# Depth From Water Reflection

Linjie Yang, Jianzhuang Liu, *Senior Member, IEEE*, and Xiaoou Tang, *Fellow, IEEE*

***Abstract*— The scene in a water reflection image often exhibits bilateral symmetry. In this paper, we design a framework to reconstruct the depth from a single water reflection image. This problem can be regarded as a special case of two-view stereo vision. It is challenging to obtain correspondences from the real scene and the mirror scene due to their large appearance difference. We first propose an appearance adaptation method to transform the appearance of the mirror scene so that it is much closer to the real scene. We then present a stereo matching algorithm to obtain the disparity map of the real scene. Compared with other depth-from-symmetry work that deals with man-made objects, our algorithm can recover the depth maps of a variety of scenes, where both natural and man-made objects may exist.**

***Index Terms*— Two-view stereo, water reflection, appearance adaptation, depth-from-symmetry, dense stereo.**

## I. Introduction

WATER reflection is a common natural phenomenon and such scenes are popular among photographers. When the landscape is reflected by the water, the mirror scene and the real scene make up a symmetric scene. This symmetric scene can be used to produce a 3D scene with the property of symmetry. The principle idea is that the real scene and the mirror scene can be regarded as one seen from two viewpoints if the observer is not in the symmetry plane, meaning that the symmetric scene is equal to a pair of stereo images.

Much work has been done about shape from symmetry [1]–[5], which focuses on man-made symmetric objects. However, to the best of our knowledge, no published study has attempted to recover the depth from this special symmetry of water reflection where both natural and man-made objects may present. First, it is challenging to obtain correspondences between the real scene and the mirror scene in the water because of their appearance difference.

Manuscript received June 29, 2014; revised November 23, 2014; accepted January 6, 2015. Date of publication January 28, 2015; date of current version February 17, 2015. This work was supported in part by the National Basic Research Program of China under Grant 2015CB352501 and in part by the Guangdong Innovative Research Team Program under Grant 201001D0104648280. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrea Cavallaro.

L. Yang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: yl012@ie.cuhk.edu.hk).

J. Liu is with the Media Laboratory, Huawei Technologies Company Ltd., Shenzhen 518129, China, and also with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: liu.jianzhuang@huawei.com).

X. Tang is with the Shenzhen Key Laboratory of CVPR, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: xtang@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2397591

Fig. 1. Example images with water reflection.

The two scenes have different illuminations and the mirror scene is usually distorted by the fluctuation of the water (see Fig. 1). Second, in a stereo system, if the object is too far away from the camera, its disparity will be close to 0 and the system cannot differentiate its depth from an object at infinity. Water reflection images often contain objects that are far away from the camera and the reconstruction of the scene depth from such images is not trivial.

In this paper, we propose a novel approach to reconstructing the depth layout from images with water reflection. The pipeline of our algorithm is shown in Fig. 2. We first propose an appearance adaptation scheme to reduce the appearance difference between the real scene and the mirror scene, which is essential for depth reconstruction. We then design a stereo matching algorithm to recover the depth of the scene with man-made and/or natural objects.

## II. Related Work

The related work mainly includes two topics: symmetry detection and 3D reconstruction from symmetry. Symmetry detection is a fundamental and long-lasting topic in computer vision. Feature-based detection algorithms have been developed to detect planar bilateral symmetric objects [6]–[8]. These approaches first find scale and affine-invariant features and then vote to obtain the axis of symmetry. Liu et al. [9] provide a comprehensive survey of the literature.

Early approaches to 3D reconstruction from symmetry often require user interaction or only reconstruct specific objects. Gordon et al. [10] deal with shape from symmetry and reconstruct objects marked with a regular grid.

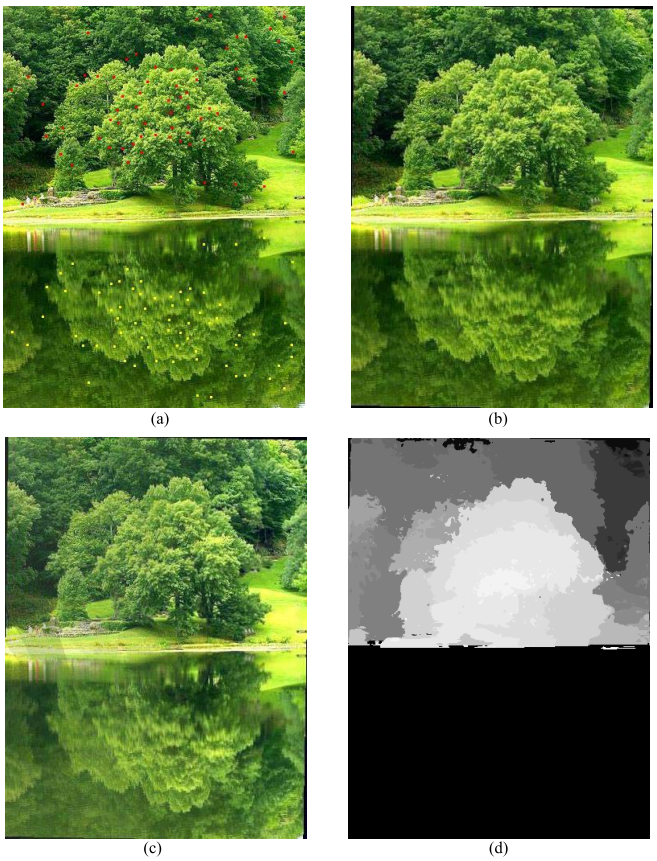



Fig. 2. The pipeline of our algorithm. (a) An original image with detected symmetric keypoints. (b) The rectified image. (c) The transformed image with the appearance adaptation. (d) The recovered depth map.

Later, Mitsumoto et al. [11] build a symmetric scene using a mirror and reconstruct a 3D model with manually labeled correspondences. Franccois et al. [3] also require labeled point correspondences to reconstruct a symmetric object. Jiang et al. [5] calibrate the camera using a pyramid frustum, then recover a set of 3D points with the underlying symmetry, and finally adopt user interaction to reconstruct a complete 3D structure. Xue et al. [4] detect symmetric line pairs from a symmetric piecewise planar object and then recover a depth map through Markov random fields.

