České vysoké učení technické v Praze
Fakulta elektrotechnická

Czech Technical University in Prague
Faculty of Electrical Engineering

Ing. Daniel Sýkora, Ph.D.

Dodání hloubky do ručních kreseb

Adding Depth to Hand-drawn Images

Summary

The human visual system can easily perceive depth in hand-drawn images thanks to understanding of high-level structure encoded in the drawing. Such a knowledge is typically hidden to the computer and thus algorithmic addition of depth becomes a challenging task. In this talk we present new depth assignment techniques which resolve this problem by exploiting a set of sparse user-specified constraints that express pair-wise relationship between selected parts in the scene. Resulting depth values are then computed as a solution to an optimization problem which enforces aforementioned constraints while taking into account smoothness of the resulting depth field. It is formulated as a quadratic programming (QP) problem of which solution is computationally demanding. To allow for incremental workflow we sub-divide the original QP into a sequence of simpler problems that can be solved notably faster and thus enable interactive response.

Souhrn

Pro lidský vizuální systém je vnímání hloubky v ručních kresbách přirozené. Hlavní roli v celém procesu hraje zejména chápání geometrické struktury obrazu založené na zkušenosti z reálného světa. Tato informace je typicky v paměti počítače nedostupná, čímž se problém algoritmického dodání hloubky do ruční kresby stává mimořádně obtížný. V této přednášce představíme nové algoritmy, které umožní problém efektivně řešit s využitím nápovědy dodané uživatelem. Její podstatou je řídká specifikace relativních vztahů mezi vybranými částmi obrazu (bod A leží blíže k pozorovateli než bod B). Výsledná hloubková mapa je vypočtena jako řešení optimalizačního problému, který si klade za cíl vyhovět uživatelem specifikovaným vztahům a zároveň předpokládá hladké rozložení hloubek v místech, kde původní obraz nebosahuje tvarové kontury. Problém je formulován jako úloha kvadratické programování (KP), jejíž řešení je výpočetně náročné. Pro zajištění interaktivní odezvy je původní KP úloha rozdělena do několika jednodušších podúloh, jejichž řešení lze získat v podstatně kratším čase.

Klíčová slova: počítačová grafika, počítačové vidění, zpracování obrazu, rekonstrukce hloubkových map, konverze ze 2D do 3D, ruční kresby, kvadartické programování

Keywords: computer graphics, computer vision, image processing, depth map reconstruction, 2D-to-3D conversion, hand-drawn images, quadratic programming

Contents

1 Introduction

Recovering depth from a single image remains an open problem after decades of active research. In this talk we focus on a specific variant of the problem where the input image is hand-crafted line drawing. As opposed to previous attempts to provide complete 3D reconstruction either by imposing various geometric assumptions [LS96, VM02, LFG08] or using sketch-based interfaces to create the 3D model incrementally [IMT99, KH06, NISA07, JC08, GIZ09], we seek for a height field or a bas-relief-type approximation consistent with the observer's perception of depth in the scene. Although such approximation cannot provide full 3D modelling capabilities it is sufficient for numerous important tasks that can arise in 2D pipelines. It enables to maintaining correct visibility and connectivity of individual parts during interactive 2D shape manipulation [IMH05], deformable image registration [SDC09a], or fragment composition [SBv05] (see Figure 1.1 top). In the context of image enhancement it helps to improve perception of depth [LCD06], generate 3D-like shading [Joh02] and global illumination effects [SKv*14], or produce stereoscopic imagery [LMY*13] (see Figure 1.1 bottom).

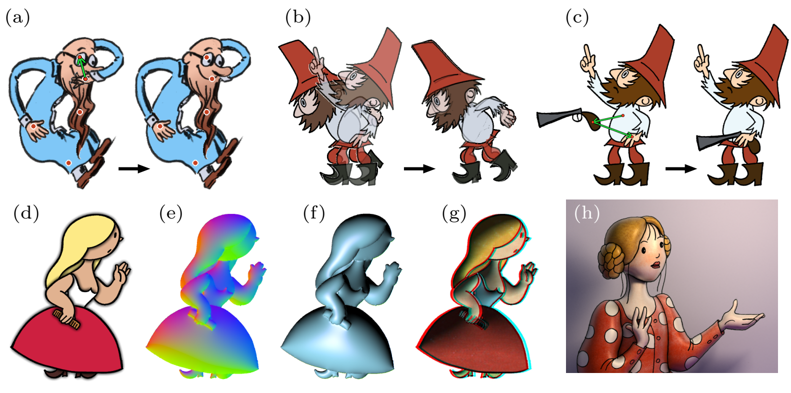

Figure 1.1: Applications—depth maps generated using our approach can be utilized to maintain correct visibility in (a) as-rigid-as-possible shape manipulation, (b) deformable image registration, and (c) fragment composition. They can also help to (d) enhance perception of depth, (e) interpolate normals with respect to depth discontinuities, and produce (f) 3D-like shading, (g) stereoscopic images, or (h) global illumination effects.

