Signal Processing: *Image Communication* 24 (2009) 17–30

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

3DTV view generation using uncalibrated pure rotating and zooming cameras

Songkran Jarusirisawad *, Hideo Saito

Department of Information and Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

ARTICLE INFO

Article history:
Received 10 October 2008
Accepted 19 October 2008

Keywords:
Free viewpoint video
View interpolation
Projective grid space
Trifocal tensor

ABSTRACT

This paper proposes a novel method for synthesizing free viewpoint video captured by uncalibrated pure rotating and zooming cameras. Neither intrinsic nor extrinsic parameters of our cameras are known. Projective grid space (PGS), which is the 3D space defined by the epipolar geometry of two basis cameras, is employed for weak camera calibration. Trifocal tensors are used to relate non-basis cameras to PGS. Given trifocal tensors in the initial frame, our method automatically computes trifocal tensors in the other frames. Scale invariant feature transform (SIFT) is used for finding corresponding points in a natural scene between the initial frame and the other frames. Finally, free viewpoint video is synthesized based on the reconstructed visual hull. In the experimental results, free viewpoint video captured by uncalibrated hand-held cameras is successfully synthesized using the proposed method.

1. Introduction

In most of the free viewpoint video creation from multiple camera systems, cameras are assumed to be fixed. This is guaranteed by mounting the cameras on poles or tripods for the duration of the capturing, and calibration is done only before starting video acquisition. During video acquisition, cameras cannot be moved, zoomed or rotated. Field of view (FOV) of each camera in these systems must be wide enough to cover the area in which the object moves. If this area is large, the moving object's resolution in the captured video and in the free viewpoint video will decrease.

Allowing cameras to be zoomed and rotated during capture is more flexible in terms of video acquisition. However, in this case, all cameras must be dynamically calibrated at every frame. Doing strong calibration at every frame with multiple cameras is possible by using special markers [14]. Marker size must be large enough compared to the scene size to make calibration accurate. When the capturing space is large, it is unfeasible to use a huge artificial marker.

In this paper, we propose a novel method for synthesizing free viewpoint video in a natural scene from uncalibrated pure rotating and zooming cameras. Our method does not require special markers or information about intrinsic camera parameters. For obtaining geometrical relation among the cameras, projective grid space (PGS) [28], which is 3D space defined by epipolar geometry between two basis cameras, is used. All other cameras are weakly calibrated to the PGS via trifocal tensors. We approximate background scene as several planes. Preprocessing tasks including the selection of 2D–2D correspondences among views and the segmentation of the background are manually done only once at the initial frames (see Fig. 3). For the other frames, the homographies that relate these frames to the initial frame are automatically estimated. Trifocal tensors of the other frames are then recomputed using these homographies. Scale invariant feature transform (SIFT) [18] is used for finding corresponding points between the initial frame

* Corresponding author. Tel.: +81 8065703942.
E-mail addresses: kpriony@hotmail.com, songkran@ozawa.ics.keio.ac.jp (S. Jarusirisawad), saito@ozawa.ics.keio.ac.jp (H. Saito).

0923-5965/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2008.10.003

and the other frame for homography estimation. We recover the shape of the moving object in PGS by silhouette volume intersection method [15]. The recovered shape in PGS provides dense correspondences among the multiple cameras, which are used for synthesizing free viewpoint images by view interpolation [3].

1.1. Related works

One of the earliest researches of free viewpoint image synthesis of a dynamic scene is virtualized reality [13]. In this research, 51 cameras are placed around hemispherical dome called a 3D Room. The 3D structure of a moving human is extracted using multi-baseline stereo (MBS) [24]. Then, free viewpoint video is synthesized from the recovered 3D model.

Moezzi et al. synthesize free viewpoint video by recovering visual hull of the objects from silhouette images using 17 cameras [21]. Their approach creates true 3D models with small polygons. Each polygon is separately colored thus requiring no texture-rendering support. Their 3D model can use standard 3D model format, such as virtual reality modeling language (VRML), delivered though the Internet and viewed with VRML browsers. In terms of computation time, real-time systems for synthesizing free viewpoint video have also been developed recently [7,8,23].

Many methods for improving quality of free viewpoint image have been proposed. Carranza et al. recover human motion by fitting a human shape model to multiple view silhouette input images for accurate shape recovery of the human body [2]. Starck and Hilton optimize a surface mesh using stereo and silhouette data to generate high accuracy virtual view images [30]. Saito et al. propose an appearance-based method [27], which combines the advantages of image-based rendering and model-based rendering. Zhang and Chen [33] propose a self-reconfigurable cameras array system that captures video sequences from an array of mobile cameras and renders novel views on the fly and reconfigures the camera positions to get a better rendering quality.

In all of the systems mentioned above, calibrated cameras are used. Cameras in these systems are arranged to the fixed positions around a scene and calibrated before capturing video. During video acquisition, the cameras cannot be moved or zoomed (except using special hardware or markers [33]). FOV of all cameras must be wide enough to cover the whole area in which the object moves. If the object moves around a large area, the moving object's resolutions in the captured video will be insufficient to synthesize a good quality free viewpoint image.

The image-based visual hulls method presented by Matusik et al. [20] is a real-time free viewpoint video method from uncalibrated cameras (only fundamental matrices are estimated). This method reconstructs visual hull of the object using epipolar geometry in image space instead of 3D space, so it does not suffer from quantization artifacts of voxels like in ordinary visual hull. This method can create new views in real-time from four cameras. However, this method applies only to the case where the cameras are fixed.

Eisert et al. [5] propose an automatic method for Euclidean reconstruction from a sequence of input frames where camera poses are unknown but the cameras' intrinsic parameters are previously estimated. Their algorithm starts by finding an approximate model from two initial frames which is then used as an approximate model for cameras pose estimation. After the pose of all cameras are known, an accurate 3D model is then reconstructed using volumetric reconstruction.

Pollefeys et al. [25] and Rodriguez et al. [26] present systems that create 3D surface models from a sequence of images taken with an uncalibrated hand-held video camera. The projective structure and motion is recovered by matching corner features in the image sequence. The ambiguity on the reconstruction is automatically upgraded from the projective space to the metric space through self-calibration. Dense stereo matching is carried out between the successive frames. The input images are used as surface textures to produce photo-realistic 3D models.

Another technique for view synthesis from uncalibrated images is designed to create in-between images from dense correspondences among two or more reference images without reconstructing the 3D model [1,3,29]. In these methods, correspondences between images are assigned manually or from stereo matching algorithms.

View synthesis from uncalibrated cameras proposed in [11,12] are the combination of the image-based and model-based methods. A 3D model is reconstructed in PGS [28] instead of the Euclidean space for making dense correspondence among views, which provide information for image interpolation in the same way as [3].

In our previous work [11], we proposed that PGS can be used for synthesizing free viewpoint images from uncalibrated pure rotation and zoom cameras. We used fixed PGS, which is defined by two background images of the whole scene. In [12], we extended [11] by using dynamic PGS defined by the current frames. PGS defined on background images covers the whole scene, but PGS defined on the current frames covers only the current area of interest. Thus, this gives more accurate 3D reconstruction result using the same number of voxels.

In both the previous works, fundamental matrices are used for calibrating non-basis cameras to PGS. Using fundamental matrices results in point transfer problem when 3D points lie on the trifocal plane, as will be shown in Section 3. Hence, in our previous works, we have to arrange the cameras in a non-horizontal setting.

In this work, we extend the idea from [12] that uses PGS defined from the current input images instead of the initial frame. We show that using trifocal tensors give a more stable result and we can use any camera configuration to render free viewpoint video.

2. Overview

To reconstruct a 3D model without strong camera calibration, we utilize PGS [28], which is a weak calibration framework based on epipolar geometry. Fundamental

matrix and trifocal tensors for weakly calibrating cameras, can be estimated from 2D–2D correspondences.

Because our cameras are not static, the fundamental matrices and trifocal tensors must be estimated for all frames. One straightforward way for calibration is finding 2D–2D correspondences among cameras and compute the fundamental matrix and trifocal tensors at every frame. Corresponding points among views can be found by a keypoint detector and descriptor, such as the SIFT [18]. However, robustness of feature point matching of a 3D scene dramatically deceases as the viewpoint between the two images increases [22] because the images of a 3D scene from different views have different appearances due to motion parallax and perspective distortion.

