Robotica (2016) volume 34, pp. 1923–1947. © Cambridge University Press 2014
doi:10.1017/S0263574714002665

# Scale-invariant optical flow in tracking using a pan-tilt-zoom camera

Salam Dhou and Yuichi Motai*

*Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284-3068, USA*

(Accepted October 29, 2014. First published online: December 9, 2014)

**SUMMARY**

An efficient method for tracking a target using a single Pan-Tilt-Zoom (PTZ) camera is proposed. The proposed Scale-Invariant Optical Flow (SIOF) method estimates the motion of the target and rotates the camera accordingly to keep the target at the center of the image. Also, SIOF estimates the scale of the target and changes the focal length relatively to adjust the Field of View (FoV) and keep the target appear in the same size in all captured frames. SIOF is a feature-based tracking method. Feature points used are extracted and tracked using Optical Flow (OF) and Scale-Invariant Feature Transform (SIFT). They are combined in groups and used to achieve robust tracking. The feature points in these groups are used within a twist model to recover the 3D free motion of the target. The merits of this proposed method are (i) building an efficient scale-invariant tracking method that tracks the target and keep it in the FoV of the camera with the same size, and (ii) using tracking with prediction and correction to speed up the PTZ control and achieve smooth camera control. Experimental results were performed on online video streams and validated the efficiency of the proposed method SIOF, comparing with OF, SIFT, and other tracking methods. The proposed SIOF has around 36% less average tracking error and around 70% less tracking overshoot than OF.

KEYWORDS: Object tracking; Optical flow; Scale-invariant feature transform; Pan-tilt-zoom.

## Definitions of Symbols

| | |
|---|---|
| $p_{c,f}$ | The center OF feature point at frame $f$. |
| $G$ | Symmetric gradient matrix. |
| $g$ | Local image intensity gradient vector. |
| $I$, $J$ | Image intensity maps. |
| $w$ | Radius of the circle window around KLT feature point. |
| $\varepsilon$ | Cost function minimizes the sum of the squares of intensity difference. |
| $I_x$, $I_y$, and $I_t$ | Derivatives of image intensity values along the $x$, $y$, and $t$ dimensions respectively. |
| $\delta$ | Displacement of a 2D point. |
| $\beta$ | Regularization constant, with which large values lead to a smoother flow. |
| $V = [u, v]^T$ | Optical flow vector. |
| $q_{c,f}$ | SIFT feature point at frame $f$. |
| $G(u, v, c, \sigma)$ | Gaussian function of image $i$, $c_i$ is a coefficient selected to get a fixed number of convolved images per octave, $\sigma$ is the scale. |
| $D(x, y, \sigma)$ | Difference of Gaussian. |
| $L(u, v, c_i, \sigma)$ | Gaussian-blurred images with coefficient $c_i\sigma$. |
| $\theta(x, y)$ | Orientation. |
| $T(P)$ | Transformation matrix. |
| $R$ | Rotation matrix. |

* Corresponding author E-mail: ymotai@vcu.edu

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

| | |
|---|---|
| $P$ | 3D points. |
| $t_f$ | Translation vector. |
| $\hat{\xi}$ | Motion twist. |
| S | 3D rotation vector. |
| $\Phi$ | 3D translation vector. |
| $E = \exp(\hat{\xi})$ | Exponential function. |
| $(x, y)_{OF}$ | Feature point tracked with OF. |
| $(x, y)_{\text{SIFT}}$ | Feature point tracked with SIFT. |
| $M_f$ | Motion matrix at frame $f$. |
| $r(f)$ | Vector from the origin of the target frame to the point $p_c$ at frame $f$. |
| $p_c(f)$ | 2D center point at frame $f$. |
| $w_v$ and $w_p$ | Weights. |
| $\sigma_{f,p}$ | Scale of a keypoint $p$ at frame $f$. |
| $\Gamma_f$ | Vector of the scales of keypoints at frame $f$. |
| $\overline{\Gamma}$ | Average of the ratio of scales between frames $f-1$ and $f$. |
| $N_P$ | Total number of matched keypoints. |
| $\lambda_f$ | Focal length at frame $f$. |
| $X_f$ | State of the system at frame $f$. |
| $\bar{X_f}$ | Predicted state of the system at frame $f$. |
| E | Transition matrix that describes the expected relationship between the current state and the predicted state. |
| $\theta^-_{x,f}$ | Predicted tilt at frame $f$. |
| $\theta^-_{y,f}$ | Predicted pan at frame $f$. |
| $\bar{\lambda_f}$ | Predicted focal length at frame $f$. |
| $\Theta_{f,\text{ comp}}$ | Computed value using SIOF. |
| $\Theta_{f,\text{ meas}}$ | Measured value using Polhemus motion estimator. |
| $\gamma$ | Approximated marginal value |
| $\alpha$ and $G_i$ | Intermediate parameters to calculate the marginal value $\gamma$. |

## 1. Introduction

Real-time tracking using a single Pan-Tilt-Zoom (PTZ) camera has many advantages in the field of computer vision. Compared with traditional tracking with a fixed camera, PTZ camera-based tracking has the potential to (i) increase the Field of View (FoV) by pan-tilt rotating and focal length adjustment, and (ii) follow the target's uncontrolled motion in dynamic environments. To achieve these improvements, this paper proposes a tracking method, Scale-Invariant Optical Flow (SIOF) that uses highly descriptive groups of feature points in a twists model to recover the motion and scale of the target. The camera then rotates to track the target's motion and adjust the focal length to preserve its size.

There are many methods conducted to improve the scale-invariant property using feature tracking. Scale-Invariant Feature Transform (SIFT) is one of the efficient methods for scale-invariant feature tracking using highly distinctive features points.[1] The methods[2,3] use tracking and matching using SIFT feature points to align an image to its nearest neighbors in a large image dataset. Size-preserving tracking methods using the existing structure-from-motion (SFM)[4] successfully achieve a scale-invariant tracking. However, a big need arises to develop a single tracking system that can handle dynamic motion scenarios and track larger transformations of the object movement. In our method, we use two different feature trackers to take advantage of the matching criterion for each of them. Also, we depend on a twist module that uses 3D feature points to generate the 6 degrees of freedom required to control the camera motion.

The proposed tracking method establishes point correspondences of Optical Flow (OF) and SIFT and recovers the 3D position, scale, and orientation of the target at every captured frame. Using groups of different kinds of feature points allows us to obtain a semi-region-based tracking, which is lightweight and can handle robust image transformations. We also use Kalman filter model to predict the scale and motion of the target in the next frame and speed up the PTZ process. The contribution

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

of this paper is twofold: First, we propose a scale-invariant tracking method that depends on an advanced feature tracking criterion. Second, we use prediction and correction model to speed up the PTZ control. Our overall proposed flow works as follows: The user selects a target area on the screen using a pointing device. Feature points are detected on the selected area. The camera tracks the target area to keep it in the FoV of the camera by rotating the camera in pan and tilt angles. The proposed method adjusts the focal length of the camera to keep the target appearing with the same size.

The remainder of this paper is organized as follows. Section 2 presents a relevant study to our topic. Section 3 shows the motion detection techniques discussed: OF and SIFT. Section 4 presents the proposed tracking method and its use to control the camera's PTZ operations. Section 5 presents the method's evaluation criteria. The proposed method's ability and performance is tested and compared with other methods in Section 6. Section 7 concludes the paper.

## 2. Relevant Study

We review some background studies related to our study. In Section 2.1, target tracking is reviewed. Section 2.2 reviews the scale-invariant tracking.

### *2.1. Target tracking*

Tracking of objects to keep them in the FoV with an optimal quality is an important problem.[5,6] Targets, such as human body,[1,7–10] can be tracked by a PTZ camera using different approaches, such as using normal Kalman filter[8] or performing segmentation of main target from the background.[11] Other approaches used for target tracking are based on extended Kalman filter framework,[10,12] quaternion Kalman filter,[13] or Kalman filters through stereo vision.[14] Optical flow is the distribution of apparent velocities of brightness patterns' motion in an image.[15] One of the important methods used for calculating OF is the Kanade–Lucas–Tomasi method (KLT).[16–18] KLT uses sparse feature point and does not involve much computational cost, so it is considered good for online tracking environments. H-Probabilistic Multi-Hypothesis Tracking (PMHT)[19] derives a stable tracking algorithm that uses the entire image as its input data avoiding peak picking and other data compression steps required to produce traditional point measurements. It links a histogram interpretation of the intensity data to the tracking method of PMHT. Mean Shift[20] is used in tracking as in refs. [21–23], where it is used to find the most probable target position in the current frame. Camshift is a method that combines the basic Mean Shift algorithm with an adaptive region-sizing step. It is a video face-tracking method and used in a perceptual user interface as in ref. [24]. Using multiple tracking concepts has been introduced in several studies to compensate for the shortcomings of each other.[25,26] In ref. [25], region fitting has been combined with dense OF and SIFT. In ref. [26], groups of low-level features, such as interest points, edges, and homogeneous and textured regions, are combined on a flexible and opportunistic basis to sufficiently characterize an object and allow robust tracking.

Pan-tilt-zoom cameras are used in many applications. In ref. [27], a PTZ camera is used with an omnidirectional imaging device for sensory tracking system to detect and track any moving object within its 360° FoV. In ref. [28], an adaptive fuzzy particle filter (AFPF) method is adapted to use for object tracking with a PTZ camera. In ref. [29], a constant-time image alignment algorithm based on spherical projection and projection-invariant selective sampling using a PTZ camera is presented. In ref. [30], video rate methods are used for zoom control during tracking to preserve the size of tracked objects. In ref. [31], low-level reactive method that appears as part of repertoire of a skilled camera operator is used based on probabilistic reasoning to minimize the chance of losing the target while maximizing zoom level at the same time. Visual servo control methodology[32,33] is used as a tracking method where multiple objects are kept in the FoV of a PTZ camera.[34] This is done using a set of task functions developed to regulate the mean and variance of image features and achieve the desired feature point velocities. In ref. [35], multiple PTZ cameras are used to infer relative positioning and orientation. Table I summarizes the tracking methods related and compared in this study.

### *2.2. Scale-invariant tracking*

A large amount of studies have been done to improve the scale-invariant property using feature tracking. SIFT is one of the efficient methods for scale-invariant feature tracking using a range of detectors.[1] SIFT relies on a Histogram of Gradients (HoG) to detect its highly distinctive features. Scale-invariant features were used for vision-based mobile robot localization and mapping.[36]

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table I. Comparison of tracking methods.

| Methods | Procedure | Advantages | Disadvantages |
|---|---|---|---|
| OF[15–18] | The distribution of apparent velocities of brightness patterns' motion in an image. | Light weight, fast, and large number of feature points. | Not applicable for large displacement, large image transformation, and scale change. |
| SIFT[1] | Range of detectors SIFT relies on a Histogram of Gradients to detect its highly distinctive features. | Highly distinctive features, accurate match, and scale invariant. | Slow, high computations, descriptors are computed at alternative locations. |
| SURF[41] | Uses sums of 2D Haar wavelet responses and makes an efficient use of integral images. | Highly dtinctive features, accurate match, scale invariant, and faster and robust to image transformation. | Descriptors are computed at alternative locations, which unlikely coincide with the features used for tracking. |
| H-PMHT[19] | Links a histogram interpretation of the intensity data with the tracking method of probabilistic multi-hypothesis tracking. | Stable and multi-target. | Not applicable for large displacement, large image transformation. and scale change. |
| Proposed SIOF | Establishes SIOF point correspondences and recovers the 3D position, scale, and orientation of the target at every frame. | Highly distinctive features, accurate match, scale invariant. | Not suitable for multitarget tracking. |

SIFT-flow[2,3] is a tracking and matching method which uses SIFT features to align an image to its nearest neighbors in a large image corpus containing a variety of scenes. Some tracking methods[37,38] combined SIFT and mean shift for tracking.

