3D Pose Tracking With Multitemplate Warping and SIFT Correspondences

Shu Chen, Luming Liang, Wenzhang Liang, and Hassan Foroosh, *Senior Member, IEEE*

***Abstract*— Template warping is a popular technique in vision-based 3D motion tracking and 3D pose estimation due to its flexibility of being applicable to monocular video sequences. However, the method suffers from two major limitations that hamper its successful use in practice. First, it requires the camera to be calibrated prior to applying the method. Second, it may fail to provide good results if the inter-frame displacements are too large. To overcome the first problem, we propose to estimate the unknown focal length of the camera from several initial frames by an iterative optimization process. To alleviate the second problem, we propose a tracking method based on combining complementary information provided by dense optical flow and tracked scale-invariant feature transform (SIFT) features. While optical flow is good for small displacements and provides accurate local information, tracked SIFT features are better at handling larger displacements or global transformations. To combine these two pieces of complementary information, we introduce a *forgetting factor* to bootstrap the 3D pose estimates provided by SIFT features, and refine the final results using optical flow. Experiments are performed on three public databases, i.e., the Biwi Head Pose dataset, the BU dataset, and the McGill Faces datasets. The results illustrate that the proposed solution provides more accurate results than baseline methods that rely solely on either template warping or SIFT features. In addition, the approach can be applied in a larger variety of scenarios, due to circumventing the need for camera calibration, thus providing a more flexible solution to the problem than existing methods.**

***Index Terms*— 3D pose estimation, appearance model, human motion tracking, template warping.**

Manuscript received October 13, 2014; revised February 19, 2015, April 13, 2015, and June 2, 2015; accepted June 26, 2015. Date of publication July 2, 2015; date of current version October 27, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61100139 and Grant 61040009, in part by the Construct Program of the Key Discipline in Hunan Province, in part by the Ministry of Science and Technology Innovation Fund for Technology-Based SMEs under Grant [2011] 242, in part by the Research Foundation of Education Bureau of Hunan Province, China, under Grant 16B258, in part by the 2013 Nanning Xiaogaodi Special Funding under Grant 2013022, and in part by the One Hundred Billion Yuan Industry Key Research Project in Guangxi Province under Grant 11107006-13. The work of H. Foroosh was supported by the U.S. National Science Foundation under Grant IIS-1212948 and Grant IIS-091686. This paper was recommended by Associate Editor J.-M. Odobez.

S. Chen is with the School of Information Engineering, Xiangtan University, Xiangtan 411105, China, and also with the Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Xiangtan 411100, China (e-mail: xtuchenshu@gmail.com).

L. Liang is with the Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO 80401 USA (e-mail: llmpass@gmail.com).

W. Liang is with GuangXi Cast Animation Company, Ltd., Nanning 530007, China (e-mail: 368697538@qq.com).

H. Foroosh is with the Computer Science Division, University of Central Florida, Orlando, FL 32816 USA (e-mail: foroosh@cs.ucf.edu).

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.2015.2452782

I. Introduction

3D POSE tracking from video sequences has a long history of research in computer vision due to its importance in a diverse set of applications, such as human–machine interaction, virtual reality, and visual surveillance, to name a few. Depending on the application area and the practical implementation constraints, the tracking methods fall into one of the following two categories: those using multiple-view video sequences and the ones based on using monocular video sequences. Having observations from multiple video cameras often resolves ambiguities. Thus, multiple-view 3D pose tracking often leads to a more accurate solution than tracking based on a monocular video sequence. However, multiple-view methods introduce several additional problems, including handover across cameras during tracking, calibration among the cameras, and of course, increased price and complexity of implementation. Although, tracking 3D pose from a monocular video sequence is more challenging with inherent ambiguities, the approach is more appealing, because of its flexibility and a wider range of applications. The template warping approach is popular in 3D pose tracking, because it leads to a dense set of correspondences, which can result in more precise results. With a specific 3D appearance model, template warping can achieve a very good performance. However, two main problems hinder the use of template warping in 3D pose tracking.

1) The focal length needs to be carefully estimated beforehand by some offline calibration method, which is time consuming and not practical in many applications.
2) When inter-frame motions are relatively large, the method may fail to provide a correct answer.

To alleviate these problems and to take full advantage of the flexibility of the template warping method, we make two major extensions to this approach.

1) We consider the focal length of the camera as an unknown parameter that can be estimated directly from the first few frames of the video by imposing some constraints.
2) We modify template warping so that it can handle large inter-frame displacements.

Template warping commonly assumes that the brightness or gradient constancy constraint applies to the video sequence. This implicitly implies that the convergence to the local minimum yields the correct solution. Unfortunately, this assumption is violated when the displacements are too large. As a result, the traditional template warping methods would lead to inaccurate results, especially when the tracked object performs drastic motions. Since the error is accumulated,

this also often leads to considerable drifts and loss of tracking. In this paper, we propose a framework to deal with the above two problems. The contributions are as follows.

1) The focal length is estimated online as a parameter optimized by an iterative procedure, therefore, eliminating the need for offline calibration, which is costly, time consuming, and reduces the user-friendliness and applicability of the method.
2) We estimate the object's 3D pose using a combination of template warping and scale-invariant feature transform (SIFT) feature correspondences, taking thus full advantage of the benefits of both methods in terms of handing both small and large displacements.
3) An online template update method is proposed to enable templates to represent the object's current appearance.

II. Related Work

Over the past two decades, many techniques have been developed for vision-based 3D tracking. Existing techniques can be roughly divided into two categories: 1) feature-based tracking and 2) model-based tracking [1]. Feature-based methods aim to recover the pose from a single image frame through classification or regression techniques [2]. The classifier is learned from training data that are generated offline with a synthetic model, or acquired by a camera from a small set of known poses [3]–[5]. Due to the large number of degrees of freedom (DoF), it is impractical to densely sample the entire state space. These are related to techniques for multiple-view object class detection [6], [7]. Demirkus *et al*. [6] propose a multistage semiautomatic framework for labeling temporal face classes, i.e., head poses, in unconstrained videos. Their approach provides only coarse viewpoint estimates. Since 3D sensing devices have become available, new methods have been developed to exploit the additional depth information, in order to enhance the robustness of the pose estimation methods [8]. Unfortunately, 3D sensing may not be always available in every application, particularly when portability is important, e.g., for mobile robots. Many discriminative methods have been proposed that focus on particular articulated object classes such as hands or bodies [9], [10]. Fan *et al*. [9] developed a pose locality constrained representation to model the 3D human body and use it to improve 3D human pose reconstruction. Belagiannis *et al*. [10] address the problem of 3D pose estimation of multiple humans from multiple views. More recently, deep learning has been employed in 3D tracking. Taylor *et al*. [11] model the problem as that of learning a nonlinear embedding, and use convolutional neural networks (CNNs) combined with neighborhood component analysis to regress toward a point representing the pose. Toshev and Szegedy [12] instead formulate the pose estimation as a joint regression problem and propose a cascade of deep neural networks (DNNs) to predict pose. Given a large number of training samples, DNNs can achieve better performances, but big data collection is a time-consuming task and might not be available in some cases.

Feature-based tracking methods can be difficult to extend to more general object types, since they incorporate large amounts of prior information, often extracted in a learning stage, and actively impose prior assumptions. Model-based tracking methods, on the other hand, refine an estimate (e.g., from a previous iteration) based on current time measurements. Model-based tracking always assumes simple shape models, and tracking can be performed with a 3D texture mesh or 3D feature mesh. Most popular tracking methods in this category match expected to observed edges by projecting a 3D wireframe model in the image [13]. Many extensions have been proposed that exploit both texture information and particle filtering in order to reduce sensitivity to background clutter and noise. Morency *et al*. [14] propose a probabilistic framework, called generalized adaptive view-based appearance model, integrating frame-by-frame head pose estimation, differential registration, and key-frame tracking. Some researchers recently focus on combining different cues to improve tracking performance. Pauwels *et al*. [15] exploit dense motion, stereo cues, sparse key-point features, and feedback from the model to track rigid objects with 6 DOFs. The method can be inherently parallelized and efficiently implemented using Graphic Processing Unit (GPU) acceleration. Unfortunately, GPUs are power hungry and might not be available in many scenarios. Brox *et al*. [16] propose the combined use of complementary concepts for 3D tracking, i.e., region fitting on the one hand, and dense optical flow as well as tracked SIFT features on the other. Recently, a similar approach is proposed, which blends a matching algorithm with a variational approach for optical flow [17].

