Synthesizing Oil Painting Surface Geometry from a Single Photograph

Wei Luo[1*],   Zheng Lu[2*],   Xiaogang Wang[1],   Ying-Qing Xu[3],

Moshe Ben-Ezra[4],   Xiaoou Tang[1], and Michael S. Brown[2]

[1]The Chinese University of Hong Kong   [2]National University of Singapore   [3]Tsinghua University   [4]Microsoft Research Asia

{lw010|xtang}@ie.cuhk.edu.hk   {luzheng|brown}@comp.nus.edu.sg   yqxu@tsinghua.edu.cn   mosheb@microsoft.com

Abstract

We present an approach to synthesize the subtle 3D relief and texture of oil painting brush strokes from a single photograph. This task is unique from traditional synthesize algorithms due to its mixed modality between the input and output; i.e., our goal is to synthesize surface normals given an intensity image input. To accomplish this task, we propose a framework that first applies intrinsic image decomposition to produce a pair of initial normal maps. These maps are combined into a conditional random field (CRF) optimization framework that incorporates additional information derived from a training set consisting of normals captured using photometric stereo on oil paintings with similar brush styles. Additional constraints are incorporated into the CRF framework to further ensures smoothness and preserve brush stroke edges. Our results show that this approach can produce compelling reliefs that are often indistinguishable from results captured using photometric stereo.

1. Introduction

One of the aesthetic qualities associated with oil paintings is its dynamic nature to light. This is attributed, in part, to the 3D relief of the brush strokes. Artists have exploited the ability of oil paint to produce 3D relief in the form of thick and distinct brush strokes to enhance the expressiveness of the subject matter. This technique is most notable in the works of Impressionist and post-Impressionists. In addition, oil paintings are often varnished as a final step resulting in a specular surface that helps to accentuate the texture and relief of the brush strokes. Not surprisingly, museums and art galleries take great care when lighting and positioning oil painting artwork for their patrons' viewing experience.

Outside a museum setting, the most common access to

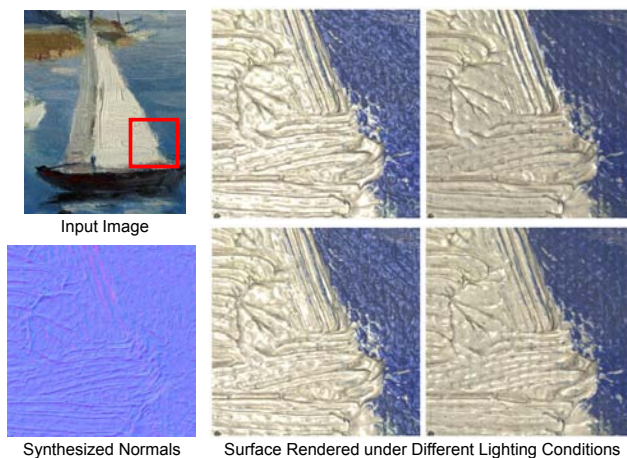*Wei Luo and Zheng Lu contributed equally to this paper.

Figure 1. Left: input image of an oil painting. Bottom: our synthesized normals. Right: zoomed-in region of the reconstructed surface using our synthesized normals with recovered surface albedo. The four images show the synthesized surface rendered under different lighting directions.

oil painting artwork is through photographs. A photograph of an oil painting, however, can only capture the object under a static lighting condition and cannot capture the rich lighting effects attributed to the painting's relief.

In this paper, we propose a framework to reconstruct the surface relief of an oil painting from a single photograph together with a training set of similar oil paintings whose normals have been estimated using photometric stereo. Our framework first applies intrinsic image decomposition on the input to acquire the shading image from which a pair of base normal maps are estimated. These normals and the original input are integrated into a conditional random field (CRF) that is able to combine the learned texture-normal relation from the training data to produce plausible results in the inference process. Moreover, long-range priors can be incorporated into the CRF to take advantage of specific structures of the painted surface. From this synthesized sur-

face normals a 2.5D height field is generated that can produce a visually realistic reconstruction that is hard to distinguish from that captured by high resolution 3D scans (see Figure 1).

The remainder of this paper is organized as follows: Section 2 discusses related work; Section 3 describes the details of our framework; Section 4 presents our results. A summary of this work is presented in Section 5.

2. Related Work

We are unaware of prior work attempting to synthesize a 2.5D relief of an oil painting from a single photograph, however, several works related to this task are discussed in the following.

Shape from shading Shape from shading (SfS) [14] is a well-known technique that recovers 3D shapes by using shading information in an image with known lighting direction and surface reflection model. In the case of oil paintings, directly applying SfS does not produce satisfactory result. This is due to several reasons. First, one needs to remove albedo effects before applying SfS by using techniques such as intrinsic images decomposition [3, 23]. However, a clean separation of shading and reflectance components is usually hard to obtain [12]. Second, even with an accurate lighting direction estimation, one intensity value in the image may correspond to different normals, leading SfS to recover a wrong surface [15]. In the case of oil painting surfaces, this one-to-many ambiguity may wrongly recover a convex stroke as a concave region. Our method, however, uses an initial normal map and its mirror reflection estimated from shading as a guidance for searching the best-matched normals in the training set. This makes our results less sensitive to the mentioned problems.