Several studies are conducted to improve SIFT since it is computationally expensive and the perfect scale invariance is difficult to be achieved in practice because of sampling artifacts, noise, and other real-time challenges. A slight variation of the descriptor employing an irregular histogram grid has been proposed to reduce the negative effect of scale error[39] and increase feature-matching precision. To get a better performance, authors in ref. [40] proposed a scale-invariant tracking method using strong corner points in scale domain to track a smaller object in a better performance than SIFT. Similar to SIFT, Speeded-Up Robust Features (SURF)[41] is another scale- and rotation-invariant feature tracking method that relies on the sums of 2D Haar wavelet responses and makes an efficient use of integral images.

Different target-tracking and size-preserving techniques are proposed in the literature. In refs. [42, 43], the tracking method used depends on 3D tracking using twists and exponential maps. In ref. [25], twists and exponential maps are used along with a combined region and feature-based 3D tracking. In ref. [44], joint point feature correspondences and object appearance similarity are used. Yao *et al.*[4] proposed a size-preserving tracking algorithm by applying the existing SFM method based on the paraperspective projection model to achieve target scale estimation. The scale property of SIFT feature points has been used in several applications such as visual homing. In ref. [23], a mobile robot moves to a home position using SIFT keypoints extracted from omnidirectional images and matched to a recorded set of keypoints at the Home location. Churchill and Vardy[45] present results of real-time homing experiment using the scale difference field in panoramic images computed from SIFT matches.

## 3. Motion Detection

We present the motion detection strategies used in our proposed method to detect the target motion. Optical flow is described in Section 3.1 and SIFT is described in Section 3.2. In Section 3.3, the interconnection between OF and SIFT as SIOF is discussed.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

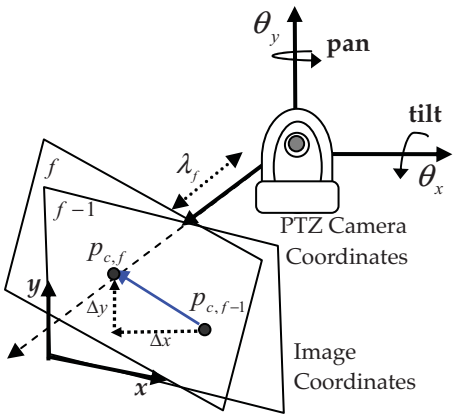



Fig. 1. The geometric relation between frames *f-1* and *f*. The point $p_{c,f-1}$ in *f-1* moved to a new location $p_{c,f}$ in *f*. This movement causes a displacement of $(\Delta x, \Delta y)$. The tracking method controls the PTZ camera to rotate in angles $\theta_x$ and $\theta_y$ to accommodate the target movement.

### *3.1. Optical flow*

The KLT method[16,17] is used to track feature points by finding their correspondences in a sequence of frames. The KLT method finds image features at locations where the minimum eigenvalue of the $2 \times 2$ symmetric matrix $G$,

$$G = \int_w gg^T dp_{c,f},$$

is above some threshold, where $g$ is the local image intensity gradient $2 \times 1$ vector: $g = [\partial I/\partial x, \partial I/\partial y]^T$, and $w$ is the radius of the circle window around the KLT feature point; $p_{c,f}$ is the center point at frame *f*. Image locations which satisfy the above criterion are called "corners" and are easily tracked. KLT finds feature correspondences by finding the displacement $\delta(\Delta x, \Delta y)$ that minimizes the sum of the squares of the intensity in the following cost function:

$$\varepsilon = \int_w [I_f(p_{c,f} - \delta) - I_{f-1}(p_{c,f-1})]^2 dp_{c,f}, \tag{1}$$

where $I_{f-1}$ and $I_f$ are two image intensity maps that are adjacent in time for point $p_{c,f-1}$ and $p_{c,f}$ respectively. Figure 1 shows the tracking process of point $p_c$ in two frames.

Smoothness in the flow is assumed over the whole image. The flow is formulated as a global energy function that is then sought to be minimized. This function is given for 2D image streams as:

$$E = \iint_w [(I_x u + I_y v + I_t) + \beta^2(\|\nabla u\|^2 + \|\nabla v\|^2)]dxdy, \tag{2}$$

where $I_x$, $I_y$, and $I_t$ are the derivatives of the image intensity values along the $x$, $y$, and $t$ dimensions respectively. $V = [u,v]^T$ is the OF vector. The parameter $\beta$ is a regularization constant. Large values of $\beta$ lead to a smoother flow.

### *3.2. Scale-invariant feature transform*

Scale-invariant feature transform is a feature-based approach that is widely used in matching and tracking applications.[1] SIFT features, defied by $q_{c,f}$, are highly invariant to image scaling and rotation, and partially invariant to illumination change and 3D camera viewpoint.

To find keypoints, the image is convolved with variable-scale Gaussian filters, and then the difference of consecutive Gaussian-blurred images is taken. The maxima/minima of the Difference of Gaussians (DoG) at different scales $c_i\sigma$ are keypoints $q_{c,f}$, where $c$ is the point number and $f$ is the frame number. Let a DoG image be as $D(x, y, \sigma)$:

$$D(x, y, \sigma) = L(x, y, c_i\sigma) - L(x, y, c_j\sigma), = G(x, y, c_i\sigma) * I(x, y), \tag{3}$$

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

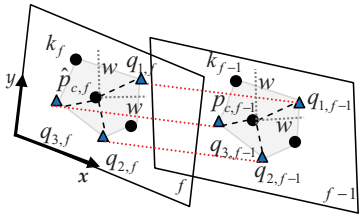



Fig. 2. Feature point tracking using OF and SIFT feature points. Features are tracked from frame *f-1* to frame *f* with OF feature point $p_{c,f}$ and SIFT feature points $q_{p,f}$.

where $*$ is the convolution operation, $I(x,y)$ is the original image, $G(x, y, c_i\sigma)$ is the Gaussian function, $c_i$ is the coefficient selected to get a fixed number of convolved images per octave, and $L(x, y, c_i\sigma)$ is the Gaussian-blurred images with coefficient $c_i$. Each keypoint is assigned with one or more orientations based on local image gradient directions. For an image sample $L(x, y)$ at scale $\sigma$, the orientation $\theta(x, y)$ is:

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right). \tag{4}$$

The descriptor vectors for keypoints with locations, particular scales, and orientations make the keypoints highly distinctive. Eight-bin-Histograms are built and each descriptor contains a $4 \times 4$ array of 16 histograms around the keypoint. This leads to a SIFT feature vector with ($4 \times 4 \times 8 = 128$) elements. This vector is normalized to enhance the invariance to changes in illumination.

### *3.3. Scale-invariant optical flow*

We form higher-level feature points in our proposed SIOF method that uses the properties of two feature points' trackers. Using two feature trackers OF and SIFT helps in handling the individual limitations of these methods. For example, OF using the KLT method cannot handle large displacements and big image transformations such as illumination and scaling changes. SIFT also cannot represent the real pixel location of a keypoint on the image. That is why we describe how OF feature points and SIFT feature points can form SIOF groups that are used for frame-to-frame matching. The following sections show how the characteristics of OF and SIFT feature trackers are fused in the proposed SIOF method.

*3.3.1. Feature point grouping.* The proposed SIOF comprises two features sets of OF and SIFT to form group features for the improvement of frame-to-frame correspondence matching. These group features are fewer in number but more distinct in form.[46] Group features ensure more accuracy by avoiding the mismatches that may happen when using single tracking criterion.

Figure 2 shows OF/SIFT feature points grouping when tracking from frame *f-1* to frame *f*. Optical flow feature points $p_{c,f}$ and SIFT feature points $q_{p,f}$ exist in both frames. Let the OF center point at frame *f-1* be $p_{c,f-1}$, and $p_{c,f}$ is its correspondence at frame *f*. Let $w$ be the radius of a search window around $p_{c,f-1}$. In large displacement cases, the corresponding point $p_{c,f}$ exists out of the search window of radius $w$ at frame *f*. In other words, the displacement $\delta(\Delta x, \Delta y)$ of $p_{c,f-1}$ in the next frame is larger than the radius of the window, $\sqrt{\Delta x^2 + \Delta y^2} > w$. We find the SIFT points $q_{p,f-1}$ that are the nearest neighbors to $p_{c,f-1}$ and reside within the window $w$. If no SIFT feature points reside within $w$, a search is conducted within $2w$ and so on until we find enough SIFT feature points (at least three in this implementation). Thus, $q_{p,f-1}$ will be selected if it maintains the following condition:

$$d(q_{p,f-1}, p_{c,f-1}) < iw, \quad i = 1, 2, ..., s/w, \tag{5}$$

where $d(q_{p,f-1}, p_{c,f-1})$ is the distance in pixels between $q_{p,f-1}$ and $p_{c,f-1}$ and $s$ is the image size. Optical flow and SIFT points within this specified window of radius $iw$ are chosen to be used as one feature group. In our implementation, the radius $w$ is initially set to 10 pixels. SIOF feature groups are denoted by $k$ and are tracked from frame to frame through the frame sequence.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

*3.3.2. Group matching using descriptor.* After each group is formed from features, a representative matching criterion for each group has been established. The matching criterion has a set of attributes that distinguish each feature group from others. For OF feature points in group $k$, the attributes used are the average image intensity maps $\bar{I}_f^k$, as described in (1), and the average smoothness in the flow assumed over group $k$ represented a global energy function $\bar{E}_f^k$, as described in (2). For SIFT feature points, the attributes used are the average orientation histograms with 8 bins in the feature points of group $k$: $\bar{\theta}_f^k$, as described in (4) and the mean RGB color of the same 16 subparts $\bar{\phi}_f^k$ in group $k$. The orientation histograms and 16 RGB color part have been explained extensively in ref. [1]. The OF and SIFT attributes lead to a high-level descriptor vector of each group $k$ at frame $f$ as follows:

$$D_f^k = \left[ \bar{I}_f^k \ \bar{E}_f^k \ \bar{\theta}_f^k \ \bar{\phi}_f^k \right].$$

To match between groups in subsequent frames, we seek a value of energy that minimizes the following overall equation:

$$\begin{aligned} \mathrm{E}\left(\Delta D_f^k\right) = & \int_k \gamma_1 \left| \bar{I}_f^k - \bar{I}_{f-1}^k \right|^2 dk + \int_k \gamma_1 \left| \bar{E}_f^k - \bar{E}_{f-1}^k \right|^2 dk \\ & + \int_k \gamma_2 \sqrt{\left[ \left( \bar{\theta}_f^k - \bar{\theta}_{f-1}^k \right) + \left( \bar{\varphi}_f^k - \bar{\varphi}_{f-1}^k \right) \right]^2} dk, \end{aligned} \tag{6}$$

where $\gamma_1$ and $\gamma_2$ are weights to adjust the significance of the effect of the minimizer of $\bar{I}_f^k$ and $\bar{E}_f^k$ to $\bar{\theta}_f^k$ and $\bar{\varphi}_f^k$. The weights $\gamma_1$ and $\gamma_2$ are calculated based on the desired significance of tracking. If the matching criterion is desired to be based on OF matching criterion more than SIFT criterion (such as the similarity of the average intensity maps), then the associated weight $\gamma 1$ is assigned to a value higher than the value of $\gamma 2$. If the desired matching criterion is SIFT matching criterion, then $\gamma 2$ is assigned to a value higher than $\gamma 1$. In our work, the values of both $\gamma 1$ and $\gamma 2$ were assigned 0.5 to give the same significance as that of the matching criteria. The representative criterion in (6) includes the attributes of OF and SIFT feature points which improves the uniqueness for each feature group $k$.

