# ShapePalettes: Interactive Normal Transfer via Sketching

Tai-Pang Wu
The Hong Kong University of Science and Technology

Chi-Keung Tang
The Hong Kong University of Science and Technology

Michael S. Brown
Nanyang Technological University

Heung-Yeung Shum
Microsoft Research Asia

Figure 1: An example of 3D markup using *Shape Palettes*. The 'question mug' is created by linking 2D primitives drawn in the freehand view to their corresponding 3D shape on the "shape palette" (the sphere). For example, the 3D orientation of the top and bottom curves of the mug correspond to the 3D orientation of the green curve on the shape palette. Specifying this relationship will generate a cylinder-like structure. Adding only a few more strokes for extra details and the 'question mug' is created.

## Abstract

We present a simple interactive approach to specify 3D shape in a single view using "shape palettes". The interaction is as follows: draw a simple 2D primitive in the 2D view and then specify its 3D orientation by drawing a corresponding primitive on a *shape palette*. The shape palette is presented as an image of some familiar shape whose local 3D orientation is readily understood and can be easily marked over. The 3D orientation from the shape palette is transferred to the 2D primitive based on the markup. As we will demonstrate, only sparse markup is needed to generate expressive and detailed 3D surfaces. This markup approach can be used to model freehand 3D surfaces drawn in a single view, or combined with image-snapping tools to quickly extract surfaces from images and photographs.

**Keywords:** Human-computer interaction, interactive modeling, image-based modeling.

## 1 Motivation

Humans have a remarkable ability to infer 3D structure from 2D images. This ability is applicable to items ranging from photographs and paintings to simple sketches and line art. Often only a few 2D strokes are necessary to express the 3D shape of an object – the observer instantly makes the analogy of each 2D stroke to its corresponding 3D shape and can infer the overall 3D surface. While this inference is done mentally without effort, it remains difficult to convey this information easily via a 2D interface. The problem lies not in *what* to specify, but in *how* to specify the 3D information.

To address this issue of *how* to specify 3D information, we introduce an approach based on what we term *shape palettes*. In our approach, the user draws a simple 2D primitive in the single view. To assign 3D information to this primitive, the user then draws a corresponding primitive on a shape palette. The shape palette is an image of some familiar object that provides salient 3D orientation information that can be easily understood and marked. For example, a single sphere serves as an excellent shape palette as its shape is universally understood and it provides all possible 3D orientations (in a half-plane).

Similar in fashion to how color palettes are used for color selection, the shape palette provides a familiar metaphor for linking 3D information to the 2D input. As we will demonstrate, this simple interaction approach can be used to create expressive 3D surfaces. Moreover, only sparse markup is necessary to derive dense 3D structure. This interaction style has a variety of uses, from single-view surface modeling, to image-based markup for 3D pop-up or image-relighting.

Figure 1 shows an example of a 'mug' that was quickly prototyped using only a few freehand strokes marked up using a shape palette. The first image is the sketched strokes with corresponding markup on the shape palette (second image). The third image is the dense 3D information derived from the sparse markup. The last two images are views of the 3D model that was generated by stitching together a reflected copy of the derived 3D surface. Figure 2 shows another example targeting modeling from a photograph. In this example, 2D features in the photograph are related to their corresponding 3D shape using the shape palette. The examples in Figure 1 and Figure 2 were generated in a matter of minutes with little effort.

## 2 Contributions and Related Work

Our first contribution is the *shape palette* metaphor, which is an intuitive and straightforward way to address the problem of how to specify 3D markup in a single view. A shape palette is simply a 2D image that corresponds to some 3D geometric information easily inferred by the user. One very good example used throughout this paper is a Lambertian shaded sphere. Using a 2D drawing interface, 3D information from the shape palette is transferred to a single 2D view by direct 2D interaction. This allows 3D markup to be specified using the familiar palette metaphor, similar to how color is specified by most drawing tools.