Bump mapping and image stylization Bump mapping is a well established technique used to enhance the lighting effect of a 3D surface by permuting surface normals (e.g., [5, 19]). The map controlling the normal perturbation (i.e., the bump map) can be computed using the gradients of an input image that has a distinct texture. This approach typically produces noisy results for oil paintings, as the content of the painting can be indistinguishable from the stroke relief. Image stylization techniques (e.g., [13, 18]) are able to impart artist styles onto input images, videos and renderings of 3D models. These approaches typically decompose the input into features that can be parameterized to guide synthetic brush strokes or particles that simulate brush strokes. These approaches are not designed to simulate the 2.5D relief of the strokes themselves. In addition, the input is assumed to be significantly different than the desired stylized output. Our goal, however, is not to change the style of the input, but instead to synthesize a 2.5D surface with a similar look and feel.

Imaging relighting, surface modeling, and manipulation Other related work involves those targeting image relighting (e.g., [6, 28, 27]) and surface modeling and manipulation (e.g., [11, 10]) using surface normals. While sharing a commonality of working with normals, these interactive approaches would be impractical for specifying individual paint strokes. As such, our approach performs surface reconstruction through surface normal synthesis from training data. Hence, our work is most related to techniques designed for constrained texture synthesis (e.g., [7, 20]), and learning based super resolution (e.g., [8, 24, 22, 25]). Like these techniques, our approach exploits a training set of exemplar patches and uses a learning based method. The main difference is our focus on synthesizing normals versus pixel values.

3. Framework

This section briefly describes our oil painting surface synthesis framework. The overview of the framework and training data collection are described first, followed by the details of normal synthesis and the surface reconstruction algorithm. Implementation details are given at the end of the section.

3.1. Overview

An overview of our framework is shown in Figure 2. The user provides an image X which will be used to recover the shading image S and reflection image R through intrinsic image decomposition. The reflectance R will be used as texture map on the final rendered surface. The shading image, S, is used to estimate a pair of initial normal maps, N_1 and its reflection N_2. We refer to N_1, N_2 as the *base normals* to differentiate from the final synthesized surface normals N. In addition, to help tune the training data, our framework requires the lighting direction of the input image, which is estimated from the input image itself with the user's help. We then indirectly compute the height field H by synthesizing the surface normals N. Our normal synthesis is a learning based approach exploiting a training set of exemplar patches whose placement is guided by X, N_1 and N_2. A CRF is formulated to compute the patch placement onto the output. Through the integration of the synthesized normals, our method is able to synthesize a convincing 2.5D geometry that appears smooth and visually plausible.

3.2. Data Collection

Capture Our training data consists of surface normals and associated albedos obtained from oil paintings (provided by an art student) that cover the variation of strokes of input paintings. Note that the training set is not required to have comparable aesthetic quality with the input. The training data is also assumed to be of approximately the same pixel
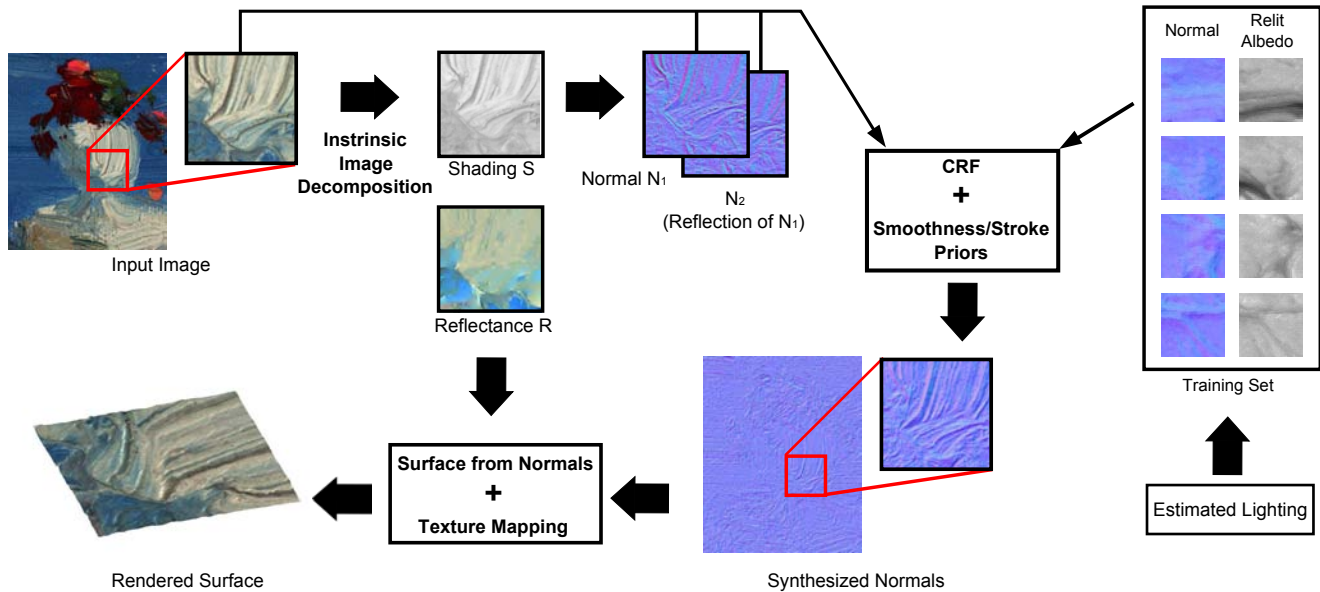
Figure 2. Overview of our framework.