Scale-invariant optical flow feature groups are tracked through the sequence of frames. The distance measure $d_{f-1,f}^k$ defined in (7) is used to find the displacement of SIOF feature group $k$ between frames *f-1* and *f*. The location of correspondences is computed as minimum $x$ and $y$ coordinates that cause the energy function $\mathrm{E}(\Delta D_f^k)$ to have a minimal value, plus the coordinates of the point on the upper left corner of the grouped SIOF feature point $k$:

$$d_{f-1,f}^k = \arg\min_{x,y} \left\{ \Delta D_f^k \right\} + \min(x_{f-1,p}, y_{f-1,p}). \tag{7}$$

This finds location in the $x$ and $y$ directions of SIOF feature point $k$ when moving from frame *f-1* to frame *f*. The new location of the feature group can be used in locating the individual OF feature point locations in the sequence of frames.

*3.3.3. Individual feature correspondences.* After matching SIOF feature groups, the location of individual OF feature points needs to be estimated and represented by real locations. The average distance between $p_{c,f}$ and the nearby feature points is preserved since these are moving in one SIOF feature group. The corresponding feature point $\hat{p}_{c,f}$ in frame $f$ is estimated as follows:

$$\hat{p}_{c,f} = p_{c,f-1} + d_{f-1,f}^k, \tag{8}$$

where $d_{f-1,f}^k$ is defined in (7). In this case, SIFT feature points, which handle large displacements, can keep OF feature points from being mismatched. These estimated individual OF feature points are used in the next section to find PTZ values in each frame of the sequence. The following algorithm summarizes the process of forming and tracking SIOF feature groups:

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

SIOF algorithm

1. Let $p_{c,f-1}$ be a feature point at frame *f-1*

$$\text{If}\sqrt{\Delta x^2 + \Delta y^2} > w,$$

select $q_{p,f-1}$ that are the nearest neighbors to $p_{c,f-1}$ so that

$$d(q_{p,f-1},\ p_{c,f-1}) < iw, \quad i = 1,\ 2,\ ...,\ s/w \tag{9}$$

2. Form the descriptor vector of SIOF feature group $k$ at frame $f$ as follows:

$$D_f^k = \left[\, \bar{I}_f^k\ \bar{E}_f^k\ \bar{\theta}_f^k\ \bar{\phi}_f^k \,\right].$$

Match between feature groups in subsequent frames by finding a value of energy that minimizes the overall Eq. (6).

3. Estimate the displacement of SIOF feature groups:

$$d_{f-1,f}^k = \arg\min_{x,y} \left\{\Delta D_f^k\right\} + \min(x_{f-1,p}, y_{f-1,p}). \tag{10}$$

Find the real location of the corresponding individual OF feature points:

$$\hat{p}_{c,f} = p_{c,f-1} + d_{f-1,f}^k. \tag{11}$$

## 4. Controlling PTZ Operations Using Proposed Scale-Invariant Optical Flow

We describe the process of controlling pan-tilt-zoom using SIOF tracker. Section 4.1 describes the motion model using SIOF. Section 4.2 describes the PTZ operations controlled by the proposed scale-invariant OF. Section 4.3 describes the prediction of PTZ operations using Kalman filter.

### *4.1. Motion model using scale-invariant optical flow*

In this section, we present the tracking technique that is used to estimate change in the target motion, scale, and position relative to the camera. Twist representation is adapted in this work for motion modeling. This model uses feature correspondences from OF and SIFT as its input and then estimates the rotation and translation of target at every captured frame. Figure 3 shows the feedback loop of PTZ estimation procedure using SIOF.

The 3D rigid body motion can be a transformation matrix:

$$T(P) = RP + t,$$

where $t$ is the 3D translation vector and $R$ is the rotation matrix. We use twist representation to represent rigid body motions[29,30,33] as:

$$\hat{\xi} = \begin{bmatrix} \Omega & \Phi \\ 0_{1\times 3} & 0 \end{bmatrix},$$

where $\Omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$

and

$$\Phi = \left[\, \varphi_1\ \varphi_2\ \varphi_3 \,\right]^T, \tag{12}$$

$\omega_i$ is a 3D unit vector in the direction of the rotation axis and $\phi_i$ determines the translation along the rotation axis. The six parameters $(\phi_1, \phi_2, \phi_3, \omega_1, \omega_2, \omega_3)$ correspond to the 6 degrees of freedom we wish to estimate.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

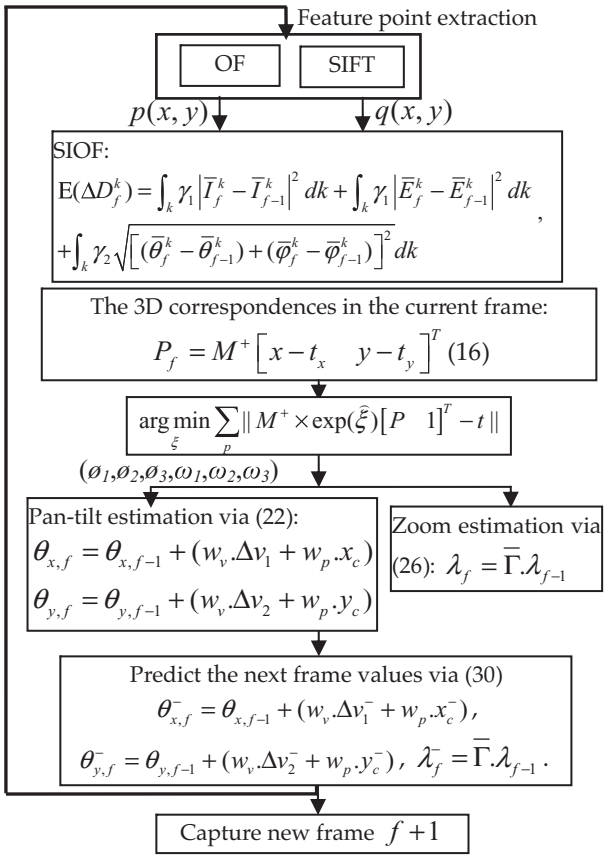



Fig. 3. The feedback loop of PTZ estimation using SIOF starts with feature points extraction in $f$ and detection of 3D correspondences. PTZ estimation and prediction are next steps. Finally, a new frame $f+1$ is captured and the same process is conducted again.

The motion of the points $p_r(x_r, y_r)$ of the region $\Re$ is described by the consecutive evaluation of exponential functions of all involved twists using homogeneous coordinates:

$$\begin{bmatrix} x_r & y_r & z_r & 1 \end{bmatrix}^T = \hat{\xi} \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T . \tag{13}$$

To estimate the parameters in $\hat{\xi}$, we need to have the 2D points and their 3D correspondences that exist on the edges of the region $\Re$. We obtain these correspondences from the feature interpolation discussed in Section 4. The coordinates of 2D points are extracted by OF tracking at every frame. To get the transformed 3D correspondences, we consider the SFM approach.[47] In this approach, 2D points are used to find the corresponding 3D points $P$ and the recovered motion matrix $M$ at every frame $f$. According to SFM, we assume that the following relation holds:

$$p_r = MP + t, \tag{14}$$

where $p_r$ is the coordinate of 2D points representing each feature group, and $M$: $2F \times 3$ is the 3D motion matrix. The motion matrix can be written as $M = [i_1 ... i_F | j_1 ... j_F]$, where $i_f$ and $j_f$ are the camera 3D orientation vectors.[47] The 3D structure of the target can be computed at every frame captured given the new camera coordinates, the horizontal and vertical translation. Using (14), the 3D points $P$ of the target are estimated at frame $f$ as follows:

$$P = M^+(p_r - t) \tag{15}$$

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

where $^{+}$denote the matrix pseudo inverse. As $p_r = [xy]^T$, we subtract the translation in the $x$ and $y$ directions of all feature points, so the 2D coordinates can be written as: $[x_r - t_x y_r - t_y]^T$. Thus, the 3D points $P$ in (15) can be written as follows:

$$P = M^{+} \left[ x_r - t_x \; y_r - t_y \right]^T . \tag{16}$$

This provides the method of recovering the 3D correspondences $P$, given the motion matrix $M$ and coordinates of the feature points. Now we need a transformation T to apply to all points $P$ such that the total distance over all correspondences is minimized in the following least squares notation:

$$\arg\min \left\| M^{+} \times \left[ x_r \; y_r \right]^T - t \right\|_2 , \tag{17}$$

by substituting (16) in (17), we got

$$\arg\min_{\xi} \sum_{p} ||M^{+} \times \exp(\widehat{\xi}) \left[ P \; 1 \right]^T - t||. \tag{18}$$

Equation (18) states a nonlinear least squares problem that finds the transformation T to apply for all points $P$. As stated in (12), $\widehat{\xi}$ has the 6 degrees of freedom $(\phi_1, \phi_2, \phi_3, \omega_1, \omega_2, \omega_3)$ needed for tracking. These six parameters and the highly descriptive SIFT feature points will be used in Section 4.2 to control pan-tilt and zoom of the camera.

### *4.2. Pan-tilt and zoom using SIOF*

Here we propose the method of controlling PTZ operations altogether based on SIOF as discussed in Section 4.1. We present the control of pan-tilt angle in the following Section 4.2.1, and adjust the focal length of the camera using the estimated motion by SIOF model in Section 4.2.2. Section 4.2.3 presents the use of Kalman filter in PTZ prediction.

*4.2.1. Pan-tilt computation.* Here we estimate the camera's pan and tilt angles so that it can rotate horizontally and vertically to track the target. The angle vector $\theta$ is required to make image center coincident with the target centroid.

$$\theta = \left[ \theta_x \; \theta_y \right]^T , \tag{19}$$

where $\theta_x$ is the tilt angle, which is the rotational angle about the $x$-axis and $\theta_y$ is the pan angle, which is the rotational angle about the $y$-axis as illustrated in Fig. 1. Pan and tilt angles are measured based on the location of the center of the image.

Let $p_{c,f}$ denote the center point in frame $f$. The velocity of the center point $p_c$ is the first derivative of the point $p_{c,f}$, which is given by

$$P(f) = \omega \times r(f), \tag{20}$$

where $r(f)$ is the vector from the origin of the target frame to the point $p_c$ at frame $f$,

$$P(f) = \omega \times (p_{c,f}(f) - P_{c,f}), \tag{21}$$

where $P_{c,f}$ is the 3D coordinate with 2D center point at frame $f$ with $z$ as the homogenous scaler. $P_{c,f}$ is the corresponding 3D point as expressed in Fig. 1.

