M. Tech (SPDD) ASHWANI KUMAR 2021

VISUAL OBJECT TRACKING BASED ON NORMALISED CROSS CORRELATION AND LEAST SQUARE ESTIMATION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
SIGNAL PROCESSING & DIGITAL DESIGN

Submitted by:

ASHWANI KUMAR
2K19/SPD/04

Under the supervision of

PROF. JEEBANANDA PANDA

DEPT. OF ELECTRONICS & COMM. ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JULY, 2021

VISUAL OBJECT TRACKING BASED ON NORMALISED CROSS CORRELATION AND LEAST SQUARE ESTIMATION

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
SIGNAL PROCESSING & DIGITAL DESIGN

Submitted by:

ASHWANI KUMAR
2K19/SPD/04

Under the supervision of

PROF. JEEBANANDA PANDA

DEPT. OF ELECTRONICS & COMM. ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JULY, 2021

DEPT. OF ELECTRONICS & COMM. ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE's DECLARATION

I, Ashwani Kumar, Roll No. 2K19/SPD/04, student of M.Tech. (Signal Processing & Digital Design), hereby declare that the project Dissertation titled "Visual Object Tracking Based on Normalized Cross-Correlation and Least Square Estimation " which is submitted by me to the Department of Electronics & Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **(ASHWANI KUMAR)**
Date:

DEPT. OF ELECTRONICS & COMM. ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Visual Object Tracking Based on Normalized Cross-Correlation and Least Square Estimation" which is submitted by Ashwani Kumar, Roll No. 2K19/SPD/04, Department of Electronics & Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **(PROF. J PANDA)**
Date: **SUPERVISOR**

ACKNOWLEDGEMENT

A successful project can never be prepared by the efforts of the person to whom the project is assigned, but it also demands the help and guardianship of people who helped in completion of the project.

I would like to thank all those people who have helped me in this research and inspired me during my study.

With profound sense of gratitude, I thank Prof. Jeebananda Panda, my Research Supervisor, for his encouragement, support, patience and his guidance in this research work.

I take immense delight in extending my acknowledgement to my family and friends who have helped me throughout this research work.

ASHWANI KUMAR

ABSTRACT

In the discipline of computer vision, visual object tracking has been a hot topic for some time due to various applications such as observational analysis, athletics analysis, machine navigation, driverless vehicle, human-machine interfacing, and medical imaging etc. Various algorithms have been developed for visual object tracking in the direction of attaining good accuracy, speed and various other challenges that are faced in the implementation of good tracker like object shape, occlusion, changing scale etc. For tracking objects of interest particular detectors are used to detect objects and then tracked accordingly. Tracking process is generally completed by first Detecting and then Tracking. Observational analysis of objects in changing environments is the need of the hour. Tracked objects can also be classified while being tracked other than tracking only. Object tracking is achieved by observing objects' spatial and temporal changes in a series of video frames along with its position and dimensions in frame. In observational analysis, object identification in sequence of frames is significant for tracing and analysing their behaviour. Moving object detection in a series of video frames is of prime importance in and tracking and background subtraction is a basic method for foreground separation. Various problems are faced in object tracking like illumination variation, object deformation, scale change, occlusion, background clutter, out of view, rotation etc. Although with the advent of deep learning technology and availability of high computational power in general purpose and high end embedded computers with very high graphical processing units, it is becoming feasible and more practical to design application of computer vision requiring object tracking which are faster and free from various problems faced in traditional methods at the cost of high computational requirements in training Deep Learning networks but the scarcity of computational power and faster response time in low end machines has led to research on this topic for an optimized algorithm that satisfies our requirement of faster and sufficient accurate tracker in the low end machines. To achieve this task, we have proposed a very simple and elegant fusion of cross-correlation with linear regression for finding the spatio-

temporal location of the object in the series of frames. This fusion is cheap in computational cost which makes this algorithm faster compared to the existing ones at a bit of accuracy cost.

Keywords—object detection, object tracking, cross correlation, Linear regression, least square estimation.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 INTRODUCTION

1.1 GENERAL INTRODUCTION

In computer vision discipline object tracking is an ongoing research issue for monitoring objects of interest in moving frames. Object tracking fits into various important applications of computer vision problems such as athletic observational analysis, machine navigation, autonomous vehicles, human-machine interfaces, and medical imaging, among others. Object detection in a video series is required for tracking, followed by associating the same object in subsequent frames. Because of the lack of computational power and the need for faster response times on low-end machines, we must determine the most appropriate tracker. This necessitates a thorough examination of the computational comparisons of various trackers available in object tracking algorithms, as well as their implementations. In a wide range of applications, including safety, investigation, entrée point management, traffic control, and so on. This process needed the detection of items of interest, the tracking of those things, and lastly the comprehension of their behavior. To deal with such issues, we need to optimize our trackers to meet the needs of apps, and in order to do so, we need to know their computing costs. We present our object tracking method in this study, which is based on normalized cross correlation [3] and the least square estimation [4] methodology. Our research contributed to the fusion of normalized cross correlation and least square estimation techniques, which detects the desired object using cross correlation of the target object with the first frame and then tracks it in a directed gradient of object movement by solving linear square estimation technique on the centroid of the bounding box of detected objects in subsequent frames.

1.2 WHAT IS OBJECT TRACKING?

Tracking can be termed as finding the position and dimension of an object in a series of video frames.

Although the description appears straightforward, tracking is a relatively broad term in computer vision and machine learning that encompasses theoretically related but technically distinct topics. For instance, following concepts, commonly studied to meet object tracking are distinct but connected.

Dense Optical Flow: This algorithm is used to compute motion vector of each pixel in a series of video frames.

Sparse optical flow: These techniques compute the position of specific feature points across series of video frames, such as the Kanade-Lucas-Tomashi (KLT) feature tracker.

Kalman Filtering is one such tracker that uses prior knowledge of object motion to anticipate the position of a moving object in next frames. The guidance of missiles was one of the first implementations of this technique! Kalman filter was first deployed in the apollo 11 mission in guiding the vehicle to lunar surface.

Meanshift and **Camshift** are class of techniques that estimates position of maxima points in a density function.

Single object trackers: these types of trackers, marks starting location of interested object by rectangle to catch its position. The initial object is afterward followed using the tracking method in following frames. SOTs are typically used along with an object detector to start with object detection.

In case of deployment of faster object detector, we can use detection in each frame of video and use tracking method to associate each detected box across a sequence of video frames for tracking purpose.

Detection vs. Tracking

In practice we see that open CV object detection works in real time and can detect faces in every frame of video sequence so what is the point of using tracking than detection?

Speed: Generally tracking algorithms are faster than detection techniques for the obvious reasons. On tracking a previously detected object in the video frame we have all information about the object's appearance, size, speed, direction etc which we can use in estimating the same object in the next frames of sequence in a small neighbourhood region. A good tracker will use all of the knowledge acquired till current frame in estimating the object in next frame but a detection algorithm will at all times start from the beginning. As a result, for efficient algorithm development an object detection is performed at every nth frame after n-1 frames of tracking.

So, why can't we use it in first frame and tracking in the following frames? by doing so we can lose track of the object for the various problems that we encounter in tracking like obstruction behind another object, or if it moves too quickly etc.

tracking algorithms accumulate errors while computing tracking path and finally tracked object loses track from actual object and drift away, so a detection is performed often to remedy these difficulties with tracking methods. Numerous of instances of the entity are used to train detection algorithms. As a result, they are more knowledgeable of the object's broad class. Tracking algorithms have better knowledge for tracking a specific type of object.

When detection fails, tracking supersedes: While using a face detector in a video stream the face detection may go fail on object obstruction but a good tracking algorithm can deal with such occlusions to some extent. Dr. Boris Babenko, the MIL tracker's author, demonstrated MIL tracker performance in such obstructions.

Object detection produces a series of Bounding Boxes around detected object, which is then tracked. The item, on the other hand, has no identity. While running a detector on particular frames of a video it will give us specific spots of detected objects without identification independently in each frame; the detected spots in frame n have no connection with the detected spots in frame n+1, but in tracking the same detected objects are connected.

CHAPTER 2 LITERATURE REVIEW

2.1 ELEMENTARY PHASES FOR OBJECT TRACKING

2.1.1 Object Detection:

Techniques find objects of interest in moving frames and cluster them. There are various methods to accomplish object detection such as difference of frames, flow of optics and background removal [1].

2.1.2 Object Classification:

to further analyse the scenario, we need to classify the different objects depending on the Motion, Shape, texture and Colour [1].

2.1.3 Object Tracking:

The object tracking techniques associates the same objects appeared in successive frames with previous ones. These techniques can be of point tracking, kernel tracking and silhouette tracking etc. type.

Issues that we need to take care in object tracking are:

1. Shape information is lost when 3D data is projected onto a 2D image.
2. image noise
3. fast motion
4. partly blockings
5. intricate Shapes.

