Research Article

Robust Object Tracking Based on Simplified Codebook Masked Camshift Algorithm

Yuanyuan Zhang,¹ Xiaomei Zhao,² Fengjiao Li,¹ Jiande Sun,³
Shuming Jiang,¹ and Changying Chen¹

¹Information Research Institute, Shandong Academy of Sciences, Jinan 250014, China

²Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

³School of Information Science and Engineering, Shandong University, Jinan 250100, China

Correspondence should be addressed to Jiande Sun; jd.sun@sdu.edu.cn

Received 26 January 2015; Revised 5 June 2015; Accepted 10 June 2015

Academic Editor: Fernando Torres

Copyright © 2015 Yuanyuan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Moving targets detection and tracking is an important and basic issue in the field of intelligent video surveillance. The classical Codebook algorithm is simplified in this paper by introducing the average intensity into the Codebook model instead of the original minimal and maximal intensities. And a hierarchical matching method between the current pixel and codeword is also proposed according to the average intensity in the high and low intensity areas, respectively. Based on the simplified Codebook algorithm, this paper then proposes a robust object tracking algorithm called Simplified Codebook Masked Camshift algorithm (SCMC algorithm), which combines the simplified Codebook algorithm and Camshift algorithm together. It is designed to overcome the sensitiveness of traditional Camshift algorithm to background color interference. It uses simplified Codebook to detect moving objects, whose result is employed to mask color probability distribution image, based on which we then use Camshift to predict the centroid and size of these objects. Experiment results show that the proposed simplified Codebook algorithm simultaneously improves the detection accuracy and computational efficiency. And they also show that the SCMC algorithm can significantly reduce the possibility of false convergence and result in a higher correct tracking rate, as compared with the traditional Camshift algorithm.

1. Introduction

Moving object detection and tracking is the basis of object recognition and behavior understanding and has very broad application and research prospects. There are mainly three different categories of object detection algorithms, such as interframe difference methods [1], optical flow methods [2], and background subtraction methods. Background subtraction methods are the most popular ones in real world because of their high detection accuracy and medium computational complexity. Classical background subtraction algorithms include kernel density estimation [3], Gaussian Mixture Background Modeling [4], and Codebook background modelling [5].

Codebook algorithm was first proposed in 2004 by Kim et al. [5], and it has been one of the most advanced motion detection methods because of its high memory

utilization, high computation efficiency, and strong robustness. Many improvements have been made based on Codebook algorithm. For example, Wu and Peng [6] proposed a modified Codebook algorithm based on spatiotemporal context which improves the detection accuracy by adding the correlation of the spatiotemporal pixels. However, the computational complexity of the whole algorithm has been increased at the same time. Tu et al. [7] made simplifications to accelerate the computational speed by introducing box-based Codebook model in RGB space to represent the matching field of the codewords. However, these simplifications decreased the detection accuracy. Most of the improvements to Codebook can improve either the detection accuracy or computational efficiency, but not both of them.

Camshift algorithm is a classical object tracking algorithm. Camshift is evolved from Mean Shift algorithm. It performs tracking according to the color information of

an object and has very good real-time performance and high robustness. Mean Shift algorithm was first proposed in 1975 by Fukunaga and Hostetler [8]. Cheng [9] expended the algorithm and enlarged its application range. After that, Comaniciu and Meer [10] successfully applied it to image segmentation and object tracking. Bradski [11] established Camshift algorithm based on Mean Shift, which cannot only predict the centroid position of an object but also adaptively alter the size of an object frame. Currently, the improvement on Camshift algorithm exists in the following aspects: to improve the accuracy by improving the features of a histogram [12–14], to reduce computation time by increasing convergence velocity [15, 16], to increase robustness for objects rotation [17], and to solve the problem of background color interference. The improvement for Camshift algorithm in this paper concentrates on the issue of background color interference. In the literature, Camshift and Kalman combined algorithm [18–20] is easy to fail when object movement is nonlinear. The tracking accuracy of Camshift and interframe difference combined algorithm [21, 22] could be affected by a low performance interframe difference motion detection algorithm.

