Robust Human-Computer Interaction for Unstable Camera Systems

Hao ZHU††, Student Member, Qing YOU††, and Wenjie CHEN††(a), Nonmembers

SUMMARY A lot of vision systems have been embedded in devices around us, like mobile phones, vehicles and UAVs. Many of them still need interactive operations of human users. However, specifying accurate object information could be a challenging task due to video jitters caused by camera shakes and target motions. In this paper, we first collect practical hand drawn bounding boxes on real-life videos which are captured by hand-held cameras and UAV-based cameras. We give a deep look into human-computer interactive operations on unstable images. The collected data shows that human input suffers heavy deviations which are harmful to interaction accuracy. To achieve robust interactions on unstable platforms, we propose a target-focused video stabilization method which utilizes a proposal-based object detector and a tracking-based motion estimation component. This method starts with a single manual click and outputs stabilized video stream in which the specified target stays almost stationary. Our method removes not only camera jitters but also target motions simultaneously, therefore offering an comfortable environment for users to do further interactive operations. The experiments demonstrate that the proposed method effectively eliminates image vibrations and significantly increases human input accuracy.

key words: video stabilization, human-computer interaction, object proposal

1. Introduction

Human-computer interactions are necessary in many video systems. An automatic option may not be available in many cases. For example, an model-based detector requires prior information, like object category, appearance, motion patterns and so on, which may not be given in prior. For a motion-inspired detection method, proper foreground-background motion may not exist at any time to find moving targets. For a scene containing multiple objects with similar appearance, selecting a certain target also requires additional interactive operation. Therefore, manual target selection, like clicking and drawing bounding box, is still widely used for abundant real-world applications. For example, in a tracking function of a drone, users may need to select a target at the initialization process.

Obviously, one can comfortably conduct interactive operations on a stationary image and input very accurate information to vision systems. However, such an ideal interactive environment can not be reached for many real-world applications. Pausing a streaming video will bring additional risk and cost. If frames during the pausing process are dropped, the input information specified in the paused image may be inconsistent to the current target due to appearance change. If these frames are buffered, the vision systems will require additional memories and computations to absorb them. Furthermore, a lot of vision systems, like a real-time tracking platform, cameras and streams can not be stopped at all. All these restrictions force users to operate on streaming videos.

Cameras in conventional surveillance systems are always mounted on stable platforms and kept stationary. However, today video cameras have been embedded in mobile phones, vehicles, drones and et al. These video systems share a common ground that cameras are mounted in an unstable environment. Therefore camera jitters and platform vibrations can not be avoided, which will lead to unstable images in video streams. Figure 1 gives a representative example.

Unfortunately, human-computer interactions are sensitive to these jitters and vibrations. Unstable cameras will lead to inaccuracy input information, which may fail to meet the requirement of initialization process in many vision systems.

Not many works investigated the interaction accuracy in practical applications. In [1], the authors collect and model human clicks in different operating protocols. The results show errors in human inputs are unbearable to initialize object trackers. To compensate the input errors they proposed a proposal-based detection method for arbitrary
object categories. This work finds a bounding box given a human click. But it may fail to achieve a perfect region as user’s wish. For example sometimes it may output a region of one’s whole upper body rather than the head only. Therefore in many cases, users still need to give initial information by manually drawing an accurate bounding box. Obviously, drawing a box is more difficult, and takes more operations, than making one click on a video stream. Besides the camera jitters, target motion also leads to an inaccurate input. This is because one always needs some time to draw a box and a moving target may move away from the original position during this process.

In this paper, we collect hand drawn bounding box, which is one of the most widely used operation methods, by several people on streaming videos and give a deep look into the user inputs in practical vision applications. This is the only publicly available dataset of hand drawn bounding boxes. The collected data demonstrates a fact that erratic camera-object motion are fatal to human inputs. Figure 2 gives examples of human input distribution around the ground-truth region. We notice that one can hardly give accurate information on unstable videos, especially those captured by hand-held and vehicle-based cameras.

