Single-View Relighting with Normal Map Painting

Makoto Okabe[1] Gang Zeng[2] Yasuyuki Matsushita[3]
Takeo Igarashi[1,4] Long Quan[2] Heung-Yeung Shum[3]

[1]The University of Tokyo, Tokyo, Japan
{makoto21,takeo}@ui.is.s.u-tokyo.ac.jp

[2]HKUST, Hong-Kong, China
{zenggang,quan}@cse.ust.hk

[3]Microsoft Research Asia, Beijing, China
{yasumat,hshum}@microsoft.com

[4]PRESTO, JST

Abstract

This paper presents an interactive technique for changing the illumination of objects in a real photograph. [1] *Because lighting effects are very sensitive to surface normal perturbations but hard to automatically analyze, we rely on a pen-based interface to quickly draw an approximate normal map over a photo. The photo and the approximate normal map are then given as input to a novel algorithm that refines the map and assigns reflectance to every pixel. We present relighting results for a variety of scenes, and use our technique to match the illumination of multiple photographs.*

1. Introduction

Illumination has often been used by painters and photographers as a tool for eliciting striking visual impressions. While physical control of illumination is sometimes possible, changing the illumination of a photo after it has been taken can prove a challenging task. One possible solution is to photograph the same scene under many different illumination conditions and then combine these photos to create new images of the same scenes [16, 14, 13, 6]. Unfortunately, these types of datasets are hard to obtain because they require a large number of photos, usually under controlled illumination conditions. The other approach is to obtain a three-dimensional (3D) model of the scene using a range scanner and apply inverse rendering techniques [15]. However, such data is not always available, especially it is difficult for the scenario of using existing photographs. When just a single photo of a scene is available, with no prior information about its shape or illumination, the relighting problem becomes especially difficult.

[1]This work was done while the first and second authors were visiting Microsoft Research Asia.

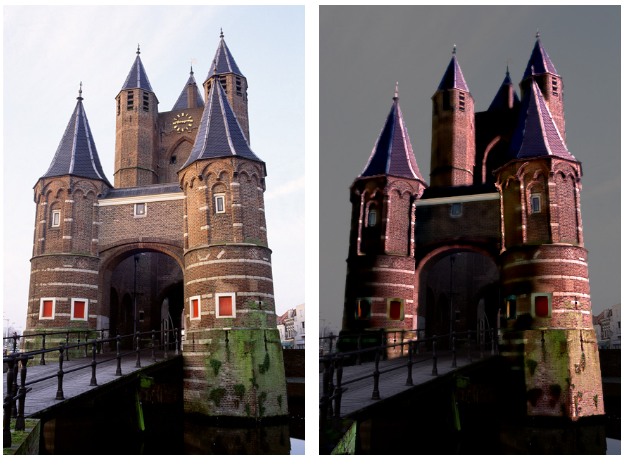


Figure 1. A photograph(left) is relit as a novel image(right) by a user specified lighting condition.

The colors and intensities within a photo are created from the interaction of light with surfaces in the scene. This interaction depends on the reflectance properties of the surfaces, as well as their shape. As a result, accurate relighting can be done only by analyzing how surface reflectance, surface shape, and the light source position(s) affect a given photo. This is a highly under-constrained problem because every pixel in the image is a function of incident lighting, surface orientation and reflectance, none of which are known.

Our approach is to efficiently extract a representation that is specifically designed for re-lighting applications. Since we use a single photograph as input, our work is closely related to prior work on single-view modeling [17, 19, 1, 4, 3, 9]. These methods emphasized the detailed acquisition of full 3D scene models using a variety of interactive tools. Unlike these approaches, we rely on a reduced scene representation that is based on surface normals and is much easier to interactively specify. We propose a novel user interface (UI) to assign surface normals to a photograph based on the fact that a human is capable of perceiving surface orientation more precisely than depth of the scene [10]. The surface normal representation has been used in the scenario of illumination control for cel animation [8]. In our scenario of relighting a single photograph, this representation is also useful in manipulating photographs where the dominant pixel intensity variations are caused by local lighting effects. We show that our technique works well on photographs of many complex scenes.

Our work is inspired by a simple and efficient UI of photo-editing applications [17] that allows a user to alter the appearance of an image through simple painting operations, such as brushing and flood-fill. We adopt a pen-tablet input device because it allows a quick and intuitive input method as used in the context of free surface modeling by [7]. We take the advantage of the feature of a detectable pen angle in recent products, e.g., *Wacom's Intuos2*. Pen angle is directly associated with the surface normal of a scene to achieve an intuitive input of surface normals. We provide a set of input methods to efficiently assign surface normals to the scene. The assigned surface normal is then used to estimate its surface reflectance and a more precise surface normal. Once the scene is represented by our data structure where each pixel encodes surface reflectance and surface normal, the scene can be re-rendered under different illumination conditions (Figure 1).

