VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Chengxi Li

RESEARCH ON CORRELATION FILTERS OF VISUAL TRACKING ALGORITHMS

Information Technology
2017

FOREWORD

Up to now, I have studied in Vaasan Ammattikorkeakoulu (Vaasa University of Applied Sciences) for two years and it is one of the most memorable experiences in my life. I would like to take this opportunity to express my appreciation to everyone who has helped me.

First, Dr. Yang Liu is the supervisor of my thesis and the study life. Without his help, I am afraid that I could not reach so far and have a clear plan of my future career. I also learn to be modest and understand that there is always someone who is better than me, and I should keep forging ahead. I am full of gratitude to him for offering me a chance to study aboard and a better platform to make progress.

Furthermore, I would like to express my appreciation to all the staff in VAMK including Dr. Chao Gao, Mr. Jukka Matila, and Mr. Santiago Chavez Vega. I also want to express my gratitude to Dr. Jun Shu in Hubei University of Technology and the other teachers who have given me direction and contributed a lot to my thesis.

Also, I would like to thank all the fellows in the Robotics Lab including Xiaotian Li, Xiuyang Li and Ziye Zhou for helping me in research and study. In particular, I want to give thanks to my girlfriend Jinling Chen for her constant support. I sincerely hope all the fellows will have great achievements and bright futures.

Chengxi Li

Vaasa, Finland

26/11/2017

VAASAN AMMATTIKORKEAKOULU

UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Chengxi Li
Title	Research on correlation filters of visual tracking algorithms
Year	2017
Language	English
Pages	68
Name of Supervisor	Yang Liu

This thesis focuses mainly on the visual tracking algorithm, the Kernelized Correlation Filters (KCF) algorithm and Discriminative Scale Space Tracker (DSST) algorithm. They are widely applied in many fields. Moreover, the visual tracking framework with KCF and DSST outperforms in the perspective of tracking speed and accuracy, which has drawn increasing attention.

Although these target tracking algorithms achieve long-term and accurate tracking of the target, there are still many problems in the practical application environment such as stability, adaptability and real-time performance. In view of these problems, some improved methods are proposed. Aiming at the problem that the detection module in the algorithm needs to detect the lack of accuracy of the fast-moving object, a Kalman filter is used to estimate the approximate appearance area of the target in the current frame. This approximate area is taken as the target detection area of the algorithm. Although the speed of the algorithm has a certain impact, but the accuracy of the algorithm has a certain degree of improvement

In this thesis, the Kalman filter is proposed to be utilized in the visual tracking framework with KCF, which is more robust to movements of the target area. Furthermore, the simulation results with test beds based on Matlab and OpenCV 3.3 show that the proposed framework outperforms the conventional KCF and DSST-based visual tracking framework. Experiments show that the two algorithms have their own advantages in the matching rate, the matching speed and the number of frames successfully tracked. And the improved algorithms are more effective than the original ones.

Keywords Correlation Filter, Computer Vision, Visual Tracking

CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

KCF	**Tracking with Kernelized Correlation Filters**
DSST	**Discriminative Scale Space Tracker**
DARPA	**Defense Advanced Research Projects Agency**
UAV	**Unmanned Aerial Vehicle**
OpenCV	**Open Source Computer Vision Library**
VSAM	**Video Surveillance and Monitoring**
NSFC	**National Natural Science Foundation**
MILBoost	**Multiple Instance Learning Boost**
TLD	**Tracking-Learning-Detector**
PSR	**Peak Side Lobe Ratio**
FFT	**Fast Fourier Transform**
DFT	**Discrete Fourier Transform**
VOT	**Visual Object Tracking**
OTB	**Online object tracking**
API	**Application Programming Interface**
MOSSE	**Minimum Output Sum of Squared Error Filter**
CN	**Color Name**
LCT	**Long-term Correlation Tracking**
SRDCF	**Learning Spatially Regularized Correlation Filters for Visual Tracking**

CSK	**Exploiting the Circulant Structure of Tracking-by-detection with Kernels**
DCF	**Discriminative Correlation Filter**
RGB	**Red, Green, Blue**
HSV	**Hue, Saturation, Value**
CFT	**Correlation Filter-based tracking**
HOG	**Histogram of Oriented Gradient**

LIST OF APPENDICES

1. INTRODUCTION

1.1 Purpose

With the development of information technology in recent decades, computers have infiltrated into almost all areas of people's life and work as the most significant tools of informationization. Target tracking, considered as one of the key technologies of information society, has attracted increasing attention from industrial and academic communities. Visual tracking necessarily requires greater accuracy in the application, tracking speed, and other accuracy. Therefore, this paper introduces the technology of visual tracking, and applies and improves the correlation filter tracking algorithm in practice. The feasibility of the algorithms is tested through experiments and simulations. Moreover, the entire experiment shows the possibility of using visual algorithms for a wide range of future social applications.

1.2 Overview Structure

This thesis consists of six chapters. In the first chapter, we mainly introduce the relevant research background and related background knowledge, including the introduction of computer vision and mainstream vision tracking algorithm, and their current research and application. The second chapter gives the general structure of visual tracking algorithm, introduces the theoretical basis of the project, logic and classic mainstream algorithms. The third chapter introduces and analyzes the theoretical basis of the correlation filter algorithms and expounds how to use the theory to modify and enhance the algorithm. The fourth chapter introduces KCF algorithm, including mathematical theory and test results. The fifth chapter introduces DSST algorithm, including mathematical theory and test results. The sixth chapter, based on the same test set, tests and compares the above two algorithms in terms of accuracy and speed, so that the KCF algorithm can be applied and improved according to the current situation of the author. The seventh chapter introduces the Kalman filter and the Kalman filter based on the kcf algorithm. The eighth chapter briefly discusses the difficulty of completing this thesis, as well as the future direction and expectations.

1.3 Introduction to Computer Vision

It is well-known that humans obtain the majority of information in the world through their eyesight. Vision is one of the important means of human perception and interaction. With the development and application of computer science, human beings gradually try to use the machine to simulate the function of human vision, so that the machine has the analytical ability of artificial intelligence. Due to the research and development of computer technology, human beings can convert environmental images into digital signals through cameras and realize the whole process of digital signal processing by computers. As a result, a whole new subject of computer vision emerged. Computer Vision's main research direction is to use cameras and computers to replace the human eye to identify, track and detect the target, and further analysis and research objectives, to achieve the purpose of artificial intelligence, involving mathematics, image processing, signal processing and computer applications and many other disciplines.

Since the 1950s, people began to study the statistical pattern recognition of two-dimensional images. In 1965, LR Roberts extracted three-dimensional structures of cubes such as cubes, wedges and prisms from digital images through computer programs. People conducted in-depth study of the three-dimensional structure. The scope of the study ranged from the edges, waiting for the extracted corners, to the geometric elements of lines, planes, surfaces, etc. - until the image brightness, texture, motion, and imaging geometry and to establish a variety of data structures and inference rules. In the mid-1970s, the Artificial Intelligence Laboratory at MIT officially launched Machine Vision Courses. In 1977, David Marr proposed a computational vision theory that is different from the "building block world" analysis - the famous Marr vision theory. This theory became a very important theoretical framework in the field of machine vision in the 1980s. From the eighties onwards, the subject visual theory framework and the visual integration theory framework not only produced the concept of object recognition based on the perception feature, but also appeared many new research methods and theories. The general two-dimensional information processing, or for the three-dimensional image model and algorithm research have greatly improved. In the 1990s, machine vision theory was further developed and began to be applied in the industrial field. At the same time, the application of machine vision theory in multi-view geometry has been rapidly developed.

During this period, the research on computer vision has gone through the development stage from laboratory to practical application. With the rapid development of such disciplines as artificial intelligence, parallel processing and neuronal networks, the computer vision system has been further promoted and involved in many complex visual processes. In a sense, this is the golden age of computer vision.

1.3.1 Open Source Computer Vision Library (OpenCV)

In the practical stage of this thesis, we will use a large number of OpenCV /1/function library, and hereby briefly explain its content.

OpenCV is an open source library for image processing, analysis, and machine vision. Whether you are doing research or commercial applications, OpenCV can be your ideal repository because it's completely free for both.

The library is written in C and C ++ and can be run on windows, Linux, mac OS systems. All of the library's code is optimized and computationally efficient because it is more focused on designing as an open source library for real-time systems. OpenCV uses C language to optimize, and, in the multi-core machine above, it will run faster. One of its goals is to provide a friendly machine vision interface function that enables complex machine vision products to accelerate. The library contains over 500 interface functions spanning areas such as industrial product testing, medical image processing, security, user interface, camera calibration, 3D imaging, machine vision and more.

Figure 1. OpenCV Logo/1/

1.4 Introduction to Visual Tracking

It is generally acknowledged that human beings obtain the majority of information from the outside world through vision. The visual observation is an important means of human perception. With significant development of computer vision, the intelligent machines can mimic the human visual perception behaviors. Specifically, the intelligent machines observe the environment with deployed cameras, process the observed images and learn to accomplish various objectives. In the field of computer vision, the target tracking in video has also become an increasingly important topic.

Due to the improvement of computer processing speed and the development of computer vision theory, video target tracking techniques are adopted in multiple fields. The target tracking includes target detection and feature extraction, which can obtain target motion parameter information (location, speed, etc.) and target behavior. As a multidisciplinary and advanced technology, moving target tracking combines theoretical knowledge in many different fields. At present, target tracking is widely used in the following aspects:

(l) Intelligent monitoring of video targets

In recent years, monitoring cameras are widely deployed for surveillance, which generate videos containing a lot of redundancies and consuming huge memories for storage. To tackle this problem, intelligent video target monitoring system is proposed to extract useful information from raw videos, thus better utilizing the videos with much less memories.

(2) Human-computer interaction technology

Human-computer interaction mainly depends on corresponding software and external devices of the computers. Conventional external devices are keyboards and monitors. With the development of computer technology, human-computer interaction becomes more intelligent, where the computers can respond to people by observing physical features. For example, Microsoft's Kinect technology, as well as smart phones with complex human posture recognition technology and so on. These technologies simplify manual operation and bring convenience and enjoyment to human life.

(3) Intelligent traffic detection

With the rapid development technology and urban construction, has been continuously expanded and traffic volume has increased rapidly. However, along with road congestion, frequent traffic accidents and sudden cases increase, so only with more intelligent tools could strengthen traffic management and safety precautions. On this occasion, the visual tracking can capture high-definition images. The rush hour can help the relevant departments to grasp the traffic flow timely in each section and prevent the serious traffic jam in time, and grasp the traffic safety situation.

(4) Industrial construction

In the factory production, robots with machine vision function replace human beings to finish the sorting work of products in the production line. In this way could save a lot of human resources and improve the efficiency of industrial production. In industrial production, the advantages of machines instead of the human eye is very obvious, such as strong spatial resolution of machine vision, strong grayscale resolution, fast, continuous work and so on. So if we can do a good job "downsizing efficiency", machine vision in labor-intensive manufacturing companies can quickly replace the manual.

In addition, target tracking is also applied in medical diagnosis, national defense construction and meteorological analysis and so on. Generally speaking, target tracking techniques extract the objects of interest through processing the raw data.

1.5 Programming Tools used in this thesis

1.5.1 Python

Python/2/ is an open source scripting language that places special emphasis on the speed of development and the clarity of the code. It can be used to develop a variety of programs, from simple script tasks to complex, object-oriented applications have a place to show their talents. Python is also considered as the best language to get started as a beginner programmer because it's free, object-oriented and extensible while enforcing strict coding standards. For this paper focus on algorithm implementation, python is undoubtedly the best implementation tool.

1.5.2 MATLAB

MATLAB/3/ is an abbreviation of Matrix Laboratory and is a commercial math software produced by The MathWorks, USA. MATLAB is a high-level technical computing language and interactive environment for algorithmic development, data visualization, data analysis, and numerical computation. In addition to common functions such as matrix operations, drawing functions / data images, MATLAB can be used to create user interfaces and programs written in other languages, including C, C ++, Java, Python, and FORTRAN.

Although MATLAB is mainly used for numerical calculations, it is also suitable for a wide range of applications such as control system design and analysis, image processing, signal processing and communication, financial modeling and analysis using a large number of additional toolboxes. There is also an accompanying software package Simulink, provides a visual development environment, commonly used in system simulation, dynamic / embedded system development and so on.

