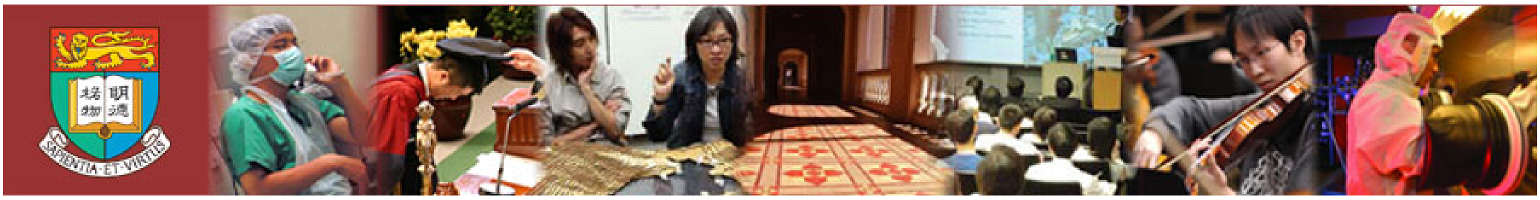


Object-Based Rendering and 3-D Reconstruction Using a Moveable Image-Based System

Zhen-Yu Zhu, Shuai Zhang, Shing-Chow Chan, *Member, IEEE,* and Heung-Yeung Shum, *Fellow, IEEE*

***Abstract*—This paper proposes a movable image-based rendering (M-IBR) system for improving the viewing freedom and environmental modeling capability of conventional static IBR systems. The system supports object-based rendering and 3-D reconstruction capability and consists of three main components.**

1) **An improved video stabilization method to reduce the shaky motion frequently encountered in movable IBR systems. It employs local polynomial regression (LPR) to automatically select an appropriate bandwidth for smoothing the estimated motion.**
2) **A novel view synthesis algorithm using a new segmentation and mutual-information (MI)-based algorithm for dense depth map estimation, which relies on segmentation, LPR-based depth map smoothing, and MI-based matching algorithm to iteratively estimate the depth map. The method is very flexible and both semiautomatic and automatic segmentation methods can be employed. They rank fourth and sixth, respectively, in the Middlebury comparison of existing depth estimation methods. This allows high-quality renderings of outdoor scenes with improved mobility/freedom to be obtained.**
3) **A new 3-D reconstruction algorithm that utilizes the sequential structure-from-motion technique and the dense depth maps estimated previously. It relies on a new iterative point cloud refinement algorithm based on Kalman filter for outlier removal and the segmentation-MI-based algorithm to further refine the correspondences and the projection matrices. The mobility of our system allows us to recover more conveniently 3-D model of static objects from the improved point cloud using a new robust radial basis function-based modeling algorithm to further suppress possible outliers and generate smooth 3-D meshes of objects. Experimental results show that the proposed 3-D reconstruction algorithm significantly reduces the adverse effect of the outliers and produces high-quality renderings using shadow light field and the model reconstructed.**

***Index Terms*—3-D reconstruction, image-based rendering (IBR), movable IBR systems, tracking and stabilization.**

I. Introduction

IMAGE-BASED rendering/representation (IBR) [1]–[9], [12] is a promising technology for rendering new views of scenes from a collection of densely sampled images or videos. It has potential applications in virtual reality, immersive television, and visualization systems.

While there has been considerable progress recently in the capturing, compression and transmission of image-based representations [12], [13], [31], [32], [52], most multiple camera systems are not designed to be movable so that the viewpoints are somewhat limited and usually cannot cope with moving objects and perform 3-D reconstruction of objects in open environment. Apart from many system design issues, there are also many important problems and difficulties in realizing these systems. This motivates us to study in [21] the design and construction of a movable image-based rendering (M-IBR) system based on plenoptic videos. In particular, a linear camera array consisting of eight video cameras was mounted on an electrically controllable wheel chair and its motion can be controlled manually or remotely by means of additional hardware circuitry. The system can potentially provide improved viewing freedom to users and ability to cope with moving objects and perform 3-D reconstruction. Moreover, multiview displays are becoming available [13]. It is predicted that 3-D or multiview applications will become another important means of information exchange.

In this paper, we study the object-based rendering and 3-D reconstruction using a movable IBR system so as to provide improved viewing freedom and object modeling of stationary objects in an open environment. In particular, we developed a software system for such purposes, which includes the following.

1) An improved video stabilization method to reduce the shaky motion caused by the vibration of the wheel chair and the roughness of the ground surface. Our method employs local polynomial regression (LPR) to automatically select an appropriate bandwidth or window for smoothing the displacement, which avoids the trial-and-error selection in conventional methods.
2) A view synthesis algorithm based on object tracking-based segmentation algorithm to preserve discontinuities and a new combined segmentation-mutual-information (MI)-based algorithm for dense depth map estimation. It relies on segmentation, LPR-based depth map smoothing, and MI-based matching algorithm to iteratively estimate the depth map. The method is very flexible

Manuscript received March 18, 2011; revised October 11, 2011; accepted November 29, 2011. Date of publication May 7, 2012; date of current version September 28, 2012. This work was supported in part by the General Research Fund of the Hong Kong Research Grant Council. This paper was recommended by Associate Editor C. N. Taylor.

Z. Y. Zhu, S. Zhang, and S.-C. Chan are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: zyzhu@eee.hku.hk; szjeff@eee.hku.hk; scchan@eee.hu.hk).

H.-Y. Shum is with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: hshum@microsoft.com).

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/TCSVT.2012.2198133

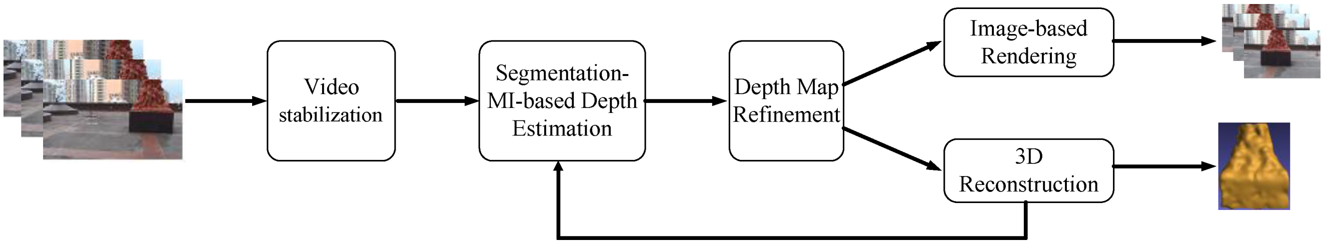


Fig. 1. Block diagram of the proposed object-based rendering and 3-D reconstruction algorithm using the M-IBR system constructed.

and both semiautomatic and automatic segmentations can be used. The semiautomatic and automatic versions rank 4 and 6, respectively, in the Middlebury comparison of existing depth estimation methods. Using the depth maps captured and the object-based approach [11], high-quality renderings of outdoor scenes along the trajectory can be obtained, which considerably improved the viewing freedom.

3) A 3-D reconstruction module for objects, which employs the estimated dense depth maps to obtain dense point correspondences from multiple views for 3-D reconstruction. For stationary objects, the M-IBR system can be driven around the object to obtain sufficient correspondences from different views, which can be integrated together for 3-D reconstruction. To this end, the sequential structure-from-motion (S-SFM) technique is first adopted to estimate the locations of the M-IBR system so as to obtain an initial set of fairly reliable 3-D point cloud from the 2-D correspondences. New iterative Kalman filter (KF)-based and segmentation-MI-based algorithms are proposed to fuse the correspondences from different views and remove possible outliers to obtain an improved point cloud. More precisely, the proposed algorithm relies on the KF to track the correspondences across different views so as to suppress possible outliers while fusing correspondences from different views. With these reliable matched points, the camera parameters and hence the image correspondences can be further refined by reprojecting the updated correspondences to successive views to serve as prior features/correspondences for MI-based matching. By iterating these processes, an improved point cloud with reliable correspondences can be recovered. Simulation results show that the proposed algorithm significantly reduces the adverse effect of the outliers and generates a more reliable point cloud. To recover the 3-D model from the improved point cloud, a new robust radial basis function (RBF)-based modeling algorithm is proposed to further suppress possible outliers and generate smooth 3-D surfaces from the raw 3-D point cloud. Compared with the conventional RBF-based smoothing, it is more robust and reliable.

The system flow of the proposed system is summarized in Fig. 1. Experimental results show that high-quality renderings of outdoor dynamic plenoptic videos and 3-D geometry of objects can be obtained using the proposed system and algorithms. Since dynamic objects may only be partially visible, only partial 3-D models can be recovered. Potential applications of the system include indoor/outdoor environmental modeling and multiview video recording.

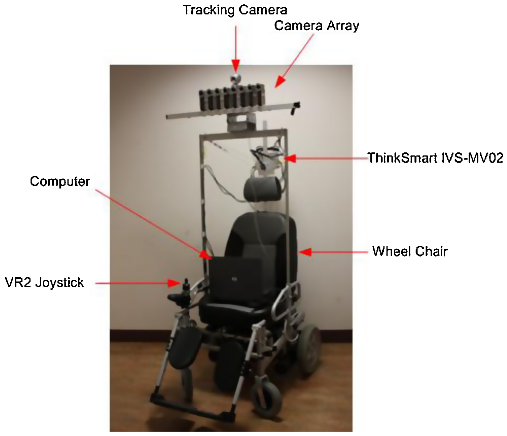


Fig. 2. Proposed M-IBR system.

This paper is organized as follows. The design and development of the proposed prototype M-IBR system are summarized in Section II. The problems of video stabilization and object segmentation/tracking are discussed in Section III. Section IV is devoted to the segmentation-MI-based dense depth map estimation algorithm. The 3-D reconstruction of scene object(s) using SFM, iterative KF-based and MI-based algorithm, and smoothed robust RBF is presented in Section V. Finally, conclusions are drawn in Section VI.

II. Construction of the Proposed M-IBR System

As mentioned previously, the M-IBR system consists of a linear array of cameras mounted on an electrically controllable wheel chair so as to cope with moving objects in a large environment and to improve the viewing freedom of users. Fig. 2 shows the M-IBR system that we have constructed. It consists of a linear array of eight Sony HDR-TGIE high-definition (HD) video cameras that are mounted on a FS122LGC wheel chair.

The motion of the wheel chair is originally controlled manually through a VR2 joystick and power controller modules from PG drives technology [14]. To make it electronically controllable, we examined the output of the joystick and generated the (x-, y-) motion control voltages to the power controller using a Devasys USB-I2C/IO microcontroller unit [15]. By appropriately controlling these voltages, the motion of the wheel chair can be controlled electronically. Moreover, by

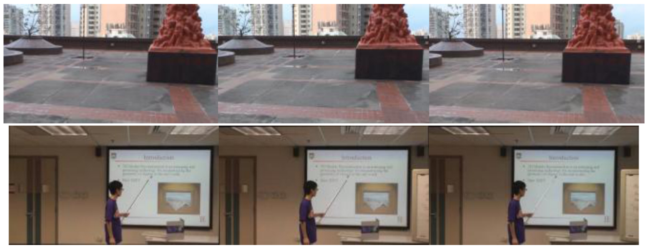


Fig. 3. Snapshots of the plenoptic videos at a given time instance: the upper row is the *podium* outdoor video from camera 1 to camera 3 and the lower row is the *presentation* indoor video from camera 1 to camera 3.

using the wireless LAN of a portable notebook mounted on the wheel chair, its motion can be controlled remotely to improve the viewing freedom. Fig. 3 shows snapshots of an outdoor and indoor plenoptic videos captured by the proposed system called *podium* and *presentation*, respectively. The resolution of these real-scene plenoptic videos is 1920 × 1080i with 25 f/s in a 24-bit RGB format.