3D pose trackers are subject to error accumulation, which eventually results in tracking failure and precludes running over long sequences. A solution to this problem is to introduce one or more key-frames that are images of the target object or scene for which the camera has been registered beforehand. At runtime, incoming images can be matched against the key-frames to provide a position estimate that is drift-free [6], [18]. Vacchetti *et al*. [18] suggested a method to merge online and offline key-frames for stable 3D tracking. These approaches are more accurate and subject to only bounded drift over time; however, they lack the higher precision of dynamic approaches. A problem with model-based tracking methods is that they have a hard time to deal with abrupt large displacements. To alleviate this problem, coarse-to-fine strategies [19] and nonlinearized models [20] have been proposed; however, these methods may still fail to track when the relative motion of a small-scale structure is larger than its own scale. Brox and Malik [21] present a way to approach this problem by integrating rich descriptors into the variational optical flow settings. Our approach tackles large displacements by integrating feature-based and intensity-based methods.

III. Overview of the Proposed Approach

The basic goal of our method is to estimate the 3D pose of an object under perspective projection from monocular video sequences without prior calibration of the camera or the knowledge of its focal length. The pose of the object is extracted by a combination of template warping and

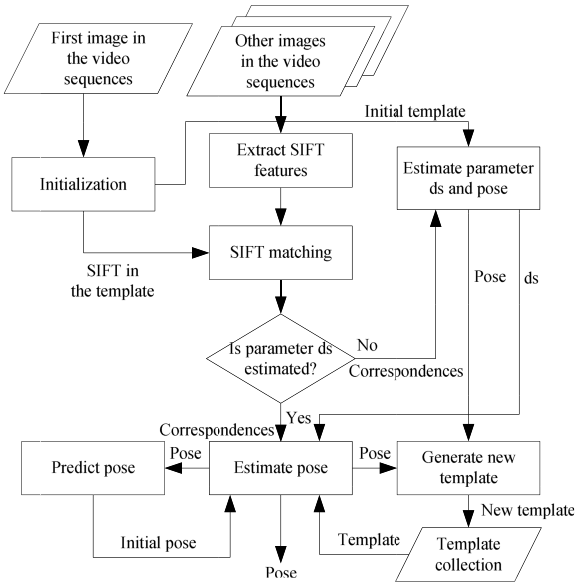


Fig. 1. Diagram of data flow in our system.

feature-based tracking, which can achieve accurate results in the presence of large displacements. Our approach for pose estimation is illustrated schematically in Fig. 1 and is described as follows.

1) By using a simple user interface, we generate a 3D texture template in the first frame, and extract SIFT features in this template. The 3D local coordinates of each pixel in the appearance template are then estimated (see Section IV-B).
2) The scale change parameter ds (defined below) is estimated by an iterative process, which follows two steps.
   a) Assuming image brightness is unchanged, we estimate pose by template registration.
   b) Assuming a rigid object shape, we estimate ds by SIFT correspondences (see Section V-C1).
3) The SIFT [22] features are extracted from the incoming frame and are used to establish correspondences between the 3D template and the 2D image for each view as described in Section V-B. The least median of squares method is employed to filter the outliers. The pose is estimated by combining template warping with a SIFT-correspondence-based tracker, as depicted in Section V-C2.
4) After estimating the pose, an autoregression is performed to predict the pose for the next frame. If the estimated result is precise, a new appearance template is generated by forward kinematics that warps the 3D model onto the image plane according to the estimated pose, and this new template is added into the template collection.

IV. Preliminaries

A. Perspective Projection Model

We use a simplified perspective camera model as described in [23]. In this model, the coordinates of a point (x, y, z) in the scene are projected to the image location (u, v) with

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{s} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{1}$$

Here, s is the scale factor that is defined as $s = z/f$, where z is the z-coordinate of the point in the scene and f is the focal length of the camera lens. The change in s (Δs) and the corresponding change in z (Δz) satisfy $\Delta s = \Delta z/f$. Thus, we have

$$\frac{1}{f} = \frac{\Delta s}{\Delta z} = \frac{ds}{1} \Rightarrow ds = \frac{1}{f} = \frac{\Delta s}{\Delta z} \tag{2}$$

where ds represents the change in s per unit change in depth z. Zou *et al.* [23] describe a simple method of estimating ds. However, it is customized only for human motion tracking. In our work, ds is treated as an unknown parameter, and is estimated by an optimization process.

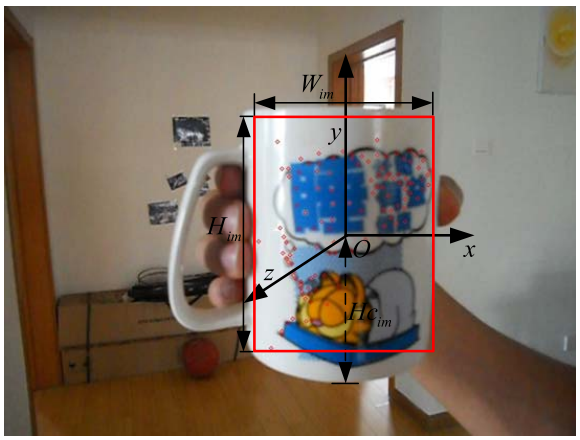


Fig. 2. Result of initialization. The red circles indicate the extracted SIFT feathers.

B. 3D Appearance Model

The 3D model is given by the user (for example, a cylinder, super quadric, or polygonal model), or is estimated by an initialization process. The initial template corresponds to only the visible part of the 3D appearance model in the first frame, and the other parts of the 3D appearance model are extracted as a new template by warping the 3D model onto the image plane after tracking each frame. In our work, we used a graphical user interface to select the projection of the tracked object in the first frame. The tracked object in the first frame is assumed to be fronto-parallel to the image plane. A marked image is shown in Fig. 2, in which the red rectangle depicts the position of the tracked object, the color circles represent the SIFT features obtained on the tracked object. The texture of the object is sampled from within the marked rectangle in the image. A local coordinate system is attached to the tracked object. The orientation of the local coordinate system is shown in Fig. 3, and the origin is located at the center of the tracked object. We denote the 3D local coordinates of each pixel in the appearance model by (x_s, y_s, z_s), and estimate them as follows.

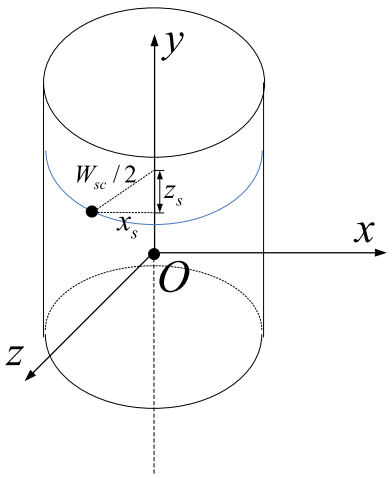


Fig. 3. Estimating z_s of pixels assuming that the appearance model is a cylinder.

Assuming that the tracked object is a cylinder, its height and width in the scene are denoted by W_{sc} and H_{sc}, respectively. W_{sc} can be set to any number, H_{sc} is set according to the ratio of W_{sc} to H_{sc}. For instance, given $W_{\text{sc}} = 10$, $\text{Ra} = H_{\text{sc}}/W_{\text{sc}} = H_{\text{im}}/W_{\text{im}}$, then, $H_{\text{sc}} = \text{Ra} \times W_{\text{sc}}$. H_{im} and W_{im} are the Euclidean lengths of the height and the width of the tracked object in the image plane, respectively. Since the tracked object is a cylinder and is initially parallel to the image plane, the scale factor s of each pixel along the same vertical line is the same. The scale factor s of each pixel in the left and right red vertical lines is estimated by $s_o = H_{\text{sc}}/H_{\text{im}}$. The scale factor of each pixel in the central vertical line is estimated by $s_c = H_{\text{sc}}/(2 \cdot Hc_{\text{im}})$, where Hc_{im} is the Euclidean length of the dotted line in the image plane (see Fig. 2). Based on the above definition, we have $s_c < s_o$, because the pixels in the central vertical line are closer to the camera than the pixels in the left and right vertical lines. Other pixels along the vertical lines between the outer vertical lines and the central line can be estimated accordingly since the scale factor s between s_o and s_c changes almost linearly.

Given the scale factor of a pixel has been obtained as s_p, using the above approach. The x and y coordinates of the pixel in the local coordinate system (x_s, y_s) are estimated by $(s_p \cdot x_l, s_p \cdot y_l)$, where x_l and y_l are the 2D local coordinates in the image plane, which are estimated by $x_l = x_{p\text{im}} - x_{o\text{im}}$, and $y_l = y_{p\text{im}} - y_{o\text{im}}$, where $(x_{p\text{im}}, y_{p\text{im}})$ and $(x_{o\text{im}}, y_{o\text{im}})$ are the image coordinates of this pixel and the origin of coordinates O, respectively. As shown in Fig. 3, z_s is estimated by $z_s = ((W_{\text{sc}}/2)^2 - x_s^2)^{1/2}$. We use the same method to obtain the 3D local coordinates of SIFT features as (x_f, y_f, z_f).