In the later chapters of the dissertation, using MATLAB to build a mathematical model of tracking algorithm is undoubtedly the most suitable and most powerful tool.

1.6 Overview of Thesis

In this thesis, the correlation filtering target tracking framework is the core of the paper, and the application of tracking module in video target tracking is taken as the main research content. Through elaborating, improving and optimizing the target tracking algorithm, it tries to optimize the specific application problems in the field of video tracking. The main contents of the thesis include the following aspects:

(1) The classical correlation filtering detection algorithm is analyzed and compared. The paper elaborates several commonly used theories of target detection algorithms, which pave the way for the following chapters to enhance and modify the algorithms.

(2) The theoretical content and system framework of KCF target tracking algorithm are introduced. The theoretical knowledge and application of KCF are introduced in detail, and the research field and development direction of KCF tracking algorithm are discussed.

(3) The theoretical content and system framework of DSST target tracking algorithm are introduced. The theoretical knowledge and application of DSST are introduced in detail, and the research field and development direction of DSST tracking algorithm are discussed.

(4) According to the actual situation and projects, deciding the algorithm which is more suitable to improve. Aiming at a series of problems appearing in the algorithm, and an improved scheme is proposed. An improved method based on Kalman filtering is designed for KCF tracking, which could improve the accuracy.

2. VISUAL TRACKING ALGORTIMS

Target tracking is one of the main research directions in the field of computer vision. Through various methods, it estimates the state of continuous vision in order to get the position, contour and trajectory of the target, and finally provide the advanced target state analysis basis. With the advent of high performance computers and cheap video capture devices, and the increasing demand of computer vision technology, the research of target tracking has a wide range of prospects.

2.1 Related research of Visual Tracking

Although there are still many challenges in video tracking, there is not yet an absolutely robust live video tracking algorithm. However, with the constant improvement and development of core algorithms such as computer vision, sensor technology and image processing, and many core algorithms have been proposed in the field of video tracking. Some achievements have been made and rapid progress has been made in this field.

In general, the US military and the NSFC have paid special attention to the research and application of target detection, recognition and tracking algorithms, especially in the research and application of complex environments. In 1991, the Carnegie Mellon University of America funded the use of visual information on UAVs for defensive system research funded by the Defense Advanced Research Projects Agency (DARPA). In 1997, DARPA again invited many U.S. universities to participate in the research and development of Video Surveillance and Monitoring (VSAM), a major project of the video surveillance system/4/.

Subsequently, many foreign universities and research institutes and researchers also joined the video target detection and tracking algorithm among them. For example, Ercan/5/ at Pennsylvania State University used Gibbs Markov random fields and 2D Gaussian distributions to model the texture and color of the target respectively for tracking. Katja/6/ at the Swiss Polytechnic University proposes an adaptive particle filter algorithm that uses a weighted color histogram method that takes into account factors such as color and appearance variation to effectively handle rapidly moving non-rigid targets in complex environments. Eindhoven University of Technology proposes a feature background subtraction method based on luminance statistics/7/. The corresponding background image is reconstructed according to the global and local luminance changes of the input image, and the residual background luminance variation is calculated by the luminance statistical method. Babenko/8/ proposed an online MILBoost algorithm at the University of California, San Diego. At the same time, a group of positive negative samples are extracted and put into the training pool respectively. The weak classifiers are trained by using the likelihood function. The classifier target position is estimated, the tracker is updated, and the MIL lookup table is updated. MIL can better handle drift issues. Kalall/9/ of the University of Surrey proposed a tracking-learning detector (TLD) framework that effectively combines detectors and trackers, learning continuously using PN learning/10/ to lock targets, obtaining the latest appearance characteristics of the target, and using the results of the learning Probe and tracking updates. Jame/11/ at the University of California proposed using a self-learning mode to automatically select reliable frame to learn. Custom progress learning must first determine the simple frame containing the target, when faced with online tracking marked as simple or complex, learning also need to review the previous frame, tracking algorithm, whether online or offline learning can achieve higher accuracy.

In short, many universities and research institutes carry out video target detection and target tracking research, and the research results are impressive.

2.2 Composition of the target tracking system

The purpose of the target tracking is to estimate the target's state in the continuous video from the information of the initial frame. As shown in the figure, a typical tracking system mainly includes three parts: appearance model, motion model and search strategy/12/.

(1) Appearance model

The appearance model mainly consists of the target representation and the statistical model. The target representation is mainly used to design a robust target descriptor through different kinds of visual features. The main purpose of the statistical model is to confirm the goal through the learning of these descriptors.

(2) Motion model

The motion model is a dynamic estimation problem, mainly used to describe the change in the state of the target over time.

(3) Search strategy

Search strategy is mainly to find the target in the current call the most likely location of the method

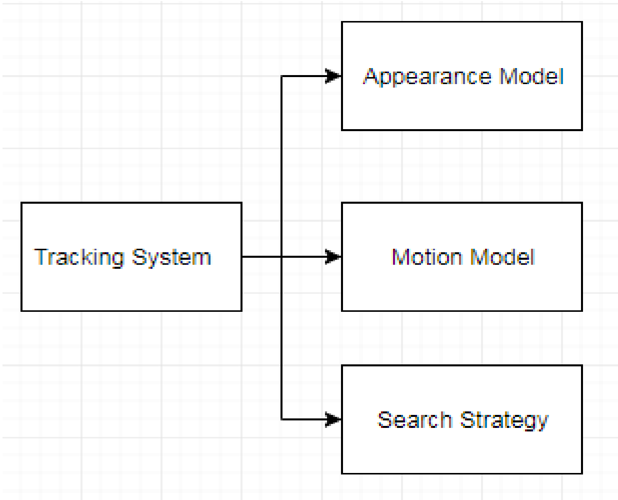


Figure 2. Universal structure of target tracking system

2.3 Difficulties in Tracking Algorithms

/13/At present, the main difficulty of the target tracking algorithm lies in the problem of data association. The complexity of the background, the change of the target ratio, the similarity of the target color and the background color, the stability of the background, the multi-objective interaction and the occurrence of various special situations all bring difficulties to the target tracking. Current goal tracking issues include the following:

(1) Blocking Problems: Blocking is a common occurrence in multi-target tracking. The target can be obscured by a stationary object in the background, obscured by another target, or obscured or obscured by itself. The fuzzy process can be divided into two stages: one is that the target enters the occlusion process, and the target information is lost more and more during this process; the other is to evade the target and the target gradually shelters and the target information is gradually restored. The blocking performance of the target information is gradually lost. The key to tracking the algorithm is to search for enough target information to determine the target. Therefore, occlusion can cause great difficulty in the reliability of target tracking, which may lead to unstable tracking

or even target loss. One of the difficulties that has long been a multi-goal tracking problem is that most systems are currently unable to handle the more serious occlusion issues and cannot provide a standard for determining when to stop and when to start tracking again, and at a loss, without corresponding The goal of the recovery is based on the goal of the guide.

(2) The complexity of the background: The complexity and level of stability of the scene in which the target is located can affect the effectiveness of the target tracking. Disturbances in the background mainly include changes in light brightness, changes in background color, changes in background objects, and the presence of objects with similar target features in the background, which may increase target tracking and may cause the tracking to converge to interfering locations; shadowing issues, Shadows belong to the non-moving target area, but unlike the background color, it is difficult to detect the moving target.

(3) Differences between target appearance changes and different target appearances Target appearance characteristics include information such as target shape and texture. For non-rigid targets, the loss of target information makes it easy to track faults due to changes in the target's ratio and shape, as well as the uncertainty of the target's motion (maneuvering target). In surveillance video, the appearance of the target is often very similar, how to choose the appropriate characteristics to better distinguish between different target appearances, in order to achieve accurate data association.

(4) Real-time requirements: video images contain a lot of data, in order to ensure real-time target tracking requirements, you must select a small amount of calculation algorithms, but tracking target tracking is another important perfor-

mance tracking accuracy assurance is often complicated calculation and processing based on two Conflict between people. A good moving target tracking system must take into account the balance of these two performance indicators.

2.4 Mainstream tracking algorithms

At present, there are many target tracking algorithms and systems in the field of computer vision. Due to the different classification criteria, target tracking algorithms are generally divided into pixel-based methods, frequency-based methods and feature-based methods. In the detection of moving objects, all the images in the video sequence are directly scanned for a long time. Therefore, in each of the pretest images, the possible movement of the target in the target area is first extracted to detect the accuracy.

To track the moving target, we need to detect the moving target. An adaptable moving target tracking system should be able to adapt to a variety of changing scenarios, but we know that this is quite difficult in real applications. At present, there are more methods to detect moving objects: inter-frame difference method, and optical flow method.

(1) Inter-frame difference method/14/

Inter-frame difference method is a method of marking a moving object by performing a differential operation on two adjacent frames in the video.

The basic idea is based on the correlation between adjacent video frames in the video sequence, the difference between two or three frames to operate to develop test standards. In general, the frame difference method subtracts image blocks in two adjacent frames from pixel to pixel. When the background information of the target does not change much, the difference between the two images is less than the threshold we give, and we think the goal is still that the difference between the two images is larger than our given threshold, and we think it contains the moving target and then mark it down and then determine where to move.

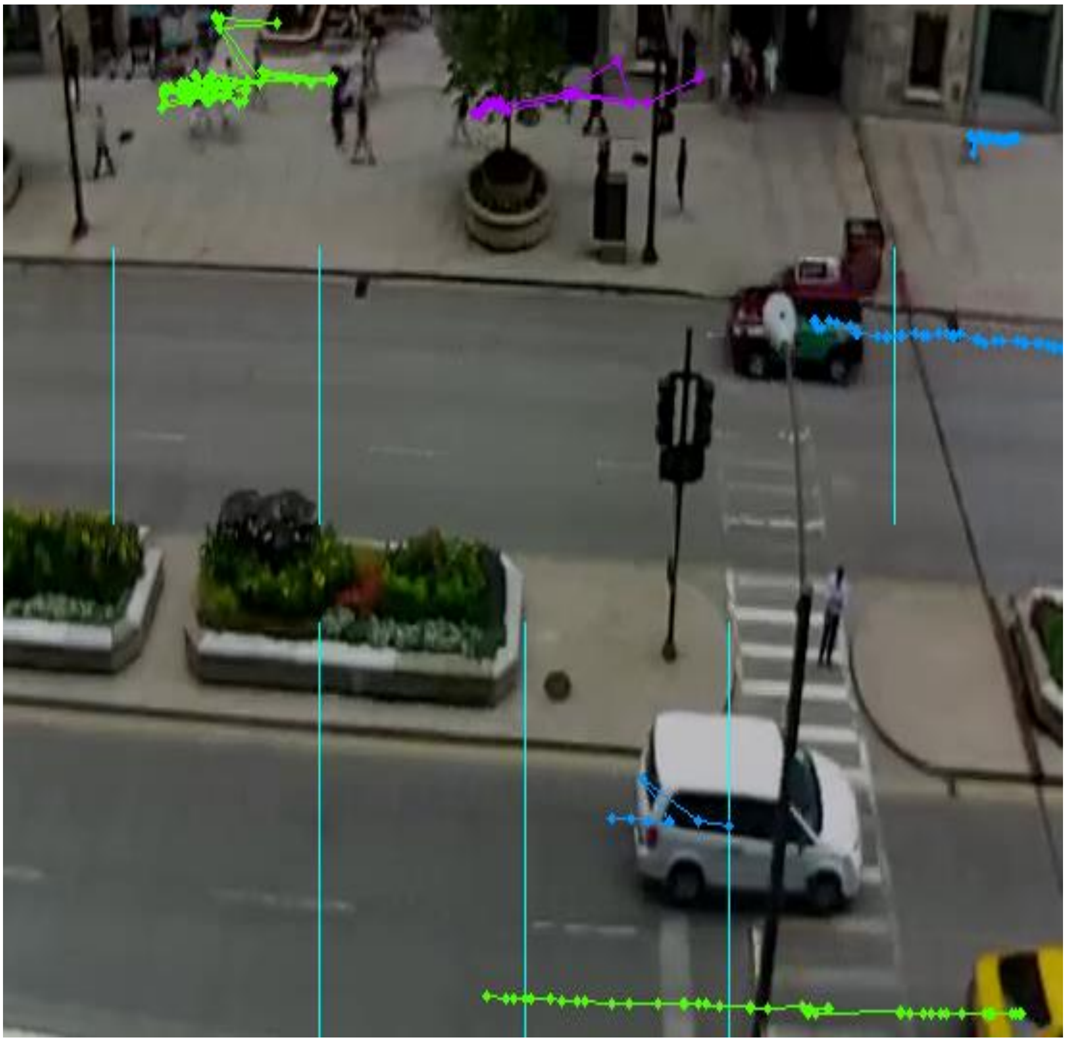


Figure 3. Inter-frame difference method (1)

The advantage of the inter-frame difference method is that the algorithm is easy to implement and the programming complexity is low. It is not sensitive to scene changes such as light and can adapt to various dynamic environments with good stability. The disadvantage is that you cannot extract a complete region of the object, the outline of the border is relatively rough, often larger than the actual object. For fast moving objects, it can even be detected as two different moving objects. For slow moving objects, no objects can be detected when the object almost completely overlaps two frames before and after.

