risser et al., 2010], or a registration algorithm may be used to align input images in specialised cases [Assa and Cohen-Or, 2012].

Example-based Painting has the user define the high-level structure manually in its entirety. Guidance determined from user interaction (ie. brush strokes, vector primitives, etc.) is then imbued with a particular visual style similarly as if they had a fill colour assigned to them (see Figure 1.9). Such approaches may be based directly off of a texture synthesis approach by manipulating its initialization [Ashikhmin, 2001], they may use texture synthesis to interactively add texture to a user-created segmentation map [Ritter et al., 2006], or they can independently synthesize individual strokes based on a database of example strokes [Lu et al., 2013].

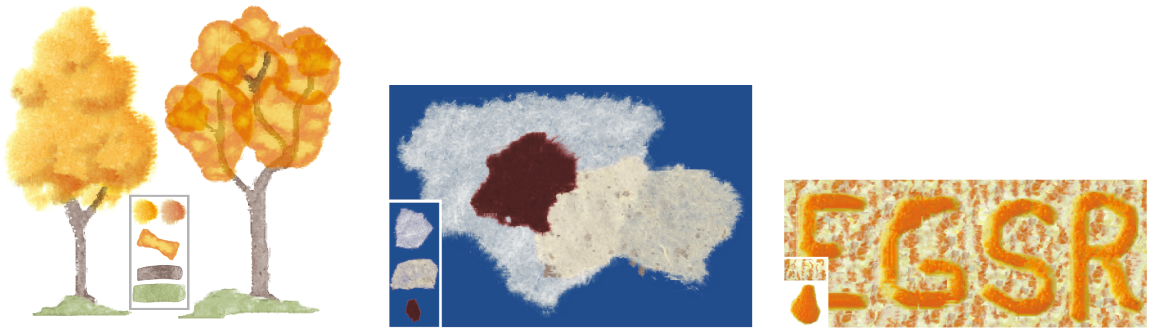


Figure 1.9: *Example-based painting results from* Painting with Texture *[Ritter et al., 2006]. Source textures are inset.*

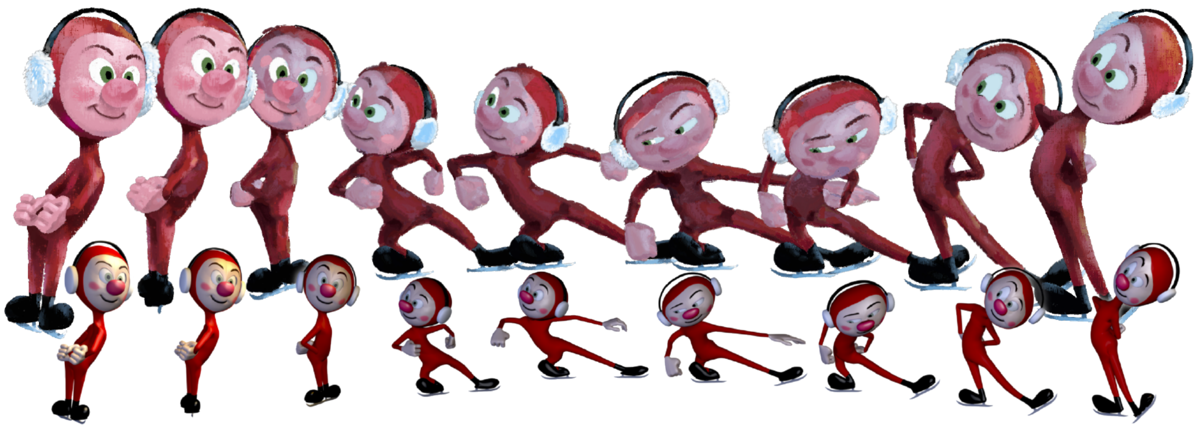


Figure 1.10: *An example of video stylisation by Bénard et al. [2013]. From the rendered input sequence on the bottom, the artist manually stylises the leftmost a rightmost frames (shown above). The stylisation is then propagated to the intermediate frames.*

Example-based Image Stylisation uses a *target image* beside the exemplar to provide structure and guide the synthesis. They generally behave as stylisation filters and may accept regular images for stylisation, or may work as *analogies* by defining a guidance map for both the target and the exemplar and synthesizing the output image on the basis of that [Hertzmann et al., 2001]. It is also applicable for video, as shown in Figure 1.10.