V. 3D Pose Tracking

We define the pose of an object as $[x, y, s_o, \alpha, \beta, \gamma]^T$, where $[x, y, s_o]^T$ are the coordinates of the center of the object (s_o is the scale factor of the point O, and the z-coordinate of this point can be estimated as $z = s_o/ds$), and $[\alpha, \beta, \gamma]^T$ are the orientation angles of the object, which correspond to the Euler rotation angles around the z-axis, y-axis, and x-axis, respectively. We assume that the order of rotation is z-y-x.

A. Pose Tracking Based on Template Warping

The basic idea of template warping is to use a texture-mapped surface model to approximate the tracked object. The model is first fitted to the initial frame to extract the reference texture, and then is warped to the subsequent image frames.

1) Template Warping: According to kinematics, for any point attached to a rigid object, the relation between local coordinates in the initial state and in the current state after rotation satisfy

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \mathbf{T} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \tag{3}$$

where $[x_s, y_s, z_s]^T$ are the local coordinates of the point in the initial frame, as described in Section IV-B and $[x_t, y_t, z_t]^T$ are the local coordinates of the point in the current frame (only rotation, no translation). $\mathbf{T}$ is the transform matrix defined as

$$\begin{aligned} \mathbf{T} = \mathbf{R}(\alpha)\mathbf{R}(\beta)\mathbf{R}(\gamma) &= \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} \\ r_{00} &= \cos\alpha \cos\beta \\ r_{01} &= -\sin\alpha \cos\gamma + \cos\alpha \sin\beta \sin\gamma \\ r_{02} &= \sin\alpha \sin\gamma + \cos\alpha \sin\beta \cos\gamma \\ r_{10} &= \sin\alpha \cos\beta \\ r_{11} &= \cos\alpha \cos\gamma + \sin\alpha \sin\beta \sin\gamma \\ r_{12} &= -\cos\alpha \sin\gamma + \sin\alpha \sin\beta \cos\gamma \\ r_{20} &= -\sin\beta \\ r_{21} &= \cos\beta \sin\gamma \\ r_{22} &= \cos\beta \cos\gamma . \end{aligned} \tag{4}$$

Thus, the coordinates of this point in the camera coordinate system can be estimated by

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5}$$

where $[x, y, z]^T$ are the coordinates of the origin O in the camera coordinate system. This point is thus projected to the image coordinates (u, v) according to (1) as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{6}$$

where s is the scale factor which satisfies the following equation, according to (2):

$$s = z_c \cdot ds \tag{7}$$

where ds is an unknown parameter which was described in Section IV-A, and we will describe how to estimate this parameter in Section V-C1. Upon substituting (5) in (6),

we get

$$u = \frac{A}{(-\sin\beta \cdot x_s + \cos\beta \sin\gamma \cdot y_s + \cos\beta \cos\gamma \cdot z_s) \cdot ds + s_o}$$
$$v = \frac{B}{(-\sin\beta \cdot x_s + \cos\beta \sin\gamma \cdot y_s + \cos\beta \cos\gamma \cdot z_s) \cdot ds + s_o}$$
$$\begin{aligned} A &= \cos\alpha \cos\beta \cdot x_s + (-\sin\alpha \cos\gamma + \cos\alpha \sin\beta \sin\gamma) \cdot y_s \\ &\quad + (\sin\alpha \sin\gamma + \cos\alpha \sin\beta \cos\gamma) \cdot z_s + x \\ B &= \sin\alpha \cos\beta \cdot x_s + (\cos\alpha \cos\gamma + \sin\alpha \sin\beta \sin\gamma) \cdot y_s \\ &\quad + (-\cos\alpha \sin\gamma + \sin\alpha \sin\beta \cos\gamma) \cdot z_s + y \end{aligned} \tag{8}$$

where $[x, y, s_o, \alpha, \beta, \gamma]^T$ are the pose parameters of the tracked object. Using (8), we define the warp function at time t as

$$\mathbf{W}(\mathbf{x}, \mathbf{p}|t) = \begin{bmatrix} W_x(\mathbf{x}, \mathbf{p}|t) \\ W_y(\mathbf{x}, \mathbf{p}|t) \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{9}$$

where $\mathbf{p} = [x, y, s_o, \alpha, \beta, \gamma]^T$, $\mathbf{x} = [x_s, y_s, z_s]$ are the coordinates of one pixel in the template. If we assume that changes in image intensity are only due to translation of local image intensity, then

$$I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t)) = I(\mathbf{x}|t_0) \tag{10}$$

where $I(\mathbf{x}|t_0)$ is the intensity of the pixel at coordinates $\mathbf{x}$ in the initial image, and $I(\mathbf{W}(\mathbf{x}; \mathbf{p}|t))$ is the intensity of the pixel at coordinates $\mathbf{W}(\mathbf{x}, \mathbf{p})$ in the time frame t. We define the residual error function as

$$E_{\text{image}} = \sum_{\mathbf{x} \in \Omega} [I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t)) - I(\mathbf{x}|t_0)]^2 \tag{11}$$

where Ω is the region of the template. Therefore, the optimal pose can be obtained by minimizing (11): $\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{\mathbf{x} \in \Omega} [I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t)) - I(\mathbf{x}|t_0)]^2$. We used the Newton–Raphson method to minimize (11) to obtain the pose of tracked object. Assuming that a current estimate of $\mathbf{p}_0$ is obtained, we iteratively estimate increments of the parameters, i.e., the following expression is minimized:

$$\sum_{\mathbf{x} \in \Omega} [I(\mathbf{W}(\mathbf{x}, \mathbf{p}_0 + \Delta\mathbf{p}|t)) - I(\mathbf{x}|t_0)]^2 \tag{12}$$

with respect to $\Delta\mathbf{p}$, and then the parameters are updated

$$\mathbf{p}_0 \leftarrow \mathbf{p}_0 + \Delta\mathbf{p}. \tag{13}$$

These two steps are iterated until the estimates of the parameters converge. Using the Newton–Raphson method, the increments $\Delta\mathbf{p}$ are estimated as

$$\begin{aligned} \Delta\mathbf{p} &= \left(\sum_{\mathbf{x} \in \Omega} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \right)^{-1} \\ &\quad \times \left(\sum_{\mathbf{x} \in \Omega} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{x}|t_0) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t))] \right) \end{aligned} \tag{14}$$

where

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial \alpha} & \frac{\partial W_x}{\partial \beta} & \frac{\partial W_x}{\partial \gamma} & \frac{\partial W_x}{\partial x} & \frac{\partial W_x}{\partial y} & \frac{\partial W_x}{\partial s_o} \\ \frac{\partial W_y}{\partial \alpha} & \frac{\partial W_y}{\partial \beta} & \frac{\partial W_y}{\partial \gamma} & \frac{\partial W_y}{\partial x} & \frac{\partial W_y}{\partial y} & \frac{\partial W_y}{\partial s_o} \end{pmatrix}. \tag{15}$$

The iterative procedure is summarized in Algorithm 1.

Algorithm 1 Template Warping

1: **repeat**
2: Warp I with $\mathbf{W}(\mathbf{x}, \mathbf{p}|t)$ to compute $I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t))$;
3: Compute the intensity difference $I(\mathbf{x}|t_0) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}|t))$;
4: Warp the gradient ∇I with $\mathbf{W}(\mathbf{x}, \mathbf{p}|t)$;
5: Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}, \mathbf{p}|t)$ using (15);
6: Compute $\Delta\mathbf{p}$ using (14);
7: Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$;
8: **until** $\|\Delta\mathbf{p}\| < \varepsilon$

2) Multitemplate Warping: It is impossible to use a single template through an entire video sequence, because the extracted initial template is often only a part of the 3D appearance model. While the tracked object performs self-rotation ($\beta \neq 0$), the template obtained in the initial frame will be occluded by other parts of the tracked object, i.e., self-occlusions exist, resulting in inaccurate tracking. Therefore, an approach based only on using the initial template would inevitably fail. In addition, it would be difficult for a single template to deal with other problems, such as a gradual lighting changes. To overcome these problems, a multitemplate warping method is introduced, which is summarized in Algorithm 2, where N_t is the current number of templates in the template collection and $N_{\max}$ is the maximum number of templates in the template collection. The residual error of registration is obtained by

$$E_i = \frac{\sum_{\mathbf{x} \in \Omega} [I(\mathbf{W}(\mathbf{x}; \mathbf{p}|t)) - I_i(\mathbf{x}|t_0)]^2}{n} \tag{16}$$

where n is the number of pixels in the template. When the residual error E is smaller than a preset threshold T, a new template is generated and added into the collection of templates. To obtain the new appearance template, the image coordinates of each point in the new template are estimated by warping the 3D model onto the image plane with the estimated pose, and the texture is extracted from the current frame. The implementation details can be found in [24].

