A Comparison Between Two Object Tracking Methods For Videos

Zhengran Zhu
Electrical and Computer Engineering
Concordia University
email: barodaslife@gmail.com
40043285

Linxuan Chen
Electrical and Computer Engineering
Concordia University
linxuanchen2017@gmail.com
40042604

***Abstract*— In this project, we studied the object tracking problem in video signals and implemented method "Meanshift object tracking using gray scale(see reference [1])" and "object tracking using fractional-gain kalman filter(see reference [4])" separately. In section II and III the theory of two methods are given separately and section IV gives comprehensive analysis according to various aspects. At the very last, the comparison between two methods are given through numerous methods. From the results, we conclude that Kalman filter method gives better performance with object tracking while the Meanshift achieves no failures through most of the testing objects.**

I. INTRODUCTION

Visual object tracking is widely used in many fields nowadays. In the past decades, plenty of tracking algorithms have been invented and implemented in practice. Kalman filter based tracking and meanshift based tracking algorithms are representative works of object tracking methods. Meanshift algorithm was initially invented by Fukunaga and Hostetler in 1975, and extended by Comaniciu et al. This algorithm has advantages of being robust to not only rigid but also nonrigid objects, scale changes and partial occlusion, However, it has some fatal drawbacks like failure to track fast moving object or object under serious occlusion. In this report we are going to introduce a proposed tracker named meanshift(MS) tracker with grey prediction which can overcome these drawbacks. In Kalman filter, the state variables are estimated with given noise and given system parameters. The initial transient gain tends to a steady state value with time. Constant gain Kalman filter has been analysed since 1968. Constant gain Kalman filter works well for stand alone as well as data fusion mode for target tracking. Further adaptive gain Kalman filter was proposed to improve the performance of constant gain Kalman filter. The performance of Kalman filter depends on innovation process (i.e. difference between actual state variable and estimated state variable). In some critical cases, Kalman filter may not converge. So, for increasing the performance of Kalman filter, several researchers and engineers have tried to develop adaptive versions. A large number of publications have been reported for constant gain, but few have been reported for adaptive Kalman gain.

II. RELATED WORK

A. Other Methods

Object tracking can be defined as the problem of approximating the path of an object in the image plane as it moves around a scene. The purpose of an object tracking is to generate the route for an object above time by finding its position in every single frame of the video. Based on tracking techniques, one can classify object tracking into 3 categories: point tracking, kernel based tracking and silhouette based tracking. All the above methods rely on accurate object detection and object classification methods. In this project, kalman filter is a method of point tracking and meanshift belongs to kernel based trakcing methods.

B. Meanshift

Meanshift is an efficient pattern matching algorithm which is frequently utilized in object tracking. The original MS algorithm is an iteration to calculate the offset of current point and to move the current point to a new location according to the offset. In MS tracking algorithm, target and candidate object models are significant as well as similarity measurement. MS based trackers need users to define a target region in the first frame as a rectangle or an ellipse. The color histogram of target region is built as

$$\hat{q}_u = C\sum_{i=1}^{n} k(||x_i^*||)^2 \cdot \delta[b(x_i) - u] \tag{1}$$

where $x^*_{i\ i=1\ldots n}$ is the pixel location and $k(x)$ represents a kernel function which is epanechnikov profile in this case. $\delta(a-b)$ is nothing but kronecker delta function which means if a is equal to$b$, the output of it is one, otherwise its zero. C is a normalization constant to ensure that the summation of q_u is equal to one, and $b(x)$ is a function mapping pixel location to indices of color histogram. In the coming frames, color histogram of candidate region isdefined as:

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n} k(||\frac{y-x_i}{h}||^2) \cdot \delta[b(x_i) - u] \tag{2}$$

Where x_i indicates pixel location of candidate region which is centered at y. We use the same kernel profile with a scale parameter h which is equal to one in this paper. C_h is obviously still a normalization constant to make sure that the sum of p_u equals to one. The similarity measurement is defined as a distance between target and candidate region. In MS algorithm, Bhattacharyya coefficient is commonly used which is given as:

$$\hat{\rho_u}(y) = \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p_u}(y)\hat{q}_u} \tag{3}$$

The relationship between Bhattacharyya coefficient and distance is:

$$d(y) = \sqrt{1 - \rho[\hat{p}_y, \hat{q}]} \tag{4}$$

which means this coefficient is a number from zero to one, and the greater the coefficient is, the more similar between target and candidate object. Thus, it becomes an optimization problem. In order to maximize Bhattacharyya coefficient, Taylor expansion is used around the candidate point $\hat{p_u}(\hat{y_0})$,the result is shown below:

$$\rho[\hat{p}_y, \hat{q}] = \frac{1}{2}\sum_{u=1}^{m} \sqrt{\hat{p_u}(\hat{y_0})\hat{q}_u} + \frac{C_h}{2}\sum_{n_h}^{2} \omega_i k(||\frac{y - x_i}{h}||^2) \tag{5}$$

where:

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

From the Taylor expansion, only the second term is correlated with maximization. If we define $g(x) = -k(x)$, the new location is defined as:

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i \omega_i g(||\frac{\hat{y}_1}{h}||^2)}{\sum_{i=1}^{n_h} w_i g(||\frac{\hat{y}_1 - x_i}{h}||^2)} \tag{7}$$

From the discussion above, the MS trackers have advantages of less computation as well as robustness of not only rigid but also nonrigid object. Nevertheless, there are some drawbacks of this algorithm, for instance it can not track effectively when occlusion occurs which will be overcame by grey prediction.