Automatic methods have been proposed recently for depth reconstruction from symmetric objects. Wu et al. [12] detect the repetitions on a building and then utilize a dense stereo method to reconstruct its 3D relief model. Koser et al. [1] match keypoints to find the symmetry plane, then reconstruct a depth map from a symmetric object using a global stereo approach. Sinha et al. [2] reconstruct 3D curved models from textureless objects in images. All these approaches focus on man-made symmetric objects (such as buildings, cars, and chairs) with the same appearance on both sides of the symmetry plane.

## III. Image Rectification

A symmetric scene has a plane of symmetry, which is called the *symmetry plane*. In this paper, the symmetry plane is the water surface. A point and its mirror point lie on different

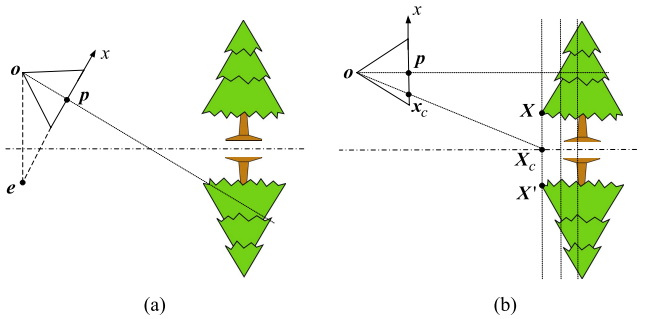



Fig. 3. The camera directions before and after the rectifying rotation, where o is the camera center and p is the principal point. (a) Before the rectifying rotation. (b) After the rectifying rotation.

Fig. 4. Symmetric keypoint detection results in two images with heavy scene distortion in the water. The keypoints in the real scene are marked in red and those in the mirror scene are marked in yellow. Better viewed on the screen with enlarged images.

sides of this plane and they are symmetric with respect to this plane. A point and its mirror point form a *symmetric pair*, and their projections on the image are called a *symmetric image pair*. When the normal of the symmetry plane is aligned with the $x$ axis of the image plane of the camera (see Fig. 3(b)), all the symmetric image pairs (say, the images of $\boldsymbol{X}$ and $\boldsymbol{X}'$ in Fig. 3(b)) lie on the corresponding scanlines, and the depth of the symmetric pair $(\boldsymbol{X}, \boldsymbol{X}')$ is determined up to a scale by the image $\boldsymbol{x}_c = (x_c, y_c)$ of their midpoint $\boldsymbol{X}_c$ if the camera parameters are known. A symmetric pair with larger $x_c$ is farther from the camera center.

When the camera is facing an arbitrary direction in which the image plane is not perpendicular to the symmetry plane, there is a vanishing point $\boldsymbol{e}$ for the lines joining the symmetric image pairs (see Fig. 3(a)). In this case, the image needs to be rectified to satisfy the condition for a stereo method. Image rectification is a well-studied problem in stereo vision [2], [13], [14]. We utilize the method in [2] to rectify the image. It applies a rectifying rotation to the image to transform the $x$ axis of the image plane to be aligned with the normal of the symmetry plane.

After applying the rotation matrix $\boldsymbol{R}$ to the camera model, the rectified image point $\boldsymbol{x}'$ becomes

$$\boldsymbol{x}' = \boldsymbol{K}\boldsymbol{R}^{\top}\boldsymbol{K}^{-1}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{x}$ is the corresponding original image point and $\boldsymbol{K}$ is the camera calibration matrix [14]. The rotation matrix $\boldsymbol{R}$ is determined by the vanishing point $\boldsymbol{e}$ [2], which is detected with the following scheme.

The vanishing point $\boldsymbol{e}$ is the intersection of the lines joining symmetric image pairs. To robustly find this point,

Fig. 5. First row: real scenes. Second row: the mirror scenes flipped. Third row: the transformed mirror scenes flipped. The real scene and the mirror scene in one column are cropped from the same image.

a set of symmetric image pairs is needed. The methods in [1], [7], [8], and [12] detect scale and affine-invariant features and then perform nearest neighbor matching. Loy and Eklundh [6] utilize the SIFT descriptor, which is rotationally invariant but not affine-invariant, to detect bilateral and rotational symmetry. In our case, where the scene includes water reflection, the perspective effect is usually weak and affine-invariant features are not necessary (see Fig. 1). We use the method in [6] to detect bilateral symmetric keypoints. These keypoints are split into two groups, those in the real scene and those in the mirror scene. They can be detected with high accuracy even in distorted scenes like those in Fig. 4. Typically, we can find 30 to 100 pairs of symmetric keypoints from an image of resolution $400 \times 500$. We then use the method in [15] to robustly calculate the position of $\boldsymbol{e}$ from the lines joining the symmetric keypoints.

## IV. Depth Computation

### A. Appearance Adaptation of Mirror Scenes

The appearance of the mirror scene in the water often appears rather different from the appearance of the real scene, which can be seen from Fig. 5. The appearance difference is caused by several aspects: brightness decrease due to the absorption of light by the water, water fluctuations and ripples, and some small objects (e.g., tree leaves) floating on the water. In the experimental section, we will show that this appearance difference problem can cause serious depth errors. To overcome this problem, we propose an effective method to adapt the appearance of the mirror scene to the real scene after image rectification. After the adaptation, the appearance of the mirror scene is much closer to the real scene, making stereo matching more robust.