B. Pose Tracking Based on SIFT Correspondences

We used a method similar to the one proposed in [24] to estimate the pose of the tracked object. First, the SIFT features are extracted from the unoccluded object region, and then SIFT correspondences between two different images are established using a nearest neighbor distance ratio matching. To speed up the matching, a k-D tree is built to search the nearest neighbors.

Given a pair of matched SIFT correspondences, $\mathbf{X}_f = [x_f, y_f, z_f]^T$ are the initialized local coordinates of the obtained feature, as depicted in Section IV-B, $\mathbf{X}'_f = [x'_f, y'_f, z'_f]^T$ are the local coordinates of this feature point in the current frame, and the relation between $\mathbf{X}_f$ and $\mathbf{X}'_f$ satisfies

$$\mathbf{X}'_f = \mathbf{T}\mathbf{X}_f \tag{17}$$

Algorithm 2 Multitemplate Warping

Set $E = 1000000000$;
for i=1:N_t **do**
  Draw template T_i from collection C;
  Use the methods proposed in section V-A-1 to estimate the pose of tracking object $\mathbf{P}_i$, and calculate the residual error E_i according to (16);
  if $E_i < E$ **then**
    $E = E_i$; $\mathbf{P} = \mathbf{P}_i$;
  end if
end for
Output the optimal pose $\mathbf{P}$;
Obtain the template T_c according to pose $\mathbf{P}$;
if $E < T$ **then**
  if $N_t < N_{max}$ **then**
    Add T_c into template collection C;
  else
    Replace the oldest template with T_c;
  end if
end if

where $\mathbf{T}$ is the transformation matrix, which is given by (4): (x'_f, y'_f) subject to

$$\begin{bmatrix} x'_f \\ y'_f \end{bmatrix} = \begin{bmatrix} x_c^f - x \\ y_c^f - y \end{bmatrix} \tag{18}$$

where (x_c^f, y_c^f) and (x, y) are the x and y coordinates of the feature point and the center of the object in the camera projection space, respectively. (x_c^f, y_c^f) can be established by

$$\begin{bmatrix} x_c^f \\ y_c^f \end{bmatrix} = \begin{bmatrix} (s+\Delta s)\cdot x_i^f \\ (s+\Delta s)\cdot y_i^f \end{bmatrix} = \begin{bmatrix} (s_o + z'_f \cdot ds)\cdot x_i^f \\ (s_o + z'_f \cdot ds)\cdot y_i^f \end{bmatrix} \tag{19}$$

where s is the scale factor of the origin of coordinates O in the current frame, denoted by s_o; (x_i^f, y_i^f) are the image coordinates of the feature point and z'_f is the local z-coordinate of the feature after rotation. According to (17), z'_f satisfies $z'_f = -x_f \sin\beta + y_f \cos\beta \sin\gamma + z_f \cos\beta\cos\gamma$. Substituting (19) into (18) yields the following system of equations:

$$\begin{aligned} x'_f &= [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma \\ &\qquad + z_f \cos\beta\cos\gamma)\cdot ds]\cdot x_i^f - x \\ y'_f &= [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma \\ &\qquad + z_f \cos\beta\cos\gamma)\cdot ds]\cdot y_i^f - y. \end{aligned} \tag{20}$$

Upon substituting (20) into (17), we get

$$\begin{cases} [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma + z_f \cos\beta\cos\gamma)\cdot ds]\cdot x_i^f \\ \quad - x = x_f \cos\alpha\cos\beta - y_f \sin\alpha\cos\gamma + y_f \cos\alpha \sin\beta\sin\gamma \\ \qquad + z_f \sin\alpha\sin\gamma + z_f \cos\alpha\sin\beta\cos\gamma \\ [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma + z_f \cos\beta\cos\gamma)\cdot ds]\cdot y_i^f \\ \quad - y = x_f \sin\alpha\cos\beta + y_f \cos\alpha\cos\gamma + y_f \sin\alpha\sin\beta\sin\gamma \\ \qquad - z_f \cos\alpha\sin\gamma + z_f \sin\alpha\sin\beta\cos\gamma. \end{cases} \tag{21}$$

We thus establish the loss function as

$$E_{\text{fp}} = \sum_{\mathbf{w}_f} \begin{bmatrix} E_x(\mathbf{w}_f, \mathbf{p}) \\ E_y(\mathbf{w}_f, \mathbf{p}) \end{bmatrix}^T \begin{bmatrix} E_x(\mathbf{w}_f, \mathbf{p}) \\ E_y(\mathbf{w}_f, \mathbf{p}) \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} &E_x(\mathbf{w}_f, \mathbf{p}) \\ &= [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma + z_f \cos\beta\cos\gamma)\cdot ds] \\ &\quad \times x_i^f - x - x_f \cos\alpha\cos\beta + y_f \sin\alpha\cos\gamma \\ &\quad - y_f \cos\alpha\sin\beta\sin\gamma - z_f \sin\alpha\sin\gamma \\ &\quad - z_f \cos\alpha\sin\beta\cos\gamma \\ &E_y(\mathbf{w}_f, \mathbf{p}) \\ &= [s_o + (-x_f \sin\beta + y_f \cos\beta \sin\gamma + z_f \cos\beta\cos\gamma)\cdot ds] \\ &\quad \times y_i^f - y - x_f \sin\alpha\cos\beta - y_f \cos\alpha\cos\gamma \\ &\quad - y_f \sin\alpha\sin\beta\sin\gamma + z_f \cos\alpha\sin\gamma \\ &\quad - z_f \sin\alpha\sin\beta\cos\gamma \end{aligned} \tag{23}$$

where $\mathbf{p}$ is defined in the same manner as in (9) and $\mathbf{w}_f = (x_f, y_f, z_f, x_i^f, y_i^f)$ represents the SIFT parameter. Therefore, the optimal pose can be obtained by minimizing (22), and the new pose of the object can be estimated using the technique described in Section V-A1.

C. Pose Tracking Based on Combining Template Warping and SIFT Correspondences

In the previous section, we introduced the details of how the pose of an object is estimated by image registration or by SIFT correspondences. Image registration uses dense correspondences, which can achieve good performance in pose extraction, but it obviously accumulates errors over time, and fails to cope with large displacements. SIFT-correspondence-based tracking uses descriptor matching, which can estimate large displacements and is free from the tracking drift, but the extracted SIFT features are very sparse in most situations and will lead to inaccurate estimation on their own. Therefore, it is crucial to combine motion-based correspondences with image registration.

1) Estimation of ds: According to (22), we define an objective function as

$$E_{ds} = \sum_{\mathbf{w}_f} \begin{bmatrix} E_x(\mathbf{w}_f, ds) \\ E_y(\mathbf{w}_f, ds) \end{bmatrix}^T \begin{bmatrix} E_x(\mathbf{w}_f, ds) \\ E_y(\mathbf{w}_f, ds) \end{bmatrix} \tag{24}$$

where $E_x(\mathbf{w}_f, ds)$ and $E_y(\mathbf{w}_f, ds)$ are defined as (23). In this function, $\mathbf{p} = [x, y, s_o, \alpha, \beta, \gamma]^T$ is known, and the parameter to be estimated is ds.

Given an initial estimation ds, we construct an iterative approach to estimate the parameter ds as follows.

1) Estimate pose $\mathbf{p}$ by minimizing (11).
2) Estimate ds by minimizing (24).
3) Repeat until convergence.

The parameter ds is estimated in the first several frames, in which the obtained Euler angle β is larger than a threshold, and a median filter is employed to obtain the robust estimation of ds.

2) Tracking: By combining the loss functions (11) and (22), the pose of the tracked object can be estimated by optimizing the following cost function:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}}[(1-\lambda)E_{\text{image}} + \lambda E_{\text{fp}}] \tag{25}$$

where $\lambda = 0.9/n$ is a forgetting factor in each stage of the iterative optimization procedure, where n is the current iteration. From the definition of λ, we know that the value of λ will steadily decrease during the iterative process. The benefits of introducing the forgetting factor can be described as follows: at the beginning of the iterative process, λ is relatively large, which indicates that the SIFT-correspondence-based component contributes more to the objective function and enables the iteration to quickly move toward the neighborhood of the ground truth. As λ decreases, image registration gradually plays a more important role than SIFT in the optimization process, which enables the estimation of $\mathbf{p}$ to be refined, thereby achieving an accurate solution.

