Ink-and-Ray: Bas-Relief Meshes for Adding Global Illumination Effects to Hand-Drawn Characters

DANIEL SÝKORA
CTU in Prague, FEE
and
LADISLAV KAVAN
University of Pennsylvania / ETH Zurich
and
MARTIN ČADÍK
MPI Informatik / CTU in Prague, FEE / Brno University of Technology
and
ONDŘEJ JAMRIŠKA
CTU in Prague, FEE
and
ALEC JACOBSON
ETH Zurich
and
BRIAN WHITED and MARYANN SIMMONS
Walt Disney Animation Studios
and
OLGA SORKINE-HORNUNG
ETH Zurich

We present a new approach for generating global illumination renderings of hand-drawn characters using only a small set of simple annotations. Our system exploits the concept of bas-relief sculptures, making it possible to generate 3D proxies suitable for rendering without requiring side-views or extensive user input. We formulate an optimization process that automatically constructs approximate geometry sufficient to evoke the impression of a consistent 3D shape. The resulting renders provide the richer stylization capabilities of 3D global illumination while still retaining the 2D hand-drawn look-and-feel. We demonstrate our approach on a varied set of hand-drawn images and animations, showing that even in comparison to ground-truth renderings of full 3D objects, our bas-relief approximation is able to produce convincing global illumination effects, including self-shadowing, glossy reflections, and diffuse color bleeding.

1. INTRODUCTION

Despite the recent success and proliferation of 3D computer-generated imagery, traditional 2D hand-drawn animation is still a popular medium for animated films. Even in modern 3D production pipelines, 2D animation plays an important role during preproduction (such as in the story, layout, and animatic phases). The key
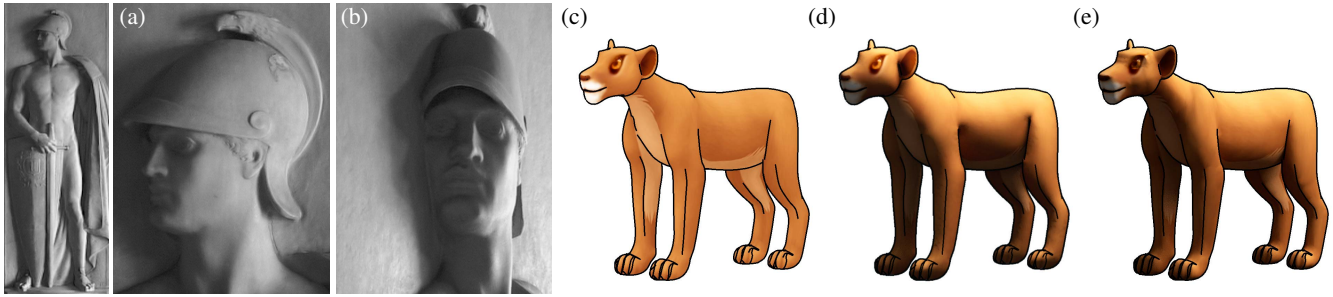
Fig. 1. Bas-relief sculpture—(a) from a frontal view the sculpture appears like a full 3D object, (b) while a side view reveals unreal depth proportions [Belhumeur et al. 1999]. (c) Input 2D character for which global illumination is computed using (d) our bas-relief like approximation and (e) the full 3D model. (Photographs © Kluwer Academic Publishers. All rights reserved.)

advantage of the 2D drawing metaphor is that it provides complete artistic freedom without the restrictions imposed by a CG environment (e.g., limited flexibility due to the 3D model and its animation rig). However, those very limitations are what make complex effects such as global illumination feasible in the computer-generated medium. Such effects are extremely labor-intensive to create for every frame of a traditionally animated film.

In this paper we introduce a new approach to the production of CG images which preserves the look-and-feel and artistic advantages of the 2D domain but at the same time delivers a look similar to that produced by a complex full 3D pipeline. Our technique is inspired by bas-relief sculptures [Read 1961] which demonstrate that fully consistent 3D models are not necessary for an appealing and convincing depiction of 3D shapes (Fig. 1a–b). The phenomenon behind this observation is known as the *bas-relief ambiguity* [Belhumeur et al. 1999] which states that an orthographic projection of a 3D object under global illumination is ambiguous under stretching or shearing transformations in depth. This is true for the lit areas of the surface, as well as cast and attached shadows. In other words the bas-relief ambiguity says that the knowledge of accurate depth values is not necessary to produce believable advanced illumination effects.

Instead, as one can observe from bas-relief sculptures, the models need to contain explicit, though possibly simplified, surface geometry (as opposed to a normal field only). In addition, the representation needs to capture consistent depth order, and appropriate separation, contact and continuity between individual surface components. Explicit geometry, ordering and separation are necessary features to capture effects produced by light bouncing between surface components, passing through gaps between surfaces, and being occluded by surfaces. Without appropriate contact, light will leak through components that are supposed to be connected/coincident and continuity is needed to control whether the surface interacts with light as a smoothly connected or discontinuous component. See Fig. 3 for an illustration of the importance of individual requirements. Our novel *ink-and-ray* framework facilitates fast creation of proxies containing the above features which we demonstrate are accurate enough to produce 3D global illumination effects including complex self shadowing, glossy reflections, and color bleeding.

To enable the proposed concept several new technical contributions have been made. A key part is the stitching phase where individual inflated layers are joined together in a prescribed relative depth order. Here a new Dirichlet type of energy with inequality constraints followed by C^1–continuous biharmonic smoothing is formulated and solved. In addition, we present an approach for

relative depth order estimation based on the concept of illusory surfaces [Geiger et al. 1998] solved via diffusion curves [Orzan et al. 2008] and a new Poisson-based formulation of the layer inflation process.

2. RELATED WORK

The bas-relief ambiguity was studied only with Lambertian reflectance, however, similar observations about the lower sensitivity of the human visual system to inconsistencies in illumination of 3D objects have been made in more complex scenarios [Ostrovsky et al. 2005]. Khan et al. [2006] demonstrated the practical utility of this phenomenon in their framework for editing materials in natural images. They reconstruct the approximate geometry of 3D objects using simple filtering of image luminance and a rough estimation of environment lighting in order to produce renderings of the objects with different materials. A similar workflow is used by Lopez-Moreno et al. [2010] to create stylized depictions of images. This type of approach is not applicable to hand-drawn characters lacking reliable shading information from which to extract depth information.

Another popular technique for approximating 3D-like shading without leaving the 2D domain is the use of estimated normal fields. *Lumo* [Johnston 2002] generates such an approximation from a line drawing by interpolating the normals on region boundaries. Extensions to the base Lumo approach [Okabe et al. 2006; Winnemöller et al. 2009; Shao et al. 2012] obtain better control over the values in the generated normal field. However, due to the lack of depth information, more complex global illumination effects such as reflections, color bleeding, and self shadowing are not supported.

Toler-Franklin et al. [2007] use curvature and directional discontinuities of estimated normal fields to emulate effects such as ambient occlusion and local self-shadowing. Similarly in *Tex-Toons* [Sýkora et al. 2011], depth layering is used to enhance textured images with ambient occlusion, shading, and texture rounding effects. Recently Vergne et al. [2012] presented a new approach to simulate complex lighting effects where a user-defined flow field deforms the input image so that it appears to be reflected from a specific surface. These methods produce a 3D-like look, however the overall appearance can still feel synthetic due to the lack of subtle global illumination effects which require some approximation of a 3D model.

In general, approaches for constructing 3D models from 2D input can be classified into two main groups: methods focused on creating fully consistent 3D models and methods producing 2.5D approximations such as height fields.
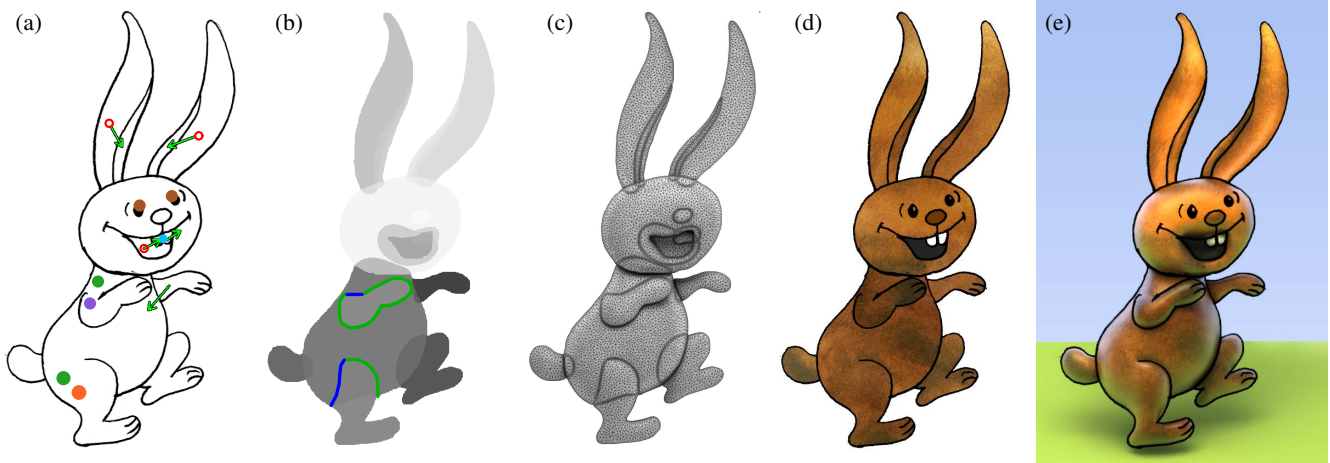
Fig. 2.   Ink-and-ray pipeline—(a) Hand-drawn image with user-specified annotations to guide segmentation and depth ordering. (b) Regions shaded according to depth order including estimated occluded silhouettes and grafting boundary conditions shown in blue and green. (c) Proxy 3D mesh. (d) Original drawing used as a texture in render. (e) Global illumination render. (Source drawing © Anifilm. All rights reserved.)

Reconstruction of 3D models from line drawings is a central challenge in the field of computer vision [Malik 1986]. In general, the problem is highly under-constrained, and additional knowledge is typically required to obtain reasonable results automatically [Wang et al. 2009; Cole et al. 2012]. The situation becomes even more complicated when the input is man-made drawings or sketches. In this scenario, important geometric rules are broken due to inaccuracies or deliberate stylization, and thus manual intervention becomes inevitable, making the process more akin to modeling from scratch than reconstruction.