resolution as the input image. A high-resolution camera [4] with four light sources is used to perform the photometric stereo described in [26]. Polarizing filters are placed on the lighting sources and over the camera lens to eliminate specular reflections. This procedure generates the surface normals and albedo from the example paintings. These normal-albedo pairs are divided in to patches of 32×32 pixels. The patches serve as the basis for our training data. We typically choose five to six regions (600×600) per painting that capture the variations of paintings such as various types of strokes, stroke sizes, stroke direction, and stroke intersections. For our implementation, there are a total of 48000 patches in the training data set.

Relighting the training set Our training data represents the oil painting in a manner that is decoupled from the environment illumination. However, a photograph of an oil painting will typically exhibit a noticeable illumination caused by a directional light. We can therefore tune our training data to better match the input image by estimating a lighting direction that matches the illumination in the photograph with user assistance. Given a rough lighting direction specified by the user, we estimate the lighting direction L using statistics of the shading image and oil painting surface normals.

Specifically, we compute the distributions of normals $p(n_x, n_y)$ from the training data. Our experiments suggest that even very distinct oil paintings exhibit consistent distribution of surface normals. Given the input photograph X, we can compute the distribution of its shading $p(S)$ from the shading image S (see next section for how to compute S). Computing $I/\rho = L \cdot (n_x, n_y, n_z)^T$ with the correct lighting direction L should yield shading direction that

resembles $p(S)$. We divide our solution space (the hemisphere of unit circle) into four equal parts (quadrants), top left, top right, bottom left, and bottom right. The user only needs to specify which quadrant the lighting direction belongs to based on shading cues in the painting. We then estimate L by dividing the specified quadrant in the solution space into 900 bins and find the best bin. Our experiments show a less than 5 degrees' error on average in lighting estimation if the user provides the right quadrant.

From L we reconstruct a new training image patch P', such that $P'(i) = \rho L \cdot n$, where ρ and n are the corresponding albedo and normal at pixel i.

3.3. Obtaining Intrinsic Images and Base Normals

Base normals are estimated as follows. We first recover the shading image using intrinsic image decomposition. We then estimate a normal map and its reflection.

Intrinsic image decomposition We obtain the intrinsic images from the input image X by adopting the scheme proposed by Tappen *et al.* [23], which takes advantage of both the color information and gray-scale pattern. In particular, by using color and pattern features, we first find and classify edges in the input image into those caused by shading and those caused by albedo change. A Markov random field (MRF) is then used to propagate the information over the entire image. We briefly describe our procedures here; for details, see [23].

To classify edges in the input image, we use two types of features, color and pattern features. Color features are extracted based on the observation that the intensity changes due to shading should affect three color channels propor-

tionally. Specifically, by normalizing the (R, G, B) triplet at each pixel to unit vector c, we can define the color feature as $F_c = \arccos(c \cdot \hat{c})$, where $\hat{c}$ is the average color vector in neighborhood of c. In our implementation, we use five scales of neighborhoods sizes, generating five features.

Pattern features are extracted to capture the fact that shading edges exhibit different shading patterns from albedo edges. We define the pattern features as $F_g = I_e * w$, where I_e is the patch centered at the edge e, and w is a local linear filter. In other words, pattern features are extracted by applying a set of linear filters to the neighborhood of an edge. In our implementation, we use Gabor filters with ten orientations and five scales. We found that stroke structures in the oil painting are well captured using these filters. In total, 50 pattern features are generated.

We then classify the edges into two classes, associated with shading or associated with albedo, by employing the above 55 features (5 color + 50 patterns) as weak classifiers in AdaBoost [9]. After the classification, each edge is treated as a node in an MRF with two possible states, corresponding to the two classes. Finally, we learn the potential between edges and propagate evidence along the network. For an oil painting, the patterns of the two edge classes are very consistent, yielding a classification accuracy of more than 90%. Given the labels of all the edges, the intrinsic images, both shading component S and reflectance component R can be recovered. See Figure 3 for an example of intrinsic images.

