International Journal of Automation and Computing 12(6), December 2015, 600-610
DOI: 10.1007/s11633-015-0908-8

Tracking and Guiding Multiple Laser Beams for Beam and Target Alignment

Peng-Cheng Zhang De Xu

Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

Abstract: An accurate and robust approach for tracking and guiding multiple laser beams is developed, which can be applied to the task of beam and target alignment. Multiple laser spots are firstly detected and recognized from the image sequences of the target and laser spots. Then, the contour tracking algorithm based on the chain code is investigated, in which the shape matching scheme based on the invariant moments is employed to distinguish different spots. When occlusion occurs in the multiple spots tracking procedure, the contour tracking combined with Kalman filter prediction is proposed to obtain the positions of multiple spots in real-time. In order to guide 3 spots to align the target, an incremental proportional integral (PI) controller is employed to make the image features of spots converge to the desired ones. Comparative experiments show that, the proposed tracking method can successfully cope with the fast motion, partial or complete occlusion. The experiment results on spots guiding also exhibit the accurate and robust performance of the strategy. The proposed visual system solves the problem of spots mixing, reduces the alignment time, improves the shooting accuracy and has been successfully applied to the experimental platform.

Keywords: Laser spot, spot segmentation, contour tracking, visual feedback, beam and target alignment.

1 Introduction

In order to apply the theories of inertial confinement fusion (ICF) to practice, the challenges associated with beam and target alignment system must be overcome[1]. It is necessary to take measures to realize rapid and precise alignment in a time as short as possible. In order to achieve higher shooting accuracy, the laser spots should be guided through continuous visual feedback in real-time[2]. Normally, the process of aligning multiple laser beams to the desired region on the target is called a beam and target alignment process, and a sensor for aligning multiple laser beams to the target is called a target alignment sensor, which mainly consists of the conjugate reflectivity mirror and the microscope lens, as shown in Fig. 1. In this work, 3 dedicated laser alignment devices are integrated to guide 3 laser beams to achieve the beam and target alignment. For each laser beam, the laser from the transmitter is amplified, processed through a filter wheel, focused by a set of convergent lenses, and then filtered via an aperture of a circular hole of 6 mm diameter. Finally, the laser is focused again by a convex lens and is reflected to the end surface of target by a high-reflectivity mirror. Therefore, the detecting, tracking and guiding technologies of laser spots become very important so that multiple laser beams can shoot the desired region of the target in the beam and target alignment procedure.

The detection and segmentation of multiple spots in image sequences is a fundamental step before the tracking or when the tracking is failed. The laser spot is assumed to be the only part of the image with a high light intensity. The common approach is to extract the spots by background subtraction. However, in practice, the intensity of laser may randomly change, the illumination in the scene may not be very stable and even the intensity of a cylindrical target may vibrate slightly. All of these may induce the changes of background. Stauffer and Grimsor[3] modeled each pixel as a mixture of Gaussians and used an on-line approximation to update the model. However, the method suffers from slow learning at the beginning and is not sensitive to the small motions. An improved Gaussian mixture model was presented by Zivkovic[4], in which not only the parameters but also the number of components of the mixture were constantly selected for each pixel. However, this method has high computational cost, especially, in the face of the high image resolution.

The tracking algorithm can provide the position feedback for the spot in real-time. In [5], the object tracking was divided into point tracking, kernel tracking and silhouette tracking. Features, such as color, edges, optical flow and texture are often chosen for tracking. They can be represented by a probabilistic model and then detected in consecutive frames. In general, the most desirable features of spot are the ones which can be used to distinguish one spot from the others. Considering the gray image, the biggest difference between different spots is the contour then the intensity. Moreover, due to the fact that the spot is non-

Regular Paper
Manuscript received December 25, 2013; accepted December 4, 2014
Recommended by Associate Editor Nazim Mir-Nasiri
This work was supported by National Natural Science Foundation of China (Nos. 61227804 and 61105036).

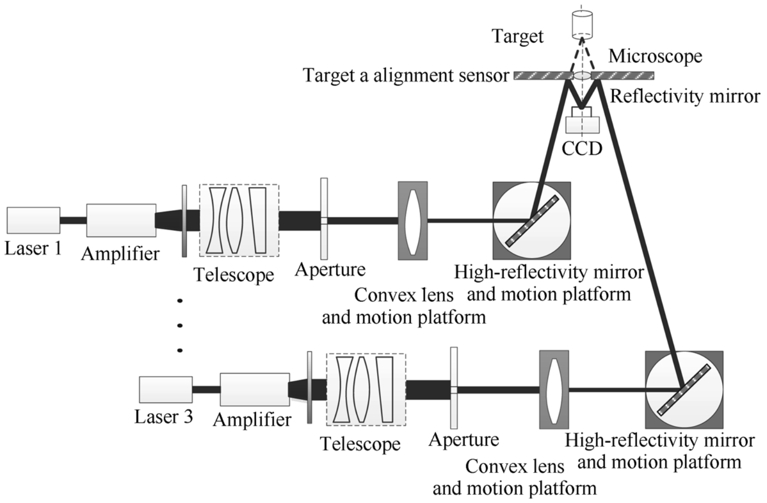


Fig. 1 Beam and target alignment system configuration, in which the microscope is used in the imaging procedure of the target while it is not used in the imaging procedure of laser spots

rigid, the tracking algorithm based on its contour is more practical. Many contour models have been reported for tracking in the previous literature, such as optical flow, level set, snakes, balloons and active contours model[6−8]. Some contour tracking methods based on boundary codes were investigated in [9, 10]. However, they are limited to the boundary-based information and are sensitive to noise. In order to overcome the drawbacks of the sensitiveness to noise and poor image contrast, a particle filtering algorithm for geometric active contours tracking was proposed in [11]. However, these techniques require a number of iterations and are computationally too expensive for real-time application on multiple spots tracking. It is necessary to develop one rapid and robust contour tracking scheme for spots.

Another important problem is that the spots may mix together and interfere with each other when multiple laser beams simultaneously shoot the target. The problem may be solved by guiding a single beam at a time, however, which is time-consuming. Therefore, occlusion handling is a difficult issue in the face of multiple spots tracking. In the previous literature, the appearance model was incorporated, or the target was treated as a blob which may merge and split, or an exclusion principle was employed by using the joint probabilistic data association filter and the particle filter was employed to avoid the high computational load[12−15]. The prior information of shape is often integrated into contour representation. Yilmaz et al.[16] proposed a non-rigid tracking method, which was achieved by evolving the contour from frame to frame with some energy functions. The contour represented by level sets was used to recover the missing object regions during occlusion. However, these methods require precise model, increase the computational complexity and may fail in the face of the rapid motions.