In general it is hard to obtain consistent depth information from hand-drawn image automatically as geometric and semantic context is typically hidden and cannot be simply inferred. Nevertheless, a few user-provided hints can notably simplify the problem. As human observers typically fail to specify absolute depth values but are much more accurate in telling whether some part of the object is in front of another and vice versa [KvDK96, Koe98], we decided to formulate an optimization problem where these pairwise relations serve as inequality constrains. The actual depth values are then reconstructed by assuming smoothness except at shape discontinuities indicated by outlines present in the original hand-drawn image. For applications where a notion of volume is necessary (e.g., stereo conversion, simulation of 3D-like shading, or globall illumination effects) an additional non-zero curvature constraint can be added to enforce inflation and topology changed to allow for a stack of multiple interconnected layers.

1.1 Related work

Reconstruction of depth information from a line drawing is a central challenge in the field of computer vision [Mal86]. In general, the problem is highly under-constrained, and additional knowledge is typically required to obtain reasonable results automatically [CIF*12]. It becomes simpler when the input drawing is clean and satisfy some basic geometric rules [LS96, VM02, LFG08, WCLT09], however, in case of cartoon images these rules are typically broken due to inaccuracies or deliberate stylization, thus manual intervention becomes inevitable. This makes the process more akin to modeling from scratch than reconstruction [GIZ09]. The user has to trace curves over the original image and/or sketch new regions directly with a mouse or a tablet. Then inflation [IMT99, OCN04, CON05, OSJ11] enhanced by topological embedding [KH06] or additional control curves [NISA07, JC08, AJC11] is used to produce final height field or a 3D model. Our approach greatly simplifies this process. There is no need for contour tracing, tedious specification of absolute depth values or control curves. From a few simple annotations, our system automatically infers single height field or a stack of inflated surfaces that preserve relative depth order and continuity.

2 Method

In this chapter we present our novel approach to depth reconstruction based on a set of sparse depth (in)equalities. We first formulate an optimization problem and show how to solve it using quadratic programming, then we proceed to an approximative solution that allows interactive feedback.

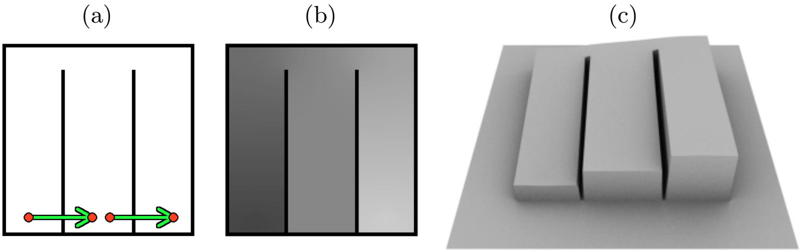

Figure 2.1: Depth from sparse inequalities—(a) input image with two user-specified depth inequalities (green arrows), (b) output depth map obtained by solving a specific quadratic program, (c) 3D visualization of the resulting height field.

2.1 Problem formulation

As an input we have an image $\mathcal{I}$ for which at pixels $p, q \in \mathcal{I}$ the user specified desired depth equalities $\{p, q\} \in \mathcal{U}_=$ and inequalities $\{p, q\} \in \mathcal{U}_>$ (see Figure 2.1a). Now the task is to assign depth values to all pixels so that the user-specified constraints are satisfied while smoothness is assumed except at locations where the intensity gradient of $\mathcal{I}$ is high (see Figure 2.1b). This can be formulated as an optimization problem where the aim is to find a minimum of the following energy function [SSJ*10]:

$$\text{minimize:} \quad \sum_{p \in \mathcal{I}} \sum_{q \in \mathcal{N}_p} w_{pq}(d_p - d_q)^2 \tag{2.1}$$

$$\begin{aligned} \text{subject to:} \quad & d_p - d_q = 0 \quad \forall \{p, q\} \in \mathcal{U}_= \\ & d_p - d_q \geq \epsilon \quad \forall \{p, q\} \in \mathcal{U}_> \end{aligned}$$

where d_p denotes the depth value assigned to a pixel p, $\mathcal{N}_p$ is a 4-connected neighborhood of p, and ϵ is some positive number greater than zero (e.g. $\epsilon = 1$). The weight w_{pq} is set as in [Gra06]:

$$w_{pq} \propto \exp\left(-\frac{1}{\sigma}(\mathcal{I}_p - \mathcal{I}_q)^2\right),$$