Our second contribution is *one* practical way to implement shape palette markup using surface normals. In our implementation, shape palettes are in fact normal maps. The transfer of 3D shape is a transfer of 3D normal information, copied from the shape palette source to its corresponding 2D primitive. Reasonably complex and expressive 3D structure can be generated using this normals transfer

**ACM Reference Format**
Wu, T., Tang, C., Brown, M., Shum, H. 2007. ShapePalettes: Interactive Normal Transfer via Sketching. *ACM Trans. Graph. 26*, 3, Article 44 (July 2007), 5 pages. DOI = 10.1145/1239451.1239495 http://doi.acm.org/10.1145/1239451.1239495.

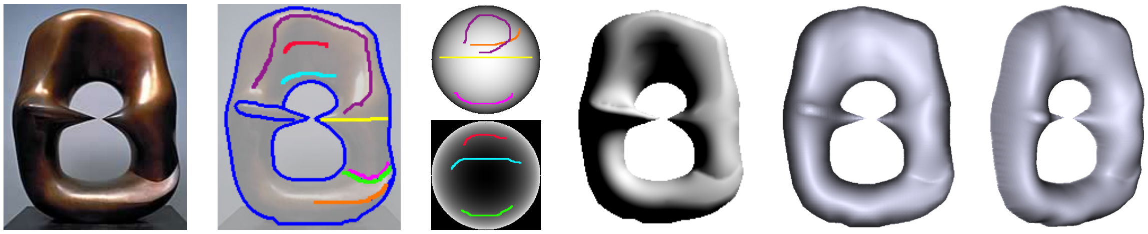


Figure 2: Markup over a photograph using shape palettes. Sparse 2D strokes are drawn over a photograph of Henry Moore's *Working Model for Oval with Points*. The shape palette provides the user a straightforward metaphor to link 3D orientation to the 2D strokes. Dense surface normals are derived from the sparse markup (middle image). A 3D surface is generated from the dense normal image as shown in the last two images. Note the overall similarity to the photograph with details preserved in the reconstruction, including the two silhouetted points which just fail to touch each other at the center of the object.

together with some simple transfer "tricks" to provide local detail. Integrating this approach with image-snapping tools, sketches and photographs of 3D surfaces can be quickly "popped up" and used for a variety of purposes.

In terms of information transfer, our method is related to [Sloan et al. 2001] that used a user-generated reference sphere to specify NPR shading for a 3D object. A similar idea on normal transfer was presented in [Hertzmann and Seitz 2005], where multiple images are required and surface normals are transferred based on orientation-consistency. Our ShapePalette idea is also inspired by Pictorial Relief [Koenderink 1998] which suggests that humans are good at assigning local surface normals for specifying local shape. Such local shapes can be, as noted in [DeCarlo et al. 2003; Ohtake et al. 2004], expressed as 3D crease curves, such as valleys and ridges, which provide adequate geometric cues for describing the overall appearance of a detailed 3D surface. These 3D crease curves correspond to salient normal directions on the 3D surface [Nalwa 1993]. Thus, specifying normals is a powerful way to encode up 3D shape. Moreover, only sparse normal information is needed from which a dense normal map can be derived that is capable of generating quality 3D surfaces that satisfy the given normal constraints. An algorithm to derive dense-from-spare normal maps is described in Appendix A. Our approach is related to [van Overveld 1996], which creates shapes by painting a dense gradient map with brushes and operators. Our method, however, is sparse in nature and could be used as a good initializer for further dense refinement using the technique presented by [van Overveld 1996].

Our work is also motivated by single-view and interactive modeling techniques. Successful techniques such as [Criminisi et al. 2000; Hoiem et al. 2005] create rectilinear scenes out of a single photo or picture. The "SmoothSketch" [Karpenko and Hughes 2006] is a successful experimental system for inferring 3D shapes from visible-contour sketches. Many sketch-based modeling techniques (e.g. SKETCH [Zeleznik et al. 1996] and TEDDY [Igarashi et al. 1999]) provide an intuitive interface for creating organic shapes. Interactive modeling techniques such as [Funkhouser et al. 2004; Nealen et al. 2005] edit existing meshes or assemble complex meshes from mesh parts. Our work is distinguished from previous approach in terms of *how* we markup the 3D shape, in our case using the shape palette.