2. Our Approach

2.1. Overview

Our method consists of two steps: Interactive surface normal assignment and simultaneous estimation of reflectance and detailed surface normal.

In the first step, a surface normal is assigned via a set of input methods designed for this purpose. Three different types of brushes and a flood-fill tool are provided to facilitate the assignment process. Once the surface normal is assigned by a user, the second step uses the normal map for automatically and simultaneously estimating the surface reflectance property and a finer surface normal map. Two techniques are proposed to achieve single-view relighting. One is a smooth flood-fill tool for surface normal assignment, and the other is an automatic refinement of surface normals and reflectance.

Our contributions are as follows. We provide a set of tools to assign surface normals to a single image. The pen angle obtained from a pen-input device is used for intuitive assignment of surface normals. The system provides three types of brushes to allow users to intuitively paint surface normals. To facilitate surface normal assignment, a flood-fill tool is developed that fills in the specified area by smoothly propagating surface normals. We have also developed an algorithm to simultaneously estimate surface reflectance and a finer surface normal from a coarse hand-drawn surface normal. Our system permits relighting of a single photograph under different illumination conditions.

Figure 2. An input image is first segmented with our interactive segmentation method. Left: Original image, middle and right: segmentation with different thresholds.

Figure 3. Important boundaries can be explicitly specified with a user input stroke. (From left to right) Left: User-defined rough strokes, Middle: segmentation result without strokes, and Right: segmentation result with strokes. Important boundaries are well preserved in the right image.

2.2. Interactive Normal Editing

Labeling Multiple Photo-object Our method begins with labeling photo-objects that are of interest for relighting. Once the labels are assigned to photo-objects, the boundaries are used for the following surface normal editing process.

To achieve multi-object interactive segmentation, we have added an interactive control to the segmentation method used in [20]. The input image is first smoothed while preserving edges by a variant of the anisotropic diffusion and bilateral filtering algorithms. After smoothing, each pixel is assigned its own segment label. These labels are merged interactively given a user defined threshold defined by a RGB Euclidean distance between average colors of the segments. A user can interactively merge/split the segments by changing the threshold. Once a desired segment is obtained at a certain threshold, a user can mark the edge as an object boundary and continue the labeling process by changing the threshold. In this way, a user can

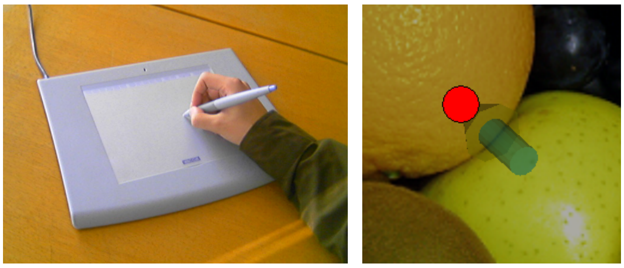


Figure 4. Pen angle is directly associated with the surface normal to allow intuitive normal painting. The cursor in the right image shows the angle of surface normal input.

efficiently label photo-objects with changing the segment boundaries (Figure 2).

We have also developed a tool to explicitly specify important boundaries by stroking. Given a stroke, the system automatically creates an area by dilating the stroke and calculating graphcut in the area using the method proposed by [12]. In the case the color gradient in the area is reasonably small, the algorithm tries to set a boundary along the user input stroke. In this way, important boundaries can be explicitly specified. Figure 3 illustrates the usefulness of boundary stroke.

Surface normal modeling Three different types of brushes and a flood-fill tool are designed for the interactive surface normal assignment. The basis of our UI is a painting UI as used in [17] for depth assignment. Our painting UI has three different brushes to facilitate the task of surface normal assignment. The first one is a *normal brush* with which a user can directly input a surface normal. As a key feature of the normal brush, we use a pen-tablet input device taking advantage of the ability of commercial pen-tablets providing information about angle of pen-input. Pen angle is directly associated with the surface normal in the image (Figure 4). This allows a user to intuitively assign surface normals in an image. The second brush is a *copy brush* that allows copying a surface normal of the specified point and paints areas with the copied surface normal regardless of the pen angle. The copy brush is useful when the same surface orientation exists at different locations, as often is the case with architecture. A user can also blur the assigned normal using a *blur brush*, which is useful to create smooth variations of the surface normal. The blur brush works as a Gaussian diffusion over the specified area of the normal map. For all brushes, the brush size can be interactively varied.

