

---

# Design of moving target detection and tracking system based on cortex-A7 and openCV

Xingli Huang<sup>1</sup>, Tianfan Zhang<sup>2,\*</sup>, Zhenghong Deng<sup>1</sup>, Zhe Li<sup>2</sup>

1. Department of Automatic Control, Institution/University  
Northwestern Polytechnical University, Xi'an 710072, China

2. Hubei Engineering University, Xiaogan 432000, China  
alitasoft@hotmail.com

---

*ABSTRACT.* Applications based on real-time image processing are affected by communication pressure, random time lag, data packet loss and other problems. The main reason is image transmission and server message feedback. This can be resolved by terminal processing. But it requires the processing terminal to be able to complete the requirements of more complex image processing tasks under limited processing power. A Camshift based optimization local algorithm considering the characteristics of the platform is proposed. It can run the algorithm in-local without transferring images to the server. And Applications such as behavioral analysis are supported because the system still has the spare. Through algorithm analysis and experiment, we can prove the validity and real-time of this algorithm based on the hardware conditions. Firstly, the upper limit of hardware condition for engineering realization is studied. And provides predictable performance basis for the implementation of low-real-time requirements in operational performance on cortex M systems. Second, the maximum processing speed of 41FPS means more additional cost space or performance redundancy. Finally, real-time recognition of personnel behavior is implemented in the current hardware environment. Provides an important reference for other behavioral analysis researchers. In this paper, the work of hardware carrier and algorithm is preliminarily completed, which can provide cutting for further optimization research and engineering realization.

*RÉSUMÉ.*pte tenu de la contradiction entre la complexité des tâches de traitement d'images et la capacité de traitement limitée des terminaux, cet article propose un algorithme d'optimisation local basé sur le décalage moyen adaptatif continu (CamShift), une méthode de segmentation d'images répandue. L'algorithme proposé peut être exécuté localement sans transfert d'images sur le serveur et supporte des applications telles que l'analyse comportementale grâce à sa capacité disponible. La validité et les performances en temps réel de cet algorithme ont été vérifiées dans diverses conditions matérielles au moyen d'analyses expérimentales. Tout d'abord, la limite supérieure de la condition matérielle pour l'application d'ingénierie a été déterminée, posant les bases de la mise en œuvre de l'algorithme dans les systèmes Cortex-M avec une faible exigence de performances en temps réel. Deuxièmement, la vitesse de traitement maximale a été identifiée à 41 images par

*seconde, ce qui indique que l'algorithme consomme plus d'espace ou présente une redondance de performances. Enfin, la reconnaissance en temps réel des comportements humains a été mise en œuvre dans l'environnement matériel actuel. Les résultats de la recherche fournissent une référence significative pour l'analyse comportementale.*

*KEYWORDS: behavior Analysis, camshift, cortex-A7, embedded system, target tracking, openCV.*

*MOTS-CLÉS: analyse du comportement, camshift, cortex-A7, système embarqué, suivi de cible, openCV.*

---

DOI:10.3166/TS.35.61-73 © 2018 Lavoisier

## 1. Introduction

Personnel behavior analysis is an important part in the field of public security image processing. Most of the current research is focused on the server end because of the complexity of such image processing and the sheer volume of computation (Barth *et al.*, 2015). And high performance hardware systems are required. The processing cost of subsequent behavioral analysis can be effectively reduced by (personnel) motion target recognition and trajectory tracking (Deng *et al.*, 2014).