The errors in inputs of initialization process will degrade performance of many vision algorithms [1]–[4]. Therefore, based on the above observation video stabilization becomes an important and necessary part of video enhancement. Its task is to remove the undesired motions. One of the key step of video stabilization is to estimate camera motion, i.e. the transformation between adjacent frames. One way to calculate global motion is the global intensity alignment approach [5], [6]. Another way, which is more widely used, is matching and tracking features, patches or regions [7]–[11] along the video frames. A general pipeline of the matching-based methods can be summarized as: 1) extract and match features; 2) estimate the global motion based on the matched pairs; then 3) smooth motions and conduct motion compensation.

Some works also use other methods to remove camera jitters. In [12], the authors use gyrosensors to estimate camera motions. However this method can not work in systems without the certain sensors. Some researchers also exploit Kinematics and kinetic models, but it only works in the vehicle-based camera systems [13]. Another idea uses visual fixation mechanism [14]. But this approach does not take wanted camera motion into consideration.

Many video stabilization methods have been successfully applied in many applications, and some of them tried to offer more abilities based on practical requirements. Some research improved the video quality by offering full-frame image size [15]. It employed a motion inpainting process to fill in missing image parts by using data from neighboring frames. Another work served aerial surveillance systems by offering stabilized streams and detecting moving objects [7].

However, all the conventional methods concentrate on the camera motion and neglect object movement, which also results in deviations in interactive operations. To offer users perfect interaction environment, both of the camera motion and the target motion should be removed. In this paper, we design an interactive method to give accurate target information. To this end, we propose a target-focused video stabilization method. The framework consists of a proposal-based detection component and a tracking-based motion estimation component. This stabilization method starts from a single human click and removes camera vibrations and target motions simultaneously. In the stabilized video, moving objects keep consistent position in the image which makes users comfortable for further operations.

We also conduct experiments to evaluate the effectiveness of the proposed stabilization method and make additional comparisons of the human inputs on original streams and the inputs on stabilized videos.

The reminder of this paper is organized as follows: we discuss the error in manual inputs in Sect. 2. The stabilization method is presented in Sect. 3. Section 4 gives details about the our experiments. Then we make a conclusion in Sect. 5.

2. Hand Drawn Bounding Box

When one want to manually select an object, drawing a bounding box is one of the most widely used methods. However, the precision of achieved target regions may fail to meet the demands of visual algorithms for several reasons. Firstly, moving targets may lead to inaccurate initializations. During the drawing process targets may move away from the original position, which will lead to erroneous results. Secondly, erratic camera motion may cause deviations. It is caused by unstable mounting methods of cameras (like hand-held cameras and vehicle-based cameras). Furthermore, users may not be able to pause video stream due to real-time requirements and mechanical limitations, e.g. selective zoom feature for mobile phones and pan tilt zoom cameras.

To give a deep look into the manual inputs in real-life applications, we build an interactive interface on a desktop environment and collect bounding boxes drawn by human users. During video playing, the subjects can draw at any
time. After each input, the video will be re-initialized from a random frame and wait for the next operation. Therefore the inputs may happen in all frames. The frame rate is fixed at 25 frames per second (fps).

We collect bounding boxes on 10 sequences collected from two datasets. Among them 6 videos are captured by hand-held cameras from TB50 dataset\(^\dagger\), and 4 are captured by cameras mounted on UAVs from UCF datasets\(^\ddagger\).

All of them contain violently erratic motion which are extremely difficult for users to draw accurate target regions. We shown the manual inputs in Fig. 2. Obviously, human inputs suffer heavy deviations and it is difficult to obtain an accurate box on these unstable videos.

We evaluate the input accuracy using center error and overlap. Center error is defined as the distance between the ground-truth region center and the input box center. Since target size is different in all sequences, we normalize all target regions to 50×50 for fair comparison. In Table 1, MEAN is average value and SD is standard deviation. Results show that inputs on all sequences suffer large errors. These regions with more than 80% errors may be unacceptable to initialize many vision algorithms. For example in [2]–[4], extensive experiments have demonstrated that even less than 20% errors would lead to significant performance degradation for tracking algorithms.

And the reason of the input error is target motions during the drawing process. As shown in Fig. 3, users need three operations to draw a bounding box with a mouse: button-down, mouse-moving and button-up. We calculate the average time cost for drawing a bounding box, which is about 10 frames in the collected data. And the average target velocity is about 10.8 pixels per frame, which means the target could move more than 100 pixels away from the original position during the drawing process.