2 Context and Content of the Thesis

The thesis presents methods that provide specific improvements in several example-based image creation scenarios. While they fall under the general headings of example-based painting and example-based image stylisation respectively, and adhere to the outlined use-cases, they introduce qualitative improvements by the virtue of including novel aspects in the problem formulation.

The core of the thesis consists of three distinct developments set in the context defined by previous sections. In this chapter, we provide a brief overview of each, along with a brief discussion of related work against which these are set. More detailed discussions are available in the thesis.

2.1 Painting by Feature: Texture Boundaries for Example-based Image Creation

Painting by Feature [Lukáč et al., 2013] is an example-based painting approach which focuses on handling of *edge features* along with interior textures. Like previous approaches [Ashikhmin, 2001; Ritter et al., 2006], it follows an interactive workflow. Compared to *Painting with Texture* [Ritter et al., 2006], the key contribution is the explicit handling of linear features using a separate line synthesis approach.

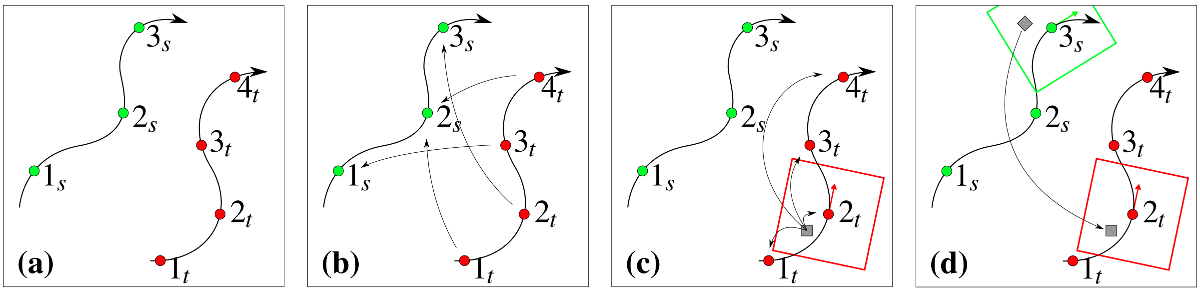


Figure 2.1: *Line feature mapping process. (a) Both the source path and the target path are sampled (respectively, the green and the red circles) at equal intervals. (b) We map the random walk to the target path, determining for each target sample the corresponding source sample. (c) We determine the colour of a pixel (grey square) in the target by finding the nearest target sample and (d) taking the value at the same relative position in the corresponding source patch (coloured squares, arrows denote patch orientation).*

In this approach, user-selected lines from the input image are used as examples to synthesize user-drawn lines in the output. The input linear feature is discretely sampled and the content of the output feature is synthesized by re-arranging these samples using a random walk biased to minimize discontinuity in the output. This novel feature is then rendered in the output image (Figure 2.1).

The line synthesis steps are then alternated with area synthesis step, each consecutive feature being interactively selected by the user and applied immediately thereafter (Figure 2.2).

This gives a better quality result than the use of segmentation masks in the previous approach, as it avoids overconstraining the texture synthesis. Free-form and rapid line and area feature selection also give the user a much greater degree of freedom in selecting the features they want to replicate and permits the use of interior lines as features, rather than merely boundaries between textures.