In terms of implementation, [Zhang et al. 2001] also used normal specification to perform 3D modeling. This method considers both height and surface normals and minimizes a thin plate energy to produce a 3D surface. While only sparse normal information is required, this approach performs normal markup in a pointwise fashion. Such pointwise normal markup is time consuming and is not particularly easy to markup for an evolving surface (as shown in our accompanying video). Moreover, a significant number of these constraints are needed to produce a quality 3D surface.


Figure 3: User interaction.

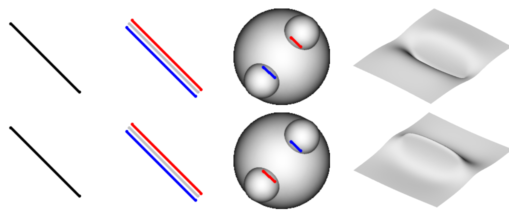


Figure 4: Creating valley (top) and ridge lines (bottom) with two strokes drawn along the black contour and two corresponding strokes on the shape palette.

## 3 Interacting with Shape Palettes

**Creating shape palettes.** A shape palette can be easily generated using orthographic projection of available 3D models, where the palette stores the inner product of a synthetic light and the 3D normal at that point (i.e. Lambertian shading). There are no restrictions on a shape palette as long as it is a familiar shape with salient 3D structure. To represent all possible normal orientations, an image containing both a concave and convex sphere is sufficient, however, the shape palette can be anything the user feels comfortable using. We find that spheres are quite intuitive. Spheres of different scales can also make marking scale a little easier, but is not necessary. In the accompanying video, we demonstrate that the markup for creating a particular shape need not be that exact. Users can experiment with different ways to draw and combine strokes to quickly materialize 3D shapes.

**User Interface.** A typical 2D drawing interface allows the user to draw on top of a blank canvas, or a canvas with a photograph/sketch for the background. For a blank canvas, 2D primitives are drawn freehand on the canvas as well as on the shape palette. For drawing over a photograph or sketch, image-snapping tools (intelligent scissor [Mortensen and Barrett 1995] for instance) can be used to help guide 2D markup on the image, while freehand drawing is used for marking on the shape palettes.

As shown in Figure 3, there are two types of 2D contours drawn on the canvas, those that form silhouette curves and those that form non-silhouettes. *Silhouettes* do not require 3D markup as their corresponding normals are considered to lie in the canvas plane and are oriented perpendicular to the 2D drawn curves. *Non-silhouettes*

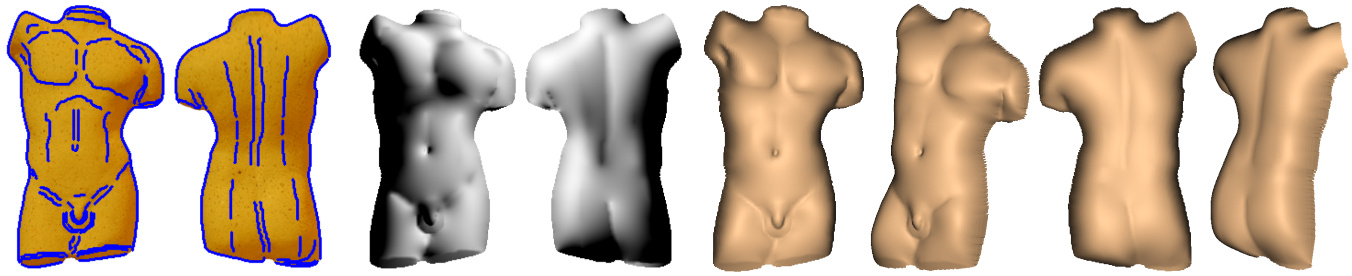
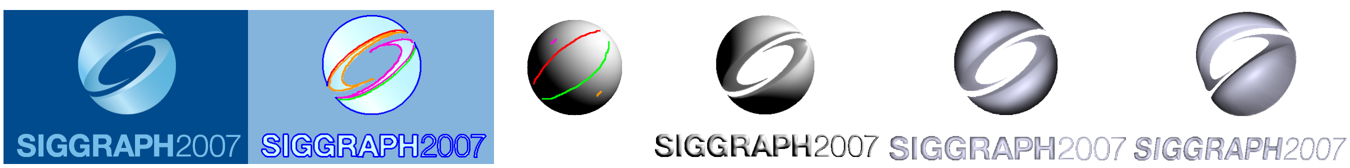


