

## Real-time system design for object tracking

Chong Liu, Zhenhao Li, Yang Sang, Qinglong Ba

School of Mechanical Engineering & Automation , Northeastern University, Shenyang 110819

congliu@me.neu.edu.cn

**Keywords:** object tracking, mean-shift, lower control, upper control

**Abstract.** This paper describes the realization of a system capable of tracking objects. The whole system design includes lower control system design and upper control system design. The rotation of the camera was controlled by an MCU in the lower control system and the tracking algorithm mean-shift was performed in a computer in the upper control system. Experiment results showed that the system could run well as a real time tracking system, and could track the object selected well.

### Introduction

Object tracking is a task required by different computer vision applications[1]. The objective of object tracking is to locate a previously specified target object in subsequent video frames[2]. Tracking an object in complex image sequences is a challenging task[3]. And to achieve the target, a number of popular object tracking algorithms such as mean-shift[1], optical flow[4], and feature matching[5] have been established. All these algorithms perform well in applications such as teleconferencing, surveillance and human-computer interface.

But in most of the applications, the tracking system needs to work in real-time. So, a kind of robustness object tracking method and a system which can manipulate the whole image sampling and digital image processing method well are very important and greatly needed.

This paper mainly studied the real-time object tracking control system design, and presents an object tracking system using mean-shift as the tracking algorithm. The whole system includes the algorithm design and control system design. The description of algorithm is in Section 2 and the whole control system design is described in Section 3. The experimental results and conclusion are finally described in Section 4 and 5.

### Tracking algorithm

The mean-shift algorithm is a fast and accurate object tracking algorithm. It is a kind of non parametric algorithm, which was originally used in the pattern recognition as an cluster analysis, and is widely used in the field of object tracking[6-10]. Mean-shift algorithm is generally refers to an iterative steps, which calculates the mean shift, moving the point to the next mean shift, then taking this as the new starting point, and continue to move until it meets certain conditions. A brief description of the mean-shift algorithm for object tracking is given below.

#### Target and target candidate model

Let  $X_i$  denote pixel locations of the target model,  $n$  be the number of pixels of the target which centered at  $X_0$ . The kernel density  $q$  is estimated by the target histograms as follow:

$$q(u) = C \sum_{i=1}^n k \left( \left\| \frac{X_i - X_0}{H} \right\|^2 \right) \delta(b(X_i) - u), i \in [1, \dots, n] \quad (1)$$

where  $\delta$  is the Kronecker delta function and  $C$  is a normalization constant, which imposes the condition that  $q_1 + q_2 + q_3 + \dots + q_m = 1$ .  $H$  is the bandwidth of the kernel density estimation.  $m$  is the number of features, which is equal to the number of gray of the picture.  $b(x_i)$  maps the value of the pixel at location  $x_i$  to the corresponding in the feature space.

Let  $x_i$  be pixel locations in the current frame and the target candidate is centered at location  $y$ . So the probability of the feature in the target candidate is:

$$p_u(Y) = C_h \sum_{i=1}^{n_h} k \left( \left\| \frac{X_i - Y}{H_h} \right\|^2 \right) d(b(X_i) - u) \quad i \in [1, \dots, n_h] \quad (2)$$

where  $H_h$  is the number of pixels of the target candidate.  $C_h$  is a normalized constant.

### Similarity function

In order to measure the distance of the two probabilities, *Bhattacharyya coefficient* is used and the similarity function is shown as (3).

$$r(p(Y), q) = \sum_{u=1}^m \sqrt{p_u(Y) q_u} \quad (3)$$

The matching process is to search for the maximum of the similarity function, and the gradient descent method is used for solving the problem. The search for the new target location in the current frame uses gradient ascent with a step size equivalent to the mean-shift method. And the target moved from the current location to the new location as in (4).

$$Y_1 = \frac{\sum_{i=1}^{n_h} X_i w_i g \left( \left\| \frac{Y_0 - X_i}{H_h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left( \left\| \frac{Y_0 - X_i}{H_h} \right\|^2 \right)} \quad (4)$$

Where  $w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(Y_0)}} d(b(X_i) - u)$ . And the tracking process is to search for the new target location by iteratively.

### Tracking control system

The whole tracking control system is shown as in Fig. 1. It is composed of 2 parts: lower control system and upper control system. The lower control system which is composed of MCU and servo motor is for the controlling of the servo motor's rotation with the camera assembled on it. And the upper control system which is composed of computer and camera is for the camera data processing and the algorithm realizing.