Let $(k, k')$ be a symmetric image pair, where $k$ is a point in the real scene and $k'$ is the mirror point. Also let $\omega_k$ and $\omega_{k'}$ be two local windows of the same size centered at $k$ and $k'$, respectively. For appearance adaptation, we regard each pair of corresponding pixels $(i, j)$ in $\omega_k$ and $\omega_{k'}$ as a symmetric image pair. Each such pixel pair is denoted as $(i, j) \in \omega_k \leftrightarrow \omega_{k'}$ in this section. Let the rectified image and the transformed image be $\boldsymbol{I}$ and $\boldsymbol{I}^t$, respectively. Assume that $\boldsymbol{I}^t$ is a linear transform of $\boldsymbol{I}$ in $\omega_{k'}$:

$$\boldsymbol{I}_i^t = a_k \boldsymbol{I}_i + \boldsymbol{b}_k \ \forall i \in \omega_{k'}, \tag{2}$$

where $a_k$ and $\boldsymbol{b}_k$ are the parameters we need to find for the final appearance transform, and $\boldsymbol{I}_i = (r_i, g_i, b_i)^\top$ with $r_i$, $g_i$, and $b_i$ being the RGB color values at $i$. We want to minimize the appearance difference between $\omega_k$ in the rectified image $\boldsymbol{I}$ and $\omega_{k'}$ in the transformed image $\boldsymbol{I}^t$. Specifically, we minimize the following cost function

$$C(a_k, \boldsymbol{b}_k) = \sum_{(i,j)\in\omega_k \leftrightarrow \omega_{k'}} (||a_k \boldsymbol{I}_j + \boldsymbol{b}_k - \boldsymbol{I}_i||_2^2 + \epsilon a_k^2), \tag{3}$$

where $\epsilon$ is a regularization factor. The minimization of (3) can be obtained by linear regression with

$$a_k = \frac{\frac{1}{|\omega_k|}\sum_{(i,j)\in\omega_k \leftrightarrow \omega_{k'}} \boldsymbol{I}_j^\top \boldsymbol{I}_i - \boldsymbol{\mu}_{k'}^\top \boldsymbol{\mu}_k}{Tr(\boldsymbol{\Sigma}_{k'}) + \epsilon}, \tag{4}$$

$$\boldsymbol{b}_k = \boldsymbol{\mu}_k - a_k \boldsymbol{\mu}_{k'}, \tag{5}$$

where $\boldsymbol{\mu}_{k'}$ and $\boldsymbol{\Sigma}_{k'}$ are the mean vector and covariance matrix of $\boldsymbol{I}$ in $\omega_{k'}$, respectively, $\boldsymbol{\mu}_k$ is the mean vector of $\boldsymbol{I}$ in $\omega_k$, $|\omega_k|$ is the number of pixels in $\omega_k$, and $Tr(\boldsymbol{A})$ is the trace of the matrix $\boldsymbol{A}$.

We use the detected symmetric keypoints described in Section III as the symmetric image pairs and obtain a sparse grid of $(a_k, \boldsymbol{b}_k)$ on the image. Finally, for an arbitrary pixel in $\boldsymbol{I}^t$, we compute its value by

$$\boldsymbol{I}_i^t = \frac{1}{N} \sum_{k\in\mathcal{N}(i,N)} (a_k \boldsymbol{I}_i + \boldsymbol{b}_k), \tag{6}$$

where $\mathcal{N}(i, N)$ is the set of $N$ keypoints nearest to pixel $i$.

Fig. 5 shows several examples of the appearance adaptation, from which we can see that the transformed mirror scenes are more close to their corresponding real scenes. Note that there are some slight streak-like artifacts caused by the nearest neighbor averaging in the transformed images, which does no harm to our following stereo algorithm.

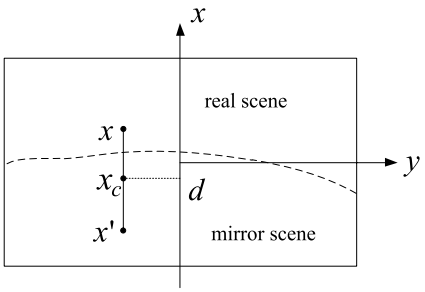



Fig. 6. The illustration of the coordinate system on the rectified image. The dashed line is the boundary between the real scene and the mirror scene. The origin of the coordinate system is at the center of the image. $(\boldsymbol{x}, \boldsymbol{x}')$ is a symmetric image pair and $\boldsymbol{x}_c$ is the midpoint. The $x$-coordinate of $\boldsymbol{x}_c$ is the disparity $d$.

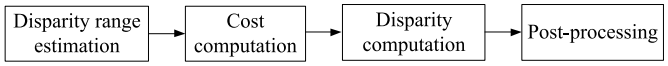



Fig. 7. The framework of our stereo algorithm.

### B. Dense Stereo From Water Reflection Scenes

Traditional dense stereo methods find the dense correspondences between two rectified images and result in a disparity map of the same size as the original image [16]. For a water reflection scene, the real scene and the mirror scene can be regarded as a pair of stereo images. However, there is no clear separation between the real scene and the mirror scene for stereo matching to start with. Fig. 6 shows the coordinate system on the rectified image and the dashed line illustrates the boundary between the real scene and the mirror scene. Although the positions of the detected sparse keypoints impose a constraint on the boundary (the boundary on the scanline $\overline{\boldsymbol{x}\boldsymbol{x}'}$ must be between $\boldsymbol{x}$ and $\boldsymbol{x}'$), it is not sufficient to predict the complete position of the boundary. We take this boundary into consideration in our stereo algorithm and output a disparity map for the real scene along with the boundary between the real scene and the mirror scene.

The framework of our stereo algorithm is shown in Fig. 7. The input to the algorithm is the rectified image $\boldsymbol{I}$ and the transformed image $\boldsymbol{I}^t$ from appearance adaptation (see Section IV-A). This stereo algorithm utilizes a 3D array $C_{\boldsymbol{x},d}$ (for the 2D image coordinates $\boldsymbol{x}$ and the 1D disparity $d$) called the cost volume to store the costs for choosing a disparity $d$ at pixel $\boldsymbol{x}$. The 2D array $C_{\boldsymbol{x},d_s}$ with a fixed disparity $d_s$ is referred to as a slice of the cost volume.