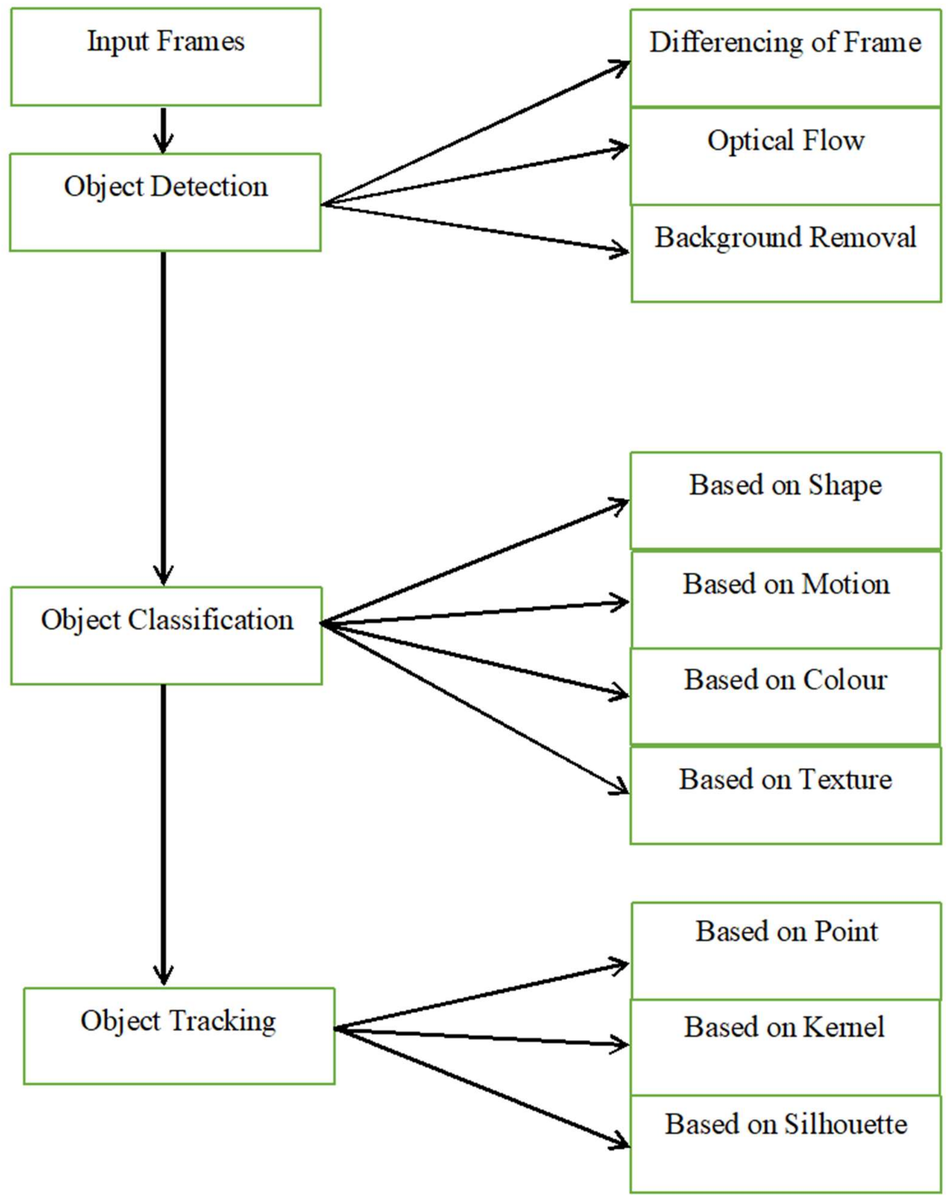

Figure 1 Elementary phases for object tracking [1].

CHAPTER 3 OBJECT DETECTION METHODS

At the outset the task of tracking starts with extraction of object of interest in the moving frames and group the similar objects and annotate them. As these objects are changing content in images form the first information in tracking that's why we focus on finding of such contents. Exhaustive description for numerous approaches is as follows:

3.1 Frame Difference:

The existence of changing items can be identified by computing the difference of two following frames of a video. this method id quite easy to implement, its simply point by point subtraction. The output of this method is not accurate because of incomplete shape outline in a changing environment of scene being imaged. [1].

3.2 Optical Flow:

in this method gradient of pixel flow is calculated and similar pixels are grouped to form objects. By this technique we get the most of the objects movement information in an image with respect to background, still it is seldom used in practical problems because of its high computational costs, noise sensitivity etc.

3.3 Background Subtraction:

in subtracting background from frame background model needs to be modelled first. Background modelling should be responsive to changing contents of foreground objects. This process creates a reference model. This reference model is subtracted from subsequent frames to detect moving contents across frames of a video images. Presently, median filter and mean filter are extensively used to implement background modelling. This can accurately detect objects provided we have modelled background properly. Background subtraction can be achieved in two ways:

1. Recursive algorithm: this method recursively keeps informed a model depending on input frame. Due to this recursive nature input frames have an outcome on present background model. This method requires less memory but any fault in model may creep in for a length of time. adaptive background, Gaussian of mixture, approximate median are some of its examples.

2. Non-Recursive Algorithm: in non-recursive method a fixed no past information is used for estimating background model. This method is very much changing because their output depends on very short past values of frames, but if a sluggish moving object needs to be tracked then memory requirements could become larger.

Table 1 Comparative Chart of object detection methods [1]

Techniques		Correctness	Calculation Time	Remarks
Background Subtraction	Gaussian Approximation	Medium	Medium	+ small buffer
				- Cannot support mix of models background
	Approximate Median	small to Medium	Medium	+ Decimation of frames not required for background modelling.
				- Latest frames are required for calculating BG model

Optical Flow	Medium	Large	+ Computes very accurately - computationally expensive
Frame Differencing	Large	Low to Medium	+ Simple. Good for fixed background. - A fixed BG is expected.

CHAPTER 4 CLASSIFICATION METHODS

The objects are classified depending on their shape, motion, colour or texture: that's why we chose these features to detect the object and classify them. Their detailed information is as follows:

4.1 Classification based on Shape:

objects can be classified depending on various features of shape related like points, lines, polygons etc. mixture of these feature points classify an object for a particular object.

4.2 Motion-based classification:

some moving objects motion depicts regular pattern of movement, which can be utilized for its classification. Optical flow is one such example for classifying an object based on its optical movement of brightness fields.

Tracking using this method is done by following changes in the brightness level at each pixel within a tolerated level of range of pixels across frames.

after these optical fields are calculated across frames Lucas kannade method ccan be used for tracking of the detected object in the moving frames.

4.3 Colour-based classification:

Colour information is not a strong feature like other features of shape, motion etc. but still a range of colour hue can make the algorithm relatively computationally cheap and simple if the moving object is quite apparent in respect of background i.e., pedestrians. For detection purpose, in this method a histogram of colour features is maintained for segmentation of moving object with background. Few mixture models can be utilized for identifying the distribution of foreground with background like Gaussian Mixture Model.

4.4 Texture-based classification

in this methodology orientation of gradients of brightness changes in a local grid of a frame is used for extracting a blob of foreground out of different oriented gradients of background. All these methods and techniques of object detection are listed and compared in table 2 below showing their accuracy, cost of computation and special remarks:

Table 2 Relative features of object classification techniques [1].

Methods	**Accuracy**	**Computational Tiime**	**Comments**
Based on Shape	Medium	small	Simple to implement, Similar shape matching via constituents, Cannot handle complex movements.
Based on Motion	Medium	Large	Identifies object from its motion field

Based on Texture	Large	Large	computation time is more with better quality of detection.
Based on Colour	Large	Large	Simple to implement, uses distribution of colour info to differentiate background with foreground object.

CHAPTER 5 OBJECT TRACKING TECHNIQUES

5.1 Techniques:

Object Tracking determines the identifying the movement of similar object across moving frames in a sequence of video. Various applications can be implemented by knowing the flow of desired object in a series of images i.e. object exclusion, object identification and monitoring, and their behavioural analysis based on movements. Basically, there are three types of tracking methods like point tracking, kernel-based tracking and silhouette-based tracking. In point tracking, we identify objects on each pixel in each frame; and in kernel-based tracking, finding the object trajectory based on its shape and subsequent similarities in upcoming frames.

tracking methods and their sub methods are shown in following diagram:

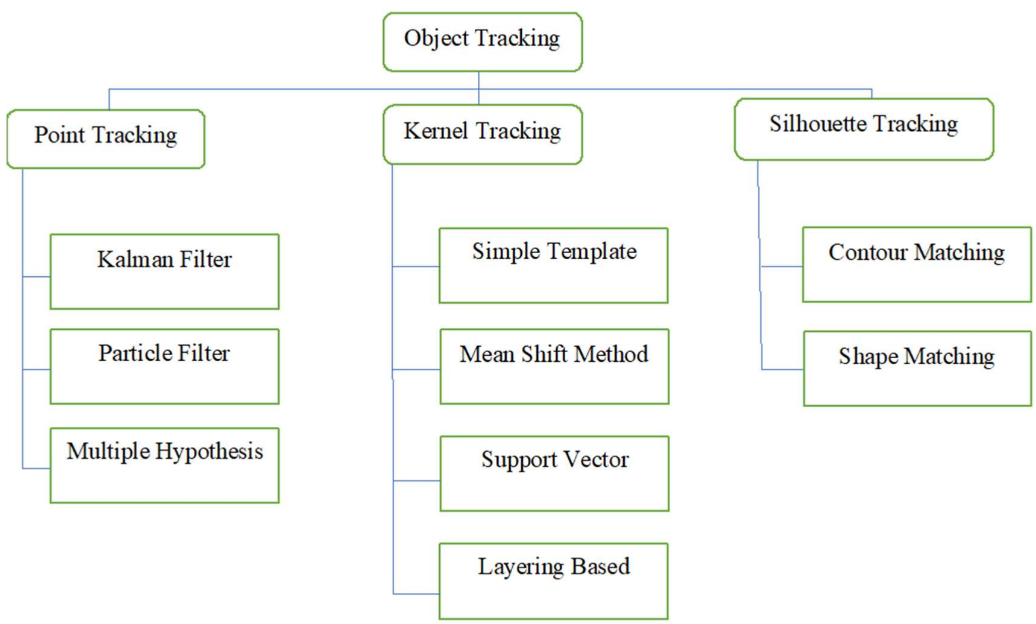

Figure 2 Object Tracking techniques [1].

5.2 OBJECT TRACKING WITH OPENCV

OpenCV has different types of tracking algorithms in its implementation. In this section we will have a brief introduction of available object tracking algorithms in OpenCV implementation

Let me start by laying out some fundamental tracking ideas. The goal of tracking is to discover an entity in the present frame after successfully tracking it in all (or almost all) preceding frames. We know how the thing has been travelling because we've been tracking it up until this point. In other words, we are aware of the motion model's parameters. Here the motion model can be understood as a process model based on the prior information tracker learns from the object's movement like speed and direction of motion. So, by learning motion model and detecting object features we can predict objects of interest in the subsequent frames of video streams.

However, we have more information than just the object's motion. In each of the previous frames, we can see how the thing appears. To put it another way, we can create an appearance model that describes how the picture appears. The model created by like this is used to explore within a short radius of the motion model's anticipated position to more correctly anticipate the object's position.

The motion model forecasts the object's approximate location. This estimate is fine-tuned by the appearance model to produce a more accurate estimate depending on appearance.

A simple template matching method can be used to compare the appearance model with object in candidate region and search for similar pattern. But in Real life, on the other hand, is not so straightforward. An object's appearance can vary substantially. So to address this issue of complex appearance trackers can be trained to learn objects appearance using a classifier that is trained online rather than hand woven features of object description.

