DOI: 10.3969/j. issn. 1673-5188. 2017. 01. 009
http://kns.cnki.net/kcms/detail/34.1294.TN.20170306.1830.002.html, published online March 6, 2017

Moving Target Detection and Tracking for Smartphone Automatic Focusing

HU Rongchun, WANG Xiaoyang, ZHENG Yunchang, and PENG Zhenming

(School of Opto-Electronic Information, University of Electronic Science and Technology of China, Chengdu 610051, China)

Abstract

In this paper, a non-contact auto-focusing method is proposed for the essential function of auto-focusing in mobile devices. Firstly, we introduce an effective target detection method combining the 3-frame difference algorithm and Gauss mixture model, which is robust for complex and changing background. Secondly, a stable tracking method is proposed using the local binary patter feature and camshift tracker. Auto-focusing is achieved by using the coordinate obtained during the detection and tracking procedure. Experiments show that the proposed method can deal with complex and changing background. When there exist multiple moving objects, the proposed method also has good detection and tracking performance. The proposed method implements high efficiency, which means it can be easily used in real mobile device systems.

Keywords

moving target detection; frame-difference method; background modeling method; camshift tracking; meanshift tracking; auto-focusing

1 Introduction

Smartphones have become an important part in model life. Most of the daily scenes contain changing background and diverse moving objects, which causes blur and low imaging quality. To get images and videos with high quality, people have paid much attention to the auto-focusing technique of camera in mobile devices.

In this paper, we propose a non-contact cell phone camera auto-focus method, which contains object detection in captured video, intelligent tracking and camera focus. For the moving object detection module, we adopt the 3-frame difference method and Gauss mixture model (GMM) [1]. For the intelligent tracking module, we adopt the optimized camshift algorithm [2], and the detected target of the front-end module is used as an input box to realize the intelligent tracking of multiple targets. The focus module conducts auto focus based on the coordinates provided by the object detection and tracking module, which is a non-contract focus method. The whole detection and tracking procedure can provide accurate object locating in real-time, and has robustness against diverse scene.

2 Target Detection

We present an object detection method combining the 3-frame difference method and GMM. The 3-frame difference method is suitable for detecting moving objects from statistic background, while the GMM method is suitable for dynamic background.

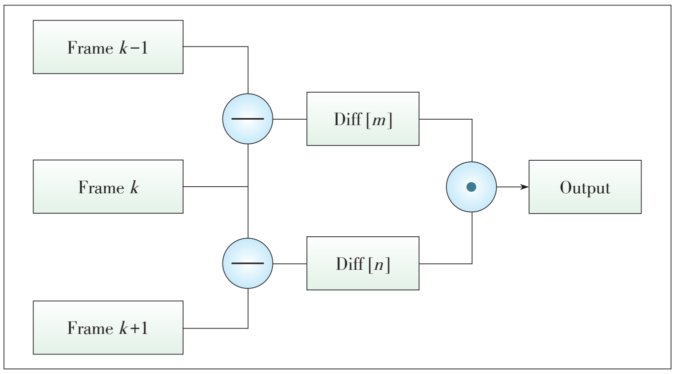
2.1 Three-Frame Difference Method

As we all know, frame difference is an effective way to detect moving objects at a low time costing level. The frame difference method, which detects the moving objects in video sequences by calculating the differences between two or more frames, can fully represent the feature of moving objects. Our proposed 3-frame difference method is shown in **Fig. 1**.

In Fig. 1, Diff[m] denotes the frame difference of fame k-1 and frame k, while Diff[n] denotes the frame difference of fame k+1 and frame k. After the frame process, the output is the dot product of Diff[m] and Diff[n]. **Fig. 2** shows the foreground detection results of the refined 3-frame difference method.

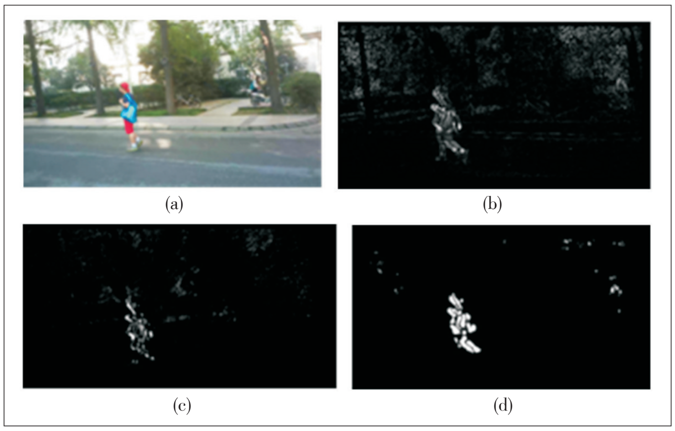
2.2 Gauss Mixture Model

The frame difference method can get an estimation of moving objects in videos. However, it is also sensitive to light change and noise corruption. The Gauss mixture model is an adaptive background modeling method, which has good perfor-



▲Figure 1. The refined 3-frame difference method.

