

Object Tracking System for Video Recording based Qt and OpenCV

Zhiyong Tang, Kun Liu, Zhenji Yang, Zhongcai Pei, Zeyang Zhang

Beihang University, Beijing 100191, China

Keywords: Qt, OpenCV, Object tracking, TLD.

Abstract. In this paper, an object tracking system was introduced for video recording based Qt and OpenCV. The system can track any moving target in dynamic background. The tracking algorithm was designed based on TLD tracking algorithm and some refactoring was implemented. The architecture of the system is clear and the system runs stability. The UI of the system was visualized and convenient. The system can be widely used in military reconnaissance, monitoring system and sports event recording.

1. Introduction

The object tracking system was widely used in the field such as military affairs, security, sporting events, etc. In these fields, a real-time, stable object tracking system was demanded. In this paper, an object tracking system was proposed, it can track any object in any long time. The system was made by C++ language, Qt library and OpenCV library. The algorithm in the system was based on TLD algorithm, and some improvement about TLD algorithm was proposed. The system can interact with user expediently.

2. Overview of the Structure

The target tracking system uses the computer vision library OpenCV version 3.1 and the graphical interface library Qt version 5.1. The computer vision library OpenCV sponsored by Intel Corporation and participate in the development, a powerful image processing capabilities [1]; The GUI library Qt is Digia company's products are widely used in the development of GUI programs. Because OpenCV and Qt are cross-platform software, contained in the operating system and that helps computer Microsoft Windows family of operating systems, but also for a variety of GNU/Linux distributions [2]

In order to get the maximum CPU utilization and simplify the application logic, the system is designed to run multiple threads in parallel. Multi-threaded implementations each thread is packaged into a separate class, and inherits the Qt QThread class. In the main() function instantiate Thread class, by calling the start() method inherited from QThread class to start a thread. The threads can communicate with each other by using QObject::connect() method to connect the signal transmission function and slot function.

The multithreading includes a front-end user interface thread, an object tracking algorithm thread and a camera platform controller communication thread. The UI thread is responsible for tracking and display real-time images captured by the camera, and receive user's various input instructions. The tracking thread is responsible for reading frames from camera, processing the tracking algorithms, passing the post-processed frames and tracking box to the user interface thread and transmitting the calculated target coordinates in the current frame picture to the camera platform controller communication thread. The camera platform controller communication thread is responsible for communication with the camera platform controller, it will transmit the target coordinates to the camera platform controller, and the camera platform controller will control the two-degrees freedom camera platform, so that the target to be tracked will always be maintained in the screen.

The relationship between the computer multithreading, camera platform controller and the two-degrees freedom camera platform was shown in Figure 1.

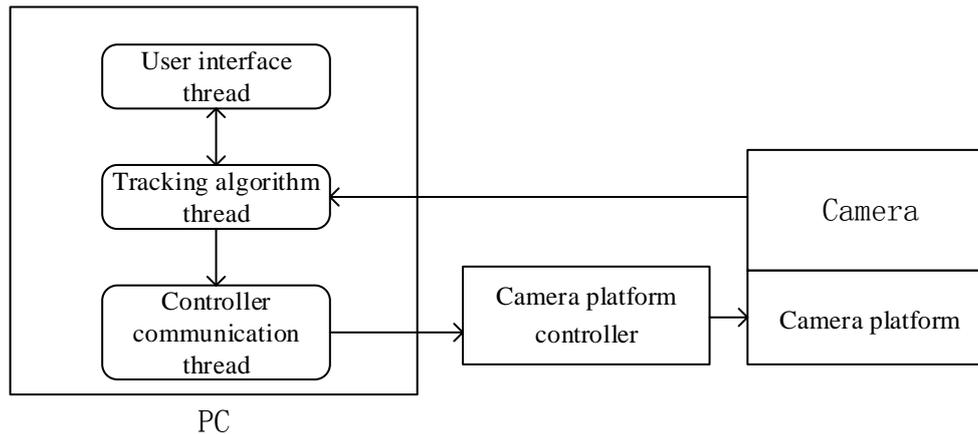


Fig. 1 The flow chart of each threads

3. User Interface Design

The design of graphical user interface includes tracking results superimposed video output, camera control buttons, training button, import and export target model button and target select button. This system provides a variety of target selection mode, including automatic target selection, manual target selection and computer-aided target selection. The automatic target selection mode automatically matches the search target in the screen based on the trained target model; the computer-aided target selection mode detects all moving objects in current frame by the moving object detection algorithm, and then the object to be tracked will be selected by user manually; in manually select mode, user should select the target by mouse manually.

During initialization of the user interface thread, a frame buffer will be set, and the frame buffer address will be passed to the tracking algorithm thread through inter-thread communication. When tracking thread is running, the target tracking box will be transmitted to the user interface thread at the end of each frame processing, The UI thread will visit the frame buffer every 30ms, and put the frame buffer to the screen.

4. Tracking algorithm design

The tracking algorithm is mainly based on the Zdenek Kalal's "Tracking - Learning - Detection (TLD)" algorithm [3] and some improvements were proposed. TLD algorithm is a novel and efficient long time visual tracking algorithm, the algorithm consisted of optical flow tracker, detector and integrator, and has online target feature learning functions. The algorithm can resist the influence made by occlusion, illumination changes, scale changes, angle changes and other factors during target tracking process The flow chart of target tracking algorithm was shown in Figure 2.