The classifier's task is to determine whether a rectangular section of a picture is an object or a non-object. The classifier takes an picture segment as input and returns a number ranging from 0 to 1 indicating the likelihood of picture segment includes the desired object. When it is totally certain that the picture segment is the background, the number is 0, and when it is definitely certain that the segment is the object, the score is 1.

The term "online" is used to describe algorithms that are trained while running. An offline classifier requires thousands of instances for training, but an online classifier is often taught with a small number of instances at run time.

objects and background instances are fed to a classifier for its training. we require thousands of positive samples and thousands of negative samples for training the classifier. Like this, the classifier crams to distinguish between what is object and is what is non object. We don't require of thousands of instances of positive and negative classes to work with while developing an online classifier.

In the following section we will see different tracking algorithms provided in openCV addressing the online training challenge.

5.2.1 BOOSTING Tracker

Boosting tracker works on the adaboost classifier in online mode, adaboost algorithm is also used in HAAR cascade face Detector. this tracker is learnt with both object and non-object instances during run time. Objects are either supplied by user or detected by an independent detector and remaining non-object patches are used as negative samples in training boosting tracker. The classifier used in boosting tracker classifies every pixel around an earlier position in a new frame and the classifier score is saved. the objects predicted position is the one with the highest score. As a result, we now have yet another positive case for the classifier. The classifier score is modified on receiving subsequent frames as positive case for the classifier.

MIL, and KCF algorithms work on similar concepts.

Cons: The performance is subpar. It doesn't always know when tracking isn't working.

5.2.2 MIL Tracker

It uses same concept as the BOOSTING tracker explained earlier. The major distinction is that it searches a trivial neighbourhood surrounding the present place for multiple possible positive examples than using only the object's present location as a positive instance.

MIL (Multiple Instance Learning) solves this situation. we don't describe positive and negative instances in MIL; instead, we specify positive and negative "bags." The positive bag's collection of photographs isn't entirely made up of positive examples as an example! Rather a single image from the positive bag is sufficient!

A positive bag, in our case, has a patch centred on the object's current location as well as patches from a small neighbourhood around it. With the tracked object's present position is erroneous, when samples from the surrounding area are placed in the +ve bag, there is a strong likelihood of bag containing minimum one picture with the object perfectly centred. For those interested in learning more about the MIL tracker's inner workings, see the MIL project page.

Advantages: The performance is quite decent. Drifting problem is also not that much as of the BOOSTING tracker, and it performs admirably when partially obscured. This might be the best tracker available if you're using OpenCV 3.0. However, if you're using a later version, KCF is a good option.

Cons: Tracking failures are not reliably reported. After a complete occlusion, it does not recover.

5.2.3 KCF Tracker

Kernelized Correlation Filters (KFC) is an acronym for Kernelized Correlation Filters. The principles introduced in the previous two trackers are expanded upon in this tracker. This tracker takes advantage of the fact that the MIL tracker's numerous positive examples have vast common zones. The commonality in positive information samples gives us special features those can be used in making this tracker faster and more precise

Pros: It outperforms BOOSTING and MIL in terms of accuracy and speed, tracking breakdown is also better than last 2 mentioned trackers.

Cons: Does not recover from occlusion completely.

5.2.4 TLD Tracker

TLD is the acronym for tracking, learning and detection. this tracker, as the name implies, can be decomposed into three subtasks namely, tracking, learning and detection. this tracker follows the object across each frame, the detector locates all previously detected appearances and, if necessary, adjusts the tracker

The learning part of the tracker calculates the detectors' fault and modifies it to avoid them for upcoming frames. The output of this tracker has a tendency to jump about. For instance, if tracker is monitoring a person and there are other persons, the tracker may momentarily monitor a person other than intended one. The plus point of this tracker is its ability to follow the intended object in terms of scale velocity and obstruction.

Pros: Works best when numerous frames are occluded. In addition, it works best when the scale changes.

Cons: There are so many false positives that it's virtually worthless.

5.2.5 MEDIANFLOW Tracker

MF tracker measures difference between the object's progressing and regressive trajectories and tracks it in both directions in time. It can accurately compute tracking errors & identify consistent paths in movie stream by reducing the Forward Backward error.

MF tracker stops tracking on recognition of tracking failure.

Pros: Admirable tracking with detection of tracking failures. This technique works very well when the object movement is foreseeable and occlusion free.

Cons: When there is a lot of movement, it fails.

5.2.6 CSRT tracker

A modified discriminative correlation filter with channel and spatial reliability map (DCF-CSR) is used in modifying the filter backing to the part of selected region out of image for monitoring. This results in improved chasing of irregular regions objects with the specified, distended and localised objects. For accomplishing this technique CSRT tracker uses HoG and Colour features. This tracker operates at reduced speed but delivers improved object tracking performance.

CHAPTER 6 PROPOSED TRACKER

6.1 BACKGROUND

In both academia and industry, machine learning is a boiling topic, with new approaches being developed on a regular basis. Even for professionals, keeping up with new tactics is challenging — and potentially intimidating for beginners — due to the field's speed and intricacy.

There are various distinct methods, each with easy explanations, graphics, and illustrations, to elucidate machine learning & provide a route for naïve individuals to learn them.

A machine learning algorithm, often known as a mathematical expression that tries to solve the given problem by computational means. For example, an online store wishes to forecast its business for the following time segment, it may employ a computational algorithm that estimates business output depending on previous business output.

Before we get started, there's one more item I'd want to mention. Let's differentiate between two types of machine learning, (1) supervised machine learning (2) unsupervised machine learning. If we wish to predict or explain about some past data then we use supervised algorithms. We can prediction of an outcome based on a new input using past data on inputs and outputs. For example, supervised machine learning techniques might be used to assist a service provider in predicting the sum of people joining for the services in the upcoming time. But, on the other hand, for connecting and grouping information bytes without knowledge of an end goal point unsupervised machine learning algorithms are used. It evaluates features information to group related things into clusters. For example, unsupervised learning techniques could be used to assist a shop who wants to group products based on comparable features without having to specify which qualities to use in advance.

6.2 CORRELATION

What Does Correlation Mean?

Correlation is the degree of common association that occurs among two or more objects. In case of signals, the same definition applies. Correlation between signals, in other words, reflects the degree to which one signal resembles another.

To put it another way, if we want to know how similar the signals 1 and 2 are, we need to determine the correlation coefficient between the two signals.

6.2.1 Correlation Types

Autocorrelation and cross-correlation are its two types, depending on whether the signals is being correlated with itself or other signal.

6.2.1.1 AUTOCORRELATION

Autocorrelation is a term that refers to the phenomenon of the provided signal is compared with its own time shifted or non-time shifted version. usually the time-shifted form of itself, in this sort of correlation. Mathematically, the autocorrelation of a signal x(t) is expressed as

$$Rxx(\tau) = \int_{-\infty}^{\infty} x(t)x \star (t-\tau)dt$$

……………..(1)

symbol $\star$is used to denote the complex conjugate form.

Likewise, the discrete time autocorrelation of the signal x[n] is expressed as

$$Rxx[m] = \sum_{n=-\infty}^{\infty} x[n]x \star [n-m]$$

………………………….(2)

The autocorrelation of any given signal can then be calculated using a graphical method. The process entails slapping the time-shifted form of signal on itself when calculating signal at each interval point. For example, for the digital given, signal is shifted by one interval point and overlapped with itself. In this calculation, we multiply and add for each shift and overlap. Figure 1 shows an example of calculation of autocorrelation.

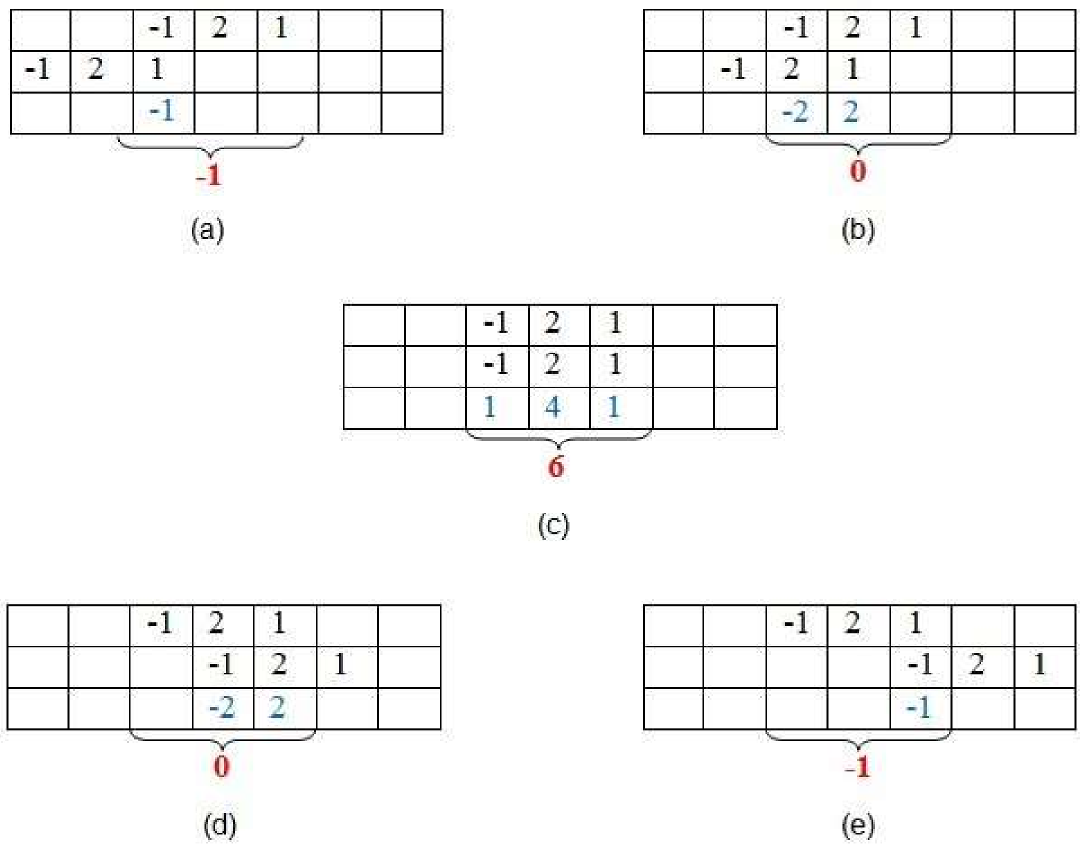

Figure 3 Graphical method of finding Autocorrelation

The provided signal is represented by the initial group of signal points. The second line of signal points is the time-shifted form of original signal. After that, multiply the appropriate samples from the initial two lines to get the signal points shown in red colour in the third line.