C. Kalman filter

Within the research area of object tracking, Kalman filter is a widely used as an optimal state estimator for the system. In this subsection we give interpretations on what a conventional kalman filter is.

For a conventional Kalman Filter, it has two steps, first step is to acquire priori estimation using system equation:

$$x_k^- = A\hat{x}_{k-1} + Bu_k \tag{8}$$

Then the variance of previous estimation, which is the uncertainty in estimated states is calculated:

$$P_k^- = AP_{k-1}A^T + Q \tag{9}$$

Then the algorithm goes into the next step witch is the estimation of posteriori states. In this step, Kalman filter gain is calculated in such a way that it balances up between measurement noise and uncertainty of prior estimation so that the posteriori error is minimized:

$$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R} \tag{10}$$

where C is the output matrix of the system and R is measurement noise. And posteriori estimation can be updated using priori estimates and error between measurement and predictive measurement:

$$x_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_{k-1}^-) \tag{11}$$

From the above equation, we see that if measurements noise R equals to 0, K_k will be equal to C^- and the posteriori estimation x_k will equal to Cy_k since there is no measurement noise. At the very last, the variance of posteriori estimation is:

$$P_k = (I - k_k C)P_k^- \tag{12}$$

Then the algorithm goes to step one to repeat the estimation.

III. THEORY

A. Meanshift Using Grayscale

GM("1","1") is a basic model of grey prediction, and the first 1 represents first order differential equation and the second 1 refers to one variable. Let $Y^{(0)} = y^{(0)}(1), y^{(0)}(2),, y^{(0)}(n)$ be an initial sequence. To predict the new location, $y^{(0)}(n+1)$, following steps are required.

Step 1: A new sequence $Y^{(1)} = y^{(1)}(1), y^{(1)}(2),, y^{(1)}(n)$ is generated by accumulated generating operation(AGO), where:

$$y^{(1)}(k) = \sum_{i=1}^{k} y^{(0)}(k), k = 1, ..., n \tag{13}$$

Step 2: Another sequence is generated by the adjacent neighbor means of $Y^{(1)}$ as $Z^{(1)} = z^{(1)}(2), z^{(1)}(3),, z^{(1)}(n)$ where $z^{(1)}(k) = (1/2)[y^{(1)}(k) + y(k1)], k = 2, ..., n$.

Step 3:

The grey prediction equation is $Y^{(0)}(k) + az^{(1)}(k) = b$ where a is named as the development coefficient while b is the grey action quantity.Let $\hat{u} = [a\ b]^T$ be the parameters sequence, then the least-square estimate parameters sequence of the grey differential equation is $\hat{u} = (B^T B)^{-1} B^T H$ where $H = [y^{(0)}(2)...y^{(0)}(n)]^T$ and $B = [-z^{(1)}(2)\ 1; ...; -z^{(1)}(n)\ 1]^T$

Step 4: The corresponding written equation is

$$\frac{dy^{(1)}}{dt} + ay_{(1)} = b \tag{14}$$

and by solving this equation, the AGO grey sequence can be obtained as follows:

$$\hat{y}^{(1)}(k+1) = [y^{(0)}(1) - \frac{b}{a}]e^{-ak} + \frac{b}{a} \tag{15}$$

Step 5: $\hat{y}_{(1)}(k+1)$ is finally calculated as follows:

$$\hat{y}^{(1)}(k+1) = (1 - e^a)[y^{(1)}(1) - \frac{b}{a}]e^{-ak} \tag{16}$$

n this paper we using previous four frames to predict the new location, and Maclaurin formula is used in this paper to compute [3]. Thus the final GM(1,1) equation is expressed as:

$$\hat{y}^{(0)}(5) = [\beta - ay^{(0)}(1)]\left[1 - 3a + \frac{(3a)^2}{2!} - \frac{(3a)^3}{3!}\right] \quad (17)$$

B. Fractional-Gain Kalman Filter

From the provided material, we know that the fractional difference defined by Grunwald-Letnikov is given as:

$$\Delta^a x_k = \frac{1}{n^\alpha}\sum_{j=0}^{k}(-1)^j \binom{\alpha}{j} x_{k-j} \quad (18)$$

where α is fractional order, h is sampling interval, k is number of samples of given signal x and $\binom{\alpha}{j}$ can be written as γ_j and is defined as:

$$\binom{\alpha}{j} = \begin{cases} 1, & \text{for} j = 0 \\ \frac{\alpha(\alpha-1)\ldots(\alpha-j+1)}{j!}, & \text{for} j > 0 \end{cases} \quad (19)$$