With sketch-based approaches for creating full 3D models, the user often traces curves over the original image or sketches regions directly with a mouse or a tablet. A concept of surface inflation [Igarashi et al. 1999] enhanced by intelligent topological embedding [Karpenko and Hughes 2006], additional control curves [Nealen et al. 2007], or replaced by geometric primitives [Gingold et al. 2009], is then used to produce an initial approximation of the 3D object. The user can add more shapes, make connections, and arrange them in the 3D space by working with side views. Some additional annotations [Gingold et al. 2009; Olsen et al. 2011] can be used to improve the shape's structure and topology. Similarly, other sketch-based methods have been proposed for producing 2.5D approximations instead of full 3D surfaces [Ono et al. 2004; Chen et al. 2005; Joshi and Carr 2008; Andrews et al. 2011]. A common drawback of these approaches is that they require tedious specification of control curves with positional and directional constraints to produce the desired results. Moreover, they typically assume the resulting surface is a height field which inherently limits the range of illumination effects (e.g., light cannot pass through holes in the object).

Sýkora et al. [2010] presented a depth assignment framework for hand-drawn images utilizing simple user-specified annotations to produce flat height fields with smooth depth transitions that can be further inflated using a reconstructed normal field. Although this approach shares interaction simplicity and produces approximations close to our bas-relief structure, the resulting surface is only a 2.5D height field with arbitrary discontinuities. As such, subsequent inflation may not respect the prescribed depth order and can easily lead to large inconsistencies such as self-intersections.

The approach of Rivers et al. [2010] allows quick creation of 2.5D models. While the result supports 3D rotation, it does not produce 3D surfaces which are necessary for simulating global illumination effects. Petrović et al. [2000] show that a simple approximation of a 3D model is sufficient for generating believable cast shadows for cel animation. Their technique bears some resemblance to our approach, but requires extensive manual intervention and working with side views to obtain the desired results.

Although some of the previous approaches could be used to create proxy 3D models for rendering, our new ink-and-ray pipeline greatly simplifies the process. The bas-relief mesh contains enough 3D information to produce global illumination effects without requiring absolute depth proportions. This enables artists to work entirely in the 2D domain. Based on a few simple annotations, the system automatically creates a stack of inflated layers which preserve relative depth order, contact, and continuity, without requiring image tracing or tedious specification of control curves.

3.   INK-AND-RAY PIPELINE

In our proposed ink-and-ray pipeline, the process of creating a bas-relief-type mesh from a hand-drawn image consists of six main steps: segmentation, completion, layering, inflation, stitching, and grafting. In this section we give an overview and provide motivation for each step.

Segmentation In the segmentation phase, the image is partitioned into a set of regions which preserve the individual components delineated by outlines in the original image (see Fig. 6b). Without segmentation, the resulting global illumination render would omit important details, producing only a simplistic balloon-like appearance (see Fig. 3a).

Completion When an extracted region includes occluded silhouettes, the hidden parts are estimated using shape completion (see Fig. 6d). This completion aids in the layering and inflation phases (see Fig. 3b and Fig. 12). It also helps to produce correct cast shadows for occluded parts (see Fig. 4).

Layering Based on the shape completion, a relative depth order can be assigned to the regions, producing a stack of layers (see Fig. 6c). Layering has a direct impact on the quality of the resulting illumination – it helps to keep the overall structure con-
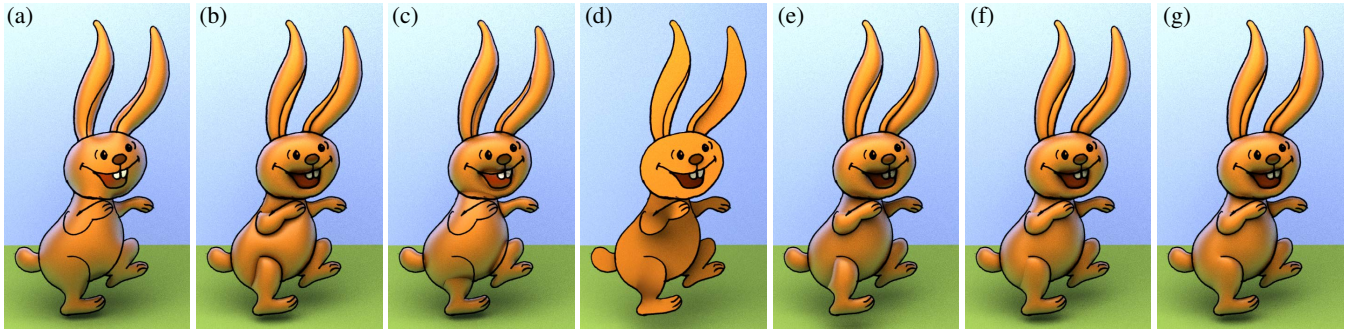
Fig. 3. Influence of the individual steps of the ink-and-ray pipeline on the resulting global illumination render (g). In each example above, one of the steps is omitted: (a) segmentation – geometric details are missing, (b) completion – occlusion destroys shape consistency, (c) layering – inflated segments penetrate, (d) inflation – lack of volume, (e) stitching – segments hover in space, (f) grafting – $\mathcal{C}^1$ discontinuities. (Source drawing © Anifilm. All rights reserved.)
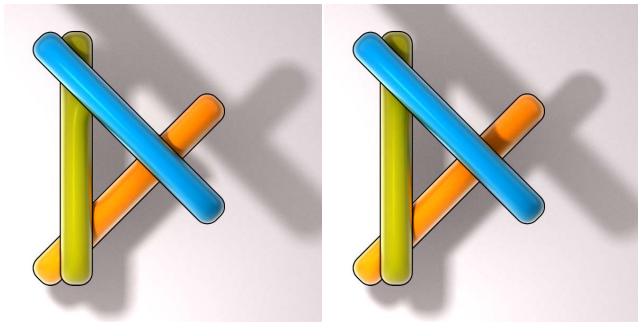

Fig. 4. Artifacts can appear when the shape of an occluded part is not completed (left). With completion, occluded parts of the reconstructed mesh are correctly reflected in the cast shadows (right).
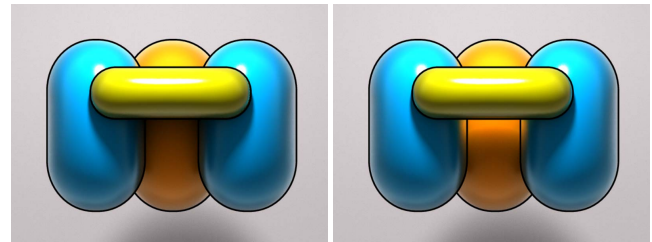

Fig. 5. Approaches that use representations based on height fields (left) can not represent holes. Our layering approach preserves plausible depth proportions and supports holes (right).

sistent by avoiding surface intersections during the stitching phase (see Fig. 3c) and allows holes which are important for plausible light transport (see Fig. 5).

Inflation Each region is inflated to produce a 3D surface with volume (see Fig. 2c). This enables direct illumination effects such as diffuse shading and glossy highlights as well as global illumination effects arising from the light interaction between objects. Without inflation the resulting render would look like a scene from a pop-up book (see Fig. 3d).

Stitching Adjacent inflated regions are stitched together at contours according to their relative depth order (see Fig. 13a–b). This influences global illumination effects such as self-shadowing, color bleeding, and glossy reflections. Without stitching, the resulting render would reveal missing contacts between individual parts (see Fig. 3e).

Grafting Finally, where the outline is missing in the original image, grafting replaces unwanted $\mathcal{C}^0$ contours with smooth $\mathcal{C}^1$ transitions (see Fig. 13c–d). This step is important in preserving the smoothness of direct illumination, preventing visible discontinuities (see Fig. 3f).

The resulting proxy can then be textured and rendered.

4. ALGORITHM

In this section, each step of the proposed ink-and-ray pipeline outlined above is described in more detail.

4.1 Segmentation

The input to our system is a scanned hand-drawn image or a digital sketch (Fig. 6a). The first step partitions the drawing into a set of regions, each of which should correspond to a logical structural component of the hypothetical 3D object depicted in the drawing (Fig. 6b).

The original image will be used as a texture during the rendering phase, and therefore it is not necessary for the segmentation to be carried out to the finest detail level (e.g., eyes can be part of the head region and the teeth together can form a single region as in Fig. 6b). On the other hand, it is important to separate out regions that represent articulated parts (for example, the limbs in Fig. 6), and in some cases to join parts broken into multiple components due to occlusion (e.g., the ear in Fig. 6 or the wolf's tail in Fig. 26).

For "clean" drawings, where regions are distinctly separated by closed outlines, the base partitioning can be done without user intervention [Sýkora et al. 2005]. In cases where this is not sufficient, the user can then perform detail reduction and region joining (see selection strokes in Fig. 6a). For articulations and more complex drawings containing rough strokes with gaps, scribble-based segmentation tools tailored to scanned images (e.g., Lazy-Brush [Sýkora et al. 2009]) or digital sketches (e.g., SmartScribbles [Noris et al. 2012]) can be used.