Estimating base normals The shading image S is used to estimate the base normal map N_1 and its reflection N_2. Specifically, by using the gradients of tS as an approximation of he surface gradient field of the input image, we construct surface M from this gradient field using the technique in [1]. The base normal maps N_1, N_2 are then obtained from M and its mirror reflection (yeilding N_2) with respect to the plane perpendicular to lighting direction L. Both base normals, N_1 and N_2, are in the CRF likelihood formulation to help guide the synthesis process as discussed in the next section. See Figure 3 for an example of the base normal maps.

3.4. Normal Synthesis through CRF Formulation

We formulate the synthesis of the normals map N as an inference of a CRF (see Figure 4(a) for our graphic model). The input image X is broken into a set of regularly sampled overlapping patches X_p. Since 3D structure of an oil painting is mostly related to painting strokes, we generate small regions of the input image, referred to as *sub-stroke*, using the superpixel technique [21]. We use both patches and sub-strokes as neighborhood structures for CRF. A typical sub-stroke may contain hundreds of patches. Our algorithm infers the conditional probability distribution over the patch label assignment from the training data given the observed
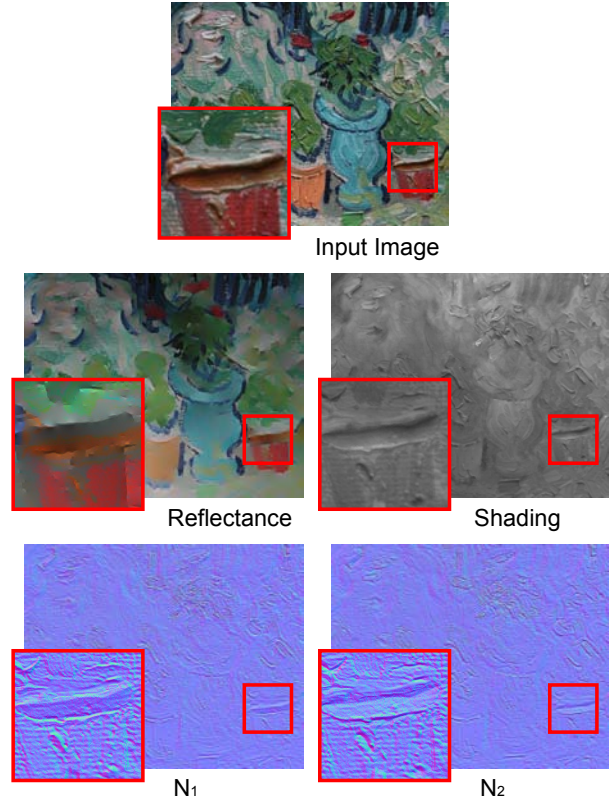

Figure 3. Top row: input image. Middle row: reflectance image (left) and shading image (right). Bottom row: base normals N_1 (left) and its reflection N_2 (right). Zoomed-in regions are also shown.

X.

We formulate our potential function for minimization based on the following considerations: 1) the synthesized normals should be similar to corresponding normals from the training data when their relit images are similar; 2) sub-stroke regions of the reconstructed surface should be less concave; 3) the synthesized normal should be smooth within and at the boundary of brush strokes. Figure 4 (b) shows an example of patches and sub-strokes. Our formulation ensures that sub-stroke a and b in Figure 4 (b) are compatible with each other and all the patches along the boundary, such as patch p in Figure 4, are smooth.

Our potential function is formulated as follows (note the dependency on the observation X is omitted for simplicity):

$$
\begin{aligned}
\Psi(Y, Z) = & \sum_p \phi_p(Y_p) + \lambda_1 \sum_p \sum_{q \in \Omega_p} \psi_{p,q}(Y_p, Y_q) \\
& + \lambda_2 \sum_s \zeta_s(Z_s) + \lambda_3 \sum_s \sum_{t \in \Gamma_s} \eta_{s,t}(Z_s, Z_t) \\
& + \lambda_4 \sum_s \sum_{t \in \Gamma_s} \theta_{s,t}(Z_s, Z_t, \{Y_{s,t}\}),
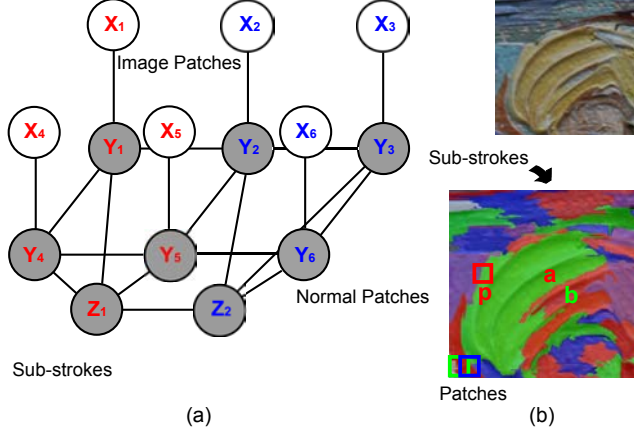\end{aligned} \tag{1}
$$

Figure 4. (a) Graphic model of our framework. (b) Top: a portion of an input image. Bottom: color squares are examples of patches and color coded regions are examples of sub-strokes obtained using super-pixel technique.

where ϕ_p and ζ_s are the data costs, $\psi_{p,q}$, $\eta_{s,t}$ and $\theta_{s,t}$ are the smoothness costs, and λ_i ($i \in \{1,2,3,4\}$) are weighting factors. The details of our formulation are described in the following.