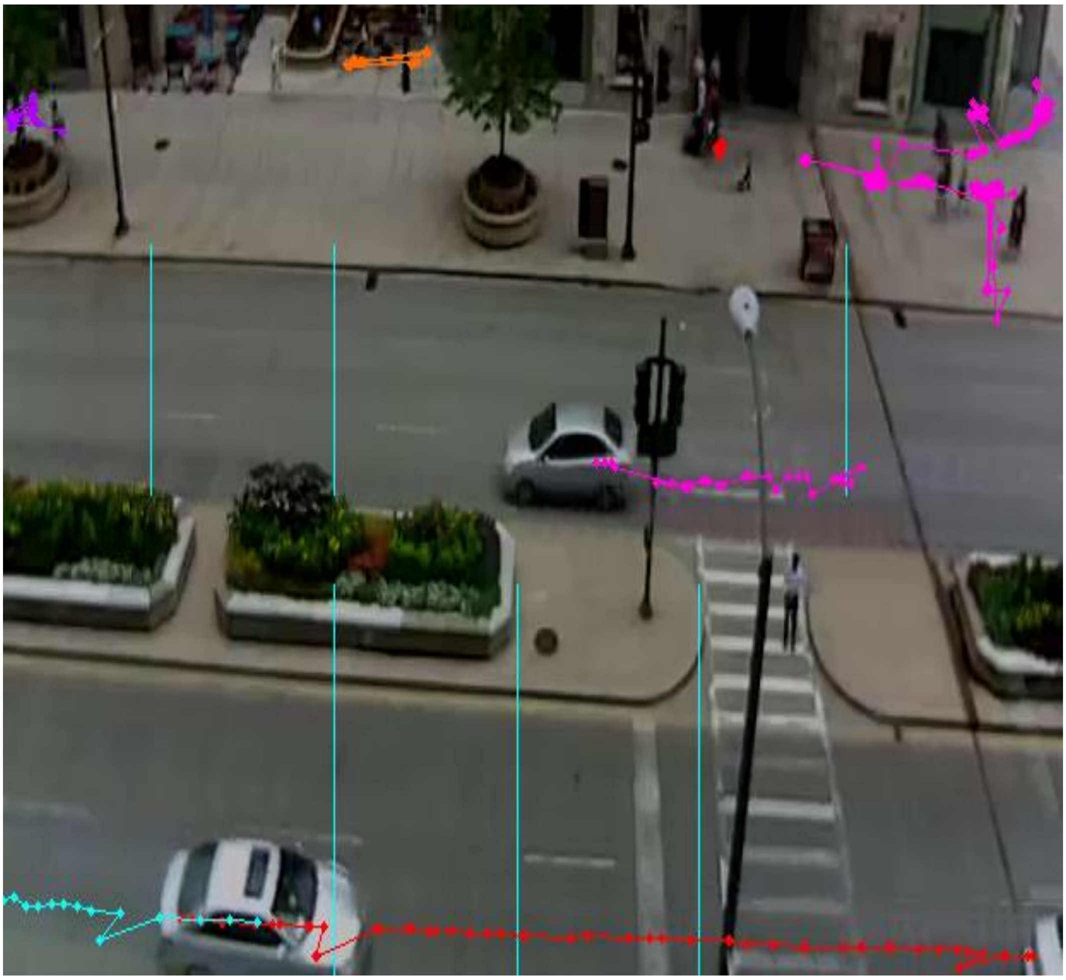


Figure 4. Inter-frame difference method (2)

(2) Optical flow method/15/

Schunck first calculated the optical flow in 1981. Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second. Consider the image below. It shows a ball moving in 5 consecutive frames. The arrow shows its displacement vector. Optical flow has many applications in areas like: Structure from Motion, Video Compression, and Video Stabilization.

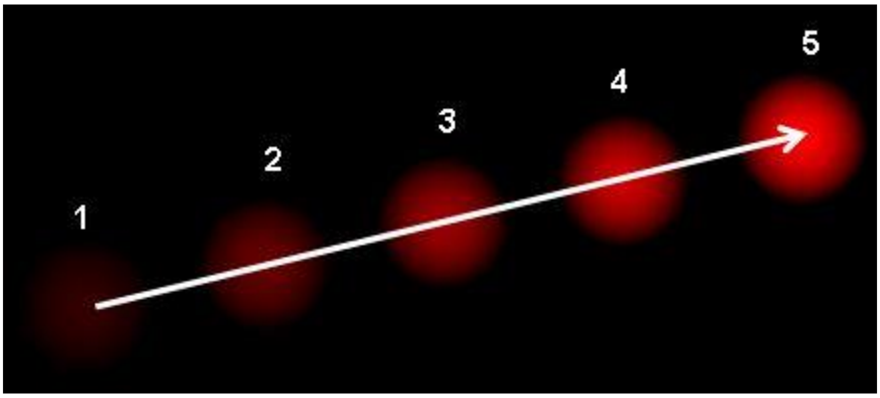


Figure 5. Optical flow method (1)

The principle of optical flow method for target detection: Give each pixel in the image a velocity vector, thus forming a motion vector field. At a specific time, the point on the image corresponds to the point on the three-dimensional object, and the correspondence can be calculated through projection. According to the characteristics of the speed vector of each pixel, the image can be dynamically analyzed. If there is no moving target in the image, the optical flow vector continuously varies over the entire image area. When there are moving objects in the image, there is relative motion between the target and the background. The speed vector formed by the moving object must be different from the speed vector of the background so that the position of the moving object can be calculated. Need to be reminded that the use of optical flow method for the detection of moving objects, a large amount of computation, cannot guarantee real-time and practicality.

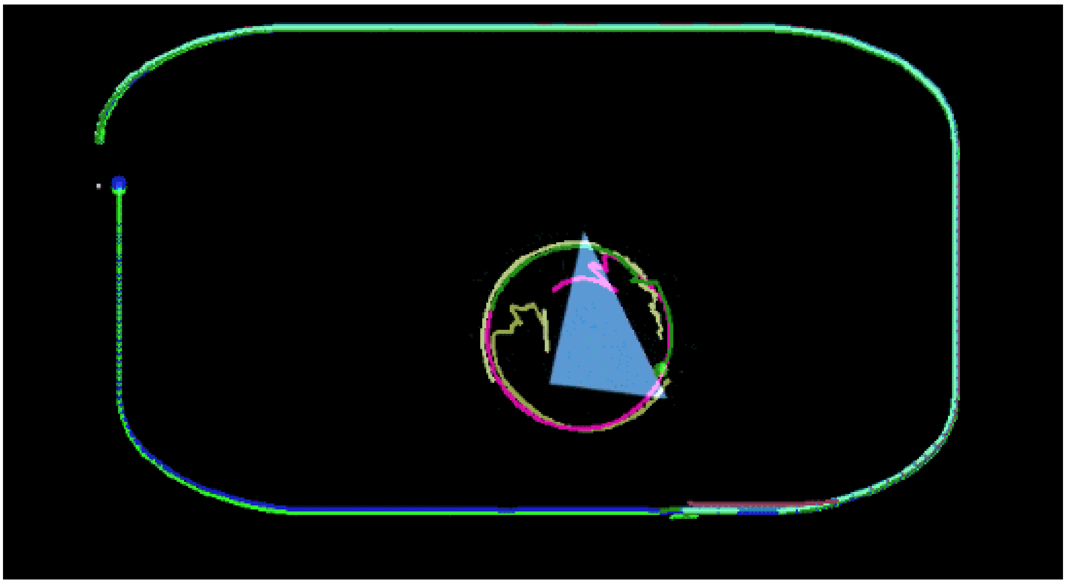


Figure 6. Optical flow method (2)

Optical flow method for tracking the principle:

1. Processing a continuous sequence of video frames;

2. For each video sequence, a certain target detection method is used to detect possible foreground targets. If there is a foreground object in a certain frame, find its representative key feature points.

3. For any two adjacent video frames afterwards, finding the best position of the key feature point in the previous frame in the current frame to find the position coordinate of the foreground object in the current frame.. So iterative, we can achieve the goal of tracking;

3. THE CORRELATION FILTER TRACKING ALGORITHMS

3.1 Introduction

Correlation is used to describe the relationship between two factors. The correlation is divided into cross-correlation (correlation, the relationship between the two signals) and auto-correlation (autocorrelation, itself in different frequency domain correlation). In 2010, David S.Bolme/16/ first used the filter in the tracking field in the article "Tracking Objects Using Adaptive Correlation Filters for Visual Objects."

3.2 Fundamental of Theory

Assuming there are two signals f and g, the correlation of the two signals is:

$$(\mathrm{f} \otimes \mathbf{g})(\tau) = \int_{\infty}^{-\infty} f^{*}(t)\mathrm{g}(\mathrm{t}+\tau)\mathrm{dt} \qquad (1)$$

$$(\mathrm{f} \otimes \mathbf{g})(\mathrm{n}) = \sum_{-\infty}^{+\infty} f * [m]\mathrm{g}(\mathrm{m}+\mathrm{n}) \qquad (2)$$

Where f * denotes the complex conjugate of f. The intuitive explanation for the correlation is to measure the similarity of two functions over time.

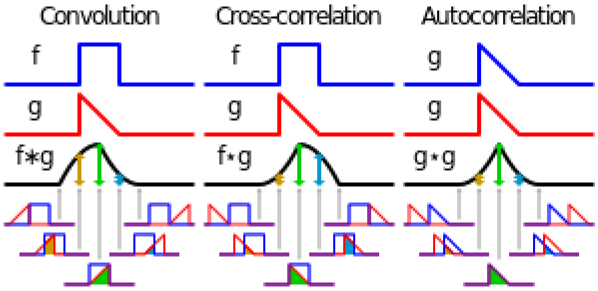


Figure 7. Basic idea of correlation filter

The simplest idea about applying and tracking correlation filters is that the more similar the two signals, the higher the correlation. Tracking is the goal of finding and tracking the maximum response to the project.

The filter proposed by the author is called the minimum output square sum error filter (MOSSE) (error square sum filter). According to the previous idea, we need to find a filter to maximize the response to the target, the following formula:

$$G = F \otimes H \tag{3}$$

Where g is the response output, f is the input image, and h is the filter template.

Simply determine the filter template h to get the response output. The above calculation is done on the convolution calculation, which is computationally expensive on a computer. Therefore, the author of the above formula for Fast Fourier Transform (FFT), making the FFT operation after the convolution into a point multiply, greatly reducing the amount of computation. The above formula becomes the following form:

$$F(g) = F(f \otimes g) = F(f) \cdot F(h)^{*} \tag{4}$$

For briefly express

$$G = F \cdot H^* \tag{5}$$

So,next step is to find H^*

$$H^* = \frac{G}{F} \tag{6}$$

However, in the actual tracking process, we need to consider the emergence of the target transform and other factors, so we need to use m target images as a reference to improve the robustness of the filtering template. Therefore, the authors propose a MOSSE model with the following formula:

$$\min H^* = \sum_{i=1}^{m} |H^* F_I - G_I|^2 \tag{7}$$

3.3 Framework of the Correlation Filter tracking algorithms

For the first frame input, the given region to be tracked is extracted feature, and then training to get the correlation filter.

For each subsequent frame, the previously predicted regions are cropped (the fast motion of the object is not good because the previous frame region is correlated), and then feature extraction is performed on the FFT transform using the cosine window function and then multiplied to the correlation filter. After the result is an FFT, the area with the largest response point is to track the new position of the target and then train and update with the new position area to obtain a new correlation filter for subsequent prediction.