First, we need to estimate the range of disparities in the image. For a symmetric image pair $(\boldsymbol{x}, \boldsymbol{x}')$ and its midpoint $\boldsymbol{x}_c = (x_c, y_c)$, since $x_c$ determines the depth of $(\boldsymbol{x}, \boldsymbol{x}')$, we refer to $x_c$ as the disparity $d$ of this pair (also see the illustration of $d$ in Fig. 6). Let $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$, $i = 1, 2, ..., n$, be the detected symmetric keypoints with their corresponding midpoints $\boldsymbol{x}_{ci} = \frac{\boldsymbol{x}_i + \boldsymbol{x}'_i}{2} = (x_{ci}, y_{ci})$, $i = 1, 2, ..., n$. We can obtain the range of disparities

$$\mathcal{D} = [\min_{i=1,2,...,n} \{x_{ci}\}, \max_{i=1,2,...,n} \{x_{ci}\}]. \tag{7}$$

Since symmetric keypoints are hard to detect on the faraway background objects including the sky and clouds, $\mathcal{D}$ may miss the disparities for the faraway objects. Therefore, we add a small range of disparity to $\mathcal{D}$ in order to compensate for the missing. The updated range of disparities is

$$\mathcal{D} = [\min_{i=1,2,...,n} \{x_{ci}\}, \max_{i=1,2,...,n} \{x_{ci}\} + \delta_d], \tag{8}$$

where $\delta_d$ is set to 3 pixels in our experiments.

Then, the costs in the cost volume $C_{\boldsymbol{x},d}$ are calculated. We calculate the matching cost of each pair of pixels on a scanline. We employ a commonly used cost function that has shown robust capacity for stereo and optical flow [17]–[19]. At disparity $d$, between a pixel $\boldsymbol{x} = (x, y)$ assumed in the real scene in the rectified image and the corresponding pixel $\boldsymbol{x}' = (x', y')$ assumed in the mirror scene in the transformed image (see Fig. 2), the matching cost is

$$\begin{aligned} C_{\boldsymbol{x},d} = {} & (1-\alpha)\min\{||I^t_{\boldsymbol{x}'} - I_{\boldsymbol{x}}||, \tau_1\} \\ & + \alpha \min\{||\nabla_x I^t_{\boldsymbol{x}'} - \nabla_x I_{\boldsymbol{x}}||, \tau_2\}, \end{aligned} \tag{9}$$

$$\text{s.t.}: \ d \in \mathcal{D}, \ y = y', \ x + x' = 2d, \ x > x', \tag{10}$$

where $\nabla_x$ is the gradient in the $x$ direction, $\alpha$ is a balancing factor between the color and gradient terms, and $\tau_1$ and $\tau_2$ are two truncation thresholds. Motivated by [19], we utilize the guided filter [20] to filter each slice of the cost volume in order to aggregate the cost over a support window with edge-preserving properties, which is to reduce noise and to make the stereo matching more robust. The filtered cost volume is denoted as $C'_{\boldsymbol{x},d}$.

The third step is disparity computation. We calculate the disparity of a pixel in a winner-take-all manner with a range constraint. Suppose the range of disparities is $\mathcal{D} = [d_1, d_2]$ obtained from (8). For a pixel $\boldsymbol{x} = (x, y)$ assumed in the real scene, the disparity is calculated by

$$d_{\boldsymbol{x}} = \operatorname*{argmin}_{i=d_1, d_1+1, ..., d'_2} C'_{\boldsymbol{x},i}, \tag{11}$$

where $d'_2 = \min\{x, d_2\}$. $d_{\boldsymbol{x}}$ cannot be larger than $x$ because a pixel in the real scene is always above the midpoint of the corresponding symmetric image pair (see Fig. 6). Note that for a pixel with $x < d_1$, there is no valid disparity for it. Such a pixel is in the mirror scene. Similarly, for a pixel $\boldsymbol{x} = (x, y)$ assumed in the mirror scene, the disparity is

$$d'_{\boldsymbol{x}} = \operatorname*{argmin}_{i=d'_1, d'_1+1, ..., d_2} C'_{\boldsymbol{x},i}, \tag{12}$$

where $d'_1 = \max\{x, d_1\}$. There is no valid disparity for a pixel with $x > d_2$ either. Such a pixel lies in the real scene. From (11) and (12), we obtain two disparity maps $D$ and $D'$, as shown in Figs. 8(a) and (b).

The last step is post-processing. The disparity map $D$ is refined through several ways. First, we check the disparity consistency of pixels in $D$ and $D'$. Let $\boldsymbol{x} = (x, y)$ and $\boldsymbol{x}' = (2d - x, y)$ be a pair of corresponding pixels, where $D(\boldsymbol{x}) = d$. If $D'(\boldsymbol{x}') \neq d$, then $\boldsymbol{x}$ and $\boldsymbol{x}'$ are considered inconsistent. For each inconsistent pixel, its disparity is set to be the same as that of the spatially closest consistent pixel on the same scanline. Since this scheme may cause streak-like

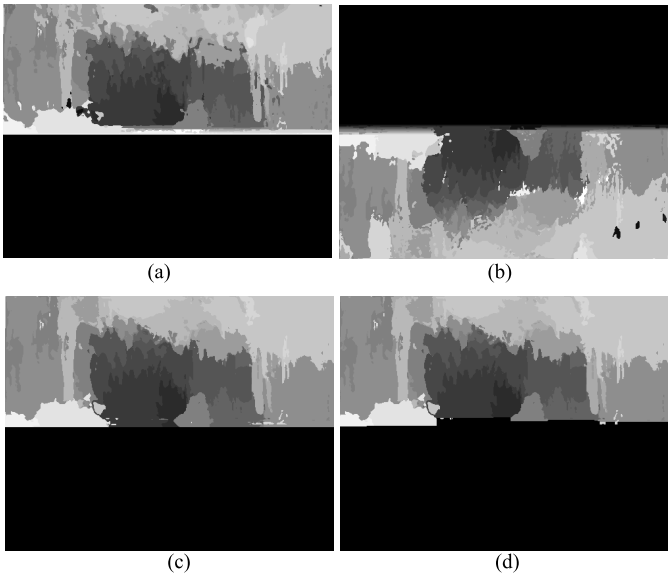



Fig. 8. (a) (b) Disparity maps $D$ and $D'$ obtained from (11) and (12), respectively, based on the fourth image in Fig. 1. (c) The disparity map $D$ after consistency checking and weighted median filtering. (d) The final disparity map. Note that the boundary between the real scene and the mirror scene is also found in (d).