In order to simultaneously improve the detection accuracy or computational efficiency of Codebook algorithm, this paper first proposes a simplified Codebook algorithm. It is called hierarchical matching 5-tuple-based Codebook algorithm which is a modification of the original 6-tuple Codebook algorithm. The average intensity is introduced as a variable into the Codebook model instead of the minimal and maximal intensities. And different matching methods between the current pixel and codeword are adopted according to the average intensity in the high and low intensity areas, respectively. Based on the simplified Codebook algorithm, this paper then proposes a concise and robust object tracking algorithm called Simplified Codebook Masked Camshift algorithm (SCMC algorithm), which combines the simplified Codebook algorithm and Camshift algorithm together. A similar work is proposed by Wang [23], which uses the results of Codebook moving objects detection algorithm to mask the manual initialization searching box. However, our experimental results show that better tracking performance can be obtained when the color probability distribution images were masked by the simplified Codebook algorithm.

2. Simplified Codebook Algorithm

Compared with original Codebook algorithm [5], our simplified Codebook algorithm has two improvements: First, maximum and minimum brightness in codeword model are substituted by average brightness. So codeword model is simplified and computation speed is increased. Second, different processing methods of high and low brightness regions are applied to the matching between current pixel and codeword, so detection accuracy is improved and the probability of false detection is reduced in the low brightness region. The simplified Codebook algorithm in this paper is called hierarchical matching 5-tuple-based Codebook algorithm.

This section will present how to detect the moving object by the proposed simplified Codebook algorithm. First, we will show process of building a codebook for a specific pixel. Then all the other pixels can repeat the same process to complete the detection for a whole image.

2.1. Initialization. We build a codebook C_L ($C_L = \{c_i \mid 1 \leq i \leq L\}$) containing several codewords for every pixel, where L is the number of codewords. The i th codeword c_i includes two parts: RGB vector $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a 5-tuple $\text{aux}_i = \langle \bar{I}_i, f_i, \lambda_i, p_i, q_i \rangle$. The 5-tuple is composed of average brightness $\bar{I}_i$, codeword accessed frequency f_i , maximal nonrepeatable time interval λ_i , the initial codeword accessed time p_i , and the eventual codeword accessed time q_i . Except for the maximum and minimum brightness replaced by the average brightness $\bar{I}_i$, all the other elements remain the same as the original Codebook algorithm.

2.2. Training Background Model. Assume the first N frames of the video are used to train background model. For a particular pixel, the sequence of pixel values for training is $X = \{\vec{x}_t \mid 1 \leq t \leq N\}$, where each element $\vec{x}_t$ is RGB vector extracted from the t th image frame. Now, we take this pixel as an example to explain the codebook training process:

- (1) Build a codebook: we build a codebook C_L for the pixel and initialize it with an empty set (let $L = 0$).
- (2) Train the codebook: the following steps are executed circularly while t changes from 1 to N :

- (1) A new pixel value is read from the sequence $\vec{x}_t = (R_t, G_t, B_t)$. Brightness is calculated through $I = \sqrt{R_t^2 + G_t^2 + B_t^2}$.

- (2) Match between the pixel value and the codebook.

Find a codeword matching $\vec{x}_t$ in $C_L = \{c_i \mid 1 \leq i \leq L\}$ based on the following two conditions:

- (a) Color distortion

$$\text{colordist}(\vec{x}_t, \vec{v}_i) = \sqrt{|\vec{x}_t| - p^2} < \varepsilon_c, \quad (1)$$

where $\vec{v}_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ is the RGB vector of the i th codeword, ε_c is the threshold of color distortion matching, and p is the projection of $\vec{x}_t$ on $\vec{v}_i$, and it can be calculated by

$$p = \frac{\vec{x}_t \cdot \vec{v}_i}{|\vec{v}_i|} = \frac{\bar{R}_i R_t + \bar{G}_i G_t + \bar{B}_i B_t}{\sqrt{\bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2}}. \quad (2)$$