Fig. 1. The camera setting in our experiment.

In pure rotating and zooming cameras, all frames from the same camera are related to each other by a homography matrix. If the fundamental matrix and trifocal tensors have already been estimated for one frame, we can compute the fundamental matrix and trifocal tensors of the other frames using the homography matrices relating these frames. This is described in Section 4. Finding correspondences using SIFT for estimating homography is easier and more robust because the capturing position of two images are the same. There is no motion parallax between these images so the two images are more similar. Accurate corresponding points can be found automatically using SIFT and the computational cost does not increase with the complexity of the 3D scene.

From this, we capture the whole background scene without the moving object at the initial frame of each camera. Then, two cameras are selected for defining PGS. The 2D–2D correspondences between cameras at the initial frame are selected manually (or automatically in case the number of correct correspondences is enough). The fundamental matrix and trifocal tensors of the initial

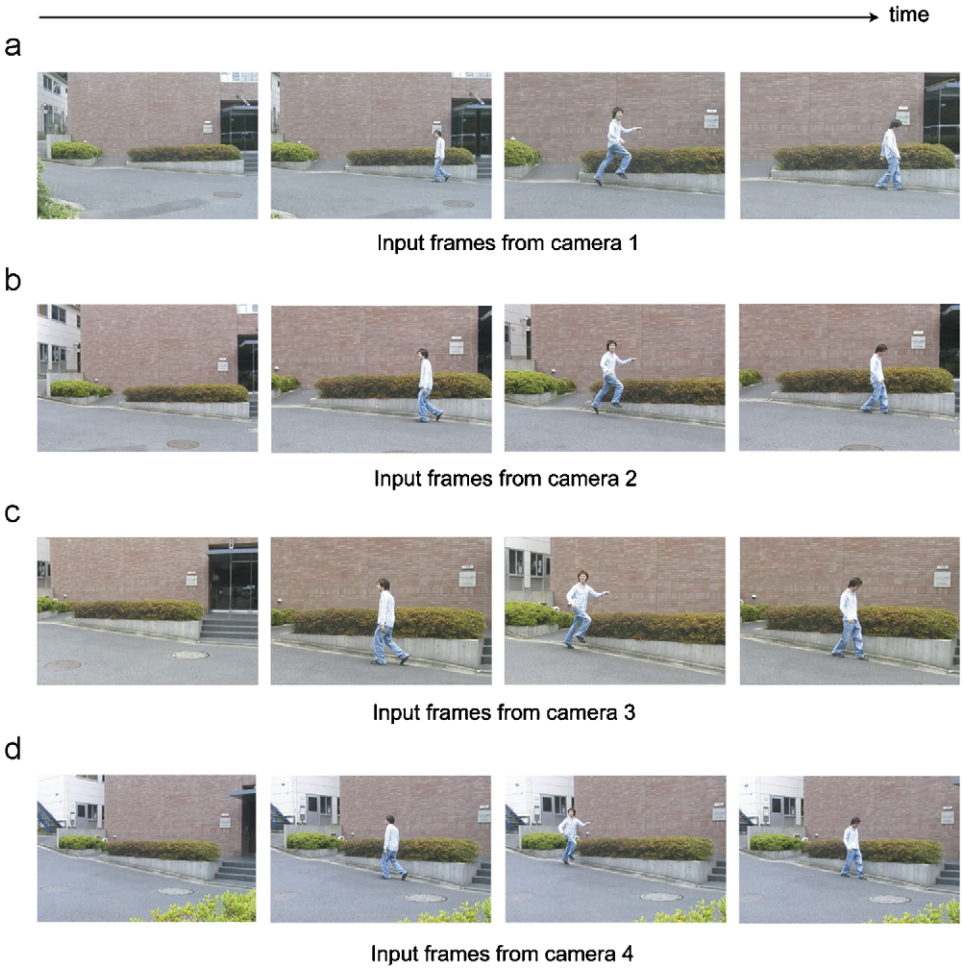

Fig. 2. Example input frames from four cameras.

frame are then estimated from these correspondences. To calibrate the other frames to PGS, homography matrices between that frame and the initial frame are estimated from 2D–2D correspondence automatically found using SIFT. Then, the fundamental matrix and trifocal tensors are re-estimated.

Fig. 1 shows the camera setting in our experiment. We use four DV cameras to capture the scene. Each camera is hand-held without tripod and each person does not change the position during capture. Because our calibration method is based on finding corresponding points with the initial frame, each camera is rotated and zoomed within the FOV of that frame.

Fig. 2 shows example input frames from each camera. We can see that each camera changes the view direction and focal length from frame to frame. The overall process is illustrated in Fig. 3 where the detail of each process is explained in the section written in the box. Our main contribution is the calibration part, which is described in Section 4. In the rest of the paper, we firstly describe PGS in Section 3. Then, we present the detailed algorithm of each step in Section 4–6. Finally, we show the experimental results and conclusion in Sections 7 and 8, respectively.

3. Projective grid space

This section describes the weak camera calibration framework for 3D reconstruction. PGS [28] allows us to define 3D space and to find the projection without knowing the cameras' intrinsic parameters or Euclidean coordinate information of a scene.

PGS is a 3D space defined by the image coordinates of two arbitrarily selected cameras, called basis camera 1 and basis camera 2. To distinguish this 3D space from the Euclidean one, we denote the coordinate system in PGS by P–Q–R axes. Fig. 4 shows the definition of PGS. x and y axes in the image of basis camera 1 correspond to the P and Q axes, while x-axis of the basis camera 2 corresponds to the R axis in PGS.

Homogeneous coordinate $\mathbf{X} = (p, q, r, 1)^{\mathrm{T}}$ in PGS is projected on image coordinate $\mathbf{x} = (p, q, 1)$ of the basis camera 1 and $\mathbf{x}' = (r, s, 1)$ of the basis camera 2. $\mathbf{x}'$ must lie

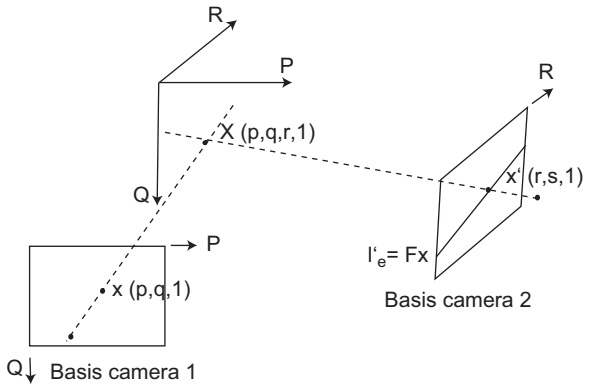

Fig. 4. Definition of projective grid space.

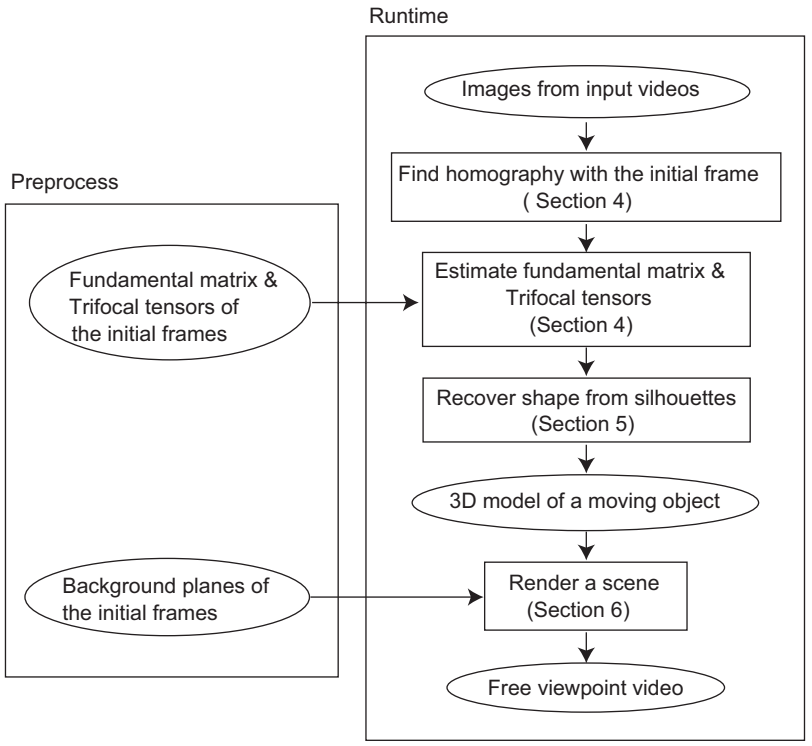

Fig. 3. Overview of our method.

on the epipolar line of $\mathbf{x}$, so s coordinate of $\mathbf{x}'$ is determined from $\mathbf{x}'^{T}F\mathbf{x} = \mathbf{0}$.