where $\mathcal{I}_p$ is the image intensity at pixel p, and σ is a mean contrast of edges. A closer inspection of (2.1) reveals that our problem can be rewritten as a quadratic program (P):

$$\text{minimize:} \quad \frac{1}{2} d^T L d \tag{2.2}$$

$$\begin{aligned} \text{subject to:} \quad & d_p - d_q = 0 \quad \forall \{p,q\} \in \mathcal{U}_= \\ & d_p - d_q \geq \epsilon \quad \forall \{p,q\} \in \mathcal{U}_> \end{aligned}$$

where L is a large sparse matrix of size $|\mathcal{I}|^2$ representing the so called Laplace–Beltrami operator [Gra06]. It can be solved directly using, e.g., using an active set method [GMSW84], however, in practice the computation can take tens of seconds even for very small images. To allow instant feedback we propose an approximative solution that yields similar results with significantly lower computational overhead.

2.2 Interactive approximation

The key idea of our approximative solution is to decompose the problem into two separate tasks: (1) multi-label segmentation (Figure 2.2a) and (2) depth assignment (Figure 2.2b). To do so we first treat the user-specified constraints as an unordered set of labels $\mathcal{L}$. Then we exploit our interactive multi-label segmentation algorithm tailored to cartoon images—LazyBrush [SDC09b].

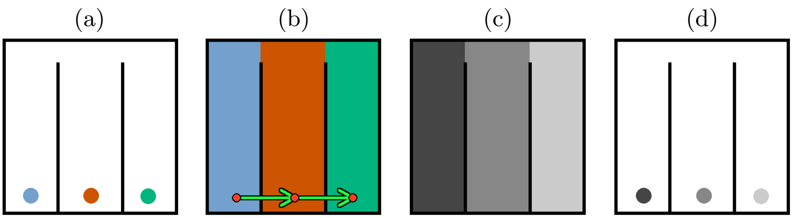

Figure 2.2: Approximative solution allowing interactive feedback: (a) multi-label segmentation, (b) topological sorting, (c) intermediate depth map, (d) boundary conditions for Laplace equation.

Once regions are separated we can apply depth inequalities $\mathcal{U}_<$, however, now all constraints that fall inside one region are associated to one node that represents the underlying region. These nodes are then interconnected by a set of oriented edges representing desired inequalities (see Figure 2.2b). Nodes and edges form an arbitrary oriented graph for which consistent depth assignment (see Figure 2.2c) exists if and only if it does not contain oriented loops. The algorithm that can solve this problem is known as topological sorting [Kah62]. Its key component is a detection of oriented loops which allows to recognize whether the newly added constraint is consistent. If not, the segmentation phase can be invoked again to create a new region and update the depths with another topological sorting step. Since segmentation and depth assignment steps are independent they can be executed incrementally until the desired depth assignment is reached.

When the absolute depths are known, we can reconstruct smooth depth transitions (see Figure 2.1b) to obtain the same result as provided by the quadratic program (2.2). To do so we use a simplified version of (2.1):

$$\text{minimize:} \sum_{p\in\mathcal{I}} \sum_{q\in\mathcal{N}_p} v_{pq}(d_p - d_q)^2 \tag{2.3}$$

$$\text{subject to:} \quad d_p = \hat{d}_p \quad \forall p \in \mathcal{U}_\circ$$

where

$$v_{pq} \propto \begin{cases} \mathcal{I}_p & \text{for } \hat{d}_p \neq \hat{d}_q \\ 1 & \text{otherwise,} \end{cases}$$

$\hat{d}$ denotes depth values from a depth map produced by the topological sorting, and $\mathcal{U}_\circ$ is a subset of pixels used in $\mathcal{U}_<$ or $\mathcal{U}_=$ (see Figure 2.2d). We let the user decide whether these constraints will be included in $\mathcal{U}_\circ$. The reason we use $\mathcal{I}_p$ instead of 0 is that we have to decide whether the depth discontinuity is real or virtual. Real discontinuities have $\mathcal{I}_p = 0$ but those which are not covered by contours (like upper gaps in Figure 2.1b) have $\mathcal{I}_p = 1$, i.e., they preserve continuity and therefore produce smooth depth transitions. The energy (2.3) can be minimized by solving the Laplace equation

$$\nabla^2 d = 0 \tag{2.4}$$

with the following boundary conditions ($q \in \mathcal{N}_p$):

$$\begin{aligned} \text{Dirichlet:} \quad & d_p = \hat{d}_p && \iff && p \in \mathcal{U}_\circ \\ \text{Neumann:} \quad & d'_{pq} = 0 && \iff && \hat{d}_p \neq \hat{d}_q \wedge \mathcal{I}_p = 0 \end{aligned}$$

for which a fast GPU-based solver exists [JCW09].