- (b) Brightness

$$\text{brightness}(I, \bar{I}_i) = \begin{cases} \text{true}, & \text{if } I_{\text{low}} \leq I \leq I_{\text{hi}} \\ \text{false}, & \text{otherwise,} \end{cases} \quad (3)$$

where $\bar{I}_i$ is the average brightness of the i th codeword and I_{hi} and I_{low} are the upper and lower bounds of brightness matching scope. ε_c , I_{hi} , and I_{low} can be calculated by the following formulas:

$$\begin{aligned} \varepsilon_c &= \begin{cases} \chi \cdot \bar{I}_i & \text{if } \bar{I}_i > \varepsilon \\ \varepsilon_{c0} = \chi \cdot \varepsilon & \text{otherwise,} \end{cases} \\ I_{hi} &= \begin{cases} \beta \cdot \bar{I}_i & \text{if } \bar{I}_i > \varepsilon \\ \bar{I}_i + \varepsilon_{b0} & \text{otherwise,} \end{cases} \\ I_{low} &= \begin{cases} \alpha \cdot \bar{I}_i & \text{if } \bar{I}_i > \varepsilon \\ \bar{I}_i - \varepsilon_{b0} & \text{otherwise,} \end{cases} \end{aligned} \quad (4)$$

where ε is the threshold to determine whether the current pixel belongs to the low brightness region or not. $\bar{I}_i$ is the average brightness of the i th codeword. χ is a variable used to calculate the threshold of color distortion matching, whose value is between 0 and 1. ε_{c0} is a constant threshold of color distortion matching in low brightness region. β is a variable used to calculate the ratio of the upper bound of brightness matching and average brightness in high brightness region. α is a variable used to calculate the ratio of the lower bound of brightness matching and average brightness in high brightness region. ε_{b0} is half of the brightness matching range when the brightness of the current pixel is lower than ε . ε_{c0} and ε_{b0} jointly guarantee the ranges of color distortion and brightness matching are not too small in low brightness area to avoid the occurrence of false detection.

- (3) If $L = 0$ or there is no matching codeword, then let $L = L + 1$ and create a new codeword $c_L(\vec{v}_L, \text{aux}_L)$, where the color vector is $\vec{v}_L = \vec{x}_t = (R_t, G_t, B_t)$ and the 5-tuple is $\text{aux}_L = \langle I, 1, t - 1, t, t \rangle$.
- (4) Otherwise, update the matched codeword $c_i(\vec{v}_i, \text{aux}_i)$ if $\vec{x}_t$ matches $c_i(\vec{v}_i, \text{aux}_i)$. Update the color vector by

$$\vec{v}_i = \left(\frac{f_i \cdot \bar{R}_i + R_t}{f_i + 1}, \frac{f_i \cdot \bar{G}_i + G_t}{f_i + 1}, \frac{f_i \cdot \bar{B}_i + B_t}{f_i + 1} \right), \quad (5)$$

and also update the 5-tuple by

$$\text{aux}_i = \left\langle \frac{f_i \cdot \bar{I}_i + I}{f_i + 1}, f_i + 1, \max \{ \lambda_i, t - q_i \}, p_i, t \right\rangle. \quad (6)$$

- (3) Regulate λ_i for every codeword c_i , and let

$$\lambda_i = \max \{ \lambda_i, (N - q_i + p_i - 1) \}. \quad (7)$$

- (4) Delete the nonbackground codeword. Assume the probability of background occurrence is bigger than 50%. Let M denote the background model which is the codebook after temporal filtering step. Specific operations can be expressed as the following formula:

$$M = \{ c_m \mid c_m \in C_L \wedge \lambda_m \leq T_M \}; \quad (8)$$

generally, $T_M = N/2$. M is the codeword set describing the background, where c_m is the m th codeword in C_L . λ_m is the maximal nonrepeatable time interval of the 5-tuple in c_m .

2.3. Foreground Detection. We match current pixel with a codeword by the same method as the training codebook. If matching exists, we update the codeword and take the current pixel as background. Otherwise, we take it as foreground.

3. Classical Camshift Algorithm

The original Camshift algorithm [11] takes the color histogram of an object as its characteristic model based on its color information. Video frame images are then changed into color probability distribution images. The centroid of the object is searched and the size of the object box is predicted on these images.