The research of Personnel behavior analysis algorithm on server side have achieved some important results. Xu *et al.* (2015) examined the accuracy level of the Microsoft™ Kinect sensor for assessment of various gait parameters during treadmill walking under different walking speeds. Barth *et al.* (2015) developed a automatically segment single strides methodologies from continuous movement sequences and boost F-measure by 15%. The related research of hardware restriction condition and low network dependence. Garcíagarcía *et al.* (2018) propose the implementation of a 3D object recognition system which will be optimized to operate under demanding time constraints and recognize objects in a scene in less than 7 seconds. Liu *et al.* (2011) using a low cost FPGA-based System-on-Chip to detect and recognize more than 10 obstacles of different types in a time limit of 25 mSec. Liu *et al.* (2018) design a food recognition system employing edge computing-based service computing paradigm to overcome some inherent problems of traditional mobile cloud computing paradigm, such as unacceptable system latency and low battery life of mobile devices. Reducing response time that is equivalent to the minimum of the existing approaches and lowering energy consumption which is close to the minimum of the state-of-the-art. These kinds of research have made some progress in their respective areas. However, the following issues still exist: The platforms used such as the NVIDIA™ Tegra system still have the same processing power and cost as the average PC. Mainly to the static image, the change rate is low non-real-time image processing research mainly. Most of the complex task handling involved is still focused on the server side, or requires the prior use of additional design for training. The deployed platform requires the ability to deploy a running environment such as DCNN requirements. More or less network dependencies exist and do not take into account real-time field decision-making needs. Therefore, further research is necessary.

In this paper, a real-time motion target recognition and tracking algorithm is

proposed, which can be processed locally in the image acquisition terminal without the need to transfer the image to the server or wait for feedback instructions. This improves the responsiveness of the terminal and provides a research basis for a complete localized analysis of personnel behavior. Specifically, there are the following points of work: Firstly, with the general research does not consider the hardware platform or the use of PC as a research carrier, this paper takes into account the actual needs of engineering implementation. The computing performance of Cortex A7 embedded system is between Cortex M and PC. The upper limit of hardware condition for engineering realization is studied. And provides predictable performance basis for the implementation of low-real-time requirements in operational performance on cortex M systems. Secondly, the processing speed of this algorithm can be stabilized in the 25 FPS, and the proposed algorithm is proved by experiments under the given conditions. The processing speed of 41FPS can be reached under simple scene and single moving target. This means that the requirements for hardware system and power consumption can be reduced at a lower frame rate. Thus further reducing the implementation cost of the system. This provides a specific performance metrics and reference for other researchers. Finally, a personnel behavior analysis algorithm based on target tracking algorithm is presented in the research prospect. In the current hardware environment, the real-time recognition of several kinds of personnel behavior is realized, which provides an important reference for other behavioral analysis researchers.

It expounds the main thrust of this paper and the present situation of field research in chapter I. In the second chapter, the overall design of the system is given. The hardware design, algorithm design and implementation of the system are described in detail in the third to fourth chapter. In the fifth chapter, the algorithm and hardware platform are tested. The framework of Behavior recognition algorithm and the effect of three kinds of recognition are given. Finally, summarizes the shortcomings of this study, and looks forward to future research.

## **2. The overall design of the system**

The system is composed of video acquisition device, embedded hardware development platform, the embedded operating system and the upper applications; in which the video acquisition device and embedded hardware platform forms the hardware part of the system; The embedded operating system and the upper the application forms software part of the system. The system is based on API interface provided by the OpenCV to process the image data with the capacity of the embedded processor. The structure diagram of tracking system is shown in Figure 1.

### **2.1. Video collection equipment**

According to the performance of the system and the functional requirements, ZC301P USB camera is selected as the video capture device. The device is hot-pluggable, the maximum resolution is 640×480, the maximum frames per second is

30 FPS/s. In the process of image, the processor calls the camera interface function provided by OpenCV to supervise and control the equipment.

## 2.2. The embedded hardware platform

The Cortex-A7 processor is a very energy-efficient applications processor designed to provide rich performance in entry-level to mid-range smart phones, high-end wearable, and other low-power embedded and consumer applications.