The recognition of multiple spots is also a key problem during multiple spots tracking. In order to distinguish different spots, shape representation and matching techniques should be considered. A number of successful shape matching algorithms were proposed. One of the most popular methods is to use Hausdorff distance, although it is very sensitive to outliers. Some methods compare shape by the feature vector which contains the descriptors such as area, geometric moments, shape matrix, appearance via gray histograms, optical flow vectors, etc.[17], while others directly do with the aid of pixel brightness[18]. Belonggie et al.[19] proposed the shape context for shape matching and object recognition, which described the contour points by histogram in the log-polar space. The similarity between two shapes was computed by a sum of matching errors between corresponding points. However, a large amount of calculations will be needed and they are hard to satisfy the real-time requirement.

The motivation of this paper is to develop the detection, tracking and guiding schemes for multiple laser spots based on a beam and target alignment experimental platform. An accurate and real-time system for multiple laser beams shooting is presented, which consists of spots segmentation, spots contour tracking even under occlusion and spots guiding based on visual feedback control.

The rest of this paper is organized as follows. Section 2 describes multiple spots detection and tracking strategy, in which the spot segmentation, the contour tracking and the shape matching algorithms are introduced. In order to accomplish the beam and target alignment task, Section 3 presents multiple spots guiding scheme. Section 4 provides the experimental configuration and the analysis of experiment results. Finally, the paper is concluded in Section 5.

2 Multiple spots detection and tracking strategy

The achievement of the beam and target alignment based on visual feedback needs to obtain the positions of multiple spots in real-time. The accurate positions need to be determined by the detection and tracking stages. As shown in Fig. 2, the proposed strategy is initialized with a status flag $L = 1$. Each spot is recognized by its shape and is numbered after the detection. Meanwhile, its location, width, height, image moments and so on are also calculated. If all spots are found, then flag L is set to 0. When a new image is captured, the tracking algorithm is employed to obtain the new features of spots. When occlusion occurs, the tracking scheme combined with a prediction mechanism will determine the new positions of spots. If the shape matching between the current contour and the previous one is failed, then, flag L changes to 0. Otherwise, the positions of the current spots will be provided to the guiding stage. Then, the spots in the next frame will be tracked until stopping.

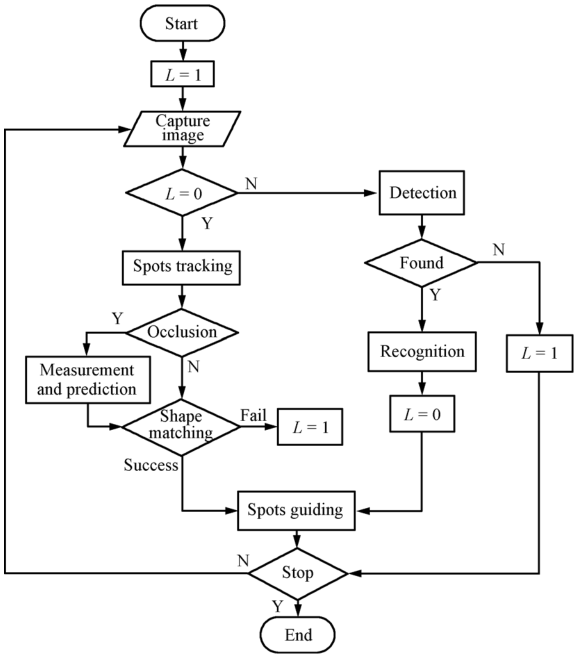


Fig. 2 Block diagram of multiple spots detection, tracking and guiding strategy

3 Spots detection stage

3.1 Spots detection stage

First of all, the moving spots are segmented from the beam and target images. However, noise may be introduced and the differential image may be incomplete by stationary background subtraction. Many solutions in the previous literature have been proposed for real-time foreground detection from moving background. Here, the adaptive Gaussian mixture model (GMM) is employed to obtain the spots background[20]. A Bayes decision rule for classifying background and foreground is formulated and the learning strategy is introduced to adapt to slight changes in background. The probability of each pixel value at time t can be written by

$$p(x_t) = \sum_{i=1}^{K} w_i \eta(x_t, \mu_i, Q_i) \tag{1}$$

where K is the number of Gaussian distributions, w_i is the weight of the i-th Gaussian component, μ_i is the mean value, Q_i is the covariance matrix and $\eta(x_t, \mu_i, Q_i)$ is the i-th Gaussian probability density function, which is represented by

$$\eta(x_t, \mu_i, Q_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |Q_i|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(x_t-\mu_i)^{\mathrm{T}} Q_i^{-1} (x_t-\mu_i)}. \tag{2}$$

In order to avoid the complex matrix computation, let $Q_i = \sigma_i^2 I$. Considering the high resolution of the spot image, no more than 3 Gaussian models are initialized.

Each pixel is firstly classified as either background or a foreground pixel by the models. The K distributions are ordered based on the fitness value w_i/σ_i. The first B distributions are selected as a model of the background, which are estimated by

$$B = \arg\min_b \left(\sum_{i=1}^{b} w_i > T \right) \tag{3}$$

where T is the threshold of background weight value. The Gaussian model that matches the current pixel value will be updated by the following formulae.

$$\begin{aligned} \hat{w}_i^{t+1} &= \hat{w}_i^t + \alpha\left(1 - \hat{w}_i^t\right) \\ \hat{\mu}_i^{t+1} &= \hat{\mu}_i^t + \alpha\left(x_{t+1} - \hat{\mu}_i^t\right) \\ \hat{Q}_i^{t+1} &= \hat{Q}_i^t + \alpha\left((x_{t+1} - \hat{\mu}_i^{t+1})(x_{t+1} - \hat{\mu}_i^{t+1})^{\mathrm{T}} - \hat{Q}_i^t\right). \end{aligned} \tag{4}$$

If no Gaussian model matches the current pixel value, then the least probable model is replaced by the formulae $w_i^{t+1} = \alpha$, $\mu_i^{t+1} = x_t$ and $Q_i^{t+1} = \sigma_0$. If the maximum number of components is reached, the component with the smallest weight will be discarded.