Patch and sub-stroke labels In our formulation, the terms p, q represent patch locations, s, t are sub-stroke locations, and Ω_p is the 4-connection neighborhood of p. The term Γ_s is the set of neighboring sub-stroke locations of s. The term Y_p is the patch label at p. The goal of our method is to paste a set of candidate patches onto the output that satisfies all the constraints. Hence, the label space for Y is the set of all the patches in the training data. The term Z_s is the binary label at s indicating whether the region satisfies base normals N_1 or N_2:

$$Z_s = \begin{cases} 1 & \text{if normal is from } N_1 \\ 0 & \text{if normal is from } N_2. \end{cases} \quad (2)$$

The terms Y_q and Z_t are defined similarly.

Data cost The term ϕ_p measures how good a patch label assignment is and is defined as follows:

$$\phi_p(Y_p) = \|\nabla X'_p - \nabla X_p\|^2 + \mu\|N'_p - N_p\|^2, \quad (3)$$

where ∇X_p is the gradient of the input image at patch location p, $\nabla X'_p$ is the gradient of the relit image component of a candidate training patch in the training data, N'_p is the normal component of the candidate patch, and N_p is normals from the base normals at p and is ether from N_1 or N_2 depending on the Z label of the sub-stroke containing the patch location. The scalar μ is a weighting factor.

The term ζ_s measures the concavity of the sub-stroke located at s. We observe that the sub-strokes are mostly convex or flat in general (see Figure 4(b)). This is because the

natural of how oil paint is applied to a flat canvas resulting in more convex shapes than concave. As a result, the less concave the region is, the more likely the normals of the region is correct. The term ζ_s is defined as follows:

$$\zeta_s(Z_s) = \begin{cases} -C(M_1) & \text{if } Z_s = 1 \\ -C(M_2) & \text{if } Z_s = 0, \end{cases} \quad (4)$$

where M_1, M_2 are the 3D meshes at sub-stroke s integrated from N_1, N_2, and $C(\cdot)$ is the concavity feature adopted by [17].

For mesh M, we compute its convex hull H_M and find the *bridges* and *pockets* of M. Bridges are convex hull facets that connect non-adjacent vertices of M; pockets are the portion of the boundary M that is not on the convex hull boundary H_M. See [17] for the detailed definition of bridges and pockets. We associate each pocket x with a unique bridge β_x, and define concavity measure $concave(x) = dist(x, \beta_x)$, which is the distance between them. $C(M_1)$ is the sum of such measures for all pockets in M_1 divided by its area. $C(M_2)$ is defined similarly.

Smoothness cost The term $\psi_{p,q}$ captures how well the normals of two patches agree with each other, and is defined as follows:

$$\psi_{p,q}(Y_p, Y_q) = \|N'_p - N'_q\|^2, \quad (5)$$

where N'_p, N'_q are the normals components of the two candidate training patches at location p, q. This term penalizes the incompatibility of the normals between two neighboring patches.

The term $\eta_{s,t}$ captures how well the relit images of two patches agree with each other directionally, and is defined as follows:

$$\eta_{s,t}(Z_s, Z_t) = \begin{cases} \|T(X'_s) - T(X'_t)\|^2, & \text{if } Z_s = Z_t \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where X'_s, X'_t are relit image component of the candidate sub-strokes located at s, t. The candidate sub-stroke is formed by candidate patches that are located within the sub-stroke region. The 2D structure tensor direction of a sub-stroke region is defined as the eigenvectors of the structure tenser matrix, which is in the form of:

$$T(\cdot) = \sum_{i \in s} \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}, \quad (7)$$

where g_x, g_y are 2D image gradients of each pixel in a sub-stroke. This term penalizes directional discontinuity of neighboring sub-strokes which share the same Z label.

Algorithm 1 Minimizing potential Ψ

Initialization: set all Z label to 0. For each patch p, set Y_p to the patch in the training data returned by 1 Nearest Neighbor search using N_p.

Till convergence:

1. Minimize

$$\Psi_1(Y \mid Z) = \sum_p \phi_p + \lambda_1 \sum_p \sum_{q \in \Omega_p} \psi_{p,q}$$
$$+ \lambda_4 \sum_s \sum_{t \in \Gamma_s} \theta_{s,t},$$

   conditioned on X and Z using Loopy Belief Propagation.