When separating articulations, the *LazyBrush* algorithm may produce artifacts (e.g., the squaring off of the top of the arm in Fig. 6b). This problem typically does not affect the final geometry as the region will be smoothly grafted to the base mesh. However, for cases where the shape is important, the silhouette completion mechanism described in the following section can be utilized (see leg in Fig. 6d).
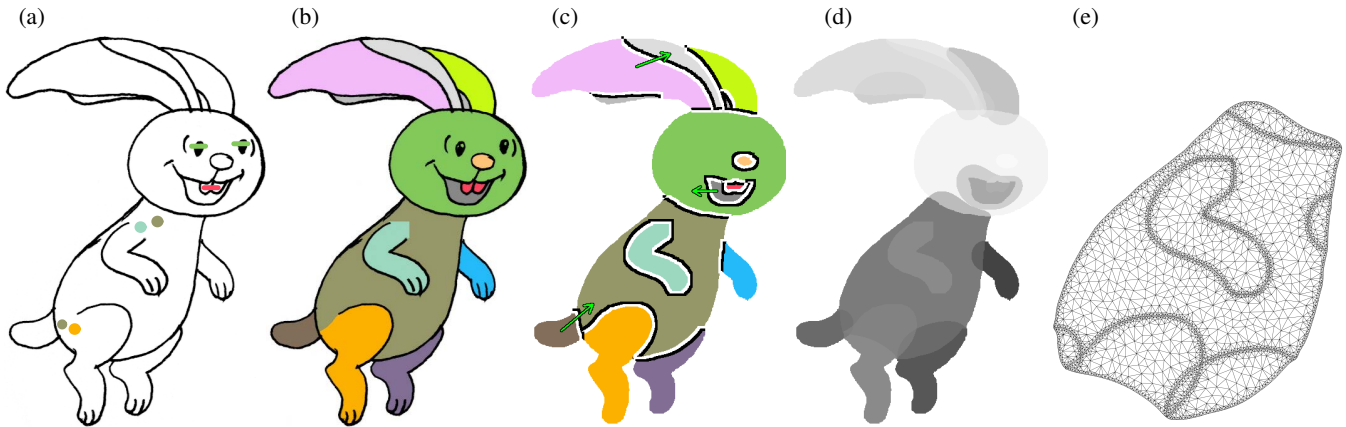
Fig. 6. Converting an input image into a set of regions with depth order—(a) the input image with user-specified annotations for region identification, (b) resulting segmentation, (c) estimation of relative depth order (black pixels mark the side of the region boundary with greater depth value, white pixels lie on the side which is relatively closer), together with relative depth corrections specified by the user (green arrows), (d) regions with depth order and illusory silhouettes in occluded parts (lighter portions are closer), (e) conforming constrained Delaunay triangulation with Steiner points at boundaries of intersecting regions. (Source drawing © Anifilm. All rights reserved.)

4.2 Completion

Once the image is partitioned into a set of regions, the next step is to complete silhouettes where hand-drawn outlines are missing in the original image either due to occlusion or articulation. The completion approach described below is inspired by techniques for reconstruction of illusory surfaces [Geiger et al. 1998].

Consider the problem of estimating the illusory surface in the classical Kanizsa square example (Fig. 7a). The boundary of the illusory square at the corner pixels is known and therefore a hypothesis can be initialized where these pixels are set appropriately to be inside ($p = 1$, white in Fig. 7b) or outside ($p = 0$, black in Fig. 7b) the square. As in Geiger et al. [1998], a diffusion process similar to that used in diffusion curves [Orzan et al. 2008] (Fig. 7c) can be applied to propagate these probabilities into the unknown pixels (gray in Fig. 7b). Finally pixels with probabilities $p > 0.5$ are classified to lie inside the illusory surface (Fig. 7d).
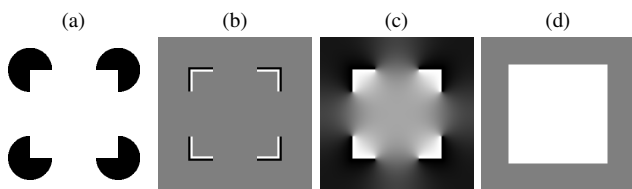

Fig. 7. Estimation of illusory surface—(a) Kanizsa square, (b) initial hypothesis: white pixels $p = 1$ are inside, black $p = 0$ are outside, gray are unknown, (c) diffusion of p into unknown pixels, (d) pixels with $p > 0.5$ lie inside the illusory surface.

4.3 Layering

The silhouette completion mechanism is used in the layering step to predict the relative depth order between regions.

Given two regions A and B (Fig. 8a), we would like to test if region B is occluded by region A. This can be determined by com-
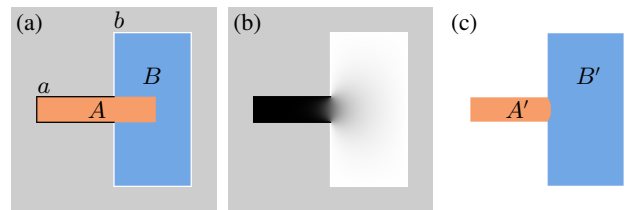

Fig. 8. Inferring relative depth order—(a) an illusory surface B' of the region B is estimated using two diffusion curves a and b with values $p_a = 0$ black and $p_b = 1$ white, (b) p after the diffusion, (c) pixels which belong to B' have $p > 0.5$. As the area of B' is bigger than area of B, we can conclude region B was occluded by region A.

puting B's illusory surface B' (Fig. 8c) and checking whether the area of B' is greater than the area of B (Fig. 8c).

The illusory surface B' is created by first constructing two diffusion curves [Orzan et al. 2008]: a with value $p_a = 0$ and b with $p_b = 1$ (Fig. 8a), which represent silhouette pixels of the corresponding regions, i.e., all boundary pixels of the region that do not touch the other region. These values are then diffused into the compound area of both regions (Fig. 8b). Interior pixels with $p > 0.5$ then correspond to the illusory surface B'.

This process can be applied to each pair of neighboring regions (see resulting predicted depth order in Fig. 6c). The user can then correct possible prediction errors using additional depth inequalities (arrows in Fig. 6c). A graph of relative inequalities is constructed [Sýkora et al. 2010] and topologically sorted to obtain absolute depth values for each region (Fig. 6d).

Once the relative depth order of regions is known, the silhouettes of occluded regions can be further refined. Given a region A that is known to be occluded by a region B (Fig. 9), the diffusion curve a can be modified to include the boundary of region B (Fig. 9a). The diffusion process is then applied (Fig. 9b) and pixels with probability p lower than 0.5 correspond to the refined illusory surface A' (Fig. 9c). B is known to be in front of A, and therefore its shape remains unchanged.
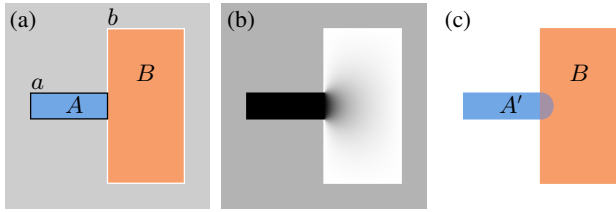
Fig. 9. Completion of occluded silhouettes—(a) an illusory surface A' of the region A occluded by the region B is estimated using two diffusion curves a and b with values $p_a = 0$ (black) and $p_b = 1$ (white), (b) p after the diffusion, (c) pixels which belong to A' have $p < 0.5$, the shape of the region B remains unchanged.

4.4 Inflation

Given a labeling of segmented regions $\Omega_i \subseteq \mathbb{R}^2$, where i is the region index, the goal of the inflation phase is to find functions $\tilde{f}_i : \Omega_i \to \mathbb{R}$ corresponding to a buckled shape. We find $\tilde{f}_i$ by solving a Poisson equation

$$-\nabla^2 \tilde{f}_i(\mathbf{x}) = c_i \quad \forall \mathbf{x} \in int(\Omega_i) \tag{1}$$

where $c_i > 0$ is a scalar specifying how much the inflated surface should buckle. Both Dirichlet and Neumann boundary constraints are supported on $\partial \Omega_i$:

$$\tilde{f}_i(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in B_D \tag{2}$$

$$\frac{\partial \tilde{f}_i}{\partial \mathbf{n}}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in B_N \tag{3}$$

where the disjoint sets $B_D \cup B_N = \partial \Omega_i$ describe where each type of boundary constraint applies. We solve this Poisson equation using the standard finite element discretization with piecewise linear shape functions, reducing the problem to a linear system solve. To produce this discretization, we employ boundary tracing [Ren et al. 2002] of Ω_i and apply conforming constrained Delaunay triangulation [Shewchuk 2002]. Additional Steiner points are added where silhouettes of other regions intersect the interior of Ω_i (Fig. 6e) and used during the stitching phase to impose constraints on coincident vertices.

The resulting $\tilde{f}_i$ produces a parabolic profile (Fig. 10a). If desired, the user can specify a custom cross-section function to convert the profile into an alternative shape. A typical example of such a function is $f_i(x) = d_i \sqrt{\tilde{f}_i(x)}$, where $d_i \in \mathbb{R}$ is a scaling factor which makes it possible to obtain spherical profiles (Fig. 10b). This can be applied, for example, when the user wants to produce a concave or flatter profile (see the ears and mouth in Fig. 2, where a lower negative d_i was used to simulate flatter concavity).

Fig. 10. Shape profiles—(a) initial parabolic profile produced by the Poisson equation, (b) spherical profile obtained by applying a cross-section function with square root.

The shape of the inflated region can further be modified by varying the boundary conditions B. Dirichlet boundary conditions are

used by default. To prevent rounding at places where the surface should remain straight, Neumann boundary conditions can be utilized (see the sleeve and lower part of the jacket in Fig. 11). To selectively change the type of boundary condition, the user clicks on two endpoints and the system automatically finds the path that connects the points along a boundary of the nearest region (see dots and curves in Fig. 11).
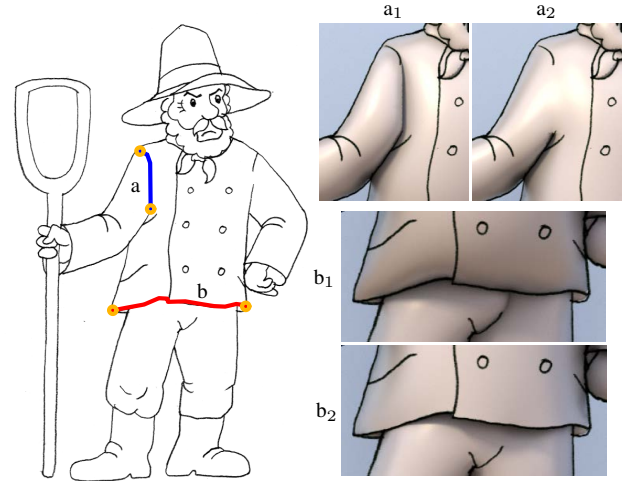

Fig. 11. Modifying boundary conditions—(a,b) the user clicks on endpoints (orange dots) and the system automatically finds the path that connects them along a boundary of the nearest region (red and blue curves). (a₁,b₁) The resulting shapes when the default settings (Dirichlet boundary conditions B_D, stitching equalities $C_{i,j}^=$, no grafting) are used. (a₂) Result when Neumann boundary conditions B_N are used together with grafting to produce $\mathcal{C}^1$-continuous stitching. (b₂) Result when Neumann boundary conditions B_N are used with stitching inequalities $C_{i,j}^{\geq}$ to avoid rounded boundaries. (Source drawing © Anifilm. All rights reserved.)