However, the foreground region segmented by the Gaussian mixture model is not sufficient to be clear. Firstly, a binary image is obtained through adaptive threshold segmentation, in which the value of each pixel is compared with the weighted average value around the pixel.

$$I_b(u,v) = \begin{cases} 255, & \text{if } (I_b(u,v) \text{ - } I_{\bar{b}}(u,v)) > I_T \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $I_{\bar{b}}(u,v)$ is the convolution of pixel by one Gaussian kernel operator and I_T is the constant value. Then, the region-based noise cleaning is applied, such as the morphological open operation including the erosion by the structuring element A then the dilation by the structuring element B, denoted by

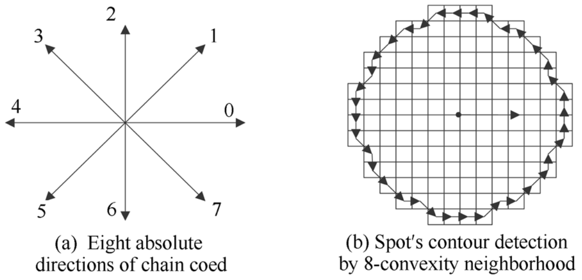
$$(X \oplus A) \ominus B. \tag{6}$$

This opening operation can generally remove very small regions, eliminate thin protrusions and smooth the contour of the spot. Then, the shape filter is used to classify the spots. So far, all spots are numbered and each one has its own ID number.

3.2 Spots tracking stage

In this subsection, the contour tracking algorithm based on the chain code is introduced. Many applications using chain code representation have been reported in previous literatures. The first approach for representing the arbitrary geometric curve using chain code was proposed by Freeman[21]. In this approach, an arbitrary contour can be represented by a sequence of small vectors of unit length and a set of possible directions. There are two standard code definitions used to represent contour, including the crack code based on 4-connectivity and the chain code based on 8-connectivity.

In this work, the real-time and robustness are very important for multiple spots tracking. Here, the 8-connectivity chain code is employed for representing the contour of spot, which is based on the connectivity of neighboring pixels[22]. Fig. 3 illustrates the changes of 8 possible absolute directions, which are indicated by numbers. "0" indicates the direction change to the east, "1" indicates the direction change to the northeast, "2" indicates the direction change to the north, "3" indicates the direction change to the northwest, "4" indicates the direction change to the west, "5" indicates the direction change to the southwest, "6" indicates the direction change to the south and "7" indicates the direction change to the southeast. A change between two consecutive chain codes means a change in the direction of the contour. Thus, each spot′s contour can be coded by the chain code in the image space.



(a) Eight absolute directions of chain coed

(b) Spot's contour detection by 8-convexity neighborhood

Fig. 3 Spot′s contour tracking based on chain code

Generally, the basic principle of the tracking based on chain code of 8 directions is to separately encode each connect component. It can be divided into 3 steps as follows.

Firstly, the initial center of the tracked spot should be specified so that the first border pixel can be found. The initial position can be obtained from the detection procedure. Starting from the first center, the first border pixel can be detected along the u-axis direction. In order to search a new contour in a small region of interest (ROI) when the tracking is failed, one suitable search window is set. A copy of the current spot should be created in order to determine whether the current spot is similar to the next one. If no similar spot is detected, this copy is set to the current spot.

Secondly, the next border pixel is considered by updating the pixel coordinates along the 8 directions. The coordinate transformation rules are described as

$$\begin{cases} u_{i+1}=u_i+1,\ v_{i+1}=v_i, & \text{if } d_i=0\\ u_{i+1}=u_i+1,\ v_{i+1}=v_i-1, & \text{if } d_i=1\\ u_{i+1}=u_i,\qquad v_{i+1}=v_i-1, & \text{if } d_i=2\\ u_{i+1}=u_i-1,\ v_{i+1}=v_i-1, & \text{if } d_i=3\\ u_{i+1}=u_i-1,\ v_{i+1}=v_i, & \text{if } d_i=4\\ u_{i+1}=u_i-1,\ v_{i+1}=v_i+1, & \text{if } d_i=5\\ u_{i+1}=u_i,\qquad v_{i+1}=v_i+1, & \text{if } d_i=6\\ u_{i+1}=u_i+1,\ v_{i+1}=v_i+1, & \text{if } d_i=7 \end{cases}$$

where u_i, v_i are the coordinates of the current pixel and d_i is the indication of the direction change. Meanwhile, the spot features, such as the center of mass, surface, bounding box and image moments, are calculated by their corresponding increments. This continues until the encoder returns to the starting border pixel. Compared to other contour-based methods, it does not go through all the neighbors of the pixel, so faster. So far, the boundaries of the spot′s contour are represented by the chain codes.

Thirdly, if the tracking is failed, a search area will be required. Thanks to the short image sampling interval, the moving distance of the spot during two consecutive frames is limited in a small range. So the dynamic window technique is applied to searching. Once the spot is found in the new frame, the center of the dynamic window will be updated. Therefore, if the tracking is failed or the spots go out of the field of view, the searching region in the next frame can be confined in this dynamic window. If the spot is still not detected in this small window, the searching window will be changed to the entire image region[23]. In addition, when the tracking is failed, not every pixel will be treated as a starting point for contour detection, but a large searching step is separately set along the u-axis and v-axis directions. If the new contour is similar enough to the previous one, the chain codes of spot contour will be updated. This tracking procedure continues until stopping.

During the multiple spots tracking procedure, if the relative chain codes of spot contour are considered for matching, it must be independent of the choice of the first boundary pixel. Usually, the normalized differential chain codes instead of the relative ones are used to represent the contour boundary and this can be computed by subtracting each element from the previous one[24]. However, the chain codes of spot contour are usually sensitive to noise, so the shape matching by chain code is impractical. In this work, considering the real-time and simplification, the image moments of spot have been employed for shape matching. The Hu invariant moments that are invariant to translation, rotation and changes in scale are adopted to evaluate the resemblance of two contours[25].

The image moments are obtained by

$$m_{pq} = \sum_{u}\sum_{v} u^p v^q I(u,v) \tag{7}$$

where the order of the moment is $p+q$, the central moments μ_{pq} are calculated by

$$\mu_{pq} = \sum_{u}\sum_{v} (u-\bar{u})^p (v-\bar{v})^q I(u,v),\ \bar{u} = \frac{m_{10}}{m_{00}},\ \bar{v} = \frac{m_{01}}{m_{00}}. \tag{8}$$

Then, the normalized central moment η_{pq} can be determined by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\lambda}}, \quad \lambda = \frac{p+q}{2}, \quad p+q \geq 2. \tag{9}$$

Further, 7 Hu invariant moments that contain image moments up to order 3 are shown as

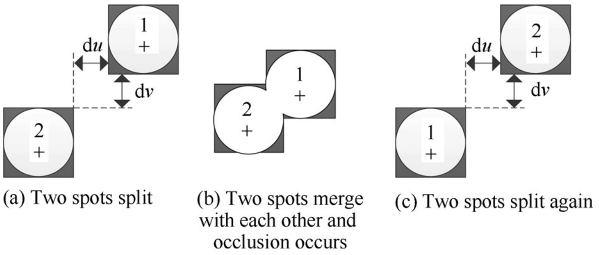
$$\begin{aligned}
h_1 &= \eta_{20} + \eta_{02}\\
h_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2\\
h_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2\\
h_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2\\
h_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) +\\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)\\
h_6 &= (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} +\\
&\quad \eta_{12})(\eta_{21} + \eta_{03})\\
h_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) +\\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2).
\end{aligned} \tag{10}$$

The Hu moments advantages of invariance to position, size and orientation make the spot matching become practical. Therefore, the 7 Hu moments combined with the spot′s width, height and surface are applied to shape matching. However, the spot′s shape is not very stable, but slightly changes in a small range. So a certain changing range is set for each moment. If the changes of moments are beyond a certain range, the corresponding shape will be rejected.