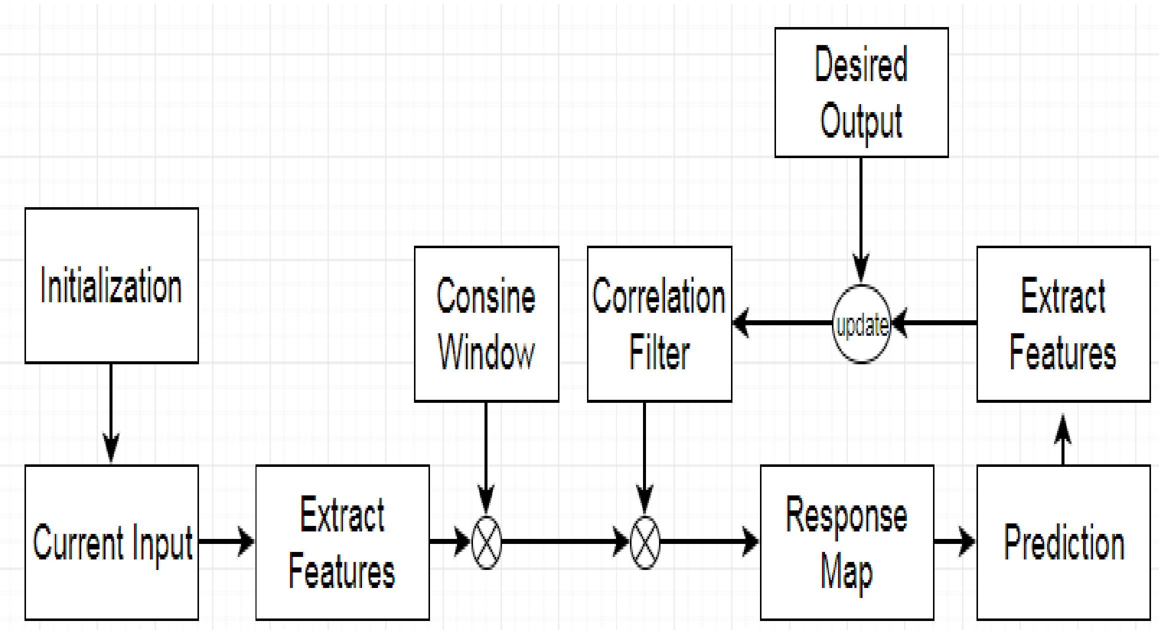


Figure 8. Flowchart of correlation filter tracking

3.4 Summary of related algorithms

In International Conference on Computer Vision and Pattern Recognition of 2010, David S.Bolme in the article "visual object tracking using adaptive correlation filters" for the first time used the relevant filtering in the field of tracking, and based on the article, many improved algorithms have appeared, the effect of tracking getting better and better. Here is a brief description of these algorithms.

1. Minimum Output Sum of Squared Error Filter (MOSSE) algorithm, first use correlation filtering for object tracking, and it uses gray-scale features up to 669fps, beyond the rest algorithms for general accuracy. This article also mentions that Peak Side Lobe Ratio (PSR) is used to determine if a target is occluded or tracked. In the latter algorithm, PSR is also used to determine blocking or tracking failures. There is also an improved PSR algorithm.

2. Exploiting the Circulant Structure of Tracking-by-detection with Kernels (CSK) /17/introduced the notion of a cyclic matrix and a core based on MOSSE, which is a prototype of KCF but uses gray scale features at speeds of 320 fps and greatly improved MOSSE over MOSSE, The solution to the problem of sample redundancy caused by sparse sampling is used in the algorithm, and the circulate matrix and kernel techniques shine brightly in the field of target tracking of relative filtering.

3. Kernel Correlation Filter (KCF) /18/can be said to be a perfect CSK. The thesis has a solid theoretical foundation and a complete formula derivation of Ridge Regression, Cyclic Matrix, Nuclear Techniques, and Rapid Detection. The paper looks perfect. The HOG Features, the code is used fHOG, the single-channel is converted to multi-channel, the kernel has three Gaussian kernels, linear kernels and polynomial kernels, the highest accuracy Gaussian kernel, linear kernel slightly lower than the Gaussian kernel, but far away from the speed Higher than Gaussian nucleus.

4. Color Name (CN)/19/ is based on the CSK. It uses the color features. The color description is different in different languages. The 11 words described in color in English are the closest to human vision, in the paper, we also prove that the color space of CN is better than RGB, HSV and so on. In order to improve the computing speed, PCA is also used to reduce the 10-dimensional features to 2 dimensions and improve the model tracking Program.

5. Discriminative Scale Space Tracker (DSST)/20/ is based on the MOSSE proposed mainly for the scale change problem, the article uses the scale-dependent filter, in principle and KCF similar to the relevant filter through the relevant point to find the largest point corresponding to the image as the target, DSST, uses 33 different scales, relatively slow in time, but the accuracy of the scale estimates is relatively high.

6. Long-term Correlation Tracking (LCT)/21/, based on DSST, added a DSST-based confidence filter that solves the problem of long-term goal tracking using stochastic fern classifiers in top-level domain (TLDs) and therefore involves the use of PSR to determine whether a goal is ambiguous and blocking occlusion target random fern classifier, compared with the former algorithm greatly improved the accuracy, but because each frame training random fern classifier, the speed is too slow.

7. Learning Spatially Regularized Correlation Filters for Visual Tracking (SRDCF) /22/is a solution based on the Discriminative Correlation Filter (DCF) boundary effect. Joined the space punishment project, made a breakthrough effect. However, the addition of spatial penalties undermines the closed solution of the ridge regression equation, which can only be solved by the Gauss-Seidel iterative method, and the calculation speed is very slow.

After this, the deep learning tracking algorithms that appeared and are widely used, the accuracy made a great progress. But the deep learning tracking algorithms are not the main topic of the article, so not too much from this field will be mentioned.

4. TARGET TRACKING FRAMEWORK WITH KCF

In this section, we mainly study the KCF (Kernel Correlation Filter) algorithm. This chapter first discusses the theoretical basis of the algorithm, and then describes in detail the working process of KCF framework.

4.1 Kernel Correlation Filter

KCF is a discriminant-based tracking method. In this method, the target detector is trained during the tracking process. The target detector is used to detect if the predicted position of the next frame is a target. Then the new test results are used to update the training set and update the target detector. In the training of the target detector, the target area is selected as a positive sample and the surrounding area of the target is a negative sample. Therefore, the closer to the target area, the more likely it is to be a positive sample.

According to the details of publication named "High-speed tracking with kernelized correlation filters." /18/ the algorithm will be divided into several parts to introduce.

4.2 Problem Formulation

The algorithm abstracts the tracking problem as a solution to the regression model, representing the input of the target image as z, the weight w, and the output as $f(\omega) = \omega^T x$. The aim is to get the the solution that minimizes the smallest mean square error of the output f (x_i) of the classifier model and the expected regression y_i on sample x_i:

$$\min w \sum (f(x_i) - y_i)^2 + \lambda ||\omega||^2 \tag{8}$$

$$\omega = (X^h X + \lambda I)^{-1} X^H y \tag{9}$$

λ is a regularization of excessive regularization parameter. Where $X^H = (X^*)^T$ denotes the complex conjugate, T denotes transpose, each row of X is x_I.

4.3 KCF Algorithm

Since these equations are too complicated to complete, the author of the publication cleverly used the cyclic matrix to solve these problems.

According to the circulant matrix, it can be diagonalized by the discrete Fourier Matrix, so that the inverse of the matrix can be transformed into the inverse characteristic of the eigenvalue. Formula (9) can be transformed into the frequency domain computation and the computational speed of the Discrete Fourier Transform (DFT), which is inverted into the airspace to obtain the response Y_{max}.

According to formula (9):

$$x^H = (X^*)^T = \left(F^* diag(\hat{x}^*) F^{H^*}\right)^T$$

$$= F diag(\hat{x}^*) F^H \tag{10}$$

And then expand formula (9)

$$\omega = \left(X^h X + \lambda \mathrm{I}\right)^{-1} X^H y$$

$$=(F diag(\hat{x}^* \cdot \hat{x} + \lambda) F^H)^{-1} X^H y$$

$$=\left(F diag\left(\frac{(\hat{x}^*)}{\hat{x}^* \cdot \hat{x} + \lambda}\right) F^H \right)\mathrm{y} \tag{11}$$

Use the circulant matrix property:

$$\omega = C(F^{-1}(\frac{(\hat{x}^*)}{\hat{x}^* \cdot \hat{x} + \lambda}))y \tag{12}$$

$$F(\omega) = F(C(x)y) = \frac{(\hat{x} \cdot \hat{y})}{\hat{x}^* \cdot \hat{x} + \lambda} \tag{13}$$

Accordingly, the linear regression coefficient ω can be obtained by the above inverse Fourier transform.

The above is only a solution to the linear problem. By introducing a kernel that extends the problem to non-linear space, the authors extend the solution to non-linear problems. There is no special explanation here.

Based on the derivation of the previous chapter, it’s easy to get the flow of the algorithm:

Tracking

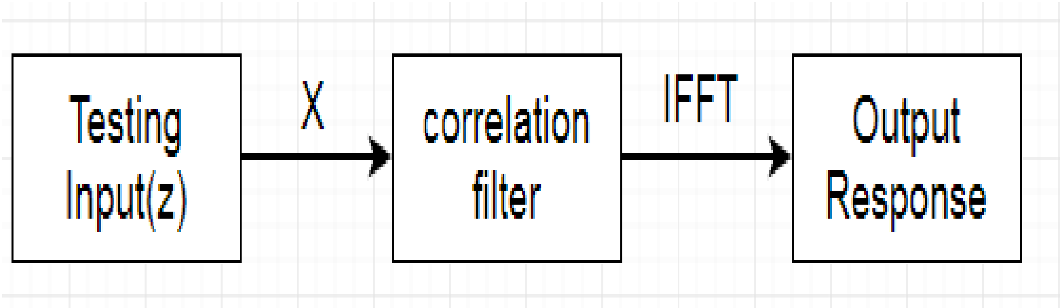


Figure 9. KCF Tracking flow

Training

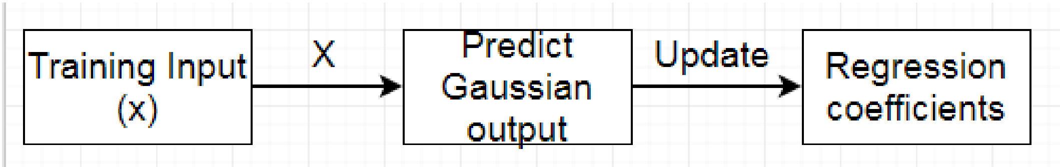
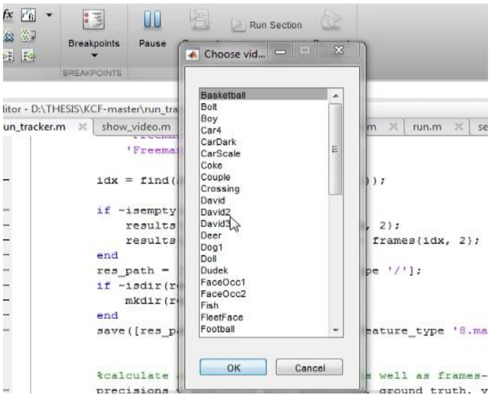
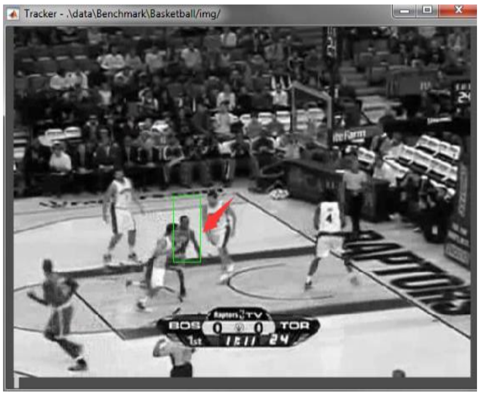


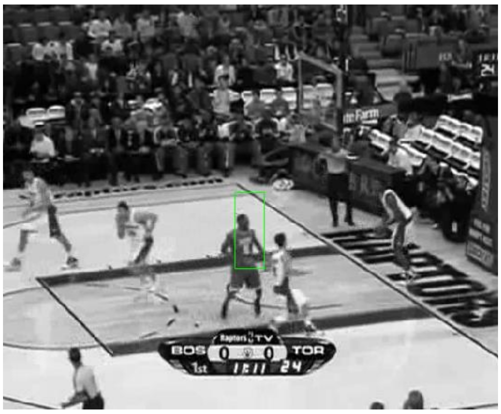
Figure 10. KCF Training flow

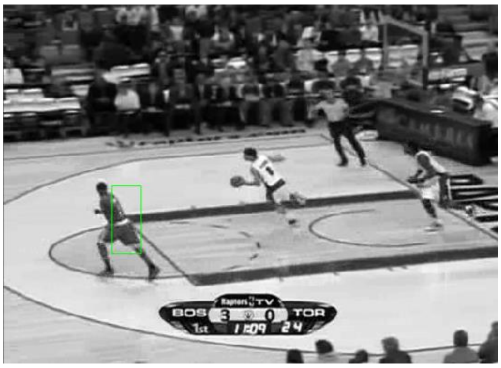
4.4 Results

Here are the brief results which show the effect of the algorithm. The algorithm tested by Matlab and the test bench functions also created by MATLAB. Here are some classical examples to show the results.