In a word, the camera-motion during the operating process leads to errors in the human inputs. We can safely conclude from the above analysis that, manual initialization on the videos containing erratic motions is not accurate enough for practical visual applications. To break this limitation, we propose a target-based video stabilization framework to remove camera and object motion simultaneously. This method starts from a manual click and offers stabilized video streams for accurate and sophisticated interactions.

3. Stabilization Pipeline

From the above observation, drawing bounding boxes on video streams, especially those containing erratic motions, is a challenging task in practical use. Since many computer vision applications need accurate initial information, we proposed a target-focused video stabilization method offering a stable human-computer interactive interface to achieve accurate manual inputs.

This work starts from simple interactive operation, i.e. a manual click, as input for an object proposal based object detection algorithm. The detection result initializes a tracking-based motion estimation process. Then the transformation between two images is described by a homography. Our method outputs a stabilized video stream, in which the target stays approximately stationary. This enables users to draw an accurate bounding box or do more complicated operations. The pipeline is shown in Fig. 4.

In the following parts of this section, we present the proposal-based object detection algorithm and the tracking-based motion estimation method.

3.1 Proposal-Based Object Detection

Here the task is to find a target region according to a single manual click, which is obtained from the interactive interface and therefore may contain deviations. Since the target may belong to arbitrary categories, the detection method should be suitable to detect different classes of objects. The detected region would be used to initialize the following tracking process. We use a proposal-based detection method, which is based on [1]. We revised the framework into a light-weight version to produce fast and approximate target regions.

Object proposal algorithm generates all object-like regions in an image[16]. We give each of the candidates a Gaussian confidence map which is weighted by the initial click. And all the confidence maps will be aggregated to

---

Footnotes:
\(^\dagger\)http://cvlab.hanyang.ac.kr/tracker_benchmark
\(^\ddagger\)http://crcv.ucf.edu/data/UCF_Aerial_Action.php

---

Table 1  Accuracy of hand drawn bounding boxes on unstabilized videos.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Center error MEAN</th>
<th>Center error SD</th>
<th>Overlap MEAN</th>
<th>Overlap SD</th>
<th>Duration (frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlurBody</td>
<td>39.50</td>
<td>30.78</td>
<td>0.25</td>
<td>0.15</td>
<td>10.52</td>
</tr>
<tr>
<td>BlurCar</td>
<td>21.08</td>
<td>12.01</td>
<td>0.29</td>
<td>0.14</td>
<td>10.35</td>
</tr>
<tr>
<td>BlurFace</td>
<td>28.02</td>
<td>18.52</td>
<td>0.27</td>
<td>0.16</td>
<td>10.90</td>
</tr>
<tr>
<td>BlurOwl</td>
<td>33.48</td>
<td>14.25</td>
<td>0.21</td>
<td>0.13</td>
<td>11.03</td>
</tr>
<tr>
<td>Couple</td>
<td>28.95</td>
<td>25.49</td>
<td>0.26</td>
<td>0.13</td>
<td>12.15</td>
</tr>
<tr>
<td>Human9</td>
<td>60.58</td>
<td>36.21</td>
<td>0.13</td>
<td>0.16</td>
<td>10.48</td>
</tr>
<tr>
<td>UCF-MAN1</td>
<td>60.99</td>
<td>57.70</td>
<td>0.11</td>
<td>0.07</td>
<td>12.98</td>
</tr>
<tr>
<td>UCF-MAN2</td>
<td>30.58</td>
<td>16.88</td>
<td>0.25</td>
<td>0.11</td>
<td>12.57</td>
</tr>
<tr>
<td>UCF-CAR1</td>
<td>27.93</td>
<td>21.39</td>
<td>0.34</td>
<td>0.18</td>
<td>13.95</td>
</tr>
<tr>
<td>UCF-CAR2</td>
<td>16.95</td>
<td>09.95</td>
<td>0.41</td>
<td>0.13</td>
<td>12.32</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>34.81</td>
<td>24.38</td>
<td>0.25</td>
<td>0.13</td>
<td>11.73</td>
</tr>
</tbody>
</table>

---

Fig. 3  Drawing bounding box on a moving target may lead to heavy deviations due to target movement. From left to right are three operations to draw a bounding box: button-down, mouse-move and button-up.
achieve a target likelihood map, based on which each candidate will have a likelihood score. We select the candidate with highest likelihood score as the crude target region \(B^*\). Therefore the detection task can be formulated as a maximum likelihood estimate.