Equation (21) can be written in homogeneous coordinates as described in (13), the resulting equation can be written as follows:

$$\begin{bmatrix} P \\ 0 \end{bmatrix} = \begin{bmatrix} \Omega & \Phi \\ 0_{3\times 1} & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}.$$

We use the recovered translation $\Phi$ and point position $p_{c,f}(x_{c,f}, y_{c,f})$ to determine pan-tilt angle values. The weighted sum of translation and position of the center point at frame $f$ is estimated in the

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

way that the camera movement should respond to the translation of the object and be proportional to the location of center of the target in the 2D frame. So we represent this weighted sum as follows:

$$\theta_{x,f} = \theta_{x,f-1} + (w_v \Delta\varphi_1 + w_p x_c),$$
$$\theta_{y,f} = \theta_{y,f-1} + (w_v \Delta\varphi_2 + w_p y_c), \tag{22}$$

where $\theta_{x,f-1}$ and $\theta_{y,f-1}$ are the pan and tilt angles in the previous frame *f-1* respectively; $\Delta\varphi_1$ and $\Delta\phi_2$ are the horizontal and vertical translations between frame *f-1* and *f* respectively; and $w_v$ and $w_p$are the weights assigned to weighted sum to determine the contribution of the translation $\hat{v}$ and the center point $p_{c,f}(x_{c,f}, y_{c,f})$ in the movement of the camera. These weights are computed experimentally and their summation is 1. Each weight parameter is computed based on the desired contribution of the translation of the object and its location to ensure having the target in the center of the image. Thus, the pan and tilt angles are measured depending on the recovered translation and rotation in the *x*- and *y*-axis directions.

Equation (22) shows the pan and tilt computation by calculating the position and displacement of a moving target over time in every two consecutive frames. The values of the computed pan $\theta_{y,f}$ and tilt $\theta_{x,f}$ are converted into integer form to be used in the command of controlling the camera movement.

*4.2.2. Focal length computation.* The scale property in SIFT keypoints that exist in SIOF feature groups are used to estimate the focal length of the camera. This is achieved by comparing the scales of the keypoints in consequent frames. Let $\sigma_{f-1,p}$ be the scale of a keypoint $p$ in frame *f-1*, and let $\Gamma_{f-1}$ be the vector of the scales of keypoints in frame *f-1* in the sequence:

$$\Gamma_{f-1} = \left[\sigma_{f-1,1}\ \sigma_{f-1,2} \cdots \sigma_{f-1,n}\right]. \tag{23}$$

Similarly, $\Gamma_f$ is a vector of scales in frame *f*:

$$\Gamma_f = \left[\sigma_{f,1}\ \sigma_{f,2} \cdots \sigma_{f,P}\right] \tag{24}$$

when the keypoints in frame *f-1* are matched to their correspondence keypoints in frame *f*, the scales of the keypoints in both frames are compared. We take the average of the ratio of scales for the matched keypoints between frames *f-1* and *f*,

$$\overline{\Gamma} = \left(\sum_{p=1}^{P} \sigma_{f-1,p}/\sigma_{f,p}\right) \Big/ N_P, \tag{25}$$

where $\overline{\Gamma}$ is the average of the ratio of scales between frames *f-1* and *f* and $N_P$ is the total number of matched keypoints in both the frames. When the target is moving closer to the camera, the scales of the matched keypoints in frame *f* have higher values than the scales of the matched keypoints in frame *f-1*, and according to (25), $\overline{\Gamma} < 1$. On the other hand, when the target is moving far away from the camera, the scales of the matched keypoints in frame *f* have lower values than the scales of the matched keypoints in frame *f-1*. Thus, the average ratio of scales will have a value larger than 1 (i.e., $\overline{\Gamma} > 1$). The focal length is adjusted by zooming in or out to keep the object appear with the same size as shown in (26),

$$\lambda_f = \overline{\Gamma}\lambda_{f-1}, \tag{26}$$

where $\lambda_f$ is the focal length of the camera in frame *f*. Thus, (26) finds the focal length of the camera at frame *f* based on the keypoint scales and the focal length of camera at frame *f-1*.

*4.2.3. PTZ prediction using Kalman filter.* The proposed tracking algorithm uses Kalman filter prediction to accelerate the tracking process. The predictions of the next values of pan-tilt and focal length are accomplished altogether at the same time. The current values of pan-tilt and zoom are used in calculating the next predicted values using the Kalman filter.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Because the latency is roughly equivalent to the period of one control cycle, the state of the system should be predicted by a control cycle in advance in order to resolve the latency issue. The state of the system at frame $f$ can be described by the vector $X_f$ given in (27), which consists of the target's position $(x_f, y_f, \lambda_f)$ and displacement $(\Delta x_f, \Delta y_f, \bar{\Gamma}_f)$ in the $x$–$y$ plane,

$$X_f = \left[ x_f \; y_f \; \lambda_f \; \Delta x_f \; \Delta y_f \; \Gamma_f \right]^T. \tag{27}$$

The predicted state $X_f^-$ is defined in (28) as

$$X_f^- = \mathrm{E} X_{f-1} + w_f, \tag{28}$$

where E is the transition matrix that describes the expected relationship between the current and future states. We use the traditional motion model in which the target's velocity is assumed to remain constant, so the future position can be described by the equation $p_f = p_0 + v.dt$. In this scenario, $dt$ is the number of frames seperating consecutive control cycles.

We aim to predict the target image location in the next frame $f$ and the zoom value that keeps the target in the same size. The predicted parameters $(x_f^-, y_f^-, \lambda_f^-)$ can be obtained by providing the values of the parameters $(x_{f-1}, y_{f-1}, \lambda_{f-1})$ to the prediction module as follows:

$$\begin{aligned} x_f^- &= x_{f-h} + u_{O,f-h}\Delta t, \\ y_f^- &= y_{f-1} + v_{O,f-h}\Delta t, \\ \text{and} \quad \lambda_f^- &= \lambda_{f-h}\bar{\Gamma}_{f-h}, \end{aligned} \tag{29}$$

where $h$ is the finite window horizon used for prediction. The predicted values are used to generate the desired camera angle motion $\theta_{x,f}^-$ and $\theta_{y,f}^-$ and focal length $\lambda_f^-$ so that we can calculate the camera parameters for tracking at frame $f$ as follows:

$$\begin{aligned} \theta_{x,f}^- &= \theta_{x,f-h} + (w_v \Delta\phi_1^- + w_p x_c^-), \\ \theta_{y,f}^- &= \theta_{y,f-h} + (w_v \Delta\phi_2^- + w_p y_c^-), \\ \text{and} \quad \lambda_f^- &= \overline{\Gamma}\lambda_{f-h}, \end{aligned} \tag{30}$$

where $(x_c^-, y_c^-)$ is the predicted center of image; and $w_v$ and $w_p$ are as described in (22). The correction phase is accomplished after each predicted state to provide an optimal estimate of the current state based on position and displacement measurements.

## 5. Method Evaluation Criteria

Here we present the evaluation criteria that we used to evaluate SIOF tracking method comparing with other related methods. In Section 5.1, position estimation and tracking errors are presented. In Section 5.2, we present the prediction overshoot analysis, and in Section 5.3, we discuss the computational complexity and storage requirements.

### *5.1. Position estimation and tracking errors*

The position and tracking errors determine the accuracy of estimating the target position, and thus the tracking accuracy. Errors in PTZ estimation are considered tracking errors. These tracking errors are measured quantitatively by computing the root mean square (RMS) errors of the difference between the computed and measured values using the Polhemus Liberty motion estimator.[48]

Equation (31) calculates the absolute difference between the computed and measured values at frame $f$ as $err_f$, and the total RMS error as the total error for all frames $F$: $err_{\text{total}}$:

$$err_f = \sqrt{(\Theta_{f,\text{meas}} - \Theta_{f,\text{meas}})^2},$$

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

$$err_{\text{total}} = \sqrt{\sum_{f=1}^{F} (\Theta_{f,\text{meas}} - \Theta_{f,\text{meas}})^2 \Big/ F}, \tag{31}$$

where $\Theta_{f,\text{comp}}$ corresponds to the values computed using SIOF and $\Theta_{f,\text{meas}}$ corresponds to the measured values. We use (31) to calculate position and PTZ tracking errors.

### *5.2. Tracking overshoot analysis*

We also analyze the tracking overshoot for the predicted values. Overshoot is defined as the cases in which the predicted value exceeds a certain marginal value. We derive marginal values based on the estimate process of the uncertainty point estimators or predictors.[49] We approximate marginal value ($\gamma$) through the total number of frames $F$ as:

$$\gamma = \pm\alpha\sqrt{1 + \left(\sum_{f=1}^{F} G_f G_f^T \Big/ F^2\right)}, \tag{32}$$

where $\alpha$ is the standard deviation estimator, and $G_f$ is the Jacobian matrix of predicted values with respect to inputs. Let the absolute value of the measured parameter be $\theta_f$, and $m = \sum_{f=1}^{F} \theta_f$. Then the values of $\alpha$ and $G_i$ are calculated as follows:

$$\alpha = \sqrt{\sum_{f=1}^{F} (m - \theta_f) \Big/ F},$$

where $G_f = [\sqrt{\theta_1}/\theta_1 \ \ldots \ \sqrt{\theta_F}/\theta_F]$.

The marginal value $\gamma$ is used to define upper and bounds of the ground truth measurement. The bounds are defined by adding or subtracting the marginal value to/from the measured value. It is considered an overshoot when the predicted values of pan-tilt-zoom exceed those of bounds. In the experimental results in Section 6.5, the analysis of the prediction overshoot is presented.

### *5.3. Computational complexity and storage requirement*

The computational and storage requirements of SIOF are dominated by the need to calculate PTZ values at each frame *f*. For a tracking method having $F$ frames and $P$ feature points, the computational complexity and storage requirements are calculated for OF, SIFT feature extraction, and SIOF.

The computational complexity of OF for one frame of size $n \times n$ is[50]]: $O(n^2)$. The computational complexity of SIFT for one frame of size $n \times n$ is[51]]: $O(n^4)$. The computational complexity of SIOF for one frame of size $n \times n$ is the summation of OF and SIFT: $O(n^4 + n^2) = O(n^4)$, given that SIOF uses both OF and SIFT. The computational complexity of OF, SIFT, and SIOF is experimentally measured and discussed in Section 6.7.

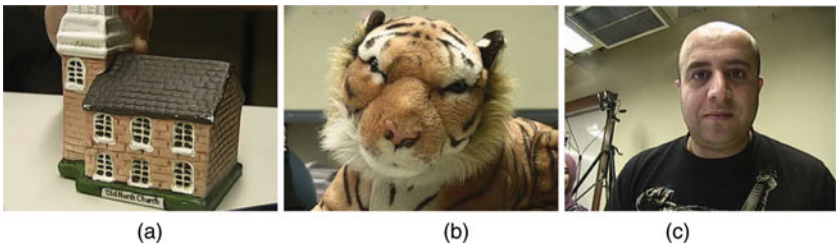
## 6. Experimental Results

We have conducted several experiments using the proposed SIOF on different video streams with different objects moving with different speeds. Section 6.1 shows the scene specifications and motion detection using SIOF described in Section 4.1. Section 6.2 shows the results of the PTZ tracking described in Section 4.2. Section 6.3 shows a comparison of SIOF with OF and SIFT. Section 6.4 shows a tracking comparison of SIOF with OF and SIFT over different prediction intervals. Section 6.5 evaluates the tracking overshoot. Section 6.6 compares SIOF with other tracking methods. Section 6.7 evaluates the computational complexity of SIOF compared with other methods.