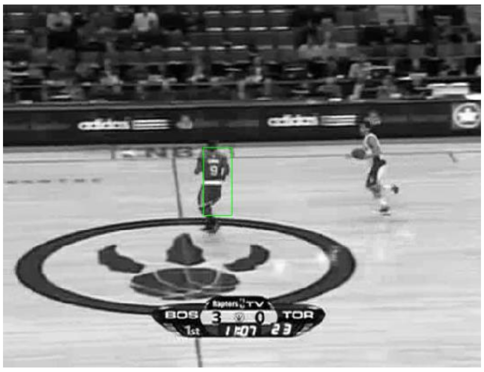














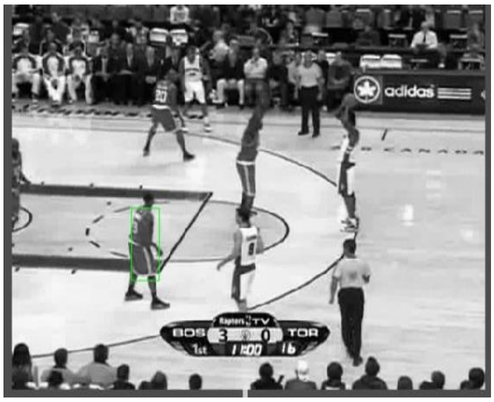



Figure 11. KCF Test Results (Basketball-dataset)

Figure 12. KCF Test Results (Coke-dataset)

5. TARGET TRACKING FRAMEWORK WITH DSST

In this section, we mainly study the DSST (Discriminative Scale Space Tracker) algorithm/20/. This chapter first discusses the theoretical basis of the algorithm, and then describes in detail the working process of DSST framework.

5.1 Discriminative Scale Space Tracker

Robust scale estimation is a challenging issue in visual object tracking. Most existing methods do not handle large-scale changes in complex image sequences. The Linköping University Visual Laboratory proposed a novel method to estimate the robustness scale in the frame by frame. The proposed method works by learning discriminate Correlation filters based on a scale pyramid representation. Utilized separate filters for translation and scale estimation, and showed significantly improved performance over a complete scale search.

DSST (Accurate Scale Estimation for Robust Visual Tracking) took the first place on VOT 2014 with its simple algorithm and excellent performance, including the KCF algorithm, both which are filter-based algorithms. This algorithm is based on MOSSE improvements, and the highlight are the addition of scaling. The MOSSE algorithm has been described in the previous section, so here not repeating anymore.

The algorithm design two identical filters respectively, to achieve the target tracking and scaling, as a translation filter and scaling filter. /23/The former locates the target of the current frame, and the latter estimates the target ratio of the current frame. The two filters are independent to each other, so different types of features and features calculations can be selected for training and testing. The Linköping University Visual Laboratory official website pointed out that the highlight of the algorithm is a standard method that can be ported to any algorithms.

5.2 Problem Formulation

The DSST algorithm designs the input signal f as a d dimensional eigenvector and constructs the optimal correlation filter h by establishing a minimized cost function as follows:

$$\varepsilon = ||\sum_{l=1}^{d} h^l * f^l - g\,||^2 + \lambda \sum_{l=1}^{d} ||h^l||^2 \qquad (14)$$

5.3 DSST Algorithm

At the moment, l represents a certain dimension of the feature, and λ is a regular coefficient whose role is to eliminate the effect of the zero-frequency component in the f spectrum and to avoid the numerator of the above formula to be zero, as follows:

$$H^l = \frac{\bar{G}F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda} = \frac{A_t^l}{B_t} \qquad (15)$$

Since each pixel in the patch requires a linear equation to solve for the d×d dimension, computation is time-consuming. In order to get a robust approximation, update the numerator A_t^l and denominator B_t respectively in the above formula:

$$A_t^l = (1-\eta)A_{t-1}^l + \eta \overline{G_t} F_t^l \qquad (16)$$

The η is the learning rate. And in the new frame, the target position can be obtained by solving the maximum correlation filter response:

$$y = F^{-1}\left\{\frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B+\lambda}\right\} \tag{17}$$

The highlight of this algorithm is a scale search and target estimation method based on one-dimensional independent correlation filters. The specific operation is as follows: In a new frame, a two-dimensional position-correlation filter is first used to determine a target's new candidate position, and then a one-dimensional scale-correlation filter is used to obtain the current center position as the center point to obtain different proportions of candidate patches, the most suitable scale.

The Linköping University Visual Laboratory in its paper published in the algorithm flow is very clear and intuitive, direct reference here.

Algorithm 1 Proposed tracking approach: iteration at time step t.

Input:

Image I_t.
Previous target position $\boldsymbol{p}_{t-1}$ and scale s_{t-1}.
Translation model A_{t-1}^{trans}, B_{t-1}^{trans} and scale model A_{t-1}^{scale}, B_{t-1}^{scale}.

Output:

Estimated target position $\boldsymbol{p}_t$ and scale s_t.
Updated translation model A_t^{trans}, B_t^{trans} and scale model A_t^{scale}, B_t^{scale}.

Translation estimation:

1: Extract a translation sample z_{trans} from I_t at $\boldsymbol{p}_{t-1}$ and s_{t-1}.
2: Compute the translation correlation y_{trans} using z_{trans}, A_{t-1}^{trans} and B_{t-1}^{trans} in (6).
3: Set $\boldsymbol{p}_t$ to the target position that maximizes y_{trans}.

Scale estimation:

4: Extract a scale sample z_{scale} from I_t at $\boldsymbol{p}_t$ and s_{t-1}.
5: Compute the scale correlation y_{scale} using z_{scale}, A_{t-1}^{scale} and B_{t-1}^{scale} in (6).
6: Set s_t to the target scale that maximizes y_{scale}.

Model update:

7: Extract samples f_{trans} and f_{scale} from I_t at $\boldsymbol{p}_t$ and s_t.
8: Update the translation model A_t^{trans}, B_t^{trans} using (5).
9: Update the scale model A_t^{scale}, B_t^{scale} using (5).

Figure 13. The Algorithm Flow in Publication/20/

In this paper, the above graph formula (5) (6), corresponding to the formula in this article (14) (15).

5.4 Results

The test enviroment is the same as the previous test. Here are some classic examples to show the results. The scale changes can be seen clearly.

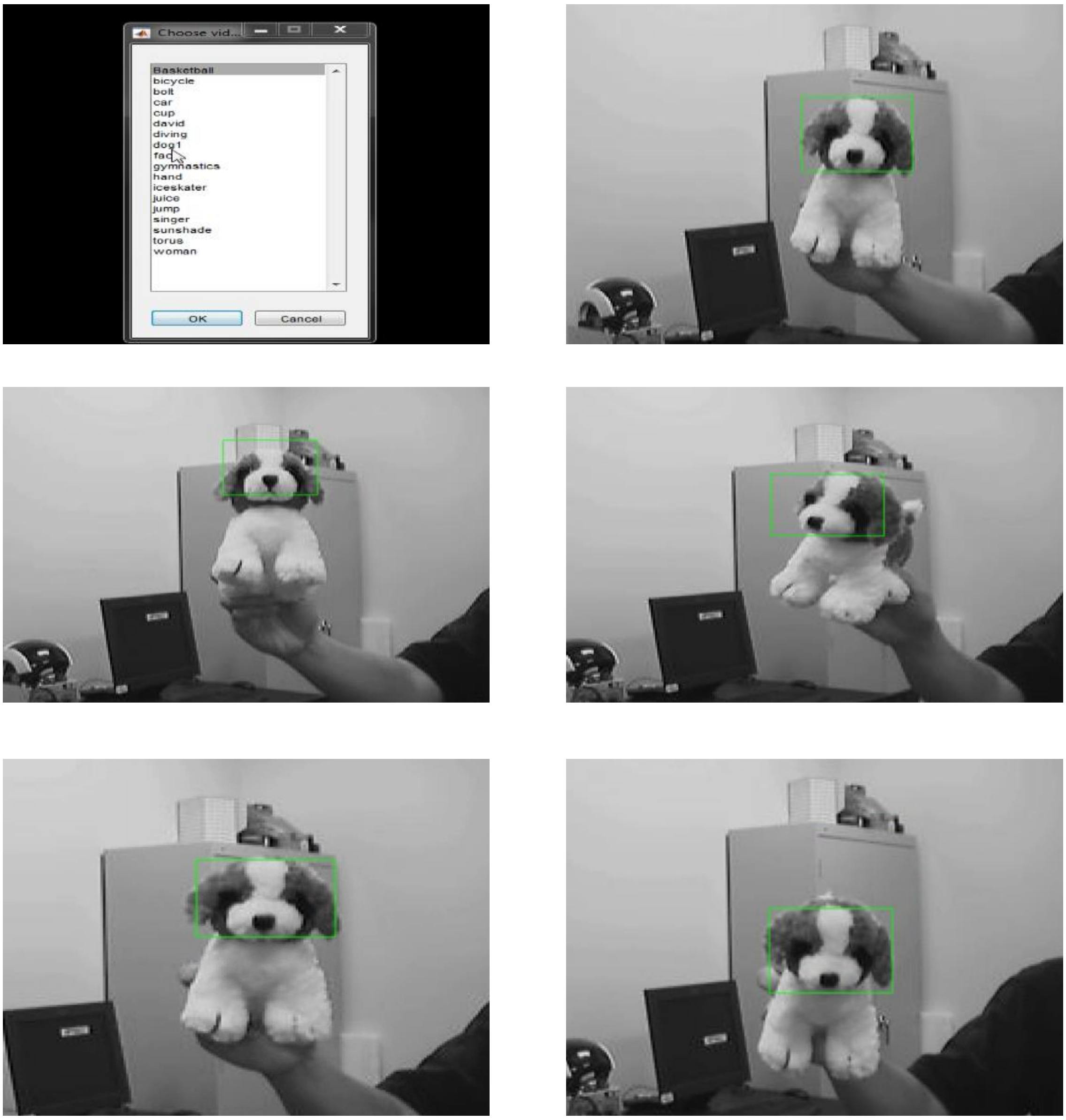


Figure 14. DSST Test Results (Normal)