3.3 Occlusion handling

Importantly, the tracking should continue even in the event that the spot is partially or even completely occluded. Especially, the mixing phenomenon between different spots is very common. When multiple spots move to near the desired positions in the guiding procedure, occlusion may occur. In this case the spots may even completely merge for a long time.

Here, the bounding box distance is used for determining whether the spots merge or split, as shown in Fig. 4. It is more stable as opposed to the Euclidean distance between the centers of spots. Therefore, occlusion detection is performed by determining the distance between the bounding boxes of different spots. When occlusion occurs, multiple spots mix each other, which induces sudden changes in shapes. So the mass centers of spots calculated by the common contour become inaccurate. In the previous literatures, this case is defined as a filtering and data association problem.



(a) Two spots split (b) Two spots merge with each other and occlusion occurs (c) Two spots split again

Fig. 4 Bounding box distances between different spots

The most common expression of the filtering and data association process is the state space approach, which can model the discrete dynamic system by linear difference equations[26]. Kalman filter technique can predict motion information and reduce the computational complexity, so it is employed to predict and update the spot′s features in this work. The features of the spot are defined by the state sequences x_k $(k = 0, 1, \cdots)$, which are specified by the dynamic equations $x_k = f_k(x_{k-1}, v_k)$. The available measurements z_k $(k = 1, 2, \cdots)$ are related to the corresponding states through the measurement equations $z_k = h_k(x_k, n_k)$. The functions v_k and n_k are the noise sequences, which are assumed to be independent and are of Gaussian distribution with zero mean. The functions f_k and h_k are linear. The dynamic equations are defined by $x_k = Fx_{k-1} + v_k$ and the measurement equations are defined by $z_k = Hx_k + n_k$. F is the transition matrix and H is the measurement matrix.

The contour tracking technique combined with the Kalman filtering is applied when occlusion occurs. The weighted sum of the measured position and the predicted position is used for determining the spot′s position in the next frame.

Typically, a Kalman filter can be divided into the predicted phase and the corrected phase[26]. In the first phase, a priori state estimation of the spot at time k is evolved from the state at time $k-1$ according to

$$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_{k-1} \tag{11}$$

where F_k is the state transition matrix, B_k is the control matrix which is not used in this work. Meanwhile, a posteriori error covariance matrix can be estimated by

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\mathrm{T}} + Q_k \tag{12}$$

where Q_k is the process noise covariance matrix.

In the update phase, the state estimation is updated by the formula as

$$x_{k|k-1} = x_{k|k-1} + K_k(z_k - H_k x_{k|k-1}) \tag{13}$$

where H_k is the measurement matrix, K_k is the optimal Kalman gain matrix, which is a function of the relative certainty of the measurements and current state estimate. With a high gain, the filter places more weight on measurement. With a low gain, the filter follows the prediction more closely. At the extreme, a gain of zero causes the measurement to be ignored. The gain can be set up to achieve better performance and is expressed by

$$K_k = P_{k|k-1}H_k^{\mathrm{T}}(H_kP_{k|k-1}H_k^{\mathrm{T}} + R_k)^{-1} \tag{14}$$

where R_k is the measurement noise covariance matrix and the posteriori estimate covariance matrix is updated by

$$P_{k|k} = (I - K_kH_k)P_{k|k-1}. \tag{15}$$

The prediction step and correction step are executed recursively. From (15), the Kalman gain is inversely proportional to the measurement covariance matrix R_k. Therefore, the smaller R_k is, the greater the gain becomes and the higher the weight of the predicted value becomes. Likewise, the closer the priori estimate error $P_{k|k-1}$ gets to zero, the higher the weight of the measurement becomes.

In Kalman filter algorithm, the moving model of spot should be constructed. Because of the small changes on spot′s surface in consecutive frames, only 4 input states are taken into account, which are the positions and the velocities of the spot along the u-axis and v-axis directions in the image space. The position and velocity of the spot are evolved from the state at time $k-1$ according to

$$\begin{bmatrix} u_k \\ v_k \\ \Delta u_k \\ \Delta v_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{k-1} \\ v_{k-1} \\ \Delta u_{k-1} \\ \Delta v_{k-1} \end{bmatrix} \tag{16}$$

where Δu_k and Δv_k are the position increments during the continuous sampling time along the u-axis and v-axis directions, and they can replace the velocities, u_k and v_k are the pixel coordinates of the mass centers of spot. The state transition matrix is a 4×4 one, the measurement matrix is a 2×4 one, the measurement noise covariance matrix R_k is a 2×2 diagonal one, the process noise covariance matrix Q_k is also a 4×4 diagonal one, and the posteriori error estimate covariance matrix P_k is a 4×4 one.

When occlusion occurs, the entire contour including the merge boundaries can be detected by the chain code. At this moment, the position of spot is the weighted sum of the mass centers of the common contour and the prediction position. Importantly, the practical implementation of Kalman filter is often difficult to obtain a good estimation of the noise covariance matrices Q_k and R_k. The autocovariance least squares technique was used to estimate the covariance[27]. More practically, in order to reduce the computation cost, an occlusion rate is defined to determine the values of noise covariance, shown as

$$\zeta(t) = \begin{cases} \left|\frac{N_s}{N(t)}\right|, & \text{if occlusion occurs} \\ 1, & \text{otherwise} \end{cases} \tag{17}$$

where N_s is the number of pixels of spot before occlusion, $N(t)$ is the number of pixels in the common area of multiple spots under occlusion. So, $N(t)$ is greater than N_s. Here, it is assumed that the noise covariance is inversely proportional to occlusion rate $\zeta(t)$. When occlusion occurs, the measurement noise covariance becomes greater, and then the weight of the predicted value becomes higher. As the common area of multiple spots becomes smaller, the occlusion rate becomes greater and the noise covariance becomes smaller, so the weight of the measurement becomes higher. According to the occlusion rate, the Kalman filter is adjusted adaptively.