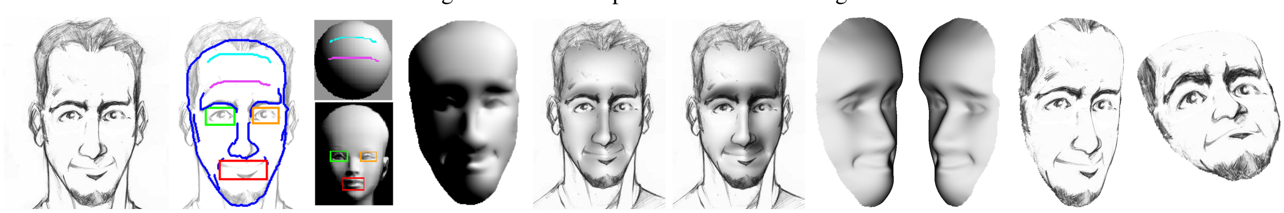
Figure 5: 3D modeling of a statuette: (left to right) user supplied strokes; two views of the derived dense normal maps shown using Lambertian shading; four views of the 3D shape.

Figure 6: Parthenon frieze (modeling relief texture conveying human shapes): (left to right) Input image with input strokes; derived dense normal map shown using Lambertian shading; four views of the texture-mapped surface




Logo | Strokes on logo | Strokes on palette | Dense normals | View 1 | View 2

Figure 7: 3D markup of the SIGGRAPH logo.



Sketch | Strokes on palettes | Dense normals | View 1 | View 2 | View 3 | View 4 | View 5 | View 6

Figure 8: Face modeling. 3D markup of more complex shapes, such as eyes and mouth, can be easily done using the "face shape palette". The face model is used for image-based relighting (views 1 and 2). Other views (without and with texture-mapped) are shown.

curves have corresponding markup on the shape palettes. In our implementation, non-silhouette curves are typically drawn in a tit for tat fashion with the shape palette, where the user draws a 2D stroke on the canvas and then draws a corresponding 2D stroke on the shape palette.

**Markup via Palettes.** For a non-silhouette curve, the user draws a corresponding primitive on a shape palette. Normals from the shape palette are transferred to the 2D primitive. This markup can be casually applied to specify orientation information throughout the 2D view. The following simple "tricks" can be quickly learned to add fine surface details in the form of ridge and valley lines. As shown in Figure 4, two contours (red and blue) bound the input contour where a ridge or valley is desired. Generating such bounding contours pairs can be controlled via a hot-key. Corresponding strokes on the shape palette are drawn for the contour pair. If the corresponding strokes are of normals pointing away from one another, a ridge will be generated; normals pointing towards one another will form a valley. As shown in the accompanying video, a user can experiment with different locations of strokes drawn on the palettes and can visualize the corresponding 3D effect almost instantly.

Given these sparse normal constraints, we can derive a dense normal field. Our implementation of the dense-from-sparse-normals algorithm is outlined in the appendix. A 3D surface is generated by integrating the dense surface gradients, or by other surface-from-dense-normals algorithms such as [Kovesi 2005; Frankot and Chellappa 1988; Goldman et al. 2005; Wu et al. 2006].