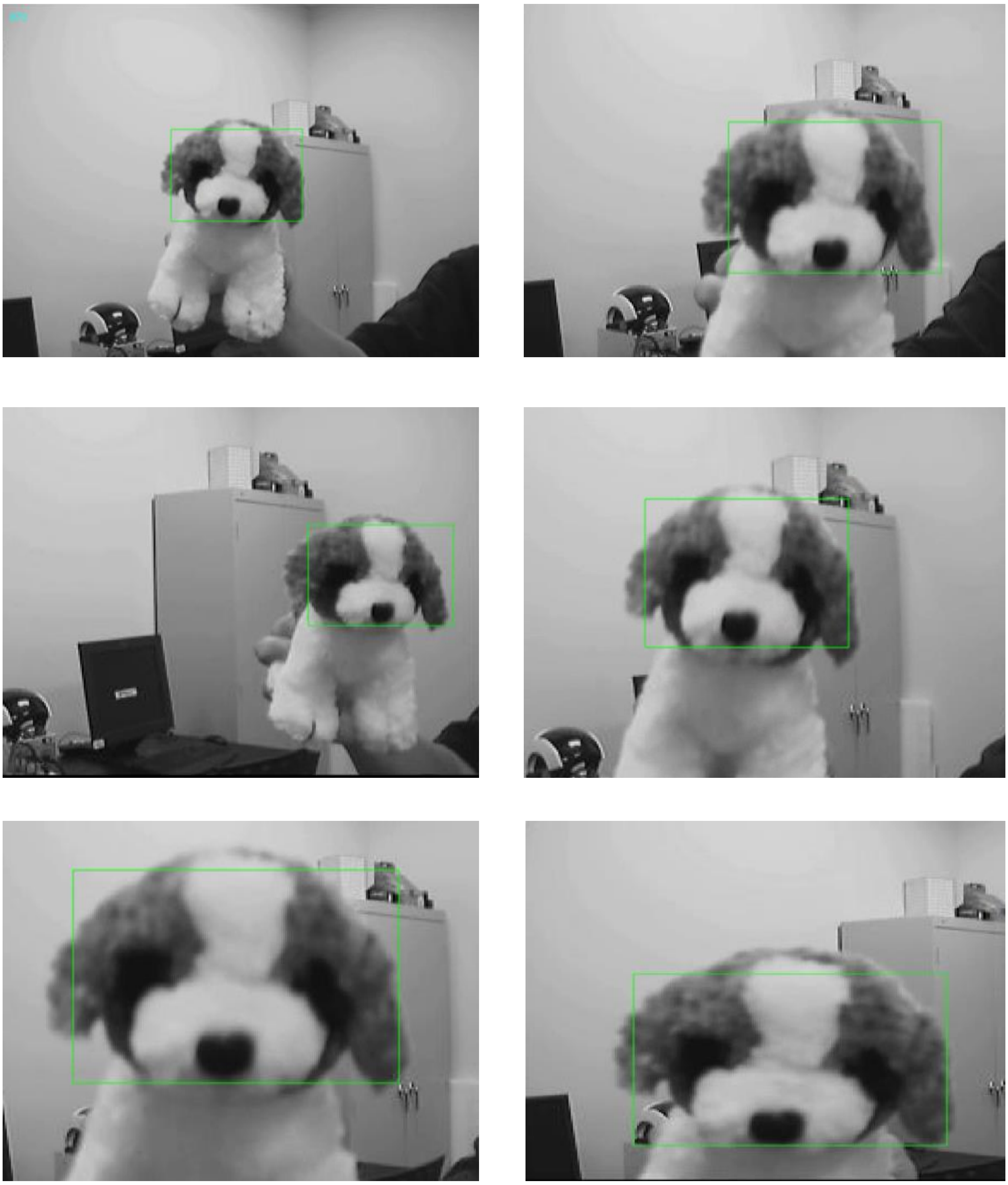


Figure 15. DSST Test Results (Scale Change)

6. COMPARING BETWEEN KCF AND DSST

6.1 Standard and Dataset

Visual tracking test, usually refers to a common single-target tracking, the first frame gives the location of the database location, which is manually marked, and then needs to track the tracking object in the subsequent frame, the following is the VOT tracking algorithm requirements/24/:

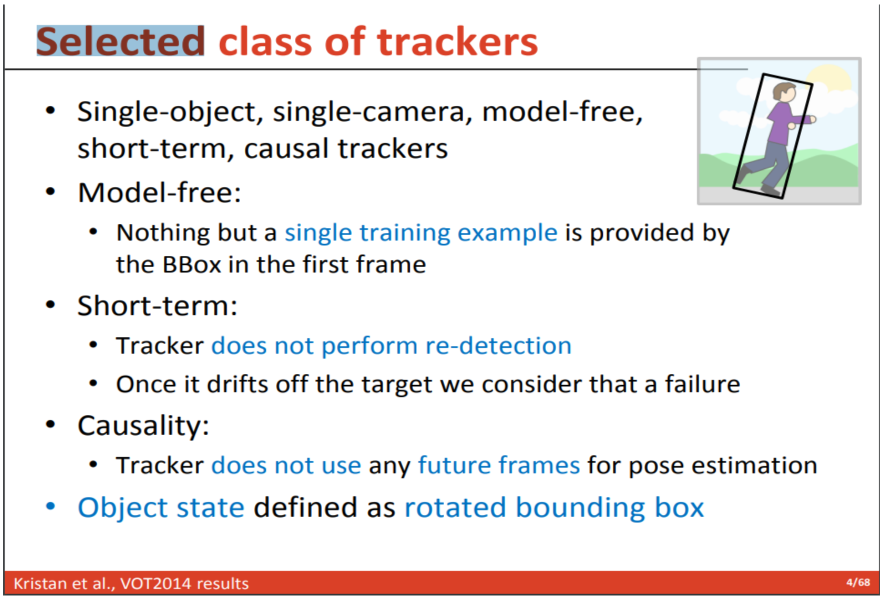


Figure 16. Selected class of trackers/25/

In the visual tracking test environment, the most commonly used is the OTB database, but also the VOT competition database. People tend to use the VOT database because the fine sequence annotation and good metric. Therefore, tests of this paper selects part of the VOT database for speed comparison and the accuracy of the two algorithms./26/

6.2 Result of Precision and Speed

Results shows in the table below:

Data-Set	**KCF-algorithm**	**DSST-algorithm**
dog	Precision: 100%, FPS: 158.56	Precision: 100%, FPS: 40.5
bolt	Precision: 98.9%, FPS: 175.32	Precision: 94.3%, FPS: 50.1
david	Precision: 11.4%, FPS: 82.81	Precision: 99.6%, FPS: 8.79
face	Precision: 73.0%, FPS: 87.80	Precision: 100%, FPS: 9.16
singer	Precision: 81.5%, FPS: 74.59	Precision: 100%, FPS: 5.01

Table17. Comparing table between algorithms

The speed and precision of these algorithms can be seen from the above table. The KCF algorithm detection speed, but in some cases the detection accuracy is low. In contrast, The DSST algorithm consistently maintain high accuracy, but at a slower rate. The specific application needs to be decided according to the specific situation.

```
% if no overlap, set to zero
overlap_height(overlap_height < 0) = 0;
overlap_width(overlap_width < 0) = 0;

% remove NaN values (should not exist any)
valid_ind = ~isnan(overlap_height) & ~isnan(overlap_width);

% calculate area
overlap_area = overlap_height(valid_ind) .* overlap_width(valid_ind);
tracked_area = positions(valid_ind,3) .* positions(valid_ind,4);
ground_truth_area = ground_truth(valid_ind,3) .* ground_truth(valid_ind,4);

% calculate PASCAL overlaps
overlaps = overlap_area ./ (tracked_area + ground_truth_area - overlap_area);

% calculate PASCAL precision
PASCAL_precision = nnz(overlaps >= PASCAL_threshold) / numel(overlaps);
end
```

Figure 18. Performance measure function

7. THE KCF AND KALMAN FILTER-BASED TARGET TRACKING FRAMEWORK

According to current projects, these programs focus more on tracking speed. In other words, these projects do not require much accuracy with respect to scale changes and complex background tracking. Given this situation, developing the KCF algorithm rather than the DSST algorithm is a smarter alternative.

7.1 Proposed Framework

According to the principle and characteristics of the correlation filter tracking algorithm, when the algorithm tracks the moving speed of the target object fast, the tracking state is unstable, and even the drift of the target area leads to completely error. Based on this problem, the inclusion of motion compensated and predictive filters can effectively reduce or even eliminate this situation.

7.1.1 Kalman filter

The Kalman filter/27/, is known as a linear quadratic estimation (LQE). It is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone. By using Bayesian inference and estimating a joint probability distribution over the variables for each timeframe to work. The filter is named after Rudolf E. Kálmán, one of the primary developers of its theory.

The observed system/28/ needs to be described by equations, because some of the parameters here are used in the equations behind the filter. Set the system's state equation and observation equation is:

$$\mathrm{X(k) = A \cdot X(k-1) + B \cdot U(k) + w(k)} \tag{18}$$

$$\mathrm{Z(k) = H \cdot X(k) + y(k)} \tag{19}$$

Where $\mathrm{X(k)}$ represents the control volume, $\mathrm{w(k)}$ represents the process noise that complies with Gaussian distribution, $\mathrm{z(k)}$ represents the observed value of the system, and $\mathrm{y(k)}$ represents the Gaussian distribution measurement. The covariance of the noise, hereafter R, A, B, H, represents the system parameters.

The Kalman filter used in the practical application is a constantly updated process. Each time a new observation value is updated, two things are updated back and forth: "system state" (x) and "error covariance" (P). Since each calculation uses only the last result and the new measurement, such filtering occupies a small amount of computing resources.

Each iteration, by the prediction and correction of two parts. The prediction is to predict the system state and error covariance $\mathrm{X(k|k-1)}$ and $\mathrm{P(k|k-1)}$ at this time based on the results of the previous iteration $\mathrm{X(k|k-1)}$ and $\mathrm{P(k|k-1)}$:

$$\mathrm{X(k|k-1) = A \cdot X(k-1|k-1) + B \cdot U(k)} \tag{20}$$

$$\mathrm{P(k|k-1) = A \cdot P(k-1|k-1)} \cdot A^T + \mathrm{Q} \tag{21}$$

The Kalman gain K(k) is then calculated and used together with the actual measurement Z(k) this time to correct the system state $X(k|k-1)$ and the error co-variance $P(k|k-1)$ latest X(k|k) and P(k|k):

$$K(k) = P(k|k-1) \cdot H^T \cdot (H \cdot P(k|k-1) \cdot H^T + R)^{-1} \quad (22)$$

$$X(k|k) = X(k|k-1) + K(k) \cdot \big(z(k) - H \cdot x(k|k-1)\big) \quad (23)$$

$$P(k|k) = (I - K(k) \cdot H) \cdot P(k-1|k-1) \quad (24)$$

X(k|k) is the filter value, which will be used in the next time as $X(k-1|k-1)$ and $P(k-1|k-1)$ at the next iteration.

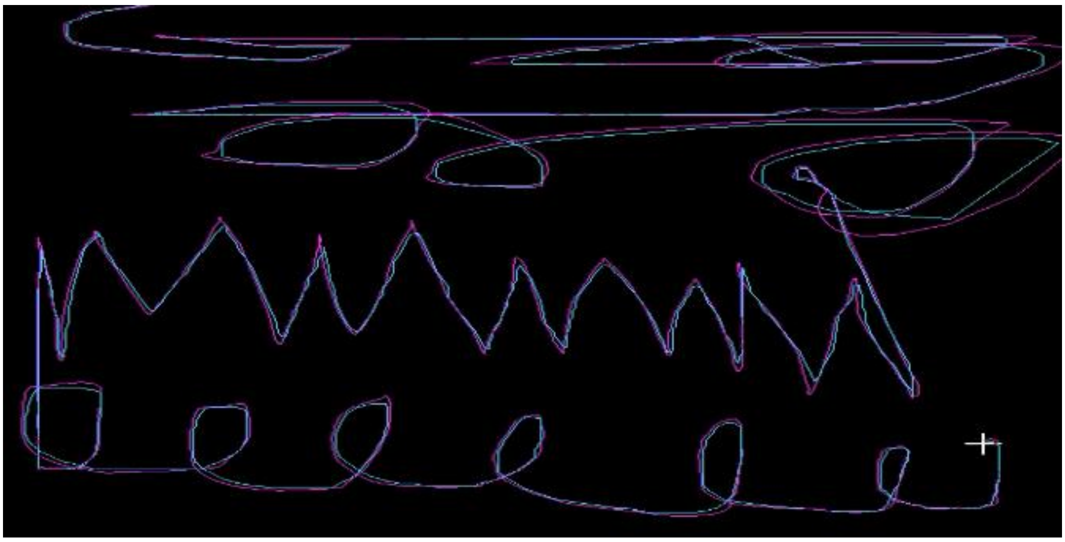


Figure 19. Kalman filter sample

7.2 Implementation

7.2.1 Test bench with Python

Python is a generic, advanced programming language that is generally clearer, direct, and code-oriented. Although it is not so efficient as other programming languages (such as C ++), it is easy to achieve your goal. The OpenCV API for KCF algorithms allows all C ++ APIs to be used and creates scripts that can be run remotely on a local computer or an embedded device. Based on this issue, Python is more suitable for learning and deep logic development

Using the tracking API provided by the latest version of OpenCV 3.3, the author successfully achieved the KCF algorithm and tested the tracking under various conditions. The test results are attached below.