A preliminary system was illustrated in [21] where the HD videos were captured in real time into the storage cards of cam-corders. They can be downloaded to a PC for further processing, such as calibration, depth estimation, and rendering using the object-based approaches [10], [11], [16], [17]. For real-time transmission, the cam-corders were equipped with a composite video output that can be further compressed and transmitted. To illustrate the concept of multiview conferencing, a ThinkSmart IVS-MV02 Intelligent Video Surveillance System [18] was used to compress the (320 × 240) 30 f/s videos online, which can be retrieved remotely through the wireless LAN for viewing or further processing. The system was built from Analog Device DSP and it can achieve real-time compression at a bit rate of 400 kb/s. This paper greatly extended the previous work [21] by introducing an improved video stabilization and segmentation-MI-based depth estimation algorithms. Moreover, a systematic approach for 3-D reconstruction is proposed.

III. Video Stabilization and Object tracking

A. Video Stabilization

To ensure good tracking of objects and to obtain more image samples for high-quality rendering, the wheel chair is usually driven steadily during capturing. However, one problem with M-IBR system is that the ground surfaces may not be smooth and the whole mechanical structure can vibrate considerably during motion. In our M-IBR system, the shaky motion of the camera array of the outdoor environment seems to come from the roughness of the ground surfaces and the vibration of the mechanical structure during the movement. Besides, the video captured may also appear shaky when the system is moving and about to settle down in indoor environment. To reduce these annoying effects, video stabilization [23]–[26], [34], [53]–[55] is frequently employed to eliminate the undesired motion fluctuation in the captured videos.

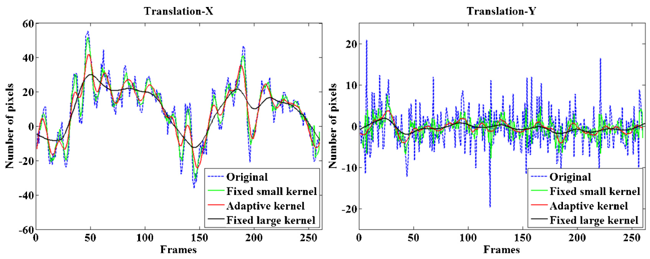
The development of video stabilization can be traced back to the work of Ratakonda [23], who first proposed a method using profile matching and subsampling to produce a low-resolution stabilized video stream in real time. Later, Chang *et al.* [24] presented an approach based on optical flow. Matsushita *et al.* [25], [26] developed an improved method for compensating the shaky motion of camera and proposed an approach called motion inpainting to fill in the missing areas. Agarwala *et al.* [53]–[55] proposed a novel stabilization framework based on estimating camera motion, generating the virtual 3-D camera path and view synthesis. Recently, scale-invariant feature transformation (SIFT) is widely used in solving the problem of video stabilization [34]. The features extracted by SIFT are affine invariant and insensitive to the change of the scale and luminance.

As mentioned above, our M-IBR system was driven steadily during capturing. Therefore, the undesired motion fluctuation will usually appear as high-frequency components compared to the intentional motion. As a result, the problem of video stabilization can also be viewed as the removal of high-frequency components in the estimated velocity. To this end, one needs to estimate the global motion of the camera, say, by means of optical flow on the video sequence so that this annoying high-frequency local motion can be removed to stabilize the videos.

The proposed algorithm is divided into three major steps as follows.

1) *Global motion estimation*: first, the geometric transformation between a location $\boldsymbol{x} = [x_1, x_2]^T$ in a frame with that in an adjacent frame, $\boldsymbol{x}'$, is modeled by an affine transformation $\boldsymbol{x}' = \boldsymbol{T}[\boldsymbol{x}] = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{t}$, where $\boldsymbol{t} = [t_{x_1}, t_{x_2}]^T$ is the translational component and the affine rotation, scaling, and stretch are represented by the matrix $\boldsymbol{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$. In homogeneous coordinates, $\boldsymbol{x}_h = [x_1, x_2, 1]^T$, $\boldsymbol{T}$ can be conveniently represented by a matrix multiplication $\boldsymbol{T}_h\boldsymbol{x}_h$, where $\boldsymbol{T}_h = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix}$. $\boldsymbol{T}$ is estimated from the tracked features in adjacent video frames using the SIFT [33], instead of the Lucas–Kanade tracker in [21].
2) *Local smoothing of motion*: the intentional motion, which is assumed to be slow and smooth, is then estimated by smoothing the global motion estimated using LPR with adaptive bandwidth selection [36]. Unlike conventional methods, the bandwidth or window size for smoothing can be automatically determined. This will be further discussed below.
3) *Video completion*: the uncovered areas are filled using motion inpainting [25], [26].

We now describe each step in more detail. Let $\{I_t(\boldsymbol{x})|t = 0, \cdots, N\}$, where $\boldsymbol{x} = [x_1, x_2]^T$, $1 \le x_1 \le \aleph_1$, $1 \le x_2 \le \aleph_2$, be a video sequence consisting of N video frames with resolution $\aleph_1 \times \aleph_2$ captured by our M-IBR system. Consider the global motion transformations up to time instant t, $\{\boldsymbol{T}_0^1, \ldots, \boldsymbol{T}_{t-1}^t\}$, where $\boldsymbol{T}_i^{i+1}$ is the coordinate transformation from the ith to the $(i + 1)$th frame. If $\boldsymbol{T}_i^{i+1}$ is smoothed separately, a smoothed transformation chain $\{\hat{\boldsymbol{T}}_0^1, \ldots, \hat{\boldsymbol{T}}_{t-1}^t\}$ is obtained and the tth compensated image frame I_t' can be



(a) (b)

Fig. 4. Motion smoothing results for (a) horizontal and (b) vertical directions. The original motion path and the smoothed motion path with different methods are shown. The blue dotted lines correspond to the shaky original motion path. Green and black lines correspond to the smoothed motion path using the method in [25] with a small and a large kernel sizes, respectively.

obtained as follows:

$$I'_t\left(\prod_{i=0}^{t-1}(\boldsymbol{T}^i_{i+1}\hat{\boldsymbol{T}}^{i+1}_i)[\boldsymbol{x}]\right) = I_t(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{T}^i_{i+1}$ and $\hat{\boldsymbol{T}}^{i+1}_i$ denote, respectively, the transformation from frame $i+1$ to i and the smoothed transformation from frame i to $i+1$. In order to avoid error accumulation due to the cascade of original and smoothed transformation chains, [25] proposed to compute directly the transformation $\tilde{\boldsymbol{T}}_t$ from the current frame $I_t(\boldsymbol{x})$ to the corresponding motion compensated frame $I'_t(\boldsymbol{x})$ using only the neighboring transformation matrices as $\tilde{\boldsymbol{T}}_t = \sum_{i\in\Omega_t} \boldsymbol{T}^i_t \otimes G(i)$, where $\Omega_t = \{f : |f-t| \le \eta\}$ are the indices of neighboring frames, $G(x) = (\sqrt{2\pi}\sigma)^{-1}e^{-x^2/2\sigma^2}$ is a Gaussian kernel, 2η is the support of Ω_t or window size, and $\otimes$ denotes the element-wise convolution operation.

It can be seen that the selection of the kernel size affects the degree of smoothing. A large kernel size will lead to the problem of oversmoothing, while a small kernel size may not be able to remove the high-frequency undesirable motion. The green and black lines in Fig. 4 illustrate the effect of using a small kernel size of $\eta = 3$ and a large kernel size of $\eta = 20$, respectively, using the method in [25].

To address this issue, we propose a new method for choosing adaptively the kernel size using LPR with adaptive bandwidth selection. The close relationship between curve fitting and video stabilization has been recognized, for example, in [34], where a local parabolic fitting is used to compute the smoothed motion path. However, the kernel size is also fixed. The advantage of our method is that the kernel size can be adaptively selected from the data.

LPR is a very flexible and efficient nonparametric regression method in statistics, and it has been widely applied in many research areas, such as data smoothing, density estimation, and nonlinear modeling. Given a set of noisy samples of a signal, the data points are fitted locally by a polynomial using the least-squares (LS) criterion with a kernel function having certain bandwidth parameters. Since signals may vary considerably over time, it is crucial to choose a proper kernel size or local bandwidth to achieve the best basis-variance tradeoff. In this paper, we used the refined intersection of confidence intervals (R-ICI) method to perform bandwidth selection. Here, we follow the homoscedastic data model of the time series

$$Y_i = m(X_i) + \sigma(X_i)\varepsilon_i \tag{2}$$

where $\{(Y_i, X_i)|i = 1, 2, \cdots, n\}$ are a set of univariate observations, $m(X_i)$ is a smooth function specifying the conditional mean of Y_i given X_i, and ε_i is an independent identically distributed (i.i.d.) additive white Gaussian noise. The problem is to estimate $m(X_i)$ and its kth derivative $m^{(k)}(X_i)$ from the noisy sample Y_i so as to achieve smoothing. Since $m(X_i)$ is a smooth function, we can approximate it locally as a general degree-p polynomial at a given point x_0

$$\begin{aligned} m(x) &\approx m(x_0) + m'(x_0)(x-x_0) \\ &\quad + m''(x_0)(x-x_0)^2/(2!) + \cdots + m^{(p)}(x_0)(x-x_0)^p/(p!) \\ &= \beta_0 + \beta_1(x-x_0) + \cdots \beta_p(x-x_0)^p \end{aligned} \tag{3}$$

where x is in the neighborhood of x_0 and β_k $(k = 0, 1, \cdots, p)$ is the kth polynomial coefficient. The coefficient vector $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_p]^T$ at location x_0 can be found by solving the following weighted least-squares regression problem:

$$\min_{\beta}\left\{\sum_{i=1}^{n} K_h(X_i - x_0)\left[Y_i - \sum_{k=0}^{p}\beta_k(X_i - x_0)^k\right]^2\right\} \tag{4}$$

where $K_h(X_i - x_0) = K(\frac{X_i - x_0}{h})/h$, $K(\cdot)$ is a kernel function with bandwidth parameter h, which emphasizes the influence of neighboring observations around x_0 in the estimation. The parameter h is adaptively chosen at different locations x_0 so as to adapt to the local characteristics of the signal (i.e., the intentional motion path). Differentiating the objective function in (4) with respect to $\boldsymbol{\beta}$ and setting the derivative as zero, we get the following LS solution in the matrix form:

$$\hat{\boldsymbol{\beta}}(x_0, h) = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y} \tag{5}$$

where

$$\boldsymbol{X} = \begin{bmatrix} 1 & (X_1 - x_0) & \cdots & (X_1 - x_0)^P \\ 1 & (X_2 - x_0) & \cdots & (X_2 - x_0)^P \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (X_n - x_0) & \cdots & (X_n - x_0)^P \end{bmatrix}$$

$\boldsymbol{y} = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix}^T$, and $\boldsymbol{W} = \text{diag}\{K_h(X_i - x_0)\}$ is the weighting matrix.

By estimating $\hat{\boldsymbol{\beta}}(x_0, h)$ with an optimized bandwidth h at different x_0, we obtain a smoothed representation of the data from the noisy observations. In the context of video stabilization, a key problem of applying LPR is thus to select an optimal bandwidth parameter h to achieve the best bias-variance tradeoff in estimation. Here, we use the R-ICI bandwidth selection algorithm [35] to select the optimal bandwidth. The basic idea of the R-ICI adaptive bandwidth selection method is to calculate a set of smoothing results with different bandwidths and then to examine a sequence of confidence intervals of these smoothing results to determine and refine the optimal bandwidth. In this paper, the kernel $K(u)$ is chosen as the Epanechnikov kernel $K(u) = (3/4)(1 - |u|^2)_+$ and the bandwidth parameter set for R-ICI is $\mathrm{H} = \{h_j : |h_j = a^j/N, j = 1, \cdots, 10\}$ with $a = 1.2$. N is the total number of frames.