4.1 Optical flow tracker.

The optical flow field is a projection of the move field in the two-dimensional image, and the optical flow is the pixel motion vector in grayscale mode. The different between simple optical flow method is that the TLD algorithm using optical flow tracking twice for each frame. Firstly, the algorithm selects 100 tracking points in previous frame, and then track these points from previous frame to current frame, get the position of these points in current frame, and then track these points from current frame to previous frame, get the position of these points in previous frame. So that we will get three groups of tracking points, they are tracking points P_1 in the previous frame, tracking points P_2 in the current frame obtained from points P_1 , and tracking points P_3 obtained from points P_2 . We can remove half of the points by comparing the similarity of each points between P_1 and P_2 , and then remove half of the rest points by comparing the distance of each points between P_1 and P_3 . Thus the 100 track points will be left up to the last 25 points. These feature points will be used to estimate the object region in the current frame. And we can estimate the change of the scale of the target region by calculate the change of distance of each feature points.

4.2 Triple detector.

The triple detector consists of variance classification, random fern classifier and nearest neighbor classifier, it can filter the image patch that most likely be the target object in the frame. Before running the triple detector, it is needed to split the frame into image patch that has different scales and different locations.

The variance classifier will calculate the variance of the target area first, and then calculate the variance of each image patches, then exclude the patches those the variance is less than half of the variance of the target area. The rest of image patches will be transmitted to the random fern classifier. Variance classifier can exclude the background patches effectively.

The random fern classifier is similar to random forest classifier, each a separate random fern can be seen as a weak classifier, the combination of several random ferns together to form a strong classifier. Random fern classifier can effectively exclude a large number of image patches that are not the target area. The rest of the image patches will be sent to the nearest neighbor classifier.

The nearest neighbor classifier judges a patch whether the target area or not by calculation the correlation between the sample image and patches. The algorithm has the advantage that simple and accuracy. The nearest neighbor classifier will save the target area of each frame as positive sample, and then compare the sample with image patches. Finally, output the image patches that go through the nearest neighbor classifier as the output of the triple detector.

In the original TLD algorithm, the detector will search for the target in global frame image, which is bound to affect the real-time performance of the algorithm, so some improvement has been proposed in this system. In this system the detector will just search the area near the last target region, and it can improve the search speed distinctly. In addition, in the original TLD algorithm, the number of samples will continue to increase in the long tracking process, which affects the efficiency of the algorithm, so this system uses a mechanism of the sample scores in each frame process, the samples that has the largest similarity of the selected target object will acquire one score, and the samples have low scores will be removed, so we can maintain the sample in a fixed amount. The mechanism guarantees the algorithm's real-time performance. Figure 2 is the sketch of the triple detector. We can see that the detector can detect the object region effectively.



Fig. 2 The sketch of the triple detector

4.3 Integrator.

The integrator will remove the target region that clearly wrong, such as the region that has sudden change in location or scale. It will integrate the output of the optical flow tracker and the triple detector and output the result as the final result of target region.

Integration is also responsible for handling the case of failure situation of the optical flow tracker for triple detector. When one fails, the integrator will use the other one result as the final result. When both of the two parties are invalid, the integrator will report the target moving out of the tracking range, close the optical flow tracker and variance classifier and use the random fern classifier and nearest neighbor classifier to scan the target object in the frame until the target appears again.

4.4 Learning process.

After tracking processing of each frame, the algorithm performs a learning process. The target area will be used as the positive samples, and the region near the target region will be used as the negative samples. The positive and negative samples will be used to train the random fern classifier and nearest neighbor classifier as shown in Figure 3.

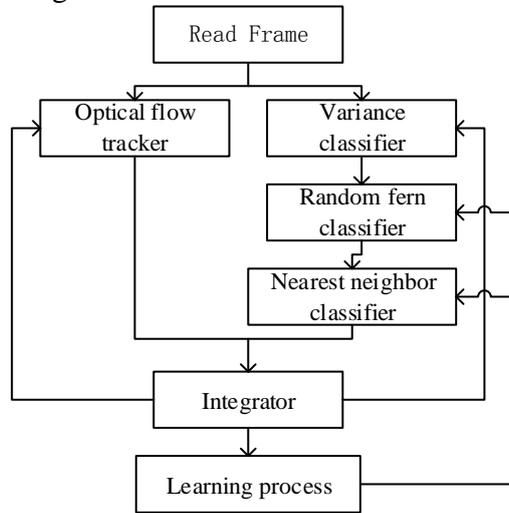


Fig. 3 Target tracking algorithm flowchart

5. Summary

After testing, the system can complete any target tracking tasks effectively. It has intuitive and convenient user interaction and can be widely used in military reconnaissance tracking, security monitoring, sporting events recording and other situations that require real-time tracking. Through improved the TLD algorithms, the real-time performance has been enhanced. Furthermore, since this system it is built with open source cross-platform software, so it can be easily transplanted to different operating environment. For example, it can be transferred into the UAV platform and augmented reality systems. Because the system uses a modular design and low coupling degree, therefore it can easily be secondary development, and has the advantages of easy maintenance and stable operation.

References

- [1] Liu D. Embedded video surveillance system based on Qt4 and OpenCV [J]. Modern Electronic Technology, 2013, 36(8): 104-108.
- [2] Wang Y, Li Y, Huang L. Wireless video surveillance system based on Qt and OpenCV [J]. Microcomputer and Application, 2015, 34(10): 67-69.
- [3] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2012, 34(7): 1409-1422.
- [4]. Xie J Z, Liao K, Xiong J L, et al. A Method for Artificial Mark Recognition in Augmented Reality System[C]. Applied Mechanics and Materials. 2014, 568: 617-621.