3 Extensions

In this chapter we demonstrate how to extend the original approach presented in Chapter 2 to handle surface inflation, a stack of multiple interconnected height fields, and how to estimate relative depth order automatically.

3.1 Inflation

A key assumption of the method presented in Section 2.1 is that the resulting height field has nearly zero curvature, i.e., it tends to produce flat shapes (see Figure 2.1c). However, for applications such simulation of 3D-like shading or stereo conversion a buckled surface is preferred as it better conveys the perception of volume in the drawing. Such a buckled shape can be enforced by setting non-zero right hand side in (2.4), i.e., solving a Poisson equation:

$$\nabla^2 \tilde{d} = c \tag{3.1}$$

where $c \in \mathbb{R}$ is a scalar specifying how much the inflated surface should buckle. The resulting $\tilde{d}$ produces a parabolic profile (Figure 3.1a). If desired, the user may specify a custom cross-section function to convert the profile into an alternative shape. A typical example of such a function is $f(x) = s\sqrt{\tilde{d}(x)}$, where $s \in \mathbb{R}$ is a scaling factor which makes it possible to obtain spherical profiles (Figure 3.1b). This can be applied, for example, when the user wants to produce a concave or flatter profile.

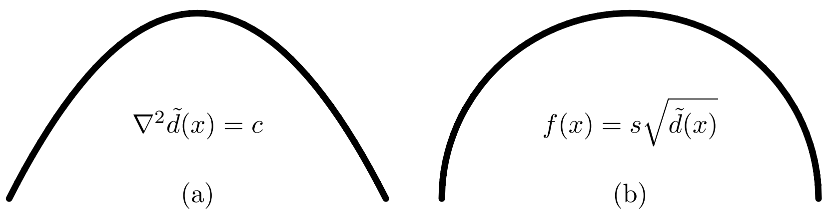

Figure 3.1: Inflation—(a) initial parabolic profile produced by the Poisson equation, (b) spherical profile obtained by applying a cross-section function with square root.

3.2 Layering

Another assumption of the original formulation discussed in Section 2.1 is that the resulting depth map is only a 2D function (single height field). To achieve more realistic results in applications such as simulation of global illumination effects, a stack of multiple height fields is necessary. This allows, e.g., to simulate appealing shadow effects as well as plausible light transport through holes (see Figure 3.2).

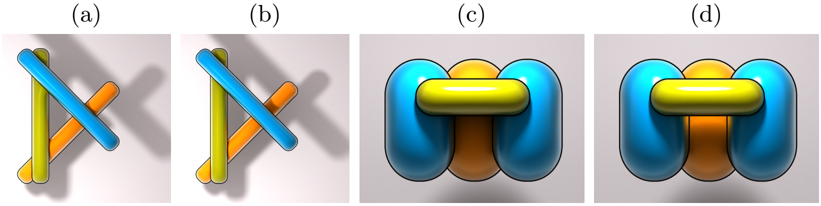

Figure 3.2: Layering—single height field cannot be used for simulation of more complex illumination effects such as cast shadows or light propagation through holes (a,c). Multiple layers has to be created and stitched together to support them (b,d).

To create such multi-layer structure we use illusory surfaces [GPR98] to estimate occluded boundaries of regions obtained during the segmentation phase (Section 2.2).

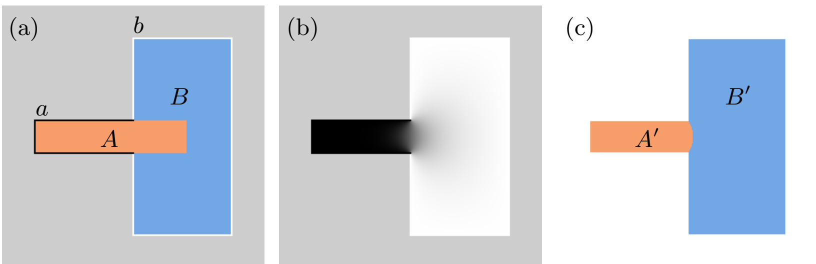

Figure 3.3: Illusory surfaces—(a) an illusory surface B' of the region B can be estimated using two curves a and b with probabilities $p_a = 0$ (black) and $p_b = 1$ (white), see [GPR98], (b) p after the diffusion, (c) pixels which belong to B' have $p > 0.5$. As the area of B' is bigger than area of B, we can conclude region B was occluded by region A.