This work was supported by ZTE Industry-Academia-Research Cooperation Funds.



▲Figure 2. Object detection results in the foreground: (a) Input frame; (b) by 3-frame difference; (c) by multiplying 2-frame difference; and (d) the binary image.

mance in complex changing scenes.

GMM represents every pixel by a mixture of Gauss models with different parameters. The number of Gauss models is set empirically. Each Gauss model has a weight value, which varies during the modeling procedure. By adjusting the parameters of Gauss models and the weight value, GMM can deal with slightly changing background and noise interruption. For every pixel in an image, the probability density function of GMM is defined as follows:

$$f\left(X_t = x\right) = \sum_{i=1}^{K} \omega_{i,t} * \eta\left(x, \mu_{i,t}, \Sigma_{i,t}\right), \tag{1}$$

$$\eta\left(x, \mu_{i,t}, \Sigma_{i,t}\right) = \frac{1}{(2\pi)^{\frac{n}{2}}\left|\Sigma_{i,t}\right|^{1/2}} e^{-\frac{1}{2}\left(X_t - \mu_{i,t}\right)^T \Sigma_{i,t}^{-1}\left(X_t - \mu_{i,t}\right)}, i = 1,2,...,K, \tag{2}$$

where X_t is the color of pixel. K is the number of Gauss distribution chosen to model the current image, $\eta\left(x, \mu_{i,t}, \Sigma_{i,t}\right)$ is the ith Gauss distribution at the time t, with an average value $\mu_{i,t}$ and a covariance value $\Sigma_{i,t}$. $\omega_{i,t}$ is the weight value of the ith Gauss distribution at time t, with $\sum_{i=1}^{K} \omega_{i,t} = 1$.

As the parameters of the mixture model of each pixel change, we can determine which of the Gaussians of the mixture are most likely produced by background processes. Here we choose the Gaussian distributions that have the most supporting evidence and the least variance. The distributions numbered 1 to B are chosen as the background model and B is expressed as

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

where T is a measure of the minimum portion of the data, which should be accounted for by the background. By identifying the background, we get an estimation of moving objects. **Fig. 3** shows the background modeling result of GMM. In Fig. 3b, the object information can easily be seen. It is not a pure background image due to the fact that there are no enough image frames to make a good mixture Gauss model. In Fig. 3c, the background is clear and with no object information. It is because that the mixture Gauss model is well build after enough image frames.

2.3 Target Detection in Foreground

After the separation of background and foreground by the above two methods, we need eliminate the false alarms in foreground images.

The post-processing of foreground images includes noising smoothing [3], threshold segmentation [4] and morphological processing including erosion and dilation [5]. After this procedure, the processed image becomes a binary image, in which 0 represents the background pixel and 1 represents the object pixel. Here we get the binary images obtained by both the frame difference method and GMM. The final binary image is obtained by combining them together:

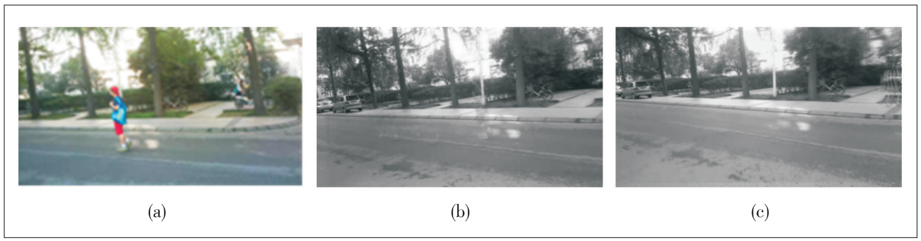
$$Result = Output\ of\ Diff\quad Output\ of\ GMM, \tag{4}$$

where $Output\ of\ Diff$ is the binary image from the frame difference method, and $Output\ of\ GMM$ is the binary image of the GMM detection. Then we process the region in which the pixels are equal to 1 by edge detection. It gives the coordinate of the pixel block's boundary and section area. The coordinate can be used to realize auto-focusing.