artifacts, we apply a weighted median filter to the inconsistent pixels in order to remove these artifacts and preserve the object boundaries. The weights of the filter are the same as those in the bilateral filter in [19]. Finally, we check the validity of the disparity in $D$ using the constraint $x \geq d_{\boldsymbol{x}}$ for a pixel $\boldsymbol{x} = (x, y)$, and set all the invalid pixels to be in the mirror scene (see Figs. 8(c) and (d)). After all these steps, the final disparity map $\tilde{D}$ is then the result of our stereo algorithm. Note that $\tilde{D}$ naturally contains the boundary between the real scene and the mirror scene, since only the pixels with valid disparities belong to the real scene and the others belong to the mirror scene.

## V. Experimental Results

This section starts by analyzing the feasibility of depth reconstruction from water reflection, and we then evaluate our algorithm on water reflection images, as well as evaluate the effectiveness of the appearance adaptation. Successful and failed examples are presented to explore the limit of this algorithm.

Typically, a water reflection image is taken by a person standing by the water (river, lake, etc). In this situation, the distance from the camera to the water surface is about a person's height $h$. Let us consider a common camera on an iPhone 5. The pixel density $\rho$ of the sensor is about 1.9 $\mu$m/pixel and the focal length $f$ is 4.1 mm. Let $l$ be the depth of a symmetric pair. Observing the geometry in Fig. 9, we can obtain the disparity $d$ of the symmetric pair as

$$d = \frac{fh}{l\rho}. \tag{13}$$

The partial derivative of $d$ with regard to $l$ is

$$\frac{\partial d}{\partial l} = -\frac{fh}{l^2\rho}. \tag{14}$$

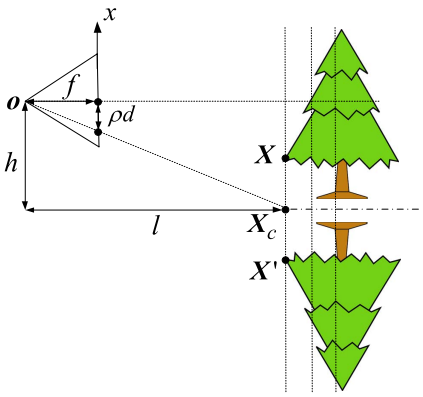



Fig. 9. Illustration of the geometry of the focal length $f$, pixel density $\rho$, disparity $d$, person's height $h$, and depth $l$ of the symmetric pair.

When the disparity decreases by one pixel, the depth of the symmetric pair increases $\frac{l^2\rho}{fh}$. Thus, objects closer to the camera are more discriminative in the disparity map. Suppose that $h$ is 1.8 meters. At the distance of 100 meters, one pixel difference in disparity corresponds to approximately 2.6 meters in depth; at the distance of 1000 meters, one pixel difference in disparity corresponds to 260 meters in depth. Therefore, we can roughly conclude that objects within 100 meters can be differentiated by their disparities. In water reflection images, some faraway objects like mountains and clouds cannot be differentiated by their disparities (see Fig. 13(b)).

We then evaluate our algorithm on water reflection images. We downloaded 81 such images from the internet, Some of which are shown in Fig. 1 and Fig. 5. The disparity ranges for these images are 7 to 50 pixels. Our algorithm is implemented in Matlab and takes approximately 40 seconds to obtain the disparity map for an image of size $400 \times 500$. The parameters for the appearance adaptation $s$, $\epsilon$, $N$ are set to 11 pixels, 0.0001, and 10, respectively, where $s$ is the length of the sides of the square window $\omega_k$. The parameters $\alpha$, $\tau_1$, $\tau_2$ in (9) are set to 0.9, $\frac{10}{255}$, and $\frac{6}{255}$, respectively.

Fig. 10 displays 10 samples of the rectified images and the corresponding disparity maps generated by our algorithm. Our algorithm can handle various scenes including mountainous landscapes, buildings, plants, and caves. Note that the objects closer to the camera contain more details in the disparity map. Some parts of the images (especially in the sky) have incorrect depth estimation, which is further discussed later.

In the experiments, we find that the appearance adaptation scheme is essential to the success of the algorithm. In Fig. 11, we compare some disparity estimation results with and without the appearance adaptation. We can see that the results with the appearance adaptation are much better. Without the appearance adaptation, the depth maps often contain incorrect disparities, especially in the parts where the real scene and the mirror scene have large appearance differences. We also compare our scheme with the normalized cross correlation (NCC), which is a widely-used matching cost for images with contrast variations [21], [22]. The results with NCC are obtained without the appearance adaptation. Fig. 11(d) illustrates that NCC tends to smooth the depth maps, reducing the details,