The illusory surface B' of region B is created by constructing two curves: a and b with initial values $p_a = 0$ and $p_b = 1$ (see Figure 3.3a), these represent probabilities whether the underlying pixels belong to the silhouette of region B. These values are diffused into the compound area of both regions (Figure 3.3b). Pixels with $p > 0.5$ are then treated as interior pixels of the illusory surface B'. Illusory surfaces can also be utilized to predict relative depth order between neighbour regions. This can be done by checking whether the area of the illusory surface B' is greater than the area of its visible portion B (Figure 3.3c). This measurement can be applied to each pair of neighboring regions to create initial depth inequalities which can be further corrected by the user.

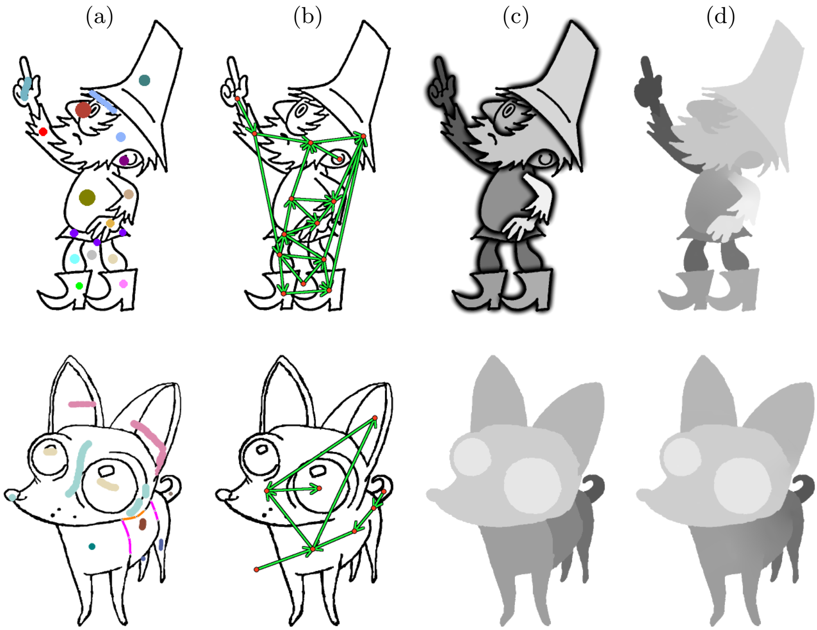

Figure 3.4: Results—depth from sparse (in)equalites: (a) user-specified depth equalities and virtual contours, (b) depth inequalities, (c) depth visualization used during the interactive session, (d) final depth map.

Once occluded boundaries are estimated we apply inflation (Section 3.1) to produce a set of buckled surfaces f_i and arrange them together to satisfy the relative depth order specified by the user. This can be accomplished by solving for a function g_i such that the sum of $f_i + g_i$ satisfies equalities (specifying that two regions should exactly meet at given boundary points) or inequalities (specifying that one region should be above/below another one). The problem is similar to that solved in Section 2.1 only the position of constraints and topology of the resulting surface is different. More details can be found in [SKv*14].