Coarse-to-fine warping was also used for improving the performance, in which the pose was estimated at a range of image resolutions [25]. First, the original image was down-sampled to create a series of different resolution images (a Gaussian image pyramid), and then, we started at a coarse resolution and iterated to convergence at each level before projecting the current solution to the next level of the pyramid. This strategy is more efficient and can converge to the correct solution from a distant initialization. To cope with the lighting variations and the object self-rotation, the whole tracking was also implemented in the multitemplate warping framework as described in Section V-A2. The data flow of pose estimation is shown in Fig. 4.

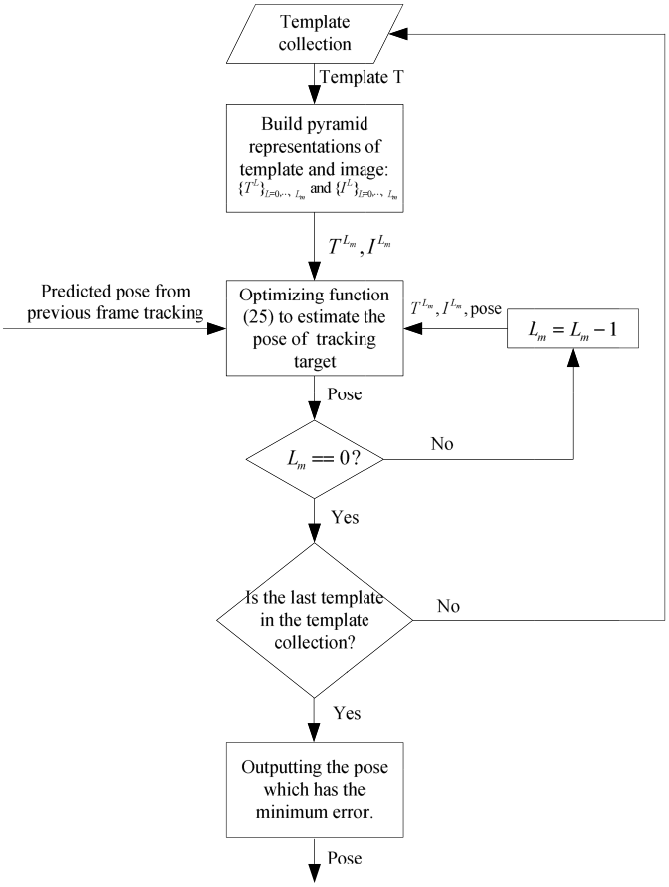


Fig. 4. Data flow of pose estimation.

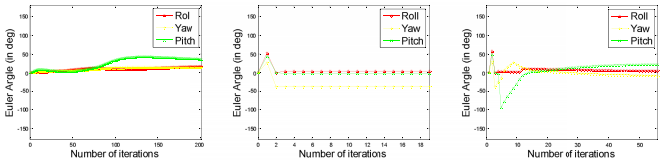


Fig. 5. Sequence of estimated Euler angles within the iterative process. (a) Optical flow. (b) SIFT-based tracker. (c) Our approach. Number of iterations from optical flow, SIFT-based tracker, and our proposed approach are 201, 19, and 56, respectively.

VI. Experiments

In order to demonstrate the effectiveness of the proposed approach, we performed extensive experiments on various kinds of object motions under different scenarios. The user specifies the model in the initial frame, and its positions in the subsequent frames are computed according to the estimated motion, and the estimated pose. In all experiments, the original image is converted into a gray image and is used as the input source. The number of templates is set to 6, T is set to 10, and the number of levels of the image pyramid is set to 3. In our experiments, we used two measures to eliminate any potential drift in the solution. First, the initial template is always included in the collection, and is never replaced. Second, two thresholds, T_{image} and T_{fp} are used to prevent selecting an inaccurate new template: the new template must satisfy $E_{\text{image}} < T_{\text{image}} \cap E_{\text{fp}} < T_{\text{fp}}$, where E_{image} and E_{fp} are obtained using (11) and (22).

A. Qualitative Evaluation

Fig. 5 demonstrates the sequence of estimated angles within the iterative process of an experiment, where a mug undergoes large displacements and rotations. From Fig. 5, we find that the convergence of optical flow is very slow, while the SIFT-based tracker exhibits a faster convergence. Both of these two approaches converge to apparently incorrect solutions. Our approach quickly moves to the neighborhood of plausible pose after 14 iterations. In the final stage of the iterations, our approach further refines the result based on template warping. The evolution of the pose is shown in Fig. 6. Fig. 7 shows the corresponding tracking results. As shown in Fig. 7(d), the computed optical flow is inaccurate due to the large motions. Fig. 7(c) indicates that in this scene there are no enough matched SIFT correspondences on the object for tracking. The pose estimation also fails if only the SIFT correspondences are used, as shown in Fig. 7(e). However, due to the combination of both methods, our algorithm better tracks the object [Fig. 7(f)]. This experiment demonstrates that there are scenes where none of the two methods used individually can track the object correctly. There is clearly a difference between the usage of correspondences from optical

Fig. 6. Visual results during the iterations. (a) Optical flow. From left to right: pose estimated after 20, 50, 90, 140, and 180 iterations. (b) SIFT-based tracker. From left to right: pose estimated after 1, 2, 3, 9, and 16. (c) Our proposed approach. From left to right: pose estimated after 2, 10, 20, 30, and 40 iterations.

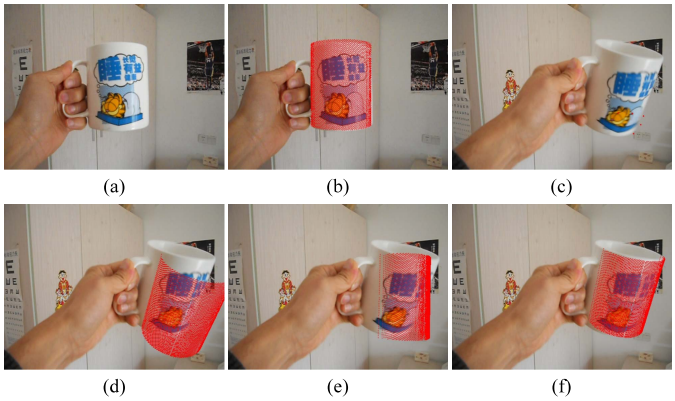


Fig. 7. Proposed algorithm is able to track the large motion of a mug. (a) Input image (frame 1). (b) Object pose at frame 1. (c) Image to be estimated (frame 2); the red circles indicate the matched SIFT correspondences at frame 2. Template matching or SIFT-based tracking alone cannot handle this situation. (d) Inaccurate pose estimated at frame 2 by optical flow. (e) Inaccurate pose estimated by SIFT-based tracker. (f) Estimated pose using a combination of template warping and SIFT correspondences.

flow and SIFT. Optical flow requires dense sampling in time which leads to accurate object tracking, but it cannot deal with large transformations and motions. For the SIFT tracker, on the other hand, a large displacement is not a problem. SIFT correspondences are usually more reliable. However, a sufficient number of them is needed in order to correctly estimate the pose. Therefore, by combining the two methods, one can achieve much better tracking.

We illustrate a comparative result of single-template warping and multitemplate warping for our algorithm in the presence of a partial occlusion in Fig. 8. In this video sequence, the jar controlled by a hand rotates drastically, and partial occlusions occur in some frames. One template warping produces plausible results but fails to recover the precise pose when part of the jar is self-occluded. Although, SIFT features can be uniquely matched between images, SIFT is not invariant with respect to angle changes. In this situation, the extracted SIFT correspondences are not enough to guarantee the precise pose estimation. On the other hand, matching pixels for parts of the template are missing in the case of self-occlusions. Such missing pixels and mismatches disturb the template warping process. However, in multitemplate warping we learn

Fig. 8. Comparison of tracking results for a test video sequence with self-occlusions. (a) Estimated pose using a combination of template warping and SIFT correspondences. (b) Estimated pose using a combination of multitemplate warping and SIFT correspondences.

Fig. 9. Frames 90, 132, 251, and 292 from a monocular sequence with a moving object which was occluded by a moving hand. (a) Input frames. (b) Extracted SIFT features. (c) Estimated pose using optical flow. (d) Estimated pose using SIFT-based tracker. (e) Estimated pose using our approach.

a set of view-based templates online, allowing to recover some invisible parts of the object, and take into account the change in appearance of the tracked object. The object in the tracked frame can be precisely registered by one of the templates.

Another demonstration of the occlusion is shown in Fig. 9. In this experiment, the swaying mug is occluded by a moving hand. Optical flow almost fails due to the estimation errors caused by the occluded areas. These errors are produced by unreliable, intensity matching. The SIFT-based tracker works well when there are a few SIFT regions on the mug, but it also fails if the matched SIFT features are not sufficient to estimate a precise pose. With the combination of the two sources, our method demonstrates higher robustness to occlusions. Although our approach can handle certain types of occlusions where a suitable number of extracted SIFT correspondences are obtained, failures can occur when an entire side of the object is occluded.

