2008 International Symposium on Parallel and Distributed Processing with Applications

Intelligent Target Tracking and Shooting System with Mean Shift

Xiang Xiang[1], Wenhui Chen[2], Du Zeng[3]
International School of Software[1], School of Electronic Information[2,3]
Wuhan University
Wuhan, Hubei, 430079, P.R.China
eglxiang@gmail.com[1], chenwh1986@yahoo.com.cn[2], xiaopengyou274@163.com[3]

Abstract

Tracking moving targets in sequence images is an essential key technology and one of the hot research topics in Computer Vision. This system is based on an embedded system platform named Embedded Star and makes full use of the OpenCV (Intel® open-source computer vision library)to implement and optimize the Mean Shift tracking algorithm. At last, it achieves the objective of real-time tracking and shooting of moving targets, and can be used in sports photography, real-time monitoring and so on. The test data has indicated that it can direct the camera through controlling the cloud terrace to track both rigid and non-rigid targets.

1. INTRODUCTION

In modern life, sports match video has become the favourite of the majority of the TV audiences, and occupies a large proportion in the existing Internet video resources. As appeared in the Beijing 2008 Olympic Games, audiences always expect better vision experiences during large-scale sports events. Therefore, the television cameramen have to be "the busiest people outside the playground". Since science technology should be people-oriented, researches on real-time tracking systems have become a hot issue.

At the same time, it has been long-cherished by the human being to make machines have visions. Objects in the real world are all three-dimensional, whereas the images human eyes can capture are two-dimensional[1]. In fact, the vision system of the human being can extract three-dimensional information from two-dimensional images[2], so the human being are able to perceive the three-dimensional world. However, it's rather difficult to make machines to have that kind of capability. During the process of the researches on this problem, the subject of Computer Vision springs up and develops gradually. Within the subject, tracking moving targets among sequenced images has been an indispensable key technique. One category of tracking algorithms is deterministic methods [3], and the other is probabilistic methods [4] , such as Mean Shift.

Mean Shift is a clustering approach proposed by Fukunaga and Hostetler[5] named "valley-seeking procedure" and clustering techniques have been extensively investigated[6-9]. However, Mean Shift was not focused until 20 years later when Cheng[10] introduced it to the image analysis community. Later Comaniciu and Meer[11-12] related it to feature space analysis. Recently, it has been widely used in obeject tracking[13-18] .

This paper illustrates the principle of Mean Shift and the implementation of the intelligent target tracking and shooting system. Its functions are as follows.

1) *Tracking and Shooting the Rigid Objects*

The Mean Shift algorithm which the system adopts supports to track and shoot the majority of rigid objects.

2) *Tracking and Shooting the Non-rigid Objects*

The Eigen value template is constantly being updated, and not greatly affected by the deformations of objects. Therefore, the system can track and shoot the non-rigid objects.

3) *Synchronous Video Compressing and Storage*

It can compress the tracking images with either xvid mpeg4 or divx mpeg4 codec and store them as video synchronously.

Therefore, this system can be used in sports photography, real-time monitoring, automatic driving, intercommunion between users and machine, video meeting and so on.

2. PRINCIPLE

The algorithm flow of Mean Shift in target tracking is illustrated in Figure 1. In the initial frame, a rectangle including all characteristics of the target, which is called target region of the tracking target, is selected by users through the mouse. The target region is also where the core function works. The region size is equal to the size of the bandwidth of the core function. Suppose the system tracks the target in colorized sequence images, then the range of the image pixel will be the space of the RGB color. According to the histogram method, the sub-space of the RGB color space(R, G, and B), can be fixed into k equal intervals, each of which is called a bin.

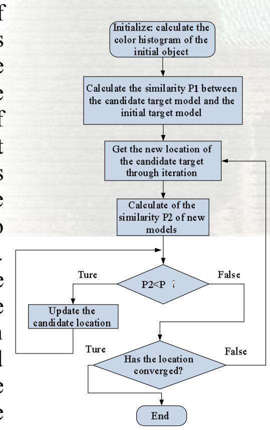


Figure1 Algorithm Flow

This work has been awarded the Second Prize in the 2008 Intel Cup Undergraduate Electronic Design Contest - Embedded System Design Invitational Contest. It is supported by Department of Education of the People's Republic of China, Department of Industry and Information Technology of the People's Republic of China, ©Intel Corporation, Evoc Intelligent Technology CO., LTD, and Wuhan University.

978-0-7695-3471-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ISPA.2008.167

Those bins constitute the feature space, and the number of the Eigen value is $m = k^3$.