Specifically, the likelihood score of each candidate \(L(B_i; c)\) can be computed by

\[
L(B_i; c) = \alpha_i \sum_{x \in B_i} T(x; B_i; c),
\]

(1)

where \(B_i\) is the \(i\)th window among \(N\) candidates and \(c\) is the input click position. \(\alpha_i\) is a normalization factor which is negatively correlated with the region size. \(T(x; B_i; c)\) indicates the value of pixel \(x\) on the target likelihood map, which is computed by

\[
T(x; B_i; c) = \sum_{i=1}^{N} \lambda(B_i|c)p(x|B_i)
\]

(2)

where \(\lambda\) is a Gaussian weight term centered at the initial click position, and \(p(x|B_i)\) is a Gaussian distribution which is modeled by each candidate window and centered at the box center. Here we simplify the framework in [1] by removing the saliency term and motion cue. Furthermore, only candidates containing the initial click are used in Eq. (2). As shown in the results in [1], these changes may cause performance decrease on some sequences.

Sometimes the proposal-based process may generate proper bounding boxes, but in some cases this method can hardly find perfect regions. Since only bottom cues, like gradients and boundaries, employed and no prior models included, the proposal method may be confused by sophisticated image structures. Some outputs may be partial regions of the whole object and some outputs may include undesired regions which are connected to the genuine targets. This is also why drawing a bounding box is still necessary in practical systems.

Although the detected region may not be the accurate part that one wants to select, the crude target region is proper to initialize the following tracking-based motion estimation process, and the inaccuracy will be absorbed by the final interactive operation on stabilized streams.

### 3.2 Tracking-Based Motion Estimation

Our method utilizes the crude target region, generated by the proposal-based detection, to initialize a tracking process. Instead of estimating global motion, which is generally used in other video stabilization methods [9], [10], [15], [17], we use the trajectory of the crude object region to estimate transforms between frames.

The stabilization process starts from frame \(j\) in which the initial click enters. The detected region \(B^*_j\) is set as an anchoring position and is also used to initialize a tracking process. For rigid objects, we can extract and describe feature points within \(B^*_j\), and then match them to features within a search region in frame \(j + 1\). Based on the matched feature pairs, an affine transformation matrix can be achieved. This pipeline is similar to previous works [11]. However, \(B^*_j\) is only a small region and may not contains enough feature points for the matching. Furthermore, for non-rigid objects, the matching process may produce a wrong transformation due to inconsistent motions between different object parts. Therefore, we directly track the image patch \(B^*_j\) in following frames to estimate the transformation.

In previous works, global transformation is usually represented by a \(3 \times 3\) homography matrix. In our work, the global motion is achieved by tracking results. In our framework, we use the method tracking-learning-detection (TLD) [18] as the tracker. Then the transformation can be represented by a 2-parameters translation matrix,

\[
H_{(j,j+1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_k & y_k & 1 \end{bmatrix},
\]

(3)

where \((x_k, y_k)\) is the offset vector between the centers of \(B^*_j\) and \(B^*_{j+1}\). For a new frame \(n\), we calculate a cumulative transformation,

\[
H_{(j,j+1)} = H_{(j,j+1)}H_{(j-1,j)}.
\]

(4)

Then each pixel \(p = (x_k, y_k, 1)\) can be projected by

\[
p' = p \times H_{c}
\]

(5)
Motion estimation may fail due to tracking failures. In this case, we stop updating cumulative transformation matrix and a model-based object detector will search the target in whole image. TLD includes a cascaded detector which is trained and updated in the tracking process. When the target re-appears in the image region, the detector will re-initialize the tracking process. Consequently the stabilization process is also recovered from the same frame and the cumulative transformation update is continued.

Through the tracking-based motion estimation, the stabilized images are anchored at the target position. Both camera motion and target motion are removed so that one may feel like the target is stationary in the image region. This is ideal environment for human computer interactions, based on which users can draw accurate target regions.