The three brushes provide an intuitive and efficient normal-input mechanism, however, it is still a time-consuming task when filling-in a large image area. To accelerate normal assignment, our system provides a *smooth flood-fill* tool that can fill up the selected image area by propagating sparse user input.

Smooth Flood-fill Tool To facilitate the surface normal assignment process, we developed a flood-fill tool that propagates sparse surface normal input to smoothly fill in the specified image area. We use intensity similarity measure to propagate the coarsely specified surface normal $\vec{N} = (N_x, N_y, N_z), ||\vec{N}|| = 1$ as Levin *et al.* used in the context of color propagation [11]. The propagation is bounded by the pre-defined label and user-specified area where intensity change is due to the variation in surface orientation. It is achieved by minimizing the energy functional J:

$$J(U) = \sum_{\vec{r}} \left(U(\vec{r}) - \sum_{\vec{s} \in A(\vec{r})} w_{\vec{r}\vec{s}} U(\vec{s}) \right)^2, \tag{1}$$

where $\vec{r}$ and $\vec{s}$ denote pixel labels, and $A(\vec{r})$ represents a set of adjacent pixels of $\vec{r}$. The weighting factor $w_{\vec{r}\vec{s}}$, which sums up to one, represents the similarity between $\vec{r}$ and $\vec{s}$, which is defined in [11] by

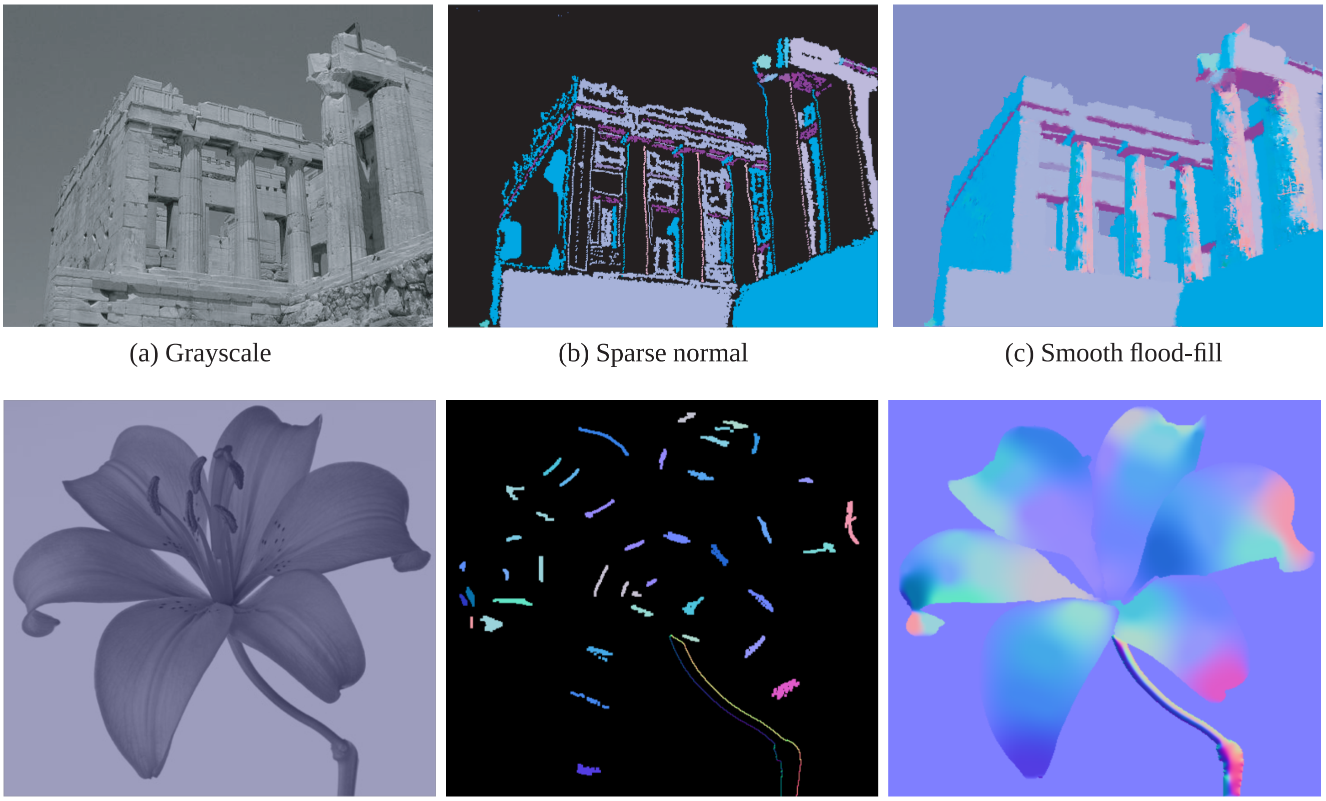
$$w_{\vec{r}\vec{s}} \propto 1 + \sigma_{\vec{r}}^{-2} (Y(\vec{r}) - \mu_{\vec{r}})(Y(\vec{s}) - \mu_{\vec{r}}), \tag{2}$$