2. Minimize

$$\Psi_2(Z \mid Y) = \lambda_2 \sum_s \zeta_s + \lambda_3 \sum_s \sum_{t \in \Gamma_s} \eta_{s,t}$$
$$+ \lambda_4 \sum_s \sum_{t \in \Gamma_s} \theta_{s,t},$$

   conditioned on X and Y using Loopy Belief Propagation.

Return: Labels Y_p. A best cut (see Section 3.5) at each overlapping region is found between patches to ensure smoothness.

The term $\theta_{s,t}$ captures how well the normals of two sub-strokes agree with each other, and is defined as follows:

$$D_p = \min \left(\|N_p' - N_1 p\|^2, \|N_p' - N_2 p\|^2 \right) \quad (8)$$

$$\theta_{s,t}(Z_s, Z_t, \{Y_{s,t}\}) = \begin{cases} \sum_{p \in \{Y_{s,t}\}} D_p, & \text{if } Z_s \neq Z_t \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where $\{Y_{s,t}\}$ is the collection of patches that belong to both sub-stroke s and t, and $N_1 p$, $N_2 p$ are two base normals at p. This term is used to smooth normals between neighboring sub-strokes. The idea is that the patches at the sub-stroke boundaries should share the same Z label and should be similar to N_1 or N_2.

Optimization To minimize our potential function shown in Equation 1, we find the label assignments for patches and sub-strokes by alternatively minimizing the two potential functions using Loopy Belief Propagation [29], as stated in Algorithm 1. Typically in our experiments, convergence is obtained after 3 to 4 iterations.

3.5. Implementation Details

We use K nearest neighbor search (KNN) to find patch candidates in the training data where K is chosen to be 40. We employ PatchMatch [2] to speed up the KNN search, since in our case, both the input image and the training set contain densely sampled patches. As a result, each KNN search takes 0.02s on our training set. For a 2000×2000 image, it takes 15 minutes in searching and 5 minutes in

inference. Graphcut [16] is used to find the best cut in the overlapping region between neighboring patches. However, in our case, we use normals instead of using intensity or colors. As the last step of our framework, the height field is reconstructed from the synthesized normals using the technique presented in [27].

4. Results

Results from our framework are shown in Figure 5, Figure 6, and Figure 7. Figure 5 shows our results for an oil painting that has also been captured using photometric stereo. We show two zoomed-in regions of the input image, surface normals obtained from photometric stereo, our synthesized surface normals, and zoomed-in regions of the reconstructed surface height maps. We also show a zoomed-in region of reconstructed surface (with texture map) rendered under different lighting conditions. The input image and surface normals from photometric stereo of this example are captured in the same way as described in Section 3.2. We are able to synthesize plausible normal map that creates very convincing texture-mapped results.

To numerically evaluate the synthesized normals, we calculate the mean square root error (MSE) between normals from photometric stereo and our result as well as the base normals N_1, N_2. The MSE of our recovered result is 0.07, implying a less than 5 degrees' error on average, whereas the MSE of base normal maps N_1, N_2 are 0.37 and 0.52 respectively, implying a more than 20 degrees' error on average for the rough approximation made directly from the shading image S.

We also relight region A and B in Figure 5 to evaluate errors from lighting estimation. We use 18 different lighting directions at least 30 degrees away from frontal light. The MSE of the recovered normals is below 6 degrees' error on average. For lighting directions as far as 60 degrees from frontal, average MSE is below 9 degrees. Our algorithm works in a copy-and-paste manner and thus is able to tolerate moderate lighting estimation errors.

Figure 6 shows our results obtained from an input image of Van Gogh's work. We show the input image, our synthesized normals and reconstructed surface height map with color coding. We also show several zoomed-in regions of reconstructed surface rendered with/without texture map (the reflectance image R). As shown by this example, our framework is able to produce very plausible surface from a single photograph, making it possible to simulate the aesthetics of the painting outside a museum setting. With the painting relief produced by our technique, users can change the illumination and viewing directions.