4. Experiments

4.1 Experimental Settings

To evaluate the performance of the proposed video stabilization method, we test it on 10 video sequences collected from TB50 and UCF as explained in Sect. 2. All of these sequences contain erratic image vibrations and are challenging to achieve accurate manual inputs. In our implementation, we employ edgebox [16] and use 1000 candidates for each frame in the proposal-based detection component and select TLD [18] as the tracker in the motion estimation process. All parameters in edgebox and TLD are default settings.

In this section, we also compare the proposed method with a traditional stabilization approach [10], which estimates global camera motion by feature matching. To test the effectiveness of the proposed method for improving human computer interactions, we collected hand draw bounding boxes under the same protocol as explained in Sect. 2. We collected more than 1000 human inputs and randomly selected equal number of inputs for each sequence. Altogether 600 hand drawn bounding boxes are used in our evaluations.

The experiments are performed using MATLAB on a laptop with i5-4210U @1.7GHz and 8GB RAM. The proposal-based detection process takes 0.5s on average. The detection algorithm can be further parallelized [1] and the proposed method is flexible in selecting tracking algorithms. Our stabilization process can run at about 20 fps. Most time is consumed by TLD tracking and the image transformation process has linear time complexity. Furthermore, an implementation programed by more efficient language, like C/C++, can further accelerates the proposed method. Therefore the proposed method has a high potential to achieve real-time performance.

4.2 Stabilized Results

We show the stabilization results in Fig. 5. The cross center is position of the ground-truth region center in the first frame of stabilization process. The black blanks clearly demonstrate the jitters in these streams. The proposed method not only removes the camera motion, but also removes the object motion by the tracking-based motion estimation. The stabilized videos gives stationary targets in the images. Their positions are consistent through the video stream.

An exception is the result of Human9 as shown in Fig. 5. The target gradually and smoothly moves away from the anchoring position. This is caused by tracking drift. After the tracking region drifts from the target to background, the video is stabilized by estimating the background movement. In Sect. 4.4, we collect practical hand drawn bounding boxes and demonstrate the drifted results still offer stable human computer environment.

4.3 Comparison with Traditional Method

We compare the traditional method and the proposed method using two measures: the peak signal-to-noise ratio (PSNR), which reflects the quantification of global stability of the whole frame and can be considered as a measure of the similarity between two consecutive images, and target drifting error (TDE), which evaluates the stability of the tracking target and is defined as the distance between the target position in current frame and the original position in the first frame where the stabilization process starts. PSNR between frame $I_{t-1}$ and $I_t$ is calculated by

$$\text{PSNR}(I_{t-1}, I_t) = 10 \log \frac{255^2}{\text{MSE}(I_{t-1}, I_t)},$$

where the mean squared error (MSE) is a measure of the average departure per pixel in the overlapped region [19]. Our method starts stabilization when the human click comes, therefore in the evaluation section we use frames after stabilization begins. We calculate average values on all sequences and the results are given in Table 2 (PSNR: larger is better, TDE: small is better). The PSNR and TDE having suffix –A are based on all frames after stabilization starting in the video sequences, while the values with suffix –1 are based on the frames between the stabilization starting and the first matching failing. PSNR-1 and TDE-1 exclude the influence of reset mechanisms and enable a more fair comparison.

PSNR results show that both of the proposed and the traditional methods remove the jitters caused by camera shakes at comparable level. However, when evaluated by TDE, our method is far ahead to the traditional one. This is because traditional method removes global camera motion but omits the local target motion. However, target movements in stable videos still degrade the human input accuracy. In perfect interactive environment, target-image position should be consistent. As shown in Fig. 5, although most targets move all the time, their centers are still anchored at the initial positions. Undoubtedly, drawing bounding boxes on a more stationary object is more comfortable and easier for users.

The proposed method significantly improves the TDE
Fig. 5  Stabilization results. We show stabilized images for every 100 frames, except for Couple, UCFCar2 and UCFMan2 the step length is 50 frames since these streams are shorter than the others.

Table 2  Stabilization performance.