where $Y(\vec{r})$ is the intensity at pixel $\vec{r}$, $\mu_{\vec{r}}$ and $\sigma_{\vec{r}}^2$ are the mean and variance of the intensities respectively in $N(\vec{r})$. Equation (1) is evaluated twice for $U \leftarrow N_x/N_z$ and $U \leftarrow N_y/N_z$, and the surface normal is re-normalized after computation.

The actual computation is performed by solving a sparse system of linear equations which is derived by taking partial derivative of Equation (1) by U. For propagating surface normal smoothly, $w_{\vec{r}\vec{s}}$ can be adjusted according to the image condition. For example, when $w_{\vec{r}\vec{s}}$ is proportional to 1.0, all the intensity information on the image are ignored and painted normal values are simply interpolated on the image coordinate. This mode is useful for a surface with a complex texture whose intensity variations are not directly related to normal variations. Figure 5 shows the results of our flood-fill method with and without intensity similarity measure. As shown in the figure, the assigned surface normal is smoothly propagated.

2.3. Refining Reflectance and Normals

Once the hand-drawn surface normal map is obtained, an automatic method for simultaneously estimating surface reflectance and a fine surface normal map is applied. The method takes an input image $I(\vec{r})$ and the user-assigned normal map $\vec{N}(\vec{r})$ as input. We assume that the scene is governed by a local lighting effect, more precisely, illumination can be approximated by a dominant directional lighting that is represented with a directional vector $\vec{l}$ and magnitude I_p, a small ambient lighting term I_a that approximates interreflections, and shadowed area is not dominant in the image.



(a) Grayscale (b) Sparse normal (c) Smooth flood-fill

(d) Grayscale (e) Sparse normal (f) Smooth flood-fill

Figure 5. (a, b, c) shows smooth flood-fill on Greek Building. (a) Grayscale image, (b) Coarse surface normal specified by a user, (c) result of smooth flood-fill using intensity similarity. (d, e, f) shows smooth flood-fill on Flower. (d) Grayscale image, (e) Coarse surface normal specified by a user, (f) result of smooth flood-fill when $w_{\vec{r}\vec{s}}$ is proportional to 1.0. The proposed interpolation method is used on petals. In (c, f), XYZ component of surface normal is represented by RGB channel respectively.

We use the following local illumination model in this method:

$$I(\vec{r}) = I_a k_a + I_p S_p\left(k_d\langle\vec{l},\vec{n}\rangle + k_s H_p\right), \tag{3}$$

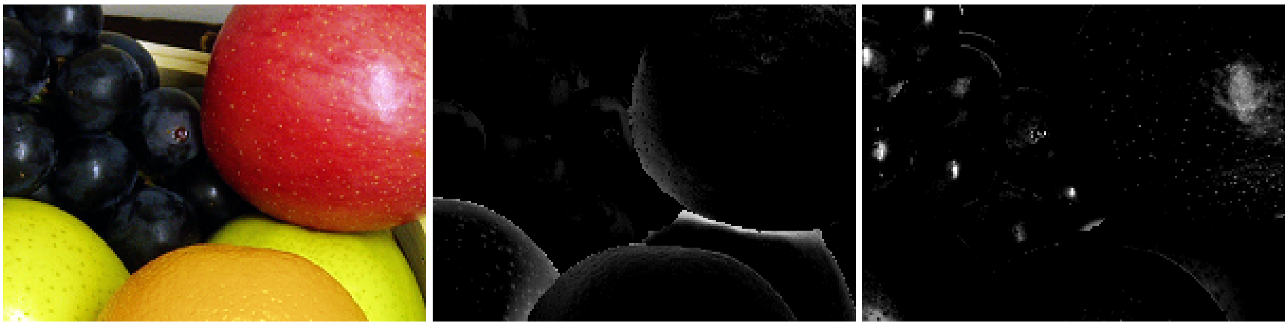
where k_a, k_d and k_s are coefficients to describe albedo of ambient, diffuse and specular reflection respectively, $\vec{n}$ is unit vectors of normal directions, $S_p(\vec{r}) \in [0,1]$ represents a shadowing term, and $H_p(\vec{r})$ is a term that governs intensity of specular highlight.