### *6.1. Scene specifications and motion detection*

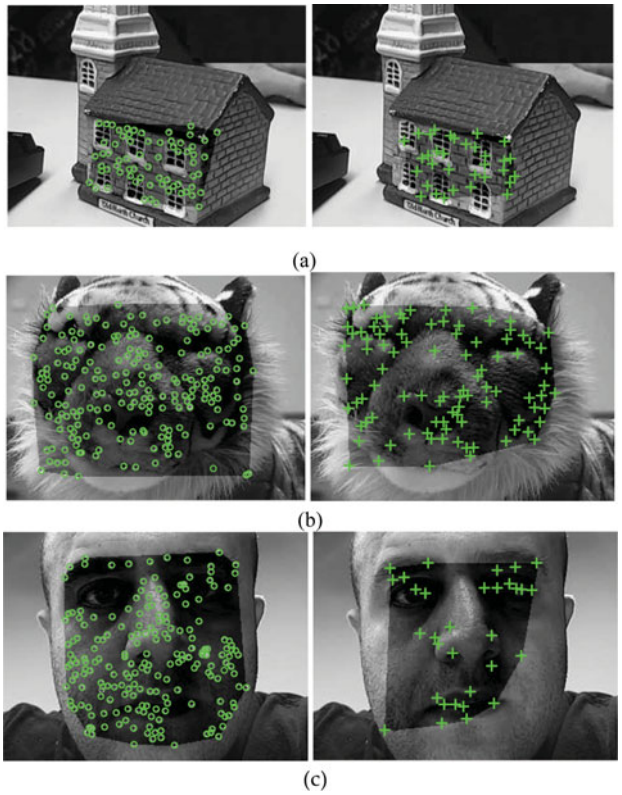
We applied the proposed tracking method on different online image streams captured by the PTZ camera, Canon VB-C60. Those image streams were taken in the same place with the same lighting conditions. Objects in the image streams were moving freely without position or speed constrains. Figure 4 shows a single image from each image stream used. As can be seen, the image streams contain textured objects with different colors.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press



(a) (b) (c)

Fig. 4. Objects used in generating video streams: (a) "Building" as a polyhedron object of nearly flat faces and straight edges, (b) "tiger" as an object of a rich texture, and (c) face as the relatively hardest object in this study.



(a) (b) (c)

Fig. 5. Feature points extraction for image streams: (a) building, (b) tiger, and (c) face. The images on the left show OF feature points, while the images on the right show SIFT feature points. The feature points are surrounded by a gray-shaded mask showing the selected region to track.

Figure 5 shows the feature points extracted on a selected area by user. The grayscale of the images is used to increase the performance of the system. When the user selects a target area on the screen on an object of interest, the keypoints are detected on the selected area. Then the camera tracks the target area selected to keep the object in the FoV of camera. SIOF, which uses groups of OF and SIFT feature points, is applied on the selected area to track the target. The images on the left show the feature points tracked by OF and marked by "o." The images on the right show feature points tracked by SIFT and marked by "+." A shadow mask surrounds the feature points.

Table II shows the properties of the image streams in terms of the number of frames, the number of feature points detected, the frame generation rate, and the frame size. Image streams were taken with different object movement speeds: slow (30–50 mm/sec), moderate (50–70 mm/sec), and fast (70–110 mm/sec). We used a PC of Intel Core 2 Duo 3.06 GHz CPU and 3-GB RAM. The frame rate and image size were the same for all the image streams: 5 frames/sec as the frame rate and 320 × 240 pixels as image size. Each row in the table shows the average specifications of five experiments conducted.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table II. Image streams properties.

| Image streams | Speed | AVG # frames | AVG # features |
|---|---|---|---|
| Building | Slow | 1369 | 166 |
| | Mod | 1150 | 165 |
| | Fast | 1123 | 153 |
| Tiger | Slow | 1631 | 247 |
| | Mod | 1123 | 234 |
| | Fast | 1220 | 231 |
| Face | Slow | 1847 | 154 |
| | Mod | 1221 | 134 |
| | Fast | 1202 | 127 |

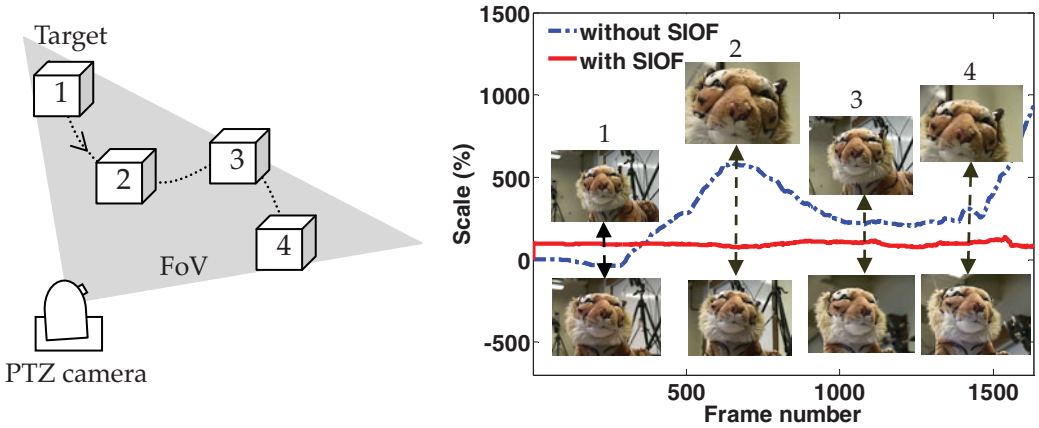



Fig. 6. Tracking outcomes at different time stamps using/without using SIOF: (a) Movement of the target and its position during tracking. (b) Scale of the target using/without using SIOF. Without using SIOF (dashed blue), the scale of the object changes during movement compared with its value in the first frame of the sequence. Using SIOF (red solid), the object is kept with the same size during tracking.

In Table II, the average number of frames for each type of image streams is shown. As seen, the length of the streams varies from one dataset to another. For comparison, if two tracking methods based on two datasets of different lengths are compared, we consider the length of the shorter dataset in the comparison and discard the extra frames of the longer dataset. The proposed method accuracy was tested and compared with other tracking methods by comparing its values with the values measured using the Polhemus Liberty motion estimator.[32] The Polhemus Liberty is an electromagnetic motion tracker that tracks 6 degrees of freedom. It has sensors that are attached to the target to track its motion with respect to the source. The Polhemus Liberty estimates the 3D position and the 3D motion of the target at a frequency of up to 240 Hz. Regarding the average number of feature points, SIOF uses the total number of feature points tracked using both OF and SIFT. All image streams were taken with the same frame rates and frame size. For every tracking method, different datasets were taken with fixing of all parameters such as motion path, speed, and position. Each tracking method is used to estimate PTZ parameters and control the camera by rotating and/or adjusting the focal length accordingly to track the target in real time.

### *6.2. PTZ tracking using SIOF*

Here we show the experimental results of the SIOF tracking method. SIOF lets the camera track the selected target using pan-tilt to keep the target in the FoV of camera. SIOF also controls zoom to adjust the focal length to keep the object appear in the same size. Figure 6 shows one of the "tiger" image streams during tracking using SIOF. The curves show the scale of the target while moving. The scale indicates the size of the object at frame $f$ compared with the size of the object at the first frame. In this experiment, the target is carried and moved manually on a defined path of movement while the camera is stationary. Figure 6(a) describes the path of movement. The numbered cubes indicate the

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

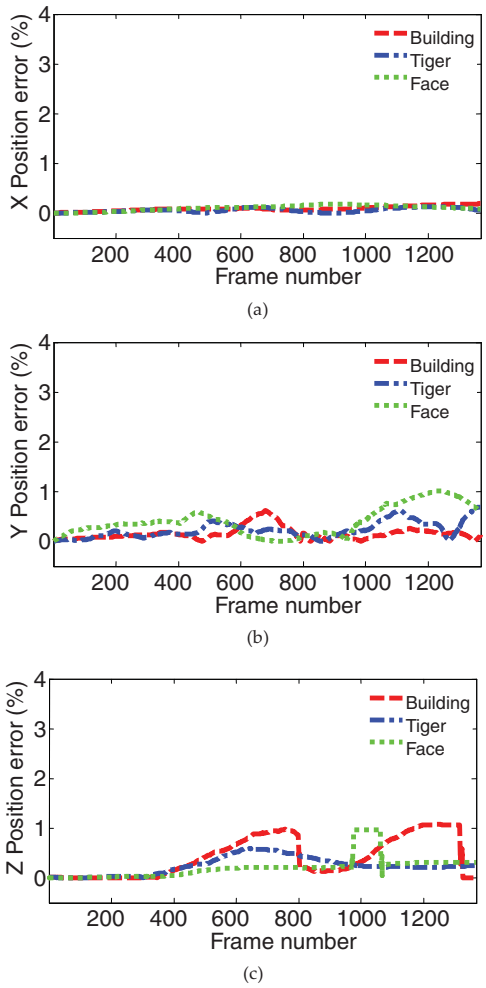



Fig. 7. Position error estimation using SIOF: (a) *X*-position error, (b) *Y*-position error, and (c) *Z*-position error. All errors are presented in percentage (error in the *X*-, *Y*-, or *Z*-direction divided by the total distance of the movement in that direction).

positions of the target during tracking, which correspond to the images of the target in Fig. 6(b). In this setting, two datasets have been taken. One dataset has been taken while the zoom function was inactive and the other dataset has been taken while the zoom function was active. Since the object was carried and moved manually, it is noted that the target has some rotation around its centroid in the second dataset (when the zoom function was active). This caused no effect on the tracking result or comparison.

As shown in Fig. 6(b), the tracking using SIOF controls the zoom operation of the camera. Using SIOF, the system estimates the scale of the target in the first frame, then it adjusts the focal length to keep its size the same throughout the tracking process. The scale of the target using SIOF is around 100% all over the tracking process, which means that the size of the target is kept about the same as its size in the first frame.