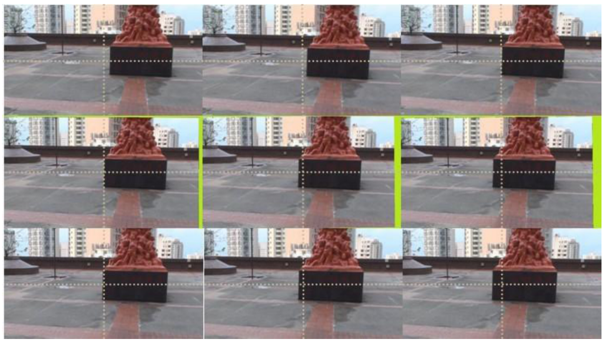


Fig. 5. Video stabilization result. The first row shows the original images captured by our system, the second row shows the stabilized images without video completion, and the third row shows the completed results.

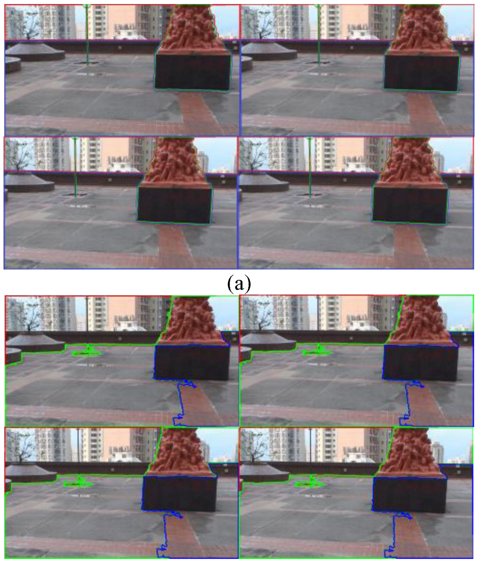
Due to page limitation, the details of the algorithm are omitted and interested readers are referred to [35] and [36].

After video stabilization, some of the pixels at the boundaries may be missing, as illustrated in the second row of Fig. 5. These missing areas can be filled in or completed by video completion using motion inpainting [26], which can propagate the motion field into the missing areas rather than simply propagating the RGB color values.

Fig. 4(a) and (b) shows the original and smoothed translational motion in the x and y directions (i.e., t_{x_1} and t_{x_2} in $\boldsymbol{T}$). Green and black lines show the motion obtained by a fixed small kernel and a fixed large kernel using the method in [25], respectively. The red lines show the motion obtained by using the LPR with the R-ICI (LPR-R-ICI) bandwidth selection method over time. It can be seen that the proposed LPR-R-ICI method is able to suppress the high-frequency components while preserving the smooth intentional motion. Example original images, stabilized results, and inpainted results using the proposed method are shown in Fig. 5.

B. Object Tracking Using Level-Set Method

For rendering (intermediate view synthesis) and 3-D reconstruction of a given object in the scene, it is usually advantageous to segment the object for further processing so as to preserve depth discontinuities. Following the object-based approach [21], we first segment the object at a given frame using a semiautomatic segmentation method. Object tracking is then employed to track and segment the object automatically in subsequent frames and adjacent views. In our system, the initial segmentation for each camera is performed by means of Lazy snapping [22]. Then, the object at other time instants of each camera is tracked using the level-set method [10]. Example video tracking results are shown in Fig. 6(a). It can be seen that major depth discontinuities are well delineated. If automatic segmentation methods are used to obtain the initial segmentation, the segmented part may sometimes involve more objects. Moreover, part of background and foreground will be grouped together. Although such segmentation is, in principle, consistent in terms of image intensity, depth discontinuities may not be preserved as well. As an illustration, a graph-cut-based automatic segmentation method [37] is used to obtain an initial segmentation for tracking. Fig. 6(a) and (b) compares the example results using the semiautomatic-based and automatic-segmentation-based tracking. It can be seen that the former gives a better result than the latter, for example, at the light pole and foreground of the scene. To obtain a better rendering result, the semiautomatic method will be adopted in this paper, though our framework also works for automatic methods. We now describe the proposed depth estimation method.



(a)

(b)

Fig. 6. Segmentation results using the level-set-based tracking method. (a) Initial segmentation obtained by lazy snapping. (b) Initial segmentation obtained by the graph cut method.

IV. Segmentation and MI-Based Dense Depth Map Estimation

Conventional depth estimation techniques are mostly based on computing the correspondences from stereo or multiple views using feature point matching. More recent algorithms employ Markov random field [39] to model the observation and estimate the depth map by maximizing the *a posterior* probability. In particular, graph-cuts (GC)-based [40] and belief-propagation (BP)-based [41] methods for performing the optimization have been widely used because of their good performances. The success of these methods depends critically on how the physical phenomena, such as occlusion, edges, color correlation, are modeled. Techniques, such as occlusion penalization [47], visibility checking [41], [46], and structural information [42]–[46], are areas of active research. Another popular direction is to combine segmentation with GC or BP [42]–[46].

In this paper, we proposed a modified MI-based dense matching algorithm by utilizing prior segmentation information. The segmentation information, which can be obtained semiautomatically and automatically, considerably reduces possible matching errors arising from occlusion. Apart from its flexibility, the proposed algorithm only involves the selection of few parameters and it works well for indoor scenes and outdoor scenes.

A. MI Matching

Since we wish to perform IBR and 3-D reconstruction of selected object(s) in the scene using multiview videos, the first step is to establish dense 2-D correspondences between adjacent views so as to generate a dense point cloud for 3-D reconstruction or depth maps for rendering. In our M-IBR system, there are eight cameras and hence eight views are obtained at each time instance for depth estimation. Here, a modified MI-based dense matching algorithm with segmentation is employed. As we have segmented the images into several parts, the whole matching process is performed on each image segment. In the sequel, we shall use "image" to denote the segmented parts in the image.

MI of two random variables X and Y is defined as $I(X;Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p_X(x)p_Y(y)}\right) dxdy$, where $p_X(x)$ and $p_Y(y)$ are the probability density function (pdf) of X and Y. $I(X;Y)$ can also be expressed in terms of the entropy and joint entropy of X and Y as follows:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{6}$$

where $H(X,Y) = -\int_Y \int_X p(x,y) \log p(x,y) dxdy$ is the joint entropy of X and Y, and $H(X) = -\int_X p(x) \log p(x) dx$ and $H(Y) = -\int_Y p(y) \log p(y) dy$ are the entropy of X and Y, respectively. Intuitively, MI measures the information that X and Y share by measuring how much knowing one of these variables reduces the uncertainty about the other.

In [27], a free-form deformation method using spline function was proposed for 2-D shape registration in pattern recognition systems. We now extend it to our segmentation-MI-based depth estimation algorithm. More precisely, the intensities of the two rectified image segments, say A and B, from two views to be matched are treated as random variables with pdfs, $p_A(i_A)$ and $p_B(i_B)$, and joint pdf $p_{AB}(i_A, i_B)$. B is then deformed by means of the disparity transformation function $T(B)$ with parameters to be determined. Ideally, when the two images are registered, the MI, $I(A; T(B))$, will be maximized. Therefore, by maximizing $I(A, T(B))$ using the parameters of $T(\cdot)$, the two original image segments can be registered to infer their correspondences. Consequently, the entropies can be calculated from the probability density functions as follows:

$$\begin{aligned} H(p_A(i_A)) &= -\int_{I(A)} p_A(i_A) \log p_A(i_A) di_A \\ &= -\iint_{I(A),I(B)} p_{A,T(B)}(i_A, i_B) \log p_A(i_A) di_A di_B \end{aligned} \tag{7}$$

$$\begin{aligned} H(p_{T(B)}(i_B)) &= -\int_{I(B)} p_{T(B)}(i_B) \log p_{T(B)}(i_B) di_B \\ &= -\iint_{I(A),I(B)} p_{A,T(B)}(i_A, i_B) \log p_{T(B)}(i_B) di_A di_B \end{aligned} \tag{8}$$

$$\begin{aligned} &H(p_{A,T(B)}(i_A, i_B)) \\ &\quad = -\iint_{I(A),I(B)} p_{A,T(B)}(i_A, i_B) \log p_{A,T(B)}(i_A, i_B) di_A di_B \end{aligned} \tag{9}$$

where i_A and i_B are the intensity valuables of A and $T(B)$ and their ranges are $I(A)$ and $I(T(B)) = I(B)$, respectively. The latter follows from the fact $T(\cdot)$ is a disparity transformation that does not change the range of the intensity values. To proceed further, one needs to determine the corresponding pdfs. A powerful method for approximating the pdfs is the kernel method [51], which approximates the pdfs directly from the image data as follows:

$$p_A(i_A) \approx \frac{1}{V} \iint_\Omega G_1(i_A - I_A(x_1, x_2)) dx_1 dx_2 \tag{10}$$

$$p_{T(B)}(i_B) \approx \frac{1}{V} \iint_\Omega G_1(i_B - I_{T(B)}(x_1, x_2))) dx_1 dx_2 \tag{11}$$

$$\begin{aligned} &p_{A,T(B)}(i_A, i_B) \\ &\quad \approx \frac{1}{V} \iint_\Omega G_2(i_A - I_A(x_1, x_2), i_B - I_{T(B)}(x_1, x_2)) di_A di_B \end{aligned} \tag{12}$$

where $G_1(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/(2\sigma^2)}$, $G_2(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-(x_1^2/\sigma_1^2 + x_2^2/\sigma_2^2)/2}$, $I_A(x_1, x_2)$ and $I_B(x_1, x_2)$ are the intensities of A and B at pixel location $\boldsymbol{x} = [x_1, x_2]^T$, $I_{T(B)}(x_1, x_2) = I_B(T^{-1}(x_1, x_2))$, T^{-1} is the inverse function of T, Ω is the image domain, and V is the area of Ω. In practice, the integrals are approximated by summing over the pixel coordinates.

For accurate matching, the transformation in our approach is carried out in two steps, namely, global and local transformations. In global transformation, which is performed first, the parameters of a global transformation are determined by matching the two images so as to model their relative scale, translation, and rotation. It can be derived as follows:

$$E_{\text{global}} = -\iint_{I(A),I(B)} p_{A,T(B)}(i_A, i_B) \log \frac{p_{A,T(B)}(i_A, i_B)}{p_A(i_A) p_{T(B)}(i_B)} di_A di_B. \tag{13}$$

In the local transformation step (refinement), local deformation is performed, which is represented by a 2-D spline function. The transformation parameters, which are the displacement vectors at a regular grid to interpolate the spline function, are determined by minimizing the function in (13).