Throughout this paper, we only consider gray images to describe our idea. For color images, the same process is applied for each red, green, and blue components.

Light source estimation Our second step begins with estimating the dominant lighting direction $\vec{l}$ in the original scene. Traditionally, techniques from shape from shading [2] have been used to estimate the direction of the light source. Assuming a Lambertian image formation model [5], most of these techniques try to simultaneously recover both shape and the direction of the light source. In our case, the problem can be more constrained by the user-defined surface normal information and pre-defined label map which represents the regions of objects. Assuming that the area which shows diffuse reflection is dominant in each label i, lighting direction $\vec{l}$ can be estimated by first minimizing $E(\vec{l})$:

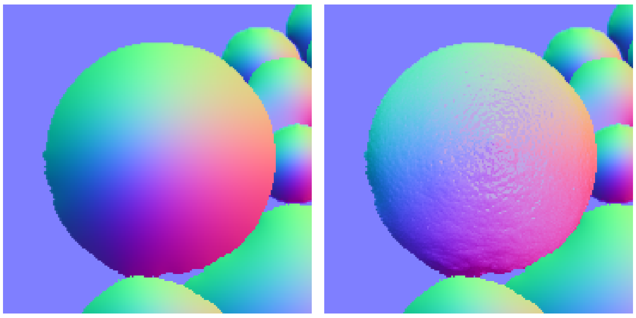
$$E(\vec{l}) = \sum_i \sum_{\vec{r}} (I(\vec{r}) - A_i - D_i\langle\vec{l},\vec{N}(\vec{r})\rangle)^2, \tag{4}$$

where $A_i = I_a\overline{k}_i$ is a constant ambient, $D_i = I_p\overline{k}_i$ is the diffuse component. $\overline{k}_i$ is a constant albedo in label i. $E(\vec{l})$ can be minimized by solving a linear system of equations which is derived by its first derivatives about A_i and D_i. Once A_i and D_i are determined, $\vec{l}$ is then estimated by searching for the minimum energy $E(\vec{l})$ as $\vec{l} = \arg\min\{E(\vec{l})\}$. In this way, lighting direction $\vec{l}$ is estimated. At the same time, A_i and D_i are assigned to each label.



(a) Input image (b) Shadow S_p (c) Highlight $I_p k_s H_p$

Figure 6. Result of shadow detection and highlight detection.



(a) Coarse surface normal (b) Refined surface normal

Figure 7. Result of surface normal refinement. Detailed surface normal is estimated with the proposed method.

Determining shadow and highlight Using the estimates A_i, D_i and $\vec{l}$, our method detects the shadowed and specular highlight areas by evaluating deviation from the diffuse reflection model as follows:

$$S_p(\vec{r}) = \begin{cases} 1, & if\ I(\vec{r}) > \overline{k}_i(I_a + t_s I_p) = A_i + t_s D_i \\ \frac{I(\vec{r}) - A_i}{t_s D_i}, & else\ if\ I(\vec{r}) > \overline{k}_i I_a = A_i \\ 0, & otherwise, \end{cases} \tag{5}$$

$$I_p k_s H_p(\vec{r}) = \begin{cases} I(\vec{r}) - A_i - D_i, & if\ I(\vec{r}) > A_i + (1 + t_h) D_i \\ \frac{(I(\vec{r}) - A_i - (1 - t_h) D_i)^2}{4 t_h D_i}, & else\ if\ I(\vec{r}) > A_i + (1 - t_h) D_i \\ 0, & otherwise, \end{cases} \tag{6}$$

where t_s and t_h are small positive thresholds that control the smoothness of the shadow and highlight. Figure 6 shows the result of shadow and highlight detection.

Refinement of Reflectance and Normals The hand-drawn surface normal is limited to representing rather coarse surface normals because it is still hard for a user to assign fine geometric details, e.g., bumps on a piece of fruit, such as an orange's surface. This method estimates

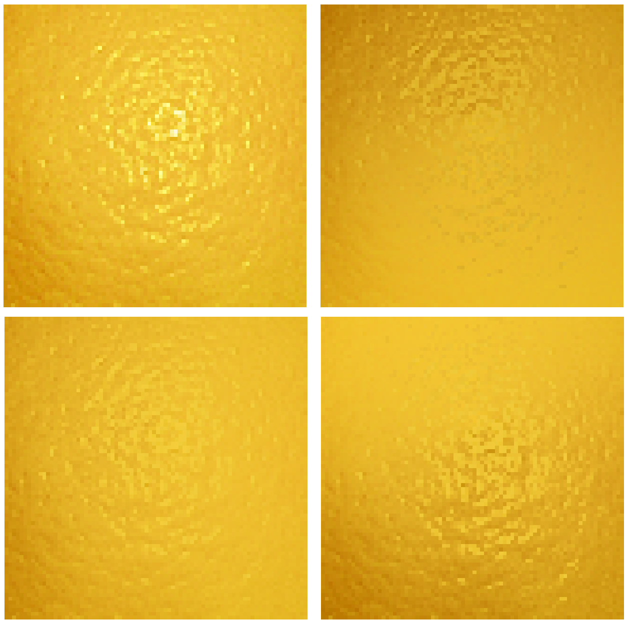


Figure 9. Close-up view of highlight in relighting result. The top-left image shows the original image, and the rest show relit results.