4 Multiple spots guiding based on visual feedback

The objective of the spots tracking is to guide the laser spots to accomplish the beam and target alignment. From the previous section, the positions of spots are obtained in real-time, so the position errors in pixels between the current and the desired ones are known. Fig. 5 illustrates the desired positions of 3 spots. Note that due to the pose relationship between the target alignment sensor and the high-reflectivity mirror, the two axes of the sub-coordinate system of each spot are not perpendicular to each other. The control objective is to minimize the errors by choosing an appropriate control vector at each sampling time. The control scheme for eliminating the position deviations of the spots is the discrete incremental proportional integral (PI) controller. The linear control law is given by

$$\begin{bmatrix} \Delta u_x(k) \\ \Delta u_y(k) \end{bmatrix} = K_p \left(\begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix} - \begin{bmatrix} \Delta x(k-1) \\ \Delta y(k-1) \end{bmatrix} \right) + K_i \begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix} \tag{18}$$

where parameters $\Delta u_x(k)$ and $\Delta u_y(k)$ are the output of the PI controller at the k-th control cycle, $\Delta x(k)$ and $\Delta y(k)$ are the positions errors in Cartesian space at the k-th control cycle. K_p is the proportional factor, which is a diagonal matrix. K_i is the integral factor, which is also a diagonal matrix. The Ziegler-Nichols method is employed to tune the PI parameters.

The control structure is shown in Fig. 6. The coordinates of the mass centers in pixels are obtained from the camera system. Then, the errors $\Delta u(k)$ and $\Delta v(k)$ in the image space between the current position and the desired one are calculated. The movements of the spot are achieved by the rotational motions of the two servo motors with a high-reflectivity mirror along the pitch and yaw directions. The rotational increments of two motors $\Delta x(k)$ and $\Delta y(k)$ can be acquired by

$$\begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix} = \begin{bmatrix} p_u & 0 \\ 0 & p_v \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta v(k) \end{bmatrix} \tag{19}$$

where parameters p_u and p_v are the proportional relationship between the translational motions of the spot in pixels

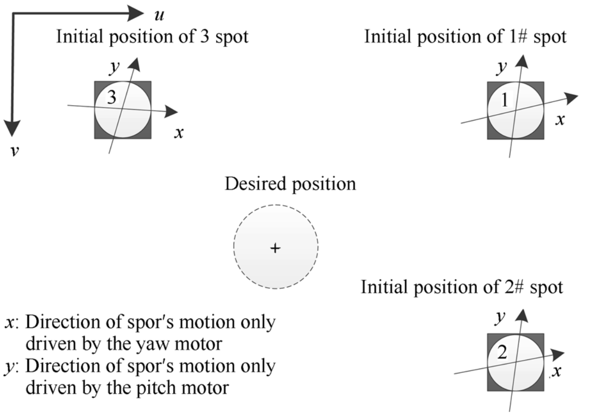

u
v
Initial position of 3 spot
3
y
x
Initial position of 1# spot
1
y
x
Desired position
+
Initial position of 2# spot
2
y
x
x: Direction of spor's motion only driven by the yaw motor
y: Direction of spor's motion only driven by the pitch motor

Fig. 5 The initial and desired positions of 3 spots in the beam and target alignment procedure and the axes of the sub-coordinate system of each spot are not perpendicular to each other

and the rotational motions of the motor. The values of p_u and p_v can be separately calibrated off-line by multiple motions, as shown in (20).

$$p_u = \frac{\sum\limits_{i=1}^{N} \frac{\Delta x_i}{\Delta m_i}}{N}, \quad p_v = \frac{\sum\limits_{i=1}^{N} \frac{\Delta y_i}{\Delta n_i}}{N} \qquad (20)$$

where Δx_i is the increment of the motor′s motion only along the pitch direction and Δm_i is the corresponding increment of the spot′s motion in the image plane. Similarly, Δy_i is the increment of the motor′s motion only along the yaw direction and Δn_i is the corresponding increment of the spot′s motion in the image plane. Meanwhile, the corresponding unit vector can be obtained by line fitting. In the control procedure, the rotational increments $\Delta x(k)$ and $\Delta y(k)$ are obtained by the increments in image space, which can be treated as the input of PI controller. Finally, the outputs $\Delta u_x(k)$ and $\Delta u_y(k)$ are used as the input to the motor′s controller.

5 Experiment and analysis

5.1 Experiment system

The whole experimental configuration consists of 3 beam and target alignment subsystems, as shown in Fig. 7. Each laser subsystem of laser is composed of a laser transmitter, a set of convergent lenses, a convex lens with a translational motion platform and a high-reflectivity mirror with two rotational motions platform. Before passing through the aperture, the laser needs to be adjusted manually. Then, it is focused again by a convex lens with a translational motion platform. Finally, the laser is reflected by a high-reflectivity mirror which is driven by one pitch motor and one yaw motor. In addition, the camera system could simultaneously image the laser spots and the cylindrical target, which could provide the position feedback to ensure the alignment accuracy. Therefore, the imaging camera and the guiding motors including the pitch and yaw ones constituted one visual servoing system. Importantly, after being reflected by the high-reflectivity mirror, the laser is reflected again by the mirror on the target alignment sensor, and then it is captured by the camera system. So, the charge-coupled device (CCD) plane and the target′s end surface should be conjugate relative to the plane of target alignment sensor so that the laser could shoot the target′s surface after the target alignment sensor is removed. It effectively prevents multiple laser beams from directly shooting the target.

The above manipulators are the Aerotech′s motor named ANT95-50-XY in the translational direction, one named ANT 130-360-R in the yaw direction and one named ANT-20G-90 in the pitch direction. The accuracy of the rotational motor is ±50 μrad and the resolution is 0.25 μrad. In order to guarantee that the reflected spots are not out of view, the rotational angle of each motor is limited within 0.015 rad. The imaging detector is GRAS-50S5M-C from Point Grey Research. The camera could run at 15 frames per second for 2448(H) × 2048(V) in pixels. The software system has two processes including a user interface process and an application one. The interprocess communication is achieved by shared memory and the socket technology. Meanwhile, multi-threading technology is used in multiple spots tracking procedure. The experiment was implemented on a PC (Intel i7 2.80 GHz).