Other cameras (non-basis cameras) are said to be weakly calibrated once we can find the projection of a 3D point from the same PGS to those cameras. Either fundamental matrices or trifocal tensors between the basis cameras and the non-basis camera can be used for this task. The key idea is that 3D points in PGS will be projected onto the two basis cameras first to make 2D–2D point correspondence. Then, this correspondence is transferred to a non-basis camera by either the intersection of epipolar lines computed from fundamental matrices (Fig. 5) or point transfer by trifocal tensor (Fig. 7).

However, point transfer using fundamental matrices gives less accurate results if a 3D point lies near the trifocal plane (the plane defined by three camera centers). Thus, trifocal tensors are used for weakly calibrating non-basis cameras in our implementation of PGS. For completeness, we will explain how 3D points in PGS can be projected onto the non-basis cameras using fundamental matrices and discuss the drawbacks first. Then, we will explain about projecting 3D points in PGS to a non-basis camera using trifocal tensor.

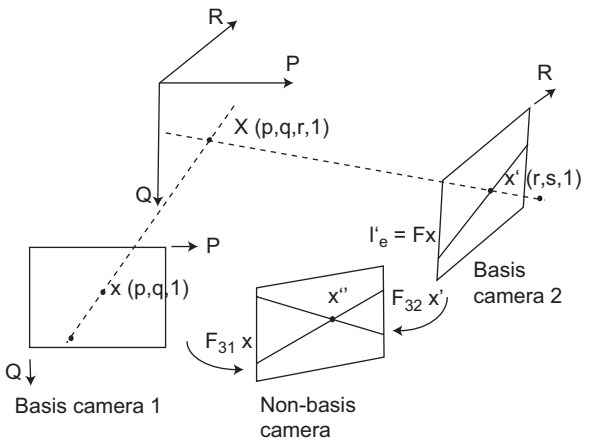

Fig. 5. Point transfer using fundamental matrices.

3.1. Weakly calibrating non-basis camera using fundamental matrices

When using fundamental matrices, the fundamental matrices between the basis cameras and a non-basis camera are estimated from at least seven point correspondences. The projected point in the non-basis camera is computed from the intersection of two epipolar line from the basis cameras. If the projected point in basis camera 1 and basis camera 2 is $\mathbf{x}$ and $\mathbf{x}'$, respectively, the correspondence in the non-basis camera will be

$$\mathbf{x}'' = (\mathbf{F}_{31}\mathbf{x}) \times (\mathbf{F}_{32}\mathbf{x}') \tag{1}$$

as illustrated in Fig. 5.

However, point transfer using fundamental matrices will fail when two epipolar lines are collinear. This happens when point $\mathbf{X}$ lies on the trifocal plane. Even in the less severe case, the transferred point will also become inaccurate for the points lying near this plane. This deficiency of point transfer using fundamental matrices can be avoided by arranging two basis cameras at different heights from the other cameras, like in Fig. 6(b). By arranging cameras this way, 3D points in the scene will not lie on the trifocal plane, and the intersection of epipolar lines will be well-defined. This approach is also used in [10–12].

3.2. Weakly calibrating non-basis camera using trifocal tensor

Trifocal tensor τ_i^{jk} is a homogeneous $3 \times 3 \times 3$ array (27 elements) that satisfies

$$l_i = l'_j l''_k \tau_i^{jk} \tag{2}$$

where l_i, l'_j and l''_k are the corresponding lines in the first, second and third image, respectively. For more details about tensor notation, refer to Appendix A.

Trifocal tensors can be estimated from point correspondences or line correspondences between three images. In case of using only point correspondences, at least seven point correspondences are necessary to estimate the trifocal tensor. Given point correspondence $\mathbf{x}$ and $\mathbf{x}'$, we can find corresponding point $\mathbf{x}''$ in the third

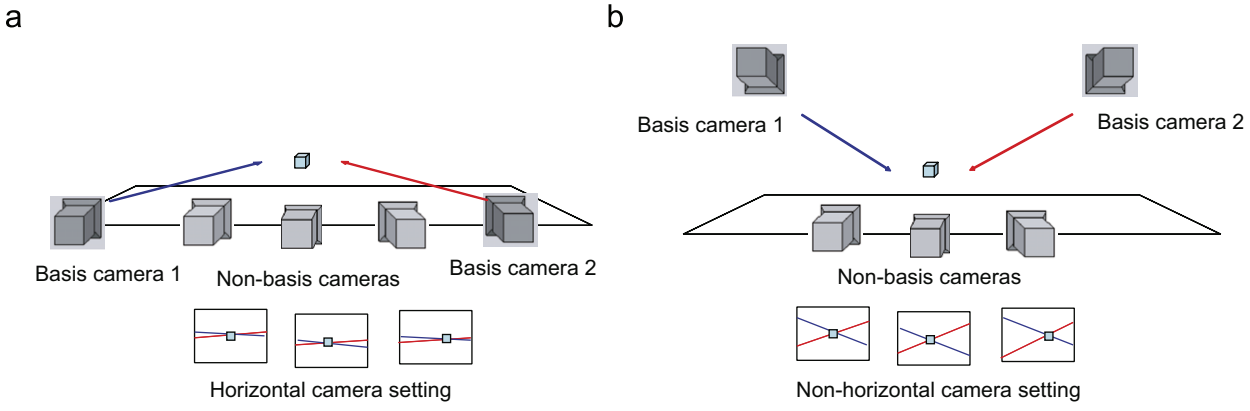

Fig. 6. Camera settings. (a) Bad arrangement of cameras for using epipolar transfer. (b) Good arrangement of cameras for using epipolar transfer.

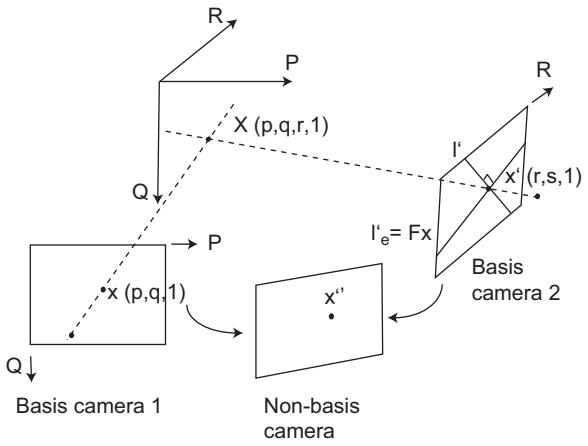

Fig. 7. Point transfer using the trifocal tensor.

camera using

$$x''^k = x^i l'_j \tau_i^{jk} \tag{3}$$

where $\mathbf{l}'$ is the line in the second camera that pass though point $\mathbf{x}'$.

Since, $x^i l'_j \tau_i^{jk} = 0^k$ and the point $\mathbf{x}''$ is undefined when $\mathbf{l}'$ is the epipolar line corresponding to $\mathbf{x}$. We can choose any line $\mathbf{l}'$ that pass point $\mathbf{x}'$, except the epipolar line corresponding to $\mathbf{x}$. A convenient choice for selecting the line $\mathbf{l}'$ is to choose the line perpendicular to the epipolar line of $\mathbf{x}$.