more precise surface normals and reflectance using an input image I and the hand-drawn coarse surface normal map $\vec{N}$.

Assuming that $\vec{l}$ is known and constant albedos $\overline{k}_i$ are obtained for labelled region i, we show that the surface reflectance and normal can be refined by the proposed refinement method. Due to the inaccuracy of the coarse surface normal $\vec{N}$ and constant albedo $\overline{k}_i$ in each label i, Eq. (3) becomes

$$e(\overline{k}_i, \vec{N}) = I(\vec{r}) - I_a \overline{k}_i - I_p S_p \left(\overline{k}_i \langle \vec{l}, \vec{N} \rangle + k_s H_p \right), \tag{7}$$

where e is the residue caused by the error in surface normal and surface reflectance. We use the residue e to refine both surface albedo $\overline{k}_i$ and normal $\vec{N}$. To determine the source

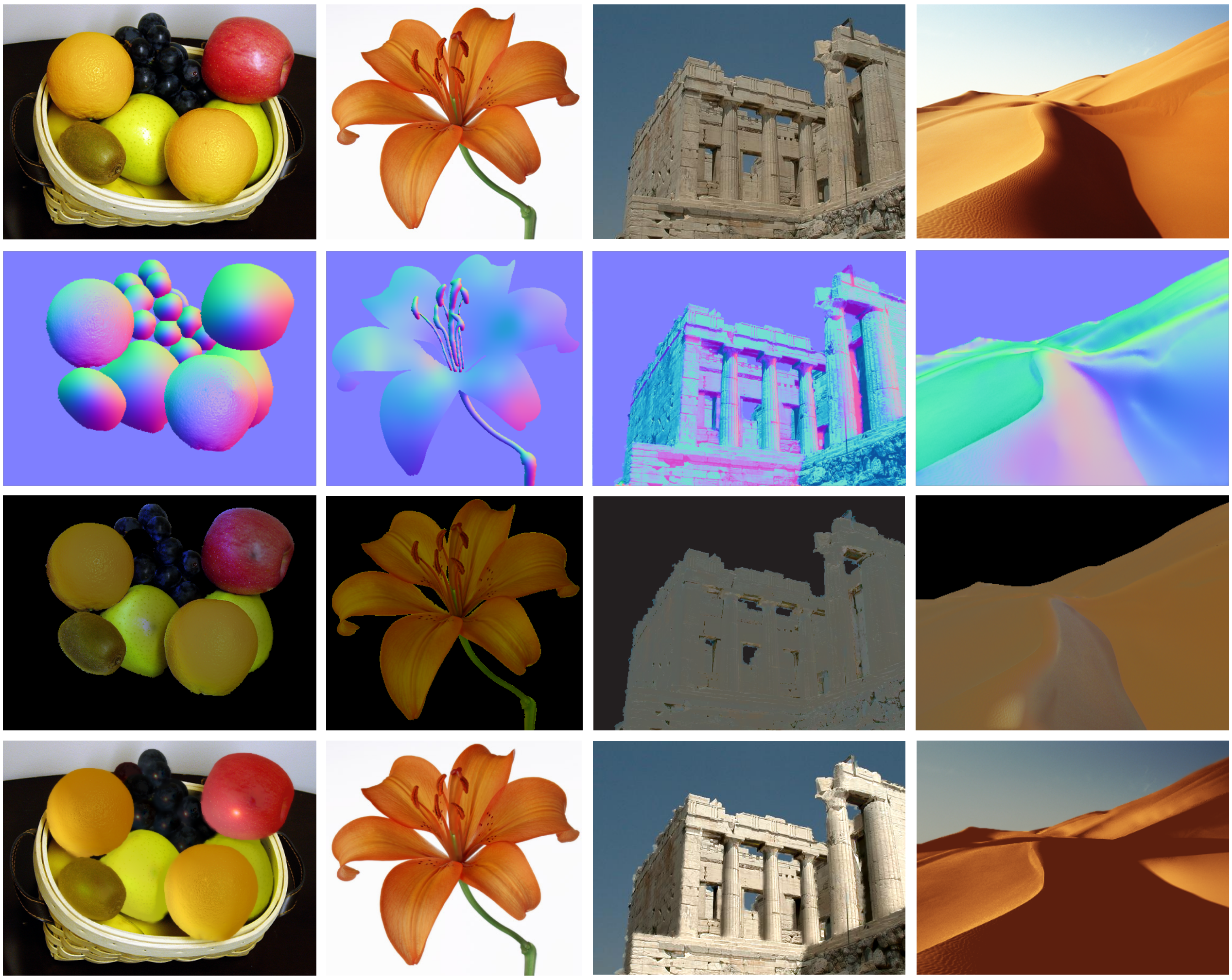


Figure 8. Result of single-view relighting. Top row shows original images, and the bottom row shows corresponding relighting results. The second and third row show the refined surface normal map and reflectance images respectively.

of the error, i.e., surface normal or reflectance, we propose to use a *chromaticity similarity* measure in the original image. In our method, the chromaticity similarity α between a pixel $\vec{r}$ and adjacent pixels $\vec{s} \in A(\vec{r})$ is evaluated in the specified image area as follows:

$$\alpha(\vec{r}) = \frac{1}{n} \sum_{\vec{s} \in N(\vec{r})} \text{Similarity}\big(\vec{C}(\vec{r}), \vec{C}(\vec{s})\big), \qquad (8)$$

where $\vec{C}(\cdot) = (R,G,B)/(0.3R + 0.6G + 0.1B)$ represents chromaticity and n represents the number of adjacent pixels $\vec{s}$. The function $\text{Similarity}(\cdot,\cdot)$ measures the distance between two chromaticities. In our case, we use the inner product of normalized vectors to measure the chromaticity similarity as $\text{Similarity}(\vec{a},\vec{b}) = \langle \vec{a},\vec{b} \rangle / (||\vec{a}||||\vec{b}||)$.

The residue e is then distributed to surface normal $\vec{N}$ and reflectance $\overline{k}_i$ weighed by α for each pixel by plugging $e(k,\vec{N}) = (1-\alpha)e(\overline{k}_i,\vec{N})$ and $e(\overline{k}_i,\vec{n}) = \alpha e(\overline{k}_i,\vec{N})$ into Equation (7) and solving respectively for each k and $\vec{n}$. To obtain the refined surface normal $\vec{n}$ from $e(\overline{k}_i,\vec{n})$ with Equation (7), the coarse normal $\vec{N}$ is rotated on the plane spanned by $\vec{l}$ and $\vec{N}$. There are two solutions of $\vec{n}$, and the solution of $\vec{n}$ which has a smaller deviation from $\vec{N}$ is taken by evaluating $\vec{n} = \text{argmin}\langle \vec{n},\vec{N} \rangle$. In this way, fine reflectance and normal map which are necessary for relighting are obtained. Figure 7 shows the result of surface normal refinement. Detailed surface normal of bumps of the orange is estimated with the proposed method.