Similarly we have the kalman filter:

$$x_k = ax_{k-1} + bu_k + w_k \quad (20)$$

and

$$z_k = hx_k + v_k \quad (21)$$

where w_k is system noise and v_k is output noise at time instant k, a is transition matrix, h is measurement matrix. The difference between estimated output $\hat{z}_k$ is known as innovation and is given as:

$$innovation = z_k - \hat{z}_k = z_k - h\hat{x}_k^- \quad (22)$$

The posteriori estimated state $\hat{x}_k$ is given as:

$$\hat{x}_k = \hat{x}_k^- + K_{new}(z_k - h\hat{x}_k^-) \quad (23)$$

where

$$K_{new} = K_k + f_k = K_k + \Delta^\alpha K_k \quad (24)$$

where $\Delta^\alpha K_k$ is calman gain and is calculated by minimize posteriori error covariance P_k:

$$P_k = E(x_k - \hat{x}_k)^2 = E(x_k - \hat{x}_k^- - (K + \Delta^\alpha K)(z_k - h\hat{x}_k^-))^2 \quad (25)$$

To find value of gain K:

$$\frac{dE(x_k - \hat{x}_k^- - (K + \Delta^\alpha K)(z_k - h\hat{x}_k^-))^2}{dK} = 0 \quad (26)$$

K_new can be written as below:

$$K_{new} = K + E\sum_{j=0}^{k}(-1)^{j+1}\binom{\alpha}{j}K_{k-j} \quad (27)$$

To summerize, the modified Kalman gain contains two terms, The first term represent the normal kalman filter gain, while the other stans for fractional difference of previous values of Kalman gain. Fractional derivative of previous Kalman gain incorporates the variations of input signal with time. This helps in reducing the settling time of proposed FOGKF(Fractional-Gain Kalman Filter) during the abrupt variations.

IV. Results

A. Experimental Setup

In this project, we first give results separately for each methods. Then a comparison is given over RMSE, running time and video sources. For the simulation of Fractional-Gain Kalman Filter for vehicle tracking, we use background extraction method, which average through all the frames in the video to extract backgrounds. Consequently, the moving objects in each frame is considered as noises, and thus we can retrieve the position of each object. As being said above, the paper for Fractional-Gain kalman filter specifies in vehicle tracking, which means the background extraction may not be as accurate as many other object detection methods.

B. Meanshift Using Grayscale

Algorithm:

initialize: Calculate the object model *qu*and initialize the location y^0.

Procedure:

(1)Input the coming frames and the frame number is n.

If $(n < 4)$

Calculate the candidate model, and y is the previous location.

Else

Predict the location yp, and replace the parameter y above.

(2)Do meanshift iterations, and find the current location y1.

(3)Compute Bhattacharyya coefficient (y1) between target and candidates, and compare the coefficient with threshold T to determine if occlusion happens.

If $\rho(y1) < \rho T)$

Replace the computed location y1 by the predicted location yp. (4)Go back to step 1 in procedure.

Output: Locations of the object in all frames.

In order to test the performance of proposed tracker under serious occlusion, we select frames from frame number 417 to 500 which contain three different scenarios. The target object is a girls face.

We extract the results of original MS tracker and MS tracker with grey prediction applied on the same frames which is frame 438, frame 463 and frame 478. The results with blue rectangle are obtained by traditional MS tracker, and the red one is generated by proposed tracker. As we can see here, the traditional MS tracker is robust to partial occlusion and generates almost the same results as proposed tracker does. However, when serious occlusion happens, the traditional MS tracker feedbacks the males face as the result which is obviously incorrect, while MS tracker with grey prediction is