To summarize, given a 3D point $\mathbf{X} = (p, q, r, 1)^{\mathrm{T}}$ in PGS and tensor τ defined by basis camera 1, basis camera 2 and the non-basis camera, we can project point $\mathbf{X}$ to the non-basis camera as follows (see Fig. 7):

(1) Project $\mathbf{X} = (p, q, r, 1)^{\mathrm{T}}$ to $\mathbf{x} = (p, q, 1)^{\mathrm{T}}$ and $\mathbf{x}' = (r, s, 1)^{\mathrm{T}}$ on basis camera 1 and basis camera 2, respectively. s is obtained by solving $\mathbf{x}'^{\mathrm{T}} F \mathbf{x} = \mathbf{0}$.
(2) Compute epipolar line $\mathbf{l}'_e = (l_1, l_2, l_3)^{\mathrm{T}}$ of $\mathbf{x}$ on basis camera 2 from $\mathbf{l}'_e = F\mathbf{x}$.
(3) Compute line $\mathbf{l}'$ that pass $\mathbf{x}'$ and perpendicular to $\mathbf{l}'_e$ by $\mathbf{l}' = (l_2, -l_1, -rl_2 + sl_1)^{\mathrm{T}}$.
(4) The transferred point in the non-basis camera is $x''^k = x^i l'_j \tau_i^{jk}$.

3.3. Camera position in PGS

In Fig. 8, the 3D camera position of basis camera 1 in PGS is $(\mathrm{C1}_x, \mathrm{C1}_y, \mathrm{e12}_x)$, where $(\mathrm{C1}_x, \mathrm{C1}_y)$ is the camera center in basis camera 1, and $(\mathrm{e12}_x, \mathrm{e12}_y)$ is the epipole of basis camera 1 in basis camera 2. In the same way, the camera position of the basis camera 2 is $(\mathrm{e21}_x, \mathrm{e21}_y, \mathrm{C2}_x)$, where $(\mathrm{e21}_x, \mathrm{e21}_y)$ is the epipole of basis camera 2 in basis camera 1, and $(\mathrm{C2}_x, \mathrm{C2}_y)$ is the camera center in basis camera 2. For the non-basis camera, 3D camera position in the PGS is $(\mathrm{e1}_x, \mathrm{e1}_y, \mathrm{e2}_x)$ where $(\mathrm{e1}_x, \mathrm{e1}_y)$ and $(\mathrm{e2}_x, \mathrm{e2}_y)$ are epipoles on basis camera 1 and basis camera 2, respectively.

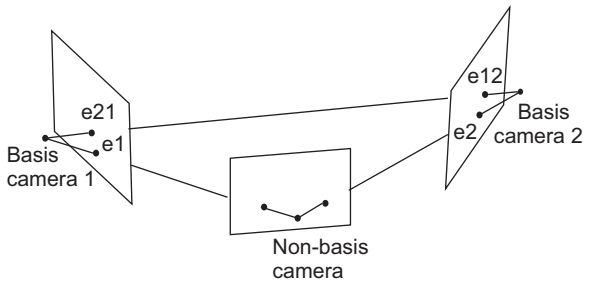

Fig. 8. Camera position in projective grid space.

4. Weak camera calibration

To weakly calibrate cameras to PGS, the fundamental matrix between the two basis cameras, and the trifocal tensors between the two basis cameras and the other non-basis camera need to be computed.

For example, in our experiment we use four hand-held camera inputs as shown in Fig. 2. If we select cameras 1 and 4 to be the basis cameras defining PGS, this means that we need to compute fundamental matrix between cameras 1 and 4, and two trifocal tensors defined by cameras 1, 4 and 2 and cameras 1, 4 and 3, respectively, for all frames.

Our approach for calibration includes two phases: preprocessing and runtime. During the preprocessing phase, we select one initial frame and estimate the fundamental matrix and trifocal tensors from manually selected correspondences. During runtime, our method can compute the fundamental matrix and trifocal tensors of the other frames automatically.

To demonstrate the process, we will explain the three camera case. Generalizing to more than three cameras is straightforward by increasing the number of non-basis cameras. Let ψ, ψ' and ψ'' represent the initial frames of basis camera 1, basis camera 2 and the non-basis camera, respectively. Let $\hat{\psi}$, $\hat{\psi}'$ and $\hat{\psi}''$ represent the other frames of the same camera.

4.1. Preprocessing phase

For the initial frames ψ, ψ' and ψ'', we zoom out all cameras to capture the whole area of a scene without an actor. The 2D–2D corresponding points for estimating fundamental matrix F between ψ and ψ' and trifocal tensor τ_i^{jk} of ψ, ψ' and ψ'' are assigned manually. Once the fundamental matrix and the trifocal tensor are estimated, PGS is completely defined. These images will be used as the reference image for calibrating the other input frames to PGS, as will be described in Section 4.2. Fig. 9 shows the initial frames ψ, ψ' and ψ''.

4.2. Runtime phase

Let $\hat{F}$ be the fundamental matrix from $\hat{\psi}$ to $\hat{\psi}'$. Let $\hat{\tau}_i^{jk}$ be trifocal tensor of $\hat{\psi}$, $\hat{\psi}'$ and $\hat{\psi}''$. We wish to compute $\hat{F}$ and $\hat{\tau}_i^{jk}$ automatically. The straightforward way is to estimate

Fig. 9. Initial frames.

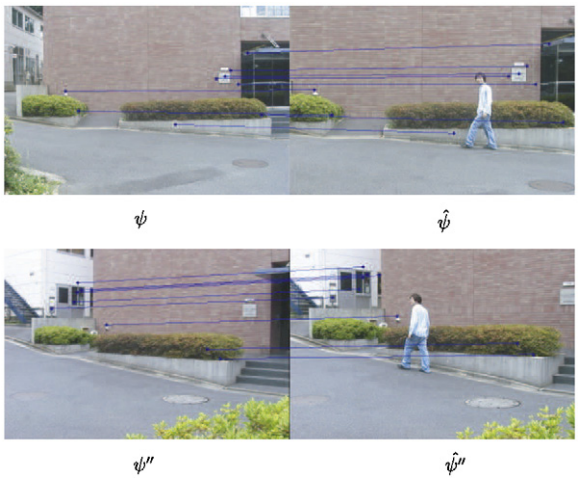

Fig. 10. Corresponding points found using SIFT for estimating homography.

from corresponding points among $\hat{\psi}$, $\hat{\psi}'$ and $\hat{\psi}''$. However finding such correspondences is error prone and difficult to achieve robustly in cases where the scene is a 3D scene and the baseline between cameras is large, as shown in [22].

We assume that the person recording the input video will not change position during capture. Thus, we may also assume that each camera is only rotating and zooming. The image coordinate $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{x}''$ of ψ, ψ' and ψ'' are transformed to the image coordinate $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}'$ and $\hat{\mathbf{x}}''$ of $\hat{\psi}$, $\hat{\psi}'$ and $\hat{\psi}''$ via homography matrices:

$$\hat{\mathbf{x}} = H\mathbf{x} \tag{4}$$

$$\hat{\mathbf{x}}' = H'\mathbf{x}' \tag{5}$$

$$\hat{\mathbf{x}}'' = H''\mathbf{x}'' \tag{6}$$

Under these point transformations, the fundamental matrix F will transform according to

$$\hat{F} = H'^{-\mathrm{T}} F H^{-1} \tag{7}$$

while the trifocal tensor τ_r^{st} will transform according to

$$\hat{\tau}_i^{jk} = (H^{-1})_i^r H_s'^j H_t''^k \tau_r^{st} \tag{8}$$

For the detailed proof of Eqs. (7) and (8), refer to [9]. From Eqs. (7) and (8), this means that we can estimate the fundamental matrix $\hat{F}$ and $\hat{\tau}_i^{jk}$ from the homographies between the initial frame given that the initial F and τ_i^{jk} are known.