3. Results

To evaluate the performance of the proposed method, we have conducted extensive experiments on various pho-

Figure 10. Adjustment of photometric consistency in superimposition application. The foreground (the girl in the top-left image) is superimposed on the different background(bottom and right image). The illumination condition of foreground (girl) is aligned to that of the background with our relighting method.

tographs. We have developed a complete system to achieve the goal and all the relighting results are rendered without any other photo-editing tools. Figure 8 displays the result of proposed relighting method. The images in the top row show the original input images, the second row shows the normal maps used for relighting, the third row shows the estimated reflectance images, and the relighting results are shown in the bottom row. To produce the relighting result, Equation (3) is used for rendering.

For surface normal assignment, like existing 3D modellers and image-based modeling methods that allow a user to assign depth values, in practice, our method is limited to rather simple scenes where image segmentation is feasible. As mentioned in [17], image segmentation usually takes many hours. The editing time of our results is also roughly a few hours on average. Ignoring the image segmentation process, we found that the normal assignment is much easier than existing methods that assign full 3D or depth by 3D modellers and depth painting method.

At the boundaries of objects where surface normals are not smooth, sometimes small artifacts due to the inaccuracy of the surface normal boundary are observed. To handle this problem, pixels that have a significant difference in chromaticity between themselves and the adjacent pixels compared to that in the original image are replaced with the smooth interpolation of adjacent pixel values weighed by the chromaticity similarity in the original image. To render highlights, we used a simple Phong model in the experiment: $I_p k_s H_s(\vec{r}) = I_p k_s \left(\vec{v} \cdot \vec{m}(\vec{r})\right)^{n_i}$, where $\vec{v}$ and $\vec{m}$ are unit vectors of viewing and mirror reflection directions. n_i is a user-defined shininess factor. In this way, H_s is computed for each pixel under a new lighting condition. Figure 9 shows the close-up view of the orange surface in the fruit scene. As we can see clearly in the image, highlight and small bumps on the orange surface are naturally rendered.