Finally, Figure 7 shows our reconstructed surface from the image of a lost artwork, Van Gogh's *A Painter on His Way to Work*. This enables the user, especially artists, to enjoy the 3D relief by viewing the artwork under different
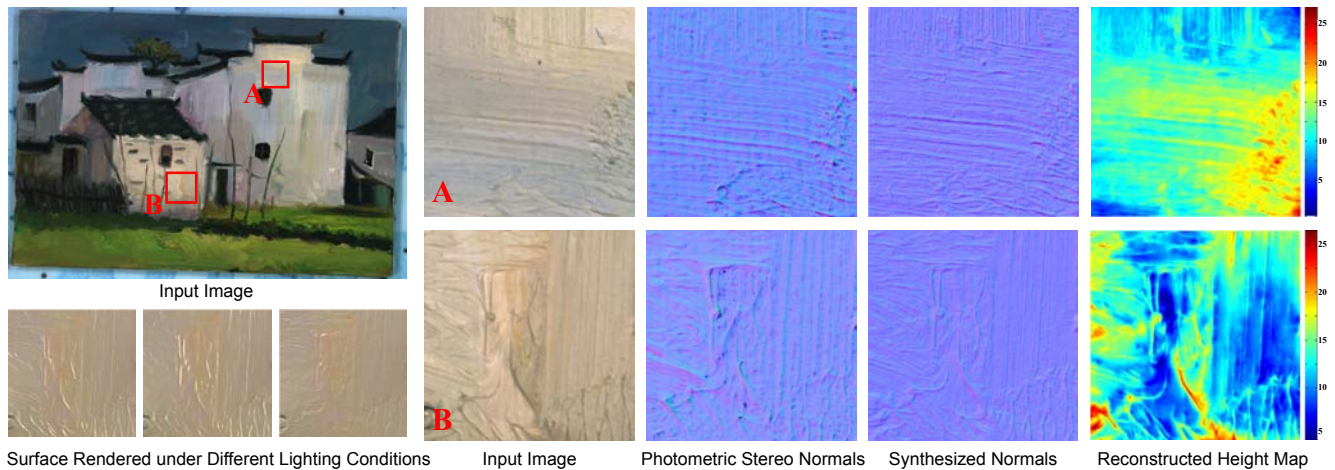
Figure 5. Example: house. Top left: input photograph. Bottom left: reconstructed surface from our synthesized normals texture mapped with recovered reflectance image and rendered under different lighting conditions. Right: two zoomed-in regions (A and B), from left to right: input image, normals from photometric stereo, our synthesized normals, and reconstructed surface height map with color coding.
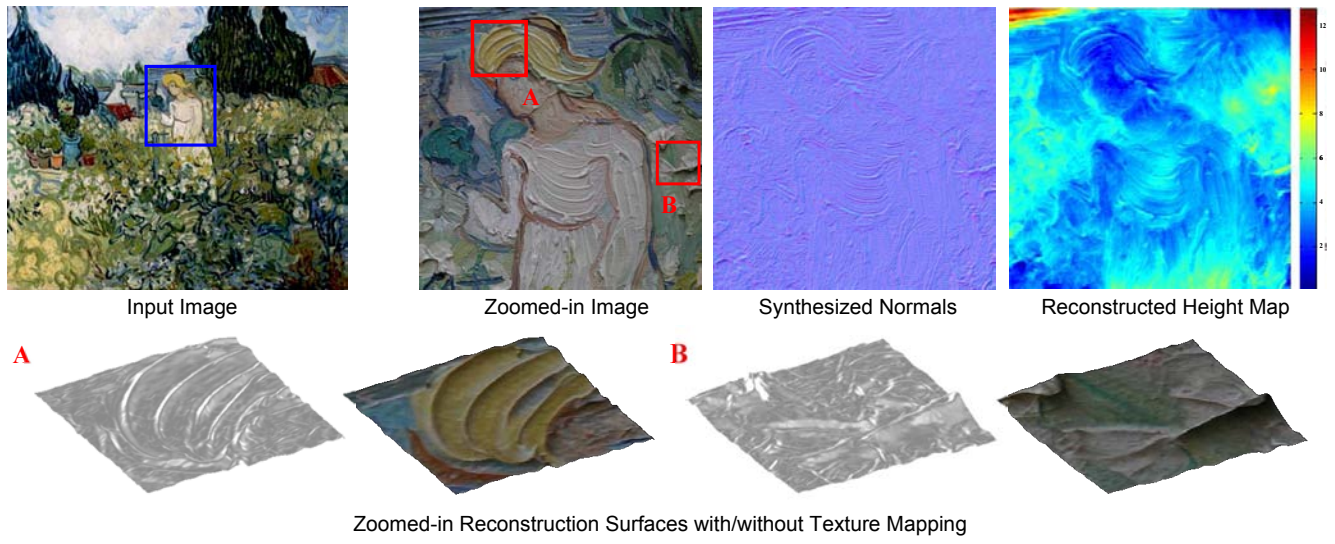

Figure 6. Example: Van Gogh's painting, *Marguerite Gachet in the Garden*, 1890. Top left: input image. Top right: zoomed-in region of the input image, our synthesized normals, and reconstructed surface height map with color coding. Bottom: two zoomed-in regions of reconstructed surface with/without texture mapping under different lighting directions.

viewing angles and illumination. For the large amount of paintings that are hard or even impossible to get physical access (e.g., those destroyed or kept in private), such 3D effect would be impossible to achieve without our technique.

5. Conclusion and Discussion

We have presented a technique to synthesize the 2.5D relief of an oil painting from a single photograph. We use a learning based approach to infer the normal field using a conditional random field based from the original 2D input image. Our approach relies on a training set of surface normals captured from real oil paintings that can be tuned to

match the input image's lighting condition. Our approach is able to produce visually plausible synthesis of oil paintings.