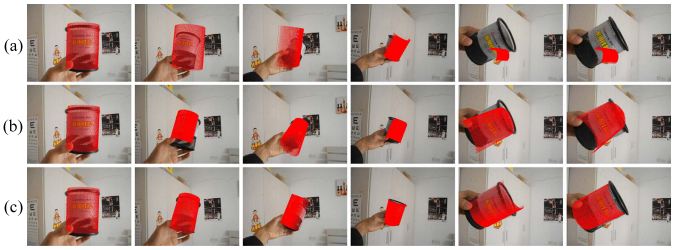


Fig. 10. Comparison of some tracking results on a low-frame-rate video (frames 30, 70, 130, 260, 390, and 400). (a) Estimated pose using template warping. (b) Estimated pose using SIFT-based tracker. (c) Estimated pose using our approach.

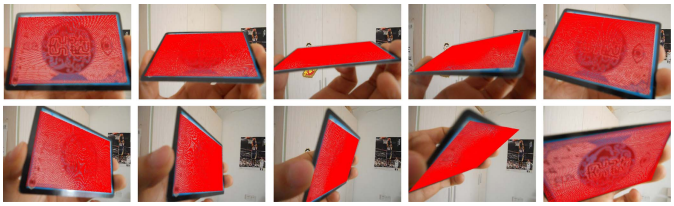


Fig. 11. Results of pose estimation for a test video sequence with strong perspective effects (every 100th frame from 100 to 1000, arranged from left to right and top to bottom).

In Fig. 10, we show the effectiveness of our approach for low-frame-rate video tracking. The test video sequences are obtained from the recorded sequences by down-sampling to 3 frames/s. At the beginning of the shot, where the motion was small, the template warping technique yielded good results. However, from frame 70, the template warping method failed to accurately register the body template to the observed images, and the error accumulation resulted in drifts and tracking failures. As shown in Fig. 10, such drifts can be avoided by SIFT correspondences, which are based on matching the image directly to the model and do not suffer from small errors in previous frames. However, the SIFT-based tracker also cannot estimate accurate poses in some frames, e.g., in frames 130 and 390, due to insufficient matched correspondences. Our approach outperforms these two approaches in the video tracking, it prevents error accumulations over time, and therefore estimates poses with higher accuracy.

We also performed experiments to confirm that the proposed approach can be used to track videos with strong perspective effects. In the test video, the card is very close to the camera. Therefore, the perspective effect is severe. The visual tracking results are shown in Fig. 11. Since a perspective camera model is adopted in our approach, we consistently achieved highly accurate tracking for the test sequences—assuming, of course, rich texture for SIFT and good template initialization.

Fig. 12 illustrates the tracking results of a long video sequence (total of 1612 frames). In this sequence, the object experiences all kinds of transformations, arbitrarily large displacements, and view/scale changes. These conditions make the tracking quite challenging. From the visual tracking results, we clearly observe that our approach tracks the target with high accuracy. Note that there is also a large-scale variation in the target relative to the camera. Fig. 13 shows

Fig. 12. Results of pose estimation from a long video sequence (every 100th frame from 100 to 1000, arranged from left to right, top to bottom).

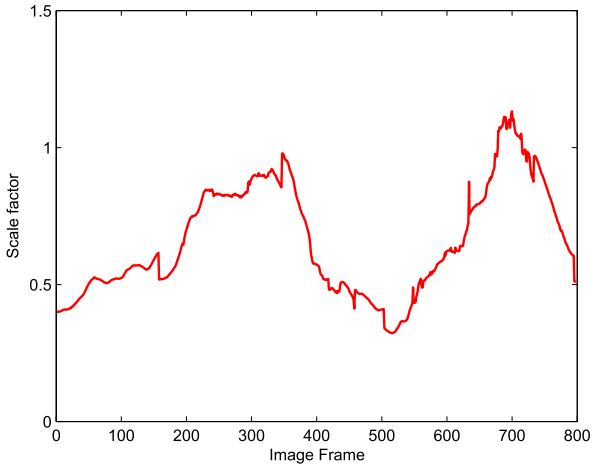


Fig. 13. Results of the estimated scale factor for the first 800 frames.

the variations in the scale factor s in the first 800 frames as the subject moves toward and away from the camera; the trajectory of the estimated scale parameter is mostly smooth.

We also applied the proposed approach to the human motion tracking. A human body is an articulated object with a large number of DoFs, and humans always perform versatile unconstrained motions. Therefore, this is a more challenging problem. In these experiments, we compared the proposed approach with optical-flow-based [26] and SIFT-correspondence-based [24] approaches.

In our method, a person is modeled as a composite of rigid segments. Each segment is connected to one or several other segments, represented as Euler angles. The human skeletal model is configured as a tree structure, in which, the root joint corresponds to the clavicle, chest is approximated as a plane, and other segments are all approximated as cylinders. The model contains the following body parts: head, shoulders, upper arms, and lower arms. Therefore, the human body model has 16 DOFs of dynamic parameters: 2 DOFs for each body part (5×2), without modeling self-rotation of limbs around its axis, 3 DOFs for its global position (translation), and 3 DOFs for its orientation (rotation). In order to effectively obtain the model parameters, we constrain the pose of the body in the template frame to one in which the body stands straight and is parallel to the image plane. Fig. 14 shows the result of initialization.

The experiment in Fig. 15 shows comparisons of a human upper body motion sequence. Although the estimated optical flow is extremely precise, as indicated by the successful

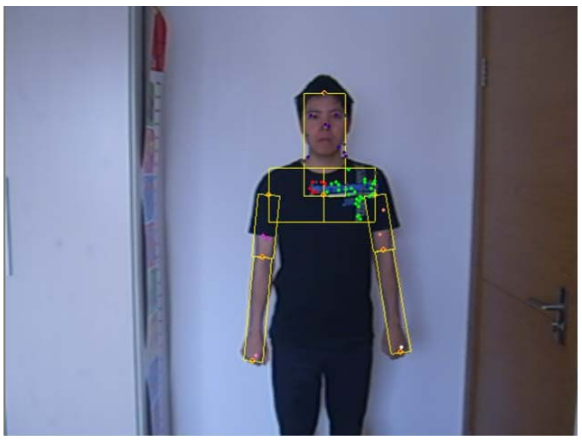


Fig. 14. Result of initialization. The yellow solid circles filled with red indicate the located joints, the filled color circles indicate obtained SIFT features, and features extracted from the same segment have the same color.

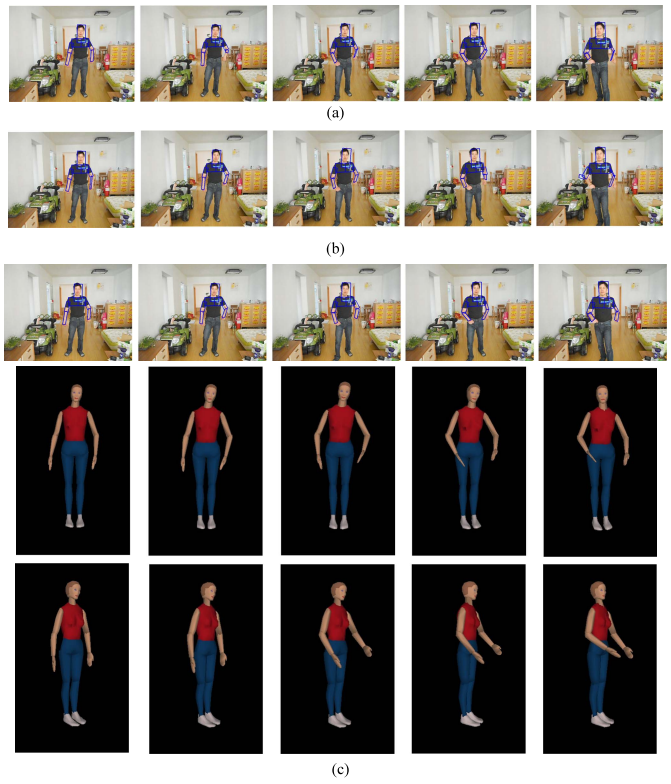


Fig. 15. Comparisons of our proposed approach with the optical-flow-based and SIFT-correspondence-based approaches. Each row shows the estimated 2D results obtained by projecting the reconstructed 3D pose onto the image plane for five frames (20, 40, 60, 80, and 112). (a) Result of tracking using the SIFT-correspondence-based algorithm. (b) Result of tracking using the optical-flow-based algorithm. (c) Result of tracking using our proposed algorithm; the second row shows the estimated pose from the front view, and the third row shows the estimated pose from a different view.

tracking of the torso over 80 frames, it is not robust to fast motions. In frame 112, due to the fast swing of the right hand, the optical-flow-based approach obviously fails to track the hand motion. SIFT features provide useful information about the pose. However, due to their sparsity and the locality of the information, the model is not constrained enough.