The implementation of the classical Camshift algorithm can be depicted as follows:

- (1) Initialize the position of the centroid and the size of the bounding box of the object.
- (2) Compute the color histogram of the bounding box.
- (3) Compute the color probability distribution image for the current frame.
- (4) Predict the position of the centroid with Mean Shift algorithm.
- (5) Predict the size of the bounding box.

The original Camshift algorithm can easily converge the bounding box to an object position, when there are significant differences between the object color and background color. Under such circumstance, pixel value of the object area is much higher than those of background on the color probability distribution image. However, when the object color is similar to the background color, the pixel value of the object area is no longer distinctive to those of background on the color probability distribution image. The algorithm will not guarantee that it will correctly converge the bounding box to an object position because it is color sensitive. This phenomenon is shown in Figures 1 and 2, respectively.

The video image sequences in Figures 1 and 2 are downloaded from the ITEA CANDELA project [24]. Figures 1(a)–1(d) and 2(a)–2(d) are color probability distribution images, and Figures 1(e)–1(h) and 2(e)–2(h) are tracking

FIGURE 1: Tracking results of Camshift when obvious difference exists in object color and background color.

FIGURE 2: Tracking results of Camshift when no obvious difference exists in object color and background color.

result images. From those figures, we can easily see that the original Camshift algorithm can track effectively when the object is significantly different to the background color. However, if the object color is similar to the background color, the algorithm may fail, just like Figure 2(h) shows.

4. Simplified Codebook Masked Camshift Algorithm

To overcome the background interference problem of Camshift algorithm, this paper proposes a SCMC algorithm, which is short for Simplified Codebook Masked Camshift algorithm. SCMC algorithm combines the simplified Codebook algorithm and the Camshift algorithm together. The detection result of simplified Codebook is utilized to mask the color probability distribution images. The main purpose is to filter out the background interference on tracking.

The implementation process of SCMC algorithm can be summarized as follows:

- (1) Detect moving objects with simplified Codebook algorithm.
- (2) Perform median filtering to the detection results to filter out noise and make the object connectable.
- (3) Compute the color probability distribution image for the current frame.
- (4) Mask the color probability distribution images with the processed foreground images in step (2).

Pixel values in background area are assigned as 0, and the pixel values in foreground area remain unchanged to guarantee Camshift algorithm converges only to the area of the moving object. The masking procedure of color probability distribution images is shown in Figure 3.

FIGURE 3: Illustration of the masking procedure of a color probability distribution image.

FIGURE 4: Flowchart of the proposed SCMC algorithm.

- (5) Search the centroid of the object with Camshift algorithm and predict the size of the bounding box.

The whole process of the proposed CMC algorithm is depicted in the flowchart of Figure 4.

5. Experimental Results and Analysis

The experiments were carried out on an ordinary PC with configuration of Intel(R) Core(TM) i3, 3.0 GB RAM, and 64-bit Windows 7 operating system. The programming environment is Microsoft Visual Studio 2010 and OpenCV 2.4.4.

5.1. Results of Simplified Codebook Algorithm. We use five different videos to illustrate the detection performance of the proposed simplified Codebook algorithm. Video #1 is captured by us. Video #2 is the famous video called Waving Tree [25]. Video #3 is chosen from the project of PETS2000 [26]. Videos #4 and #5 are selected from the ITEA CANDELA project [24].

The detection results of Video #1 and Video #2 using the proposed simplified Codebook algorithm are depicted in Figure 5. In order to show the superiority of the proposed algorithm, we also performed comparisons with several other motion detection algorithms such as Gaussian Mixture Model and original Codebook model. The real foreground extracted by hand is also depicted in Figure 5.

From Figure 5, we can see that detection results of the proposed simplified Codebook algorithm are better than the Gaussian Mixture Model and the original Codebook model. The original Codebook model may lead to false detection in the region with low brightness. The simplified Codebook algorithm can reduce the influence of low brightness area by using the method of hierarchical matching.