In this paper, the global deformation function T_G is chosen as an affine transformation. The model parameters can be obtained by minimizing the objective function in (13). Let B' be the transformed image obtained by the affine transformation after the first step. The local transformation $T_L(B')$, which is a 2-D spline function, is parameterized by the displacement vectors at a uniform grid of control points $\{C\}$, $\boldsymbol{P}_c(m,n) = [P_{c_1}(m,n), P_{c_2}(m,n)]^T$ for $m = 1, \ldots, \hat{M}, n = 1, \ldots, \hat{N}$. If $(\aleph_1, \aleph_2)$ is the resolution of the input image, the spacing of the control points in the x and y directions is $\Delta_1 = \aleph_1/\hat{M}$ and $\Delta_2 = \aleph_2/\hat{N}$, respectively. The deformation of any pixel in the image is obtained by spline interpolation of those at the grid points $\{C\}$. Therefore, the deformation of pixel (x_1, x_2), $\boldsymbol{P}(x_1, x_2) = [P_1(x_1, x_2), P_2(x_1, x_2)]^T$ can be written as follows:

$$\boldsymbol{P}(x_1, x_2) = \Sigma_{\mu=0}^{3} \Sigma_{\gamma=0}^{3} \beta(l) \beta(v) \boldsymbol{P}_c(m + \mu, n + \gamma) \tag{14}$$

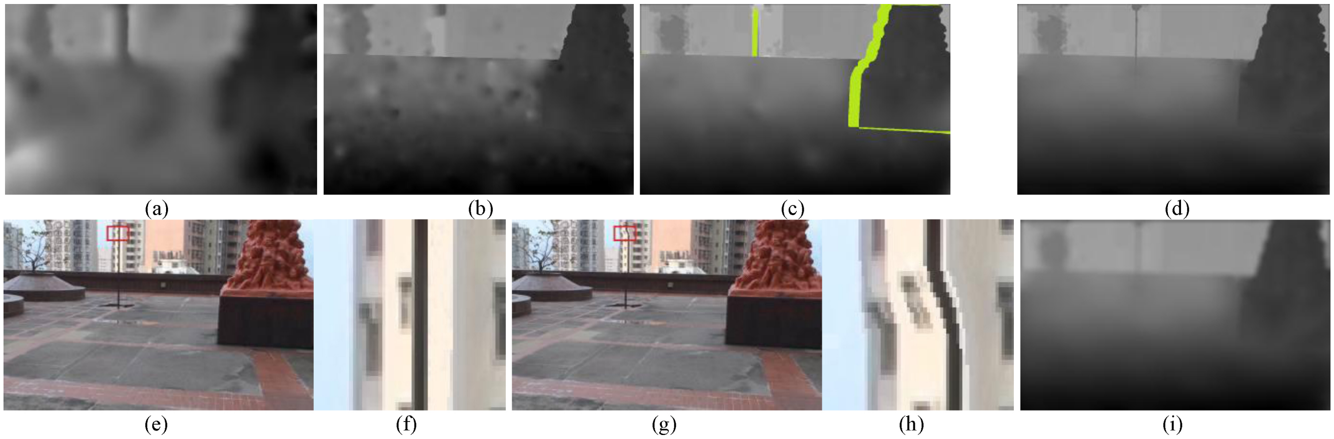


Fig. 7. (a) Example depth map obtained by using MI matching without segmentation information. (b) Depth map obtained by using automatic segmentation MI matching. (c) Depth map obtained by using semiautomatic segmentation MI matching. Green areas in (c) are the occlusion areas detected by our algorithm. (d), (i) Refined depth maps of (c) by inpainting and smoothing (c) using SK-LPR-R-ICI and 25 × 25 ideal low-pass filter, respectively. (e), (g) Renderings obtained by (d) and (b). (f), (h) Enlargements of the red boxes in (e) and (g), respectively.

for $1 \le x_1 \le \aleph_1, 1 \le x_2 \le \aleph_2$, where $l = (x_1/\Delta_1) - \lfloor x_1/\Delta_1 \rfloor$, $v = (x_2/\Delta_2) - \lfloor x_2/\Delta_2 \rfloor$, β (u) is the cubic β-spline function, $\{(m+\mu, n+\gamma)|(\mu, \gamma) \in [0, 3]\}$ are the neighboring control points of (x_1, x_2). The collection of $\boldsymbol{P}_c(m, n)$ forms the parameters of the transformation function given by $T_L(B'(x_1, x_2)) = B'(x_1+P_1(x_1, x_2), x_2+P_2(x_1, x_2))$. With (13) and (14), one gets the local matching data term to be minimized

$$E_{\text{data}} = -\iint_{I(A),I(B)} p_{A,T_L(B')}(i_A, i_B) \log \frac{p_{A,T_L(B')}(i_A, i_B)}{p_A(i_A) p_{T_L(B')}(i_B)} di_A di_B. \tag{15}$$

In order to reduce the variance of the variables, an additional smoothing term and other prior constraints can be added to the data term above. A popular smoothing term is the L_2 norm of the displacement vectors $E_{\text{smooth}} = \Sigma_{(m,n)}(\|\boldsymbol{P}_c(m, n)\|^2 + \|\nabla \boldsymbol{P}_c(m, n)\|^2)$. Since there are a few model parameters in affine transformation, the smoothing term is only needed in local matching.

If the pair of images being registered does have distinct geometric features as correspondences, incorporating this feature information can greatly improve accuracy and efficiency. In our algorithm, the scale invariant features can be used as a feature term. These constraints can be conveniently integrated into our registration framework. More precisely, assuming that the total number of feature points is R, and their corresponding locations at A and B' are $\mathbf{x}_{A,r}$ and $\boldsymbol{x}_{B',r}$, $r = 1, \cdots, R$, respectively, then the following energy term can be incorporated as a feature term:

$$E_{\text{feature}} = \sum_{r=1}^{R} D(\boldsymbol{x}_{A,r}, \boldsymbol{x}_{B',r}) \tag{16}$$

where $D(\boldsymbol{x}_{A,r}, \boldsymbol{x}_{B',r})$ is an appropriate distance measure such as the Euclidean distance between $\boldsymbol{x}_{A,r}$ and $\boldsymbol{x}_{B',r}$. Therefore, the object function in the local transformation is as follows:

$$E_{\text{local}} = E_{\text{data}} + E_{\text{smooth}} + E_{\text{feature}}. \tag{17}$$

The L-BFGS algorithm [28] is used to solve for the unconstrained nonlinear optimization problems. An advantage is that the explicit evaluation of the Hessian matrix is not required, since it can be recursively estimated. Moreover, it was found to be much faster than using the conventional level set method [27]. Because there is no need to compute the whole Hessian matrix, the storage space of L-BFGS is less than other conventional algorithms, such as belief propagation. Because of this reason, the L-BFGS method can deal with large problems such as 1080P resolution images. Meanwhile, the L-BFGS method can converge to a local optimum in nonconvex problems under mild conditions as demonstrated in [28]. This is an important advantage over other conventional algorithms. Previous studies also analyzed and demonstrated the efficiency of the L-BFGS [28], especially in terms of function evaluations.

As mentioned previously, the segmented parts are processed one by one and they will be integrated to form the final depth map for matching one view to the other at each time instant.

B. Depth Map Refinement

Compared with the traditional MI method, the segmentation-MI-based method simplifies the preservation of depth discontinuities, and the smoothing and inpainting of depth maps. This is illustrated in Fig. 7 where example depth maps obtained by the MI algorithm without segmentation [Fig. 7(a)], with automatic segmentation [Fig. 7(b)], and semiautomatic segmentation techniques [Fig. 7(c)] are shown. The depth maps obtained by incorporating the segmentation information [Fig. 7(b), (c)] are considerably better than the one without segmentation information [Fig. 7(a)]. Moreover, the depth discontinuities at object boundaries and smoothness at flat regions are seen to be better preserved for the semiautomatic approach, which will significantly reduce the artifacts during rendering. However, due to noise, occlusion, and lower reliability of the matching process at low texture areas, the resulting depth maps may still contain errors. These issues will be addressed below through further refinement of the depth maps.

1) *Occlusion Detection and Inpainting:* Let the depth map obtained by the MI-based matching algorithm from the stabilized images $I'_{i,t}(\boldsymbol{x})$ to $I'_{i+1,t}(\boldsymbol{x})$ be $\Gamma^i_{i+1,t}(\boldsymbol{x})$, where

$x_1 = 1, 2, \cdots, \aleph_1$ $x_2 = 1, 2, \cdots, \aleph_2$, $i = 1, \ldots, M$ and $t = 0, \ldots, N$. Similarly, a depth map can be obtained from matching $I'_{i+1,t}(\boldsymbol{x})$ to $I'_{i,t}(\boldsymbol{x})$, which gives $\Gamma^{i+1}_{i,t}(\boldsymbol{x})$. If a pixel is not occluded, then the depth values of the same pixel in $\Gamma^{i}_{i+1,t}(\boldsymbol{x})$ and $\Gamma^{i+1}_{i,t}(\boldsymbol{x})$ should be similar to each other. If the absolute value of their difference is larger than a certain threshold, this pixel is considered to be occluded. In this paper, this threshold value is chosen as two pixels. Therefore, we can obtain a refined depth map $\Gamma_{i,t}(\boldsymbol{x})$ of each image after occlusion detection. For a comprehensive survey of occlusion detection algorithms, please see [38].

After these occluded pixels are detected, we need to inpaint the depth values at these occlusion areas [e.g., the green areas in Fig. 7(c)]. The occluded areas are inpainted by interpolation using the samples inside the corresponding segments, which avoids blurring at depth discontinuities if the conventional interpolation techniques are used.

2) *Smoothing of Depth Maps:* As mentioned earlier, depth map may contain an invalid value due to noise and regions with low texture, and others. Therefore, the depth maps should be further smoothed to reduce such estimation errors. Here, we adopt 2-D LPR with adaptive bandwidth selection [36], which is a 2-D generalization of the LPR-R-ICI method introduced before, for smoothing depth maps of different segments. It enables us to preserve the discontinuity at object boundaries while performing smoothing at flat areas. More precisely, we treat the depth map as a 2-D function $Y(x_1, x_2)$ of the coordinate $\boldsymbol{x} = [x_1, x_2]^T$ with $x_1 = 1, 2, \cdots, \aleph_1$ and $x_2 = 1, 2, \cdots, \aleph_2$

$$Y_i = m(\boldsymbol{X}_i) + \sigma(\boldsymbol{X}_i)\varepsilon_i \tag{18}$$

where $(Y_i, \boldsymbol{X}_i)$ is a set of observations with $i = 1, \cdots, n$. $\boldsymbol{X}_i = [X_{i,1}, X_{i,2}]^T$ is a 2-D explanatory variable. $m(\boldsymbol{X}_i)$ is a smooth function specifying the conditional mean of Y_i given $\boldsymbol{X}_i$, and ε_i is an independent identically distributed (i.i.d.) additive white Gaussian noise. The problem is to estimate $m(\boldsymbol{X}_i)$ from the noisy sample Y_i. Since $m(\boldsymbol{X}_i)$ is a smooth function, we can approximate it locally as a general degree-p polynomial at a given point $\boldsymbol{x} = [x_1, x_2]^T$

$$m(\boldsymbol{X} : \boldsymbol{x}) = \sum_{\kappa=0}^{p} \sum_{k_1+k_2=\kappa} \beta_{k_1,k_2} \prod_{j=1}^{2} (X_j - x_j)^{k_j} \tag{19}$$

where $\boldsymbol{\beta} = \{\beta_{k_1,k_2} : k_1 + k_2 = \kappa$ and $\kappa = 0, \ldots p\}$ is the vector of coefficients. The polynomial coefficient at a location $\boldsymbol{x}$ can be determined by minimizing the following weighted LS problem:

$$\min_{\beta} \sum_{\mathrm{i=1}}^{\mathrm{n}} K_H(\boldsymbol{X}_i - \boldsymbol{x})[Y_i - m(\boldsymbol{X}_i : \boldsymbol{x})]^2 \tag{20}$$

where $K_H(\cdot)$ is a suitably chosen 2-D kernel. When $\boldsymbol{x}$ is evaluated at a series of 2-D grid points, we obtain a smoothed depth map from the noisy depth estimates Y_i. Similar to (4), (20) can be solved using the LS method and the solution is as follows:

$$\hat{\boldsymbol{\beta}}_{LS}(\boldsymbol{x}, h) = (\Xi^T \Omega \Xi)^{-1} \Xi^T \Omega \Psi \tag{21}$$

where $\Omega = \mathrm{diag}\{K_H(\boldsymbol{X}_1 - \boldsymbol{x}), \cdots, K_H(\boldsymbol{X}_n - \boldsymbol{x})\}$ is the weighting matrix, $\Psi = [Y_1, Y_2, \cdots, Y_n]^T$

$$\Xi = \begin{bmatrix} 1 & (\boldsymbol{X}_1 - \boldsymbol{x})^T & vech\{(\boldsymbol{X}_1 - \boldsymbol{x})(\boldsymbol{X}_1 - \boldsymbol{x})^T\} & \cdots \\ 1 & (\boldsymbol{X}_2 - \boldsymbol{x})^T & vech\{(\boldsymbol{X}_2 - \boldsymbol{x})(\boldsymbol{X}_2 - \boldsymbol{x})^T\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (\boldsymbol{X}_n - \boldsymbol{x})^T & vech\{(\boldsymbol{X}_n - x)(\boldsymbol{X}_n - \boldsymbol{x})^T\} & \ldots \end{bmatrix}$$

and $vech(\cdot)$ is the half-vectorization operation. The following Gaussian kernel is employed in this paper:

$$K_H(\boldsymbol{u}) = \frac{1}{h^2(2\pi|\det C^{-1}|)} \exp\left(-\frac{1}{2}\boldsymbol{u}^T \boldsymbol{C}\boldsymbol{u}\right) \tag{22}$$