For all pixels in the target region of the initial frame images, it is called the description of the target model to calculate the probability of each Eigen value in the feature space; for the subsequent images, it is called the description of the candidate model to calculate that in the candidate region where the target is possible to exist. Then the Epanechikov function is commonly selected as the core function. The similarity function is used to measure the similarity of the target model of the initial frame and the candidate model of the current frame. In the following, the Mean Shift vector of the target, which is obtained through calculating the max of the similarity function, will be the vector where the target transfers from the initial position. Because of the convergence of the Mean Shift Algorithm, as long as the Mean Shift vector is calculated iteratively, the ultimate point would be converged to the real position of the target(a static point). At the same time, the purpose of tracking would be achieved.

2.1. Description of Target Model

Suppose the center point of target region is x_0, n pixels in the region is expressed as $\{x_i\}_{i=1\dots n}$, and the number of the Eigen value bin is m. So for the target model, the probability density of the Eigen value $u = 1\dots m$ is estimated as

$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\frac{x_i^s - x_0}{h}\right\|^2\right)\delta\left[b\left(x_i^s\right) - u\right] \quad (1)$$

where $k(x)$ is the contour function of the core function; because of the impact of the shelter or the background, the pixels near the center of the target model is more reliable than those outside: $k(x)$ is the centre pixels larger weight values, and pixels away from the center smaller weight values; the role of $\left\|\frac{x_0 - x_i}{h}\right\|$ in the function $k(x)$ is to eliminate the impact brought by the calculating of targets of different sizes, so the target described by the ellipse is normalized into a unit circle; $\delta(x)$ is the Delta function, and the role of $\delta[b(x_i) - u]$ is to judge whether the color value of the pixel x_i in the target region belongs to the u st bin: the result 1 represents true, and 0 represents false; C is a standardized constant coefficient, making $\sum_{u=1}^{m} q_u = 1$, so:

$$C = \frac{1}{\sum_{i=1}^{n} k\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right)} \quad (2)$$

2.2. Description of Candidate Model

The region which is possible to include the moving target in the second frame and each subsequent ones is called as the candidate region, the central coordinate of which is y, also known as the central coordinate of the core function. The pixels in the region is expressed as $\{x_i\}_{i=1\dots n_h}$. The description of the candidate region is called as target candidate model. For it, the probability density of the Eigen value $u = 1\dots m$ is:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u] \quad (3)$$

Of which, $C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)}$ is a standardized constant coefficient.

2.3. Similarity Function

The similarity function described the degree of similarity between target model and candidate model, and the probability distribution of the two models is exactly the same in ideal circumstances. There are many such functions; however, Bhattachyarya coefficient in Mean Shift algorithm is a choice better than other similarity function. In this system, Bhattacharyya coefficient is adopted as the similarity function, which is defined as

$$\hat{\rho}(y) \equiv \rho(\hat{p}(y), \hat{q}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \quad (4)$$

Its value is between 0 and 1. The value of $\hat{\rho}(y)$ is greater, it represents that the two models are more similar. The candidate model of different candidate regions in the current frame should be calculated. Finally, the region making $\hat{\rho}(y)$ the max is the position of the target in the current frame.

2.4. Target Location

To make $\hat{\rho}(y)$ the max, the location of the target center in the current frame is set equal to that in the former frame, which is y_0. The optimal matching target would be searched from this point, the center is y. First, calculate the target candidate model $\hat{\rho}(y)$; second, expand the above formula (3) at $\hat{\rho}(y)$ to get Taylor series, so the Bhattahcyarya coefficient is similar to:

$$\rho(\hat{p}(y), \hat{q}) = \frac{1}{2}\sum_{u=1}^{m} \sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (5)$$

And $w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}}\delta[b(x_i) - u]$ (6)

In the Formula (5), only the second addition part is changed with y. Therefore, here the second part would be analyzed. Suppose:

$$f_{n,K} = \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (7)$$

This expression is similar to the density estimate of core function, and the only difference is that it includes an extra weight w_i. So in order to make the Formula (5) the max, Formula (7) should first be made the max. In the following, the Mean Shift vector of Formula (7) would be calculated to get the vector from the center y_0 of the candidate region to the real target region y :

$$m_{h,G}(y) = y_1 - y_0 \frac{\sum_{i=1}^{n_h} x_i w_i g(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2)}{\sum_{i=1}^{n_h} w_i g(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2)} - y_0 \qquad (8)$$

where $g(x) = -k^{'}(x)$, $m_{h.G}(y)$ is the vector of the target center from y_0 to y , Mean Shift approach is starting at y_0. In the Mean Shift Algorithm, the search moves from y_0 to where the color changes most between the two models. Hence, this kind of search is better than the busy but blind searches adopted by the common algorithms.