Since the scene structure is represented by surface normals in our method, it is inherently difficult to generate cast shadows that are not local lighting effects. However, when the scene surface is smooth, we can compute cast shadow by taking the integral of surface normals. We have adopted a method proposed by Robertson *et al.* [18] to achieve cast shadow rendering. The relighting results with cast shadow is shown on the bottom row of Figure 8.

Our method is useful for illumination alignment for photo-object superimposition (Figure 10). With our relighting method, photometric consistency between foreground and background is adjusted and visible in the illustration. In this example, we superimposed the foreground onto the background image with adjusting the illumination condition of the foreground.

4. Conclusion and Future Work

This paper presents a new approach to single-view relighting. We provide a set of input methods for interactive assignment of surface normal. An automatic method for estimating surface reflectance and detailed surface normal from the coarse surface normal input is also proposed. The proposed method uses surface normals to represent the scene geometry in an image, which is sufficient for relighting where local illumination effect is dominant. Two key techniques are developed; (1) Smooth flood-fill tool which propagates sparse normal input to smoothly fill-in the specified image area, and (2) Algorithm to refine surface reflectance and normal with the input image and the hand-drawn coarse surface normal. The applicability of the method is evaluated by relighting a variety of photographs and the application of illumination adjustment in photo superimposition.

There are many promising directions for future research. One is to investigate more efficient tools for surface normal assignment. Our current system provides normal modeling tools based on detectable pen angle of a pen-tablet device. It may be also possible to develop UIs to assign surface normals in a copy-and-paste manner by referencing well-known shapes such as cube, sphere, etc. We are also interested in other interactive tools such like Teddy system [7].

To simplify the problem, we limit our method to apply to scenes which are governed by a dominant directional lighting with an ambient lighting and are composed of surface materials that can be approximated by the Phong's model. Extending the method to work with more complex lighting models and BRDF models is the other direction of future research, which also makes the refinement algorithm of reflectance and normal maps more complex but may produce more physically correct relighting results.

References

[1] S. Boivin and A. Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *SIGGRAPH '01*, pages 107–116. ACM Press, 2001.

[2] M. J. Brooks and B. K. P. Horn. Shape and source from shading. In *Int. Joint Conf. Aritficial Intell.*, pages 932–936, 1985.

[3] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. *Int'l J. of Computer Vision*, 40(2):123–148, 2000.

[4] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics*, 30(Annual Conf. Series):11–20, 1996.

[5] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1990.

[6] M. Fuchs, V. Blanz, and H.-P. Seidel. Bayesian relighting. In *Rendering Techniques 2005: Eurographics Symposium on Rendering*, pages 157–164, 2005.

[7] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99*, pages 409–416. ACM Press/Addison-Wesley Publishing Co., 1999.

[8] S. F. Johnston. Lumo: illumination for cel animation. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 45–ff, New York, NY, USA, 2002. ACM Press.

[9] S. B. Kang. Depth painting for image-based rendering applications. *Tech. report, CRL, Compaq Cambridge Research Lab, 1998*, 1998.

[10] J. J. Koenderink. Pictorial relief. *Phil. Trans. of the Roy. Soc.: Math., Phys, and Engineering Sciences*, 356(1740):1071–1086, 1998.

[11] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, 2004.

[12] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[13] Z. Lin, T.-T. Wong, and H.-Y. Shum. Relighting with the reflected irradiance field: Representation, sampling and reconstruction. *Int. J. Computer Vision*, 49(2-3):229–246, 2002.

[14] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 519–528. ACM Press, 2001.

[15] S. R. Marschner and D. P. Greenberg. Inverse lighting for photography. In *Proceedings of the Fifth Color Imaging Conference, Society for Imaging Science and Technology*, Nov. 1997.

[16] V. Masselus, P. Peers, P. Dutré, and Y. D. Willems. Relighting with 4d incident light fields. *ACM Trans. Graph.*, 22(3):613–620, 2003.

[17] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *SIGGRAPH '01*, pages 433–442. ACM Press, 2001.

[18] P. K. Robertson. Spatial transformations for rapid scan-line surface shadowing. *IEEE Computer Graph. Appl.*, 9(2):30–38, 1989.

[19] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz. Single view modeling of free-form scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 990–997. IEEE Computer Society, 2001.

[20] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004.