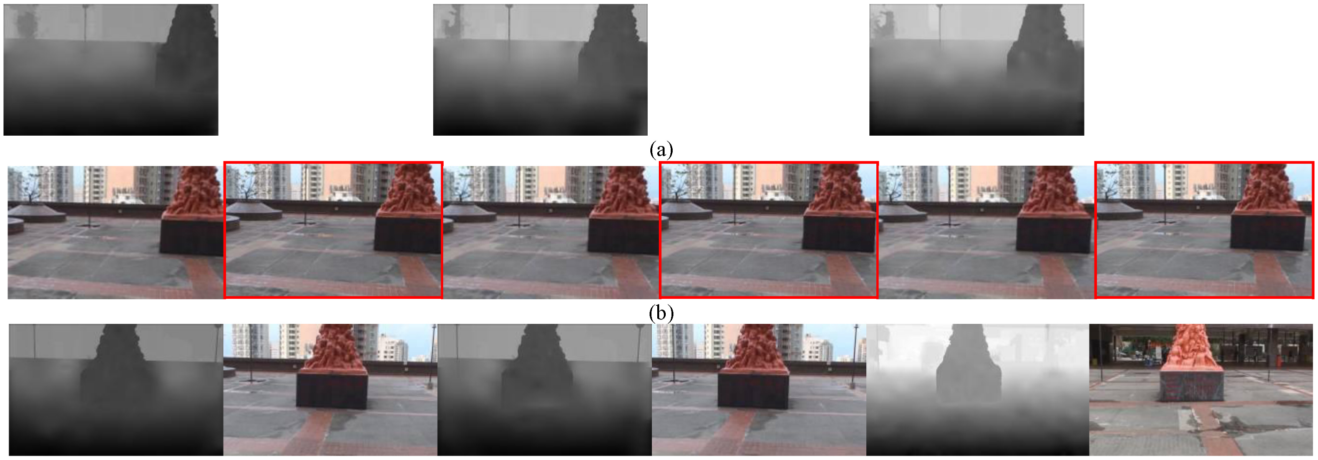
where the positive definite matrix $\boldsymbol{C}$ and scalar bandwidth h determine, respectively, the orientation and scale of the smoothing. Since the Gaussian kernel is not of compact support, it should be truncated to a sufficient size $\aleph_K \times \aleph_K$ to reduce the arithmetic complexity. Usually, $\boldsymbol{C}$ is determined from the principal component analysis of the data covariance matrix at $\boldsymbol{x}$. When h is small, noise in the depth map may not be removed effectively. On the contrary, a large-scale kernel better suppresses additive noise at the expense of possibly blurring of the depth maps. Here, we adopt the iterative steering kernel regression (ISKR) method in [36], which was shown to have a better performance than the conventional symmetric kernel [36], especially along image edges. In the ISKR method, the local scaling parameter was obtained as $h_i = h_0\gamma_i$, where h_0 and γ_i are, respectively, the global smoothing parameter and the local scaling parameter. The scale selection process is fully automatic and it can be performed by using the data-driven adaptive scale selection method with the R-ICI rule mentioned in Section III. The resulting method is called the SK-LPR-RICI algorithm and more details can be found in [36]. The depth map smoothed by SK-LPR-RICI is denoted by $\tilde{\Gamma}_{i,t}(\boldsymbol{x})$.

As an illustration, we also smooth the example depth map by a 25 × 25 ideal low-pass filter and the result is shown in Fig. 7(e). Comparing Fig. 7(e) with (d), we can see that the discontinuity of the object boundaries using SK-LPR-R-ICI is well preserved, while the object boundaries are blurred by the lowpass filter due to its fixed size and relatively large support for noise suppression. In order to illustrate the effect of these errors in the depth maps on the rendering qualities, example renderings are also shown in Fig. 7(f) and (h) according to the depth maps obtained from Fig. 7(d) and (b), respectively. It can be seen that inaccurate depth values produce obvious distortion of the light pole in Fig. 7(h) and (i). By combining this new MI-based depth estimation algorithm with our movable IBR system, we can obtain the depth map [Fig. 8(a), (c)] and synthesized views [Figs. 8(b), 9(b)] at nearby locations of the trajectory in indoor/outdoor environment.

V. 3-D Reconstruction and Modeling

A. Structure From Motion

In order to perform 3-D reconstruction, the camera must be calibrated to determine the intrinsic parameters, as well as their extrinsic parameters, i.e., their relative positions and poses. In


(a)

(b)

(c)

Fig. 8. Rendering results obtained by the proposed algorithm. (a) Depth maps corresponding to images in (b). The highlighted images in (b) show the rendered views from the adjacent views in (b) using depth maps in (a). (c) Depth maps at other positions.

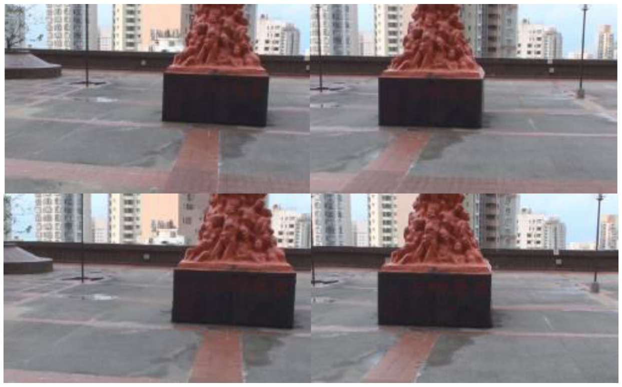


Fig. 9. Example rendering results. The first row shows the original images captured by our M-IBR system. The second row shows renderings with a step-in ratio of 1.15 times.

this paper, we employ the plane-based calibration method [19] and the SFM method [20] to determine the camera projection matrices, which connect the world coordinate and the image coordinate, of our M-IBR system. This is accomplished by using a sufficient large checkerboard calibration pattern to determine the intrinsic parameters and therefore only the extrinsic parameters need to be computed by structure-from-motion. This greatly reduces the degree of freedom of the camera projection matrices and hence improves the accuracy of calibration.

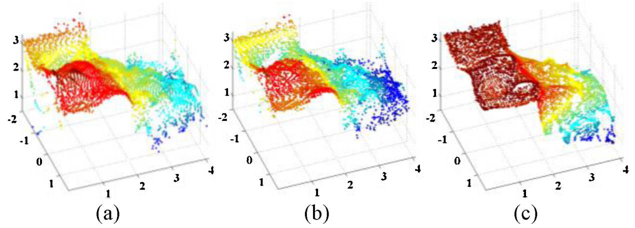
SFM combined with self-calibration [20] is a useful geometry reconstruction method, which can estimate 3-D object positions and projection matrices without any prior knowledge of the camera motion and structure of the scene. Sequential methods (S-SFM) and factorization methods (F-SFM) are two commonly used approaches in SFM. S-SFM works with each view sequentially by incorporating the results obtained in previous views. In contrast, F-SFM works by computing camera pose and scene geometry using all image measurements simultaneously. F-SFM is in principle more accurate once it converges to the global minimum, but it requires accurate initialization and is computationally more expensive. In practice, sequential methods are usually adopted and the factorization method can be used as a refinement if necessary. Moreover, most factorization methods only assume simplified linear camera models, e.g., orthographic, weak perspective, and paraperspective. Therefore, we shall employ the S-SFM method in this paper.

Our S-SFM algorithm consists of four major steps: 1) tracking of 2-D feature points in the whole image sequence using SIFT; 2) determination of an initial solution for the camera motion (extrinsic parameters), since the intrinsic parameters are known from calibration; 3) extending and optimizing the solution for every additional view; and 4) optimization of the camera motion globally using bundle adjustment. A comprehensive summary of the S-SFM algorithm can be found in [20].

After S-SFM, cameras are fully calibrated. In addition, an initial 3-D point cloud of the object in the scene can be obtained. However, in order to reconstruct a more accurate 3-D model, we need to refine the 3-D point cloud to remove outliers, and so on.

B. Point Cloud Generation and Refinement

After the dense depth map estimation, a set of point correspondences from multiple views are obtained. For 3-D reconstruction of stationary objects, the M-IBR system can be driven around them to obtain more views for reconstruction. Using the S-SFM technique, the camera projection matrices of the M-IBR system can be estimated. This allows a set of 3-D points to be computed from their correspondences in adjacent views through triangulation [20]. More precisely, from the depth map between views i and $i+1$, one can get a set of corresponding image points from views i and $i+1$ and their 3-D locations with the help of the estimated camera parameters and triangulation. To determine more accurate location of a 3-D point that is visible to all cameras, we need to track its correspondences across multiple views. Suppose that we start with a pair of correspondences between views 1 and 2. Let its estimated 3-D location obtained by triangulation be $z(1)$. Using the depth map between views 2 and 3, one



(a) (b) (c)

Fig. 10. Iterative refinement of point cloud. (a) Initial point cloud. (b) Point cloud after outlier detection and Kalman filtering. (c) Point cloud after the proposed iteration method.

can also determine another correspondence of this point and its estimated location $z(2)$. By continuing this operation repeatedly for subsequent views, we get more estimates $z(i)$ from views i and $i+1$, for $i = 1, \cdots, M-1$, where M is the total number of views. An example set of 3-D points obtained is shown in Fig. 10(a). However, outliers may exist due to errors in mutual segmentation-MI-based matching and estimating the projection matrices in S-SFM and occlusion, and so on. Therefore, the estimated locations cannot simply be averaged. In this paper, a Kalman-filter-based method is proposed to track the location $z(i)$ and detect possible outliers so that the point cloud can be refined by fusing different views. Moreover, an iterative method for further refining the point cloud will be introduced.