5.2 Experimental result and analysis

In the spots detection experiments, the noise variance was set to 6.4, the initial weight value was 0.04, the variance threshold was 0.64, the threshold of background weight value was 0.9, and the learning rate was set to 0.005. The procedure of spots segmentation based on the Gaussian mixture model is shown in Figs. 8 (e) to 8 (g). As a contrast, the procedure of stationary background subtraction is exhibited in Figs. 8 (b) to 8 (d). Compared to the stationary background subtraction, the segmentation results based on GMM are more stable and precise, especially in the face

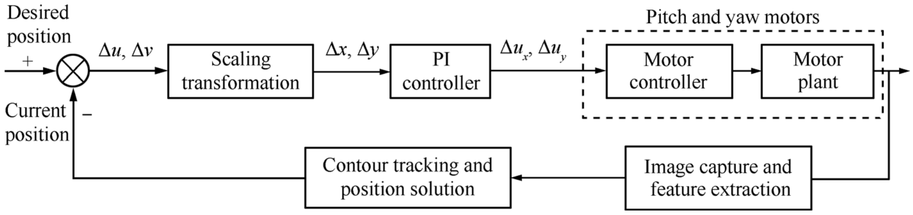


Fig. 6 Control architecture of spots guiding based on visual feedback

of the complex beam and target images or the rapid movement of spot. Then, the adaptive threshold operation, the morphological open operation including the erosion with a 3×3 operator and the dilation with a 5×5 operator are used to clean noise.

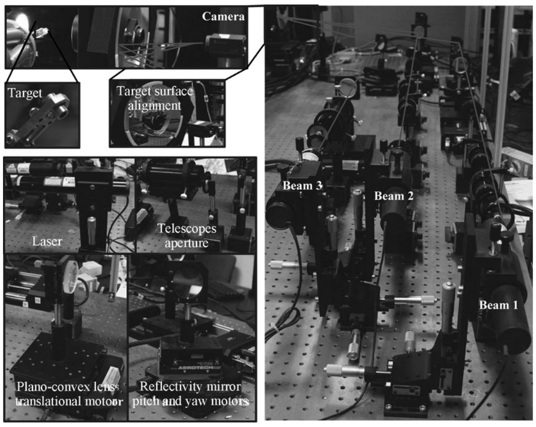


Fig. 7 Experimental system

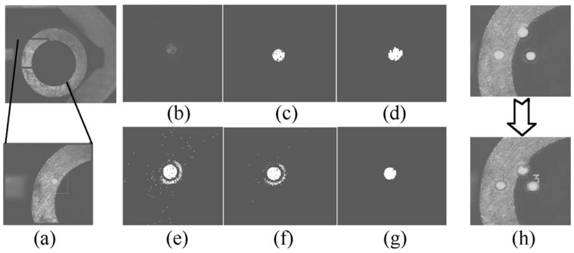


Fig. 8 The spot segmentations based on stationary background subtraction and Gaussian mixture model. (a) The beam and target images; (b)−(d) The spot after the stationary background subtraction, the spot after the threshold segmentation and the spot after the morphological operation, respectively; (e)−(g) The spot after subtraction based on GMM, the spot after adaptive threshold segmentation and the spot after the morphological operation, respectively; (h) The image segmentation and contour extraction of 3 spots.

In this work, 3 laser beams are used for tracking and guiding. The first laser spot is about 51 pixels in width, 49 pixels in height and 1 980 pixels in surface. The second spot is about 70 pixels in width, 65 pixels in height and 3 560 pixels in surface. The third spot is about 64 pixels in width, 64 pixels in height and 3 202 pixels in surface. In the tracking procedure, if the tracking is failed, the size of searching window would be set to 15 times the spot′s size and the corresponding searching steps along u-axis and v-axis directions would be respectively set to 0.75 times the spot′s width and height. In addition, the tracking times have been tested. The average times spent on searching the contour based on chain code are 1.373 ms for the first spot, 1.134 ms for the second spot and 1.235 ms for the third spot. The whole tracking times including the spot segmentation are 38.051 ms for the first spot, 39.806 ms for the second spot and 41.160 ms for the third spot. The real-time contour tracking technique based on the chain code could meet the requirement of the experimental system.

The experiments on handling the problem of spots mixing have been conducted. As discussed in Subsection 3.3, whether the spots merge with each other is determined by the bounding box distances. In the Kalman filter algorithm, the initial values of the diagonal elements of matrix R_k are set to 3, the initial values of the diagonal elements of matrix Q_k are $1\mathrm{e}^{-6}$ and the initial values of the diagonal elements of matrix P_k are 0.1. When multiple spots mix with each other in the guiding procedure, the center of the searching window calculated by the common contour would have a sudden shift. However, thanks to the prediction mechanism based on Kalman filter, the sudden shift would be decreased. A range of tracking results combined with Kalman filter prediction is shown in Figs. 9 (a)−9 (f), in which the trajectory of the tracking center is marked with the solid line. Fig. 10 exhibits the tracking results of 3 spots in the face of merging and splitting. The proposed algorithm is able to track multiple spots in the presence of partial or complete occlusion, which increases the feedback accuracy.

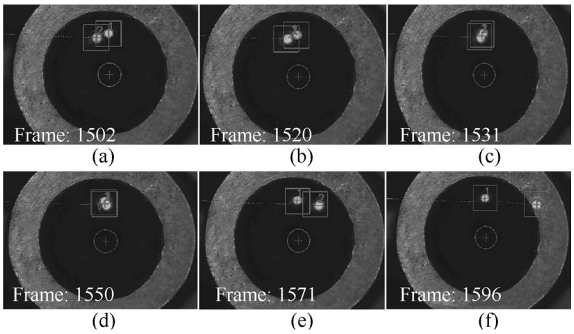


Fig. 9 The tracking image sequences under two spots mixing, in which the first spot remained in motion, merged with the second spot and then split during its movement. The second spot remained motionless. (a)−(f) Image sequence frames 1 502, 1 520, 1 531, 1 550, 1 571, and 1 596 are shown.

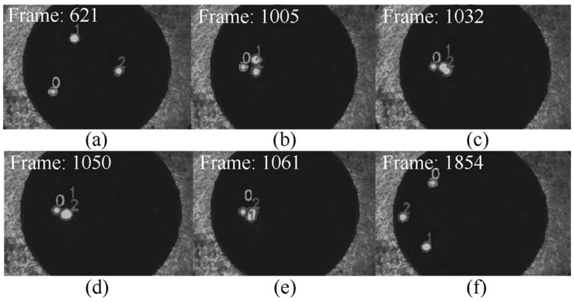


Fig. 10 The tracking results in the case of 3 spots mixing. These are the images sequences for the 3 laser spots in the cases of merging, splitting, partially occluding and completely occluding. (a)−(f) Image sequence frames 621, 1 005, 1 032, 1 050, 1 061, and 1 854 are shown, in which the entire contour is detected under the occlusion.

In order to show the better performance of the proposed tracking algorithm, the comparative experiments have been carried out as follows.