## 4 Shape Palettes in Use

When specifying markup between the 2D view and a shape palette, the length of the corresponding strokes and the relative scale between the 2D view and shape palette are not important. Transfer is applied as simple 1D copying along the corresponding strokes. This is done by parameterizing the two strokes as normalized splines (with length $[0-1]$) and transferring data between these splines. In addition, our dense-from-sparse normal algorithm does not require accurate normal specification to capture coarse shape and degrades gracefully in the face of bad input. The accompany video shows examples where the same overall surface is obtained with similar,

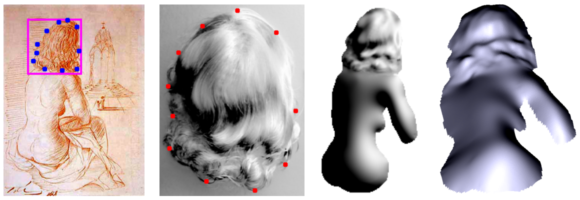


Figure 9: Normals from a “hair shape palette” are transferred to the corresponding region in the 2D view. Additional markup is shown in Figure 3. The output dense normals and one view of the 3D surface are shown.

Figure 10: With the locations and normals of *Oval*, we render the geometry using ray tracing with the Phong illumination model to make the object transparent.

but not exact, markup.

We demonstrate the ability of our 3D markup approach in the following examples. Unless otherwise stated, all results are generated using the sphere shape palette. Figure 5 shows an example where strokes are drawn on two matchable views of a statuette. Figure 6 shows an example of modeling relief texture using a photo input. This relief texture is complicated as both overlapping curved surfaces and flat surfaces are present. Note that all underlying surface orientation discontinuities are preserved in the reconstruction. The appearance of the reconstructed relief is faithful to the original relief. As shown in Figure 7, our shape palettes approach is ideal for quick 3D prototyping. The user draws a short stroke on the palette to transfer normals of nearly constant orientations to model the flat part of the SIGGRAPH logo.

Figure 8 shows a natural extension of the shape palette idea where patches of normals are transferred to the 2D view. In this example, a “face shape palette” was created by orthographic projection of an existing 3D face model. The eyes and mouth shape are transferred to the single view by specifying matching points between the 2D view and shape palette. Normals from the shape palette are warped to their corresponding regions using Thin Plate Spline (TPS) [Bookstein 1989]. The derived dense normal image are used to shade and relight the sketch. Related works such as [Johnston 2002; Okabe et al. 2006] which relight cartoons and real scenes using dense normals can benefit from our markup approach. Other views of the recovered surface (with and without texture) are also shown. Shown in Figure 9 is a result where a hair-structure shape palette is used for normal transfer.

The output dense normal maps produced by our method are suitable for generating special effects that require geometric information. Figure 10 shows an example which simulates the effect of environment matting for an opaque-turned-to-transparent object, where a few sketch strokes are all it takes to produce the surface normals needed for ray tracing.

## 5 Discussion and Limitations

To examine the usability of our approach, we compare our system with a related work by [Zhang et al. 2001] in terms of the number of primitives drawn and the interaction time used, given that surfaces of comparable visual quality are produced. The comparison is performed by a user who is reasonably familiar with both

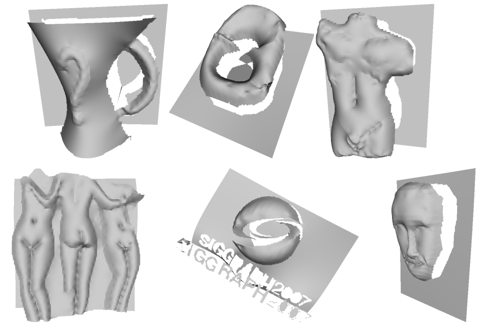


Figure 11: Surface results produced by using [Zhang et al. 2001]. Interaction details are shown in Table 1.

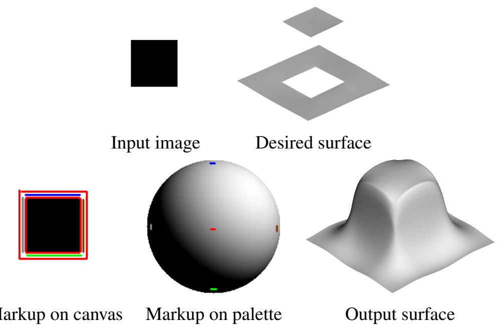



Figure 12: This example shows a limitation of our system. Occlusion boundary with unknown depth difference cannot be handled.