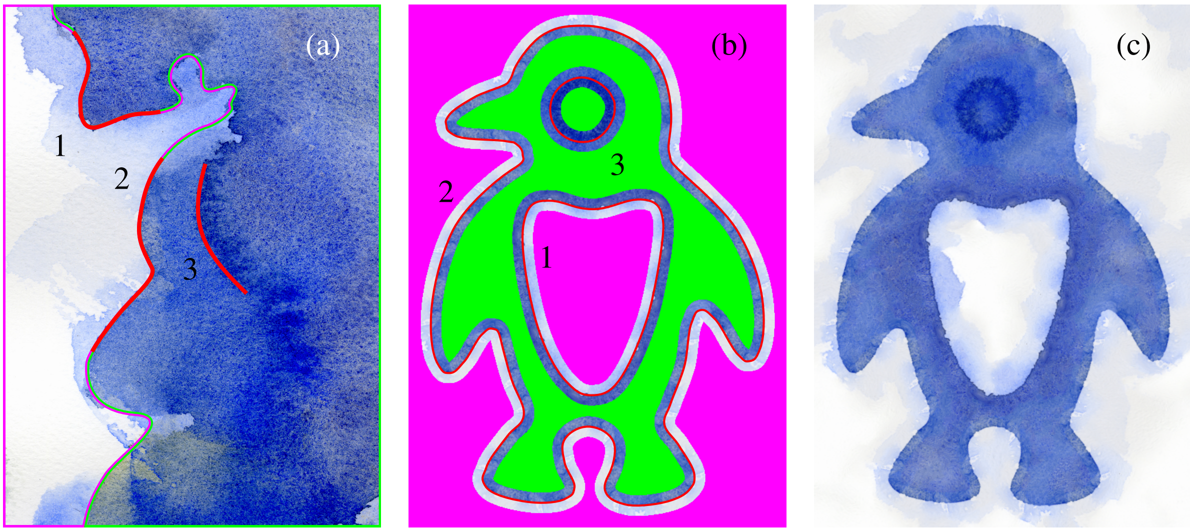


Figure 2.2: *Image creation workflow overview. (a) Annotated source image: two area features delimited by the pink and green outlines, and three line features indicated by the red curves and the numbers. (b) Line feature synthesis along user-specified paths from the corresponding numbered line features of (a). The pink and green areas represent the parts to be filled in by the corresponding area features of (a). (c) Final result after texture transfer by the fill tool. Source credits:* Alessandro Andreuccetti *via deviantART,* mrjive *via OpenClipArt*

2.2 Color Me Noisy: Example-based Rendering of Hand-colored Animations with Temporal Noise Control

Color Me Noisy [Fišer et al., 2014] is a video stylisation approach based on example-based texture synthesis. The approach was borne out of discussions with artists who work with traditional animations and felt that the previous approaches [Bénard et al., 2013], with their focus on maximizing temporal coherence of the result, lost a significant artistic quality bestowed by temporal noise which is usually an artefact of hand-made animation. The approach is therefore based on stylising a video sequence to match the visual style of the example while ensuring that a controllable amount of temporal noise is present without becoming disturbing to the viewer.

The basis of the approach is the observation that the perceived amount of noise corresponds with the spatial extent of image perturbations. Having confirmed this in a perceptual study, we developed a multi-scale re-synthesis approach which preserves features based on a locally-variant spatial threshold, while introducing perturbations on the appropriate scale to introduce randomness (Figure 2.3).

This principally improves on the previous state of art by enriching the stylisation with an additional channel of artistic expression, as the amount of noise may be controlled both spatially and temporally to convey emotion or other artistic intent.