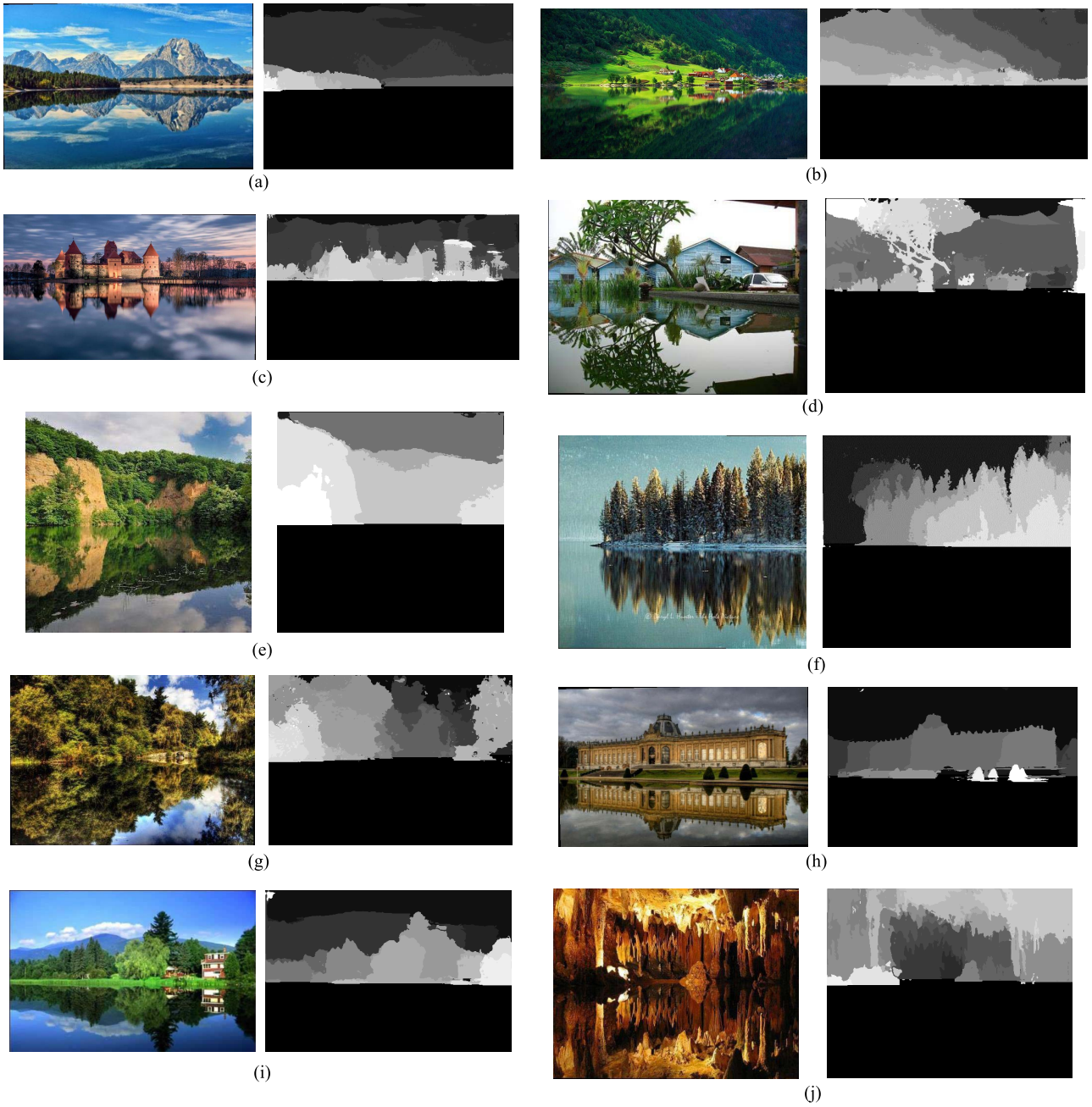



Fig. 10. Rectified image and their disparity maps generated by our algorithm.

such as the profile of the building in the fourth image, which leads to poor disparity estimations.

We also conduct a quantitative experiment with 40 manually labeled images to compare the performance with and without the appearance adaptation, using the NCC cost. The images are first segmented into superpixels using multiscale normalized-cut [23]. Then the superpixels in the real scene are manually adjusted to match the corresponding patches in the mirror scene, from which proper disparities are computed. We only label the superpixels with high confidence in the matching. On average, 57.2% of the pixels in the real scene are labeled for each of the 40 images (most of the unlabeled pixels belong to the textureless sky). In Table I, performances of two criteria for the three matching costs are displayed. The first criterion is the pixel-level accuracy (PLA) that the predicted disparity is regarded as correct only when it equals to the labeled disparity, while the second criterion (PLA ($\pm 1$)) is the accuracy that the predicted disparity is regarded as correct when it is within $[-1, +1]$ pixel range of the labeled disparity. Since slight inaccuracy exists in human labeling, PLA ($\pm 1$) is a more convincing criterion in this scenario. The algorithm with the appearance adaptation outperforms the one without the appearance adaptation and the one with the NCC cost, which shows the advantage of the appearance adaptation for the reconstruction from water reflection.