In our experiments, we use the implementation for trifocal tensor estimation from [19]. To estimate homography matrix, corresponding points between ψ, ψ', ψ'' and $\hat{\psi}, \hat{\psi}', \hat{\psi}''$ are necessary. We employ SIFT for finding such correspondences. Example corresponding points that are automatically found using SIFT are shown in Fig. 10. In Fig. 10, the left image is the initial frame and the right image is the other frame which will be calibrated to PGS. RANSAC [6] is used to reject outliers in correspondences. The lines show corresponding points that will be used for estimating homography.

Finding correspondences between two images captured from the same position but with a change in focal length and rotation is more robust than finding correspondences between different views. This is because the two images captured from the same position will not have motion parallax. This is the motivation behind our calibration method.

5. 3D reconstruction

In this section, we describe how we reconstruct a 3D model of a human actor. We use an appearance-based approach for synthesizing free viewpoint video [27]. An appearance-based rendering is a combination of model-based rendering and image-based rendering. The reconstructed model is used for making a dense correspondence between the two original views for image interpolation.

We reconstruct a visual hull of a human actor in PGS using the silhouette volume intersection method [15]. To get a silhouette of human actor, we have to generate a virtual background for background subtraction. In the initial frame, we capture a background scene without human actor. In the later frames, a homography matrix, which is estimated for camera calibration as described in Section 4, is used for warping the initial frame to the current frame as a virtual background. Then background subtraction can be done as shown in Fig. 11. The RGB color $\mathbf{I}$ of a pixel p in the input image is compared to the RGB color $\mathbf{I}_{bg}$ of the same pixel in the warped background image by computing

$$\theta = \cos^{-1}\left(\frac{\mathbf{I}\cdot\mathbf{I}_{bg}}{|\mathbf{I}||\mathbf{I}_{bg}|}\right) \tag{9}$$

$$d = |\mathbf{I} - \mathbf{I}_{bg}| \tag{10}$$

The pixel p is segmented as a foreground pixel if $\theta > \theta_T$ or $d > d_T$ where θ_T and d_T are some thresholds. We then apply morphological operations to reduce the segmentation errors.

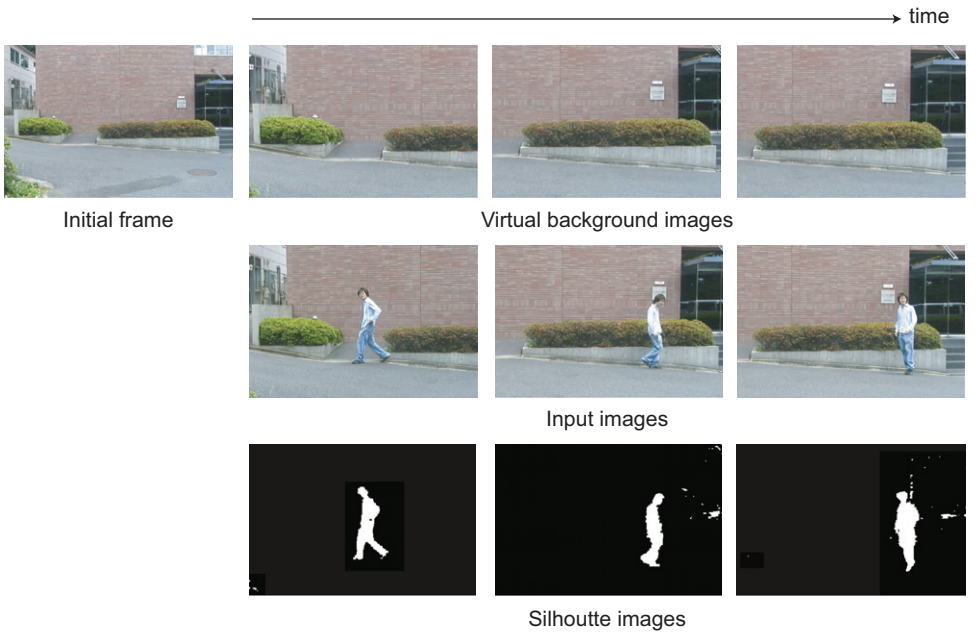

Fig. 11. Generating silhouette images of a human actor.

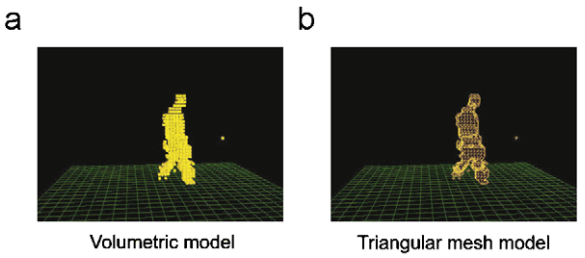

Fig. 12. 3D models of human in projective grid space.

Each voxel in PGS is projected onto silhouette images to test voxel occupancy. The surfaces of the volumetric model are extracted to a 3D triangular mesh model as shown in Fig. 12 using the Marching cube algorithm [17]. This 3D triangular mesh model will be used for making dense correspondences for view interpolation.

6. Free viewpoint video rendering

Our method can synthesize free viewpoint video using interpolation between the two reference views. Free viewpoint video is rendered in two steps. Background planes in a scene are rendered first. A moving object is then rendered and overlaid to the synthesized planes. The following subsections explain the details of the two rendering phases.

6.1. Background rendering

Our background scene is represented by several planes. Fig. 13 shows how we segment background scene.

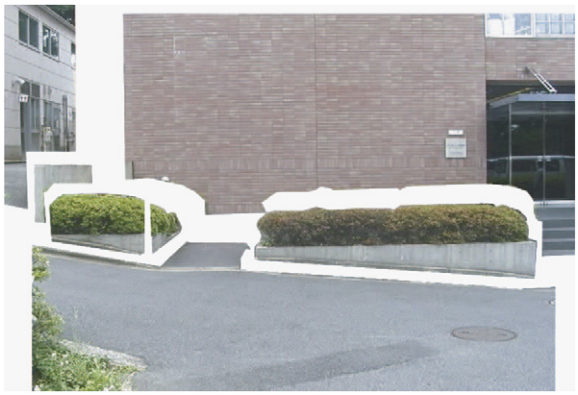

Fig. 13. Background scene is segmented into several planes.

During preprocessing, the initial frames that we used for calibration are manually segmented into several planes. The 3D positions of points that lie on those planes are reconstructed by specifying the corresponding points between basis camera 1 and basis camera 2. If $(p,q)^{\mathrm{T}}$ and $(r,s)^{\mathrm{T}}$ are correspondences in basis camera 1 and basis camera 2, respectively, then the 3D position in PGS of this point will be $(p,q,r)^{\mathrm{T}}$.

These 3D positions in PGS are projected onto both reference views. The 2D positions of these points on free viewpoint image are determined using linear interpolation

$$\begin{pmatrix} x \\ y \end{pmatrix} = w \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + (1-w) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{11}$$

where w is a weight, ranging from 0 to 1, defining the distance from the virtual view to the second reference

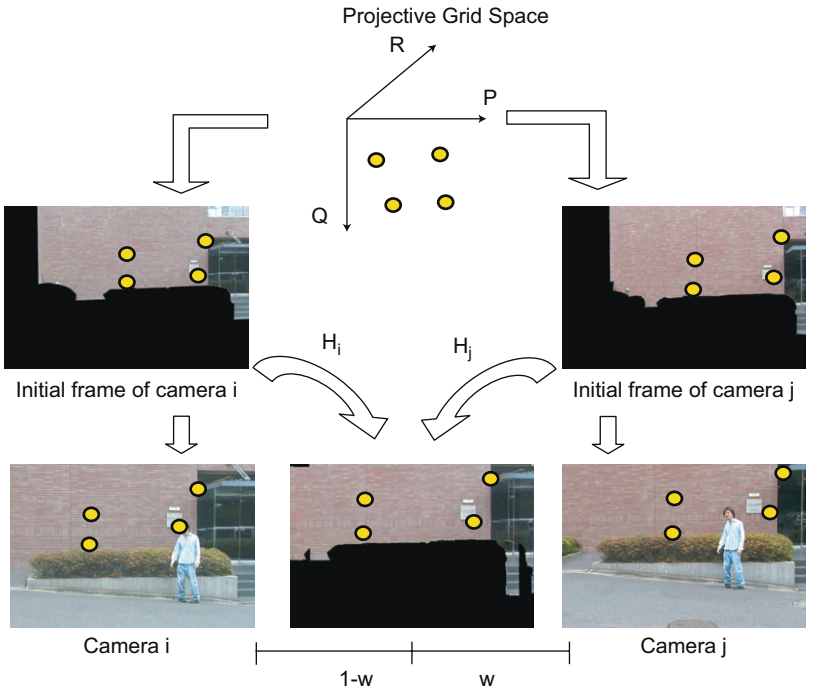

Fig. 14. Rendering a plane on a free viewpoint image.

view. $(x_1, y_1)^T$ and $(x_2, y_2)^T$ are the corresponding points on the first reference view and the second reference view, respectively. Corresponding points between the initial frames of the reference view and the virtual view are used for estimating a homography. The plane in the background image that is segmented during preprocessing is warped to the virtual view. Warped planes from two reference views are then blended together. In case that the scene consists of more than one plane, two or more planes in the virtual view are synthesized in this way and merged together. Fig. 14 illustrates how the plane is rendered in the free viewpoint image.