TABLE I
AVERAGE PIXEL ERROR OF THREE APPROACHES OVER THE TEST SEQUENCES

Joint	Average pixel error		
	Optical flow based approach	SIFT correspondences based approach	Proposed approach
Head	13.6	19.4	10.3
Right Shoulder	13.5	16.8	9.8
Right elbow	19.7	23.5	16.2
Right wrist	30.3	29.8	21.3
Left Shoulder	14.8	15.1	10.5
Left elbow	21.3	22.6	15.4
Left wrist	26.2	28.3	23.5

The SIFT-correspondence-based approach is unstable in all images. Table I shows the quantitative comparison of the average pixel error for 2D image positions of joints for different trackers. The average pixel error is defined as $[\sum_{i=1}^{60}((x_i - x_i')^2 + (y_i - y_i')^2)^{1/2}]/60$, where (x_i, y_i) are the coordinates of the manually labeled ground truth joints in the ith frame, and (x_i', y_i') are the estimated coordinates of joints in the ith frame. Our proposed approach outperforms both the optical-flow-based and the SIFT-correspondence-based trackers.

B. Quantitative Comparison

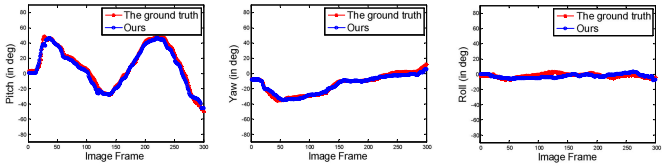
1) Datasets: We evaluated the performance of our approach on three different datasets: the Biwi Kinect Head Pose dataset from [27], the BU dataset from [28], and the McGill Faces dataset from [6]. In our experiments, the 3D model of the target is represented by a cylinder as in [29], the appearance model is established using the method described in Section IV-B.

Biwi Kinect Head Pose dataset contains over 15K images of 20 people (6 females and 14 males—4 people were recorded twice). The head pose range covers about a $\pm 75°$ yaw and a $\pm 60°$ pitch. The subjects were asked to freely turn their head around, trying to span all possible yaw/pitch angles they could perform. The sequences were annotated using the automatic system of www.faceshift.com, i.e., each frame is annotated with the center of the head in 3D and the head rotation angles. All of the sequences and the corresponding ground truth are publicly available at http://www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html.

BU dataset consists of 45 sequences (nine sequences for each of five subjects) taken under uniform illumination, where the subjects perform free head motion including translations and both in-plane and out-of-plane rotations. All the sequences are 200 frames long (approximately 7 s) and contain free head motion of several subjects. Ground truth for these sequences was simultaneously collected via a flock of birds 3D magnetic tracker. The video signal was digitized at 30 frames/s at a resolution of 320×240. All of the sequences and the corresponding ground truth are publicly available at ftp://csr.bu.edu/headtracking/.

McGill Faces dataset consists of 60 unconstrained (real-world) videos from 60 unique subjects and the corre-

Fig. 16. Tracking results of a test video sequence from Biwi Kinect Head Pose dataset (frames 50, 100, 150, 200, 250, and 300, arranged from left to right and top to bottom).



(a) (b) (c)

Fig. 17. Comparison between the estimated pose and the ground truth. Left image shows the pitch comparison, middle image shows the yaw comparison, and right image shows the roll comparison. (a) Pitch comparison of the estimated rotation parameters with the ground truth. (b) Yaw comparison of the estimated rotation parameters with the ground truth. (c) Roll comparison of the estimated rotation parameters with the ground truth.

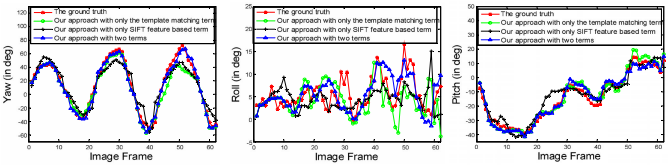
sponding labels (face location, gender, and head pose) for each video frame. For each subject, a 60-s video with 30 frames/s (1800 frames per subject) at a 640 × 480 resolution was recorded using a Canon PowerShot SD770 camera. The face scale changes (on average from 113 × 104 to 222 × 236) not only from one video to another, but also within the same video sequence. Each video sequence is shot under different illumination and background conditions. Furthermore, the subjects are free to move as they wish, resulting in arbitrary face scales, expressions, viewpoints, and local and/or global occlusions. All of the sequences and the corresponding ground truth can be found at http://www.cim.mcgill.ca/~rfvdb.

2) Results:

a) Results on the Biwi Kinect Head Pose dataset: In this experiment, we compare the performance of our method with an approach proposed in [27], which uses an exponential weighting scheme with λ set to 5, where the parameter λ specifies the steepness of the change. Since our approach is a rigid object tracking method, it is not suitable for nonrigid object tracking. In the Biwi Kinect Head Pose database, nonrigid deformations occur in some sequences, so the statistical results do not take into account the tracking results of those frames. The discarded frames are: frames 667 to 746 in sequence 4, frames 429 to 451 in sequence 14, frames 501 to 701 in sequence 15, frames 670 to 762 in sequence 16, frames 209 to 298 in sequence 17, and frames 157 to 371 in sequence 19. Fig. 16 shows some examples of the visual tracking results using our approach for one of the test sequences, and Fig. 17 shows the quantitative comparison of the estimated rotation parameters with the ground truth. Our proposed method

TABLE II
MEAN AND STANDARD DEVIATION STATISTICS OF POSE ESTIMATION, TOGETHER WITH AVERAGE RUNTIME

Technique	Pitch	Yaw	Roll	Time(ms)
The approach proposed by Gabriele Fanelli et al.	3.5±5.8	3.8±6.5	5.4±6.0	44.7
Our approach	3.2±5.5	3.8±6.1	4.6±4.7	3726



(a) (b) (c)

Fig. 18. Comparison of tracking rotational angles of three approaches with the ground truth. The comparison starts from frame 55, and frames in the graph are at the rate of every 10 frames relative to the original frame rate. The left image shows the yaw comparison, the middle image shows the roll comparison, and the right image shows the pitch comparison. (a) Comparison of the estimated yaw of three approaches with the ground truth. (b) Comparison of the estimated roll of three approaches with the ground truth. (c) Comparison of the estimated pitch of three approaches with the ground truth.

consistently achieves a higher tracking accuracy throughout the test sequences. The statistics of tracking results are shown in Table II. Although the computation cost of our approach is higher than the approach proposed in [27], our method achieves a more robust pose-tracking performance.

We also conducted an experiment on sequence 3 in Biwi Kinect Head Pose database to compare our approach with the two cases where only the template-matching term and the SIFT-feature-based term are used in our cost function. The video is down-sampled to one-tenth of the original frame rate. The comparison done after ds estimation, i.e., we use first several frames in the sequence to estimate ds, and then apply each approach, respectively, to track the target. In this experiment, it takes 54 frames to estimate ds, because there is very little motion at the very beginning. Fig. 18 shows the quantitative comparison of the estimated angles with the ground truth. The estimated results of only adopting the template-matching term is extremely precise, as indicated by the successful tracking over 40 frames. However, drift occurs from frame 44 and it fails as the motion between frames 43 and 44 is larger than the tracked structure itself. Fig. 19 illustrates this case. The SIFT-feature-based approach works well all through the sequence tracking; however, the estimated angles are not very accurate. This is attributed to the inaccurate matching of SIFT correspondences or insufficient matching SIFT pairs. Our approach bootstraps tracking using SIFT-feature-based term, then, uses the template-matching term for the final tracking. This combination enables tracking of fast movements with higher accuracy. As shown in Fig. 18, the system with the combined cues is close to the ground truth even when the frame rate is small.

b) Results on the BU dataset: In these experiments, we compared our method with three other tracking methods: 1) Xiao 03 [29]; 2) Cascia 00 [28]; and 3) Morency 08 [14]. To test the sensitivity of our method to the number of templates adopted, the experimental results for our approach with

Fig. 19. Tracked results of three approaches on frames 43 and 44. Left: estimated pose from frame 43. Right: estimated pose from frame 44. (a) Our approach with only the template-matching term. (b) Our approach with only the SIFT-feature-based term. (c) Our approach with two terms.