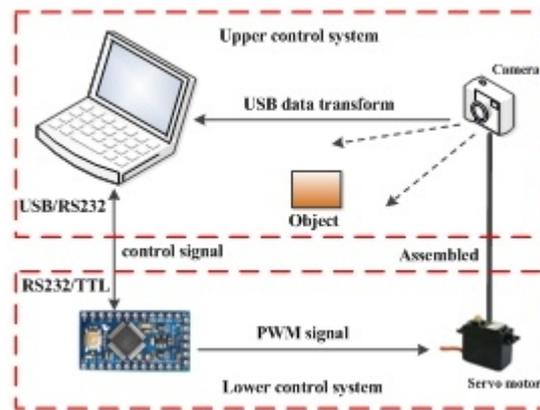


Fig 1. The whole object tracking system

### Lower control system

A kind of low cost camera equipped on a toy car was used to evaluate the algorithm of object tracking which is shown as in Fig. 2. The rotation of the camera on the car was controlled by a servo motor. And the movement of the servo motor was controlled by the MCU(arduino pro mini) through motor driver. The whole system was powered by a 7.4V battery.

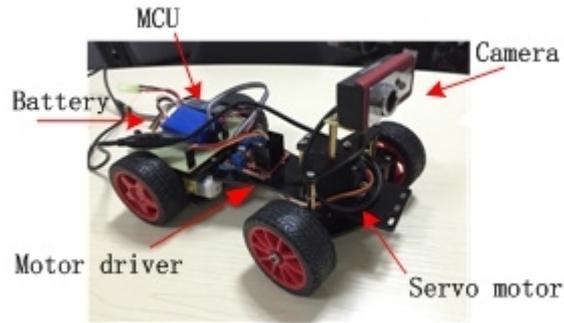


Fig 2. Lower control system

The lower control system receives data from the upper control system, and the data are the angles the upper control system wants the camera to rotate. Since the camera may rotate in different directions such as rotate left or right, the direction information is also included in the data received from the upper control system. And here the 8 bits data received from the upper control system through serial port was defined as Fig. 3. The first 2 bits are used to describe the direction information, and the two bits data could describe 4 directions. The last 6 bits are used to describe the angle information, so it means the servo motor could only rotate within 0 degrees to 63degrees in each direction.

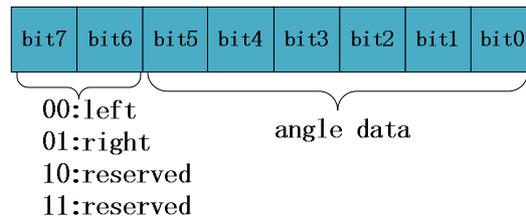


Fig 3. Format of the data received from upper control system

The MCU keeps monitoring the serial port since the system start. If it receives any data, the first 2 bits of the data are used first to get the direction information and the last 6 bits of data which include the angle information are then used to generate the PWM(Pulse Width Modulation) signal to make the servo motor rotate to a certain angle.

### Upper control system

A computer together with the camera was used as the upper control system. The data sampled from the camera were all processed in the computer according to the algorithm in Section 2. The algorithm was programmed in MATLAB, and the Image Acquisition Toolbox was used.

Since the camera in Fig. 2 could not move in the vertical direction, this paper only concerns about tracking the object which could only move in oriental direction. We divided each frame the camera captured into 3 parts: left, middle and right parts as in Fig. 4. And the objective of the tracking system is to make the camera rotate to keep the object in the middle part of the frame. The boundaries were selected according to the real experiment results, and these could not be too close or too far. Because if the boundaries are too close, the object will move out of the middle part easily and will swing between the left and right parts while the camera rotates. And if the two boundaries are too far, the camera will lose the target easily.

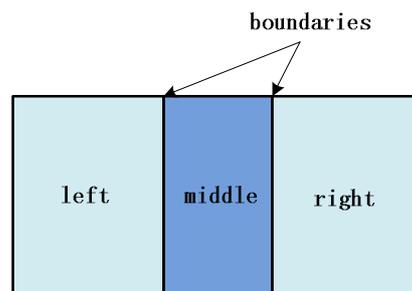


Fig 4. The 3 parts of a frame

When the boundaries are decided, the system will run as the following steps:

Step1. Capture a frame  $f_0$

Step2. Select the object to track in  $f_0$

Step3. Get the center point  $X_0$  of the object in  $f_0$  and the coefficients  $q$ ,  $H_h$  and  $C_h$  for the target model

Step4. Capture the next frame  $f_n$

Step5. Use Mean shift algorithm to get the new location of the object in current frame  $f_n$

Step6. Get the center  $X_n$  of the tracked object in the frame  $f_n$

Step7. Calculate the distance =  $X_n - X_0$  and find out  $X_n$  belongs to which part of  $f_n$ . If  $X_n$  belongs to the left part distance =  $|distance|$  and if  $X_n$  belongs to the right part then distance = distance+64

Step 8. Set the new location center to  $X_0$

Step 9. Send the value *distance* to serial port

Step 10. Go to step4 and loops.