6.2. Moving object rendering

Free viewpoint images of a moving object is synthesized by view interpolation method [3]. The 3D triangular mesh model in PGS is used for making a dense correspondence and also for testing occlusion between the reference images.

To test occlusion of triangular patches, the z-buffer of each camera is generated. All triangular patches of a 3D model are projected onto the z-buffer of each camera. The values in the z-buffer for each pixel store the 3D distance from the focal point of a camera to the projected triangular patch. If some pixels are projected by more than one patch, the shortest distance is stored. The distance of point $\mathbf{a}(p_1, q_1, r_1)$ and $\mathbf{b}(p_2, q_2, r_2)$ in PGS is defined as

$$D = \sqrt{(p_1 - p_2)^2 + (q_1 - q_2)^2 + (r_1 - r_2)^2} \quad (12)$$

To synthesize a free viewpoint image, each triangular mesh is projected onto the two reference images. Any patch whose distance from the focal point of the input camera is greater than the value stored in the z-buffer is decided to be occluded. In the case that a patch is occluded in both input views, this patch will not be interpolated in a free viewpoint image. If a patch is seen from one or both input views, this patch will be warped and merged into a new view image. The position of a warped pixel in a new view image is determined using Eq. (11).

To merge warped triangular patches from two reference views, RGB colors of the pixel are computed by the weighted sum of the colors from both warped patches. If a patch is seen from both input views, the weight used for interpolating RGB color is the same for determining the position of a patch. In case that the patch is occluded in one view, the weight of the occluded view is set to 0 while the weight of the other view is set to 1. Fig. 15 shows an example of free viewpoint image of a moving object.

6.3. Hole filling

To combine the background with the moving object, the free viewpoint image of the moving object is rendered on top of the background. There might be some holes in the combined image because of the areas that are not visible in both reference views. These holes are easily noticed and also degrade the quality of the final output video. We use linear interpolation to fill out these holes. The hole filling process finds holes that are adjacent to

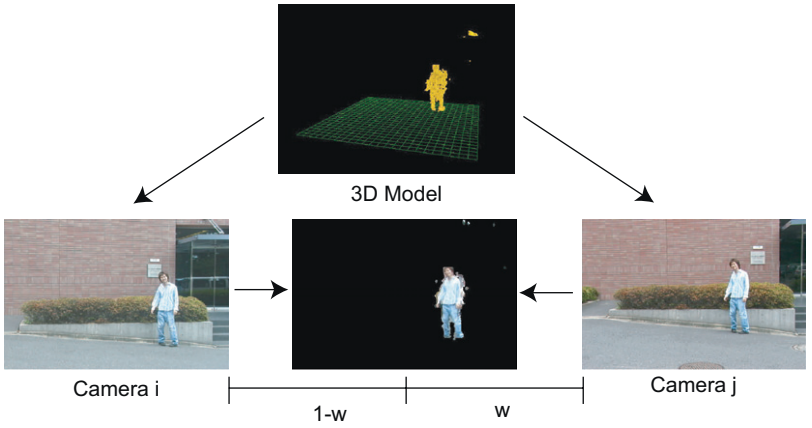

Fig. 15. Rendering a moving object on a free viewpoint image.

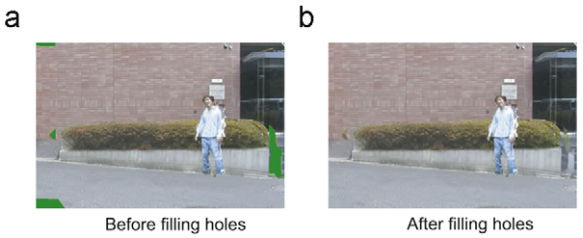

Fig. 16. Hole filling in the interpolated image. The green color pixels are holes that are not visible in both reference views.

some color pixels, and then interpolate that hole pixel using the average of the colors of nearby pixels. The process will stop when there are no more holes in the output video. Fig. 16 show an example image before and after filling holes.

7. Experimental results

In this section, we show our experimental results by synthesizing free viewpoint video from uncalibrated pure rotating and zooming cameras using the proposed method. We use four Sony-DV cameras with 720×480 resolution. All cameras are hand-held and captured without tripod as in Fig. 1. Note that the cameras used in our experiments are almost on the same horizontal line, which is not a suitable camera setting if the fundamental matrices are used for transferring correspondences (see Fig. 6). Video synchronization is done during digitization by Adobe Premiere Pro 2.0 (Adobe Premier Pro is a registered trademark of Adobe, Inc.).

We synthesize free viewpoint video from 300 consecutive frames by our proposed method. During the capturing process, each cameraman stood still, zoomed the camera and changed the view direction within the range of the initial frame independently. We zoom in and out approximately $1\times$ to $2\times$. The rotation angle of the cameras during capture from the left most view to the right most view is approximately 45°. Example input frames are shown in Fig. 2. There is no artificial markers placed in the scene. Only natural features are used for finding corresponding points. After the initial frame, our method can correctly calibrate all other frames to PGS and synthesize free viewpoint video without manual operation. Fig. 17 shows some example frames from the resulting free viewpoint video.

We select one frame from the input video and create new view images at several virtual camera ratios as shown in Fig. 18. The ratio between two views is given under each frame for different virtual views.

7.1. Subjective evaluation

From the results, we successfully create new view images from pure rotating and zooming cameras. Even there are artifacts, such as blurred texture or missing part of the moving object, overall quality is acceptable given that only four cameras are used for 3D reconstruction and the baseline between cameras is large (approximately 1.5–2.0 m). In this section, we give more detailed analysis of the cause of each artifact and discuss about potential solutions.

Fig. 19(a) shows that hole filling does not give a satisfactory result. If a hole appears near a particular object or dense textures, the result seems to be unconvincing. One possible solution is using information from the other views (not reference views) to fill holes.

In Fig. 19(b), there are some blurred textures or ghosting (double imaging) on the moving object in the synthesized image because of the inaccuracy of the reconstructed triangular mesh model. If the reconstructed mesh model is different from the real object, the warped textures from both reference cameras will be misaligned in the virtual view.

To reduce blurring or ghosting artifacts, one possible solution is to improve the accuracy of the 3D model. The straightforward way is to increase the number of cameras in the system. The newly added cameras will carve out the non-object voxels during volumetric

Fig. 17. Example free viewpoint images from consecutive 300 frames.