A similar approach to inflation was previously used in Tex-Toons [Sýkora et al. 2011]. Here normal interpolation (originally proposed in Lumo [Johnston 2002]) is reformulated based on solving a Laplace equation. A key drawback to this method is that normal estimation is required at region boundaries, which is not necessary in our Poisson-based solution. In TexToons, normal interpolation is further guided by Neumann boundary conditions at depth discontinuities. Our approach could utilize this technique, removing the need for our shape completion phase. However, if a large portion of the boundary has Neumann conditions, it can produce an undesirable inflation which does not respect the underlying structure (see Fig. 12).

4.5 Stitching

The shapes f_i obtained by inflation are all based at the same height $z = 0$ (see Fig. 13a). Note that we assume that Dirichlet boundary conditions are used at least at one point. To obtain a bas-relief-type structure, the inflated shapes need to be stitched together in a way that satisfies the relative depth order (see Fig. 13b). This requires translation in z and deformation of the initial shapes. We accomplish this by solving for functions $g_i : \Omega_i \to \mathbb{R}$ such that the sum $f_i + g_i$ satisfies *stitching constraints*. The stitching constraints can either be equality (specifying that two regions should exactly meet at given boundary points) or inequality (specifying that one region should be above/below another one).
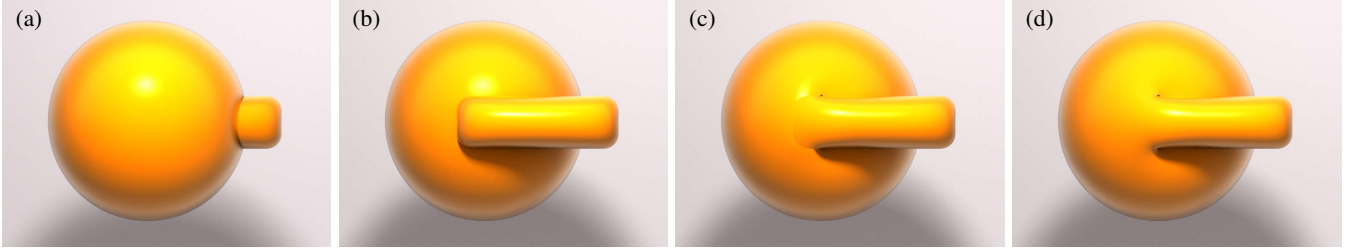
Fig. 13. Stitching & grafting—(a) Initial inflated shapes are all at the same depth. (b) Prescribed $C^=$ and $C^\geq$ constraints enforce correct absolute depth values. (c) Grafting merges meshes together, and (d) removes visible $\mathcal{C}^1$ discontinuities.
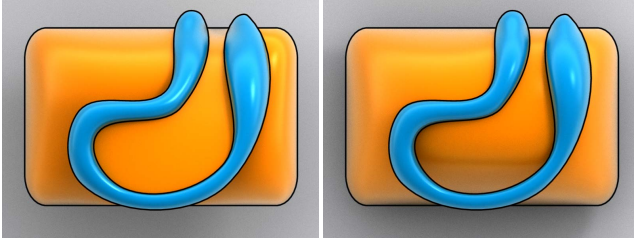

Fig. 12. If Neumann boundary conditions are used to guide the inflation at depth discontinuities, the resulting shape does not respect the underlying structure (left). A better result is obtained if the shape completion mechanism is applied prior to inflation of the occluded rectangle (right).

For two overlapping regions Ω_i, Ω_j (Fig. 14b), we define the sets $C^=_{i,j}, C^\leq_{i,j}, C^\geq_{i,j} \subseteq \partial\Omega_i \cup \partial\Omega_j$ which describe the type of applied constraints (Fig. 14c–d). Then the functions g_i are found by minimizing the Dirichlet energy:

$$\int_{\Omega_i} \|\nabla g_i\|^2 d\mathbf{x} \qquad (4)$$

subject to the stitching constraints:

$$f_i(\mathbf{x}) + g_i(\mathbf{x}) = f_j(\mathbf{x}) + g_j(\mathbf{x}) \quad \forall \mathbf{x} \in C^=_{i,j} \qquad (5)$$
$$f_i(\mathbf{x}) + g_i(\mathbf{x}) \leq f_j(\mathbf{x}) + g_j(\mathbf{x}) \quad \forall \mathbf{x} \in C^\leq_{i,j} \qquad (6)$$
$$f_i(\mathbf{x}) + g_i(\mathbf{x}) \geq f_j(\mathbf{x}) + g_j(\mathbf{x}) \quad \forall \mathbf{x} \in C^\geq_{i,j} \qquad (7)$$

Intuitively, this corresponds to seeking g_i which is as constant as possible while satisfying the constraints (Fig. 14e–f).

Finite element discretization of this problem is straightforward because, by construction, each constrained vertex is present in meshes corresponding to both Ω_i and Ω_j (Fig. 6e). The constraints can then be directly expressed as a system of linear equalities and inequalities, resulting in one quadratic program for all regions which is solved using MOSEK [Andersen and Andersen 2000].

4.6 Grafting

The $\mathcal{C}^0$ connections created during the stitching step are desirable for most parts of the proxy 3D mesh. However, for regions representing shape articulations, the user may prefer smooth $\mathcal{C}^1$ transitions (see Fig. 13c–d). In this case, a *grafting* step can be used to smoothly merge the meshes of individual regions together.

Grafting works with a subset of vertices $G_i \subseteq \partial\Omega_i$, which specify where the regions Ω_i will be smoothly connected. These *grafting vertices* can be detected automatically as boundary points that are not covered by a hand-drawn contour, or the user can use
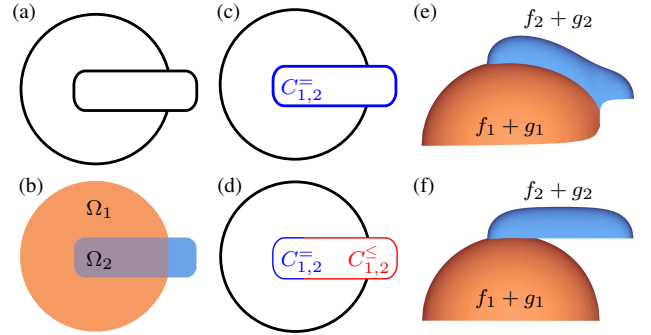

Fig. 14. Stitching—(a) an input drawing containing two regions Ω_1 and Ω_2 (b). When equality constraint $C^=_{1,2}$ is active on the whole $\partial\Omega_2$ (c), then $f_2 + g_2$ is glued together with $f_1 + g_1$ (e). When the equality constraint is replaced by inequality constraint $C^\leq_{1,2}$ on a subset of $\partial\Omega_2$ (d), the corresponding part of $f_2 + g_2$ will hover over $f_1 + g_1$. Side views are shown in images (e) and (f).

the aforementioned two-click interface to specify them manually (Fig. 11a). The regions are then processed and merged one by one in a bottom-top order. Given regions $\Omega_1, \dots, \Omega_{i-1}$ that have already been joined into region Υ, the task is to graft Ω_i (Fig. 15a). We search for vertices in Υ coincident (in x, y coordinates) with G_i, and reconnect the triangles in such a way that the visible top surface forms a connected mesh (Fig. 15b). Note that this process only changes the connectivity, no re-indexing of vertices is necessary.

After the grafting, smoothing is performed in two steps. First, a smoothing region S is determined (Fig. 15c) which ensures that only a local neighborhood of G_i is affected. This can be done using an approach similar to the illusory surface reconstruction performed during the layering phase (see Sec. 4.3). We apply harmonic regularization to a delta-type function:

$$\delta(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \notin G_i \\ 1 & \mathbf{x} \in G_i \end{cases} \qquad (8)$$

Specifically, we find $s : \Upsilon \to \mathbb{R}$ which minimizes

$$\int_{\Upsilon} \left((s - \delta)^2 + \alpha\|\nabla s\|^2 \right) d\mathbf{x} \qquad (9)$$

where $\alpha > 0$ is a scalar controlling the amount of diffusion. The smoothing region S is then defined as

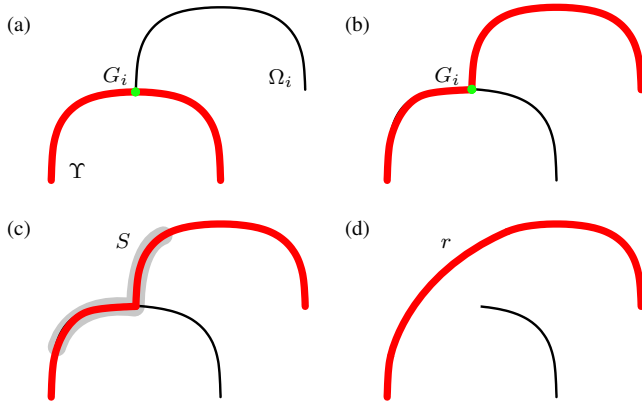$$S = \{\mathbf{x} \in \Upsilon : s(\mathbf{x}) > 0.5\} \qquad (10)$$

Fig. 15. Grafting—(a) region Ω_i is grafted on Υ at grafting vertices G_i, (b) triangles of Υ are reconnected to Ω_i at vertices coincident with G_i, (c) smoothing region S is computed, and (d) $\mathcal{C}^1$-continuous surface r is produced. For clarity sectional views are used to show the surface profile.

Once the smoothing region S is defined, the second step is to obtain a new smoothed surface r (Fig. 15d) by minimizing:

$$\int_{\Upsilon} (\Delta r)^2 d\mathbf{x} \qquad (11)$$

subject to:

$$r(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) \quad \forall \mathbf{x} \notin S \qquad (12)$$

i.e., replacing the values of the original function $f + g$ by a $\mathcal{C}^1$ biharmonic interpolant in the smoothing region S.