To conclude, all the computed signal points in red colour like this are grouped in order of time to make the autocorrelated version of signal's samples.

The computed values of autocorrelated signal will be maximum at its zero-time sample shifted version.

6.2.1.2 CROSS-CORRELATION

In cross-correlation, the signal in question is compared to a different signal to determine their similarity level. The cross-correlation of two continuous time signals x (t) and y (t) can be expressed mathematically as

$$Rxy(\tau) = \int_{-\infty}^{\infty} x(t)y \star (t - \tau)dt \qquad \text{……………………………..(3)}$$

Likewise, discrete time cross-correlation of the signals x [n] and y [n] is represented like

$$Rxy[m] = \sum_{n=-\infty}^{\infty} x[n]y \star [n - m] \qquad \text{…………………………….(4)}$$

Cross-correlation of two signals can also be computed same way as autocorrelation approach. One signal is dragged over the top of another signal and the signal points are computed at each interval point. So finally, a sum of the product is computed of the overlapping signal points.

Figure 2 illustrates the cross-correlation of two digital signals x [n] and y [n].

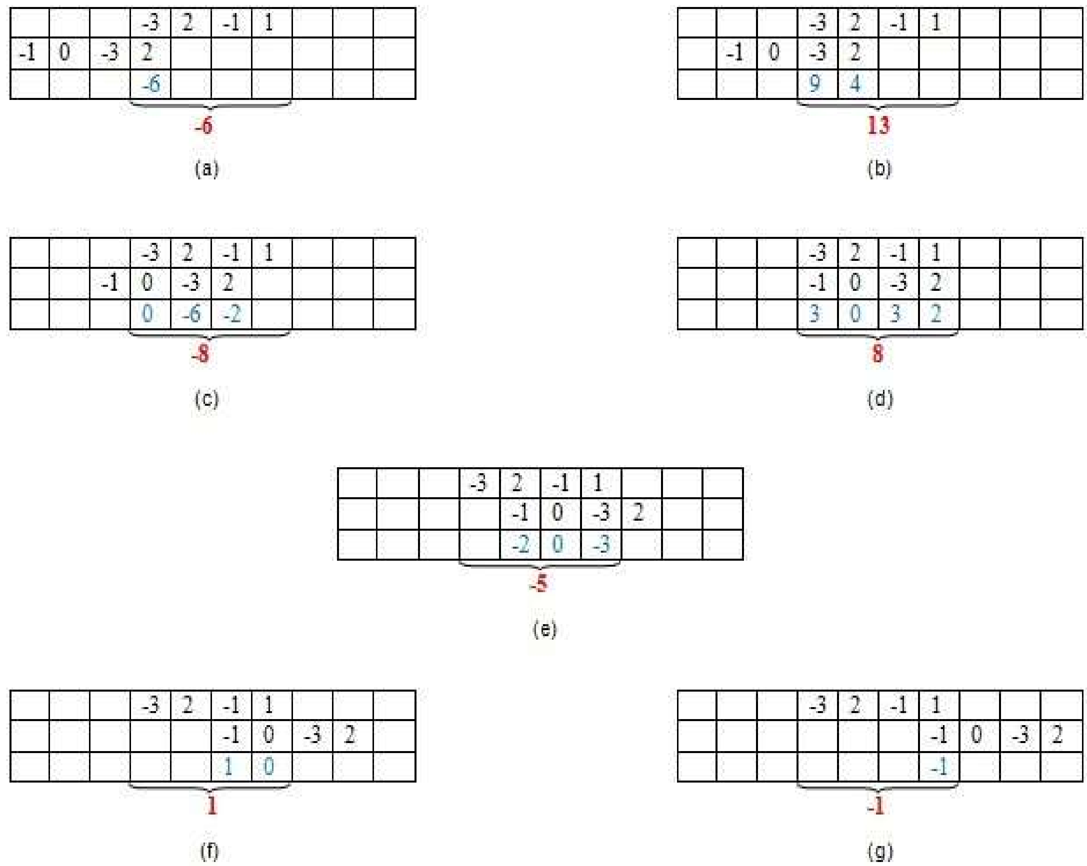

Figure 4 Calculating cross-correlation of two discrete signals

The first signal x [n] is shown by the initial group of signal points and the second signal y[n] is shown by another group of signal points.

Blue-coloured samples in the third line are then calculated by multiplying the analogous signal points in the previous lines. Lastly, all the computed signal points in red colour like this are grouped in order of time to make the cross-correlated version of signal's points. The cross-correlated signal points are at its maximum point, with value 13, when y [n] and x [n] are identical.

As a result, the maximum value of two cross-correlation signals is when the two signals under consideration are most identical to one another.

6.2.1.3 NORMALIZED CROSS-CORRELATION

When computing cross-correlation degree between two signals, it is time dependent values of signals, in signal processing area, normalization cross-correlation function is crucial because it affects the statistical features of the calculated cross-correlation.

The normalized cross-correlation of a statistical process offers a scale-free degree of the amount of statistical dependence, and its value lies from 1 to -1, 1 denotes absolute correlation and -1 denotes absolute anti-correlation of two different scaled signals.

6.3 OBJECT TRACKING AS MOTION PREDICTION

Object tracking can be accomplished or completed by prediction of object motion in sequence of video frames

6.3.1 PREDICTION IN MACHINE LEARNING

Predictive analytics is a domain of statistics that can be used in estimating for predicting the future events with the use of statistical methodologies. To do predictive analysis a number of approaches can be used like predictive modelling and machine learning also. These outcomes could include, for example, customer behaviours or market shifts. By analysing the past, predictive analytics helps in predicting the possible future events. While Machine learning is a branch of computer science that makes the computers to learn from data itself without being programmed explicitly, according to Arthur Samuel's 1959 definition. Machine learning, a type of pattern recognition that investigates idea that computers can learn from data itself and predict possible outcome. These algorithms can also transcend software commands in making extremely exact, data-driven conclusions as they become more "intelligent."

What are predictive analytics and how does it work?

Predictive modelling is at the heart of predictive analytics. As machine learning algorithm are deployed in developing predictive models so they play as heart in predictive analytics. These models may be skilled to adapt to fresh information over time, producing outcomes that company requires. Machine learning and predictive modelling are closely related fields. Predictive models are divided into two categories. Basically, classification models and regression models are found in predictive analytics which are used in classifying the class model and predicting numerical values respectively. Algorithms are used to create these models. Data mining and statistical analysis are carried out by the algorithms, which identify trends and patterns in the data.

The extensively used models are:

Decision trees, Regression (linear and logistic), Neural networks

Other models:

Time Series Algorithms, Clustering Algorithms, Naïve Bayes classifiers and Support vector machines

6.3.1.1 REGRESSION AS PREDICTION

Regression

The regression is from the type of supervised machine learning algorithms. They are utilized in predicting the futuristic arithmetical information depending on a set of historical data, for example estimating a property's future price depending on earlier rating of alike properties. linear regression is simplest method, that groups the collected information by equation ($y = m * x + b$). From this line equation and the data values we have the overall distance measure that can now be calculated and minimized to match model with numerous data samples.

Let's look at an example of linear regression that is more concrete. By combining different parameters of a building, we can predict the energy consumption of particular structures. To achieve this a multi-variable linear regression model can be used because there were multiple inputs (age, square feet, etc.). The same theory is followed in training the multi variable linear regressor as simple linear but in multidimensional space depending on the number of independent variables.

The graph shown underneath demonstrates prediction of actual building energy use by the linear regression model.

Imagine you have access to a building's characteristics (age, square footage, etc.) but not to its energy use. In this scenario, the fitted line can be used to estimate the energy consumption of a specific building. It's worth noting that linear regression can also be used to determine the most important factor that plays role in determining the final value of property that goes into prediction.

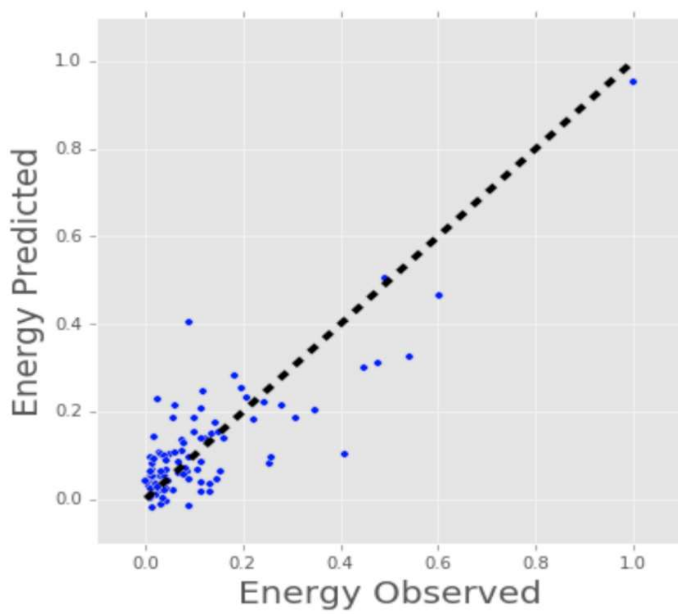

Figure 5 Energy consumption prediction with linear regression model

6.3.1.1.1 TYPES OF REGRESSION

- Regression has basically two types namely linear and logistic.

6.3.1.1.1.1 SIMPLE LINEAR REGRESSION

For identifying a connection among 2 incessant variables, simple linear regression is appropriate. There are two variables independent variable and dependent variable. It searches for statistical associations rather than deterministic ones.

Real-time example

In this example we have got a data about the marks awarded to the students and the amount of study hours they have devoted in studying. From this data we will train our linear regressor, our target will be to develop a model that will be able to predict the potential grades or marks corresponding to the study hours for any set of data. A regression line is created using the training data that will yield the least amount of error. Any fresh data is then fed into this linear equation. That is, if we provide a student's number of hours studied as an input, model should accurately predict their grade.