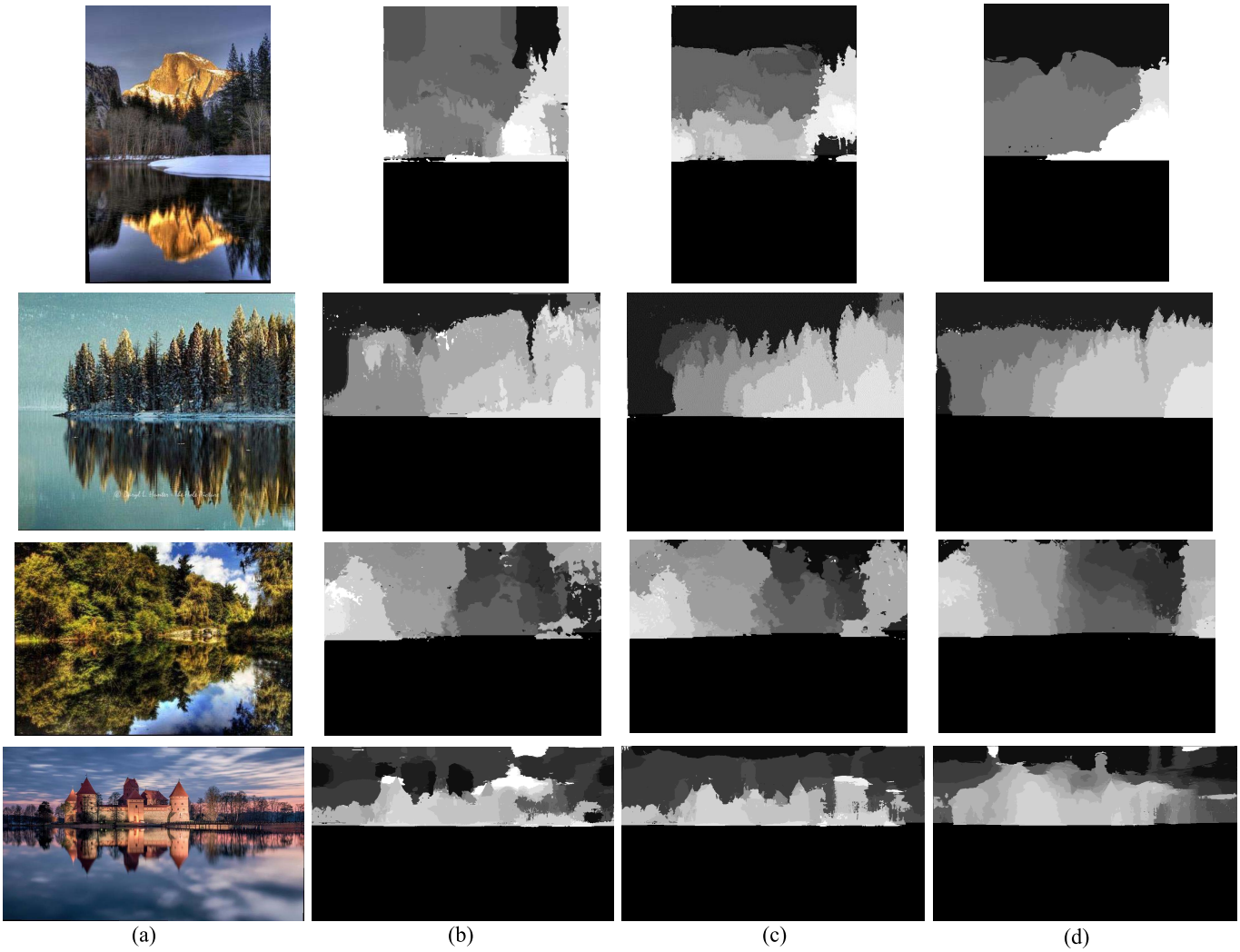



Fig. 11. Depth estimation results without and with the appearance adaptation. (a) Rectified images. (b) Depth maps without the appearance adaptation. (c) Depth maps with the appearance adaptation. (d) Depth maps with NCC cost without the appearance adaptation.

TABLE I
PERFORMANCES WITH TWO CRITERIA FOR THE THREE MATCHING COSTS

| | PLA | PLA ($\pm 1$) |
|---|---|---|
| With appearance adaptation | 0.370 | 0.757 |
| Without appearance adaptation | 0.351 | 0.707 |
| NCC | 0.304 | 0.660 |

We further calculate the average distance $\bar{d}$ between the colors of the patches around the keypoints in the real scene and those in the mirror scene, and the average distance $\bar{d}^t$ between the colors of the patches around the keypoints in the real scene and those in the transformed mirror scene, by

$$\bar{d} = \frac{1}{n|\omega|}\sum_{i=1}^{n}\sum_{(p,q)\in\omega_{k_i}\leftrightarrow\omega_{k_i'}} ||\boldsymbol{I}_p - \boldsymbol{I}_q||_2, \tag{15}$$

$$\bar{d}^t = \frac{1}{n|\omega|}\sum_{i=1}^{n}\sum_{(p,q)\in\omega_{k_i}\leftrightarrow\omega_{k_i'}} ||\boldsymbol{I}_p - \boldsymbol{I}_q^t||_2, \tag{16}$$

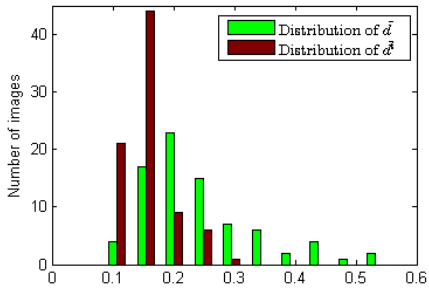



Fig. 12. The distributions of the average distances $\bar{d}$ and $\bar{d}^t$.

where $(k_i, k_i')$, $i = 1, 2, \ldots, n$, are $n$ pairs of symmetric keypoints, $\omega_{k_i}$ and $\omega_{k_i'}$ are the local windows around $k_i$ and $k_i'$, respectively, and $|\omega|$ is the size of the local window which is 121 ($11\times11$). We calculate the average distances for all the 81 images and show the distribution of $\bar{d}$ and $\bar{d}^t$ in Fig. 12. We can see that $\bar{d}^t$ is much less than $\bar{d}$, which indicates the appearance adaptation can greatly reduce the distance between the real scene and the mirror scene.

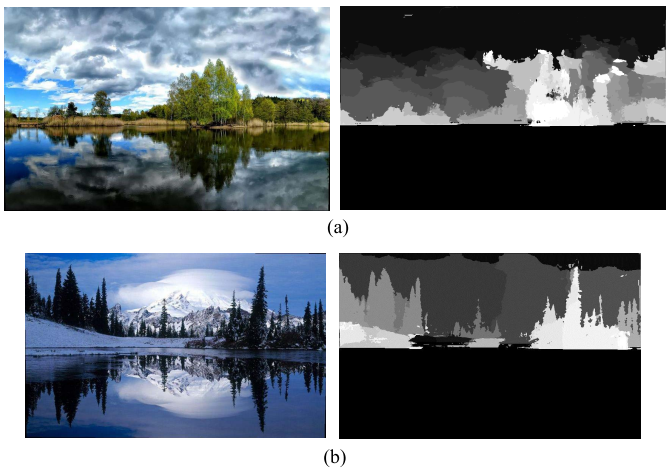


Fig. 13. (a) A failure example caused by heavy blur and distortion in the water. (b) Another failure example in which the mountain is too far away from the camera and its disparity is close to that of the background.