Fig. 5 shows the object tracking interface of the upper control system. All the control and setup buttons are on the left side. The first button in the “Object Tracking Control Panel” could start/stop the whole system and the second could start the tracking. The interrupt frequency could be set in the “Control Panel”, and camera could also be controlled manually by pressing the “KeyPressControl” button. The serial port setting is in “COM port data” panel, and the state of the system or the data lower control system returns are shown on the bottom of the panel. The figure on the right side shows the tracking interface. The object was first selected by hand and then tracked automatically. And the object surrounded by green lines shown in the figure is the object the system found and tracked.

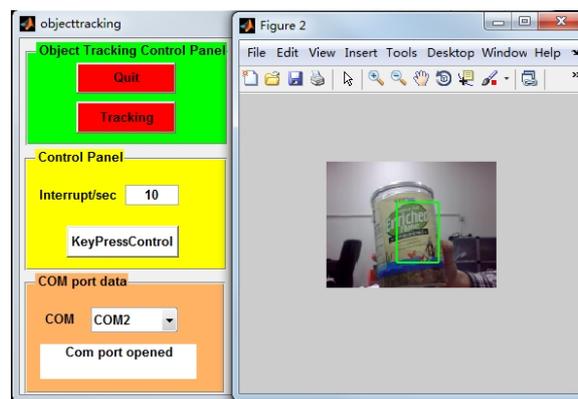


Fig 5. The tracking interface

## Experiment and results

The resolution of the camera was set to 160\*120. And the boundaries of middle part were set to 70 and 90, which means the band width of middle part is 20 pixels and the oriental center of the current frame is 80. The interrupt frequency was set to 10. The control signal of upper control system was sent to the lower control system through serial port ‘COM2’. The maximum iteration of mean-shift algorithm was set to 20, which means if the algorithm could not find the new target location within 20 iterations the algorithm will stop and search all over again.

Fig. 6 shows 14 successive frames sampled while the system was tracking an object. The object surrounded in green lines was first in the middle part of the frame, and then moved to the right side of the camera. From the first 4 figures(a-d) we could see that the object moved to the right side but the background did not change, which means although the object started moving, it was still in the middle part of the frame. Then from the last 10 figures(e-m) we could see that the camera rotated to the right, which means the system found the object was in the right part of the frame and rotated a certain angle to make the object stay in the middle part of the frame. The experiment results show that the system could track the object selected.