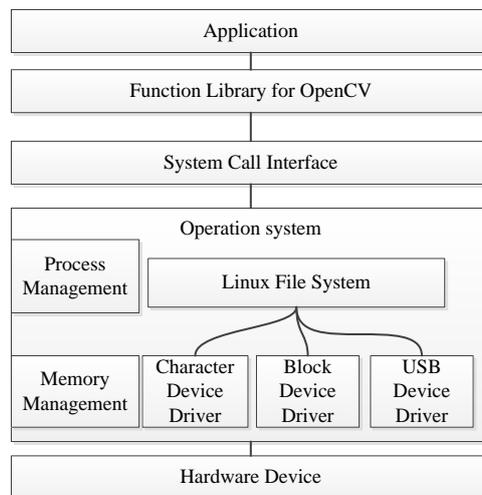


Figure 1. The structure diagram of embedded platform target tracking system

Choose A20 (Allwinner™ Dual-Core Cortex-A7 ARM CPU 1G) to be the embedded platform which is with the Cortex-A7 high-performance architecture, and equipped with Mali400mp2 GPU, 16GB NAND FLASH and 1GB DDR3 SDRAM, that can meet the requirements of real-time image processing (Xing *et al.*, 2017).

The Mali family of products combines to provide the complete graphics stack for all embedded graphics needs, enabling device manufacturers and content developers to deliver the highest quality, cutting-edge graphics solutions across the broadest range of consumer devices. The Cortex-A7 is compatible with all Mali mid-range and high-end graphics processors, the Mali-DP500 display processor and the Mali-V500 video processor.

## 2.3. The software platform for the system

Due to the lack of independent development ability for the embedded systems, we can't directly set up the development environment on the structures, so cross compile environment on the host is needed.

OpenCV as an open source application platform provides several cross-platform API for C function (Kaehler, 2008). Its platform independence makes that the program with OpenCV can be transplanted between multiple platforms, no need to modify the code. OpenCV-2.4.9 is used for the system. First cross-compiling OpenCV vision library under the Fedora, finally put the cross compiled library function on to the target board, so as to make the target board support programs written in OpenCV (Guennouni *et al.*, 2015). The flow diagram of Software platform building is as following:

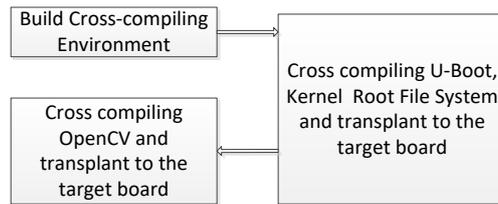


Figure 2. The flow diagram of software platform building

### 3. Localized image processing algorithms

#### 3.1. The image preprocessing

As a result, that the collected video image will be affected by external disturbance such as the weather and light, in order to reduce the influence of interference for the target detection and tracking, the video data preprocessing is needed. At the same time, in order to reduce the image file storage and the memory, the image color space conversion could reduce the computational complexity to make it convenient for subsequent processing.

Discrete pulse, the salt and pepper noise and zero mean Gaussian noise have a big influence on the image which are obtained by camera, which has large effect on target detection and tracking. So this article uses the median filter to remove impulse noise and salt and pepper noise. The definition of the median filter is as following:

$$g(x, y) = \text{med}\{f(x-k, y-i)\}, (k, i \in W) \quad (1)$$

where,  $f(x, y)$ ,  $g(x, y)$  represent the original image and processed image respectively.  $W$  is as two-dimensional template, it is usual as  $3 \times 3$ ,  $5 \times 5$  regions.

#### 3.2. Target detection module

##### (1) Analyzing for target detection algorithm

The moving target detection algorithm based on static background mainly includes the frame difference method, background subtraction, optical flow method

and statistics method. For the optical flow method and statistics method, due to the high complexity of algorithm, it needs high performance hardware, and the complexity of time is high. The frame difference method is simple and less sensitive to light and scene change, but it does not extract the complete area of the object, it can only extract the boundary; the background subtraction method is accurate and easy to implement, but it is difficult to directly obtain the static background; due to the dynamic changes of the background image, so the background should be updated selectively.

(2) The target detection algorithm

Because the operation performance and capabilities are relatively limited for the embedded hardware system, so this article chooses Surendra background update algorithm as detection algorithm, which is combined by the frame difference and background-differencing-method (Zeng, 2015).