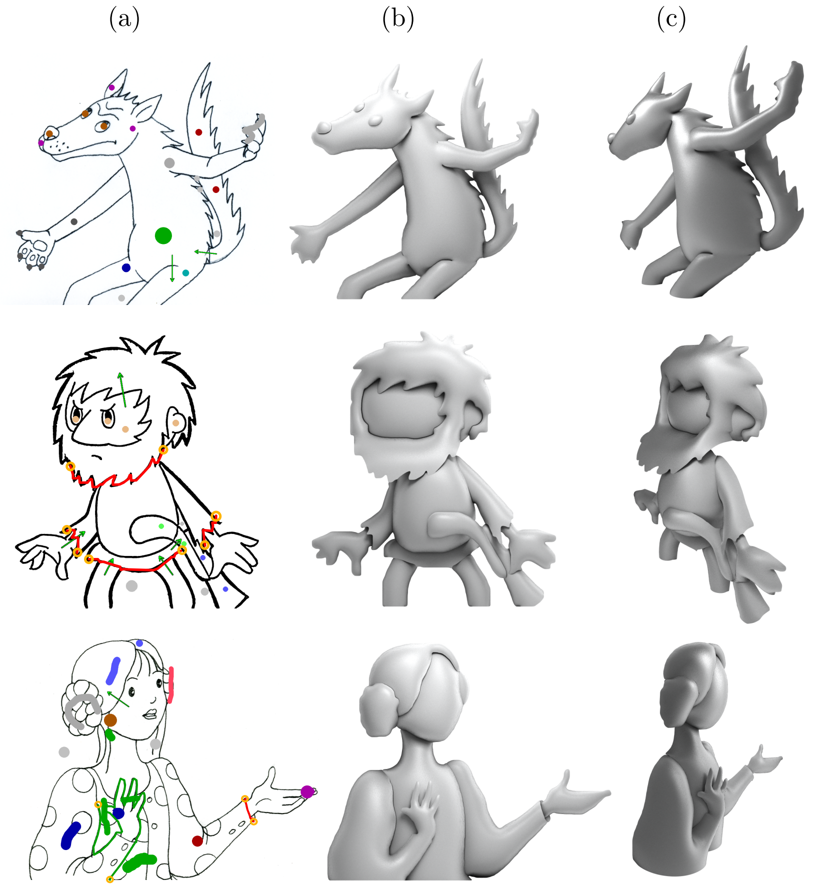

Figure 3.5: Results—multiple layers with inflation: (a) the original drawing with user-specified annotations for segmentation (color scribbles), depth inequalities (green arrows) and specific boundary conditions (red & green curves), (b) resulting bas-relief-type mesh rendered using orthographics projection, (c) sideviews rendered using a perspective camera. Note the bas-relief-type structure which is not obvious from the orthographic view.

4 Results

Examples of depth maps generated using the method described in Chapter 2 are presented in Figure 3.4. All user interactions are recorded in two separate layers—one for equality scribbles (Figure 3.4a) and second for equalities (Figure 3.4b). These two layers are hidden during the interaction phase to avoid clutter. It is typically sufficient to see only the currently added constraint together with a visualization of an intermediate depth map (Figure 3.4c).

Note that users typically do not tend to specify a minimal set of inequalities (see Figure 3.4b), instead they start with local constraints and then refine the result using more distant relations. Such an incremental process corresponds to how the human visual system works, i.e., it starts with local cues and proceeds towards more complex global relations to reach the overall perception of depth in the scene [KvDK96].

Examples of bas-relief-type meshes generated using extensions described in Chapter 3 are presented in Figure 3.5. In each example, we show the original hand-drawn image together with the user-specified annotations (Figure 3.5a). Besides segmentation scribbles, the annotations contain also additional depth inequalities to correct prediction errors. There are also user-specified boundary conditions to avoid rounding or stitching at specific locations on the resulting mesh. To create them the user needs to click on endpoints and the system automatically finds the path along a boundary of the nearest region.

5 Conclusion

In this talk we have presented a novel approach that enables artists to quickly add depth information into their hand-drawn images. A key benefit of the proposed method as compared to previous depth assignment techniques is that it does not require specification of absolute depth values. Instead it uses relative proportions which better corresponds to the depth recovery mechanism found in the human visual system and thus makes the process more intuitive and less tedious. Moreover, presented extensions enable production of bas-relief-type approximations which can be utilized to produce a visual appearance or stereo effects typically only found in complex 3D pipelines. We believe our approach will greatly simplify workflow in 2D pipelines and bring visual richness of 3D techniques into the world of hand-drawn animation.

Bibliography

[AJC11] Andrews J., Joshi P., Carr N. A.: A linear variational system for modelling from curves. *Computer Graphics Forum 30*, 6 (2011), 1850–1861.

[CIF*12] Cole F., Isola P., Freeman W. T., Durand F., Adelson E. H.: Shapecollage: Occlusion-aware, example-based shape interpretation. In *Proceedings of European Conference on Computer Vision* (2012), pp. 665–678.

[CON05] Chen B.-Y., Ono Y., Nishita T.: Character animation creation using hand-drawn sketches. *The Visual Computer 21*, 8-10 (2005), 551–558.

[GIZ09] Gingold Y., Igarashi T., Zorin D.: Structured annotations for 2D-to-3D modeling. *ACM Transactions on Graphics 28*, 5 (2009).

[GMSW84] Gill P. E., Murray W., Saunders M. A., Wright M. H.: Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Transactions on Mathematical Software 10*, 3 (1984), 282–298.

[GPR98] Geiger D., Pao H.-K., Rubin N.: Salient and multiple illusory surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (1998), pp. 118–124.

[Gra06] Grady L.: Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 11 (2006), 1768–1783.

[IMH05] Igarashi T., Moscovich T., Hughes J. F.: As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics 24*, 3 (2005), 1134–1141.

[IMT99] Igarashi T., Matsuoka S., Tanaka H.: Teddy: A sketching interface for 3D freeform design. In *ACM SIGGRAPH Conference Proceedings* (1999), pp. 409–416.

[JC08] Joshi P., Carr N. A.: Repoussé: Automatic inflation of 2D artwork. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2008), pp. 49–55.

[JCW09] Jeschke S., Cline D., Wonka P.: A GPU Laplacian solver for diffusion curves and Poisson image editing. *ACM Transaction on Graphics 28*, 5 (2009).

[Joh02] Johnston S. F.: Lumo: Illumination for cel animation. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering* (2002), pp. 45–52.

[Kah62] Kahn A. B.: Topological sorting of large networks. *Communications of the ACM 5*, 11 (1962), 558–562.