4.7 Texturing

Since only the z coordinates of the points on the mesh are modified, the $[x, y]$ coordinates can be directly used as texture coordinates $[u, v]$. Any property attached to the original image can therefore be directly transferred to the 3D proxy surface, e.g., flat colors assigned to regions [Sýkora et al. 2009] (Fig. 16a–b), or textures [Sýkora et al. 2011] with bump maps (Fig. 16d–e). The only limitation of this transfer is that occluded surfaces are assigned the same data as the occluding top layer. In most cases, this does not produce noticeable artifacts. However, in cases when correct coloring is important (e.g., due to color bleeding effects), one can employ the illusory surfaces computed during the layering phase. Data (e.g., flat colors or $[u, v]$ coordinates) can then be extended from the visible portion of the region to the occluded part using the extrapolation approach in TexToons [Sýkora et al. 2011] for filling small gaps caused by misalignment.

4.8 Rendering

Once the proxy mesh is generated, it can be rendered using an off-the-shelf global illumination engine or, for a more stylized look, simple flat shading alone can be computed with, for example, a cartoon shader (Fig. 16a). The final look can be customized by adding environment geometry with flat colors or textures (see ground and background plane in Fig. 16b–e), setting multiple light sources (Fig. 16b), specifying materials (e.g., glossy or diffuse in Fig. 16d–e), adding fur (Fig. 16c), etc. The only limitation is the use of an orthographic camera with optical axis perpendicular to the plane of the source image. Near perspective or different viewpoints may reveal the approximate bas-relief structure (see Fig. 28) and will

break the linear mapping between vertex coordinates and the original image space. This mapping is crucial for the post-processing phase where the reduced details and hand-drawn look are restored by superimposing the input image as a texture in the final render (Fig. 16).

5. RESULTS

We have implemented the ink-and-ray prototype using a combination of C++ and Matlab. Our prototype utilizes the following packages: the fast max-flow/min-cut solver GridCut [Jamriška et al. 2012] for the LazyBrush segmentation, a fast Laplacian solver [Jeschke et al. 2009] for computing diffusion curves, and the Triangle package [Shewchuk 1996] for triangulation. In Matlab, the backslash operator was used to solve linear systems and MOSEK [Andersen and Andersen 2000] for quadratic programs. The whole process from input image to output mesh takes less than 7 seconds for the *bunny* scene in Fig. 2 (0.5Mpix, 32k triangles) on a Xeon 3.5GHz with 16GB RAM and GeForce GTX 660. See the following table for timing breakdown of the individual steps:

segmentation	0.2 sec
completion & layering	0.4 sec
triangulation	0.9 sec
inflation & stitching	2.6 sec
grafting & smoothing	2.7 sec
total	6.8 sec

To render the image, we implemented a progressive path tracer with Monte-Carlo integration [Kajiya 1986] based on the OptiX ray tracing engine [Parker et al. 2010]. Although the current performance of the whole prototype pipeline is not fully interactive, it still provides reasonable enough response, making incremental edits feasible. In future work we plan to employ parallel processing and greater GPU utilization to deliver truly interactive performance.

To demonstrate the versatility of the proposed method, we selected a set of scanned hand-drawn images in various drawing styles, with different poses and structural complexity (see Fig. 2, Fig. 26, and Fig. 27). In each example, we show the original hand-drawn image, the result of the completion and layering phases, the generated proxy mesh, texture used for rendering, and the final global illumination render. The original and layered images are superimposed with the annotations made by the user (segmentation scribbles, depth ordering correction arrows, and dots specifying boundary conditions).

An example of the annotation workflow is shown in Fig. 27. The initial segmentation is either established automatically (see Fig. 2b and *robber* in Fig. 26b) or a couple of rough scribbles are drawn to help the system extract important regions (see Fig. 27c). The system next estimates the relative depth order of extracted regions and produces the initial bas-relief-type approximation. If depth ordering artifacts are present in the rendering of this initial approximation (Fig. 27d), additional depth inequalities can be specified (Fig. 27e) to correct them. Grafting and $C^{\geq}$ boundary conditions are detected automatically (see Fig. 2b and *wolf* in Fig. 26b) or specified by the user (Fig. 27g) to improve on the results of the default inflation and stitching operations (Fig. 27f). Using this type of annotation workflow, a trained user was able to specify the annotations in less than a minute for each frame. No tedious manual
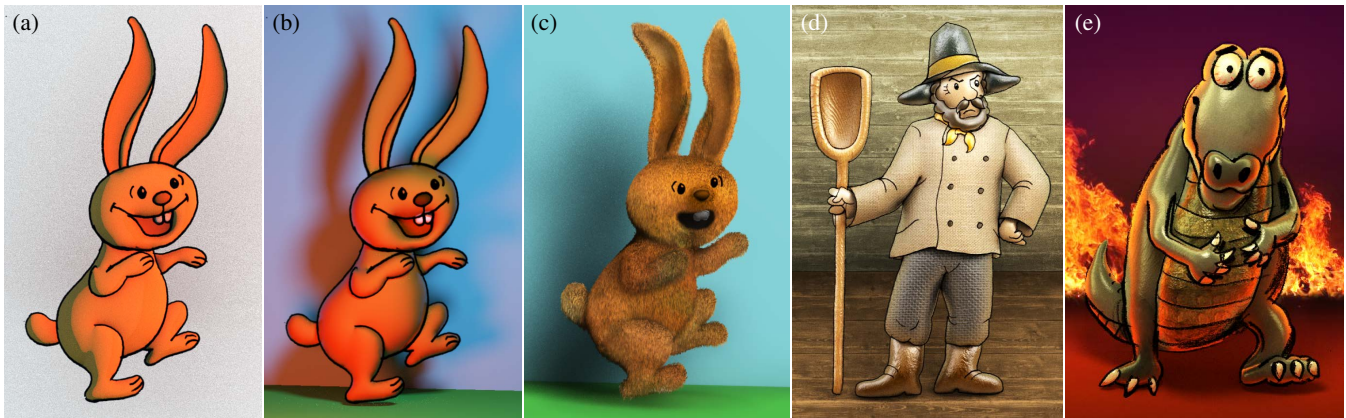
Fig. 16. The generated proxy mesh can be rendered using an off-the-self physically-based renderer (b–e). (a) One can also employ a stylized shader to simulate cartoon-like shading. Global illumination examples include: (b) flat colors, (c) fur, and (d–e) textures with bump maps applied on the mesh. Multiple light sources (b) as well as diffuse (d) or glossy (e) materials can be used as in a standard rendering pipeline. (Source drawings © Anifilm, textures © CGTextures. All rights reserved.)

orthographic   **perspective**
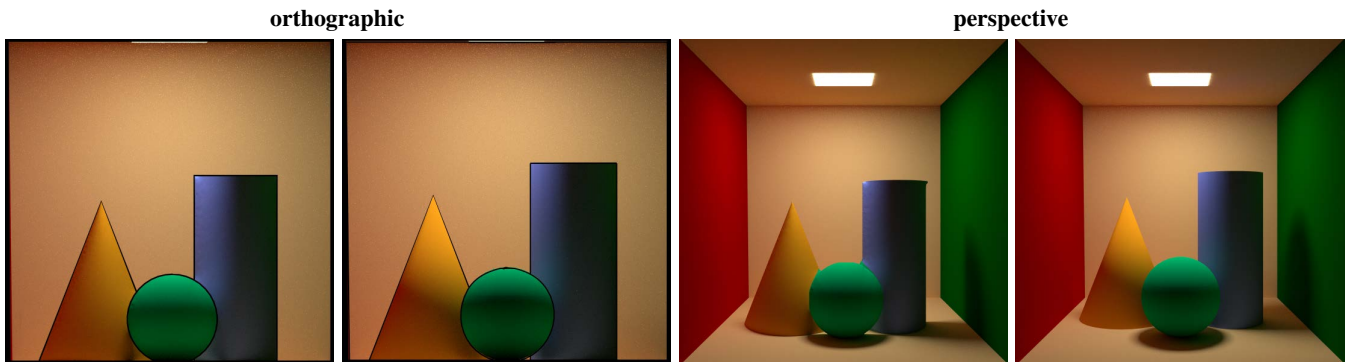

Fig. 17. A simple Cornell box—global illumination computed using our approximation (left) and full 3D model (right). Differences are less noticeable when orthographic projection is used.

labor such as rotoscoping was required. In contrast, the creation of similar 3D models using standard modeling tools can take hours.

The final render produces believable illumination despite the simplifications in the generated proxy mesh (see tilted views in Fig. 28). To show that our bas-relief approximation is able to deliver comparable render quality to that of a complete 3D model, we set up a simple Cornell box scene where a set of 3D objects (sphere, cone, cylinder) occluding each other is illuminated by an area light source (Fig. 17). We rendered visible outlines in orthographic projection using a simple cartoon shader and applied our method using the resulting 2D image as input. A global illumination render from the same viewpoint was run for the full 3D model and for our proxy mesh. The resulting images look similar under orthographic projection. Similar illumination effects are produced even when rendered with a perspective camera, but in this case the approximation does cause some artifacts, most notably in the attached shadows. To further test the quality of our results against a rendering of a full 3D model, we performed a similar ground-truth test on a cartoon model of a lion (Fig. 1c–e) and conducted a perceptual experiment (see Sec. 6).