The algorithm can be divided into the following steps (Yang, 2014):

1. Setting the first frame  $I_0$  as the background  $B_0$ ;
2. Setting the binarization threshold  $T$  (threshold is 80 in this article), the current cycle  $m=1$ , the max number of iteration  $m=MAX-STEPS$ .
3. Calculating the difference of the current frame  $I_i$  and the prev-Frame  $I_{i-1}$  according to the principle of frame difference the binary image  $D_i$  is:

$$D_i(x, y) = \begin{cases} 1, & |I_i - I_{i-1}| \geq T \\ 0, & |I_i - I_{i-1}| < T \end{cases} \quad (2)$$

where,  $I_i$  represents the current frame,  $I_{i-1}$  is the prev-Frame,  $|I_i - I_{i-1}|$  is the frame difference between the current frame and the prev-Frame,  $D(x, y)$  is the grey value in the place of  $(x, y)$  of the binary image.

4. Updating background image  $B_i$  according to  $D_i(x, y)$ :

$$B_i(x, y) = \begin{cases} B_{i-1}(x, y) & D_i(x, y) = 1 \\ \alpha I_i + (1-\alpha)B_{i-1}(x, y) & D_i(x, y) = 0 \end{cases} \quad (3)$$

where,  $B_i(x, y)$  is the grey value in the place of  $(x, y)$  of the background image,  $\alpha$  means iterative update coefficient.

5. Setting the number of iteration  $m=m+1$ , return to step 3; when  $m=MAX-STEPS$ , the algorithm ends, then making background image  $B_i$  as the current background image.

6. Making the difference between the current frame  $i$  and the current background frame  $B_i$ :

$$d_i(x, y) = |I_i(x, y) - B_i(x, y)| \quad (4)$$

$$DB_i(x, y) = \begin{cases} 255 & d_i(x, y) = T \\ 0 & d_i(x, y) = \bar{T} \end{cases} \quad (5)$$

where,  $d_i(x, y)$  is the grey value in the place of  $(x, y)$  after making difference;  $DB_i(x, y)$  is the binary result of  $d_i(x, y)$ .

(3) The image post-processing

The morphology filtering processing operation is needed to eliminate the phenomenon of the hole and isolated points in the binary image. The erode function could eliminate the effect of boundary point of the object to make the target small; the dilate function could increase the target and fill the hole of the target.

**3.3. The design of target tracking algorithm**

This paper adopts Kalman filter combined with CamShift tracking algorithm; the Kalman filter has the function of prediction and can overcome some target obscured, mutual interference between the target to predict the location of the object; CamShift can effectively solve the problem of target deformation and occlusion, and requirements of system resources is not high, time complexity is low. Algorithm process is shown as Figure 3:

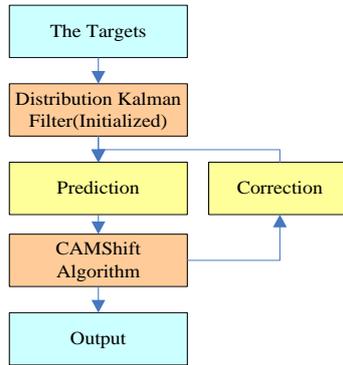


Figure 3. Tracking algorithm process

Kalman filtering algorithm includes the following two models:

The signal model:

$$X_k = A_k X_{k-1} + B_k W_k \quad (6)$$

$$Z_k = H_k X_k + V_k \quad (7)$$

where,  $X_k$  and  $Z_k$  means the state vector and the observation vector respectively;  $A_k$ ,  $B_k$  and  $H_k$  are state-transition matrix, input matrix and the observation matrix.  $W_k$  and  $V_k$  mean dynamic noise and observation noise.

The steps of CamShift algorithm are as following:

(a) Selecting appropriate target search window;

(b) Calculating the zero order and first order moments of the search window; the zero-order moment:

$$M_{00} = \sum_x \sum_y I(x, y) \quad (8)$$

The first order moment:

$$M_{10} = \sum_x \sum_y xI(x, y) \quad M_{01} = \sum_x \sum_y yI(x, y) \quad (9)$$

(c) Calculating the center of the search window:

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \quad (10)$$

(d) Setting the size of the search window:

$$s = 2\sqrt{M_{00}/256} = 0.125\sqrt{M_{00}} \quad (11)$$

(e) Iterative computation.