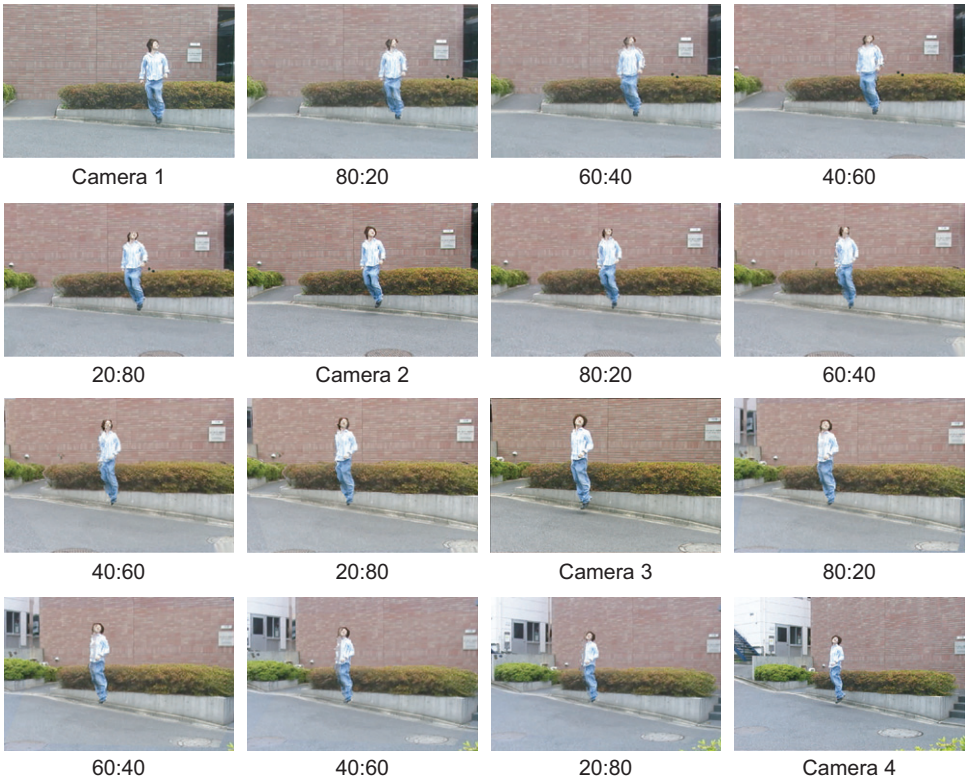

Fig. 18. Free viewpoint images from one input frame.

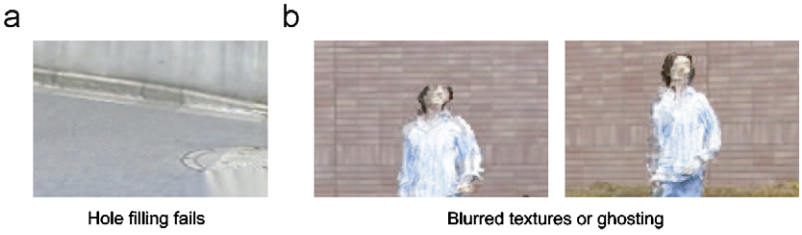

Fig. 19. Artifacts in the resulting new view images.

reconstruction, so the difference between the reconstructed shape and the real one will be reduced. However, the reconstructed visual hull gives only a coarse approximation to the actual shape of the object (concave areas cannot be reconstructed). An algorithm for optimizing meshes based on image textures and silhouettes can be applied [4,32].

Because the blurred textures occur when blending intensity of two misaligned textures, another solution is finding a good seam between textures instead of blending. Using this method, blurring or ghosting artifacts could be reduced without optimizing the 3D shape. This approach has been proposed in [16].

Another factor causing blurred textures is the trifocal tensor estimation error. Our method for computation is based on the assumption that the cameras are pure rotating and zooming. However, to show a practical application that this method is not limited to the case where the cameras are perfectly pure rotating like placing on a tripod, we use hand-held cameras that are held by a cameraman. Cameraman tries not to move the camera position, but there is still some handshake or other small movement. These contribute to the error during camera calibration.

Imperfect silhouette segmentation cause two kinds of artifacts: missing parts of the moving object and a hole-like region in the new view image. Missing parts of the silhouette images in some views cause missing parts of the moving object in the final free viewpoint image. The background area that is missegmented as the foreground area causes a phantom (no real object) in the reconstructed 3D model. This will appear as a hole-like artifact in the output video, as illustrated in Fig. 20. The color of this hole-like artifact will depend on the color of the texture in the reference cameras. Because our background is a natural scene, a completely clear silhouette is difficult to achieve using background subtraction.

7.2. Objective evaluation

This section gives objective quality measurements of our result. We use no-reference (no ground truth) evaluation method proposed in [31] to measure the error in registering scene appearance in image-based rendering. Two new view images at the center (ratio 50:50) between the two reference cameras are rendered. Each new view image is rendered using the texture only from the corresponding reference camera, as shown in Fig. 21.

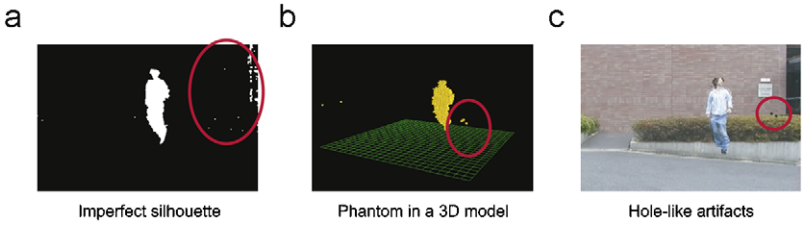

Fig. 20. The background area that is missegmented as the foreground causes a phantom in the 3D model and cause a hole-like artifact in the new view image. (a) Imperfect silhouette. (b) Phantom in a 3D model. (c) Hole-like artifacts.

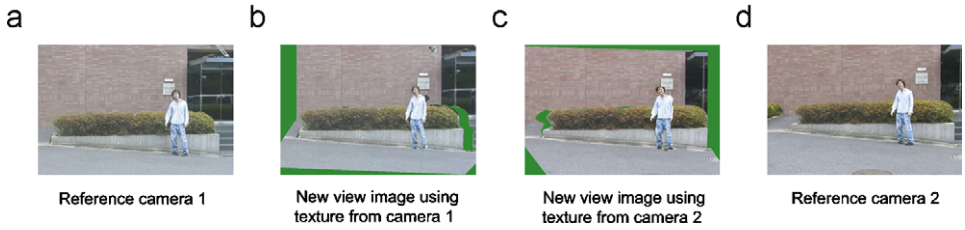

Fig. 21. New view images rendered for evaluating appearance registration errors. (a) Reference camera 1. (b) New view image using texture from camera 1. (c) New view image using texture from camera 2. (d) Reference camera 2.

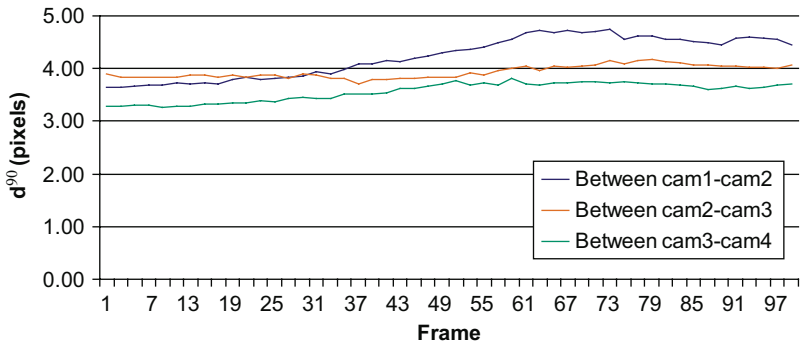

Fig. 22. d^{90} registration error of new view images.

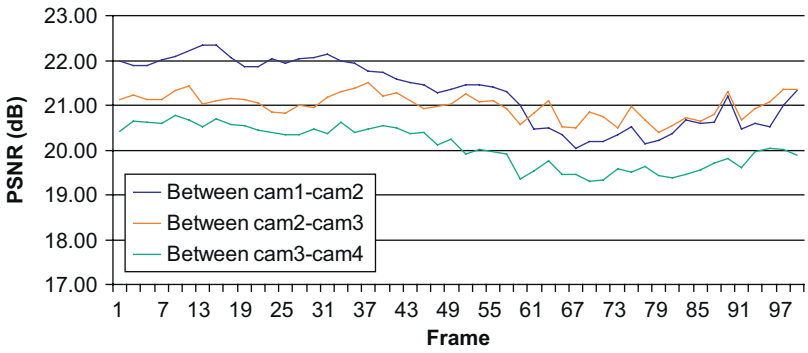

Fig. 23. PSNR registration error of new view images.