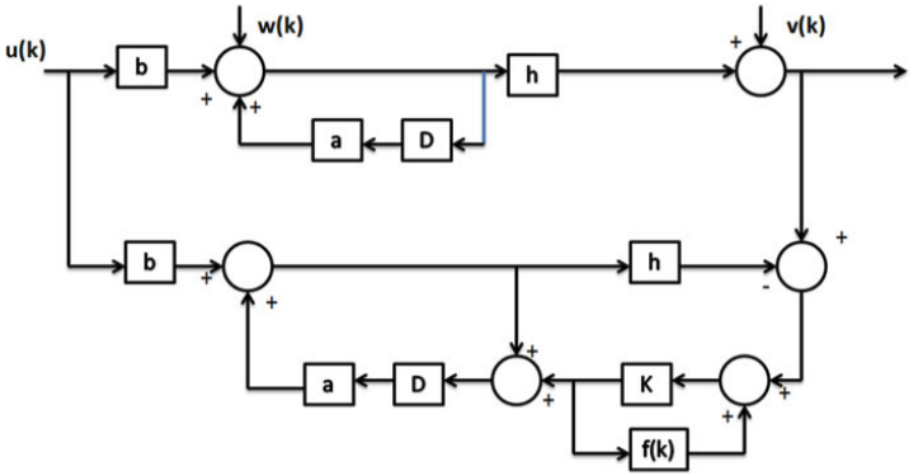

Fig. 1. Block diagram of Fractional-Gain Kalman Filter

still capable to track target object effectively with slight drift. The reason behind this phenomenon is that MS trackers are based on color histogram, when serious occlusion happens, the target object is vanished and the most similar region to be found by comparing color histogram is the males face. Although the Bhattcharyya coefficient between target and result patch generated by traditional MS algorithm is not acceptable, original tracker has no choice but to consider the most similar part as its result. Meanwhile new tracker calculates the Bhattcharyya coefficient and compares it with threshold to determine whether replacing results from traditional MS algorithm by grey predicted locations, and that is the reason why new tracker can overcome serious occlusion problem.

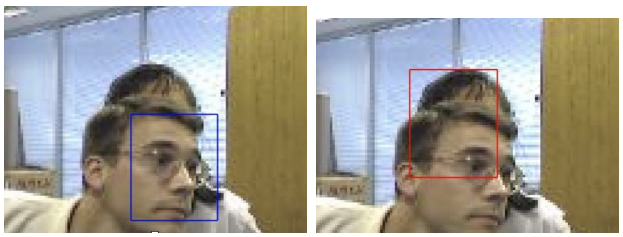

Figure 2. Results Under Serious Occlusion

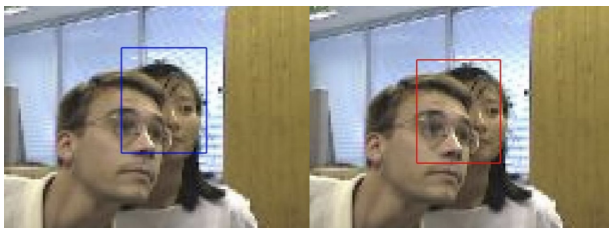

Figure 3. Results Under Partial Occlusion

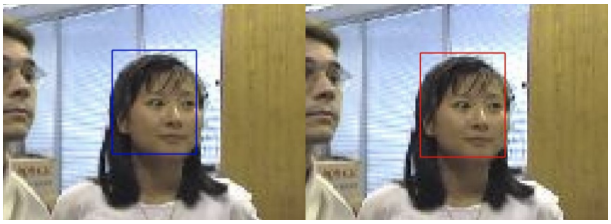

Figure 4. Results Under No occlusion

In terms of running time, as mentioned before that traditional MS tracker is only close to real time trackers which means it fails tracking fast moving objects. Although in the paper they mention that the improvement of running time is just a happen coincidence, MS tracker with grey prediction does improve the running time. The figure 5 below is running time comparison between original and proposed trackers. It is obvious that the computation time of proposed tracker is much less than that of traditional tracker, and the average computation time on each frame of proposed tracker is even less than 0.02 second per frame which means the proposed track is almost a real time tracker.

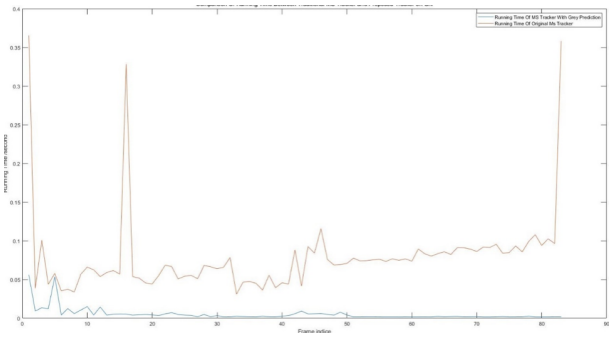

Figure.5 Running Time Comparison(orange one is traditional tracker and blue one is proposed tracker)

RMSE of distance between predicted location by two trackers and ground truth are calculated and plotted in this section. Occlusion happens from frame 10 to frame 25,

and as we can see in the figure 6, under serious occlusion, traditional tracker fails and predicted locations obtained is far away from the true locations while proposed tracker works smoothly.

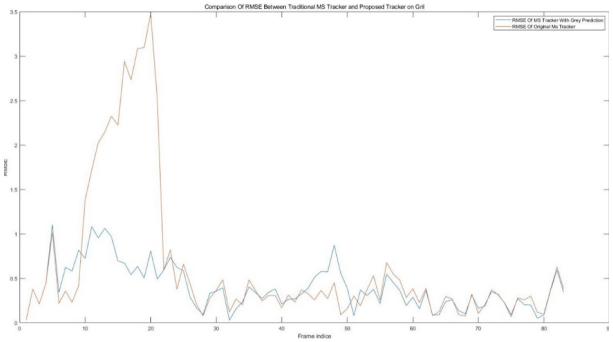

Figure.6 RMSE of Two Trackers (Orange one is traditional tracker and blue one is proposed tracker)

Another example is shown in below: Tests on video sequences Dog from Visual Tracker Benchmark with threshold=0.3. The figures below are results of applying two trackers on a video without occlusion, the performance is very similar to each other because basically when there is no occlusion, they use very similar algorithm.