## 4. The result and analysis of the experiment

### 4.1. Algorithm Testing and analysis

In the Linux environment using OpenCV-2.4.9 to make implementation and simulation for detection algorithm and tracking algorithm, test sets were collected in the crossroads with 320×240, the compile tools are the g++ and arm-linux-g++; Selecting the image without any moving target as the first frame; goal in test sets are mainly the pedestrians, vehicles, automotive and so on. The following are the effect of target detection and tracking. Original samples used in the experiment as shown in figure 4.



Figure 4. Original samples used in the experiment (the first one is the RGB image; the second is grayscale background image)

Sample time is early winter afternoon. The background frame is given as the grayscale graph, because the algorithm mainly takes the grayscale graph and Reduce the amount of computation as the main.

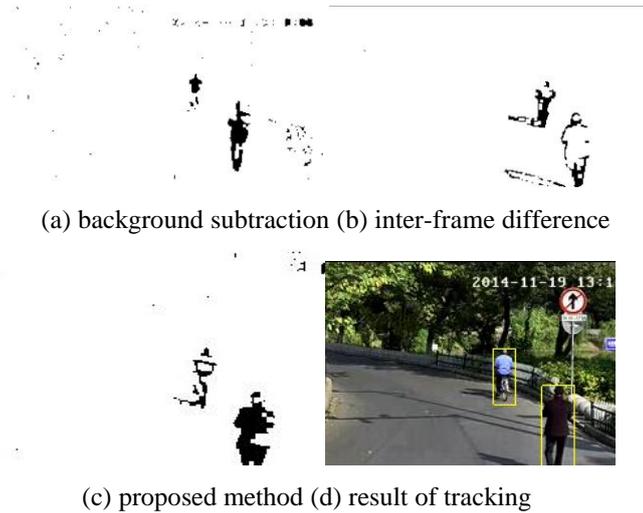


Figure 5. Three methods for experimental comparison and tracking of results

The simulation result shows that when there appears moving object in the picture, the algorithm adopted by this paper can well separate the target from the background of sports and following compared with the frame differential method and background subtraction method. There is more noise interference in the background as shown in figure 5 (a); the inter-frame difference method reduces background noise interference, but the shadow of the target is still not to be ignored as shown in figure 5 (b). This method preserves the body of the target as much as possible and the background noise is maintained at a low level. And the final recognition result is clear and acceptable as shown in figure 5 (c) and (d).

Using the cross-compile tools arm-Linux-g++ to cross-compiling the

applications and transplanting to the processor, changing the mode of the application with the command “chmod + x tracking”, then experimenting.

The analysis comparison of the processing speed for the system (hardware 25 frames/s), as shown in following:

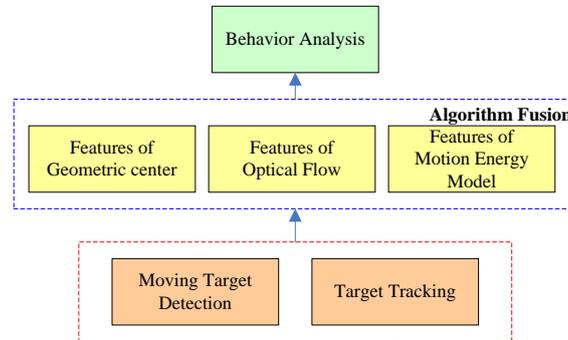
*Table 1. The statistics of target tracking time*

Tracking Number	1	2	3
Consumption Per Frame (ms)	23.9264	25.7645	26.2829

Known from the table above, the algorithm can meet the demand of the system, the experiment proved that the system can meet the requirements of the real-time video processing.