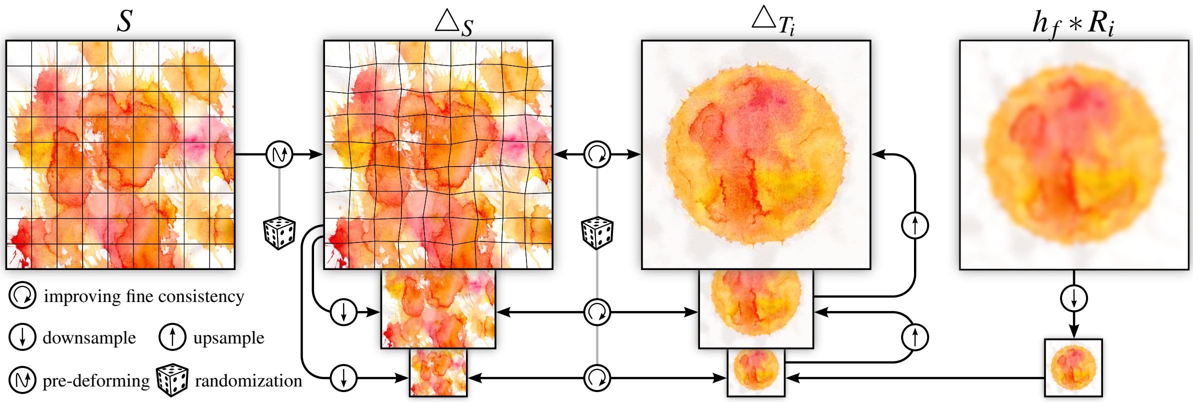


Figure 2.3: *Algorithm—the source drawing medium S is randomly pre-deformed and image pyramids of source $\triangle_S$ and target frame $\triangle_{T_i}$ are built. The coarsest level of $\triangle_{T_i}$ is initialized by downsampled reference frame R_i. The user-specified strength f of the low-pass filter h_f is used for downsampling. The algorithm starts from coarsest level of $\triangle_{T_i}$ and continues towards finer levels. At each level ℓ fine consistency between $\triangle_S$ and $\triangle_T$ is improved. During this process generalized PatchMatch is utilized to find nearest neighbour patches. The seed for the randomized search is always changed to avoid determinism.*

2.3 Brushables: Example-based Edge-aware Directional Texture Painting

Brushables [Lukáč et al., 2015] is an example-based painting approach which focuses on textures that could not be handled by previous approaches – namely textures with anisotropic directionality and fuzzy or gradual edge effects. Compared to *Painting by Feature*, Brushables handle texture edges and interiors simultaneously. This is seemingly a step back towards *Painting with Texture* but manually selecting the boundary becomes impractical for these more difficult textures.

The primary focus is therefore a significantly improved synthesis approach which handles fuzzy boundaries and incorporates an adapted shape descriptor into the synthesis, permitting the handling of edges in a soft, continuous way, as well as permitting the handling of significantly broader edges on which a simple mask-patch approach fails.

This is achieved with two new mechanisms. First, we use sign-aware direction detection and direction authoring to handle the directionality of the textures. Second, we integrate a *shape term* into the objective function, measured based on our own fast shape descriptor (Figure 2.4). In addition, these new constraints necessitate further adaptations to the synthesis process in order to avoid introducing errors or artefacts which would have been otherwise effected by the more free-form nature of the transformations involved.