Figure 7 shows the error of the target's estimated position. Figure 7(a) shows the error in the *X*-position of the target, Fig. 7(b) shows the error in the *Y*-position of the target, and Fig. 7(c) shows the error in the *Z*-position of the target. These position errors are presented in percentage, which is calculated by dividing the tracking error calculated as in (31) in either *X*, *Y*, or *Z* direction by the distance that the object moved in that direction. The tracking error $err_f$ in (31) is calculated by computing the absolute difference between the computed values and the measured values using the Polhemus Liberty motion estimator. In this experiment, we used a selection of "building," "tiger," and "face" image streams that were taken at a moderate speed.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

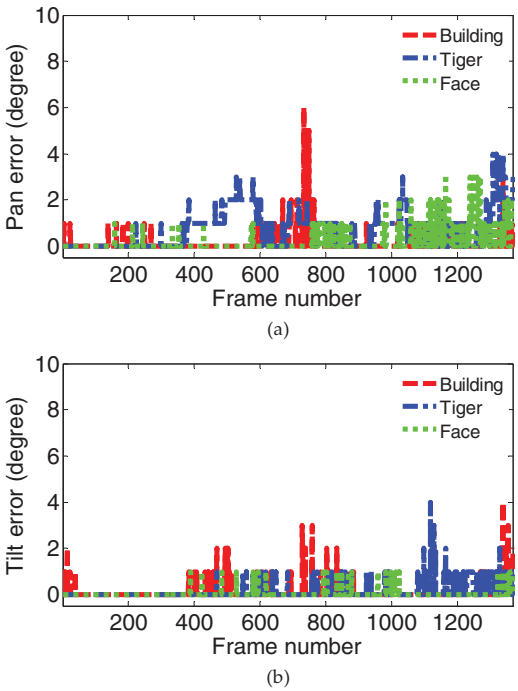



Fig. 8. Tracking errors using SIOF: (a) pan error, and (b) tilt error. Errors are measured in degrees, which is the unit of camera rotation. Pan and tilt angles are measured and rounded to the nearest integer, so when comparing pan-tilt angles between methods, error appears also as an integer.

As shown in Fig. 7, the average error of the estimated position in the *X*- and *Y*-positions of the target do not exceed 1% for *X*- and *Y*-position errors. The maximum error also does not reach 2% for the *Z*-position error. The error of the estimated position using SIOF is in general smaller in the *X*, *Y* movement directions rather than in the *Z* movement direction in all the frames.

Figure 8 shows the tracking errors ($err_f$) of the pan and tilt using SIOF. We use the same datasets and error criterion as used in Fig. 7 to compare between the computed pan-tilt values and the measured ones.

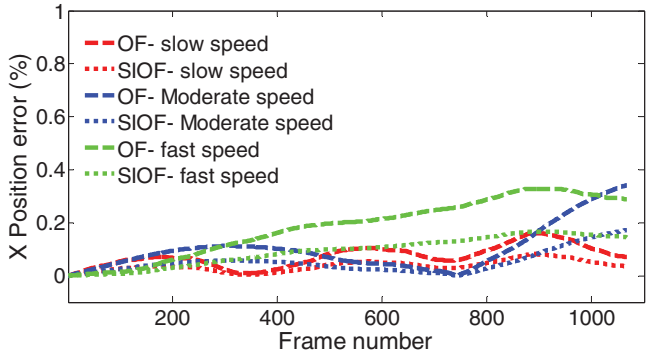
As shown in Fig. 8, the maximum tracking error of the pan and tilt angles is around 4° , except for some few frames in the "building" image stream where it reached 6°. However, the average tracking error did not exceed 2° in all directions. The motion scenarios of all targets was a free motion; it was uncontrolled by a robot or any other device. Such kind of motion has lots of noise and is expected to carry error measurements. The values of the estimated pan/tilt angles by the tracking method are rounded off to the nearest integer. The algorithm increments each of the pan-tilt angles by 1° . The goal of the camera rotation is to keep the object in the center of the FoV of the camera, so when the target displacement shifts the target from the center of the FoV, the camera rotates by the estimated pan-tilt angles. The movement of the camera is still smooth because of the active prediction function which predicts the movement of the target in the next frame window and moves the camera accordingly to overcome the hardware and computing delay, as discussed in Section 4.2.3.

Table III shows the quantitative average error $err_{\text{total}}$ from five conducted experiments of the estimated position in percentage, and pan-tilt error in degree for the three image streams as described in (31). As shown in Table III, the average *X*- and *Y*-position errors in the three image streams are no more than 0.5%. For the *Z*-position, the average error is a bit larger, reaching around 0.6%. Pan-tilt angles have no more than 1° of error in average in the three image streams. This average error may be acceptable in such a dynamic system with freely moving targets which requires the camera to perform concurrent PTZ operations. In general, the SIOF average position error is 0.25% and the average tracking error is 0.44° over the three image streams.

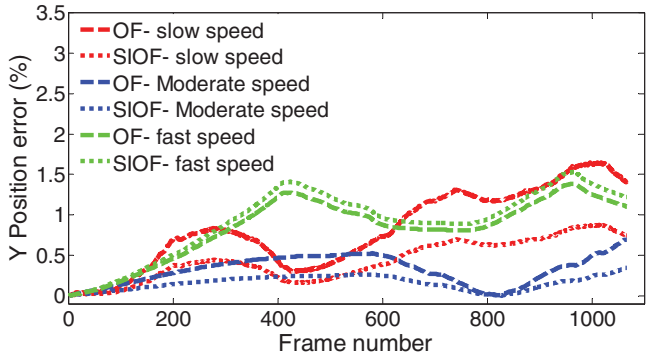
https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table III. Average error ($err_{total}$) of position and tracking operations over the three image streams.

| | | Position error (%) | | | Tracking error (degree) | |
|---|---|---|---|---|---|---|
| Streams | Speed | *X* | *Y* | *Z* | Pan | Tilt |
| Building | Slow | 0.0424 | 0.5037 | 0.280 | 0.393 | 0.358 |
| | Mod | 0.0651 | 0.1804 | 0.285 | 0.663 | 0.972 |
| | Fast | 0.1078 | 1.0163 | 0.356 | 1.021 | 1.750 |
| Tiger | Slow | 0.0564 | 0.6082 | 0.291 | 1.003 | 0.3343 |
| | Mod | 0.0754 | 0.2501 | 0.312 | 1.810 | 0.431 |
| | Fast | 0.2088 | 1.0314 | 0.399 | 1.909 | 0.679 |
| Face | Slow | 0.0490 | 0.5285 | 0.221 | 0.399 | 0.135 |
| | Mod | 0.0721 | 0.1201 | 0.378 | 0.956 | 0.621 |
| | Fast | 0.2451 | 1.0578 | 0.561 | 1.275 | 0.886 |




(a)




(b)

Fig. 9. Position error using SIOF versus OF for: (a) *X*-position error, (b) *Y*-position error, and versus SIFT for (c) *Z*-position error. All errors are presented in percentage (error in the *X*, *Y*, or Z direction divided by the total distance of the movement in that direction).

### *6.3. PTZ tracking using SIOF versus OF and SIFT*

Here we compare the SIOF tracking method with OF and SIFT in terms of the *X*-, *Y*-, and *Z*-position estimation and PTZ tracking accuracy. Figure 9 shows the estimated error in the *X*- and *Y*- position of the target using SIOF versus OF; and the estimated error in the *Z*-position of the target using SIOF versus SIFT. Figure 9(a) shows the *X*-position error, Fig. 9(b) shows the *Y*-position error, and Fig. 9(c) shows the *Z*-position error. We used one of the "building" image streams. The measured position values are also measured using the Polhemus Liberty motion estimator.

As can be seen, Fig. 9 shows that the estimated position error in *X*- and *Y*- position of the target using SIOF versus OF in (a) and (b) was almost the same. The error in the *Y*-direction was larger than the error in the *X*-direction. One of the reasons for this is that the motion in the *Y*-direction was larger than the motion along the *X*-direction. There are many factors that might be the cause of the relatively

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

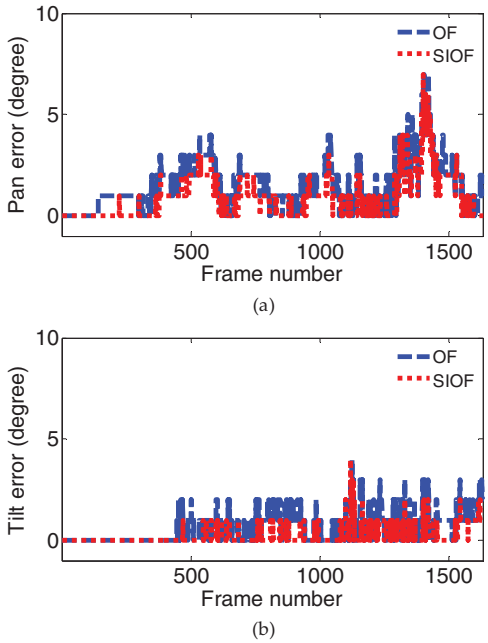



Fig. 10. Tracking error using SIOF versus OF for: (a) pan error, and (b) tilt error. All errors are presented in degrees, which is the angle of pan-tilt movement. Pan-tilt measurement is estimated and rounded off to the nearest integer, as the pan-tilt measurement sent by a command to PTZ camera should be an integer.

large *Y*-error compared with the *X*-error. A calibration error might be one of the factors. Also, since we used an electromagnetic device to measure the ground truth, the tracking measurements might be affected by the electromagnetic field in the work environment. Although we have avoided every possible cause that might cause uncertainty in the device estimation, this is still a possible factor. In Fig. 9, it can be observed that the position error rises along time. It is because the algorithm took the initial frame as the only reference during the trial. Feature points are extracted in the first frame, then they are tracked through the image sequence. The same behavior can be observed in Fig. 7.

Figure 10 shows the estimated tracking error in pan-tilt and using SIOF compared with OF using one of the "tiger" image streams. Figure 10(a) shows the pan error using SIOF versus OF, Fig. 10(b) shows the tilt angle error using SIOF versus OF.

As can be seen in Fig. 10, the estimated pan and tilt angle errors using SIOF are less than the pan and tilt angle errors using OF. The error is high in OF because the estimation of *X*- and *Y*-movement was not associated with zoom movement. Also, integer numbers for pan and tilt are used to control the pan-tilt controls of the camera. Rounding the pan-tilt angles sometimes makes a difference between the computed (OF and SIOF) and true measurements.

### *6.4. Tracking comparison over the prediction intervals using OF, SIFT, and SIOF*

Here we apply the tracking method on different prediction intervals using Kalman filter. There exist several time delay factors for PTZ command–execution such as rotation speed, re-focus time, and stabilization time. Thus, we set the three representative horizon time windows of 100, 143, and 200 milliseconds (ms). For the next PTZ measurement, the horizontal time window specified is expected to cover these mechanical displacement delays. We compare our proposed method SIOF with OF and SIFT in terms of PTZ errors using these different horizon time windows, using (31) mentioned in Section 5.1.