Our method can be applied to individual drawings as well as sequences of animated frames. We use the concept of *scribble transfer* described in [Sýkora et al. 2009] to reuse annotations in subse-

Fig. 19. Comparison with previous approaches—(a) normal map based shading used in Lumo [Johnston 2002], (b) simulation of 3D-like shading used in TexToons [Sýkora et al. 2011], (c) our approach. (Source drawing © Anifilm. All rights reserved.)
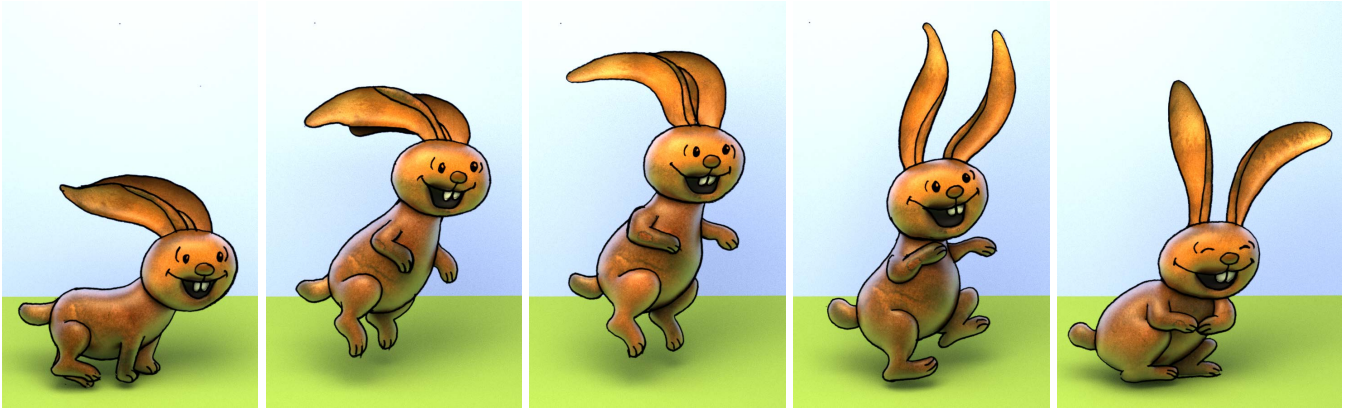
Fig. 18.   Example animation sequence created using the ink-and-ray framework. Annotations from Fig. 2a were transferred to the remaining frames using as-rigid-as-possible image registration [Sýkora et al. 2009] (each region was registered separately to handle topology changes) and the TexToons [Sýkora et al. 2011] framework was used to transfer texture coordinates from Fig. 2b. (Source drawings © Anifilm. All rights reserved.)

quent frames. Our method was then used to produce a proxy mesh for each frame. Although our approach does not address temporal coherency explicitly, the shading produced by the bas-relief-type approximation generally appears consistent over animated sequences (see Fig. 18). One reason for the lack of noticeable temporal artifacts is the typically lower frame rate (12 frames per second) found in many hand-drawn animations. The lower frame rate (1) reduces manual effort and (2) allows greater stylization, i.e., larger structural changes between individual frames are less noticeable to the viewer.

We compared our ink-and-ray method to other related approaches for enhancing hand-drawn images: Lumo [Johnston 2002] and TexToons [Sýkora et al. 2011]. To simulate Lumo we extracted a normal field from our bas-relief approximation and rendered it using the same lighting conditions. As visible in Fig. 19a, due to lack of 3D structure, Lumo does not offer important depth-related global illumination effects such as complex self shadowing, glossy reflections, and diffuse color bleeding. In TexToons, Lumo-like shading is combined together with the knowledge of relative depth order to simulate the effect of ambient occlusion. However, the result is still purely a 2D approximation and does not respect 3D structure and produces a limited stylized look (Fig. 19b).

6.   PERCEPTUAL EXPERIMENT

We conducted a subjective perceptual experiment to verify the overall visual quality of renderings generated using the output of our ink-and-ray pipeline.

6.1   Stimuli

The observers compared images rendered using our bas-relief-type approximation with those using the ground-truth (full 3D model), normal map based shading (Lumo) [Johnston 2002], and a simple inflation from silhouette as used in [Ono et al. 2004] (see Fig. 20). We projected the ground-truth 3D model using an orthographic camera and used NPR techniques to produce a synthetic 2D line drawing which was then used to generate the approximate models. The three test models (*lioness, rings, snake*) were rendered using two different global illumination conditions (diffuse and diffuse plus specular).

6.2   Experimental Design and Procedure

We conducted a two-alternatives-forced-choice (2AFC) designed experiment [David 1988] with 89 observers (20 females and 69 males; age 18 to 71 years), all reporting to have normal or corrected-to-normal vision.

Each subject was introduced to the problem before the experiment. They were told the goal was to compare different methods for adding 3D effects to 2D drawings. For a series of inputs, the observer was presented with a reference line drawing and a pair of side-by-side rendered images and asked to choose the rendered image with the most correct 3D appearance. They were told to focus on shading, shadows, lighting and reflectance. The evaluated images were presented on uncalibrated displays. The observing distance was not controlled and no time constraints were imposed. The sequences of images were randomized. The experiment took on average 10 minutes per observer.

6.3   Agreement and Consistency

The agreement between observers was quantified by a coefficient of agreement u, ranging from $u = -1/(s-1) = -0.011$, where $s = 89$ is the number of observers, (which indicates no agreement) to $u = 1$ (all observers responded the same). The obtained value for our case $u = 0.47$ is rather high, meaning the observers had similar preferences during the judgment. Accordingly, the χ^2 test of significance clearly shows the statistical significance of u ($\chi^2 = 1515.1$, $p < 0.001$, 6 degrees of freedom), meaning the measured subjective responses are not random.

The coefficient of consistency of responses within subject ζ ranges from $\zeta = 0$ (no consistency) to $\zeta = 1$ (ideally consistent responses). The average consistency over all the observers $\zeta_{avg} = 0.93$ indicates that the observers were highly consistent in their responses and most of them did not change their preference during the experiment.

6.4   Study Results

Each observer performed $m \times n(n-1)/2 = 36$ comparisons, where $m = 3$(models) $\times 2$(illuminations) and $n = 4$ is the number of tested methods. The image chosen by an observer was given a score of 1, the other a score of 0. The data was stored in a 4×4 frequency
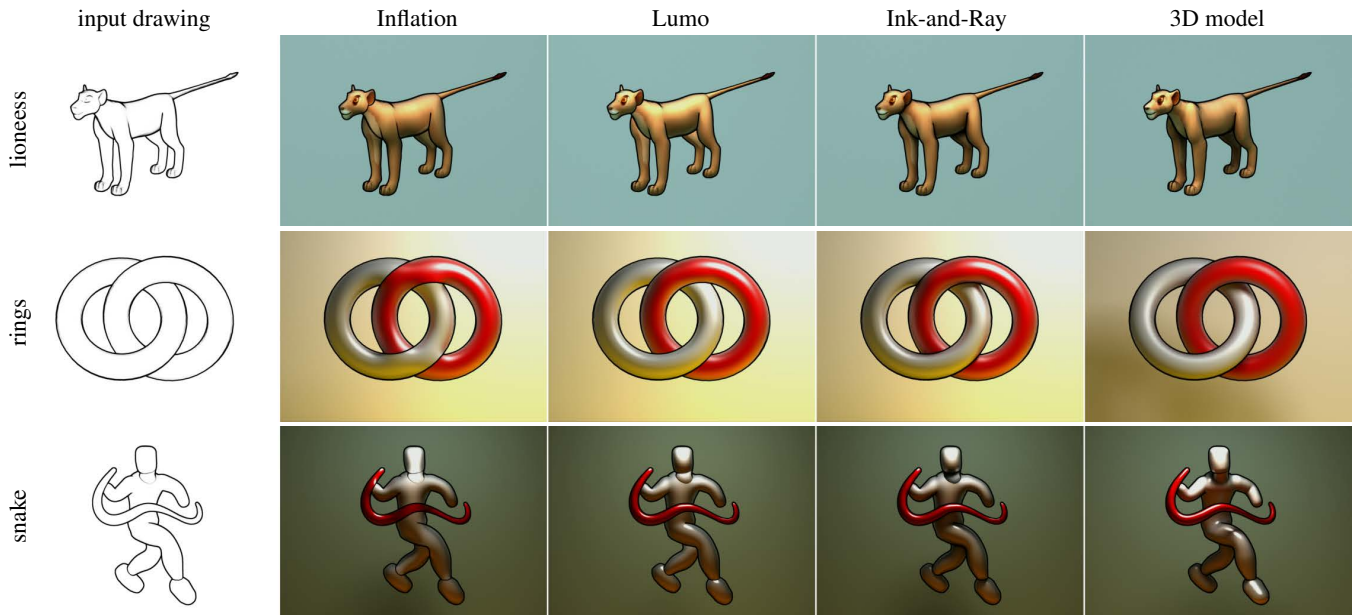
Fig. 20.   Perceptual experiment stimuli (each model is shown for one illumination condition only). (Snake 3D model © Yotam Gingold. All rights reserved.)
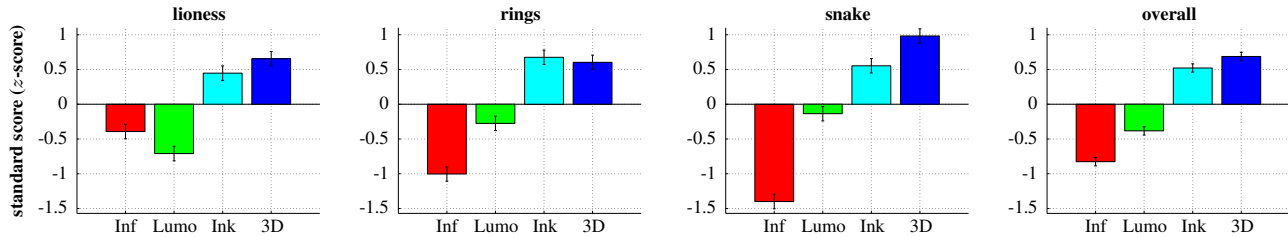

Fig. 21.   Results of the perceptual experiment (standard z-scores) per stimulus and overall results.

matrix for each observer, where the value in column i and row j represents the score of tested method i compared with method j.

The overall results as well as results for each stimulus separately are depicted in Fig. 21. This plot shows standard z-scores for all observers along with the confidence intervals. The z-scores were acquired from frequency matrices using the incomplete matrix Bradley-Terry-Luce model. The overall results using the full 3D model exhibit the best perceptual quality (the highest z-score), followed by our proposed ink-and-ray method, Lumo technique, and finally the simple inflation from silhouette.