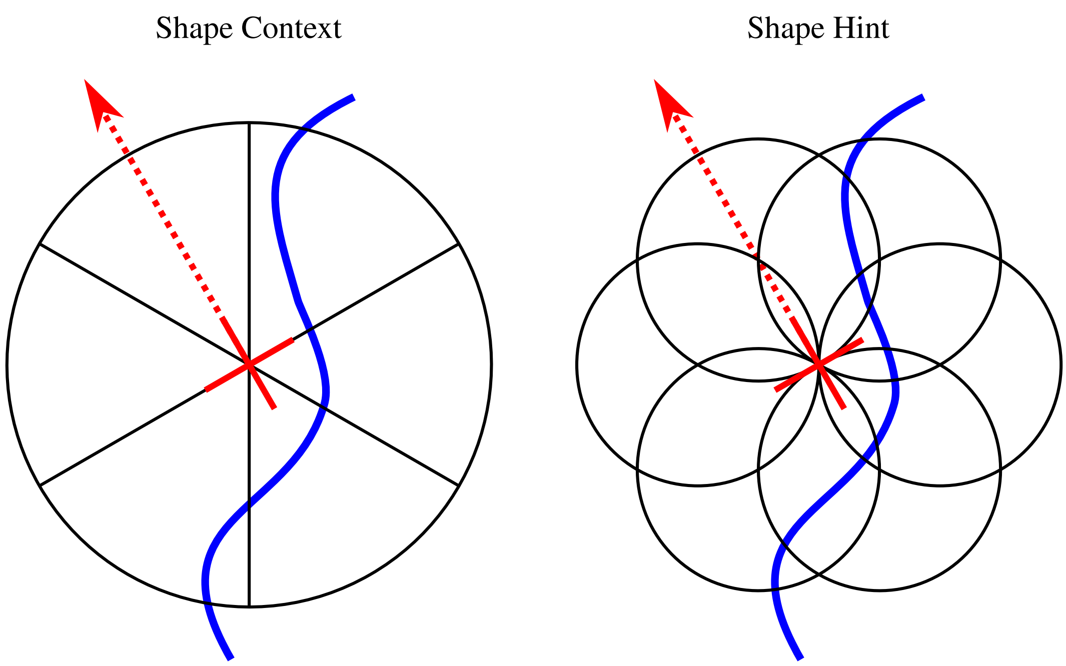


Figure 2.4: *Shape and arrangement of bins in an oriented single-layer* shape context *(left) and in our* Shape Hint *(right).*

3 Conclusion

In this chapter, we briefly summarize the developments presented in the publications mentioned above and their impact on the problem area at large.

3.1 Summary

3.1.1 Painting by Feature

Painting by Feature presented an approach for example-based painting founded on treating boundaries and curvilinear features as a special case with its own synthesis algorithm. The important conceptual point was introducing the idea of example-based toolsets with multiple complementary tools, each focusing on transferring a different aspect of the exemplar's visual style. Independent synthesis of line features is also necessary in order to represent exemplars with high-order structure, as the classical texture synthesis approaches lose this structure on 1D features when downsampling during image pyramid construction.

3.1.2 Color Me Noisy

In *Color Me Noisy*, we presented an approach for example-based video stylization. The novel aspect of this approach is *noise control*; rather than trying to suppress any sort of temporal incoherence, this approach offers a spectrum of options between full suppression to high noise, which can further be locally and temporally variant. This allows for a new channel of artistic expression that was not available with previous approaches, and permits the use of styles that could not have been emulated before, such as hand-coloured animation.

3.1.3 Brushables

Brushables presented an example-based painting approach focusing on difficult texture examples. At the core of the approach was the integration of shape into similarity function, which facilitates seamless handling of complex boundary effects. In conjunction with modern texture analysis and segmentation tools, this new approach is especially suited for painting with textures acquired “in the wild”.

3.2 Discussion

Along with the one presented in Painting by Feature, several approaches intended strictly for synthesis of curvilinear features were developed [Lu et al., 2013; Zhou et al., 2013; Lu et al., 2014]. These were more specialised and produced better results in their chosen cases but did not integrate with area texture synthesis. While more advanced texture painting approaches like Brushables can synthesise texture boundaries satisfactorily, they still tend to have trouble with highly structured regular line features, which suggests an area of future development.

The work on video stylization [Fišer et al., 2014; Bénard et al., 2013] has been followed up by further video stylisation work [Jamriška et al., 2015] which, like concurrent texture synthesis work [Kaspar et al., 2015], focused on implementing a new synthesis approach which would not only faithfully reproduce individual textural features but carefully preserve their relative frequency.

In conclusion, research, of which the thesis was a part, has in recent years both laid the groundwork for example-based image creation and successfully applied it for consumer use (e.g. in software like Adobe Photoshop). The thesis in particular presents three new methods which represent a tangible improvement on the state of art, as signified by their acceptance for publication in prestigious scientific journals.

For a more comprehensive technical presentation of the above methods, as well as an extended discussion, we invite the reader to refer to the thesis.

References

[Ashikhmin, 2001] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of Symposium on Interactive 3D graphics*, pages 217–226, 2001.