Figure 7. Results Of Frame 30

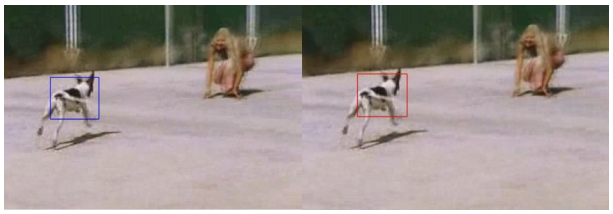

Figure 8. Results of Frame 45

Figure 9. Results of Frame 90

According to figure 10 the proposed tracker still has much better computation time than traditional MS tracker.

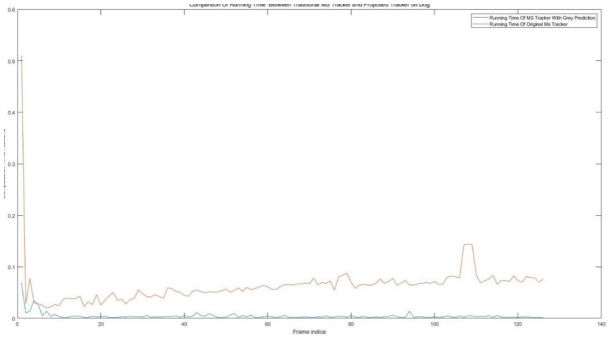

Figure 10. Computation Time of Two Trackers (Orange one is traditional tracker and blue one is proposed tracker)

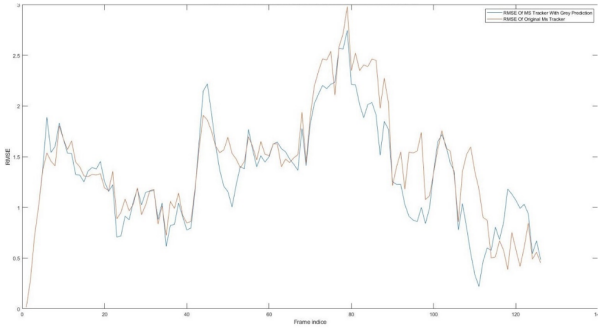

Figure 11. RMSE of Two Trackers on Dog (Orange one is traditional tracker and blue one is proposed tracker)

C. Fractional-Gain Kalman Filter

In this paper, the author used background retraction method to detect the objects(vehicles) in the first place, and then apply Fractional-gain Kalman filter to keep tracking these objects. Background retraction can be represented as below:

$$B(i,j) = median\left\{I(i,j,1), I(i,j,2), ..., I(i,j,N)\right\} \quad (28)$$

Where $B(i,j)$ is background image, N is the size of buffer and $I(i,j,N)$ is image sequence.

Consider the motion model that the tracking object is moving with a constant velocity and in a straight line. The transition matrix M at time N is given by:

$$M(k+N) = \begin{bmatrix} 1 & N & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & N \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

and state vector is described by:

$$x(k) = \begin{pmatrix} x & v_k & y & v_y \end{pmatrix}^T \quad (30)$$

where x and y are the position coordinates of the vehicle in the image at time k, v_x and v_y are the velocities of the vehicle at the correspondant position. For a stationary input(can be treated as a step input). The output is as follows:

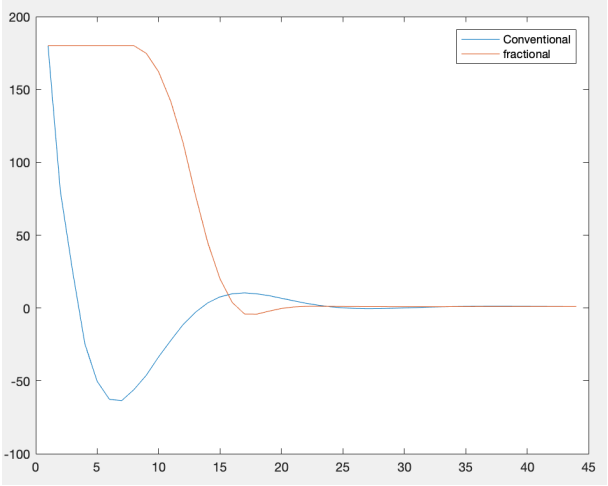

Figure.12 Output of stationary input.