We introduce the temporal filtering theory before the final detection results are output. We also conduct a coincidence comparison to the candidate foreground area which is detected in several continuous frames. The target detected in several continuous frames is determined as the true target while the others are treated as false alarms.

3 Smart Tracking

The smart tracking module includes the feature extraction of moving targets and target tracking. The key point feature is selected to perform the tracking procedure. As to the tracker, we choose the camshift tracking algorithm because it makes good



▲Figure 3. The background estimation results of GMM: (a) The original image; (b) background estimation 1; and (c) background estimation 2.

use of the color information of objects. However, the camshift does not take the texture and spatial structure of objects into consideration, which may cause inaccurate tracking result. Therefore, the Local Binary Pattern (LBP) feature descriptor [6] is introduced to the traditional camshift tracker, which builds a new effective joint histogram model of target appearance.

3.1 Camshift Algorithm

The basic idea of the camshaft, a continuous adaptive mean-shift algorithm [7], is to process all the frames with a mean-shift operator, and the last frame result (the central location and the window size of the search window) is regarded as the initial value of the mean-shift's searching window in next frame. Then the iteration continues. Assume that the video consists of n frames, the processing steps are shown in **Algorithm 1**.

Algorithm 1 Camshift Tracking

```
Input:
    Labeled target area in the first frame;
Output:
    Coordinate of targets in each frame.
1. Initialize: select the target area in the video;
2. Calculate 2D color probability distribution in the selected area;
3. for $k$=1:$n$ do
4.  Tracking the selected object using the meanshift tracker;
5.  Calculate the object coordinate in the current frame;
6.  Mark the object;
7. end for
```

3.2 LBP Operator

LBP is an operator to describe the partial texture features of an image. LBP at the coordinates (x,y) can be calculated as

$$LBP(x,y)=\sum_{n=0}^{p-1}2^n\,\mathrm{sgn}(i_n-i_c)\,, \tag{5}$$

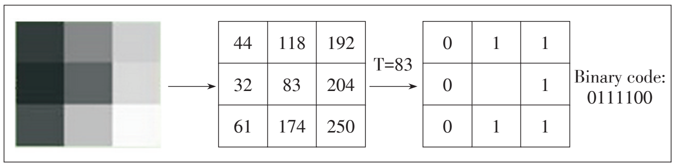
where i_c denotes the gray scale value of a pixel at the coordinates (x,y), i_n denotes the adjacent pixel's gray scale value, and p denotes the number of adjacent pixels. We usually adopt the 3×3 window with p=8. The sign function (sgn) is written as

$$\mathrm{sgn}(x)=\begin{cases}1 & (x\geqslant 0)\\ 0 & (x<0)\end{cases}. \tag{6}$$

Fig. 4 shows the processing steps of LBP, the LBP texture map is then obtained.

3.3 Target Tracking Algorithm Based on Combined Color and Texture Features

Algorithm 2 describes target tracking based on the combined color and texture feature.



44	118	192
32	83	204
61	174	250

0	1	1
0		1
0	1	1

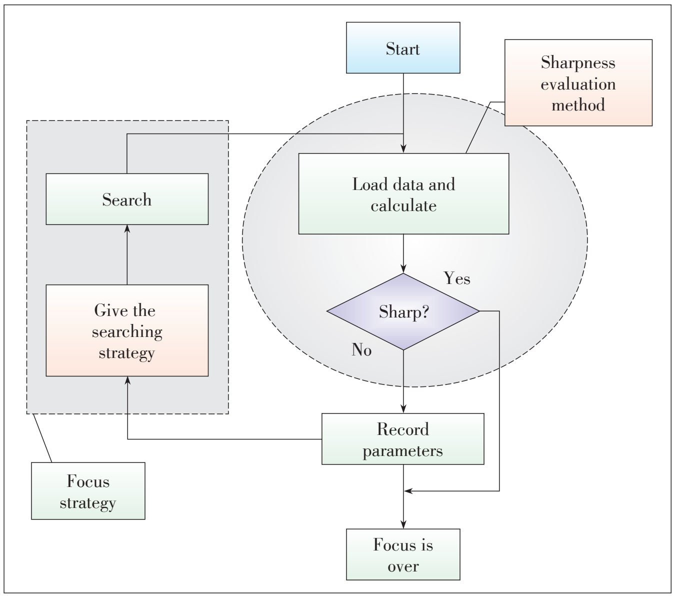
▲Figure 4. The processing steps of LBP feature.