As with existing synthesis techniques, the quality of our results is related to the similarity of the training data, thus it is important to provide a large database of styles to choose from for the use in a real world setting. In addition, though we try to minimize the inherent shading in texture maps using a recovered reflectance image, the reflectance is not perfectly decoupled. This is currently a challenging problem that is slated for future work.

Input Image   Synthesized Normals

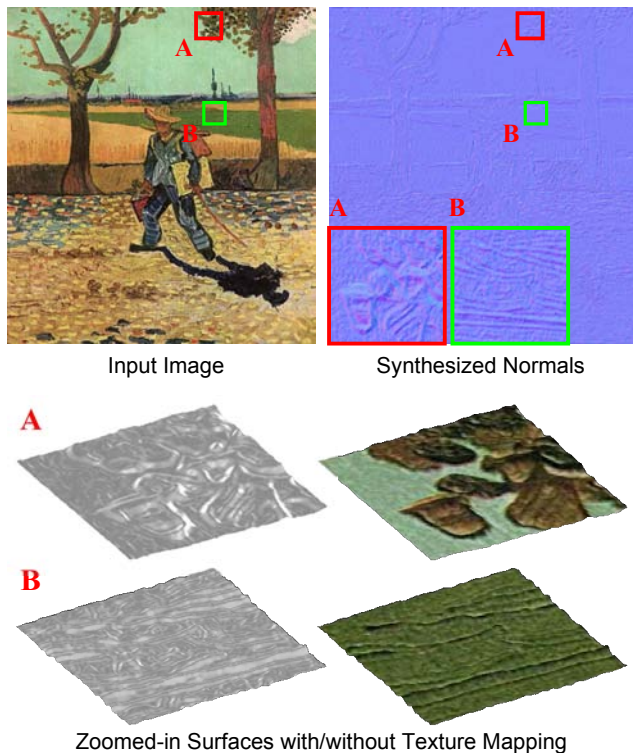Zoomed-in Surfaces with/without Texture Mapping

Figure 7. Example: destroyed artwork: Van Gogh's *A Painter on His Way to Work*, 1888. Top: input image (left) and our synthesized normals with two zoomed-in regions (right). Bottom: two zoomed-in regions of reconstructed surface with/without texture mapping under different lighting directions.

6. Acknowledgement

References

[1] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *ECCV*, 2006.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *SIGGRAPH*, 2009.

[3] H. Barrow, J. Tenenbaum, S. I. A. I. C. C. Science, and T. Division. *Recovering intrinsic scene characteristics from images*. Artificial Intelligence Center, SRI International, 1978.

[4] M. Ben-Ezra. A digital gigapixel large-format tile-scan camera. *IEEE Computer Graphics and Applications*, 2011.

[5] J. F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH*, 1978.

[6] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. In *SIGGRAPH Asia*, 2009.

[7] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.

[8] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. In *IJCV*, 2000.

[9] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Journal of Computer and System Sciences*, 1997.

[10] Y. Gingold, T. Igarashi, and D. Zorin. Structured annotations for 2d-to-3d modeling. In *SIGGRAPH Asia*, 2009.

[11] Y. Gingold and D. Zorin. Shading-based surface editing. In *SIGGRAPH*, 2008.

[12] R. Grosse, M. Johnson, E. Adelson, and W. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

[13] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH*, 1998.

[14] B. Horn. Obtaining shape from shading information. In *Shape from shading*, 1989.

[15] I. Kemelmacher-Shlizerman, R. Basri, and B. Nadler. 3d shape reconstruction of mooney faces. In *CVPR*, 2008.

[16] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *SIGGRAPH*, 2003.

[17] J. Lien and N. Amato. Approximate convex decomposition of polyhedra. In *SPM*, 2007.

[18] J. Lu, P. V. Sander, and A. Finkelstein. Interactive painterly stylization of images, videos and 3d animations. In *I3D*, 2010.

[19] N. L. Max and B. G. Becker. Bump shading for volume textures. *IEEE Computer Graphics and Applications*, 1994.

[20] G. Ramanarayanan and M. K. Bala. Constrained texture synthesis via energy minimization. *IEEE Trans. on Visualization and Computer Graphics*, 2007.

[21] X. Ren and J. Malik. Learning a classification model for segmentation. In *CVPR*, 2003.

[22] X. Tang and X. Wang. Face sketch synthesis and recognition. In *ICCV*, 2003.

[23] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *PAMI*, 2005.

[24] Q. Wang, X. Tang, and H. Shum. Patch based blind image super resolution. In *ICCV*, 2005.

[25] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *PAMI*, 2009.

[26] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 1980.

[27] T.-P. Wu, J. Sun, C.-K. Tang, and H.-Y. Shum. Interactive normal reconstruction from a single image. In *SIGGRAPH*, 2008.

[28] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum. Shapepalettes: interactive normal transfer via sketching. In *SIGGRAPH*, 2007.

[29] J. Yedidia, W. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. on Information Theory*, 2005.