#### **4.2. Additional tests and future research**

In order to further verify the effectiveness and practicability of the algorithm, tested the algorithm combined with our future research "analysis of abnormal behaviour". At first, use the algorithm for catch and track the moving targets. Then use the combinatorial analysis method to analyze the personnel behavior by these targets. The test program framework, as shown Figure 6:



*Figure 6. Algorithm diagram of abnormal behavior analysis based on the Algorithm and hardware platform*

Because of the algorithm of behavior analysis is very difficult (Deng *et al.*, 2012). We chose the interior as a test site that environment is uncomplicated (the light environment is basically constant) (Sobhani *et al.*, 2017). Two groups of tests were conducted, include walk, fall, fight with single and multiplayer situations. The test results as shown Figure 7 and Figure 8:

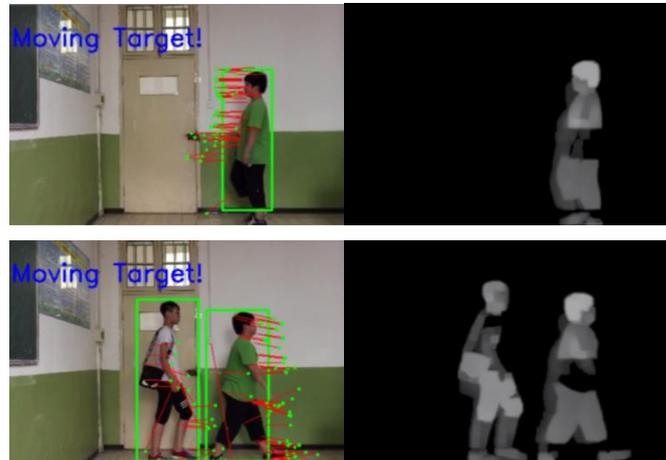


Figure 7. Moving target recognition and tracking

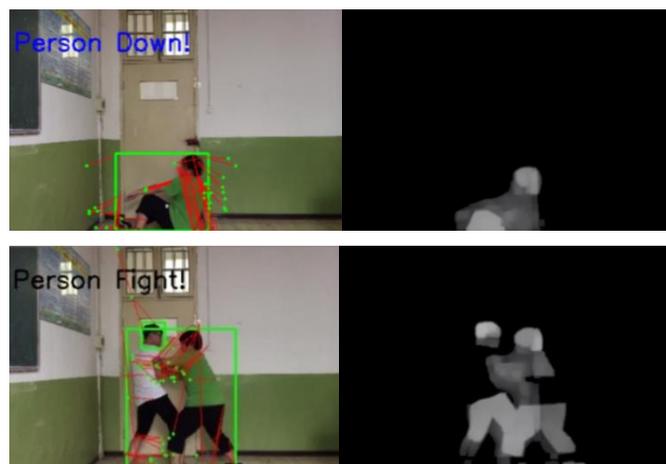


Figure 8. Targets behavior analysis: Fall and fight (Robbery)

As shown in figure 7, the effect of the tracking algorithm is satisfactory in the test environment. As shown in Figure 8, the identification of two typical abnormal behaviors is relatively accurate. There are still some problems did not get to resolve effectively (Harik *et al.*, 2017). Our team has made a preliminary inquiry into some of these issues, researcher and time delay problem in multi-sensor fusion (Li *et al.*, 2016), Li *et al.* (2017) discusses the problem of robust controller design for two-dimensional system such as image process.

## 5. Conclusions

The system uses the Cortex-A7 as the hardware platform, through transplanting OpenCV library computer vision and call the API interface provided by the OpenCV to process the video data acquired by USB camera. The system is the embedded intelligent tracking system based on the front-end processing, for the later it can expand other function. Experiment results show that the system can meet the requirements of real-time detection and tracking, it implements the design demands.

### *Acknowledgments*

*This work was supported by the Public Welfare Scientific Research Program is funded by the Wenzhou City (LYG20160020). Hubei Natural Science Fund Project (2018CFB14). Scientific Research Program Funded by Hubei Provincial Department of Education 2017 (B2017505).*