1) *KF-Based Outlier Detection and Point Cloud Fusion:* KF is the minimum mean-squares state estimator of the following linear state-space model with Gaussian innovation and measurement noise:

$$\boldsymbol{x}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t-1) + \boldsymbol{w}(t) \tag{23}$$

$$z(t) = \boldsymbol{H}(t)\boldsymbol{x}(t) + \delta(t) \tag{24}$$

where $\boldsymbol{x}(t) \in R^n$ and $z(t) \in R^m$ are, respectively, the state vector and observation vector at time t. $\boldsymbol{F}(t) \in R^{n\times m}$ and $\boldsymbol{H}(t) \in R^{m\times n}$ are the state transition and observation matrices, respectively, and the innovation $\boldsymbol{w}(t) \in R^n$ and measurement $\boldsymbol{\delta}(t) \in R^m$ noise are zero mean Gaussian noise with covariance matrix $\boldsymbol{Q}_w(t) \in \boldsymbol{R}^{n\times n}$ and $\boldsymbol{R}_\delta(t) \in \boldsymbol{R}^{m\times m}$, respectively. Assuming that $\boldsymbol{F}(t)$, $\boldsymbol{H}(t)$, $\boldsymbol{Q}_w(t)$, and $\boldsymbol{R}_\delta(t)$ are known, the standard KF update for estimating the state $\boldsymbol{x}(t)$ is given by

$$\hat{\boldsymbol{x}}(t/t-1) = \boldsymbol{F}(t)\hat{\boldsymbol{x}}(t-1/t-1) \tag{25}$$

$$\boldsymbol{P}(t/t-1) = \boldsymbol{F}(t)\boldsymbol{P}(t-1/t-1)\boldsymbol{F}^T(t) + \boldsymbol{Q}_w(t) \tag{26}$$

$$\boldsymbol{e}(t) = z(t) - \boldsymbol{H}(t)\hat{\boldsymbol{x}}(t/t-1) \tag{27}$$

$$\boldsymbol{K}(t) = \boldsymbol{P}(t/t-1)\boldsymbol{H}^T(t)\cdot[\boldsymbol{H}(t)\boldsymbol{P}(t/t-1)\boldsymbol{H}^T(t)+\boldsymbol{R}_\delta(t)]^{-1} \tag{28}$$

$$\hat{\boldsymbol{x}}(t/t) = \hat{\boldsymbol{x}}(t/t-1) + \boldsymbol{K}(t)\boldsymbol{e}(t) \tag{29}$$

$$\boldsymbol{P}(t/t) = [\boldsymbol{I} - \boldsymbol{K}(t)\boldsymbol{H}(t)]\boldsymbol{P}(t/t-1) \tag{30}$$

where $\hat{\boldsymbol{x}}(t/\tau)$ $(\tau = t-1, t)$ denotes the estimate of $\boldsymbol{x}(t)$ given the measurements $\{z(j), j \le \tau\}$ and $\boldsymbol{e}(t)$ represents the prediction error.

Here, we associate $z(i)$ with the ith observation of state space model and the true state $\boldsymbol{x}$ as the true location of the 3-D point and assume that the additive noise is zero-mean and Gaussian distributed. Since the true 3-D location across multiple camera views does not change, the state transition and observation matrices should be $\boldsymbol{F}(t) = \boldsymbol{I}_3$ and $\boldsymbol{H}(t) = \boldsymbol{I}_3$, respectively. Thus, (23) and (24) can be rewritten as follows:

$$\boldsymbol{x}(i) = \boldsymbol{x}(i-1) + \boldsymbol{w}(i) \tag{31}$$

$$z(i) = \boldsymbol{x}(i) + \delta(i) \tag{32}$$

where $\boldsymbol{w}(i)$ and $\boldsymbol{\delta}(i)$ are Gaussian distributed innovation and measurement noise with zero mean and variance $\boldsymbol{Q}_w = q\boldsymbol{I}_3$ and $\boldsymbol{R}_\delta = r\boldsymbol{I}_3$, respectively, and $\boldsymbol{I}_3$ is the 3×3 identity matrix. The KF updates are then reduced to

$$\boldsymbol{P}(i/i-1) = \boldsymbol{P}(i-1/i-1) + q\boldsymbol{I}_3 \tag{33}$$

$$\boldsymbol{K}(i) = \boldsymbol{P}(i/i-1)[\boldsymbol{P}(i/i-1) + r\boldsymbol{I}_3]^{-1} \tag{34}$$

$$\hat{\boldsymbol{x}}(i/i) = \hat{\boldsymbol{x}}(i-1) + \boldsymbol{K}(i)\boldsymbol{e}(i) \tag{35}$$

$$\boldsymbol{P}(i/i) = [\boldsymbol{I}_3 - \boldsymbol{K}(i)]\boldsymbol{P}(i/i-1) \tag{36}$$

where $i = 2, \cdots, M-1$. The initial state and covariance are initialized to $\boldsymbol{x}(1) = z(1)$ and $\boldsymbol{P}(1/1) = 10$, respectively, while $q = r = 0.1$ denotes the expected variance of the estimation error. As mentioned earlier, outliers may arise due to error in matching and estimated camera parameters. We now propose a method to detect possible outliers at each KF iteration based on the following three consistency criteria. If they are violated, $z(i)$ is considered as an outlier and the KF will be terminated.

1) Segmentation consistency: at the ith iteration, $z(i)$ is reprojected back to a 2-D point $\boldsymbol{x}_i = \boldsymbol{P}_i z(i)$ in view i, where $\boldsymbol{P}_i$ is the camera projection matrix of view i, which contains the intrinsic parameters and extrinsic parameters. For notational convenience, we have dropped the additional subscript t for denoting the tth time instant. Due to errors in computing the projection matrixes and triangulation, $\boldsymbol{x}_i$ may lie outside the segment it belongs to. In this case, $z(i)$ is considered as an outlier.
2) Location consistency: the 3-D distance between $z(i)$ and the predicted location of the KF $\hat{\boldsymbol{x}}(i)$ should be relatively small. That is, $||z(i) - \hat{\boldsymbol{x}}(i)|| \le \varepsilon_D$ for some constant ε_D. If not, it is treated as an outlier.
3) Intensity consistency: $z(i)$ is reprojected back to two 2-D points $\boldsymbol{x}_i = \boldsymbol{P}_i z(i)$ and $\boldsymbol{x}_{i+1} = \boldsymbol{P}_{i+1} z(i)$ in views i and $i+1$, respectively. They should have similar intensity values, i.e., $|\boldsymbol{I}_i(\boldsymbol{x}_i) - \boldsymbol{I}_{i+1}(\boldsymbol{x}_{i+1})| \le \varepsilon_I$ for some constant ε_I. However, to cope with intensity variation, we employ the normalized cross-correlation (NCC) as a measure for intensity consistency check. The NCC has a range of $[-1, 1]$ and in this paper, $z(i)$ is treated as an outlier if its NCC score is smaller than 0.8.

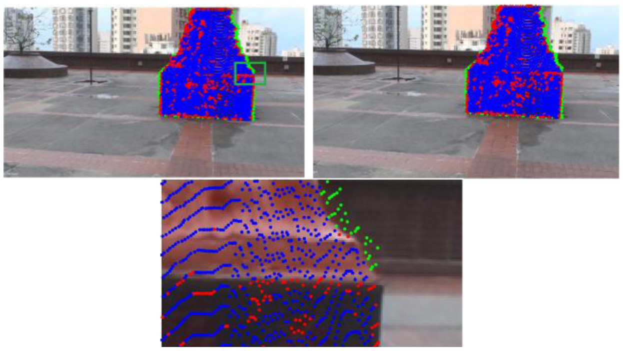


Fig. 11. Upper figures show the 3-D to 2-D reprojection at frame 20 and frame 21, respectively. Blue points are inliers. Green points are outliers detected by the segmentation consistency check. Red points are the outliers detected by intensity and location consistency checks. Lower part shows the enlargement of the highlight area in upper left. The point cloud is downsampled for better visualization.

To ensure the reliability of the extracted 3-D points, we only include those points that satisfy the consistency tests above for K consecutive number of views. K is chosen to be four in this paper since we have at most seven matches for the eight views. The KF is first applied to the first view. When it is terminated, say at the ith iteration, a set of potential 3-D points $\boldsymbol{S}_F = \{z(1), z(2), \cdots, z(i)\}$ is obtained. If i is less than K, we then proceed to the second view and so on until a consecutive of K matched views is found. If so, the matched 3-D points can be fused by computing their mean value. If not, then we shall proceed to the remaining corresponding points. The process is illustrated in Fig. 11 where the projections of the point cloud are also shown. Blue points denote the inliers. Green points show the points detected by segmentation consistency in (i). Red points show the points detected by location and intensity consistency checks in (ii) and (iii). Fig. 10(b) shows the refined point cloud after the outlier detection and Kalman filtering, where we can see that outlying points are effectively suppressed. The advantages of the KF-approach are its implementation simplicity, and flexibility, where one can process the views sequentially while performing the consistency checks.

2) *Iterative Refinement of Point Cloud:* With more reliable matched points, the camera parameters and hence the image correspondences can be further improved. This suggests an iteration method for further refining the point cloud and other parameters.

More precisely, after Kalman filtering, the 2-D matching and 3-D geometry are refined as follows.

1) The fused 3-D point cloud is first reprojected to successive views to serve as prior features/correspondences for MI-based matching. By adding to (14) the reprojection correspondences as parts of the feature term, a more reliable depth map can be computed.
2) The updated matching result is then used to update the 3-D point cloud using the KF-based outlier detection and point fusion algorithm introduced above.
3) The process will be repeated until the maximum number of iterations, L_{MAX}, is reached or no significant improvement of the 3-D geometry can be obtained. To measure the change in the 3-D geometry, a similarity measure of two consecutive 3-D point clouds is therefore needed.

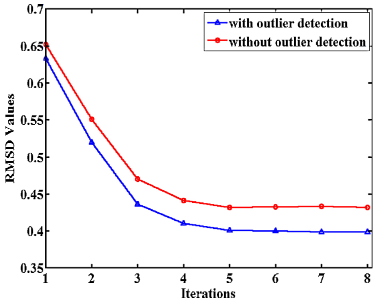


Fig. 12. Convergence behavior of the RMSD versus the number of iterations for the proposed iterative 3-D reconstruction algorithm. The blue line shows the RMSD values with the KF-based outlier detection. The red line shows the RMSD values without KF-based outlier detection.

Let the 3-D point clouds at the l and $l+1$ iterations be $\boldsymbol{M}^{(l)} = \{\boldsymbol{p}_j^{(l)}, j = 1, \ldots, \vartheta\}$ and $\boldsymbol{M}^{(l+1)} = \{\boldsymbol{p}_j^{(l+1)}, j = 1, \ldots, \vartheta\}$, respectively, where ϑ is the number of estimated 3-D points. In this paper, the similarity measure is chosen as the root-mean-square distance (RMSD) between two point clouds, which is defined as follows:

$$RMSD = \sqrt{\frac{1}{\vartheta}\sum_{j=1}^{\vartheta} D(\boldsymbol{p}'_j, \boldsymbol{p}_j)} \tag{37}$$

where $p'_j \in \boldsymbol{M}^{(l+1)}$ is the point closest to $p_j \in \boldsymbol{M}^{(l)}$ and $\boldsymbol{D}(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

The algorithm can be terminated when the minimum $RMSD$ value or the maximum number of iterations, L_{MAX}, is reached. Fig. 12 plots $RMSD$ versus the number of iterations for refining the point cloud for the statue in Fig. 11. Fig. 10(c) shows the final point cloud obtained by refining the one at Fig. 10(b). Considerable improvement in terms of the smoothness and number of matched points is observed, which demonstrates the effectiveness of the proposal iterative refinement approach.