Algorithm 2 Target tracking algorithm based on combined color and texture feature

```
Input:
The selected moving target;
Output:
Coordinate of targets in each frame.
1. Initialize: The size and position of each window;
2. for $k$=1:$n$ do
3.  Extract the texture feature;
4.  Use the texture histogram with the image to get back projection;
5.  Get the texture probability distribution map;
6.  Carry out AND operation of the texture probability distribution map and the tone probability distribution;
7.  Camshift tracking;
8.  Update the size of search area;
9. end for
```

4 Auto-Focusing

When the target detection and tracking are completed, we need to conduct auto-focusing (**Fig. 5**) according to the given coordinates from the detection. An original focal length is first



▲Figure 5. The flow chart of auto-focusing.

given to calculate the sharpness value according to the sharpness evaluation method. The mountain climbing searching (MCS) algorithm is then conducted until the auto-focus system finds the sharpest target.

4.1 Focusing Sharpness Evaluation

The high-frequency components show the detail of an image. However, the first-order difference operator is not really sensitive to the high-frequency components, but works well on detecting the low-frequency components such as the target's outline. Our proposed method adopts the refined 8-neighbour Laplacian operator [8]. The Laplacian operator is an edge detecting operator defined by the x, y second order partial derivatives of the $f(x, y)$ image, which can be described as an approximate Laplacian operator template:

$$L_4(x,y)=\nabla^2\approx\begin{bmatrix}0&1&0\\1&-8&1\\0&1&0\end{bmatrix}, \qquad (7)$$

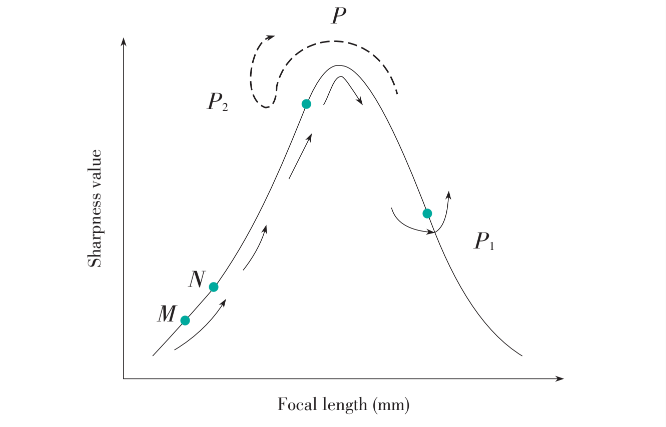
where ∇^2 is the second derivative differential operator for the Laplacian operator.

The focusing sharpness evaluation function is then given by

$$J_{laplacian8}=\sum_M\sum_N L_4(x,y)\,. \qquad (8)$$

4.2 Focusing Searching Strategy

Our method uses MCS [9] to focus the search. First, an initial focus value M is given. The sharpness value $J(M)$ is calculated by the focusing sharpness evaluation method, and the focal length value is changed for calculating a new value N and its sharpness $J(N)$. $J(M)$ and $J(N)$ are then compared. At the same time, the search direction is determined and the iterative search is continued based on the side on which the sharpness value is greater. In **Fig. 6**, we search from point M to point N. In the searching process, the searching results such as focal length and the sharpness J are saved for the next iteration. A search climbs in one direction until the sharpness value reaches the maximum and begins to decline, and the searching process for the first time follows the solid line $M-N-P-P1$ in Fig. 6. Then the second search starts from $P1$ and makes a reverse lookup until the sharpness value begins to decline over the peak once again. The dashed line $P1-P-P2$ reflects the second searching process. Every time when the searching is done, we reduce the searching step in the next iteration. The iteration is repeated until the maximum focus value is found, and then the focusing is over.



▲**Figure 6. The mountain climbing search method.**

5 Experiments

Our experiments were conducted at a DELL T5600 power station (CPU: Intel XeonE5 - 2603, 1.8GHz. Memory: 8GB. GPU: NVIDIA Tesla C2075), with which the process reached 25 *fps*.

We selected several specific videos that contain complex backgrounds and multiple targets. In this way, we examined the performance of the proposed algorithm under different circumstances.