The most popular shape representation is the shape context proposed in [19]. Yet in the face of the conditions that the strict requirement of real-time, the high resolution image and multiple targets, many complex contour-based tracking algorithms are difficult to be used in our experimental system. Shape context has been employed in the contour-based tracking. Firstly, the spot is represented by a set of points sampled from the contour. Then, the shape context is calculated by the log-polar histograms (5 bins for log$r$ and 12 bins for θ, which are the same as [28]). The point correspondences are obtained by the matching cost function with χ^2 distance, as shown in Fig 11. Finally, the similarity based on shape distance could be obtained. The translation and rotation between two frames are also calculated by the affine transformation. The contour of each spot is saved as the prior shape before matching. Each spot could be recognized by shape similarity between the current shape and the prior one. Fig. 12 reveals the times of shape matching between each spot′s shape in arbitrary frame and the corresponding prior shape. The average matching times are 406.092 ms and 410.989 ms. It is time-consuming and hard to be accepted. Furthermore, when two spots mix with each other, the shape matching based on shape context will fail. Therefore, the tracking method based on shape context matching is not practical for our system, although the shape context represent the spot′s contour better. Considering the real-time requirement, the contour tracking scheme based on the chain code detection and the recognition based on the invariant image moments are more practical for our experimental system.

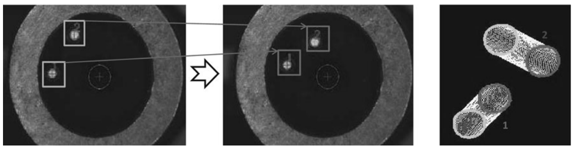


Fig. 11 Shape matching by the shape context between two frames

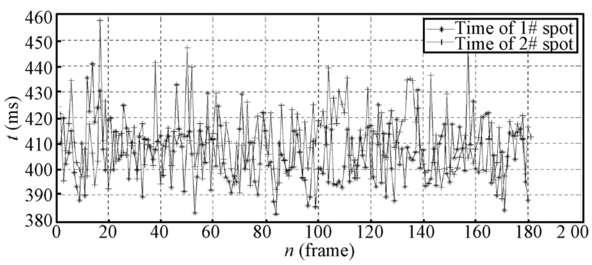


Fig. 12 Time of shape matching by the shape context in the sequential frames

In the second comparative experiment, the mean shift algorithm was employed, which is an efficient technique for tracking 2D blobs, such as the spots in this work. The mean shift method is a nonparametric statistical one for tracking with an isotropic kernel[29, 30]. Here, the kernel function is simplified as a rectangular one. In the tracking procedure, the mass center is computed. Meanwhile, a mean shift vector is determined and the searching window center shifted to the current mass center accordingly. This procedure is repeated until the number of iterations reaches the specified one or until the shifting distance was less than the specified one. Since the spot′s shape changed not too much, here the size and orientation of the tracking window do not change during the tracking procedure. In addition, the searching window is set to a rectangle region of 264 × 264 in pixels. The average searching time is 1.679 ms and the whole tracking time including pre-processing is 39.848 ms. Compared to the proposed tracking scheme, the times are almost similar. However, it is hard to continue tracking when multiple spots are close to each other, as shown in Fig. 13. Although this method has the better tracking time performance, it cannot handle the case of occlusion. Therefore, this disadvantage limited its applications on multiple spots tracking.

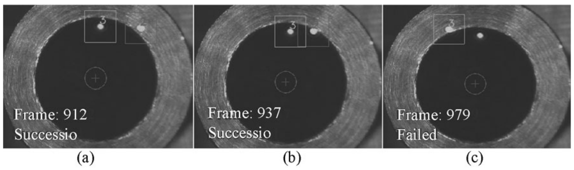


Fig. 13 Tracking image sequences by the mean-shift. When two spots mix with each other, the tracking is failed. (a)−(c) Image sequence frames 912, 937 and 979 are shown.

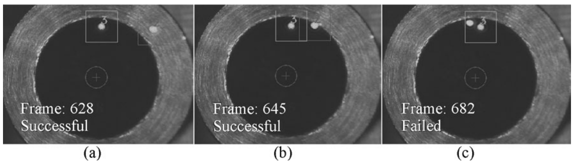


Fig. 14 Tracking image sequences by polar coordinate scanning. When two spots mix with each other, the tracking was failed. (a)−(c) Image sequence frames 628, 645 and 682 are shown.

Table 1 The comparative experiment results

Spots tracking	Average tracking time
Chain code	1.247 ms(39.672 ms)
Mean shift	1.679 ms(39.848 ms) (ROI: 264 × 264)
Polar coordinate	2.411 ms(43.054 ms) (ROI: 264 × 264)
Shape matching	Average matching time
Invariant moments	Very short
Shape context	412.253 ms

In addition, on the condition that the spot′s shape is approximately treated as a circle, the contour can be detected by scanning along the polar coordinate. The least square method combined with random sampling consensus technique is utilized to fit circle with the contour points. Then, the tracking window is updated by the new center of circle. The searching window is also set to a rectangle region of 264 × 264 in pixels. The average time of detecting the spot contour is 2.411 ms and the whole tracking time including pre-processing is 43.054 ms. By contrast, not only is its real-time performance is not better, but also it cannot handle the case in which multiple spots are very near each other, as shown in Fig. 14. In addition, when the spots shape change suddenly and cannot be treated as a circle, the tracking will be failed. The running time of the comparative experiment are concluded in Table 1.

Through many tests, the proposed contour tracking scheme proved to be robust under the adverse conditions, including rapid motions, large-scaled changes in velocity, slight vibrations of target and slight shaking of the laser spot. Therefore, the proposed tracking method is optimal and practical.

Finally, a series of images that show 3 spots guiding with PI controller are displayed in Figs. 15 (a)−15 (f). Three threads for guiding 3 laser beams execute synchronously. Three spots mix with each other when they are close to the target area. In this case, the contour tracking method combined with the Kalman filter is applied to determine the current position of each spot. The weight of the measurement is adjusted by the occlusion rate. Meanwhile, the position error of each spot between the current one and the desired one is treated as a feedback. Here, the valid shooting objective is a circular region with 150 pixels in diameter and the desired position is the center of the target′s end surface. The detection, tracking and guiding strategy have realized the high accurate shooting. The average time of 3 spots guiding is within 38 s, which meets the experimental system′s requirement.

6 Conclusions

In this paper, the main contribution is that a new detecting, tracking and guiding strategy for multiple laser spots based on beam and target alignment experimental system are developed. First, the laser spots are segmented and recognized by the contour-based analysis. Then, the contour tracking technique based on the chain code and Kalman filter prediction is proposed, in which the shape matching scheme is employed to distinguish different spots. Finally, a visual feedback system is introduced in order to guide 3 laser spots to shoot the desired position of the target. The experiment results show the rapid, accurate and robust performance of the laser shooting. In future, the new technique can be further combined with more data association approaches, such as multiple hypotheses tracking, especially, when multiple spots mix together or completely interfere with each other.