TABLE III
COMPARISON OF OUR RESULTS WITH THREE TRACKERS. THE VALUES IN THE TABLE ARE THE AVERAGE EUCLIDEAN ANGULAR ERRORS

Technique	Pitch	Yaw	Roll
Xiao 03	3.2	3.8	1.4
Cascia 00	-	-	-
Morency 08	3.67	4.97	2.91
Our approach with 3 templates	3.15	3.58	2.03
Our approach with 6 templates	3.08	3.25	1.86
Our approach with 10 templates	3.07	3.26	1.85
Our approach with 20 templates	3.07	3.25	1.85

Fig. 20. Tracking results of a test video sequence from BU dataset.

3, 6, 10, and 20 templates are reported. Fig. 20 shows some visual tracking results. Our approach successfully tracked all 45 video sequences without losing track at any point, while La Cascia *et al*. [28] reported an average percentage of tracked frames of only ∼75%. Table III shows the accuracy of different trackers, with all the results measured in degrees. Since the intensity of a face point is highly pose dependent and can change quite rapidly, depending on the head movements, all the other three approaches, which use only image registration, fail to accurately register the face template to the image observation in some frames. Our approach takes advantage of SIFT tracker, which is invariant with respect to scaling, image rotation, and moderate lighting changes. Therefore, our proposed method achieves a more robust pose-tracking performance. From Table III, we observe that our approach with six templates is more accurate than with three templates, the gain in accuracy may come from the increased number of templates, which covers a wider range of target appearance changes. However, with further increasing the number of templates, no obvious improvements were noticed. This low correlation is attributed to the smooth changes in the appearance of the object during tracking, making six templates sufficient to get good tracking results. However, it is worth noting that the optimal number of templates is not necessarily six for all videos, since that may depend on video characteristics. The number of templates should be set according to the tradeoff between the desired tracking accuracy and the computing complexity.

c) Results on the McGill Faces dataset: In these experiments, we show that the proposed approach can also be used to estimate head poses in unconstrained environments. The McGill Faces dataset contains occasion, illumination and scale changes. Since the McGill Faces dataset is not collected for motion tracking purposes, the target object in the first frame in the video sequences may not be fronto-parallel to the image plane. Therefore, to initialize the template in our approach, we used a frame that seems mostly fronto-parallel to the image plane. The pose label in each tracked frame is defined as follows: the pose is labeled as $-90° \pm \varepsilon$, $(-90°, -45°)$, $-45° \pm \varepsilon$, $(-45°, -0°)$, $0° \pm \varepsilon$, $(0°, 45°)$, $+45° \pm \varepsilon$, $(+45°, +90°)$, and $+90° \pm \varepsilon$ while the estimated yaw angle is in $(-90°, -70°)$, $(-70°, -50°)$, $(-50°, -30°)$, $(-30°, -10°)$, $(-10°, 10°)$, $(10°, 30°)$, $(30°, 50°)$, $(50°, 70°)$, and $(70°, 90°)$.

Fig. 21. Tracking results of a test video sequence from McGill Faces dataset (frames 63, 71, 85, 90, 96, and 277, arranged from left to right and top to bottom).

We present some challenging cases, and the corresponding tracking results obtained by the proposed approach in Fig. 21. Our approach can estimate the correct labels with a high accuracy of 98% while the accuracy of labeled results reported in [6] were 96.98%. We observe that the false labeling results are usually due to insufficient matched feature correspondences, full occlusion, or nonrigid deformations.

VII. CONCLUSION

In this paper, we suggest a combination of template warping and SIFT-correspondence approach for 3D pose estimation. We demonstrate that the proposed approach can overcome the drawbacks of optical-flow-based and feature-correspondence-based approaches. This approach can be used to track uncalibrated monocular video sequences, which are captured by different cameras because it does not require a prior knowledge of the focal length of the camera. Furthermore, we have demonstrated that the proposed approach can be applied to human motion tracking. Our method has great potential to be used in other applications such as action recognition and human–machine interface.

Although our proposed algorithm has many advantages, it still has one major limitation: the system is initialized manually; this disadvantage makes the approach slightly inconvenient to use. In the future, we will focus on using deep CNNs to detect key poses for template initialization and update, and will study a method to combine good detection results with our proposed tracking method.

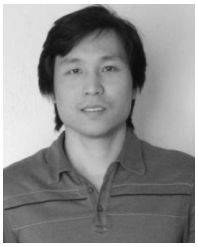
REFERENCES

[1] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: A survey," *Found. Trends Comput. Graph. Vis.*, vol. 1, no. 1, pp. 1–89, 2005.

[2] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1297–1304.
[3] M. Demirkus, D. Precup, J. J. Clark, and T. Arbel, "Probabilistic temporal head pose estimation using a hierarchical graphical model," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 328–344.
[4] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3224–3231.
[5] C. Chen and J. Odobez, "We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1544–1551.
[6] M. Demirkus, J. J. Clark, and T. Arbel, "Robust semi-automatic head pose labeling for real-world face video sequences," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 495–523, May 2014.
[7] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2879–2886.
[8] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3D deformable face tracking with a commodity depth camera," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 229–242.
[9] X. Fan, K. Zheng, Y. Zhou, and S. Wang, "Pose locality constrained representation for 3D human pose reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 174–188.
[10] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D pictorial structures for multiple human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1669–1676.
[11] G. W. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler, "Pose-sensitive embedding by nonlinear NCA regression," in *Advances in Neural Information Processing Systems*. Whistler, BC, Canada: Curran Associates, Oct. 2010, pp. 1–9.
[12] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.
[13] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 498–519, Mar. 2012.
[14] L. P. Morency, J. Whitehill, and J. Movellan, "Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation," in *Proc. 8th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Sep. 2008, pp. 1–8.
[15] K. Pauwels, L. Rubio, J. Diaz, and E. Ros, "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2347–2354.
[16] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3D tracking of rigid and articulated objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 402–415, Mar. 2010.
[17] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1385–1392.
[18] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1385–1391, Oct. 2004.
[19] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2307–2314.
[20] L. Alvarez, J. Weickert, and J. Sanchez, "Reliable estimation of dense optical flow fields with large displacements," *Int. J. Comput. Vis.*, vol. 39, no. 1, pp. 41–56, Aug. 2000.
[21] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.
[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[23] B. Zou, S. Chen, C. Shi, and U. M. Providence, "Automatic reconstruction of 3D human motion pose from uncalibrated monocular video sequences based on markerless human motion tracking," *Pattern Recognit.*, vol. 42, no. 7, pp. 1559–1571, Jul. 2009.
[24] S. Chen, W. Liang, and L. Wu, "Recovering upper-body motion using a reinitialization particle filter," *J. Electron. Imag.*, vol. 22, no. 3, p. 033005, Jul. 2013.
[25] J.-Y. Bouguet, "Pyramidal implementation of the Lucas–Kanade feature tracker," Microprocess. Res. Lab, Intel Corp., Santa Clara, CA, USA, Tech. Rep., 1999.
[26] C. Bregler, J. Malik, and K. Pullen, "Twist based acquisition and tracking of animal and human kinematics," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 179–194, 2004.
[27] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 437–458, Feb. 2013.
[28] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.
[29] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Int. J. Imag. Syst. Technol.*, vol. 13, no. 1, pp. 85–94, Sep. 2003.

Shu Chen received the Ph.D. degree from Central South University, Changsha, China.

He is an Associate Professor with the School of Information Engineering, Xiangtan University, Xiangtan, China. His research interests include computer vision, human motion tracking and recognition, and animation.



Luming Liang received the B.Sc. and M.Sc. degrees from the School of Information Science and Engineering, Central South University, Changsha, China, in 2005 and 2008, respectively, and the Ph.D. from the Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO, USA, in 2014.

He is a Scientist with Microsoft, Redmond, WA, USA. His research interests include finding shape correspondences.

Wenzhang Liang received the B.S. degree in metal material from Guangxi University, Nanning, China.

He is an Engineer with Guangxi Cast Animation Limited Company, Nanning. His research interests include pattern recognition and computer vision.

Hassan Foroosh (M'02–SM'03) received the M.S. and Ph.D. degrees in computer science, specializing in computer vision and image processing, from INRIA, Sophia Antipolis, France, in 1993, and 1996, respectively.

He currently a Professor with the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL, USA. He has authored or co-authored over 100 peer-reviewed journal and conference papers, and has been in the Organizing and the Technical Committee of many international conferences.

Dr. Foroosh was a recipient of the Pierro Zamperoni Award from the International Association for Pattern Recognition in 2004. He also received the Best Scientific Paper Award at the International Conference on Pattern Recognition of the International Association for Pattern Recognition in 2008. He was an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING from 2003 to 2008. He has also been an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING since 2011. His research has been sponsored by National Aeronautics and Space Administration, National Science Foundation, Navy, Office of Naval Research, and industry.