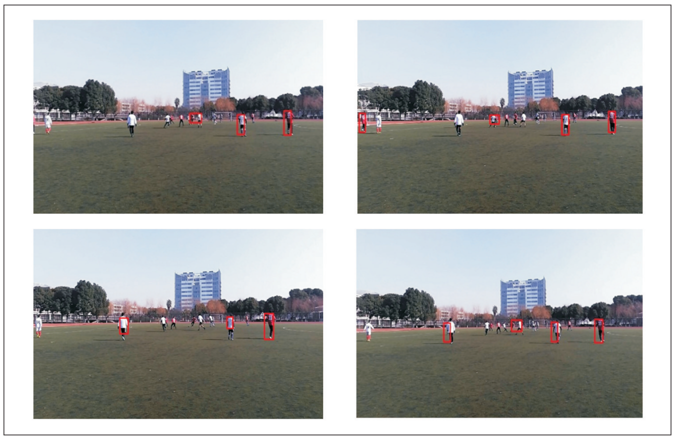
5.1 Target Detection

We captured one or more targets in the video sequences for target detection. The pictures in **Fig. 7** show that the football players were moving fast and the background contains trees, buildings and the sky. In this case, the moving targets could be easily detected by the proposed method (according to their sizes and positions).

To evaluate the performance of target detection, we introduced the detection accuracy rate (DAR) as a metric, which is defined as

$$DAR=\frac{Target\ detected\ frame\ amounts}{Frame\ amount}\,. \qquad (9)$$

The DAR under single-target and multi-target modes for test videos are shown in **Tables 1** and **2**. The multi-target mode has a higher DAR than the single target mode, although the latter reaches a high DAR. This experiment demonstrates the effectiveness of the proposed detection method. The combi-



▲**Figure 7. Multi-target detection.**

▼Table 1. Performance of target detection (single target)

File name	corridor.avi	office1p.avi	office2p.avi	outdoor.avi
Frame number	386	440	430	854
Target-detected frames	332	348	378	769
DAR	86%	79%	88%	90%
DAR: detection accuracy rate				

▼Table 2. Performance of target detection (multi-target)

File name	basketball.avi	football.avi	football2.avi	spring.avi
Frame number	1098	1165	2035	1655
Target-detected frames	1010	1048	1933	1522
DAR	92%	90%	95%	92%
DAR: detection accuracy rate				

nation of the frame difference method and GMM achieved good performance in highly complex background with multiple moving targets.

5.2 Smart Tracking

The tracking strategy is that one target was selected to track manually or automatically (according to the target's size or position) among the detected targets (**Fig. 8**). Then the tracking is kept until a new detection starts.

Similar with the target detection evaluation metric, we introduced the tracking accuracy rate (TAR), which is defined as

$$TAR = \frac{Target\ tracked\ frame\ amounts}{Frame\ amount}. \qquad (10)$$

Table 3 shows the *TAR* of four different videos. As we can see, the refined camshift method can track well in changing background.

5.3 Combining Detection with Tracking

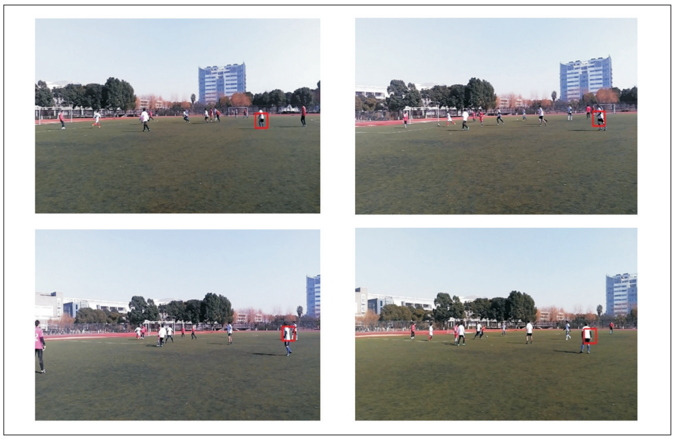
In our final combining test, the two detection and tracking modes were used to evaluate the performance of the proposed algorithm. The two evaluation modes are as follows.

1) MT Mode: It is for the multi-target scene. All the targets in the video sequences are detected and one of them is selected as the tracking object based on its size or position. The accuracy rate of target detection relies on the multi-target detection algorithms, while the tracking accuracy relies on the single target tracking algorithm.

2) S Mode: It is for the single target scene. The target is detected and tracked automatically.

Table 4 shows the performance of the two evaluation modes.

In practice, the algorithm efficiency has great influence on the performance of an auto-focusing system. The proposed method is more than 25 *fps* and has reached real-time requirements. **Table 5** shows the *fps* of each tested video under the MT and S modes. We can see that the proposed detection and tracking algorithm performs better in the S mode due to



▲Figure 8. The smart tracking is very smooth and robust.