3. IMPLEMENTATION

3.1. Hardware design

The system adopts the Embedded Star as the core platform, which is expanded with the camera, cloud terrace, video capture card, LCD touch screen, MCU, tripod, and other peripheral equipments. Its structure is illustrated by Figure 2.

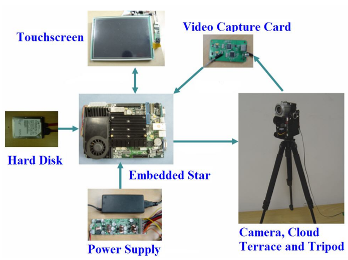


Figure 2 Hardware design

- Video collection: The analog signals generated by the camera is translated to digital signals through video capture card, and then passed through USB.
- Video output: All the interactions and outputs are friendly displayed on the 8.6-inch LCD touchscreen.
- Shooting Control: The platform uses RS-485 to control the cloud terrace. The decoder uses AT89C52 as a control chip and compliances with the strong anti-jamming communication protocol PELPELCO-D to achieve the wide-angle shooting with horizontal angle of 355 °, and vertical angle of 100 °.

3.2. Software design

First, the system uses the DirectShow to capture video; second, after the user selects an object, it begin to track and shoot, based on the improved Mean Shift algorithm; third, it controls the cloud terrace through the serial port commands, and finally delivers the task to the hardware. The software design is illustrated by Figure 3.

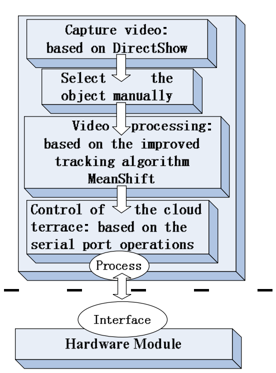


Figure 3 Software design

- Tracking algorithm: Mean Shift algorithm
- Based on Intel® Open Source Library OpenCV
- Based on OpenMP Multi-core Programming

3.3. Key features

The finished system is displayed by Figure 4:

Figure 4 Finished System

Compared with existing systems, the intelligent target tracking and shooting system has the following key features:

Table 1 Key Features of the System

1	Tracking Rigid Targets	Real-time (less complex, run faster); Robust (support shelter & distortion)
2	Tracking Non-rigid Targets	Support continuous change of shape & size (automatic adaptation of targets' shape)
3	Control of the Cloud Terrace	Control through serial port, Able to turn 355° horizontally, 100° vertically
4	Multi-format Synchronous Video Compressing and Storage	Support xvid mpeg4 /divx mpeg4 codec, can be stored as .avi video
5	Shelter Processing of a Certain Extent	Support partial (< 50%) shelter
6	Multi-core Programming and Optimization through Intel® Tools	Parallel programming through OpenMP, Performance analyze through Intel® VTune
7	Open Source Frame Based on Intel® OpenCV	Implement and optimization of library functions, encapsulated as API for other developers

4. EXPERIMENT

4.1. Test Environment

The Embedded Star, the Hitachi hard disk, the touch-screen LCD, the power adapter and DC to the ATX power modules; the cloud terrace, the tripod, the analog video camera, and the video capture card.

4.2 Test Solution and Result

4.2.1. Test 1: Tracking performance test

In order to test the performance of real-time tracking and shooting, the authors test both rigid and non-rigid objects. The result is illustrated by Table 2 and Figure 5 is the screenshot of the tracking test of human body.

Table 2 Result of tracking performance test

Test object	Real size of the selected object	Distance between the camera & object	Test result
Beverage bottle	26.2× 9.3 (cm)	About 5m	No matter the targets moves up or down, left or right, it can makes the target locate in the center of the eyesight.
Human body	170.4× 38.7(cm)	About 10m	

Figure 5 Tracking result of the frame 0, 7, 21,35,45,53

4.2.2. Test 2: Shelter test

In the process of real-time tracking and shooting, the object is likely to be sheltered, and the authors take this problem into account in the design of algorithm, doing a number of improvements and optimizations. The result of shelter test is illustrated by Table 3, which reflects that the system is robust. And Figure 6 is the screenshot of the shelter test of human body.