[Assa and Cohen-Or, 2012] Jackie Assa and Daniel Cohen-Or. More of the same: Synthesizing a variety by structural layering. *Computers & Graphics*, 36(4):250–256, 2012.

[Barnes et al., 2009] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.

[Bénard et al., 2013] Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. Stylizing animation by example. *ACM Transactions on Graphics*, 32(4):119, 2013.

[Bonneel et al., 2013] Nicolas Bonneel, Kalyan Sunkavalli, Sylvain Paris, and Hanspeter Pfister. Example-based video color grading. *ACM Transactions on Graphics*, 32(4):39, 2013.

[Chen et al., 2015] Zhili Chen, Byungmoon Kim, Daichi Ito, and Huamin Wang. Wetbrush: GPU-based 3D painting simulation at the bristle level. *ACM Transactions on Graphics*, 34 (6):200, 2015.

[Criminisi et al., 2004] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.

[Darabi et al., 2012] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, 31(4):82, 2012.

[Ebert et al., 2002] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2002.

[Efros and Leung, 1999] Alexei Efros and Thomas Leung. Texture synthesis by non-parametric sampling. *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, 2:1033, 1999.

[Efros and Freeman, 2001] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, 2001.

[Eslami et al., 2012] Saeideh M. Ali Eslami, Nicolas Heess, Christopher K. I. Williams, and John Winn. The shape boltzmann machine: a strong model of object shape. In *IEEE Conference on Computer Vision and Pattern Recognition 2012*, pages 406–413, 2012.

[Fišer et al., 2014] Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente, and Daniel Sýkora. Color me noisy: Example-based rendering of hand-colored animations with temporal noise control. *Computer Graphics Forum*, 33(4):1–10, 2014.

[Freeman et al., 2002] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[Hertzmann et al., 2001] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of SIGGRAPH 2001*, pages 327–340, 2001.

[Jamriška et al., 2015] Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Lazyfluids: Appearance transfer for fluid animations. *ACM Transactions on Graphics*, 34(4):92, 2015.

[Kaspar et al., 2015] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. *Computer Graphics Forum*, 34(2):349–359, 2015.

[Kopf et al., 2007] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2D exemplars. *ACM Transactions on Graphics*, 26(3):2, 2007.

[Kwatra et al., 2003] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures. *ACM Transactions on Graphics*, 22(3):277, 2003.

[Kwatra et al., 2005] Vivek Kwatra, Irfan A. Essa, Aaron F. Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics*, 24(3):795–802, 2005.

[Lefebvre and Hoppe, 2005] Sylvain Lefebvre and Hugues Hoppe. Parallel controllable texture synthesis. *ACM Transactions on Graphics*, 24(3):777–786, 2005.

[Lefebvre and Hoppe, 2006] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics*, 25(3):541–548, 2006.

[Liu et al., 2014] Yiming Liu, Michael Cohen, Matt Uyttendaele, and Szymon Rusinkiewicz. AutoStyle: Automatic style transfer from image collections to users' images. *Computer Graphics Forum*, 33(4):21–31, 2014.

[Lu et al., 2013] Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam Finkelstein. Realbrush: Painting with examples of physical media. *ACM Transactions on Graphics*, 32(4): 117, 2013.

[Lu et al., 2014] Jingwan Lu, Connelly Barnes, Connie Wan, Paul Asente, Radomir Mech, and Adam Finkelstein. Decobrush: Drawing structured decorative patterns by example. *ACM Transactions on Graphics*, 33(4):90, 2014a.

[Lu et al., 2014b] Jingwan Lu, Stephen DiVerdi, Willa A. Chen, Connelly Barnes, and Adam Finkelstein. RealPigment: Paint compositing by example. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering*, NPAR '14, pages 21–30. ACM, 2014b.

[Lukáč et al., 2013] Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sýkora. Painting by feature: Texture boundaries for example-based image creation. *ACM Transactions on Graphics*, 32(4):116, 2013.