Y(pred) = b0 + b1*x ………………………………………………(5)

The variables b0 and b1 should be selected so that the loss is minimised. The target of training the regressor is to find the optimal values of model so that error is minimal.

$$\text{Error} = \sum_{i=1}^{n}(actual\_output - predicted\_output) ** 2 \quad \ldots\ldots\ldots\ldots(6)$$

$$b_0 = \bar{y} - b_1\bar{x} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(7)$$

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(8)$$

Exploring ‘b1’

• for b1 > 0, a positive relationship exists between x and y.

• for b1 less than zero, a negative connection is there between x and y.

Exploring ‘b0’

- Non-inclusion of value x=0, the forecast using simply b0 will be worthless. For example, we have a dataset that shows the relationship between height and weight. Using x=0 will result in a calculation with simply b0 as an output value, which is entirely useless because a zero height for zero weight is illogical.
- Upon inclusion of x=0 in the model, ‘b0' gives the normal projected value. However, it is frequently impossible to set all predictor variables to zero.
- b0 keeps a zero mean residual value. Regression lines passes through origin in absence of ‘b0' coefficient. In this case the b1 as well as the output will be skewed.

Co-efficient from Normal equations

Other than the aforementioned calculation, the model's co-efficient can also be determined using the normal equation.

$$\text{Theta} = (X^T X)^{-1} X^T Y \quad \text{...(9)}$$

Theta is the sum of all co-efficients. The inverse of the input matrix is used to compute the normal equation. When the number of characteristics increases, it becomes extremely sluggish.

Residual Analysis

A regression model's two major components are randomness and unpredictability.

Deterministic + Statistic = Prediction

The forecaster variable in the equation takes care of the deterministic aspect. Stochastic component demonstrates that the expected and actual values are both unpredictable. There will always be certain details that are overlooked. The residual data can be used to gain this information.

Let's use an example to illustrate the idea of residue. Consider the following scenario: we have a dataset that predicts juice sales based on the temperature at a location. The value predicted by the regression equation will almost always differ from the actual value. The genuine output value will not be perfectly matched by sales. This distinction is referred to as residue.

The residue plot aids in the analysis of the model by displaying the values of the residues. It's shown as a line between the projected values and the residual.

They all have the same values. The point's distance from 0 indicates how inaccurate the prediction was for that particular number. The prediction is low for +ve value. The estimate is high for -ve value. An estimate with a 0 value is perfect. The model can be enhanced by detecting residual patterns.

Metrics for model evaluation

R-Squared value

This number can be anywhere between 0 and 1. The value '1' denotes that the perfect prediction. The 0 value means no relationship between predictor and output parameter.

1. Regression sum of squares (SSR)
indicates the difference between predicted and actual average).

$$\text{Error} = \sum_{i=1}^{n}(\text{Predicted\_output} - \text{average\_of\_actual\_output})\hat{}2$$ ………..(10)

2. Sum of Squared error (SSE)

$$\text{Error} = \sum_{i=1}^{n}(Actual\_output - predicted\_output)\hat{}2$$ ……………..(11)

3. Total sum of squares (SSTO)
It indicates deviation from mean.

$$\text{Error} = \sum_{i=1}^{n}(\text{Actual\_output} - \text{average\_of\_actual\_output})\hat{}2$$

$$\mathbf{R\hat{}2 = 1 - (SSE/SSTO)}$$ ..(12)

R-Square range

Forcing regression line to permit via a point, the R square value may become negative. This will compel the regression line to have no intercept value, resulting in an mistake greater than the average line's mistake. If the data is remote from the source, this will happen.

Correlation co-efficient (r)

R is the square root value of ‘r-squared,' as shown below. It has a range from -1 to 1.

r = +/-sqrt(r2)

‘r' is -ve for -ve ‘b1’ and +ve for +ve ‘b1’ . It doesn't have any units.

Null-Hypothesis and P-value

The null hypothesis is the first assertion made by the researcher based on past study or information.

Low P-value indicates that the close relationship between predictor and response, rejecting the null hypothesis.

High P-value: No relation between predictor and target values changes.

Regression line Obtained

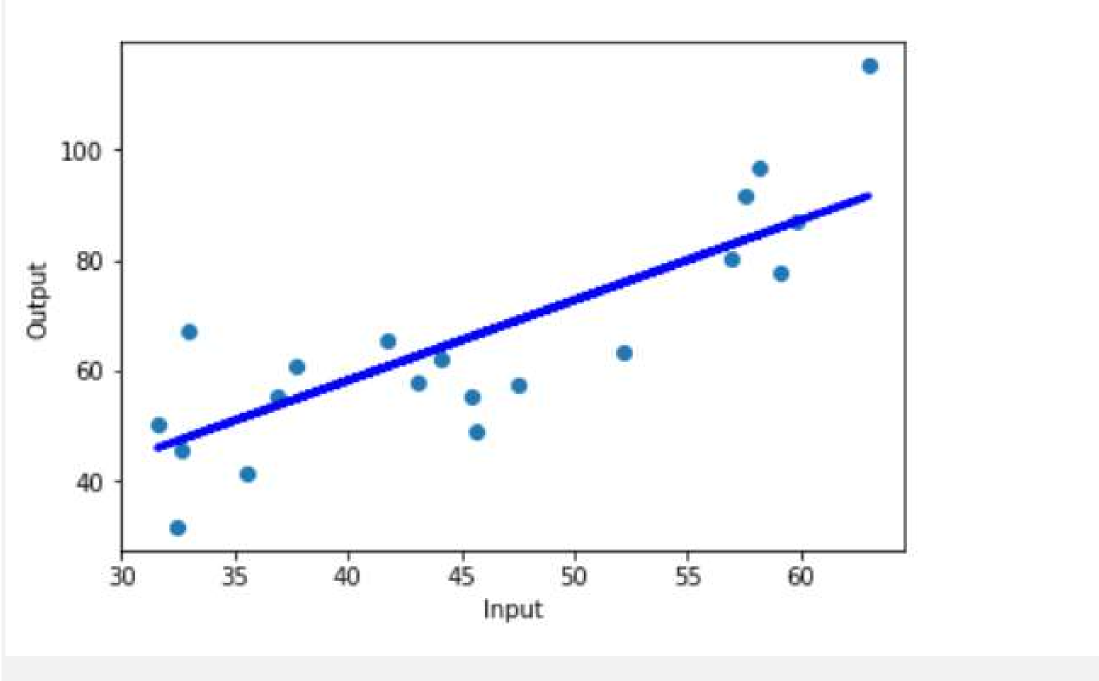

Figure 6 Final Regression line over test data

6.4 LEAST SQUARE ESTIMATION AS LINEAR REGRESSOR

Least Square Estimation:

In machine learning, signal processing, and statistics, least-square estimate is one of the most extensively used techniques. It is the most frequent method of solving linear regression, which is commonly used to represent continuous outcomes.

It can be described as a Bayes estimator with a quadratic cost or as an MMSE estimator. I've already written in-depth papers on a variety of machine learning and signal processing estimators. For more details, please see the following article.

In this post, I'll go over least squares in detail; in subsequent articles, I'll go over multi-objective, constraint, and nonlinear least-squares in greater depth.

Ordinary Least-Square (OLS)

If A is a m x n matrix with m higher than or equal to n (a tall matrix), then the following overdetermined system of equations either has a unique solution or has none.

$$Ax = b \text{.....................................(13)}$$

Figure 1: Linear Equation

The unique solution happens if b is a linear combination of columns of A, which means that if A is expressed as a column representation, b will be in A's column space. Another way to put it is that there exist scalers x1, x2,..., xn that have a solution to the following equation.

$$A = [a_1 \quad a_2 \quad \ldots \quad a_n]$$

$$x_1a_1 + x_2a_2 + \ldots + x_na_n = b \text{...........................(14)}$$

Note: Consider the assumption for the least-square issue to see why if the solution exists, it is unique. A has a left inverse if it is tall and has linearly independent columns. The unique solution is obtained by multiplying the left and right sides of equation 1 by the left inverse of A. The rest of this post will show you how to obtain this answer.

Note: Please watch the following videos to learn more about overdetermined, undetermined systems of equations, tall, wide matrices, and matrix column and row interpretation:

6.5 Least Squares Regression Line Calculator

- finding the best fit line
- equation of Least squares regression line
- Least squares technique
- Least squares regression line equation finding
- Limitations of Least square fit

This section contains some important information regarding the least square approach, including method of obtaining the least squares line and avenues for when completing the fit.

6.5.1 Finding the best fit line

You can try to construct a line that goes as close to all of the spots as feasible intuitively. It may be a straight line in some cases, indicating that we will run a linear regression. There are several approaches to this problem, with least squares estimates being the most popular and commonly employed. Here are some instances from actual life:

Your car's engine produces more combustion as you travel quicker. Perhaps the winter is bitterly cold, or the summer is scorching hot, necessitating the purchase of additional electricity for heating and air conditioning. You may conceive a slew of other scenarios in which an increase in A leads B to expand (or decay). Why do we make use of it? We can roughly forecast the outcome of a future occurrence with only a few data points. This is why knowing how to find the best-fitting line is advantageous. It will assist you in determining the B/A ratio at any given time. This calculator for finding the least square regression line demonstrates how to do so.

6.5.2 Equation of Least squares regression line

As we all know the equation for a straight line is $y = a * x + b$ with a slope, a, and an intercept, b, to make things as plain as possible. We will use this equation $y = a * x + b$ for fitting best fit line with least squares estimation.

The secret is in the method of calculating the a and b parameters. This is fantastic!

6.5.3 Least squares method

The concept of finding least square method to determine the best fit line is straightforward:

1. sketch a line with the formula f(x) = a*x + b.