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>traditional</th>
<th>proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR-1</td>
<td>32.89</td>
<td>34.96</td>
<td>34.88</td>
</tr>
<tr>
<td>PSNR-A</td>
<td>32.21</td>
<td>33.44</td>
<td>34.33</td>
</tr>
<tr>
<td>TDE-1</td>
<td>46.40</td>
<td>76.08</td>
<td>16.54</td>
</tr>
<tr>
<td>TDE-A</td>
<td>77.70</td>
<td>121.14</td>
<td>22.68</td>
</tr>
</tbody>
</table>

score without sacrificing PSNR performance. This is because both of the proposed method and the method in [10] remove the camera motions, which affects PSNR, while the proposed method simultaneously removes target motions, which affects TDE. There should be an uncommon exception that a shaking target can make the proposed method suffer a worse PSNR. However, even in this case, a low TDE will also help one to make an easy operation.

We also plot the target center offset in Fig. 6. We can clearly see the consistency of target position generated by our method. For traditional method, the abrupt jumps are caused by stabilization re-initialization due to feature matching failures.

4.4 Influence on Interactive Operations

As we declared in the above sections, the proposed stabilization method aims to offer the stable interactive environment which allows accurate manual input even on erratic-motion video streams. Therefore to test the accuracy change of human inputs on stabilized videos, we collect practical user input under the same protocol in Sect. 2.

As shown in Fig. 7, the human inputs suffer heavy deviations on original video stream caused by camera jitters and object motion. On the stabilized streams, users can easily obtain accurate bounding boxes which can cover the right target region with higher overlap. We also give comparisons in numbers in Fig. 8. Here we normalize all target regions to $50 \times 50$ to enable fair comparison.

The deviations are serious on the original streams. After the stabilization process, center error drops significantly, from 34.81 to 6.85 pixels on average, as shown in Fig. 8.
Fig. 6 Offset comparison of target centers to the original position. The peaks on the green lines are caused by stabilization re-initialization.

Fig. 7 Comparison between hand drawn boxes on the original streams and on the stabilized streams. The left images show manual inputs collected on the original videos and the right ones show results on the stabilized videos. Accuracy improvement is clear to perceive in all video sequences.

Among them, Human9 and UCF-Man1 are the most challenging sequences before stabilization while their accuracy increases most by 51 and 50 pixels respectively. The decreased center error benefits from the consistent target positions in the stabilized videos. We also notice that users feel less worried to spend more time on drawing boxes for stationary objects. The average drawing time increases from 11.73 to 15.76 frames while the average overlap increases from 0.25 to 0.64 after stabilization.

Note in Fig. 8, input errors are different on different sequences. Human9, UCF-Man1 and BlurBody are most difficult for interactive operations. They have larger center error
Fig.8 The accuracy of hand drawn bounding boxes. The blue bars represent accuracy on the original video streams and the orange bars are the stabilized streams. Both center error and overlap demonstrate significant improvement after stabilization for human input.

than the other sequences. However, our stabilization process is effective and human inputs achieve similar improvements on all these difficult situations.

We can confidently conclude that the proposed stabilization method is significantly beneficial to human input accuracy.

5. Conclusion

In this paper, we make a deep look into the human-computer interactive operation in unstable vision systems. We collected hand drawn bounding boxes and analyzed the reason causing the input error. To offer a robust interactive environment for unstable videos, we proposed a target-focused video stabilization method which can simultaneously remove the camera jitter and object motion. Experiments show that the input error is significantly reduced almost 80% and overlap is increased more than 2x. To sum up, this work offers an easy and robust interactive solution in real-life applications.

References

Hao Zhu is currently working at Hikvision Digital Technology Co., Ltd. He received his PhD degree from Beijing Institute of Technology in 2017. He received his BS degree in the School of Automation from Nanjing University of Science and Technology in 2011. He was a visiting student in the Research School of Engineering, Australian National University funded by a scholarship from the China Scholarship Council from September 2014 to September 2015. His current research interests include visual tracking and video segmentation.

Qing You is a M.S. candidate at the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology. He received his BS degree in the School of Automation from Beijing Institute of Technology in 2016. His current research interests include computer vision, machine learning and deep learning.

Wenjie Chen received the B.S. degree in Physics from Xi’an Jiaotong University, Xian, China in 1996, the M.S. degree in electronic engineering from the Beijing Institute of Technology, Beijing, China in 2000, the Ph.D. degree in electronic engineering from the Beijing Institute of Technology in 2005. Now he is an associate professor in the Beijing Institute of Technology. His research interests include machine learning, computer vision, image processing, pattern recognition, and Kalman filtering.