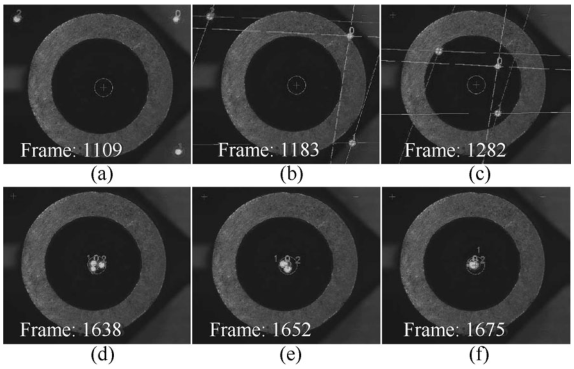


Fig. 15 The guiding procedure of 3 spots for beam and target alignment. (a)–(f) Image sequence frames of the spots guiding 1 109, 1 183, 1 282, 1 638, 1 652 and 1 675 are shown. The lines are the directions of spot motion only driven by the yaw or pitch motor. And in the experiment, the linear interpolation algorithm was applied to two motors.

References

[1] S. J. Boege, E. S. Bliss, C. J. Chocol, F. R. Holdener, J. L. Miller, J. S. Toeppen, C. S. Vann, R. A. Zacharias. NIF pointing and centering systems and target alignment using a 351-nm laser source. In *Proceedings of the SPIE 3047*, SPIE, Paris, France, pp. 248–258, 1997.

[2] P. Di Nicola, D. Kalantar, T. McCarville, J. Klingmann, S. Alvarez, R. Lowe-Webb, J. Lawson, P. Datte, P. Danforth, M. Schneider, J. M. Di Nicola, J. Jackson, C. Orth, S. Azevedo, R. Tommasini, A. Manuel, R. Wallace. Beam and target alignment at the national ignition facility using the target alignment sensor. In *Proceedings of the SPIE 8505*, SPIE, San Diego, USA, pp. 1–9, 2012.

[3] C. Stauffer, W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Fort Collins, USA, vol. 2, pp. 246–252, 1999.

[4] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition*, IEEE, Cambridge, UK, vol. 2, pp. 28–31, 2004.

[5] A. Yilmaz, O. Javed, M. Shah. Object tracking: A survey. *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[6] M. Yokoyama, T. Poggio. A contour-based moving object detection and tracking. In *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, IEEE, Tokyo, Japan, pp. 271–276, 2005.

[7] N. Paragios, R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.

[8] M. Fussenegger. Level Set Based Segmentation and Tracking of Multiple Objects, Ph. D. dissertation, Graz University of Technology, Austria, pp. 1–104, 2006.

[9] G. Wagenknecht. A contour tracing and coding algorithm for generating 2D contour codes from 3D classified objects. *Pattern Recognition*, vol. 40, no. 4, pp. 1294–1306, 2007.

[10] M. W. Ren, J. Y. Yang, H. Sun. Tracing boundary contours in a binary image. *Image and Vision Computing*, vol. 20, no. 2, pp. 125–131, 2000.

[11] Y. Rathi, N. Vaswani, A. Tannenbaum, A. Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 29, no. 8, pp. 1470–1475, 2007.

[12] A. Senior, A. Hampapur, Y. L. Tian, L. Brown, S. Pankanti, R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, vol. 24, no. 11, pp. 1233–1243, 2006.

[13] M. Isard, J. MacCormick. Bramble: A Bayesian multiple-blob tracker. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, IEEE, Vancouver, Canada, vol. 2, pp. 34–41, 2001.

[14] J. MacCormick, A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[15] L. Bazzani, D. Bloisi, V. Murino. A comparison of multi hypothesis Kalman filter and particle filter for multi-target tracking. In *Proceedings of the CVPR Workshop on Performance Evaluation of Tracking and Surveillance*, Miami, USA, pp. 47–54, 2009.

[16] A. Yilmaz, X. Li, M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004.

[17] N. Sebe, M. Lew. Robust shape matching. In *Proceedings of International Conference on Image and Video Retrieval*, London, UK, pp. 17–28, 2002.

[18] D. S. Zhang, G. J. Lu. Review of shape representation and description techniques. *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[19] S. Belongie, J. Malik, J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[20] P. KaewTraKulPong, R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance Systems*, Springer, Berlin, Germany, pp. 135–144, 2001.

[21] H. Freeman. On the encoding of arbitrary geometric configurations. *IEEE Transactions on Electronic Computers*, vol. 10, no. 2, pp. 260–268, 1961.

[22] INRIA, Visual Servoing Platform, [Online], Available: http://www.irisa.fr/lagadic/visp/image-processing.html, 2013.

[23] D. Xu, H. W. Wang, Y. F. Li, M. Tan. A new calibration method for an inertial and visual sensing system. *International Journal of Automation and Computing*, vol. 9, no. 3, pp. 299–305, 2012.

[24] M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis and Machine Vision*, Thomson Learning, Stamford, USA, pp. 750–805, 2007.

[25] M. K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

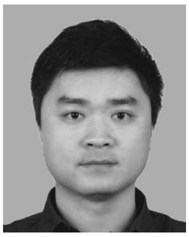
[26] G. Bishop, G. Welch, An introduction to the Kalman filter, [Online], Available: http://www.cs.unc.edu, TR95-041, UNC-Chapel Hill, USA, 2000.

[27] M. R. Rajamani, J. B. Rawlings. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, vol. 45, no. 1, pp. 142–148, 2009.

[28] Z. Liu, H. Shen, G. Y. Feng, D. W. Hu. Tracking objects using shape context matching. *Neurocomputing*, vol. 83, pp. 47–55, 2012.

[29] D. Comaniciu, V. Ramesh, P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[30] R. T. Collins. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Pittsburgh, USA, vol. 2, pp. 1–234, 2003.



Peng-Cheng Zhang received the B. Sc. degree in mechanical and automotive engineering from Hefei University of Technology, China in 2008, and the M. Sc. degree in mechanical and automotive engineering from South China University of Technology, China in 2011. He is currently a Ph. D. degree candidate in the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, China.

His research interests include target tracking and positioning, visual servoing, and computer vision.

E-mail: pengcheng.zhang@ia.ac.cn (Corresponding author)

ORCID iD: 0000-0003-2338-2788

De Xu received the B. Sc. and M. Sc. degrees in control science and engineering from Shandong University of Technology, China in 1985 and 1990, respectively, and the Ph. D. degree in control science and engineering from Zhejiang University, China in 2001. Since 2001, he has been with the Institute of Automation, Chinese Academy of Sciences, China, where he is currently a professor in the Research Center of Precision Sensing and Control.

His research interests include robotics and automation, the control of robots, such as visual control and intelligent control.

E-mail: de.xu@ia.ac.cn