Table IV shows the average error $err_{\text{total}}$ in the target's position and pan-tilt-zoom compared with the measured value across three prediction intervals for OF, SIFT, and SIOF. We used the average values from "tiger" image streams for this study. To evaluate the position and tracking errors quantitatively, the RMS error across all frames ($err_{\text{total}}$) is computed as described in (31). As shown, the error values for the *X*–*Y* position decrease when we use a longer prediction interval in OF, SIFT, and SIOF. Regarding the position estimation error, SIOF has 21% smaller error value when the

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table IV. Average error ($err_{\text{total}}$) of pan, tilt, and zoom across three horizon windows of finite prediction intervals.

| | 100 (ms) | | | | | 143 (ms) | | | | | 200 (ms) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Position error (%) | | | Track error (degree) | | Position error (%) | | | Track error (degree) | | Position error (%) | | | Track error (degree) | |
| | *X* | *Y* | *Z* | Pan | Tilt | *X* | *Y* | *Z* | Pan | Tilt | *X* | *Y* | *Z* | Pan | Tilt |
| OF | 0.14 | 0.36 | NA | 1.97 | 0.92 | 0.10 | 0.33 | NA | 1.93 | 0.89 | 0.07 | 0.29 | NA | 1.89 | 0.86 |
| SIFT | NA | NA | 0.43 | NA | NA | NA | NA | 0.38 | NA | NA | NA | NA | 0.35 | NA | NA |
| SIOF | 0.12 | 0.36 | 0.41 | 1.42 | 0.51 | 0.09 | 0.32 | 0.37 | 1.38 | 0.47 | 0.07 | 0.28 | 0.35 | 1.34 | 0.43 |

prediction interval used is 200 ms rather than 100 ms. Regarding the tracking error, SIOF has 8% smaller error value when using 200 ms as the prediction interval rather than the prediction interval of 100 ms. Comparing with OF, SIOF has around 36% lower average tracking error value when using 200 ms as the prediction interval rather than OF.

### *6.5. Tracking overshoot*

To evaluate the performance of overshoot for the predicted values, we used the derived marginal value ($\gamma$) in (32) in Section 5.2. The upper and bounds are defined by adding and subtracting the marginal value to/from the measured value. This is to determine if the predicted outcomes exceed the bounds and the percentage of overshoots over all the frames. Figure 11 shows the tracking overshoot of the tracking estimation in degrees for pan and tilt angles' estimation using one of the "tiger" image streams. As you can see in Figs. 11(a) and (b), SIOF values exceed the upper and lower bounds at some frames but it still has smaller overshoot compared with OF. In Figure 11(c), the *Z*-position estimated using SIOF has smaller overshoot than the one estimated using SIFT. In the dataset selected here, we almost got zero overshoot in the *Z*-estimated position.

Table V summarizes the overshoot percentage calculated for the three methods, i.e., OF, SIFT, and SIOF. The percentage of overshoot represents the number of frames in which the computed value exceeds the bounds, divided by the total number of frames in the image stream. As seen in Table V, the average overshoot in pan and tilt is lower in SIOF than in OF by around 70%. Regarding the *Z*-position, the overshoot that happened using SIOF is around 2% less than the overshoot happened using SIFT. Regarding the *Z*-position estimation in SIOF, the use of feature groups, as introduced in Section 3.3, allowed us to use the average scale across each feature group instead of using the average scale of individual features. This gives more stable tracking behavior. Also, in SIOF the PTZ operations are handled concurrently, which enables better tracking in SIOF than in SIFT, especially in the cases when the target is moving across the *X*- and *Y*-axis while moving toward/away (from) the camera. Thus, the overshoot percentage is lower using SIOF than using OF and SIFT, which enables SIOF to handle large movements of the target and ensures an accurate tracking.

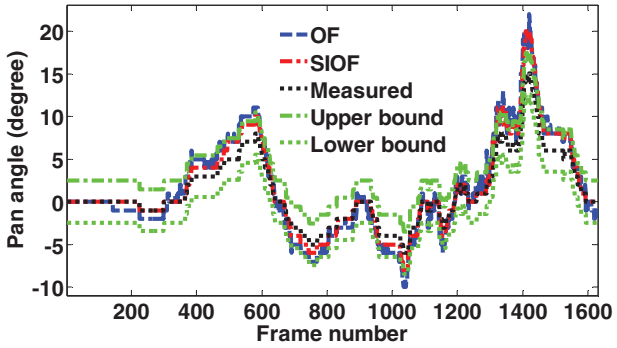
### *6.6. Comparative evaluation to other tracking methods (SURF and H-PMHT)*

We further compare our tracking method with other alternative tracking methods: SURF[41] and H-PMHT[19] which are mentioned in Section 2. Figure 12 shows the tracking error in SIOF versus SURF, and H-PMHT using one of the "face" image streams. We used the SURF functionalities provided by OpenCV such as creating an object, detecting and matching keypoints, etc. These functionalities have been used by our group to track a target using PTZ camera for comparison purposes. As seen in Figs. 12(a) and (b), SIOF has the smallest error measurement compared with other methods regarding pan and tilt errors. Regarding the zoom error in Fig. 12(c), all methods have almost similar error measurements.

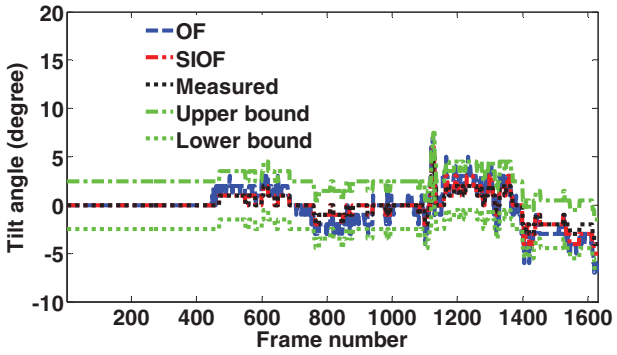
Table VI shows the quantitative comparison between the three tracking methods: (1) SURF, (2) H-PMHT, and (3) SIOF. We compare these tracking methods in terms of the PTZ errors.

As can be seen in Table VI, SIOF has the smallest pan-tilt error measurements compared with other three methods. Feature-based trackers, SURF and SIOF, are also compared in terms of zoom error. Zoom error using SURF is slightly smaller than the one using SIOF. This may be due to its

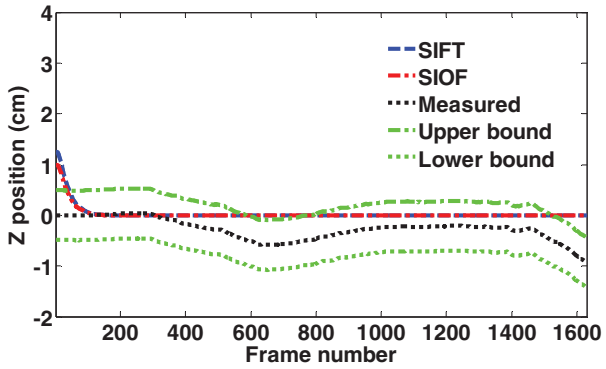
https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press




(a)




(b)




(c)

Fig. 11. Tracking overshoot for (a) pan, (b) tilt, and (c) zoom. The computed measurement by OF, SIFT, or SIOF is considered to have an overshoot if it exceeds the specified upper or lower bound (green lines).

robustness and light-weight implementation. SIOF has the second best measurement regarding zoom error after SURF.

### *6.7. Computational time*

Regarding CPU time, we have evaluated the overall performance of OF, SIFT, SURF, H-PMHT, and SIOF by measuring the average CPU time, using a PC of Intel Core 2 Duo 3.06 GHz CPU and 3-GB RAM. Table VII shows the CPU time needed to estimate tracking parameters (pan, tilt, and zoom). Time in milliseconds is counted per frame. The value after $\pm$ sign is the standard deviation of the CPU time all over the frames of the image stream.

As can be seen in Table VII, OF is fast while SIFT is slower. SIOF needs almost 15% more time than SIFT. This is because SIOF uses OF and SIFT as its features trackers. SIOF computes PTZ parameters concurrently, while OF calculates only pan and tilt and SIFT calculates only zoom. The time complexity is generally lower using the "building" image stream due to the less number of

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table V. Average percentage of tracking overshoot.

| | Tracking overshoot (%) | | |
|---|---|---|---|
| | Pan | Tilt | Zoom |
| OF | 0.154 | 0.015 | N/A |
| SIFT | N/A | N/A | 0.2103 |
| SIOF | 0.049 | 0.002 | 0.2066 |

Table VI. Average error of pan, tilt, and zoom using SURF, H-PMHT, and SIOF.

| | Average tracking error | | |
|---|---|---|---|
| | Pan (degree) | Tilt (degree) | Zoom (cm) |
| SURF[41] | 0.425 | 0.721 | 0.208 |
| H-PMHT[19] | 0.542 | 0.610 | N/A |
| SIOF | 0.296 | 0.254 | 0.288 |

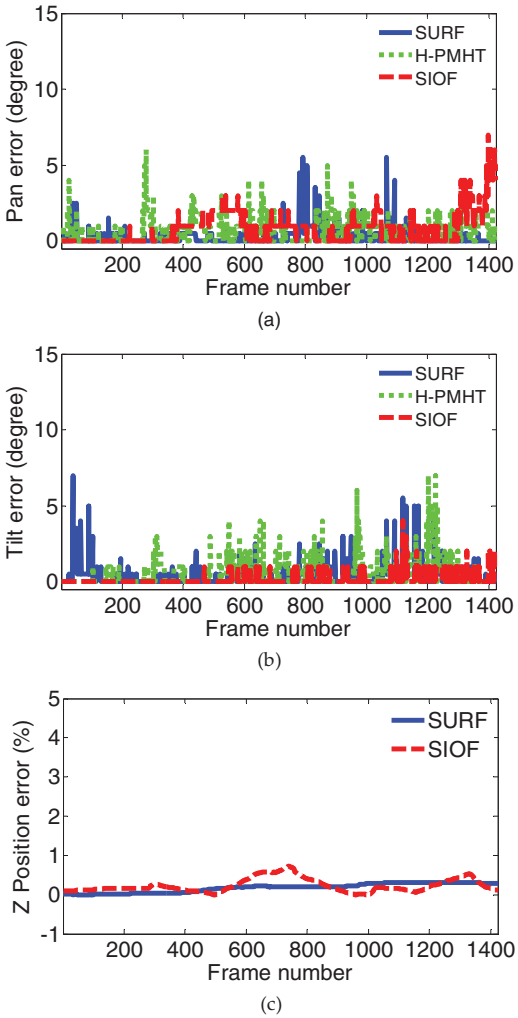



Fig. 12. Comparison between SURF,[41] H-PMHT,[19] and SIOF in terms of (a) pan, (b) tilt, and (c) zoom errors. All errors are presented in percentage (error in the $X$, $Y$, or $Z$ direction divided by the total distance of the movement in that direction).

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

Table VII. CPU time used for PTZ estimation in the three image streams using OF, SIFT, SURF, H-PMHT, and SIOF.

| | Building (ms/frame) | Tiger (ms/frame) | Face (ms/frame) |
|---|---|---|---|
| OF[17] | $4 \pm 1.3$ | $5 \pm 3.2$ | $5 \pm 1.4$ |
| SIFT[1] | $35 \pm 2.2$ | $41 \pm 2.1$ | $36 \pm 1.1$ |
| SURF[49] | $23 \pm 1.4$ | $22 \pm 2.1$ | $22 \pm 1.1$ |
| H-PMHT[39] | $14 \pm 3.1$ | $16 \pm 2.3$ | $14 \pm 1.6$ |
| SIOF | $41 \pm 2.2$ | $46 \pm 1.3$ | $43 \pm 2.2$ |

Table VIII. Average CPU time used for each step of PTZ estimation in the three image streams using OF, SIFT, SURF, H-PMHT, and SIOF.

| | Feature detection (ms/frame) | Feature matching (ms/frame) | Pan/tilt (ms/degree) | Zoom (ms/degree) |
|---|---|---|---|---|
| OF[17] | $0.35 \pm 0.11$ | $2.05 \pm 0.13$ | 1.91 | N/A |
| SIFT[1] | $15.32 \pm 0.23$ | $9.41 \pm 0.31$ | N/A | 3.49 |
| SURF[49] | $10.21 \pm 0.12$ | $7.19 \pm 0.32$ | 2.02 | 2.39 |
| H-PMHT[39] | $6.13 \pm 0.13$ | $5.11 \pm 0.22$ | 1.81 | N/A |
| SIOF | $17.39 \pm 0.19$ | $11.93 \pm 0.13$ | 2.21 | 3.51 |

feature points, while for the "tiger" image stream, it has more feature points. SURF and H-PMHT are also faster than SIFT and SIOF.