The computational difference between the simplified Codebook algorithm and the original Codebook model mainly lies in the process of calculating direct parameters from indirect parameters. The original Codebook model includes 2 multiplications, 1 division, and 1 subtraction, while the simplified Codebook algorithm includes 1 division and 3 multiplications, or 2 divisions and 1 addition. It is more likely to save a lot of operations by using the simplified Codebook

FIGURE 5: Object detection results of Video #1 and Video #2 (each row).

FIGURE 6: Object detection results of Video #3, Video #4, and Video #5 (each row).

algorithm especially when the average brightness is lower than ε . The detection speed of the simplified Codebook algorithm to Video #1 is 47 ms/frame, while detection speed of the original Codebook model is 62 ms/frame.

More experimental results of different detection algorithms of Video #3, Video #4, and Video #5 are given in Figure 6. From these figures, we can also see the superiority of the proposed simplified Codebook algorithm.

In order to illustrate the validation performance of the proposed simplified Codebook algorithm, we draw the ROC curves of three different algorithms in Figure 7. In this figure, the false positive rate is defined to be the ratio of amount number of background pixels which are falsely detected as foreground pixels to amount number of background pixels,

and true positive rate is defined to be the ratio of amount number of foreground pixels which are correctly detected as foreground pixels to amount number of foreground pixels. The validation performance is better when the area under ROC curve is larger.

5.2. Results of SCMC Algorithm. Here, we also use several videos to test the tracking performance of the proposed SCMC algorithm. Video #3, Video #4, Video #5, and Video #6 [27] are used in this part. First, we show the tracking results of a white car in Video #4 using the original Camshift algorithm in Figure 8. Figure 8(a) shows original color probability distribution images, and Figure 8(b) shows the corresponding tracking result images of the original Camshift algorithm.

FIGURE 7: ROC curves of three different algorithms.

FIGURE 8: Tracking results of a red car in the 165th, 173rd, 181st, and 189th frames in Video #4.

From Figure 8, we can easily see that serious tracking error occurs when background color and object color are slightly similar using the original Camshift algorithm.

Then we show the tracking results of the same object in Video #4 using the proposed SCMC algorithm in Figure 9. Figure 9(a) shows color probability distribution images masked by the moving object detection result using the simplified Codebook algorithm. Figure 9(b) shows the corresponding tracking result images of the proposed SCMC algorithm. In this figure, most of the background information was filtered out by the masked color probability distribution image. So the probability of false convergence is greatly reduced when using the SCMC algorithm, and higher tracking accuracy is obtained.

More comparisons about the tracking performance between the original Camshift algorithm and the proposed SCMC algorithm are shown in Figures 10, 11, and 12. We also give the tracking results of original Codebook with Camshift

algorithm and Compressive Tracking algorithm [28] in these figures. From these figures, we can see that the tracking performance can be easily influenced by the background color when using the original Camshift algorithm. However, most of the background information has been filtered out after using the SCMC algorithm. As a consequence, it largely reduces the possibility of false convergence. So better tracking performance is obtained. The tracking results are also better than those of original Codebook with Camshift algorithm and Compressive Tracking algorithm.

In Figure 13, we show the tracking results of a person using 4 different algorithms. As you can see in this figure, the tracking performance of the proposed SCMC algorithm is also better than those of original Camshift algorithm and original Codebook with Camshift algorithm. In Figure 13(c), one of the targets is missing by using the method of incremental learning for robust visual tracking. However, the person can be correctly tracked by the proposed SCMC algorithm.

(a) Color probability distribution images of masked simplified Codebook algorithm

(b) Tracking result images of the proposed SCMC algorithm

FIGURE 9: Tracking results of a red car in the 165th, 173rd, 181st, and 189th frames in Video #4 using the proposed SCMC algorithm.

(a) Original Camshift algorithm

(b) The original Codebook + Camshift algorithm

(c) Compressive Tracking algorithm

(d) The proposed SCMC algorithm

FIGURE 10: Tracking results of a red car in the 70th, 85th, 100th, and 115th frames in Video #3.

FIGURE 11: Tracking results of a black car in the 205th, 210th, 215th, and 220th frames in Video #3.