[Lukáč et al., 2015] Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Brushables: Example-based edge-aware directional texture painting. *Computer Graphics Forum*, 34(7):257–268, 2015.

[Měch and Miller, 2012] Radomír Měch and Gavin Miller. The *Deco* framework for interactive procedural modeling. *Journal of Computer Graphics Techniques*, 1(1):43–99, 2012.

[Paget and Longstaff, 1998] Rupert Paget and Ian D. Longstaff. Texture synthesis via a non-causal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998.

[Risser et al., 2010] Eric Risser, Charles Han, Rozenn Dahyot, and Eitan Grinspun. Synthesizing structured image hybrids. *ACM Transactions on Graphics*, 29(4):85, 2010.

[Ritter et al., 2006] Lincoln Ritter, Wilmot Li, Brian Curless, Maneesh Agrawala, and David Salesin. Painting with texture. In *Proceedings of Eurographics Symposium on Rendering*, pages 371–376, 2006.

[Rosenberger et al., 2009] Amir Rosenberger, Daniel Cohen-Or, and Dani Lischinski. Layered shape synthesis: Automatic generation of control maps for non-stationary textures. *ACM Transactions on Graphics*, 28(5):107, 2009.

[Rother et al., 2004] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[Shih et al., 2014] YiChang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4): 148, 2014.

[Simakov et al., 2008] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, pages 1–8, 2008.

[Wei and Levoy, 2000] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[Wei and Levoy, 2003] Li-Yi Wei and Marc Levoy. Order-independent texture synthesis. Technical Report TR-2002-01, Computer Science Department, Stanford University, 2003.

[Wexler et al., 2007] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.

[Zhou et al., 2013] Shizhe Zhou, Anass Lasram, and Sylvain Lefebvre. By-example synthesis of curvilinear structured patterns. *Computer Graphics Forum*, 32(2pt3):355–360, 2013.

4 Author's Publications

The following publications were co-authored by the author of the thesis and published in high-impact journals, as indexed by ISI.

Michal Lukáč, Jakub Fišer, Jean-Charles Bazin, Ondřej Jamriška, Alexander Sorkine-Hornung, and Daniel Sýkora. Painting by feature: Texture boundaries for example-based image creation. *ACM Transactions on Graphics*, 32(4):116, 2013
Authorship share: 50%

Cited in:

Shizhe Zhou, Changyun Jiang, and Sylvain Lefebvre. Topology-constrained synthesis of vector patterns. *ACM Transactions on Graphics*, 33(6):215, 2014

Jingwan Lu, Connelly Barnes, Connie Wan, Paul Asente, Radomir Mech, and Adam Finkelstein. Decobrush: Drawing structured decorative patterns by example. *ACM Transactions on Graphics*, 33(4):90, 2014

Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. Autocomplete painting repetitions. *ACM Transactions on Graphics*, 33(6):172, 2014

Riccardo Roveri, A. Cengiz Öztireli, Sebastian Martin, Barbara Solenthaler, and Markus Gross. Example Based Repetitive Structure Synthesis. *Computer Graphics Forum*, 34(5):39–52, 2015

Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Lazyfluids: Appearance transfer for fluid animations. *ACM Transactions on Graphics*, 34(4):92, 2015

Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. Autocomplete hand-drawn animations. *ACM Transactions on Graphics*, 34(6):169, 2015

Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente, and Daniel Sýkora. Color me noisy: Example-based rendering of hand-colored animations with temporal noise control. *Computer Graphics Forum*, 33(4):1–10, 2014
Authorship share: 15%

Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Brushables: Example-based edge-aware directional texture painting. *Computer Graphics Forum*, 34 (7):257–268, 2015
Authorship share: 50%

5 Summary

In modern computer graphics, the collective creativity of digital artists far outstrips the capabilities of available tools. In the drive to catch up, more traditional simulation-based and procedural tools are being supplemented by example-based methods, which take advantage of the cornucopia of available data as guidance.