▼Table 3. Performance of target tracking (single target)

File name	corridor.avi	office1p.avi	office2p.avi	outdoor.avi
Frame number	386	440	430	854
Target-tracked frames	379	426	427	828
TAR	98%	97%	99%	97%
TAR: tracking accuracy rate				

▼Table 4. Performance of target detection and tracking

File name	Scene type	MT mode accuracy rate	S mode accuracy rate
corridor.avi	Single target	-	>95%
office1p.avi	Single target	-	>95%
office2p.avi	Single target	-	>95%
outdoor.avi	Single target	-	>95%
runner.avi	Single target	-	>95%
basketball.avi	Multi-target	90%	-
football.avi	Multi-target	90%	-
football2.avi	Multi-target	90%	-
penquan.avi	Multi-target	90%	-

▼Table 5. Efficiency of target detection and tracking

File name	Scene type	*fps* of MT mode	*fps* of S mode
corridor.avi	Single target		>178
office1p.avi	Single target	-	>175
office2p.avi	Single target	-	>141
outdoor.avi	Single target	-	>200
runner.avi	Single target	-	>178
basketball.avi	Multi-target	>55	-
football.avi	Multi-target	>70	-
football2.avi	Multi-target	>65	-
penquan.avi	Multi-target	>55	-

the single target property. In the MT mode, the efficiency of the proposed method is also higher than 25 fps.

6 Conclusions

In summary, the proposed method works well on detecting and tracking moving objects in mobile phone video sequences. The proposed target detection method is based on the 3-frame difference method and GMM. The tracking method is a combination of the LBP feature and camshift tracker. Auto-focusing is realized by using the coordinates of the detection and tracking modules. The proposed method can perform well in many multi-target scenes. The experiments of detection and tracking show that the proposed method can be used to achieve the function of non-contact auto-focusing in mobile devices.

References

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for Real-time tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, USA, Jun. 1999, pp. 246–252. doi: 10.1109/CVPR.1999.784637.

[2] K. Nayebi, T. P. Barnwell, and M. J. T. Smith, "Time-domain filter bank analysis: a new design theory," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1412–1429, 1992.

[3] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. Eleventh ACM International Conference on Multimedia*, Berkeley, USA, Nov. 2003. doi: 10.1145/957013.957017.

[4] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, Aug. 2002. doi: 10.1109/TIP.2002.1014998.

[5] J. M. Menon and L. J. Stockmeyer, "Garbage collection in log-structured information storage systems using age threshold selection of segments," U.S. Patent No. 5,933,840, Aug. 3, 1999.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, USA: Prentice Hall, 2002.

[7] X. Tan and B. Triggs, "Fusing Gabor and LBP feature sets for kernel-based face recognition," *Analysis and Modeling of Faces and Gestures*. Berlin, Germany: Springer Berlin Heidelberg, 2007, pp. 235–249. doi: 10.1007/978-3-540-75690-3_18.

[8] G.-Y. Gong, W. Z. He, and X.-H. Gao, "Optimized mountain climb-searching of auto-focusing in infrared imaging system," *Laser & Infrared*, vol. 11, no. 026, 2007.

[9] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp.603–619, 2002. doi: 10.1109/34.1000236.

Manuscript received: 2015-11-16

Biographies

HU Rongchun (hrc@swust.edu.cn) received his master's degree in information and communication engineering from University of Science and Technology of China (UESTC) in 2007. He is a lecture and pursuing the Ph.D. degree in signal and information processing at UESTC. His research interests include machine learning and image processing.

WANG Xiaoyang (xywang_2012@163.com) received her B.E. degree in electronic science and technology from University of Electronic Science and Technology of China. She is currently a Ph.D. candidate in signal and information processing there. Her research interests include image processing, computer vision, and compressive sensing theory and applications.

ZHENG Yunchang (zhengyunchang@foxmail.com) received the B.E. and M.S. degrees in electronic science and technology from University of Electronic Science and Technology of China. His research interests include machine learning and computer vision.

PENG Zhenming (zmpeng@uestc.edu.cn) received his Ph.D. degree in geo-detection and information technology from Chengdu University of Technology, China in 2001. From 2001 to 2003, he was a postdoctoral researcher with the Institute of Optics and Electronics (IOE), Chinese Academy of Sciences. He is a professor with University of Electronic Science and Technology of China. His research interests include image processing, radar signal processing, and target recognition and tracking. Prof. PENG is a member of the IEEE and the Aerospace Society of China.