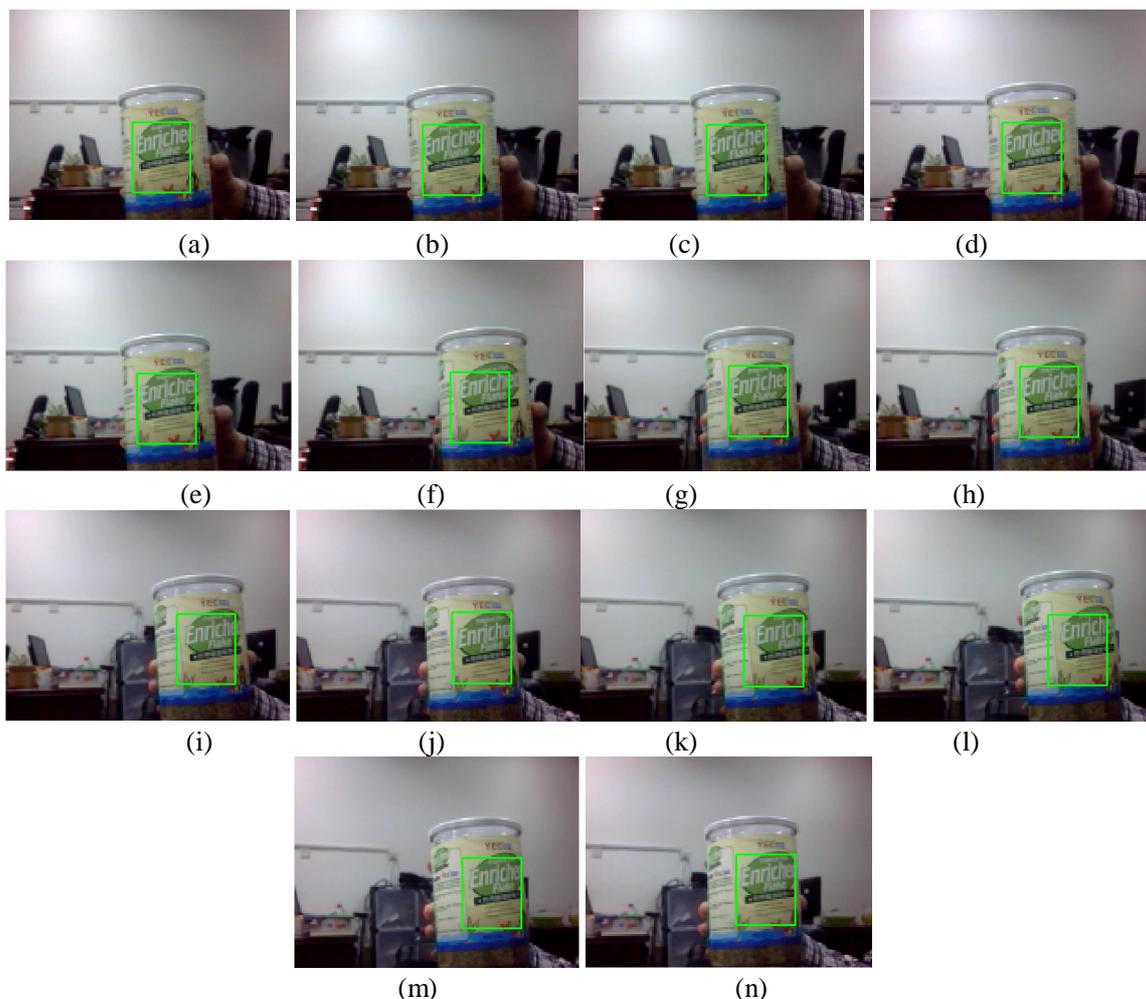


Fig 6. Frames of the tracking process of the system

## Conclusions

This paper mainly studies the real-time object tracking system design based on mean-shift algorithm. The whole system includes lower control system design and upper control system design. The lower control system could control the rotation of the camera and the upper control system realized the algorithm and could send control commands to the lower control system. The experiment results show that by using the tracking system designed by our paper, the camera could track the object selected well, which also shows that the mean-shift algorithm can be used as a real-time object tracking algorithm.

## Acknowledgements

This work was financially supported by Research Fund for the Doctoral Program of Higher Education of China under Grant 20120042120023, the University Innovation Team of Liaoning Province (LT2014006) and the Fundamental Research Funds for the Central Universities of China under Grant 20130042120027.

## References

- [1] D. Comaniciu, V. Ramesh, Mean shift and optimal prediction for efficient object tracking, Proceedings 2000 International Conference on Image Processing, p 70-3 vol.3, 2000
- [2] J. Jeyakar, R.V. Babu, K.R. Ramakrishnan, Robust object tracking with background-weighted local kernels, Computer Vision and Image Understanding, v 112, n 3, p 296-309, 2008

- [3] A.H. Mazinan, A. Amir-Latifi, Improvement of mean shift tracking performance using a convex kernel function and extracting motion information, *Computers and Electrical Engineering*, v 38, n 6, p 1595-615, 2012
- [4] J. Shin, S. Kim, S. Kang, S.W. Lee, J. Paik, B. Abidi, M. Abidi, Optical flow-based real-time object tracking using non-prior training active feature model, *Real-Time Imaging*, v 11, n 3, p 204-18, 2005
- [5] J. Shi, C. Tomasi, Good features to track, *IEEE Conference on Computer Vision and Pattern Recognition*, p 593-600, 1994
- [6] A.H. Mazinan, A. Amir-Latifi, Applying mean shift, motion information and Kalman filtering approaches to object tracking, *ISA Transactions*, v 51, n 3, p 485-97, 2012
- [7] J. S. Hu, C. W. Juan, J. J. Wang, A spatial-color mean-shift object tracking algorithm with scale and orientation estimation, *Pattern Recognition Letters*, v 29, n 16, p 2165-73, 2008
- [8] I. Leichter, M. Lindenbaum, E. Rivlin, Mean Shift tracking with multiple reference color histograms, *Computer Vision and Image Understanding*, v 114, n 3, p 400-408, 2010
- [9] H. Zhou, Y. Yuan, C. Shi, Object tracking using SIFT features and mean shift, *Computer Vision and Image Understanding*, v 113, n 3, p 345-52, 2009
- [10] T. Vojir, J. Noskova, J. Matas, Robust scale-adaptive mean-shift for tracking, *Pattern Recognition Letters*, v 49, p 250-8, 2014