ShapePalettes and [Zhang et al. 2001]. Table 1 shows the interaction details and the time required for the examples. The corresponding surface results produced by using [Zhang et al. 2001] are shown in Figure 11. Better surfaces are produced with shorter interaction time using the ShapePalette markup approach. Note that there is no way to incorporate normal patches to assign surface details (e.g., Figure 8) in [Zhang et al. 2001]. Matter of fact, the interface in [Zhang et al. 2001] could benefit from our shape palette approach for specifying normals.

Our method is not without limitations and has room for improvement. Our system can currently only model single surfaces that exhibit little self-occlusion. An example is shown in Figure 12. Distortion is observed in the output surface because of the enforcement of the integrability constraint in the applied surface-from-normals method. We also assume the input sketch or photo does not have severe perspective distortion. Even with these limitations, however, the shape palette markup approach still provides an attractive method of 3D markup given its ease and straightforward use.

## 6 Summary

We have presented the shape palette metaphor for 3D markup and introduced an implementation based on normals. We have only begun to explore the potential of this approach to materialize 3D information from 2D marking, and can already demonstrate the immediate benefits from this approach on common tasks ranging from 3D markup over photos and sketches, image-based modeling and rendering, freehand 3D modeling, to quick 3D prototyping. This technique can lead to more, possibly complementary uses with other interaction techniques in traditional and image-based modeling.

## A Dense Normals from Sparse Normals

Denote a normal using $\frac{1}{\sqrt{p^2+q^2+1}}[-p \;\; -q \;\; 1]^T$ where $p = -\frac{x_i}{z_i}$ and $q = -\frac{y_i}{z_i}$ associated with a unit normal $[x_i \; y_i \; z_i]^T$ at pixel $i$. One typical implementation of estimating $p_i$ is outlined for all pixels. Estimation of $q_i$ is similar.

| | | **ShapePalettes** | | | **[Zhang et al. 2001]** | | | |
|---|---|---|---|---|---|---|---|---|
| Input | Image size | No. of strokes on canvas | No. of patches | **Interaction time** | No. of normal constraints | No. of fairness curves | No. of discontinuity curves | **Interaction time** |
| Figure 1 | $274 \times 284$ | 14 | 0 | **1m28s** | 0 | 5 | 4 | **3m46s** |
| Figure 2 | $220 \times 250$ | 11 | 0 | **1m27s** | 32 | 7 | 3 | **6m57s** |
| Figure 5 | $244 \times 342$ | 38 | 0 | **2m20s** | 73 | 0 | 16 | **5m45s** |
| Figure 6 | $306 \times 324$ | 75 | 0 | **2m38s** | 41 | 9 | 27 | **20m27s** |
| Figure 7 | $491 \times 307$ | 23 | 0 | **4m27s** | 13 | 20 | 2 | **15m41s** |
| Figure 8 | $225 \times 334$ | 15 | 3 | **1m32s** | 10 | 1 | 6 | **5m32s** |

Table 1: The table compares the interaction details between ShapePalettes and [Zhang et al. 2001]. As shown in Figure 11, ShapePalettes can produce visually better surfaces with shorter interaction time. Note that the normal constraints used in [Zhang et al. 2001] are specified in a pointwise manner, where each normal direction needs to be adjusted with a projected line using a series of mouse click-and-drag. Please refer to the accompanying video to compare the actual operation on results shown in Figures 1 and 5.

Let $\mathbf{G} = \{p_i | i \in 1 \cdots N\}$ be a set of $p$'s and $N$ is the total number of pixels. The goal is to estimate the optimal $\mathbf{G}$ given the sparse set of known gradient values, represented by the observation set $\mathbf{O} = \{\tilde{p}_k | k \in \mathcal{S}\}$, where $\tilde{p}_k$ is known and $\mathcal{S}$ is the set of corresponding pixel locations. The associated energy function is

$$E(\mathbf{G}) = \log(P(\mathbf{O}|\mathbf{G})) + \log(P(\mathbf{G})) \quad (1)$$

where $P(\mathbf{O}|\mathbf{G})$ is the likelihood and $P(\mathbf{G})$ is the prior. The likelihood are defined as

$$P(\mathbf{O}|\mathbf{G}) = \prod_{k \in \mathcal{S}} \exp\left(-\frac{||p_k - \tilde{p}_k||^2}{2\sigma_1^2}\right) \quad (2)$$