In order to illustrate the superiority of the proposed SCMC algorithm, we perform comparison with some state-of-the-art object tracking algorithm such as original Camshift algorithm, CT (Compressive Tracking) algorithm [28], and TLD algorithm [29] in Table 1. The number of correct tracking frames of some cars in Video #3, Video #4, and Video #5 is summarized in this table. We consider correct tracking if the center of the bounding box falls into the object area.

From Table 1, we discover that the proposed SCMC algorithm can correctly track most objects in Video #3, Video #4, and Video #5, and the tracking performance is significantly superior to the original Camshift algorithm. Moreover, the correct tracking rate of SCMC is 28.2% higher than that of TLD and is 19.4% higher than CT. SCMC algorithm is therefore superior to TLD and CT algorithms when the camera is stable.

Notice that the tracking performance of the proposed SCMC algorithm for the gray car in Video #3 is worse than that of all the other algorithms in Table 1. This special case is shown in Figure 14, too. The reason is that a black car approaches the object and then leaves it in the tracking

progress, which causes the false tracking of SCMC algorithm. By contrast, gray road information near the gray car is not filtered out for Camshift algorithm, which is a benefit to the convergence of the centroid of the object on the gray car. One possible disadvantage of the proposed SCMC algorithm is that the tracking performance may be affected if there is another moving object with similar color in the video, even if the interference of the background has already been restrained.

6. Conclusion

This paper first proposes a simplified Codebook algorithm called hierarchical matching 5-tuple-based Codebook algorithm. The average intensity is introduced as a variable into the Codebook model instead of the minimal and maximal intensities. And different matching methods between the current pixel and codeword are adopted according to the average intensity in the high and low intensity areas, respectively. Based on the simplified Codebook algorithm, the proposed SCMC algorithm masks color probability distribution images

TABLE 1: Comparison of tracking results for different tracking algorithms.

Objects in videos	First frame of object shows	Frame number of object shows	Number of correct tracking frames			
			Original Camshift [11]	TLD [29]	CT [28]	Proposed SCMC
Red car in Video #3	65	89	6	89	58	89
Black car in Video #3	193	151	13	151	151	151
White car in Video #3	403	61	0	61	53	61
Crimson car in Video #4	47	98	17	25	61	98
Black car 1 in Video #4	69	113	67	34	61	113
White car 1 in Video #4	97	113	53	55	10	53
Gray car in Video #4	139	30	27	30	30	16
White car 2 in Video #4	162	30	17	1	30	30
Black car 2 in Video #4	192	37	37	31	37	37
Red car in Video #4	285	79	7	0	79	79
Black car in Video #5	36	20	20	20	13	20
Gray car in Video #5	106	26	26	26	18	26
Red car in Video #5	133	25	22	22	22	25
Blue car in Video #5	420	29	29	28	29	29
Number of total frames (901)			341	573	652	827
Correct tracking rate			37.8%	63.6%	72.4%	91.8%

(a) Original Camshift algorithm

(b) The original Codebook + Camshift algorithm

(c) Compressive Tracking algorithm

(d) The proposed SCMC algorithm

FIGURE 12: Tracking results of a crimson car in the 50th, 60th, 70th, and 80th frames in Video #4.

FIGURE 13: Tracking results of a person in the 181st, 191st, 254th, and 349th frames in Video #6.

FIGURE 14: Tracking results of a gray car in Video #3 using SCMC algorithm; the 144th, 149th, 154th, and 159th frames are shown here.

with moving objects detection results, in which the pixel values in the background are configured as 0 to filter out the interference of background on Camshift tracking. The probability of false convergence is therefore greatly reduced. The algorithm has higher correct tracking rate than classical Camshift algorithm. Moreover, its correct tracking rate is superior to TLD and CT if the position of the camera is stable. However, it also has some disadvantages: (1) it may not perform tracking if background rapidly changes and (2) it may track a false object if two objects have quite similar colors. To solve the first disadvantage, future

improvement may be the construction of more robust background modeling to implement foreground detection for rapidly changing background. For the second disadvantage, we can include some texture information into the model to increase the distinction between different objects through the improvement of characters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project was supported by the Nature Science Foundation of Shandong Province (no. ZR2014FM012 and no. ZR2014YL010) and the Shandong Science and Technology Development Program (no. 2012GSF12004, no. 2013YD20001, and no. 2014GSF120018).