## References

- Barth J., Oberndorfer C., Pasluosta C., Schulein S., Gassner H., Reinfelder S., Eskofier B. M. (2015). Stride segmentation during free walk movements using multi-dimensional subsequence dynamic time warping on inertial sensor data. *Sensors*, Vol. 15, No. 3, pp. 6419-6440. <https://doi.org/10.3390/s150306419>
- Deng Z. H., Jiao L. T., Liu L. Y. (2014). Design of gait recognition system based on FPGA and DSP. *Applied Mechanics & Materials*, No. 687-691, pp. 3861-3868. <https://doi.org/10.4028/www.scientific.net/AMM.687-691.3861>
- Deng Z. H., Li T. T., Zhang T. T. (2012). An adaptive tracking algorithm based on mean shift. *Advanced Materials Research*, No. 538, pp. 2607-2613. <https://doi.org/10.4028/www.scientific.net/AMR.538-541.2607>
- Garcia Garcia A., Ortsescolano S., Garciarodriguez J., Cazorla M. (2018). Interactive 3D object recognition pipeline on mobile GPGPU computing platforms using low-cost RGB-D sensors. *Journal of Real-time Image Processing*, Vol. 14, No. 3, pp. 585-604. <https://doi.org/10.1007/s11554-016-0607-x>
- Guenouni S., Mansouri A., Ahaitouf A. (2015). Multiple object detection using OpenCV on an embedded platform. *Information Science and Technology (CIST)*, pp. 374-377. <https://doi.org/10.1109/CIST.2014.7016649>
- Harik E. H. C., Guérin F., Guinand F., Brethé J. F., Pelvillain H. (2017). Fuzzy logic controller for predictive vision-based target tracking with an unmanned aerial vehicle. *Advanced Robotics*, pp. 1-14. <https://doi.org/10.1080/01691864.2016.1271500>
- Kaehler A. (2008): *Learning OpenCV*. O'Reilly, Beijing, China, pp. 215-223.
- Li Z., Zhang H. X., Mu D. J., Guo L. T. (2016): IEEE access analysis and synthesis of large-scale systems. *IEEE Access*, Vol. 4, pp. 7509-7518. <https://doi.org/10.1109/ACCESS.2018.2856307>
- Li Z., Zhang T. F., Ma C., Li H. X., Li X. Z. (2017). Robust passivity control for 2-D uncertain Marconian jump linear discrete-time systems. *IEEE Access*, Vol. 5. No. 1, pp. 12176-12184. <https://doi.org/10.1109/ACCESS.2017.2719918>

- Liu C., Cao Y., Luo Y., Chen G., Vokkarane V. M., Yunsheng M., Hou P. (2018). A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. *IEEE Transactions on Services Computing*, Vol. 11, No. 2, pp. 249-261. <https://doi.org/10.1109/TSC.2017.2662008>
- Liu H., Niar S., Elhillali Y., Rivenq A. (2011). Embedded architecture with hardware accelerator for target recognition in driver assistance system. *ACM Sigarch Computer Architecture News*, Vol. 39, No. 4, pp. 56-59. <https://doi.org/10.1145/2082156.2082170>
- Sobhani B., Paolini E., Giorgetti A., Mazzotti M., Chiani M. (2017). Target tracking for UWB multistatic radar sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 8, No. 1, pp. 125-136. <https://doi.org/10.1109/JSTSP.2013.2286771>
- Xing A., Jin X., Li T. (2014). Speeding up deep neural networks for speech recognition on ARM Cortex-A series processors. *International Conference on Natural Computation (ICNC)*, pp. 123-127. <https://doi.org/10.1109/ICNC.2014.6975821>
- Xu X., Mccorry R. W., Chou L., Lin J., Chang C. (2015). Accuracy of the Microsoft Kinect™ for measuring gait parameters during treadmill walking. *Gait & Posture*, Vol. 42, No. 2, pp. 145-151. <https://doi.org/10.1016/j.gaitpost.2015.05.002>
- Yang M. (2014). A moving objects detection algorithm in video sequence. *International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 410-413. <https://doi.org/10.1109/ICALIP.2014.7009826>
- Zeng H. F. (2015). Design of embedded intelligent intrusion detection system based on ARM. *Microcomputer & Its Applications*, Vol. 12, No. 12, pp. 85-87. <https://doi.org/10.3969/j.issn.1674-7720.2015.12.028>