To evaluate the significance of the results we apply the Kruskal-Wallis (non-parametric version of one-way ANOVA) test [Montgomery and Runger 1999]. The results of the test are summarized in Fig. 22. The null hypothesis "there is no difference in perceived quality of the presented stimuli" can clearly be rejected ($p \ll 0.001$), meaning the methods produce perceptually different results. The multiple comparison test (Tukey's honestly significant differences [Hochberg and Tamhane 1987]) returns an overall ranking of the conversions with an indication of the significance of the differences (see Fig. 23, top row). The results show that there is a statistically significant difference between all the tested methods. This means that inflation from the silhouette is statistically worse than the Lumo technique, the Lumo technique produces overall perceptually worse results than our proposed method, and that the rendering using the full 3D model is better than our approximation. An interesting result can be obtained when we deliberately do not take into account subjects who considered themselves as graphics experts. For this case (49 subjects; 19 females and 30 males), the difference between our method and the full 3D model is not statistically significant (see Fig. 23, bottom row). This corresponds to our empirical observation that non-expert observers typically do not recognize differences between our method and the full 3D model.

6.5   Study Summary

The results of this study show that observers prefer the renderings generated using a full 3D model (when available). However, our model was shown to be better with statistical significance than models produced by simple inflation from the silhouette and by the Lumo technique. Moreover, for non-expert observers, results generated using our ink-and-ray method can be considered perceptually comparable to those produced from a full 3D model.

7.   LIMITATIONS AND FUTURE WORK

In this section we discuss some limitations of the current prototype and propose avenues for improvements in future work.

The current implementation is unable to provide updates at interactive rates. Incremental updates are feasible, however further

	SS	$d.f.$	MS	χ^2	p
method	$439.1 \cdot 10^6$	3	$146.4 \cdot 10^6$	1233.1	$\ll 0.001$
error	$321.2 \cdot 10^6$	2132	150677		
total	$760.4 \cdot 10^6$	2135			

Fig. 22. Results of Kruskal-Wallis test (where SS denotes Sum of Squares, $d.f.$ means Degrees of Freedom, MS denotes Mean Square, χ^2 is χ^2 value, and p is p-value for the null hypothesis [Tabachnick and Fidell 2007]).

	Inflation	Lumo	Ink-and-Ray	3D model
with experts	-0.8263	-0.3829	0.5216	0.6875
without experts	-0.7093	-0.3438	0.4833	0.5698

Fig. 23. Overall performances of the tested methods (z-scores). The best result is *3D model*. In the first case when all participants are considered (top row), no methods were considered perceptually similar, however, in the second case where graphics experts were not taken into account (bottom row) our method and 3D model are considered perceptually similar (highlighted).

optimizations are necessary to deliver immediate visual feedback. One possible approach is to compute a quick on-the-fly approximation of the proxy while editing.

A key advantage of our bas-relief-type approximation is that the user is not required to draw side-views or to specify absolute depth values for any part of the drawings. However, this might be limiting in cases when the range of depths in the generated proxy mesh is either too small or large such that the resulting error starts to be visible in cast shadows and reflections (see Fig. 25). This drawback can be fixed by adding lower and upper bounds to the inequalities $C^{\geq}$ used during the stitching phase (see Sec. 4.5).

Our method assumes hand-drawn objects consist mainly of rounded shapes and does not provide larger control over sharp features. The user can globally reduce smoothness by applying an arbitrary cross section function to the inflation, however, it is currently not possible to control surface sharpness locally. To address this we plan to employ more advanced shape control approaches such as Cauchy constraints [Andrews et al. 2011].

Another practical limitation is that, due to the unreal depth proportions produced by our bas-relief-type approximation and due to the use of an orthographic camera, to be consistent, surrounding geometry must be constructed to be compatible within this context. This means the artist needs to take these factors into account and adjust surrounding geometry to follow the used bas-relief principle (e.g., tilt the ground plane to simulate perspective projection as shown in Fig. 24) or use bas-relief conversion [Weyrich et al. 2007] to deliver a similar solution automatically.
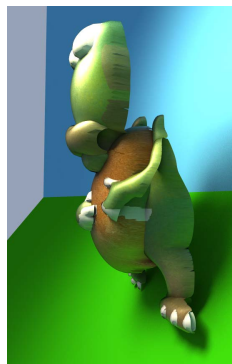

Fig. 24.

Although processing the animation frames separately produces reasonable results in general (see Fig. 18), the smoothness of transitions between individual frames still strongly depends on the coherency of the original source animation. By considering rough correspondences [Sýkora et al. 2009], temporal flickering could be reduced overall in the final rendered animation. In addition, visible
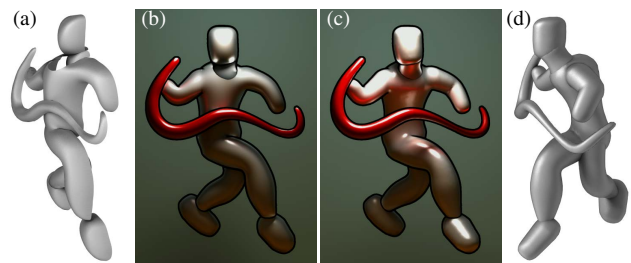
Fig. 25. Limitation—our bas-relief-type approximation (a) could contain too small (snake vs. body) or too large (neck vs. body) depth discontinuities which may influence the quality of the resulting global illumination (b). A full 3D model (d) rendered using the same lighting conditions is shown for comparison (c). (Source 3D model © Yotam Gingold. All rights reserved.)

portions of occluded objects from previous frames could be used to improve the shape completion phase.

8. CONCLUSIONS

We present a new ink-and-ray framework to enhance hand-drawn 2D artwork with global illumination effects. Our approach uses a bas-relief-type approximation to produce a visual appearance typically only found in complex 3D CG pipelines. As it requires very little manual input, we believe our approach could open a new potential for rapid prototyping in early phases of current CG pipelines, as well as support the production of a richer look for traditional hand-drawn animation.

REFERENCES

E. D. Andersen and K. D. Andersen. 2000. The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. In *High Performance Optimization*. Kluwer Academic Publishers, 197–232.

J. Andrews, P. Joshi, and N. A. Carr. 2011. A Linear Variational System for Modelling From Curves. *Computer Graphics Forum* 30, 6 (2011), 1850–1861.

P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. 1999. The Bas-Relief Ambiguity. *International Journal of Computer Vision* 35, 1 (1999), 33–44.

B.-Y. Chen, Y. Ono, and T. Nishita. 2005. Character Animation Creation using Hand-drawn Sketches. *The Visual Computer* 21, 8-10 (2005), 551–558.

F. Cole, P. Isola, W. T. Freeman, F. Durand, and E. H. Adelson. 2012. Shapecollage: Occlusion-Aware, Example-Based Shape Interpretation. In *Proceedings of European Conference on Computer Vision*. 665–678.

Fig. 26. Results—(a) the original drawing including user-defined annotations for segmentation, (b) result of the layering and completion phase with boundary condition annotations (red: Neumann and $C^{\geq}$, green: Dirichlet and $C^{\geq}$, blue: Neumann and $\mathcal{C}^1$ grafting), (c) generated bas-relief-type mesh, (d) original texture, (e) resulting global illumination render. (Source drawings: lady, wolf, dino, and farmer © Anifilm, robber © UPP. All rights reserved.)
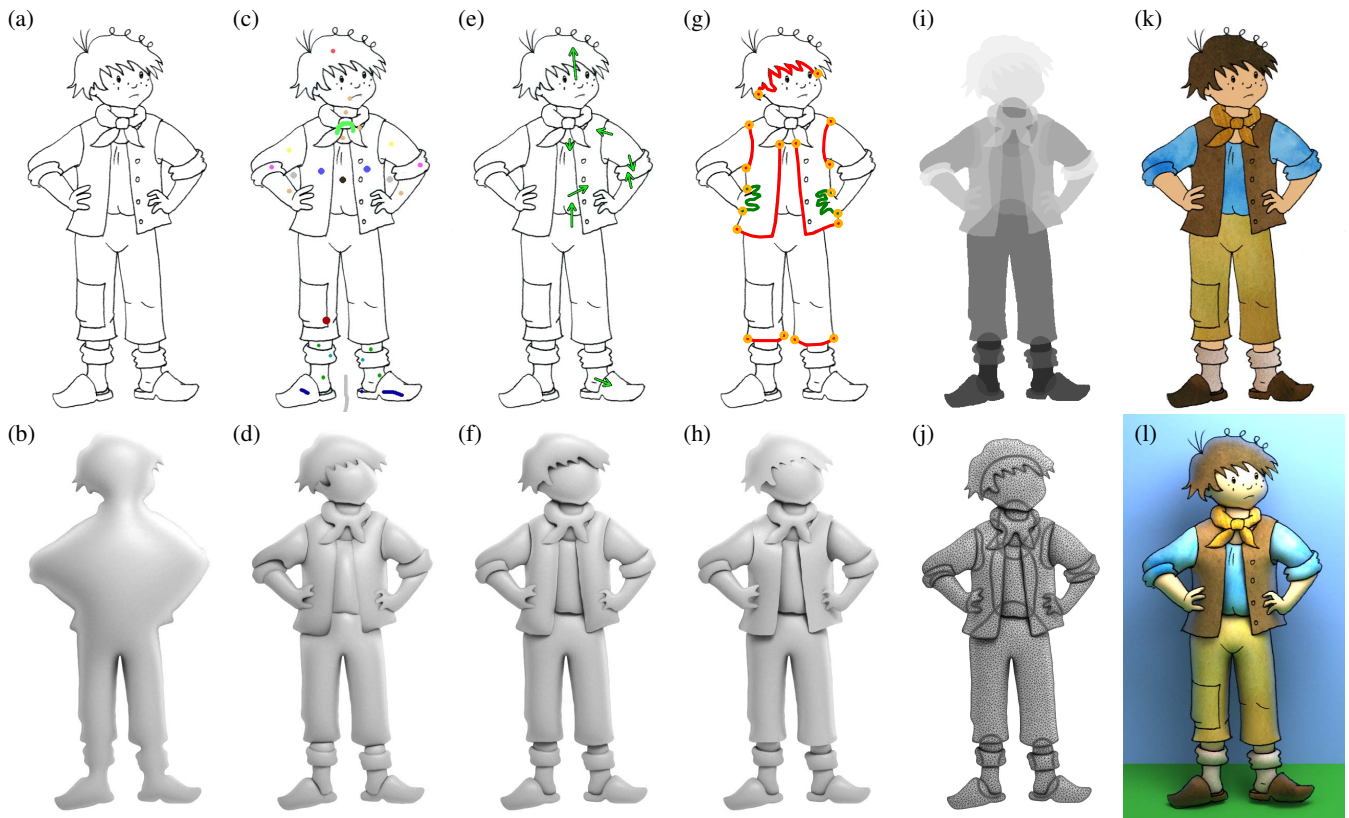
Fig. 27. Annotation workflow—(a) the original hand-drawn image, (b) simple inflation from silhouette, (c) segmentation scribbles specified by the user, (d) resulting bas-relief-type approximation with visible artifacts in depth order, (e) depth inequalities added by the user to fix incorrect depth ordering, (f) resulting bas-relief model has unwanted discontinuities using the default inflation and stitching parameters, (g) user-defined boundary conditions to improve shape of selected segments (green: $C^{\geq}$, red: Neumann with $C^{\geq}$), (h) final proxy mesh, (i) result of depth ordering and completion phases, (j) triangulation and topology of the resulting proxy mesh, (k) original texture, (l) resulting global illumination render. (Source drawing © Anifilm. All rights reserved.)
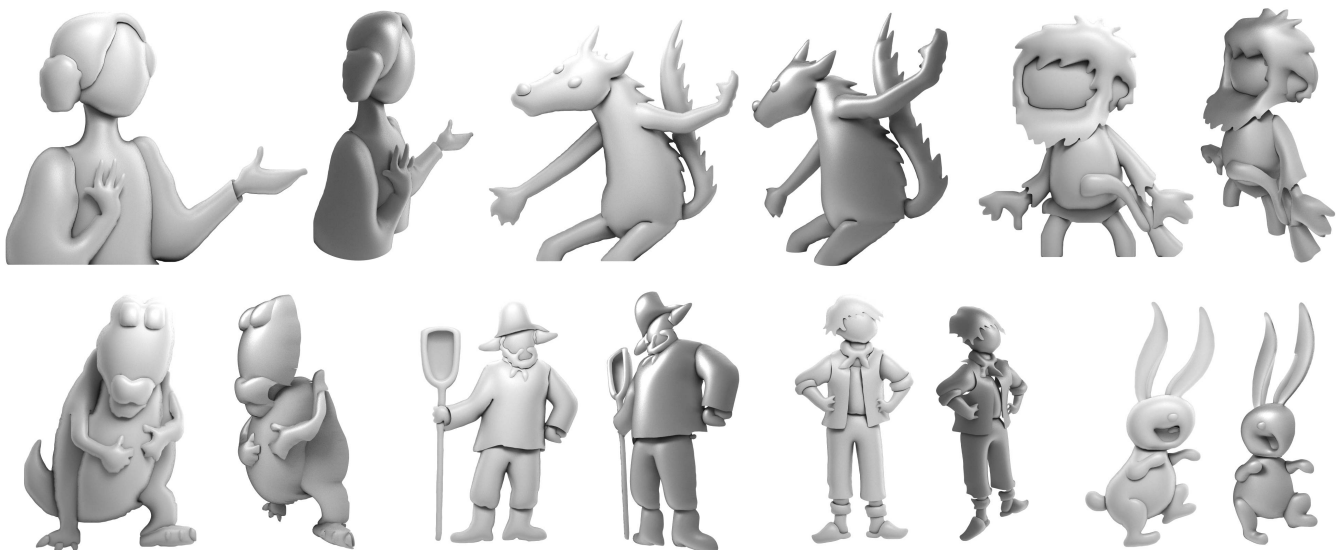