C. RBF Modeling and Mesh Generation

After the completion of the iterative refinement procedure, the final point cloud $\tilde{\boldsymbol{M}}$ may still contain holes and may not be smooth enough to get a good mesh. Therefore, further smoothing of the raw 3-D point cloud is necessary. In this paper, we employ the RBF-based modeling for smoothing and the construction of the 3-D mesh. The basic form of an RBF is as follows:

$$F(\boldsymbol{x}) = \Sigma_{j=1}^{\theta} c_j p_j(\boldsymbol{x}) + \Sigma_{i=1}^{\vartheta} \lambda_i \varphi(\boldsymbol{x} - \boldsymbol{x}_i) \tag{38}$$

where c_j is the model coefficient of the polynomial $p_j(x)$, $j = 1, \cdots, \theta$, which together form a basis of the polynomial part of the RBF; and λ_i is the RBF coefficient for the RBF $\varphi(\boldsymbol{x} - \boldsymbol{x}_i)$ with center $\boldsymbol{x}_i$, $i = 1, \cdots, \vartheta$, where ϑ is the number of data points, which is also equal to the RBFs used in the RBF. Given a set of 3-D points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_\vartheta\}$ with values $\boldsymbol{f} = [F(\boldsymbol{x}_1), F(\boldsymbol{x}_2), \cdots, F(\boldsymbol{x}_\vartheta)]^T$ and the additional conditions [29], $\boldsymbol{p}^T \Lambda = 0$, where $\boldsymbol{p}$ is the vector containing $\{p_j(\boldsymbol{x})\}$ and

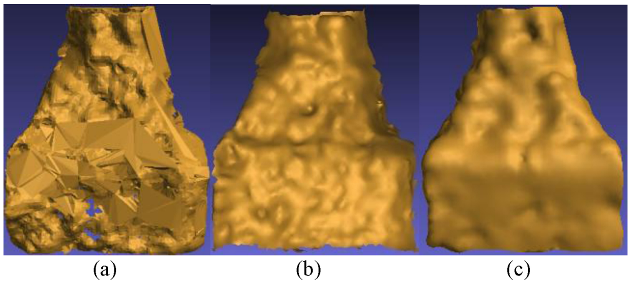


Fig. 13. 3-D reconstruction results: (a) without using RBF, (b) using RBF without outlier detection, and (c) using RBF with outlier removal.

Λ is the vector containing the RBF coefficients $\{\lambda_i\}$; the RBF coefficients satisfy the following equation:

$$\begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{p} \\ \boldsymbol{p}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\Lambda} \\ \boldsymbol{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} \tag{39}$$

where $[\boldsymbol{\Phi}]_{ji} = \varphi(\boldsymbol{x}_j - \boldsymbol{x}_i)$, $i, j = 1, \cdots, \vartheta$, and $\boldsymbol{c} = [c_1, c_2, \cdots, c_\theta]^T$. By solving the linear equation in (39), the RBF coefficients can be computed. The complexity of solving (38) is $O(\vartheta^3)$. The fast evaluation method proposed in [30] can reduce the complexity to $O(\vartheta \log \vartheta)$. The basic idea of this fast RBF algorithm is that exact interpolation is not needed in practice. Consequently, the value of $F(\boldsymbol{x}_i)$ is only required to lie in an acceptable range to achieve a given accuracy. In this paper, we also make use of this property to get rid of possible outliers left. More precisely, we set an error bar for the RBF values $\varepsilon_{i1} < F(\boldsymbol{x}_i) < \varepsilon_{i2}$, where, for simplicity, we set $\varepsilon_{i1} = -\varepsilon_{i2} = \varepsilon_i$. Consequently, the problem becomes

$$\begin{array}{ll} \text{minimize} & \Lambda^T \boldsymbol{\Phi} \Lambda \\ \text{subject to} & |F(\boldsymbol{x}_i)| < \varepsilon_i \quad \boldsymbol{p}^T \Lambda = 0 \end{array} \tag{40}$$

which is recognized as a convex constrained quadratic programming problem and it can be solved readily [29]. In this paper, ε_i is chosen as the normalized confidence value obtained from the matching results. The higher the confidence is, the closer the reconstructed points are to the original points. By comparing the reconstructed 3-D model [Fig. 13(c)] with the one without removing the outliers and smoothing [Fig. 13(a)] and the one using RBF without outlier removal [Fig. 13(b)], significant improvement is obtained. Finally, we summarize the complete system flow in Algorithm 1.

D. Experimental Results

We now present and evaluate further the experimental and timing results of the proposed algorithm. The testing is performed in an INTEL Core i7 920 CPU-based computer with 4 GB RAM and GTX295 GPU acceleration. The resolution and the frame rate of the videos are 1920 × 1080i and 25 f/s, respectively. Example video stabilization results have been presented in Fig. 5 and a demonstration video of our video stabilization algorithm can be found at http://www.youtube.com/watch?v=qPuMNjgUoWs.

Algorithm 1 System flow of the proposed algorithm

```
1) Video Stabilization
1.1. Compute SIFT features of each video sequence.
1.2. Use the feature points to determine the affine transformation
1.3. Use LPR for smoothing the parameters of the local transformations.
1.4. Stabilize the video and use motion inpainting to fill in missing areas.
2) Segmentation-MI-based Depth Estimation
2.1. Perform semiautomatic segmentation using Lazy snapping or automatic segmentation using graph cut on reference image frames.
2.2. Use the level set method to track each segment.
2.3. Apply the segmentation-MI-based depth estimation algorithm to each segment for adjacent views and detect possible occlusions.
2.4. Integrate the depth maps of the segments and perform SK-LPR-RICI smoothing and inpaint the occlusion areas.
3) IBR
3.1. Use the depth maps, mattes, and original images to render intermediate views using the approach in [11].
4) 3-D reconstruction (optional)
4.1. Compute the projection matrices using the S-SFM method and determine the initial 3-D point cloud.
4.2. Iteration refinement.
    i) Perform KF-based filtering and fusion of the point cloud.
    ii) Reproject the 3-D points to 2-D images to refine the estimation of the depth maps and projection matrices.
    iii) Go to step 2.3 if the minimum RMSD or maximum number of iteration is not reached.
4.3. Perform robust RBF-based smoothing to the point cloud and generate the mesh from the RBF model.
4.4. Render the 3-D model using shadow field to support real-time relighting and object movement.
```

The segmentation-MI-based matching algorithm has been evaluated extensively on the stereo test image sets at the Middlebury stereo page and our outdoor plenoptic video *podium*. Fig. 14 shows the stereo images, ground truth depth maps, and depth maps calculated by our method of the Teddy test images (450 × 375) [48]. Table I is a reproduction of the upper part of the evaluation at the Middlebury stereo pages. A standard threshold of 1 pixel has been used in Table I. The segmentation-MI-based matching is among the best performing stereo algorithms at the upper part of the table with the semiautomatic and automatic versions ranking the fourth and sixth, respectively. The performance difference between our algorithm and the top algorithm is very small. Moreover, our algorithm is very stable and insensitive to versatile data sets such as real data sets. And there are not too many parameters that need to be selected carefully in our algorithm. Example renderings at different views of the *podium* plenotic video have been shown in Figs. 8 and 9.

To illustrate the 3-D reconstruction of the stationary object, the M-IBR system was driven around a statue in the *podium* sequence. The 3-D model of the statue is estimated by the procedures described in step 4 of Algorithm 1. Fig. 15 gives example renderings using the reconstructed 3-D model

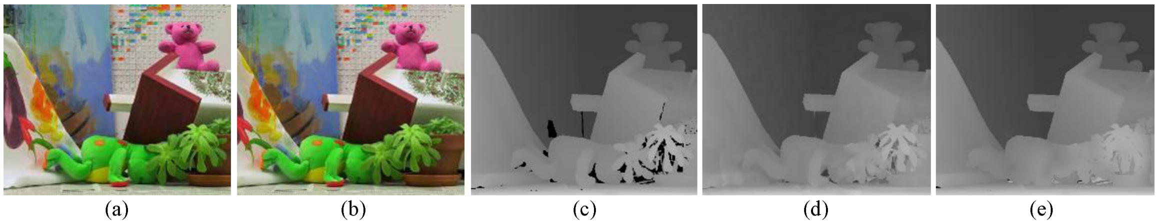


Fig. 14. Teddy test images [48] and depth maps for comparison. (a) LEFT image. (b) RIGHT image. (c) Ground truth depth map. (d) Depth map calculated by semiautomatic segmentation-based MI matching. (e) Depth map calculated by automatic segmentation-based MI matching.

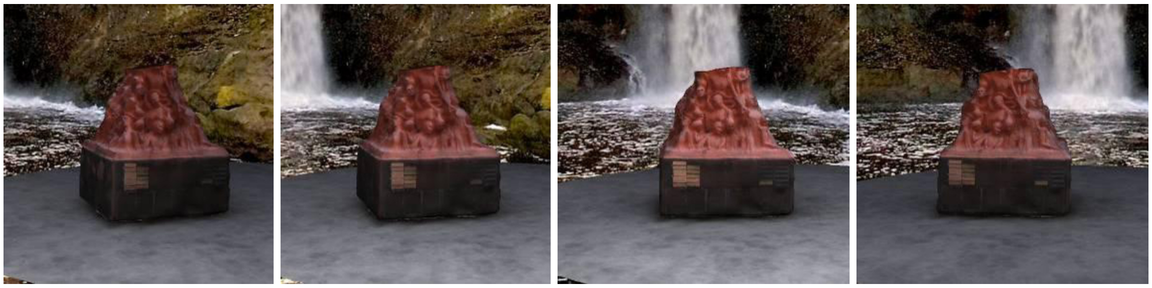


Fig. 15. Object-based rendering results using the estimated 3-D model and shadow field in different lightening conditions.

TABLE I
COMPARISON OF THE RANK USING STANDARD THRESHOLD OF 1 PIXEL ON MIDDLEBURY TEST STEREO IMAGES

Algorithm	Rank	Tsukuba	Venus	Teddy	Cones
Adapting BP [42]	6.7	1.37	0.21	7.06	7.92
CoopRegion [43]	6.7	1.16	0.21	8.31	7.18
DoubleBP [46]	8.8	1.29	0.45	8.30	8.78
Our Method (Semi-Seg)	9.6	1.30	0.18	5.10	8.88
OutlierConf [49]	9.8	1.43	0.26	9.12	8.57
Our Method (Auto-Seg)	12.2	1.30	0.24	7.91	8.88
SubPixDoubleBP [50]	13.2	1.76	0.46	8.38	8.73
SurfaceStereo [44]	13.8	1.65	0.28	5.10	7.95

and shadow field using OpenGL, which supports real-time relighting and object movement with soft shadow. The 3-D rendering speed is 60 f/s and the processing time of the whole reconstruction process is about 10 min.

More results of another sequence “conference” where a person is conducting a conference presentation can be found in [56]. The M-IBR system is used to track the motion of the speaker and its partial dynamic geometry is recovered by integrating the depth maps computed using the eight cameras at each time instant. Relighting and rendering using this dynamic partial geometry is also illustrated in [56]. More rendering results can be found in our demonstration video at http://www.youtube.com/watch?v=hZHW5XS9xAg. Moreover, the SK-LPR-RICI method can be applied to the depth map to estimate a smooth gradient field. Combining this gradient field with the depth map, a normal field corresponding to the 2-D image can be approximated. This can be used to perform real-time 2-D relighting. A demonstration video of the 2-D relighting results can be found at http://www.youtube.com/watch?v=5LRdPgnWapo. Due to page limitation, the details of the relighting algorithm will be reported in future work.

VI. CONCLUSION

A new system and associate processing algorithms for object-based rendering and 3-D reconstruction using a movable IBR system for improved viewing freedom and environmental modeling were presented. They included three major components, namely: 1) an improved video stabilization method based on LPR; 2) a new iterative segmentation and MI-based algorithm for dense depth map estimation, which supports both semiautomatic and automatic segmentation methods; and 3) a new 3-D reconstruction algorithm using the S-SFM technique and the dense depth maps estimated, which makes use of a new iterative point cloud refinement algorithm based on KF for outlier removal and the segmentation-MI-based algorithm to further refine the correspondences and the projection matrices. A new robust RBF-based modeling algorithm was developed to further suppress possible outliers and generate a smooth 3-D mesh of the object. The mobility of our system also allowed us to capture 3-D models of static objects more conveniently. Experimental results showed that high-quality renderings can be obtained by using the shadow light field and the 3-D model reconstructed. Further research will focus on improving the mechanic design of the camera array so that they can be steered to different directions, while the platform or wheel chair is moving.