The prior $P(\mathbf{G})$ is defined as:

$$\prod_i \prod_j \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_2^2}\right) \prod_i \exp\left(-\frac{||\sum_j p_j - 4p_i||^2}{2\sigma_3^2}\right) \quad (3)$$

where $\sigma_1, \sigma_2, \sigma_3$ are the respective uncertainty measurements, and $j \in \mathcal{N}(i)$ is the pixel locations of the first-order neighbors of $i$. The first exponent in the prior energy (3) enforces the smoothness of the normal orientation, while the second minimizes the surface curvature. The energy function (1) is convex. Standard numerical optimization packages can be used to solve (1).

## Acknowledgment

This research was supported in part by the Hong Kong Special Administrative Region under grant HKUST620006. We thank Zhang et al. for their single view modeling program, and Ying-Qing Xu and the anonymous SIGGRAPH reviewers for their constructive comments and recommendations.

## References

BOOKSTEIN, F. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI 11*, 6 (June), 567–585.

CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *IJCV 40*, 2 (November), 123–148.

DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph. 22*, 3, 848–855.

FRANKOT, R., AND CHELLAPPA, R. 1988. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 4, 439–451.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3, 652–663.

GOLDMAN, D., CURLESS, B., HERTZMANN, A., AND SEITZ, S. 2005. Shape and spatially-varying brdfs from photometric stereo. In *ICCV05*, I: 341–348.

HERTZMANN, A., AND SEITZ, S. 2005. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *PAMI 27*, 8 (August), 1254–1264.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph. 24*, 3, 577–584.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99*, 409–416.

JOHNSTON, S. F. 2002. Lumo: illumination for cel animation. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, ACM Press, New York, NY, USA.

KARPENKO, O. A., AND HUGHES, J. F. 2006. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Trans. Graph. 25*, 3.

KOENDERINK, J. 1998. Pictorial relief. *Royal 356*, 1740, 1071–1086.

KOVESI, P. 2005. Shapelets correlated with surface normals produce surfaces. In *ICCV05*, 994–1001.

MORTENSEN, E., AND BARRETT, W. 1995. Intelligent scissors for image composition. *Proceedings of ACM SIGGRAPH95*, 191–198.

NALWA, V. 1993. *A Guided Tour of Computer Vision*. Addison-Wesley.

NEALEN, A., SORKINE, O., ALEXA, M., AND COHEN-OR, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph. 24*, 3, 1142–1147.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. 23*, 3, 609–612.

OKABE, M., ZENG, G., MATSUSHITA, Y., IGARASHI, T., QUAN, L., AND SHUM, H.-Y. 2006. Single-view relighting with normal map painting. In *Proceedings of Pacific Graphics*, 27–34.

SLOAN, P. J., MARTIN, W., GOOCH, A., AND GOOCH, B. 2001. The lit sphere: A model for capturing NPR shading from art. In *Proceedings of Graphics Interface 2001*, B. Watson and J. W. Buchanan, Eds., 143–150.

VAN OVERVELD, C. W. A. M. 1996. Painting gradients: Free-form surface design using shading patterns. In *Graphics Interface '96*, Canadian Human-Computer Communications Society, W. A. Davis and R. Bartels, Eds., 151–158.

WU, T.-P., TANG, K.-L., TANG, C.-K., AND WONG, T.-T. 2006. Dense photometric stereo: A markov random field approach. *PAMI 28*, 11 (November), 1830–1846.

ZELEZNIK, R. C., HERNDON, K. P., AND HUGHES, J. F. 1996. Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '96*, 163–170.

ZHANG, L., DUGAS-PHOCION, G., SAMSON, J., AND SEITZ, S. 2001. Single view modeling of free-form scenes. In *CVPR01*, I:990–997.