We see that from the result, the Fractional-Gain Kalman Filter has better settling time and peak value which accords to our anticipation. For an incrementing input (similar to ramp input), we have the following result:

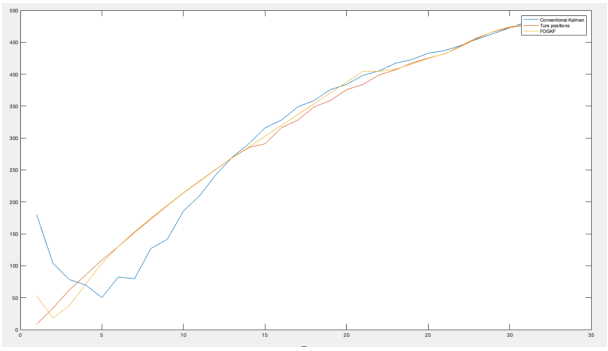

Figure.13 Position tracking

As shown in the figure. The Fractional-Gain Kalman Filter performs better over both accuracy and response
Last of all we give the result for real position tracking for conventional kalman filter:

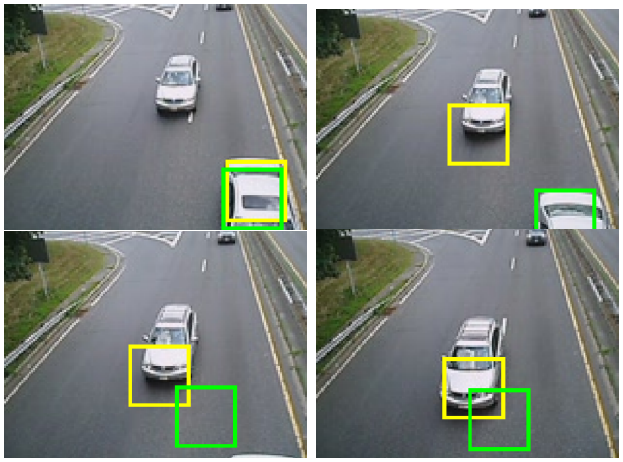

Figure.14 Object Tracking For Video Using Kalman Filter

The yellow rectangle represents for the true position of the vehicle while the green one for the estimated position by Kalman Filter. Comparing to conventional kalman filter. We have following results for Fractional-Gain Kalman Filter at the same frame number

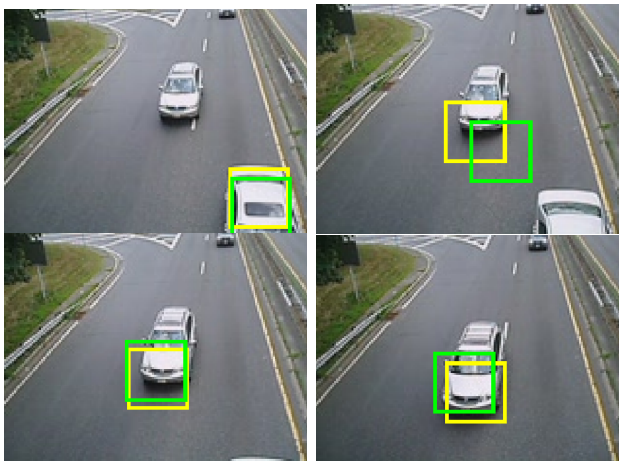

Figure.15 Object Tracking For Video Using Fractional-Gain Kalman Filter

This time, the green rectangle represents for the Fractional-Gain Kalman Filter. From the aboving comparison, we see that Fractional-Gain Kalman Filter give better tracking results. However the conventional Kalman filter barely keeps tracking in this video object. The Fractional-Gain Kalman Filter is said to be at least 17% better then conventional Kalman Filter in this paper.
To compare two Kalman filter with the Fractional Kalman filter statistically we use RMSE and running time as a reference.

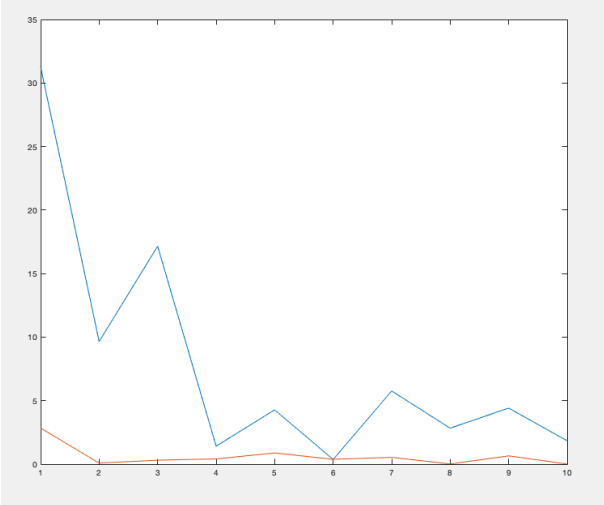

Figure.16 RMSE of the true position and estimated position(Orange for fractional)

The RMSE is the mean square error between true position(in this case we use background extraction) and estimated position, it can be defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} e_j^2} \tag{31}$$

As shown in fig.16 organge line which is fractional-gain

kalman filter performs better tracking results. The running time of two compared methods is given below:

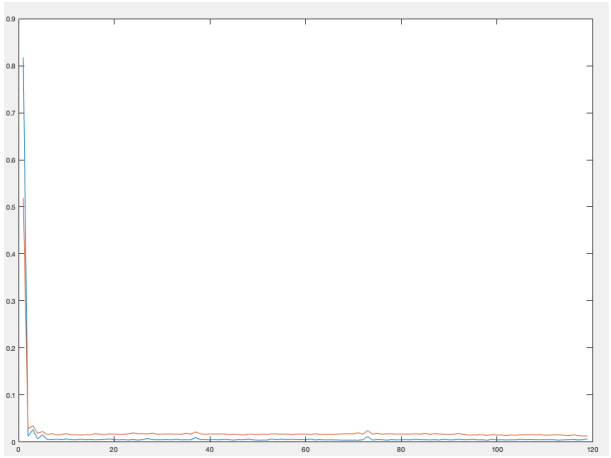

Figure.17 Runing time comparison between conventional and proposed Kalman filter(Orange for Fractional-Gain Kalman)

We see that fractional-gain Kalman filter requires more time to process since it has a fractional term in its gain. To Conclude, the Fractional-Gain Kalman Filter maintains better performance on noise soothing and responsiveness due to its adjustment ability using previous gains.

D. Results Comparison Between Two Object Tracking Methods

In order to compare MS tracker with grey prediction and tracker based on fractional-gain Kalman filter. We apply them on two video sequences at the same time and use RMSE of each frame as well as running time to quantize the performance of trackers. The first video is girl, and the second video is David3. Both are from visual tracking benchmark. Due to that the initial location of Kalman tracker is random value, we drop the first 9 frames to avoid extremely high RMSE generated by Kalman tracker. In video sequence girl, it is obvious that MS tracker generates locations which are closer to real location, and MS tracker performs more stable. However, we make an assumption that the performance of Kalman tracker can be improved by increasing precision of its embedded object detector. Theoretically speaking, if object detection algorithm embedded is accurate enough, the green rectangle which represents results of detector should be very close to ground truth. So, in video David3, we assume that Kalman tracker utilizes an accurate detector and we use ground truth directly to evaluate the tracking performance of Kalman filter. The results are shown in Figure 19, as we expected, Kalman tracker actually has batter prediction precision because it calculate and predict the location directly using some prediction algorithm while MS tracker uses similarity and threshold to find the most similar patch in the following frames. Figure 20 demonstrate the running time comparison of two trackers on two video sequences. The computation time of MS tracker with gray prediction is lower than Kalman tracker in both sequences, which means MS tracker is more real time.

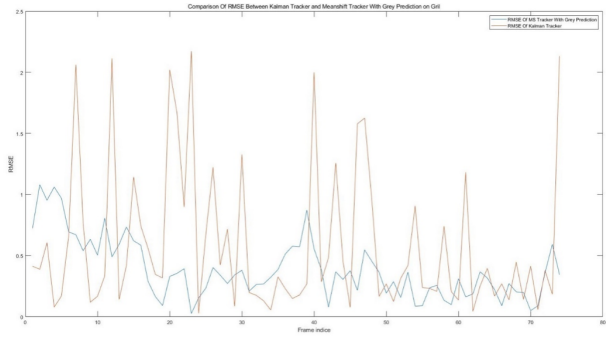

Figure 18. RMSE of Two Trackers on Girl (Orange one is Kalman tracker and blue one is MS tracker with grey prediction)

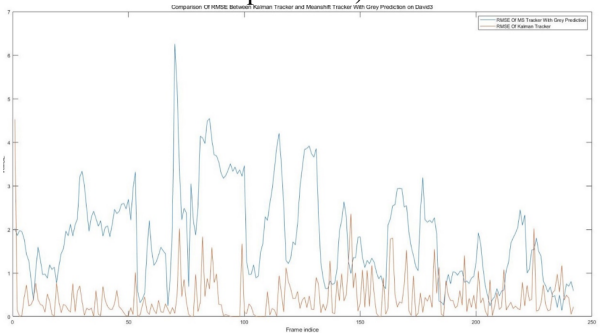

Figure 19. RMSE of Two Trackers on David3(Orange one is Kalman tracker and blue one is MS tracker with grey prediction)

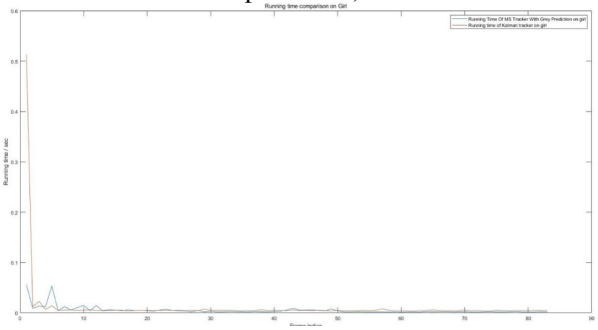

Figure 20. Running time of two trackers on Girl(Orange one is Kalman tracker and blue one is MS tracker with grey prediction)