2. Determine all di=|yi - f(xi)|

3. Make a square with them: di2.

4. Add them up: Z = di2 = d12 + d22 + d32 +...

5. Draw a line that reduces the value of Z to the smallest possible value.

6. Take pleasure in learning the origin of the least squares method's name.

It may appear a little hazy at first glance, so let's look at some photographs to help clarify things. For the same data points, three distinct lines have been fitted: (1,2), (2,6), (3,4), (4,7):

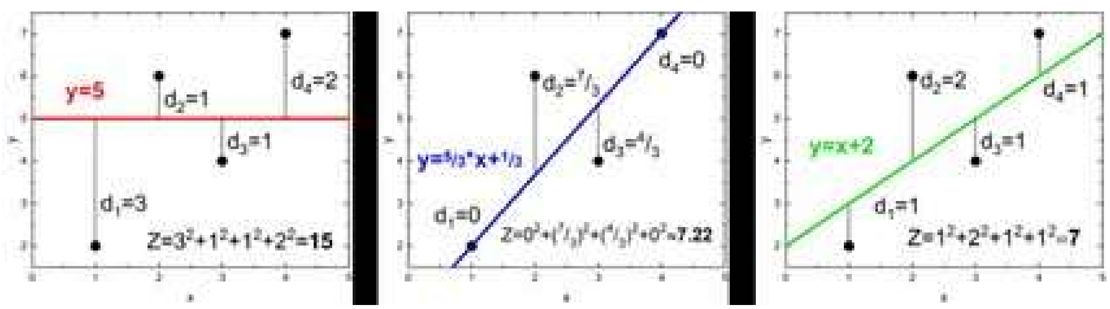

Figure 7 Best fit line by Least Square Method

Z has distinct values in each scenario, as you can see. For the third plot, it's bare bones, but can we do better? To see if that's the best option, use our least squares regression line calculator

6.5.4 Least squares regression line finding

- The regression line parameters a and b must be estimated. We may calculate a few auxiliary numbers using the normal least square method, that simplifies the end equation:

- $S_x = \sum x_i = x_1 + x_2 + x_3 + \dots$
- $S_y = \sum y_i = y_1 + y_2 + y_3 + \dots$
- $S_{xx} = \sum x_i^2 = x_1^2 + x_2^2 + x_3^2 + \dots$
- $S_{yy} = \sum y_i^2 = y_1^2 + y_2^2 + y_3^2 + \dots$
- $S_{xy} = \sum x_i y_i = x_1 y_1 + x_2 y_2 + x_3 y_3 + \dots$
- $\Delta = n*S_{xx} - S_x^2$

These coefficients produce the least square fit:

- $a = (n*S_{xy} - S_x*S_y) / \Delta$
- $b = (S_{xx}*S_y - S_x*S_{xy}) / \Delta$

You can get numerical values by solving these formulas.

- $\sigma a = \sqrt{(n/(n-2) * (Syy - a * Sxy - b * Sy) / \Delta)}$
- $\sigma_b = \sqrt{(S_{xx} / n)} * \sigma_a$

Round the parameters to the matching decimal integers. Remember to utilise scientific notation when dealing with extremely large or extremely small numbers.

Finally, the Pearson correlation coefficient can be found., `r`:

- $r = (n * S_{xy} - S_x * S_y) / \sqrt{((n*S_{xx} - S_x^2) * (n * S_{yy} - S_y^2))}$

value of r might be anything between 0 and 1. The least square fit improves as it gets closer to unity (1). Our data points reveal no linear relationship as the value approaches zero.

A quick note: We assume that the y values surrounding real dependence follow a normal distribution, that is intended in replicating by regression line.

6.5.5 Limitations of Least square fit

The least square approach is popular & commonly employed, it is important to remember that it is imperfect and might be deceptive in some situations. The factors that control the accuracy of this method are as follows:

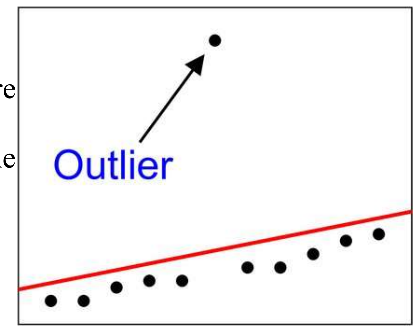

Figure 8 Effect of outlier in LSE method

- The method is prone to outliers, and the more points in your data, the greater the accuracy of the least square fit.
- A single point that obviously deviates from the overall trend will have an impact and distort the outcome.
- if the data distribution is non-linear in nature, a parabola or other related functions should be considered as the fitting curve. However, a straight line can be fit to anything with greater degree of error.

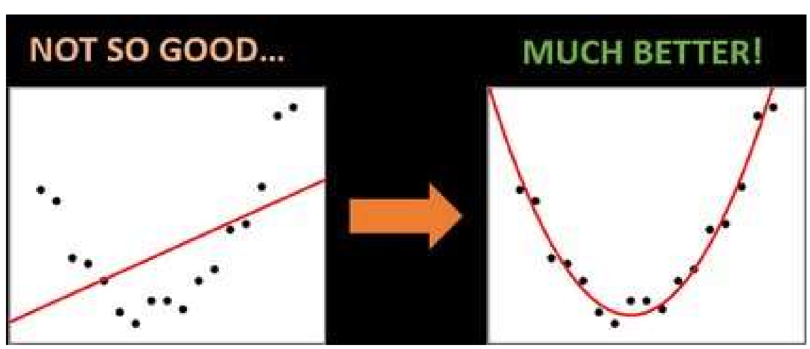

Figure 9 Limitation of Least Square Method in Non-Linear Relationship data

6.6 PROPOSED METHODOLOGY OF TRACKER

The flowchart of our methodology is shown in figure 1, As a proof of concept of our methodology we are extracting the target image by selecting a ROI in the initial frame for target detection and using this target image for tracking in the following frames by solving trajectory of target object through least square estimation and template matching with 2D normalized cross correlation.

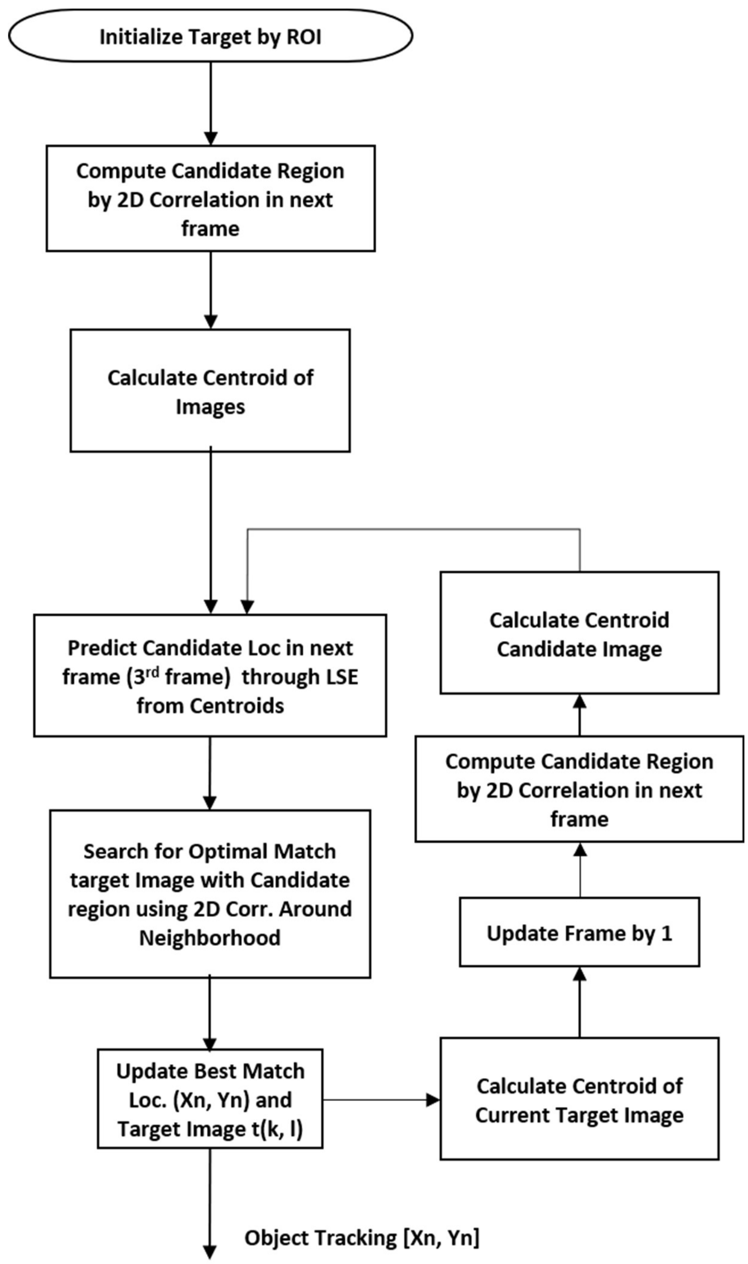

Figure 10 Proposed Methodology Algorithm

6.6.1 Foreground Extraction/ Target Modelling And Representation

In this paper we have used a ROI (region of interest) extracted by a user and represented by a rectangular bounding box of image intensity values in the initial frame for simplicity. Opencv function rectangular object representation

6.6.2 Similarity Target Matching

In our methodology, we have used 2D normalized correlation as a reference image matching with candidate region for finding the similarity score between the two. There are also other methods like Bhattacharya coefficient for finding similarity match between the two [15]. Reference image matching is a basic and simple pattern matching technique in digital image processing. With this operation one can evaluate the resemblance between reference image and a target image. In two dimensional images, reference image matching deploys a reference image, an example of a real image or a synthetized sample of the pattern. The pattern is subset of the candidate image. With this two-dimensional correlation, one finds the availability of reference image in target. The correlation method uses the correlation coefficient as a degree of resemblance between the reference image for each location (x, y) in the target image. The computed coefficient from this operation returns maximum value in the target image sub location where reference image fits better. This operation can be understood as convolution of reference image t(x, y) as a spatial filter mask, the sum of products of corresponding pixel values, with every location in the target image f(x, y). coefficient values will be highest where the sub-image of reference image fits most over the target image location. Images are normalized by subtracting the mean pixel value and dividing by the standard deviation.

$$C(k,\ l)\ =\ \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(x,\ y)t(x-k,\ y-l) \tag{15}$$

$$\begin{aligned}&Where \quad -(P-1) \leq k \leq (M-1)\\ &\qquad\qquad -(Q-1) \leq l \leq (N-1) \quad template\ window\ size\\ &compute\ f(x,\ y)\ where\ C(k,\ l)\ is\ maximum\ for\ the\ given\ t(k,\ l).\end{aligned}$$

6.6.3 Tracking Strategy

6.6.3.1 Direction Search-

we have used least Square Estimation approach for finding the track of the target object which is a class of linear regression problem Invented by the French mathematician gauss, we have assumed the motion of the target object to be linear in nature that is why we have adopted this method for solving the tracking, the major advantage of this strategy least square estimation solution is its simplicity and faster processing in minimum time.