Table VIII shows an approximate of the CPU time needed to accomplish each step of tracking such as feature detection and matching, and performing the camera's PTZ operations. Time in milliseconds is counted per frame or degrees moved by the camera. The value after $\pm$ sign is the standard deviation of the CPU time all over the image streams (building, tiger, and face). As you can see in Table VIII, feature detection and matching are the most time-consuming operations in the tracking methods. PTZ operations take very similar time across all the methods.

## 7. Conclusions

In this paper, we presented a tracking method, SIOF, which controls pan-tilt-zoom of the camera to keep the object in the FoV of the camera with the same size. The significance of this proposed method is that it presents a dynamic tracking framework that monitors the movement of a target of interest and rotates accordingly to center the target in the image with the same size. In this method, two feature trackers are used to extract feature points: OF and SIFT. The feature point correspondences are fed into a twist model to recover the 3D motion of the target. This 3D motion is used to determine the $X$, $Y$, and $Z$ positions of the target and then the PTZ functions of the camera. Since it is a dynamic system, SIOF uses prediction with Kalman filter to smooth and speed up the tracking process.

Experimental results were applied on three types of online image streams. SIOF was applied to measure $X$, $Y$, and $Z$ positions and the PTZ tracking parameters. SIOF was compared with OF and SIFT. SIOF has around 36% lower average tracking error than OF. SIOF also showed a lower overshoot in pan and tilt estimation of around 70% than OF. SIOF was also compared with other tracking approaches: SURF and H-PMHT. SIOF outperforms the mentioned methods regarding pan and tilt errors. Thus, SIOF can be used in a PTZ tracking environment since it decreases the tracking errors more than done by other methods.

## Acknowledgements

This study was supported in part by the School of Engineering at Virginia Commonwealth University (VCU), Higher Education Equipment Trust Fund from the State Council of Higher Education for Virginia, VCU's Presidential Research Incentive Program, and NSF CAREER Award 1054333.

https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

## References


1. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (Nov. 2004).
2. C. Liu, J. Yuen and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Pattern Anal. Mach. Int.* **33**(5), 978–994 (May 2011).
3. C. Liu, J. Yuen, A. Torralba, J. Sivic and W. T. Freeman, "SIFT Flow: Dense Correspondence Across Different Scenes," **In**: *Proceedings of ECCV 2008*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Vol. 5304 (Oct. 2008) pp. 28–42.
4. Y. Yao, B. Abidi and M. Abidi, "3D target scale estimation and target feature separation for size preserving tracking in PTZ video source," *Int. J. Comput. Vis.* **82**(3), 244–263 (May 2009).
5. C. Micheloni and G. L. Foresti, "Active tuning of intrinsic camera parameters," *IEEE Trans. Autom. Sci. Eng.* **6**(4), 577–587 (Oct. 2009).
6. C. F. Graetzel, B. J. Nelson and S. N. Fry, "A dynamic region-of-interest vision tracking system applied to the real-time wing kinematic analysis of tethered drosophila," *IEEE Trans. Autom. Sci. Eng.* **7**(3), 463–473 (Jul. 2010).
7. I. Mikic, M. Trivedi, E. Hunter and P. Cosman, "Human body model acquisition and tracking using voxel data," *Int. J. Comput. Vis.* **53**(3), 199–223 (Jul.–Aug. 2003).
8. D. S. Jang, S. W. Jang and H. I. Choi, "2D human body tracking with structural Kalman filter," *Pattern Recognit.* **35**(10), 2041–2049 (Oct. 2002).
9. D. Schulz, W. Burgard, D. Fox and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robot. Res.* **22**(2), 99–116 (Feb. 2003).
10. R. Rosales and S. Sclaroff, "A framework for heading-guided recognition of human activity," *Comput. Vis. Image Underst.* **91**(3), 335–367 (Sep. 2003).
11. X. D. Sun, J. Foote, D. Kimber and B. S. Manjunath, "Region of interest extraction and virtual camera control based on panoramic video capturing," *IEEE Trans. Multimedia* **7**(5), 981–990 (Oct. 2005).
12. S. G. Wu and L. Hong, "Hand tracking in a natural conversational environment by the interacting multiple model and probabilistic data association (IMM-PDA) algorithm," *Pattern Recognit.* **38**(11), 2143–2158 (Nov. 2005).
13. X. P. Yun and E. R. Bachmann, "Design implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *IEEE Trans. Robot.* **22**(6), 1216–1227 (Dec. 2006).
14. D. Beymer and K. Konolige, "Tracking people from a mobile platform," *Exp. Robot. VIII Springer Tracts in Adv. Robot.* **5**, 234–244 (2003).
15. B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.* **17**(1–3), 185–203 (Aug. 1981).
16. B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with An Application to Stereo Vision," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, (Aug. 1981), vol. 81, pp. 674–679.
17. C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," *Technical Report CMU-CS-91-132*, Carnegie Mellon University, Pittsburgh, PA (Apr. 1991).
18. J. Shi and C. Tomasi, "Good Features to Track," *Proceedigs of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, USA (Jun. 1994) pp. 593–600.
19. R. L. Streit, M. L. Graham and M. J. Walsh, "Multitarget tracking of distributed targets using histogram – PMHT," *Digit. Signal Process.* **12**, 394–404 (May 2002).
20. Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(8), 790–799 (1998).
21. D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Proc. Comput. Vis. Pattern Recognit.*, Hilton Head, SC, USA (2000) pp. 673–678.
22. L. Ido, "Mean shift trackers with cross-bin metrics," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 695–706 (Apr. 2012).
23. L. Ido, L. Michael and R. Ehud, "Mean shift tracking with multiple reference color histograms," *Comput. Vis. Image Underst.* **114**(3), 400–408 (Mar. 2010).
24. G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *IEEE Workshop on Applications of Computer Vision*, Princeton, NJ (1998), pp. 214–219.
25. 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.* **32**(3), 402–415 (Mar. 2010).
26. D. Serby, E.-K. Meier and L. Van Gool, "Probabilistic Object Tracking Using Multiple Features," **In**: *Proceedings of 17th International Conference on Pattern Recognition (ICPR 2004)*, IEEE, Cambridge, United Kingdom, Vol. 2 (2004) pp. 184–187.
27. M. Tarhan and E. Altug, "A catadioptric and pan-tilt-zoom camera pair object tracking system for UAVs," *J. Intell. Robot. Syst.* **61**(1–4), 119–134 (Mar. 2011).
28. P. D. Z. Varcheie and G. A. Bilodeau, "Adaptive fuzzy particle filter tracker for a PTZ camera in an IP surveillance system," *IEEE Trans. Instrum. Meas.* **60**(2), 354–371 (Feb. 2011).
29. D. Song, Y. Xu and N. Qin, "Aligning windows of live video from an imprecise pan-tilt-zoom camera into a remote panoramic display for remote nature observation," *J. Real-Time Image Process.* **5**(1), 57–70 (2010).
30. B. Tordoff and D. Murray, "Reactive control of zoom while fixating using perspective and affine cameras," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1), 98–112 (Jan. 2004).
31. B. Tordoff and D. Murray, "A method of reactive zoom control from uncertainty in tracking," *Comput. Vis. Image Underst.* **105**(2), 131–144 (Feb. 2007).


https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press

32. S. A. Hutchinson, G. D. Hager and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.* **12**(5), 651–670 (Oct. 1996).
33. F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robot. Autom. Mag.* **13**(4), 82–90 (Dec. 2006).
34. N. R. Gans, G. Hu and W. E. Dixon, "Keeping Multiple Objects in the Field of View of a Single PTZ Camera," *Proceedings of the 2009 American Control Conference (ACC '09)* (Jun. 2009) pp. 5259–5264.
35. I.-H. Chen and S.-J. Wang, "An efficient approach for the calibration of multiple PTZ cameras," *IEEE Trans. Autom. Sci. Eng.* **4**(2), 286–293 (Apr. 2007).
36. S. Se, D. G. Lowe and J. J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Trans. Robot.* **21**(3), 364–375 (Jun. 2005).
37. H. Zhou, Y. Yuan and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Underst.* **113**(3), 345–352 (Mar. 2009).
38. A. H. Chen, M. Zhu, Y. H. Wang and C. Xue, "Mean Shift Tracking Combining SIFT," **In**: *Proceedings of the 9th International Conference on Signal Processing, 2008*, Beijing, China, Vol. 1–5 (2008) pp. 1532–1535.
39. Y. Cui, N. Hasler, T. Thormaehlen and H. P. Seidel, "Scale Invariant Feature Transform with Irregular Orientation Histogram Binning," **In**: *Proceedings of International Conference on Image Analysis and Recognition*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Vol. 5627 (2009), pp. 258–267.
40. H. Lee, P. G. Heo, J. Y. Suk, B. Y. Yeou and H. Park, "Scale-invariant object tracking method using strong corners in the scale domain," *Opt. Eng.* **48**(1), 017204–017204-9 (Jan. 2009).
41. H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," **In**: *Proceeding of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Vol. 3951 (May 2006), pp. 404–417.
42. C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," **In**: *Proceedings of the IEEE CS Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, (1998), pp. 8–15.
43. C. Bregler, J. Malik and K. Pullen, "Twist-based acquisition and tracking of animal and human kinematics," *Int. J. Comput. Vis.* **56**(3), 179–194 (Feb. 2004).
44. Z. H. Khan and I. Y. H. Gu, "Joint feature correspondences and appearance similarity for robust visual object tracking," *IEEE Trans. Inf. Foren. Secur.* **5**(3), 591–606 (Sep. 2010).
45. D. Churchill and A. Vardy, "Homing in Scale Space," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France (Sep. 2008), pp. 1307–1312.
46. V. Venkateswar and R. Chellappa, "Hierarchical stereo and motion correspondence using feature groupings," *Int. J. Comput. Vis.* **15**, 45–69 (1995).
47. T. Morita and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(8), 858–867 (Aug. 1997).
48. Polhemus Documentation: Available at: http://www.polhemus.com/?pa-ge=Motion_Liberty. [Accessed 10 August 2010].
49. C. Chatfield, "Prediction intervals for time-series forecasting," *Principles of forecasting.* Springer US, (2001), pp. 475–494.
50. S. Lim and A. El-Gamal, "Optical Flow Estimation Using High Frame Rate Sequences," **In**: *Proceeding of the IEEE 2001 International Conference on Image Processing (ICIP)*, Thessaloniki, Greece (Oct. 2001), Vol. 2, pp. 925–928.
51. C. Liu, J. Yuen and A. Torralba, "Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA, (Jun. 2009) pp. 1972–1979.


https://doi.org/10.1017/S0263574714002665 Published online by Cambridge University Press