First frame is the original image input signal, and the area of blue rectangle selection is a useful sample of positive samples that can be viewed as favoring the KCF algorithm. In the next sequence of video images, the tester took off the glasses, put on the glasses, took off the headphones, picked up the headphones, interfered with the target with similar color, and blocked the track with a large area. However, the KCF algorithm always categorizes the correct region, proving that the initial algorithm implementation is feasible.

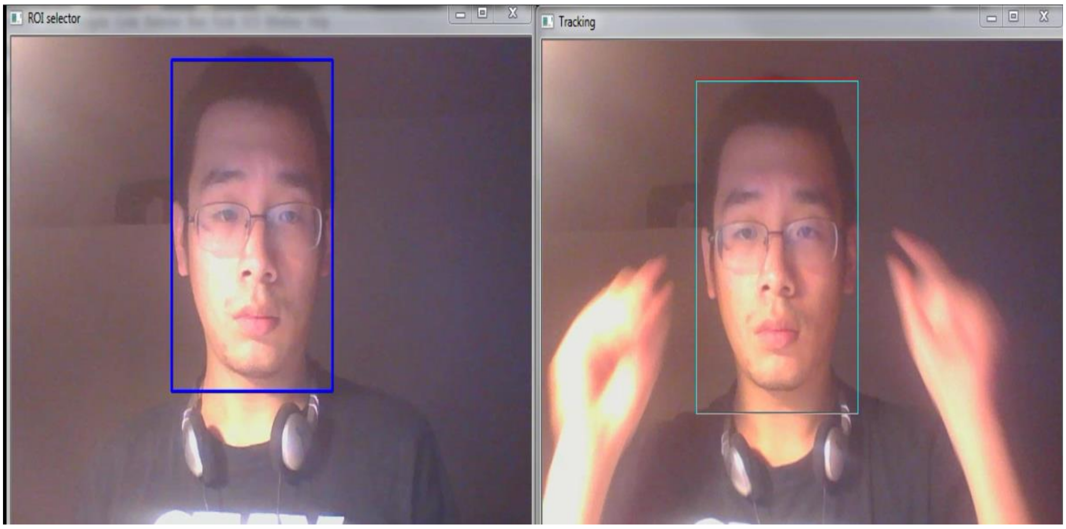


Figure 20. Target area interception

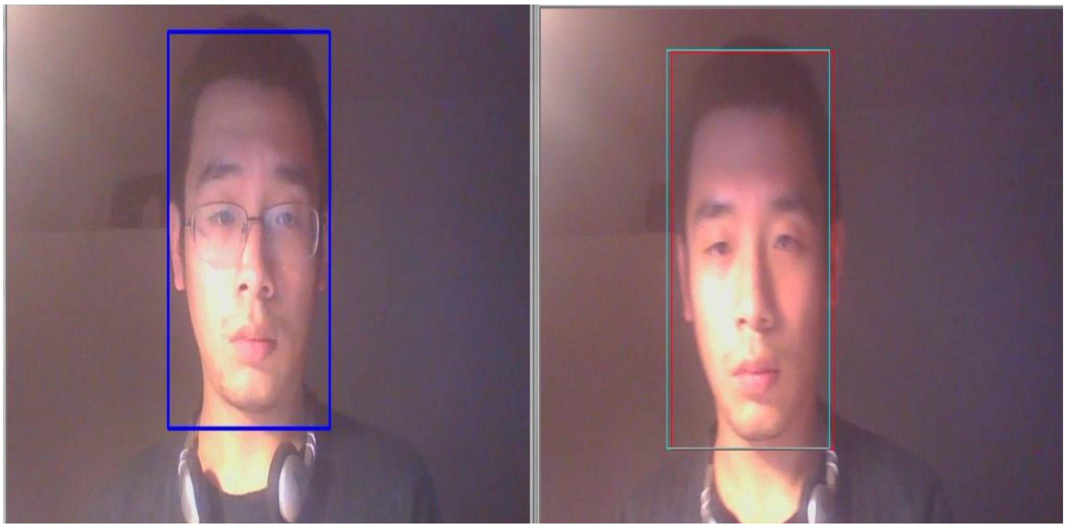


Figure 21. Tracking result of the object has small change

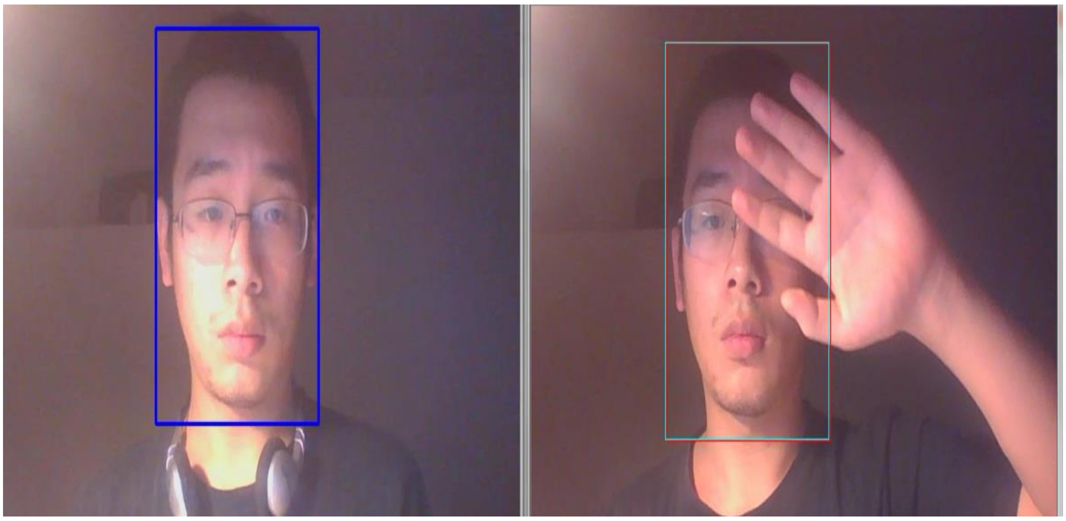


Figure 22.Tracking result of similar object interference

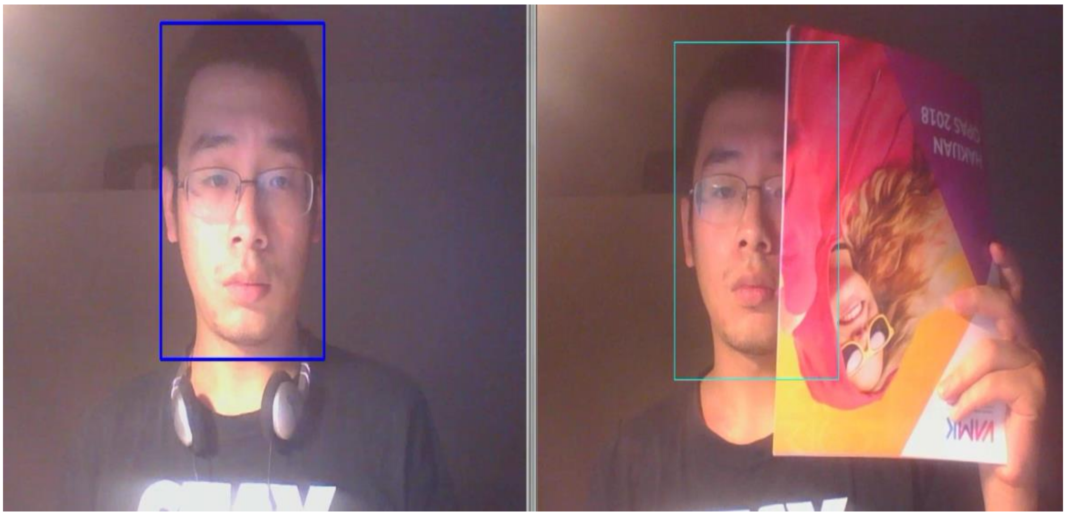


Figure 23.Tracking result of large area blocking interference

7.2.2 Test results with Kalman filter

As the problems mentioned before, Kalman filter has been added to the test platform has been built using python, and the program will use the VOT test dataset a relatively significant test sequence relative to Kalman filter for the effect. The test results are as follows (Green-with Kalman Filter, Red- without Kalman Filter):