$$\begin{aligned}&compute\ m\ and\ C\ using\ Least\ Square\ Estimation\ regression\\ &for\ the\ given\ X1,\ Y1\ (Centroid\ of\ initial\ ROI)\ and\ X2,\ Y2\\ &(Centroid\ of\ Target\ detected\ using\ 2D\ Cross\ Correlation\\ &from\ initial\ ROI\ using\ Eq\ (1)\ )\end{aligned}$$

$$[Y1\ Y2]\ =\ m\,[X1\ X2]\ +\ C \tag{16}$$

$$\begin{aligned}&Now\ Calculate\ the\ City\ Block\ Distance\ between\ X1,\ X2\\ &and\ Y1,\ Y2\ \ as\ Xd = \ |X1-X2|\ and\ Yd\ =\ |Y1-Y2|\end{aligned} \tag{17}$$

$$as\ average\ displacement\ of\ object\ movement\ in\ every\ frame$$

$$\begin{aligned}&using\ Equation\ (2)\ and\ (3)\ Predict\ Xp,\ Yp\\ &Yp = mXd + C\end{aligned} \tag{18}$$

$$and\ Xp =\ (Yd - C)/m \tag{19}$$

$$as\ Prediction\ of\ center\ location\ target\ object\ in\ next\ frame$$

6.6.3.2 Neighborhood Search-

After Predicting the centroid of the candidate bounding box in the next frame, we need to search in the local neighborhood of 25 by 25-pixel grid area from the centroid to match with the target object again using 2D correlation with the previous computed target image. upon finding the successful and optimal match, we update the prediction with the computed target location and recursively go on computation of Least Square Estimation for the prediction of the next target object in the next frame and so on.

CHAPTER 7 MEASUREMENT METHODOLOGY

Evaluation of object tracking algorithms is quite a tough and difficult job as all algorithms are of different implementation and requires a pre- and post-processing of computed data for checking their performance and other parameters. Evaluation of tracking itself can be a difficult job as it requires to be operated upon different objects under different imaging conditions. Tracking also requires overcoming of different types of issues like rotation, , light conditions, occlusion, blurred images. To avoid issues, we require a very large dataset. Each sequence of images in the data has been included by taking all issues those are about to face and taken care by a good tracking algorithm. Tracking problems has been shown in table 1.

7.1 SUCCESS EVALUATION

For evaluation of success measurement of object in a frame, a formula can be used as shown in following line for each frame: $C\,suc(f) = |rt \cap rg| / |rt \cup rg|$, $Csuc(f)$ is the success parameter for frame f ; where rt is bounding box detected by tracker and rg is that the bounding box of evidence data. We simply took intersection and divide it by the union of two bounding boxes. If magnitude of this parameter is larger than 0.5 we count it as success of object detected in successive frame f. [12]

7.2 TIME COMPLEXITY EVALUATION

To measure time complexity of each algorithm, we simply measure time for each frame: $tim(f) = t$ where $Ctim(f)$ is the algorithm time required for completion; t is the time it took to process the current frame f.

7.3 DIFFICULTIES

To evaluate algorithms performance, we need to prune our data before and after processing so that we can remove leftovers which can lead to mis-calculation of performance parameters.
to find the amount of overlap of bounding box calculated by the tracker in respect of ground truth bounding box whereas also measurement the time interval,

the difficulty was that the tracker rejects the object from the ground truth once being initialized or simply looses the object in subsequent tracking. Once a trackers becomes unsuccessful in any frame of video it show unsuccess in all the subsequent video frames also since sequence of tracking looses track box information. We also had to manage these problems by reinitialization of trackers after a number of unsuccessful frames. we set reinitialization of tracker threshold at twenty-five.

Some other problems were also identified like detection of very large bounding box by tracker while actually object was of small size this we reclassified as failure of tracker, and recorded as unsuccess so that proper account of performance can be ensured.

CHAPTER 8 RESULTS

we have compared performance of different algorithms in this section in reference to standard benchmarked results of trackers, for this we have used bar graphs diagrams which compares parameters fairly well visually to understand trackers performance.

8.1 Results Of Object Detection

The result of object detection is shown in figure 12.

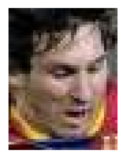

Figure 11 Sample ROI

Figure 12 Object Detected with 2D Cross-Correlation

8.2 Results Of Object Tracking

TABLE 3 COMPARATIVE QUANTITATIVE PERFORMANCE CHART BASED ON TRACKERS ACCURACY & SPEED

Trackers/ Parameters	MIL	Boost	MF	TLD	KCF	CSRT	NCC-LSE (5X5/ 25X25)
Accuracy	0.60	0.58	0.10	0.40	0.55	0.62	0.51/ 0.59
Time (sec)	0.075	0.020	0.012	0.090	0.050	0.052	0.025/ 0.041

Figure 13 Comparative Results of trackers accuracy (a)TLD (b) MIL (c) Proposed NCC-LSE

TABLE 4 TRADE-OFF BETWEEN ACCURACY, SPEED WITH WINDOW SIZES OF PROPOSED METHOD

Proposed Method Trade-off between Accuracy, Speed with Window Size		
	5X5 Search Window	***25X25 Search window***
Accuracy	0.51	0.59
Speed (FPS)	40	23.82
Time taken to process each frame (sec)	0.025	0.041

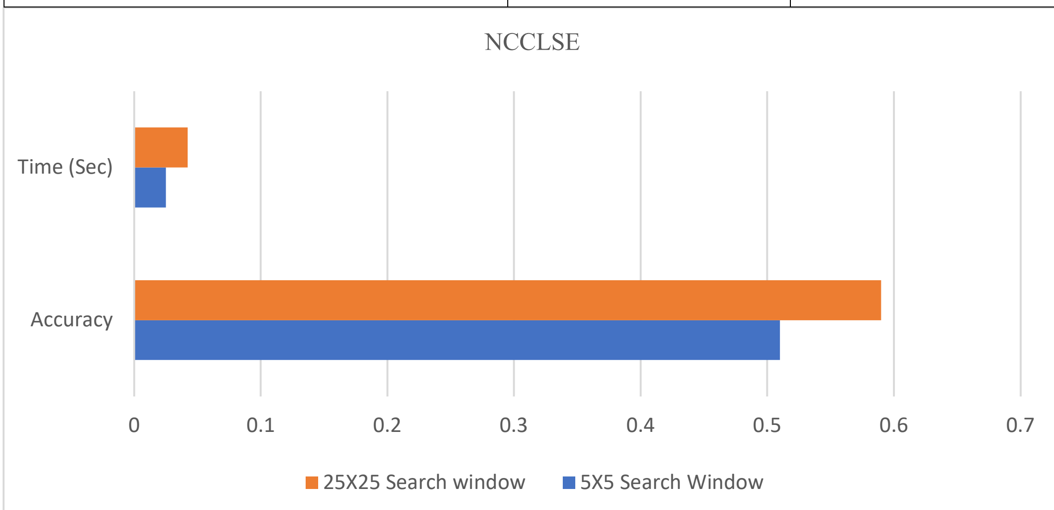

Figure 14 Proposed Methodology trade-off between Neighborhood Window Size over Accuracy & Speed (time in Seconds).

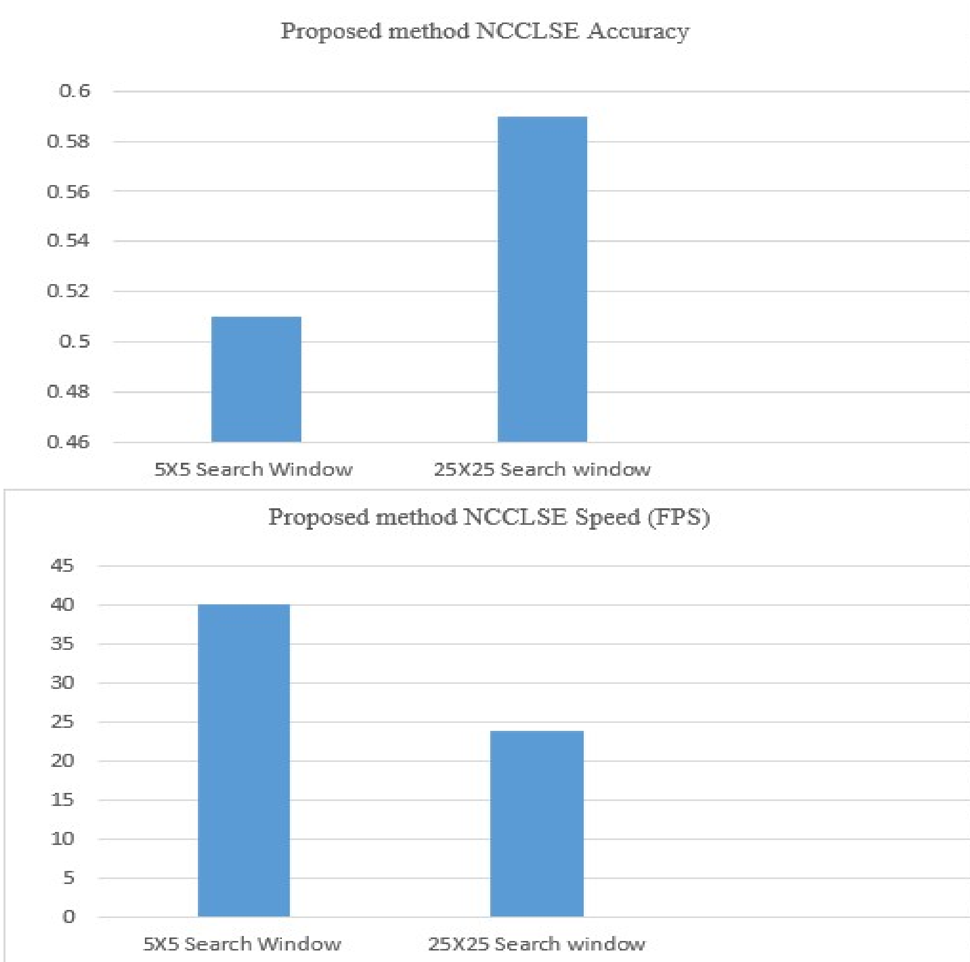

Figure 15 Proposed Methodology trade-off between Neighborhood Window Size over Accuracy & Speed (FPS).