[KH06] Karpenko O. A., Hughes J. F.: SmoothSketch: 3D free-form shapes from complex sketches. *ACM Transactions on Graphics 25*, 3 (2006), 589–598.

[Koe98] Koenderink J. J.: Pictorial relief. *Philosophical Transactions of the Royal Society of London 356*, 1740 (1998), 1071–1086.

[KvDK96] Koenderink J. J., van Doorn A. J., Kappers A. M. L.: Pictorial surface attitude and local depth comparisons. *Perception & Psychophysics 58*, 2 (1996), 163–173.

[LCD06] Luft T., Colditz C., Deussen O.: Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics 25*, 3 (2006), 1206–1213.

[LFG08] Lee S., Feng D., Gooch B.: Automatic construction of 3D models from architectural line drawings. In *Proceedings of Symposium on Interactive 3D Graphics* (2008), pp. 123–130.

[LMY*13] Liu X., Mao X., Yang X., Zhang L., Wong T.-T.: Stereoscopizing cel animations. *ACM Transactions on Graphics 32*, 6 (2013), 223.

[LS96] Lipson H., Shpitalni M.: Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design 28*, 8 (1996), 651–663.

[Mal86] Malik J.: *Interpreting Line Drawings of Curved Objects.* PhD thesis, Stanford University, 1986.

[NISA07] Nealen A., Igarashi T., Sorkine O., Alexa M.: FiberMesh: Designing freeform surfaces with 3D curves. *ACM Transactions on Graphics 26*, 3 (2007).

[OCN04] Ono Y., Chen B.-Y., Nishita T.: 3D character model creation from cel animation. In *Proceedings of International Conference on Cyberworlds* (2004), pp. 210–215.

[OSJ11] Olsen L., Samavati F., Jorge J. A.: NaturaSketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31*, 6 (2011), 24–34.

[SBv05] Sýkora D., Buriánek J., Žára J.: Sketching cartoons by example. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005), pp. 27–34.

[SDC09a] Sýkora D., Dingliana J., Collins S.: As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering* (2009), pp. 25–33.

[SDC09b] Sýkora D., Dingliana J., Collins S.: LazyBrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum 28*, 2 (2009), 599–608.

[SKv*14] Sýkora D., Kavan L., Čadík M., Jamriška O., Jacobson A., Whited B., Simmons M., Sorkine-Hornung O.: Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transaction on Graphics 33*, 2 (2014), 16.

[SSJ*10] Sýkora D., Sedlacek D., Jinchao S., Dingliana J., Collins S.: Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum 29*, 2 (2010), 615–623.

[VM02] Varley P. A. C., Martin R. R.: Estimating depth from line drawings. In *Proceedings of ACM Symposium on Modeling and Application* (2002), pp. 180–191.

[WCLT09] Wang Y., Chen Y., Liu J. Z., Tang X.: 3D reconstruction of curved objects from single 2D line drawings. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 1834–1841.

Acknowledgments

Additional information about algorithms presented in this talk can be found in the original publications [SSJ*10, SKv*14]. I would like to thank co-authors of the aforementioned papers in alphabetic order: Martin Čadík (Brno University of Technology), Steve Collins (Trinity College Dublin), John Dingliana (Trinity College Dublin), Alec Jacobson (ETH Zurich), Ondřej Jamriška (CTU in Prague), Ladislav Kavan (University of Pennsylvania), David Sedláček (CTU in Prague), Maryann Simmons (Walt Disney Animation Studios), Olga Sorkine-Hornung (ETH Zurich), Jinchao Sun (Trinity College Dublin), and Brian Whited (Walt Disney Animation Studios).

Ing. Daniel Sýkora, Ph.D.

Address:
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo nám. 13
12135 Prague 2
Czech Republic

E-mail:
sykorad@fel.cvut.cz

WWW:
http://dcgi.fel.cvut.cz/home/sykorad

Education:

- 2003–2007: Ph.D. in Informatics and Computer Scienece, Doctoral Programme at the Czech Technical University in Prague, Outstanding Dissertation Thesis Award
- 1997–2003: M.Sc. at Czech Technical University in Prague, Faculty of Electrical Engineering, Computer Engineering, ACM Student Master Thesis Award

Employment:

- 2010–present: Assistant Professor, Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague
- 2010: Graduate Associate, Walt Disney Animation Studios
- 2008–2010: Marie Curie Research Fellow, School of Computer Science and Statistics, Trinity College Dublin
- 2006–2008: Research Fellow, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague

Consultancy:

- 2012–present: Universal Production Partners, a.s.

- 2011–present: Disney Research Zurich
- 2010–present: Walt Disney Animation Studios
- 2008–present: AniFilm, s.r.o.
- 2001–2008: Digital Media Production, a.s.