ACKNOWLEDGMENT

The authors would like to thank J. L. C. Koo for his help in constructing the movable image-based rendering system.

REFERENCES

[1] S. E. Chen, “QuickTime VR: An image-based approach to virtual environment navigation,” in *Proc. Annu. Comput. Graph.*, Aug. 1995, pp. 29–38.

[2] P. E. Debevec, C. J. Taylor, and J. Malik, “Modeling and rendering architecture from photographs: A hybrid geometry—and image-based approach,” in *Proc. Annu. Conf. Comput. Graph*, Aug. 1996, pp. 11–20.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 1996, pp. 43–54.

[4] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 1996, pp. 31–42.

[5] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. Annu. Conf. Comput. Graph*, Aug. 1995, pp. 39–46.

[6] J. Shade, S. Gortler, L. W. He, and R. Szeliski, "Layered depth images," in *Proc. Annu. Conf. Comput. Graph.*, Jul. 1998, pp. 231–242.

[7] H. Y. Shum and L. W. He, "Rendering with concentric mosaics," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 1999, pp. 299–306.

[8] K. Zhou, Y. Hu, S. Lin, B. Guo, and H. Y. Shum, "Precomputed shadow fields for dynamic scenes," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 2005, pp. 1196–1201.

[9] H. Y. Shum, S. C. Chan, and S. B. Kang, *Image-Based Rendering*. New York: Springer-Verlag, 2007.

[10] Z. F. Gan, S. C. Chan, K. T. Ng, and H. Y. Shum, "An object-based approach to plenoptic videos," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 3435–3438.

[11] S. C. Chan, Z. F. Gan, K. T. Ng, and H. Y. Shum, "An object-based approach to image/video-based synthesis and processing for 3-D and multiview televisions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 821–831, Jun. 2009.

[12] S. C. Chan, H. Y. Shum, and K. T. Ng, "Image-based rendering and synthesis: Technological advances and challenges," *IEEE Signal Process. Mag.*, vol. 24, no. 7, pp. 22–33, Nov. 2007.

[13] J. Konrad and M. Halle, "3-D displays and signal processing," *IEEE Signal Process. Mag.*, vol. 24, no. 7, pp. 97–111, Nov. 2007.

[14] *PG Drive Technology* [Online]. Available: http://www.pgdt.com/products/vr2/index.html

[15] *USB-I2C/IO* [Online]. Available: http://www.devasys.com/usbi2cio.htm

[16] Z. F. Gan, S. C. Chan, K. T. Ng, and H. Y. Shum, "Object tracking for a class of dynamic image-based representations," in *Proc. SPIE Visual Commun. Image Process.*, Jul. 2005. pp. 1267–1274.

[17] Z. F. Gan, S. C. Chan, and H. Y. Shum, "Object tracking and matting for a class of dynamic image-based representations," in *Proc. IEEE Adv. Video Signal-Based Surveillance*, Sep. 2005, pp. 81–86.

[18] *IVS Technology* [Online]. Available: http://www.ivs-tech.com

[19] Z. Y. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

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

[21] S. C. Chan, Z. Y. Zhu, K. T. Ng, C. Wang, S. Zhang, Z. G. Zhang, Z. F. Ye, and H. Y. Shum, "A movable image-based system and its applications to multiview audio-visual conferencing," in *Proc. IEEE Int. Symp. Commun. Info. Technol.*, Oct. 2010, pp. 1142–1145.

[22] T. Li, J. Sun, C. K. Tang, and H. Y. Shum, "Lazy snapping," in *Proc. Annu. Conf. Comput. Graph.*, Aug. 2004, pp. 303–308.

[23] K. Ratakonda, "Real-time digital video stabilization for multi-media applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4. May 1998, pp. 69–72.

[24] H. C. Chang, S. H. Lai, and K. O. Lu, "A robust and efficient video stabilization algorithm," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 1. Jun. 2004, pp. 29–32.

[25] Y. Matsushita, E. Ofek, and H. Y. Shum, "Full-frame video stabilization," in *Proc. IEEE CVPR*, vol. 1. Jun. 2005, pp. 50–57.

[26] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, Jun. 2006.

[27] X. L. Huang, N. Paragios, and D. N. Metaxas, "Shape registration in implicit spaces using information theory and free form deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1303–1318, Aug. 2006.

[28] J. Nocedal and S. J. Wright, *Numerical Optimization*. Berlin, Germany: Springer, 1999.

[29] R. K. Beatson, J. B. Cherrie, T. J. McLennan, T. J. Mitchell, J. C. Carr, W. R. Fright, and B. C. McCallum, "Surface reconstruction via smoothest restricted range approximation," in *Geometric Modeling and Computing*. Los Angeles, CA: Nashboro Press, 2004, pp. 41–52.

[30] R. K. Beatson, W. A. Light, and S. Billings, "Fast solution of the radial basis function interpolation equations: Domain decomposition methods," *SIAM J. Sci. Comput.*, vol. 22, pp. 1717–1740, Feb. 2001.

[31] Q. Wu, K. T. Ng, S. C. Chan, and H. Y. Shum, "On object-based compression for a class of dynamic image-based representations," in *Proc. Int. Conf. Image Process.*, Sep. 2005, pp. 405–408.

[32] K. T. Ng, Q. Wu, S. C. Chan, and H. Y. Shum, "Object-based coding for plenoptic videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 4, pp. 548–562, Apr. 2010.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 2, no. 60, pp. 91–110, 2004.

[34] R. Hu, R. J. Shi, I. F. Shen, and W. B. Chen, "Video stabilization using scale-invariant features," in *Proc. Int. Conf. Info. Visualization*, Jul. 2007, pp. 871–876.

[35] Z. G. Zhang, S. C. Chan, K. L. Ho, and K. C. Ho, "On bandwidth selection in local polynomial regression analysis and its application to multi-resolution analysis of non-uniform data," *J. Signal Process. Syst. Signal Image Video Technol.*, vol. 52, no. 3, pp. 263–280, 2008.

[36] Z. G. Zhang, S. C. Chan, and Z. Y. Zhu, "A new two-stage method for restoration of images corrupted by Gaussian and impulse noises using local polynomial regression and edge preserving regularization," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 948–951.

[37] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[38] G. Egnal and R. P. Wildes, "Detecting binocular half-occlusions: Empirical comparisons of five approaches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1127–1133, Aug. 2002.

[39] L. Zhang and S. Seitz, "Parameter estimation for MRF stereo," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 2. Aug. 2005, pp. 288–295.

[40] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[41] J. Sun, Y. Li, S. B. Kang, and H. Y. Shum, "Symmetric stereo matching for occlusion handling," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 2. Aug. 2005, pp. 399–406.

[42] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. IEEE Int. Conf. Pattern Recognit.*, vol. 3. Sep. 2006, pp. 15–18.

[43] Z. Wang and Z. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 1, no. 12. Aug. 2008, pp. 1–8.

[44] M. Bleyer, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Aug. 2010, pp. 1570–1577.

[45] Y. Taguchi, B. Wilburn, and L. Zitnick, "Stereo reconstruction with mixed pixels using adaptive over-segmentation," in *Proc. IEEE CVPR*, vol. 1. Aug. 2008, pp. 2720–2727.

[46] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 492–504, Mar. 2009.

[47] V. Kolmogorov and R. Zabih, "Computation visual correspondence with occlusions using graph cuts," in *Proc. Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 508–515.

[48] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1/2/3, pp. 7–42, Apr.–Jun. 2002.

[49] L. Xu and J. Jia, "Stereo matching: An outlier confidence approach," in *Proc. ECCV*, vol. 5305. Oct. 2008, pp. 775–787.

[50] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[51] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman and Hall, 1986.

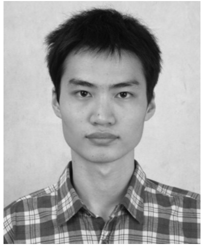
[52] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner–Ziv-based multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 713–724, Jun. 2008.

[53] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, vol. 28, no. 3. 2009, pp. 1–9.

[54] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, vol. 30, no. 1. 2011, pp. 1–10.

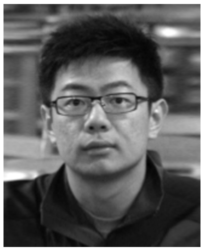
[55] B. M. Smith, L. Zhang, H. Jin, and A. Agarwala, "Light field video stabilization," in *Proc. Int. Conf. Comput. Vis.*, vol. 2. Sep.–Oct. 2009, pp. 341–348.

[56] *More Rendering Results* [Online]. Available: https://picasaweb.google.com/115086134578854328561/MIBR?authuser=0&feat=directlink



Zhen-Yu Zhu received the B.Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2008. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong.

His current research interests include digital image processing, multiple-view geometry, and 3-D reconstruction.



Shuai Zhang received the B.Eng. degree from Yanshan University, Qinhuangdao, China, in 2009, and the M.Eng. degree from the University of Hong Kong, Pokfulam, Hong Kong, in 2010.

He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, University of Hong Kong. His current research interests include multimodality data fusion, human body tracking, and statistical video processing.

Shing-Chow Chan (S'87–M'92) received the B.Sc. and Ph.D. degrees in electrical engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 1986 and 1992, respectively.

He joined the City Polytechnic University of Hong Kong, Kowloon, Hong Kong, in 1990 as an Assistant Lecturer and later became a University Lecturer. Since 1994, he has been with the Department of Electrical and Electronic Engineering, University of Hong Kong, and is currently a Professor. He has held visiting positions with Microsoft Corporation, Redmond, WA, Microsoft Research Asia, Beijing, China, University of Texas, Arlington, and Nanyang Technological University, Singapore. His current research interests include fast transform algorithms, filter design and realization, multirates and array signal processing, communications and biomedical signal processing, and image-based rendering.

Dr. Chan is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society and is an Associate Editor of the *Journal of Signal Processing Systems*. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I from 2008 to 2009 and was the Chairman of the IEEE Hong Kong Chapter of Signal Processing from 2000 to 2002. He was on the organizing committees of several international conferences as the Special Session Co-Chair of the IEEE International Conference on Acoustic Speech, Signal Processing in 2003, and the Technical Program Co-Chair of the IEEE International Conference on Field-Programmable Technology in 2002, and on the technical program committees of IEEE APCCAS in 2006, IEEE APCCAS in 2008, IEEE BioCAS in 2006, and was the Tutorial Chair of the IEEE International Conference on Image Processing in 2010.

Heung-Yeung Shum (M'90–SM'01–F'06) received the Ph.D. degree in robotics from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Currently, he is the Corporate Vice President with Microsoft Corporation, Redmond, WA, where he is responsible for search product development. Previously, he oversaw research activities with Microsoft Research Asia, Beijing, China, as well as the laboratory's collaborations with universities in the Asia Pacific region. He was responsible for the Internet Services Research Center, an applied research organization dedicated to long-term and short-term technology investments in search and advertising at Microsoft, and was a Researcher with Microsoft Research, Redmond, in 1996. He moved to Beijing, as one of the founding members of Microsoft Research, China (later renamed Microsoft Research Asia). There, he began a nine-year tenure as a Research Manager, subsequently moving on to become the Assistant Managing Director, Managing Director of Microsoft Research Asia, Distinguished Engineer, and Corporate Vice President. He has published more than 100 papers in computer vision, computer graphics, pattern recognition, statistical learning, and robotics. He holds more than 50 U.S. patents.

Dr. Shum is a fellow of the Association for Computing Machinery.