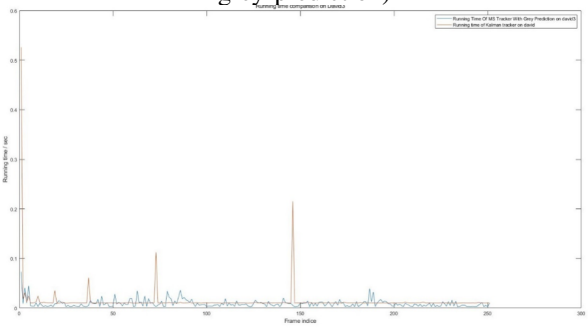

Figure 21. Running time of two trackers(Orange one is Kalman tracker and blue one is MS tracker with grey prediction)

As stated before, in this project, Kalman filter method relies greatly on object detection method to retrive the true position of the object. In some of the video sequences, the motion is not obvious or there is camera motion occurring, it hinders the background retraction method to detect the true position of the objects. Here, we show an example of the failure of object detection using background retraction method:

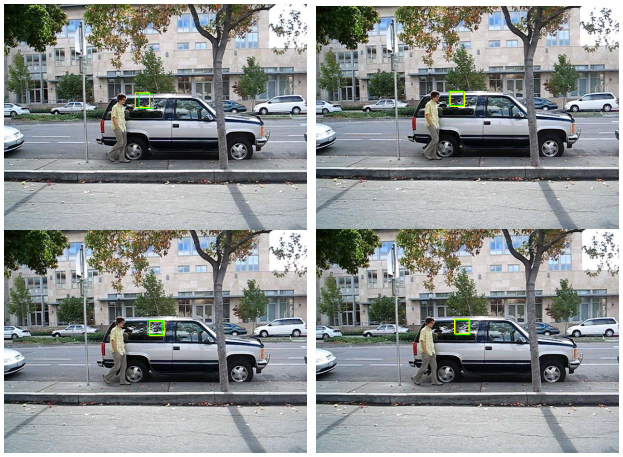

fig.21 Example of the failure of background retraction.

V. CONCLUSIONS

MS tracker with grey prediction uses predicted locations to replace error locations generated by traditional MS algorithm. The advantages of proposed tracker are lower computation time and overcoming drawbacks of traditional MS tracker. However, the threshold of determine if occlusion happens is varies from object to object, so we need to manually change it in different scenarios. Kalman tracker is mainly designed for traffic tracking, the precision of it is affected by object detector embedded. If the detection algorithm is accurate enough, it has better tracking performance. The drawbacks of Kalman tracker is the same as them of detection algorithm used which means if the detector fails, Kalman tracker fails. In conclusion, MS tracker has lower computation time and wider range of applications while Kalman tracker has better tracking performance.

VI. GROUP CONTRIBUTION

Zhengran Zhu:
Fractional-Gain Kalman Filter for vehicle tracking
Linxuan Chen:
Meanshift using grayscale for object tracking

VII. DEMO LINKS

For those may be interested: All the results(videos) are available online:
Applying traditional MS tracker on Dog :
https://youtu.be/GXmx5y0FO4I
Applying proposed MS tracker on Dog :
https://youtu.be/lJJsLYivtd8
Applying traditional MS tracker on Girl :
https://youtu.be/2x5JPLnKnyI
Applying proposed MS tracker on Girl :
https://youtu.be/EI4uV9o-B14
Applying proposed MS tracker on David3 :
https://youtu.be/-m6QX0OxncM
Applying Kalman Tracker on Girl:
https://youtu.be/f0XSjU1XAUI
Applying Kalman Tracker on David3:
http://youtu.be/zLXplfBnrSs

REFERENCES

[1] Mingming Lv , Li Wang, Yuanlong Hou, Qiang Gao, and Runmin Hou, FoMean Shift Tracker With Grey Prediction for Visual Object Tracking, hCanadian Journal of Electrical and Computer Engineering, Fall 2018, vol.41, No.4, pp.172-178

[2] GKalyan Kumar Hati, andA Vishnu Vardhanan, Review and Improvement Areas of Mean Shift Tracking Algorithm, The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), pp.1-2

[3] K. Fukunaga and L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans.Inf. Theory, vol. IT-21, no. 1, pp. 3240, Jan. 1975.

[4] HHarpreet Kaur and J. S. Sahambi, Member, IEEE Cehicle Tracking in Video Using Fractional Feedback Kalman Filter,IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, VOL. 2, NO. 4, DECEMBER 2016.

[5] Nazia Aslam, and Veena Sharma, KForeground Detection of Moving Object Using Gaussian Mixture Model,International Conference on Communication and Signal Processing, April 6-8, 2017, India

[6] Test video from http://cvlab.hanyang.ac.kr/tracker/benchmark/datasets.html

[7] https://github.com/skhobahi/Kalman-Filter-Object-Tracking