Journal Publications:

1. D. Sýkora, L. Kavan, M. Čadík, O. Jamriška, A. Jacobson, B. Whited, M. Simmons, and O. Sorkine-Hornung: Ink-and-Ray: Bas-Relief Meshes for Adding Global Illumination Effects to Hand-Drawn Characters. In *ACM Transactions on Graphics* 33(2):16, 2014.
2. 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. In *ACM Transactions on Graphics* 32(4):116, 2013.
3. G. Noris, D. Sýkora, A. Shamir, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, and R. Sumner: Smart Scribbles for Sketch Segmentation. In *Computer Graphics Forum* 31(8):2516–2527, 2012.
4. D. Sýkora, D. Sedláček, S. Jinchao, J. Dingliana, and S. Collins: Adding Depth to Cartoons Using Sparse Depth (In)equalities. In *Computer Graphics Forum* 29(2):615–623, 2010, Günter Enderle Best Paper Award.
5. D. Sýkora, J. Dingliana, and S. Collins: LazyBrush: Flexible Painting Tool for Hand-drawn Cartoons. In *Computer Graphics Forum* 28(2):599–608, 2009.
6. J. Obert, J. Křivánek, F. Pellacini, D. Sýkora, and S. Pattanaik: iCheat: A Representation for Artistic Control of Indirect Cinematic Lighting. In *Computer Graphics Forum* 27(4):1217–1223, 2008.
7. D. Sýkora, J. Buriánek, and J. Žára: Colorization of Black-and-White Cartoons. In *Image and Vision Computing*, 23(9):767–852, 2005.

Conference Publications:

1. O. Jamriška, D. Sýkora, A. Hornung: Cache-efficient Graph Cuts on Structured Grids. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 3673–3680, 2012.

2. D. Sýkora, M. Ben-Chen, M. Čadík, B. Whited, and M. Simmons: TexToons: Practical Texture Mapping for Hand-drawn Cartoon Animations. In *Proceedings of the International Symposium on Non-photorealistic Animation and Rendering*, pp. 75–83, 2011, Best Paper Award.

3. G. Noris, D. Sýkora, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, and R. Sumner: Temporal Noise Control for Sketchy Animation. In *Proceedings of the International Symposium on Non-photorealistic Animation and Rendering*, pp. 93–98, 2011.

4. D. Sýkora, J. Dingliana, and S. Collins: As-Rigid-As-Possible Image Registration for Hand-drawn Cartoon Animations. In *Proceedings of the International Symposium on Non-photorealistic Animation and Rendering*, pp. 25–33, 2009.

5. D. Sýkora, D. Sedláček, and K. Riege: Real-time Color Ball Tracking for Augmented Reality. In *Proceedings of the Eurographics Symposium on Virtual Environments*, pp. 9–16, 2008.

6. D. Sýkora, J. Buriánek, and J. Žára: Video Codec for Classical Cartoon Animations with Hardware Accelerated Playback. In *Proceedings of the International Symposium on Visual Computing*, pp. 43–50, 2005.

7. D. Sýkora, J. Buriánek, and J. Žára: Sketching Cartoons by Example. In *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 27–34, 2005.

8. D. Sýkora, J. Buriánek, and J. Žára: Unsupervised Colorization of Black-and-White Cartoons. In *Proceedings of the International Symposium on Non-photorealistic Animation and Rendering*, pp. 121–127, 2004.

9. D. Sýkora, J. Buriánek, and J. Žára: Segmentation of Black-and-White Cartoons. In *Proceedings of the Spring Conference on Computer Graphics*, pp. 233–230, 2003.

Patents:

1. O. Jamriška and D. Sýkora: Optimizing Computation of Minimum Cut in Graphs with Grid Topology. U.S. Patent No. 8,533,139, 2013.

Professional Research Services:

- Member of International Program Committee for Expressive (20013–2014), NPAR (2010–2012), IPR (2012), EGIRL (2009)
- Member of Organization Committee for Eurographics 2007
- Reviewing for SIGGRAPH (2006, 2008–2011, 2013–2014), SIGGRAPH Asia (2009–2010, 2012), Eurographics (2005–2007, 2010–2011, 2014), TOG (2012), TVCG (2010–2012), CGF (2009–2010, 2014), CG&A (2011), C&G (2012), JVCI (2013), JEI (2011–2012), STSP (2011), PG (2009–2010, 2013), Expressive (2013–2014), NPAR (2010–2012), IPR (2012), ISVC (2009), SCCG (2006–2007, 2010, 2012), EGIRL (2009), VIE (2006), VIIP (2006), Web3D (2010), WSCG (2007), CESCG (2006)