We also have some failure cases in the experiments, two of which are shown in Fig. 13. The failures are usually caused by three problems. The first is heavy scene blur and distortion in the water, as shown in Fig. 13(a). The second is too faraway objects whose disparities are very close to the background objects (clouds and the sky), making it difficult for our algorithm to differentiate their depths, as shown in Fig. 13(b). The third problem is that large textureless regions such as the sky may cause incorrect disparities (see some parts in the sky in Figs. 10(a), (c), (d), and (i)), which is a common problem in stereo vision algorithms. The problem caused by the textureless sky can be solved by detecting and removing the sky from the image using a state-of-the-art image parsing method such as [24] before performing depth reconstruction.

## VI. Conclusions

In this paper, we have proposed a framework for a novel topic: depth from water reflection. Unlike other depth-from-symmetry methods that deal with man-made objects, our approach reconstructs the depth of a scene where both natural and man-made objects may exist. Our algorithm first rectifies the image with detected symmetric keypoints to satisfy the scanline condition for stereo matching. It then applies appearance adaptation to the rectified image to reduce the appearance difference between the real scene and the mirror scene. Finally, dense stereo algorithm especially designed for this kind of symmetric scene is carried out to obtain the scene depth. Future work includes the combination of image parsing and depth from water reflection to improve scene understanding and depth reconstruction.

## Acknowledgement

The authors would like to thank Kaiming He for his valuable discussion of the manuscript, and the editor and anonymous reviewers for their constructive comments.

## References

[1] K. Köser, C. Zach, and M. Pollefeys, "Dense 3D reconstruction of symmetric scenes from a single image," in *Proc. 33rd DAGM Symp.*, 2011, pp. 266–275.

[2] S. N. Sinha, K. Ramnath, and R. Szeliski, "Detecting and reconstructing 3D mirror symmetric objects," in *Proc. 12th ECCV*, 2012, pp. 586–600.

[3] A. R. J. François, G. G. Medioni, and R. Waupotitsch, "Mirror symmetry ⇒ 2-view stereo geometry," *Image Vis. Comput.*, vol. 21, no. 2, pp. 137–143, 2003.

[4] T. Xue, J. Liu, and X. Tang, "Symmetric piecewise planar object reconstruction from a single image," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 2577–2584.

[5] N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," *ACM Trans. Graph.*, vol. 28, no. 5, Dec. 2009, Art. ID 113.

[6] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *Proc. 9th ECCV*, 2006, pp. 508–521.

[7] H. Cornelius and G. Loy, "Detecting bilateral symmetry in perspective," in *Proc. CVPR Workshop*, 2006, p. 191.

[8] H. Cornelius, M. Perd'och, J. Matas, and G. Loy, "Efficient symmetry detection using local affine frames," in *Image Analysis*. Berlin, Germany: Springer-Verlag, 2007.

[9] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. Van Gool, "Computational symmetry in computer vision and computer graphics," *Found. Trends Comput. Graph. Vis.*, vol. 5, nos. 1–2, pp. 1–195, 2010.

[10] G. G. Gordon, "Shape from symmetry," *Proc. SPIE*, vol. 1192, p. 297, Mar. 1990.

[11] H. Mitsumoto, S. Tamura, K. Okazaki, N. Kajimi, and Y. Fukui, "3D reconstruction using mirror images based on a plane symmetry recovering method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 941–946, Sep. 1992.

[12] C. Wu, J.-M. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3113–3120.

[13] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 1999, pp. 125–131.

[14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[15] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2136–2143.

[16] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, 2002.

[17] C. Lei and Y.-H. Yang, "Optical flow estimation on coarse-to-fine region-trees using discrete optimization," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 1562–1569.

[18] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[19] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3017–3024.

[20] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. 11th ECCV*, 2010, pp. 1–14.

[21] O. Faugeras *et al.*, "Real time correlation-based stereo: Algorithm, implementations and applications," Inria, Tech. Rep. EPFL-REPORT-176855, 1993.

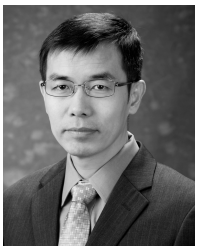
[22] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE 11th ICCV*, Oct. 2007, pp. 1–8.

[23] S. X. Yu, "Segmentation using multiscale cues," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun./Jul. 2004, pp. I-247–I-254.

[24] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 3001–3008.

**Linjie Yang** received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include computer vision and machine learning.

**Jianzhuang Liu** (M'02–SM'02) received the Ph.D. degree in computer vision from The Chinese University of Hong Kong, Hong Kong, in 1997. He was a Research Fellow with Nanyang Technological University, Singapore, from 1998 to 2000. From 2000 to 2012, he was a Post-Doctoral Fellow, an Assistant Professor, and an Adjunct Associate Professor with The Chinese University of Hong Kong. He was with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor, in 2011. He is currently a Chief Scientist with Huawei Technologies Company, Ltd., Shenzhen. He has authored over 100 papers, most of which are in prestigious journals and conferences in computer science. His research interests include computer vision, image processing, machine learning, multimedia, and graphics.



**Xiaoou Tang** (S'93–M'96–SM'02–F'09) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1990, the M.S. degree from the University of Rochester, Rochester, NY, USA, in 1991, and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1996.

He is currently a Professor with the Department of Information Engineering and an Associate Dean (Research) with the Faculty of Engineering, The Chinese University of Hong Kong, Hong Kong. He worked as the Group Manager of the Visual Computing Group with Microsoft Research Asia, Beijing, China, from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing.

Dr. Tang was a recipient of the best paper award at the IEEE Conference on Computer Vision and Pattern Recognition in 2009 and the Outstanding Student Paper Award at the Association for the Advancement of Artificial Intelligence in 2015. He was the Program Chair of the IEEE International Conference on Computer Vision in 2009 and has served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.