Table 1
Error measurements for the resulting new view images (average of 100 frames).

Virtual camera between	d^{90} (pixels)	*PSNR* (dB)
cam1–cam2	4.23	21.29
cam2–cam3	3.94	20.99
cam3–cam4	3.56	20.07

Two metrics d^{90} [31] and peak signal to noise ratio (*PSNR*) are computed over the overlapping pixels to measure the registration error in these new view images (reprojected appearances). d^{90} tells us about the overall distance of misaligned pixels between two images. The lower the value of d^{90}, the better the quality of the output in new view images. If the rendered image from one reference camera is much different from the other, then there will be visual artifacts, like blurred texture or ghosting in the blended image. We measure these values for 100 consecutive input frames between every adjacent cameras. Figs. 22 and 23 show each error metric of our new view images. Table 1 presents the average d^{90} and *PSNR* values over 100 frames.

8. Conclusion

We proposed a method for synthesizing free viewpoint video of a moving object in natural scene, which is captured by pure rotating and zooming cameras. Our method allows cameras to be zoomed and change view direction during capture within the field of view of the initial frames. Trifocal tensors are automatically estimated every frame, given the already estimated trifocal tensor in the initial frame. Our weak calibration method is done without special markers. Experimental results show that the proposed method is efficient, even when it is applied to the hand-held cameras with a small movement.

Appendix A. Tensor notation

This appendix gives an introduction to the tensors for the reader who is unfamiliar with tensor notation. For more details, refer to [9].

A tensor is a multidimensional array that extends the notion of scalar, vector and matrix. A tensor is written using an alphabet with contravariant (upper) and covariant (lower) indexes. For example, the trifocal tensor τ_i^{jk} has two contravariant indexes and one covariant index.

Considering a representation of vector and matrix using tensor notation, entry at row i and column j of matrix $\mathbf{A}$ is written using tensor notation as a_j^i, index i being contravariant (row) index and j being contravariant (column) index. An image point represented by the homogeneous column vector $\mathbf{x} = (x^1, x^2, x^3)^{\mathrm{T}}$ is written using tensor notation as x^i, while a line represented using the row vector $\mathbf{l} = (l_1, l_2, l_3)$ is written as l_i.

Writing two tensors together means doing a contraction operation. The contraction of two tensors produce a new tensor where each element is calculated from a sum of product over the repeated index. For example consider a matrix multiplication $\hat{\mathbf{x}} = \mathbf{A}\mathbf{x}$, this can be written using tensor notation as $\hat{x}^i = a_j^i x^j$. This notation imply a summation over the repeated index j as $\hat{x}^i = \sum_j a_j^i x^j$.

References

[1] T. Beier, S. Neely, Feature-based image metamorphosis, ACM Comput. Graph. 26 (2) (1992) 35–42 (in: Proceedings of SIGGRAPH'92).
[2] J. Carranza, C. Theobalt, M. Magnor, H.-P. Seidel, Free-viewpoint video of human actors, in: Proceedings of ACM SIGGRAPH'03, 2003, pp. 569–577.
[3] S. Chen, L. Williams, View interpolation for image synthesis, in: Proceedings of ACM SIGGRAPH'93, 1993, pp. 279–288.
[4] G. Eckert, J. Wingbermuhle, W. Niem, Mesh based shape refinement for reconstructing 3D-objects from multiple images, in: The First European Conference on Visual Media Production (CVMP), 2004, pp. 103–110.
[5] P. Eisert, E. Steinbach, B. Girod, Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views, IEEE Trans. Circuits Systems Video Technol. 10 (2000) 261–277 (special issue on 3D Video Technology).
[6] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography, Comm. ACM 24 (6) (1981) 381–395.
[7] B. Goldluecke, M. Magnor, Real-time microfacet billboarding for free-viewpoint video rendering, in: Proceedings of the IEEE International Conference on Image Processing, 2003, pp. 713–716.
[8] O. Grau, T. Pullen, G. Thomas, A combined studio production system for 3D capturing of live action and immersive actor feedback, IEEE Trans. Circuits Systems Video Technol. 3 (2004) 370–380.
[9] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, Cambridge, 2004 ISBN: 0521540518.

[10] Y. Ito, H. Saito, Free-viewpoint image synthesis from multiple-view images taken with uncalibrated moving cameras, in: IEEE International Conference on Image Processing (ICIP), 2005, pp. 29–32.
[11] S. Jarusirisawad, H. Saito, Free viewpoint video synthesis based on natural features using uncalibrated moving cameras, ECTI Trans. Electrical Eng. Electronics Comm. 5 (2) (2007) 181–190.
[12] S. Jarusirisawad, H. Saito, 3DTV view generation using uncalibrated cameras, in: Proceedings of the 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video, 2008, pp. 57–60.
[13] T. Kanade, P.W. Rander, P.J. Narayanan, Virtualized reality: concepts and early results, in: IEEE Workshop on Representation of Visual Scenes, 1995, pp. 69–76.
[14] H. Kato, M. Billinghurst, Marker tracking and HMD calibration for a video-based augmented reality conferencing system, in: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, 1999, pp. 85–94.
[15] A. Laurentini, The visual hull concept for silhouette based image understanding, IEEE Trans. Pattern Anal. Machine Intell. 16 (2) (1994) 150–162.
[16] V. Lempitsky, D. Ivanov, Seamless mosaicing of image-based texture maps, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 2007, pp. 1–6.
[17] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, in: Proceedings of ACM SIGGRAPH'87, 1987, pp. 163–169.
[18] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Internat. J. Comput. Vision 60 (2) (2004) 91–110.
[19] B. Matei, P. Meer, A general method for errors-in-variables problems in computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 18–25.
[20] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, L. McMillan, Image-based visual hulls, in: Proceedings of ACM SIGGRAPH'00, 2000, pp. 369–374.
[21] S. Moezzi, L.C. Tai, P. Gerard, Virtual view generation for 3D digital video, IEEE Multimedia 4 (1) (1997) 18–26.
[22] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3D objects, Internat. J. Comput. Vision 73 (3) (2007) 263–284.
[23] V. Nozick, S. Michelin, D. Arques, Real-time plane-sweep with local strategy, J. WSCG 14 (1–3) (2006) 121–128.
[24] M. Okutomi, T. Kanade, A multiple-baseline stereo, IEEE Trans. Pattern Anal. Machine Intell. 15 (4) (1993) 353–363.
[25] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual modeling with a hand-held camera, Internat. J. Comput. Vision 59 (3) (2004) 207–232.
[26] T. Rodriguez, P. Sturm, P. Gargallo, N. Guilbert, A. Heyden, J.M. Menendez, J.I. Ronda, Photorealistic 3D reconstruction from handheld cameras, Machine Vision and Applications 16 (4) (2005) 246–257.
[27] H. Saito, S. Baba, T. Kanade, Appearance-based virtual view generation from multicamera videos captured in the 3-D room, IEEE Trans. Multimedia 5 (3) (2003) 303–316.
[28] H. Saito, T. Kanade, Shape reconstruction in projective grid space from large number of images, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), vol. 2, 1999, pp. 49–54.
[29] S.M. Seitz, C.R. Dyer, Physically-valid view synthesis by image interpolation, in: Proceedings of the IEEE Workshop on Representations of Visual Scenes, 1995, pp. 18–25.
[30] J. Starck, A. Hilton, Towards a 3D virtual studio for human appearance capture, in: Proceedings of the IMA International Conference on Vision, Video and Graphics (VVG), 2003, pp. 17–24.
[31] J. Starck, J. Kilner, A. Hilton, Objective quality assessment in free-viewpoint video production, in: Proceedings of the 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video, 2008, pp. 225–228.
[32] S. Yaguchi, H. Saito, Improving quality of free-viewpoint image by mesh based 3D shape deformation, J. WSCG 14 (1–3) (2006) 57–64.
[33] C. Zhang, T. Chen, A self-reconfigurable camera array, in: ACM SIGGRAPH Sketches, 2004, p. 151.