Figure 24.Select the target and target area

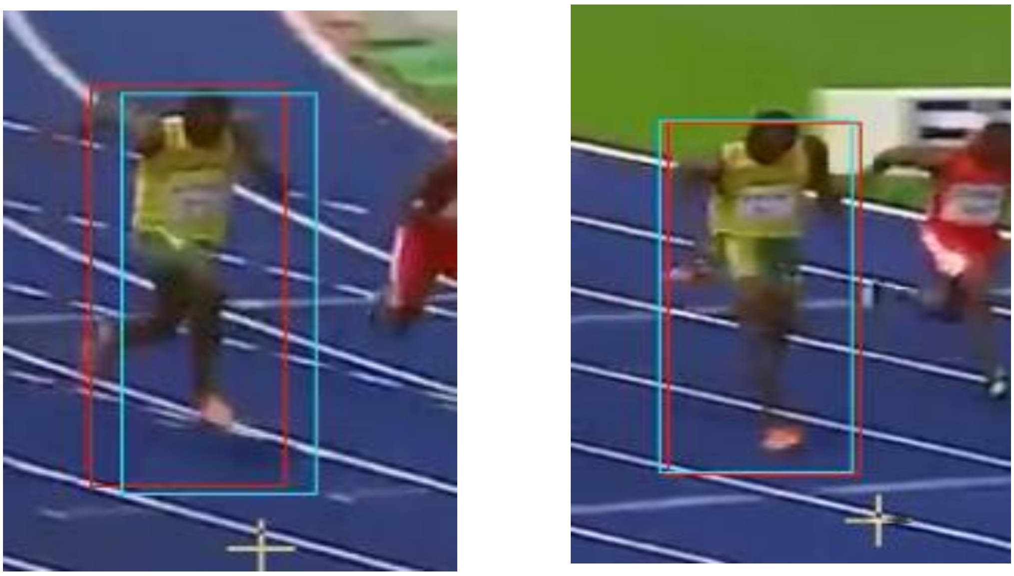


Figure 25. Tracking result of simple background

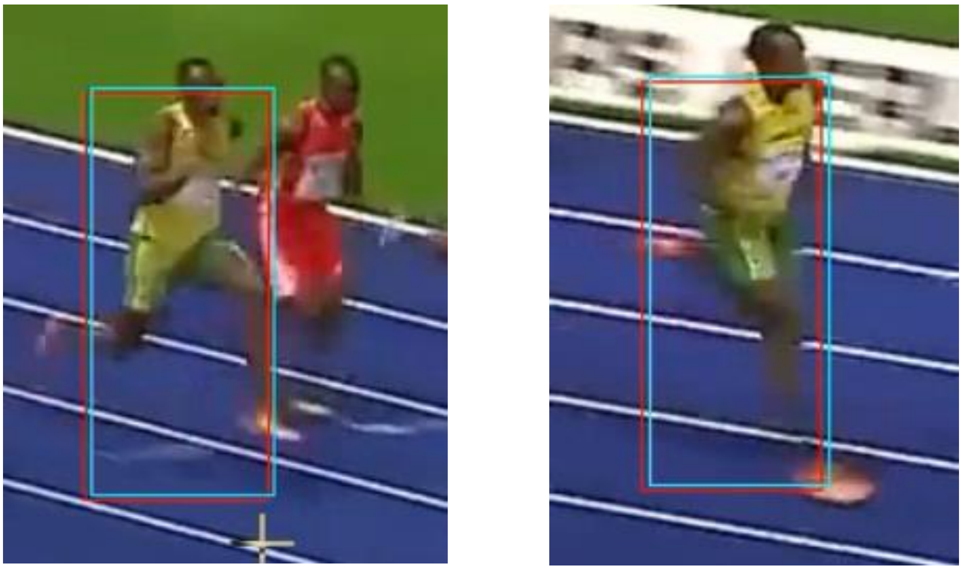


Figure 26. Tracking result of body deformation

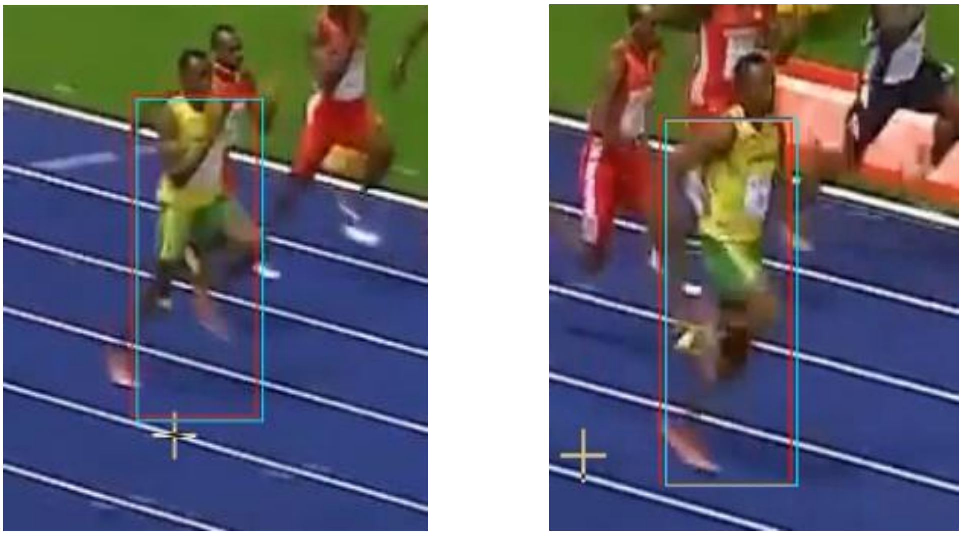


Figure 27. Tracking result of similar target interference

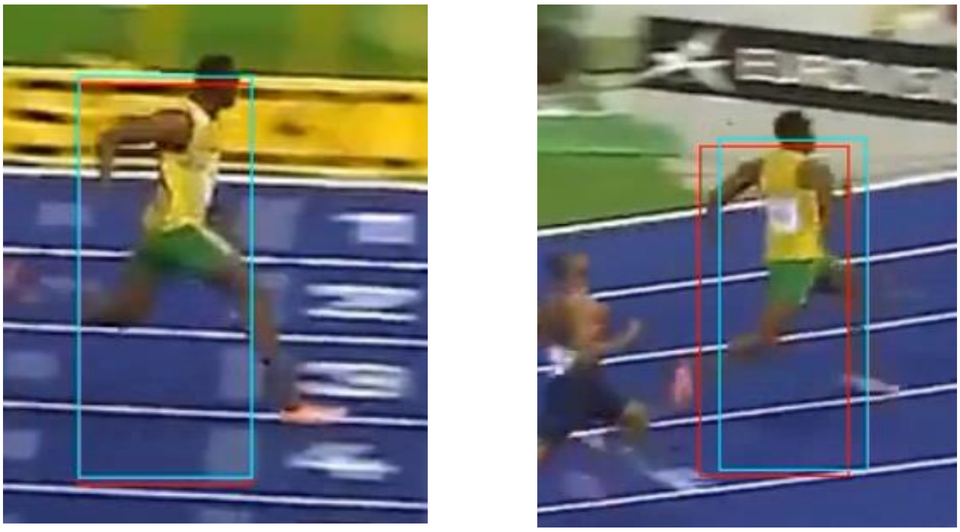


Figure 28. Tracking result of target area changes

It can be seen from the test results that the test objects in the entire test sequence move quickly throughout the test sequence, with blurred motion and unstable shape. A number of background changes occurred during the test, and the samples during the early and late tests changed significantly. However, the target is always in the tracking range, and based on the Kalman filter, there is a clear tendency to predict the movement. Therefore, it can be concluded that Kalman filter can make effect in the program.

8. SUMMARY AND PROSPECT

8.1 Prospect

In this paper, we research and analyze the current situation of video target tracking in related fields, reference some of the research results, and analyze the related applications of correlation filter target tracking algorithms. By comparing, summarizing, summarizing and experiment analyzing, some problems in target tracking are solved and some targets have been achieved. However, we must understand that the long-term goal tracking technology is a very challenging research in the field of computer vision and many problems remain to be solved. According to the research direction of this article, we can summarize the following aspects as the direction of further research in the future.

8.1.1 Improving aspects of the correlation filter algorithms

(1) Correlation filter-based tracking (CFTs) mainly through the following aspects to improve:

(2) Introducing better training schemes

(3) extract more powerful features

(4) Mitigate the impact of scale changes

(5) In combination with a part-based tracking strategy, which means dividing the target into several parts relative to the overall recognition of the target, each part can be identified

(6) Cooperating with long-term tracking.

8.1.2 Using Deep Learning Tracking Algorithms/29/

These deep learning thoughts could be used into tracking area:

(1) Use auxiliary image data to pre-train the depth model, fine-tuning while tracking online.

(2) The Convolutional Neural Network (CNN) classification network pre-trained by the existing large-scale classification data set is used to extract features.

(3) Pre-training with tracking sequence, fine-tuning while tracking online.

(4) Use of recurrent neural network for target tracking

8.2 Summary

Target tracking in video sequence is an important research topic in the field of computer vision and has been widely used in various fields of national defense and civil affairs. Target tracking integrates advanced technologies and ideas such as image processing, pattern recognition and automatic control, artificial intelligence and computer theory and is a challenging research topic. The current target tracking technology has been partially applied to the video surveillance system, but the video target tracking in practical applications is not very mature tracking algorithm, the interference of light, the target is blocked or disappeared, the target attitude changes and the rapid movement of the target There are still many deficiencies in the aspects of the need to track the stability of the target, robustness, accuracy and real-time and other aspects of improvement and research.

This thesis details the extended application of these algorithms based on the correlation filter tracking algorithm and the existing projects. Also combines correlation filter visual tracking algorithms with Kalman Filter. Somehow, it is a great challenge for me to combine those two advanced technologies to one project. And to be honest, there is still numerous things could be improved

Through the study of these algorithms, I realized that I was lacking in theory. During the mathematical thinking of these algorithms, I encountered many problems because of my own mathematical abilities. But in the process of solving and understanding math thought, I found myself learning a lot. At the same time, I also realized that when a very large and complex issue is solved as part of us, we will always find a solution. In the face of a vast problem, what needs to be done is not discouragement but finding problems to try to solve. Although there are still some restrictions on the application of these algorithms at this stage, new and emerging algorithms are still of great value for further research. In this paper, we mainly take two classical algorithms-- KCF and DSST as an example to discuss the tracking algorithm.

When people face such sophisticated and classic algorithms, the difficulty of learning and the complexity of the project itself is understandable. Especially in the face of excellent algorithms at a level much higher than existing abilities, what can do only dig deeper in this filed by lifelong learning, reading academic paper and attending advanced courses.

For any student who is interested in this area, this article suggests that they firstly thoroughly deal with variable functions, linear algebra, and probability statistics. On this basis, students can then determine the research direction and carry out their work.

We hope all students working on this topic will be well prepared for further study and future career.

9. REFERENCE

/1/ Open Source Computer Vision Library (OpenCV)

https://opencv.org/

/2/ Python Programming Language

https://www.python.org/

/3/ Matlab

https://www.mathworks.com/products/matlab.html

/4/ Collins Retal. A system for video surveillance and monitoring: VSAM final report[R].

Carnegie Mellon University: Technical Report CMU-RI-TR-00-12, 2000.

/5/ Ercan Ozyildiz, Nils Krahnstover, Rajeev Sharma. Adaptive texture and color segmentation for tracking moving objects. Pattern Recognition [J]. 2002, 35:2013-2029.

/6/ Katja Nummiaro, Esther Roller Meier, Luc Van Gool. An Adaptive Color-Based Particle Filter [J].Image and Vision Computing.2003,21 (1):99-110.

/7/ Luc Vosters, Caifeng Shan, Tommaso Gritti.Real-time robust background subtractionunder rapidly changing illumination conditions [J]. Image and Vision Computing.2012,30:1004-1015.

/8/ D B. Babenko, M. Yang, S. Belongie, Robust object tracking with online multiple nstance learning [J]. Pattern Analysis and Machine Intelligence (PAMI).2011 (33): 1619-1632.

/9/ Z. Kalal, K. Mikolajczyk, and J. Matas.Tracking-Learning-Detection [J].Pattern Analysis and Machine Intelligence. 2012,34(7): 1409-1422.

/10/ Z. Kalal, J. Mat as, and K. Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints [A]. In: Computer Vision and Pattern Recognition (CVPR),2010 IEEE Computer Society Conference on[C]. San Francisco:IEEE?2010,23:13-18.

/11/ James Steven, Deva Ramanan. Self-paced learning for long-term tracking[A]. In:Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Computer Society Conference on[C]. Portland:IEEE

/12/ Tracking Algorithms classification

http://blog.csdn.net/App_12062011/article/details/48436959

/13/ Multi-target tracking overview

http://www.c-s-a.org.cn/ch/reader/create_pdf.aspx?file_no=20101259&flag=&journal_id=cas

/14/ Temporal Difference

http://blog.csdn.net/tiemaxiaosu/article/details/51558830

/15/ Optical flow method

https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html

/16/ Visual object tracking using adaptive correlation filters
DS Bolme, JR Beveridge, BA Draper, YM Lui - Computer Vision and Pattern Recognition (CVPR) …, 2010

/17/ Exploiting the Circulant Structure of Tracking-by-detection with Kernels

Jo˜ao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista

/18/ “High-speed tracking with kernelized correlation filters.

João F. Henriques “High-speed tracking with kernelized correlation filters.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 37.3 (2015): 583-596.

/19/ Robust Tracking via Color Names and Spatial Context Learning

INSPEC Accession Number: 16002105, DOI: 10.1109/ICVRV.2015.40

/20/ Accurate Scale Estimation for Robust Visual Tracking.
In Proceedings of the British Machine Vision Conference (BMVC), 2014.

http://www.cvl.isy.liu.se/en/research/objrec/visualtracking/scalvistrack/ScaleTracking_BMVC14.pdf

/21/ Ma C, Yang X, Zhang C, et al. Long-term Correlation Tracking_CVPR2015

/22/ Danelljan M, Häger G, Khan F S, et al. Learning Spatially Regularized Correlation Filters for Visual Tracking_ICCV2015

/23/ Learning Tracking Algorithm.

http://blog.csdn.net/gxb0505/article/details/52601613

/24/ Background of Visual tracking

https://zhuanlan.zhihu.com/p/26415747

/25/ The Visual Object Tracking VOT2014: Challenge and results

www.votchallenge.net/vot2014/download/vot_2014_presentation.pdf

/26/ Visual Tracker Benchmark

http://cvlab.hanyang.ac.kr/tracker_benchmark/

/27/ Kalman Filter.

https://en.wikipedia.org/wiki/Kalman_filter

/28/ Kalman Filter simple example

http://www.jianshu.com/p/c512a2b82907

/29/ The application of deep learning using in object tracking

https://www.leiphone.com/news/201609/jtSho2bUVBvbzTFW.html

APPENDIX 1

Mouse Callback function combine with Kalman Filter

```
def mousemove(event, x, y, s, p):
    global frame, current_measurement,
 measurements, last_measurement, current_prediction, last_predic-
tion

    last_prediction = current_prediction
    last_measurement = current_measurement
    current_measurement = np.array([[np.float32(x)],
[np.float32(y)]])
    kalman.correct(current_measurement)
    current_prediction = kalman.predict()

    lmx, lmy = last_measurement[0], last_measurement[1]
    cmx, cmy = current_measurement[0], current_measurement[1]
    lpx, lpy = last_prediction[0], last_prediction[1]
    cpx, cpy = current_prediction[0], current_prediction[1]


    cv2.line(frame, (lmx, lmy), (cmx, cmy), (255, 255, 0))
    cv2.line(frame, (lpx, lpy), (cpx, cpy), (255, 0, 255))
```

APPENDIX 2

Initial Function of KCF Tracking Mode

```
ok, frame = video.read()
if not ok:
    print 'Cannot read video file'
    sys.exit()

#define the filter
kalman = cv2.KalmanFilter(4, 2)
kalman.measurementMatrix = np.array([[1, 0, 0, 0], [0, 1, 0, 0]],
np.float32)

kalman.transitionMatrix = np.array([[1, 0, 1, 0], [0, 1, 0, 1],
[0, 0, 1, 0], [0, 0, 0, 1]], np.float32)

kalman.processNoiseCov = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0,
0, 1, 0], [0, 0, 0, 1]],np.float32) * 0.8

bbox = cv2.selectROI(frame, False)
# Initialize tracker with first frame and bounding box
ok = tracker.init(frame, bbox)
```