Fig. 28. Proxy meshes generated using our approach expose their approximate nature when rendered from sideviews using a perspective camera. Note the bas-relief-type false-depth structure which is not visible in the orthographic projection.

H. A. David. 1988. *The Method of Paired Comparisons* (2nd ed.). Oxford University Press.

D. Geiger, H.-K. Pao, and N. Rubin. 1998. Salient and Multiple Illusory Surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 118–124.

Y. Gingold, T. Igarashi, and D. Zorin. 2009. Structured Annotations for 2D-to-3D Modeling. *ACM Transactions on Graphics* 28, 5, Article 148 (2009).

Y. Hochberg and A. C. Tamhane. 1987. *Multiple Comparison Procedures* (1st ed.). Wiley.

T. Igarashi, S. Matsuoka, and H. Tanaka. 1999. Teddy: A Sketching Interface for 3D Freeform Design. In *ACM SIGGRAPH Conference Proceedings*. 409–416.

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

S. Jeschke, D. Cline, and P. Wonka. 2009. A GPU Laplacian Solver for Diffusion Curves and Poisson Image Editing. *ACM Transaction on Graphics* 28, 5, Article 116 (2009).

S. F. Johnston. 2002. Lumo: Illumination for Cel Animation. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*. 45–52.

P. Joshi and N. A. Carr. 2008. Repoussé: Automatic Inflation of 2D Artwork. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling*. 49–55.

J. T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Computer Graphics* 20, 4 (1986), 143–150.

O. A. Karpenko and J. F. Hughes. 2006. SmoothSketch: 3D Free-form Shapes from Complex Sketches. *ACM Transactions on Graphics* 25, 3 (2006), 589–598.

E. A. Khan, E. Reinhard, R. Fleming, and H. Buelthoff. 2006. Image-Based Material Editing. *ACM Transactions on Graphics* 25, 3 (2006), 654–663.

J. Lopez-Moreno, J. Jimenez, S. Hadap, E. Reinhard, K. Anjyo, and D. Gutierrez. 2010. Stylized Depiction of Images Based on Depth Perception. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 109–118.

J. Malik. 1986. *Interpreting Line Drawings of Curved Objects*. Ph.D. Dissertation. Stanford University.

D. C. Mongomery and G. C. Runger. 1999. *Applied Statistics and Probability for Engineers* (2nd ed.). John Wiley & Sons.

A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. 2007. FiberMesh: Designing Freeform Surfaces with 3D Curves. *ACM Transactions on Graphics* 26, 3, Article 41 (2007).

G. Noris, D. Sýkora, A. Shamir, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, and R. Sumner. 2012. Smart Scribbles for Sketch Segmentation. *Computer Graphics Forum* 31, 8 (2012), 2516–2527.

M. Okabe, G. Zeng, Y. Matsushita, T. Igarashi, L. Quan, and H.-Y. Shum. 2006. Single-View Relighting with Normal Map Painting. In *Proceedings of Pacific Conference on Computer Graphics and Applications*. 27–34.

L. Olsen, F. Samavati, and J. A. Jorge. 2011. NaturaSketch: Modeling from Images and Natural Sketches. *IEEE Computer Graphics and Applications* 31, 6 (2011), 24–34.

Y. Ono, B.-Y. Chen, and T. Nishita. 2004. 3D Character Model Creation from Cel Animation. In *Proceedings of International Conference on Cyberworlds*. 210–215.

A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-Shaded Images. *ACM Transactions on Graphics* 27, 3, Article 92 (2008).

Y. Ostrovsky, P. Cavanagh, and P. Sinha. 2005. Perceiving Illumination Inconsistencies. *Perception* 34, 11 (2005), 1301–1314.

S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich. 2010. OptiX: A General Purpose Ray Tracing Engine. *ACM Transactions on Graphics* 29, 4, Article 66 (2010).

L. Petrović, B. Fujito, L. Williams, and A. Finkelstein. 2000. Shadows for Cel Animation. In *ACM SIGGRAPH Conference Proceedings*. 511–516.

H. Read. 1961. *The Art of Sculpture* (2nd ed.). Bollingen Foundation, New York.

M. W. Ren, J. Y. Yang, and H. Sun. 2002. Tracing Boundary Contours in a Binary Image. *Image and Vision Computing* 20, 2 (2002), 125–131.

A. Rivers, T. Igarashi, and F. Durand. 2010. 2.5D Cartoon Models. *ACM Transactions on Graphics* 29, 4, Article 59 (2010).

C. Shao, A. Bousseau, A. Sheffer, and K. Singh. 2012. CrossShade: Shading Concept Sketches Using Cross-Section Curves. *ACM Transactions on Graphics* 31, 4, Article 45 (2012).

J. R. Shewchuk. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Proceedings of ACM Workshop on Applied Computational Geometry*. 203–222.

J. R. Shewchuk. 2002. Delaunay Refinement Algorithms for Triangular Mesh Generation. *Computational Geometry: Theory and Applications* 22, 1–3 (2002), 21–74.

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

D. Sýkora, J. Buriánek, and J. Žára. 2005. Colorization of Black-and-White Cartoons. *Image and Vision Computing* 23, 9 (2005), 767–782.

D. Sýkora, J. Dingliana, and S. Collins. 2009. As-rigid-as-possible Image Registration for Hand-drawn Cartoon Animations. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*. 25–33.

D. Sýkora, J. Dingliana, and S. Collins. 2009. LazyBrush: Flexible Painting Tool for Hand-drawn Cartoons. *Computer Graphics Forum* 28, 2 (2009), 599–608.

D. Sýkora, D. Sedlacek, S. Jinchao, J. Dingliana, and S. Collins. 2010. Adding Depth to Cartoons Using Sparse Depth (In)equalities. *Computer Graphics Forum* 29, 2 (2010), 615–623.

B. G. Tabachnick and L. S. Fidell. 2007. *Using Multivariate Statistics* (5th ed.). Pearson Education.

C. Toler-Franklin, A. Finkelstein, and S. Rusinkiewicz. 2007. Illustration of Complex Real-World Objects Using Images with Normals. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 111–119.

R. Vergne, P. Barla, R. W. Fleming, and X. Granier. 2012. Surface Flows for Image-based Shading Design. *ACM Transactions on Graphics* 31, 4, Article 94 (2012).

Y. Wang, Y. Chen, J. Z. Liu, and X. Tang. 2009. 3D Reconstruction of Curved Objects from Single 2D Line Drawings. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1834–1841.

T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein. 2007. Digital Bas-Relief from 3D Scenes. *ACM Transactions on Graphics* 26, 3, Article 32 (2007).

H. Winnemöller, A. Orzan, L. Boissieux, and J. Thollot. 2009. Texture Design and Draping in 2D Images. *Computer Graphics Forum* 28, 4 (2009), 1091–1099.