This thesis is an anthology of articles published in various journals in the period of 2013 to 2015, which present three such novel methods. The first is an example-based painting approach which focuses on proper preservation of feature-rich boundaries between textures. The second is a video stylization approach which allows fine control over the amount of temporal noise in the output, and the third is another texture painting approach which focuses on handling shape-specific features.

V moderní počítačové grafice kolektivní tvořivost digitálních umělců dalece předhání možnosti dostupných nástrojů. V rámci snah o překonání těchto nedostatků jsou tradičnější simulační a procedurálně generované nástroje doplňovány novými metodami založenými na tvorbě obsahu podle příkladů, které využívají nepřeberného množství dostupných dat jako předlohy.

Tato disertace je antologií článků publikovaných v několika časopisech v období let 2013 až 2015, v nichž jsou prezentovány tři nové takové metody. První je metodou pro kreslení podle příkladu, která se zaměřuje na věrohodné zachování vizuálně bohatých rozhraní mezi texturami. Druhá je algoritmem pro stylizaci videa, který umožňuje detailní kontrolu míry časového šumu ve výstupu. Třetí je další metoda pro kreslení podle příkladu, která se zaměřuje na zachování efektů specifických tvaru.

Michal Lukáč

CONTACT INFORMATION	Department of Computer Graphics and Interaction Faculty of Electrical Engineering Czech Technical University in Prague Karlovo náměstí 13, 121 35 Prague 2, CZ	*phone:* *fax:* *e-mail:*	+420 724 149 688 +420 2 2435 7556 `michal.lukac@fel.cvut.cz`

BIOGRAPHY

Born
Martin, Slovak Republic, 2nd June 1989

Married
wife: Lenka (*1986),

EDUCATION

Czech Technical University in Prague
M.Sc., Computer Science and Engineering, June 2011.
B.Sc., Web and Multimedia, June 2009.

ACADEMIC EXPERIENCE

Czech Technical University in Prague

Postgraduate Student	**September 2011 – present**
Graduate Student	**September 2009 – June 2011**
Teaching Assistant	**January 2008 – present**
Undergraduate Student	**September 2006 – June 2009**

PROFESSIONAL EXPERIENCE

Adobe Research
Procedural Imaging Group Internship — **June 2015 – September 2015**

Adobe Research
Procedural Imaging Group Internship — **July 2014 – October 2014**

I. Certifikační Autorita, a.s.
Proof-of-concept development — **April 2011 – September 2011**

Škoda Auto, a.s.
Internship at the Digital Prototyping Laboratory — **June 2010 – September 2010**

JOURNAL PUBLICATIONS

M. Lukáč, J. Fišer, P. Asente, Jingwan Lu, Eli Shechtman, and D. Sýkora:
Brushables: Example-based Edge-aware Directional Texture Painting
Computer Graphics Forum, 2015.

J. Fišer, M. Lukáč, O. Jamriška, M. Čadík, Y. Gingold, P. Asente, and D. Sýkora:
Color Me Noisy: Example-based Rendering of Hand-colored Animations with Temporal Noise Control
Computer Graphics Forum, 2014.

M. Lukáč, J. Fišer, J.-C. Bazin, O. Jamriška, A. Sorkine-Hornung, and D. Sýkora:
Painting by feature: texture boundaries for example-based image creation
ACM Transactions on Graphics, 2013.

FUNDINGS

ViCiTis: Virtual Cities in Time and Space (SGS10/291/OHK3/3T/13)
V3C – Visual Computing Competence Center (TE01010415)

RESEARCH INTERESTS

Example-based Content Synthesis
Image Inpainting
Monte Carlo Algorithms
Markov Models

PROGRAMMING SKILLS

Profficient
C++, CUDA C, GLSL

Intermediate
C#, HLSL, JavaScript, LISP

LANGUAGES

English	*Proficient, C2 certified (Cambridge ESOL CPE)*
Russian	*Passive, intermediate*
German	*Passive, intermediate*