References

- [1] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1190–1203, 1999.
- [2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [3] A. M. Elgammal, D. Hanwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Computer Vision—ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II*, vol. 1843 of *Lecture Notes in Computer Science*, pp. 751–767, Springer, Berlin, Germany, 2000.
- [4] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pp. 175–181, Morgan Kaufmann, Providence, RI, USA, 1997.
- [5] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [6] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU—International Journal of Electronics and Communications*, vol. 64, no. 8, pp. 739–747, 2010.
- [7] Q. Tu, Y. Xu, and M. Zhou, "Box-based codebook model for real-time objects detection," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 7621–7625, IEEE, Chongqing, China, June 2008.
- [8] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.
- [9] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [10] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 750–755, IEEE, San Juan, Puerto Rico, June 1997.
- [11] B. Bradski G, "Real time face and object tracking as a component of a perceptual user interface," in *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision*, pp. 214–219, October 1998.
- [12] S. Birchfield Stanley and S. Rangarajan, "Spatio-grams versus histograms for region-based tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 1158–1163, Institute of Electrical and Electronics Engineers Computer Society, June 2005.
- [13] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 176–183, IEEE, June 2005.
- [14] Y. Xiao and X. Gang, "Camshift ship tracking algorithm based on multi-feature adaptive fusion," *Opto-Electronic Engineering*, vol. 38, no. 5, pp. 52–58, 2011 (Chinese).
- [15] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis, "Mean-shift analysis using quasi-Newton methods," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, vol. 2, pp. 447–450, IEEE Computer Society, Barcelona, Spain, September 2003.
- [16] M. Á. Carreira-Perpiñán, "Acceleration strategies for Gaussian mean-shift image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1160–1167, June 2006.
- [17] J.-Y. Zuo, Y. Liang, Q. Pan, C.-H. Zhao, and H.-C. Zhang, "Camshift tracker based on multiple color distribution models," *Acta Automatica Sinica*, vol. 34, no. 7, pp. 736–742, 2008 (Chinese).
- [18] K. Sun and S. Liu, "Combined algorithm with modified camshift and kalman filter for multi-object tracking," *Information and Control*, vol. 38, no. 1, pp. 9–14, 2009 (Chinese).
- [19] P. Juanchun, G. Lizhong, and S. Jianbo, "The hand tracking for humanoid robot using Camshift algorithm and Kalman filter," *Journal of Shanghai Jiaotong University*, vol. 40, no. 7, pp. 1161–1165, 2006 (Chinese).
- [20] L. Yuan, L. Ling, Z. Baisheng, and Y. Hongmei, "Video hand tracking algorithm based on hybrid Camshift and Kalman filter," *Application Research of Computers*, vol. 26, no. 3, pp. 1163–1165, 2009 (Chinese).
- [21] X. Liu, F. Chang, and H. Wang, "An object tracking method based on improved Camshift algorithm," *Microcomputer Information*, vol. 23, pp. 297–298, 2007 (Chinese).
- [22] W. Xiaojuan, Z. Haiting, W. Lei, and X. Liqun, "An improved Camshift hand tracking algorithm," *Journal of Shandong University (Engineering Science)*, vol. 34, no. 6, pp. 120–124, 2004 (Chinese).
- [23] C. Wang, *Intelligent Monitoring of Moving Object Detection and Tracking Algorithms*, Beijing University of Posts and Telecommunications, Beijing, China, 2012, (Chinese).
- [24] <http://www.multitel.be/image/research-development/research-projects/candela.php>.
- [25] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 255–261, September 1999.
- [26] <http://www.stats.ox.ac.uk/~wauthier/tracker/pets2000.html>.
- [27] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008.
- [28] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part III*, vol. 7574 of *Lecture Notes in Computer Science*, pp. 864–877, Springer, Berlin, Germany, 2012.
- [29] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking learning detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 1–14, 2010.