Table 3 Result of shelter test

Test targets	Selected target size	Blocking size	Test result
Human body	12×2.5 (cm)	About 50%	The system can make the target in the center of the eyesight.
Human body	12×2.5 (cm)	About 80%	

Figure 6 Tracking result of the frame 51,73,78,83

4.2.3. Test 3: CPU occupancy rate test

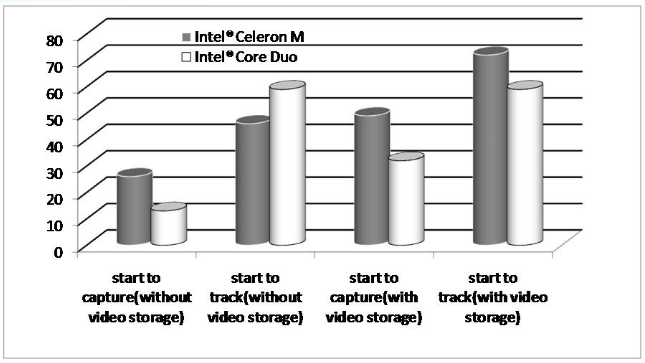
CPU occupancy rate of the entire application program, can reflect the run-time usage of resources and load situations. The test result is illustrated by Table 4.

Table 4 Result of CPU occupancy rate and memory usage test

Test targets	CPU occupancy rate
Beverage bottle	59%
Human body	67%

Especially, the authors compare the CPU occupancy rate between Core Duo and Celeron M platform. For beverage bottle, the result is illustrated by Table 5; for human body, the result is similar.

Table 5 CPU occupancy compare



5. CONCLUSION

In Test 1 and 2, the rigid target tracking indicates that the system is real-time because Mean Shift is less complex and more computationally efficient, and that the system is robust because Mean Shift supports either partial(<50%) shelters or distortions; especially, the non-rigid targets tracking indicates that the system supports continuous changes of both the shape and size for Mean Shift's automatic adaption of the target's shape. In addition, Test 3 proves much lower CPU occupancy rate on Core Duo Platform than on Celeron M Platform.

However, there still exist a number of shortcomings: first of all, because of Mean Shift's own disadvantages, the effect of tracking rapid-moving objects is not very good, which leads to the result that the tracking of targets has a certain speed limit; furthermore, since Mean Shift is based on the Eigen value of the color histogram, the system seems easily lose the target when the color of the selected target is similar with the background or the background is too complex. Therefore, in the coming time, the authors will try their best to solve the above deficiencies to improve the performance of the system.

REFERENCES

[1] Oliviver Faugeras. Three-dimensioanl Computer Vision: A Geometric Viewpoint. MIT Press, 1993.

[2] DG Lowe, "Three Dimensional Object Recognition form Single Two-dimensional Images," Artificial Intelligence, 31, 3(March 1987), pp. 355-395.

[3] G. R. Bradski, "Computer Vision Face Tracking for use in a Perceptual User Interface", IEEE Workshop on Applications of Computer Vision, pp.214-219.

[4] M.Isard and A. Blake, "Condensation - Conditional Density Propagation for Visual Tracking", Int. Journal of Computer Vision, pp.5-28, 1998.

[5] K. Fukunaga and L. Hosteler. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, IEEE Tran. IT., Vol. 21, 1975, pp 32-40.

[6] G. H. Ball, "Data analysis in the social sciences: What about the details?," in Proc. FJCC, 1965, pp. 533-559.

[7] R.O.Duda and P.E. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[8] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall, 1988.

[9] K. Fukunaga, Introduction to Statistical Pattern Recognition. Second Edition. San Diego, CA: Academic Press, 1990.

[10] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.17, 1995, pp 790-799.

[11] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications", In Proc. International Conference on Computer Vision, 1999, pp 1197-1203.

[12] D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, 2002, pp 603-619.

[13] D. Comaniciu, and Ramesh V, "Mean shift and optimal prediction for efficient object tracking," In: Mojsilovic A, Hu J, eds. Proc. of the IEEE International Conference on Image Processing, pp.70-73, 2000.

[14] D. Comaniciu, Ramesh V, and Meer P, "Real-time tracking of non-rigid obejcts using mean shift," In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-149, 2000.

[15] D. Comaiciu, V. Ramesh, P. Meer, "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5), pp.564-575, 2003.

[16] Collins RT, "Mean Shift blob tracking through scale space," In: Proc. of the Conference on IEEE Computer Vision and Pattern Recognition, pp.18-20, 2003.

[17] Shan C, Wei Y, and Tan T et al, "Real time hand tracking by combining particle filtering and mean shift," In: Proc. of the 6th IEEE International Conference on Automatic Face and Gesture Recognition, 17-19, May, pp.669-674, 2004.

[18] Maggio E, and Cavallaro A, "Hybrid Particle Filter and Mean Shift tracker with adaptive transition model," In: Proc. of IEEE Signal Proc. Society International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, March pp. 19-23, 2005.