SUCCESS

In following figure, we see the total success of different algorithms, which shows SURF and SIFT as the worst performer followed by TLD, which was not expected despite TLD being quite complex. MIL and BOOST shows satisfactory success parameters in total performance.

Scenario may be because of reinitialization from the ground truth data of other algorithms, and TLD was over self-correcting which makes detected object out of success criteria.

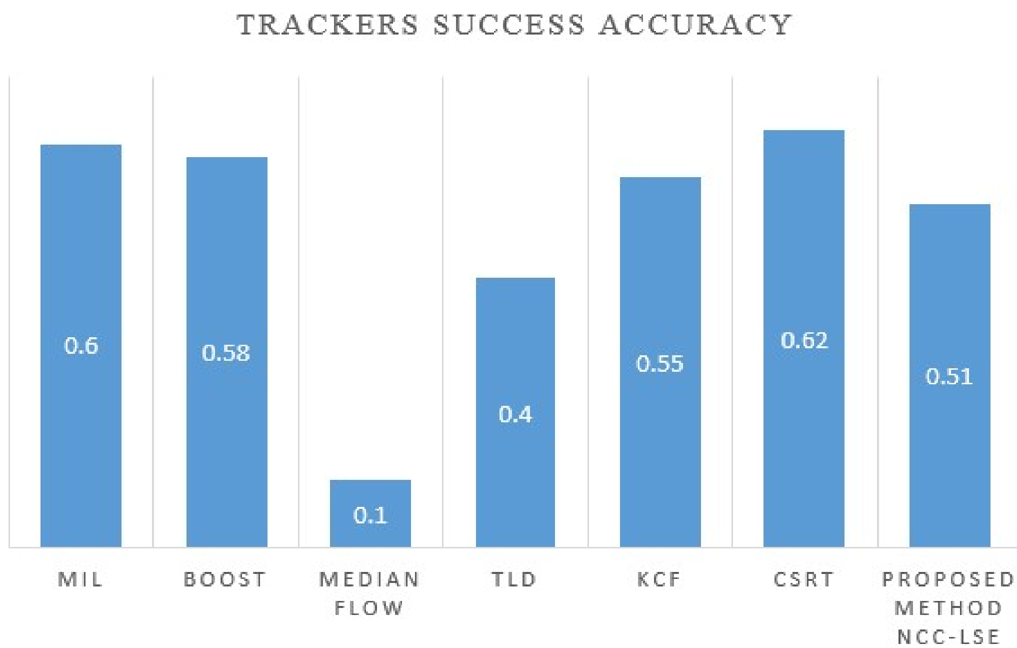

Figure 16 Comparative Trackers Accuracy.

TIME DEMANDS

Time complexity of these algorithms as can be seen from the following figure shows that SIFT and SURF are very slow, we measured the time required by each algorithm for processing a frame. Reason is that these algos utilize point values for their calculation while others only significant values, so other algorithms ORB, MIL, BOOST and MF are faster and the slowest one is TLD. TLD is slow because of complex structure and inefficient implementation in OpenCV framework.

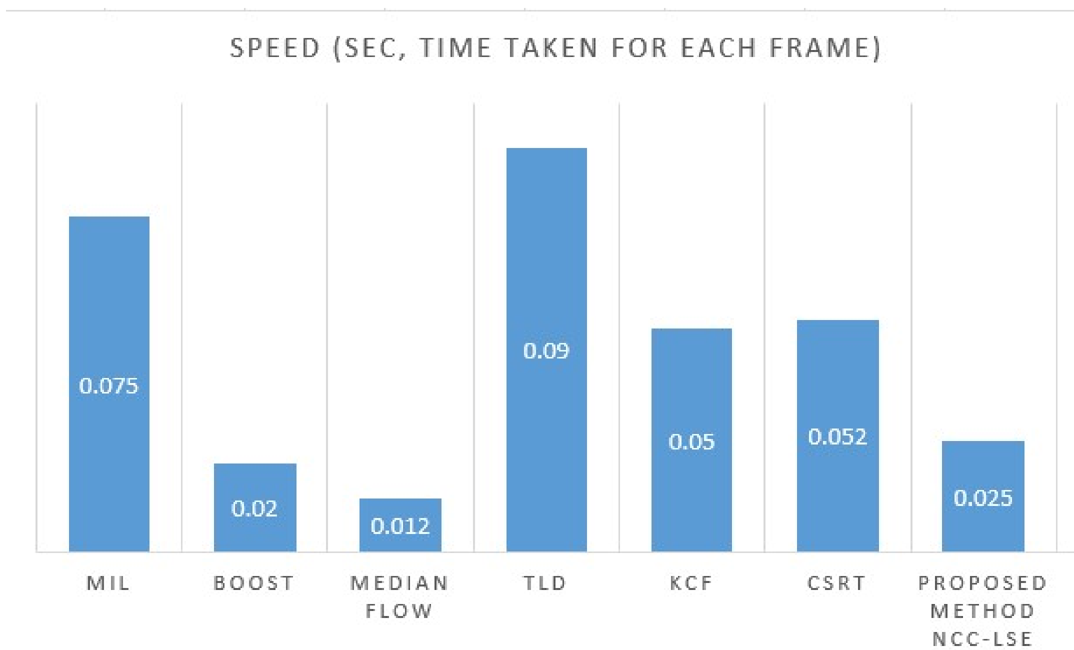

Figure 17 Comparative Trackers Speed (average time taken to process each frame).

OPTIMIZATIONS

In this project implementation of tracking algorithms basically are from OpenCV library only, so their speed of operation depends on algorithms structure and optimizations used in specific sub sections of internal computation.

In OpenCV optimization is achieved by parallelization of operations through use of virtual functions and classes.

OPTIMIZATIONS IN OPENCV

There are 2 ways of optimization in OpenCV, in first way a mathematical operation which can be broken into parallel operation are computed on parallel cores and is known as CPU parallel optimization in the second way computer use GPUs for taking graphical processing to its GPU controllers instead of CPU for computing mathematical operations.

CHAPTER 9 CONCLUSION AND RECOMMENDATION

In this project, we have proposed an integrated detection and tracking method based on fusion of normalized cross correlation with least square estimation technique. the experimental results shows that our proposed methodology is a faster algorithm in terms of speed at a comparable accuracy with other standard open CV tracking algorithms like boosting, median flow, MIL, KCF, CSRT etc.

Overall parameters like success, precision, time complexity and performance summarise the picture. Success criteria shows that TLD is not the best tracker as understood may be because of its inefficient implementation.

Precision factor shows all trackers more or less précised but a bit difference of min max range in few algos.

Time complexity figure shows SIFT, SURF as the slowest algorithms followed by TLD

The following points can be taken away from this project analysis:

MIL, BOOST, or MF for most robust applications:

MIL tracker can be used when higher accuracy with compromised FPS rate.

We can use BOOST tracker when we require faster FPS rate at the cost of compromising accuracy a bit.

Tracker MF can be used for requirement of pure speed only.

CHAPTER 10 REFERENCES

[1] Dr Rachna Verma, a review of object detection and tracking methods International Journal of Advance Research and Engineering Development Volume 4 issue 10 October 2017

[2] A Yilmaz, O Javed, M Shah,"object tracking: a survey", ACM computing surveys (CSUR), 38(4):13, 2006

[3] Feng Zhao, Qingming Huang, Wen Gao, Image Matching by Normalized Cross-Correlation, Acoustics, Speech, and Signal Processing, 1988. ICASSP-88,1988 International Conference on 2(2):II - II,June 2006, DOI:10.1109/ICASSP.2006.1660446, IEEE Xplore.

[4] Björck, Å. (1996). Numerical Methods for Least Squares Problems. SIAM. ISBN 978-0-89871-360-2.

[5] Chang Y H, Mang J. A Motion image detection method based on the inter-frame difference method[J]. Applied mechanics and materials, 2014, 490- 491: 1283- 1286.

[6] Decarlo D, Metaxas D, optical flow constraints on deformable models with application to face tracking[J], International Journal of computer vision, 2000, 38(2):99-127

[7] Piccardi M, background subtraction techniques: review[C]// IEEE international conference on systems, man and cybernetics, IEEE, 2005:3009-3104 vol.4.

[8] Guan H, Xue X Y, Zhi-Yong A N. advances on application of deep learning for video object tracking[J]. Acta automatica sinica, 2016.

[9] Viola P, jones M, Rapid object detection using a a boosted cascade of simple features[C]// computer vision and pattern recognition, 2001 an. Cvpr 2001. proceedings of the 2001 IEEE computer society conference on. IEEE, 2003:1-511-1-518 vol.1.

[10] Zhou H, Yuan Y, Shi C. object tracking using SIFT features and meanShift[J], computer vision and image understanding, 2009, 113 (3):345-352

[11] Miao Q, Wang G, Shi C, et. al. A new Framework for online object tracking based on SURF[J]. pattern recognition letters, 2011,32(13):1564-1571.

[12] Dalal N, Triggs B, histogram of oriented gradients for human detection [C]// computer vision and pattern recognition, 2005. cvpr 2005, IEEE computer society conference on. IEEE, on 2005:886-893.

[13] Barnich O, Droogenboreck M V. ViBE: a powerful random technique to estimate the background in video sequences[J]. 2009:945-948.

[14] Canny J, a computational approach to EDGE detection [J], IEEE transactions on pattern analysis and machine intelligence, 1986, PAMI-8(6):679-698.

[15] Bhattacharya A. on a measure of divergence between two statistical population Defined by their probability distribution [J]. bull CalcuttaSoc, 1943 ,35:99-109

[16] Fukunga K, Hostetler L D. the estimation of the gradient of a density function, with applications in pattern recognition[J], IEEE transactions on information theory, 1975, 21(1):32-40

[17] https://www.socscistatistics.com/tests/regression/default.aspx

[18] https://www.omnicalculator.com/math/least-squares-regression#least-squares-method

[19] https://www.omnicalculator.com/math/least-squares-regression

[20] https://www.intechopen.com/books/dynamic-data-assimilation-beating-the-uncertainties/estimation-for-motion-in-tracking-and-detection-objects-with-kalman-filter

[21] https://en.wikipedia.org/wiki/Simple_linear_regression

[22] https://www.allaboutcircuits.com/technical-articles/understanding-correlation/