Santa Clara University
Research Report

22-February-2021

Using a Kalman filter tracker in a novel way to improve object tracking accuracy in video sequences subject to noise and occlusion impediments

by
Mark Heimbach

Research Report
Department of Electrical Engineering
Santa Clara University

Table of Contents

List of Figures

Abstract

Real-time object tracking systems must be capable of tracking an object subject to abrupt changes in motion and appearance. Model uncertainty is inherent in the tracking process where the only information available may be initial object appearance and position. The constraint for real-time tracking applications reduces the applicability of online detect and tracking methods such as CNN. The model may also suffer from variations due to impediments such as physical occlusions, object deformations, resolution changes, rotations and random noise. These noise impediments can lead to inaccurate measurements and ultimately loss of track. To maintain frame-to-frame tracking, a Kalman filter is proposed. Kalman filters have been employed extensively for tracking, but lose accuracy (or become unstable) due to mismatch between a priori noise estimates and measurements. However, developing appropriate model parameters for the Kalman filter is a challenge without explicit knowledge of the tracked object and the environment in which it is observed. A cost function is therefore designed to minimize this discrepancy by adaptively updating the Kalman model based on a confidence function derived from the feature descriptor. To demonstrate the approach, the Histogram of Oriented Gradients (HOG) is used as the descriptor to profile the object according to its distribution of gradients. This method is extensible to other feature descriptors where a quantitative analysis can be made for measured object detection and position. By constantly updating the Kalman model based on relative measurement confidence, the tracking performance is improved with more accurate position estimates when compared with Kalman filters with invariant model parameters. Online video datasets with ground truth are used to validate the effectiveness of the proposed algorithm.

Introduction

Realtime object tracking is an important component in many computer vision applications such as driver assistance, traffic monitoring [1],[2], surveillance [3],[4], robotics, human-computer interaction [5], smart homes [6] and medical imaging [7]. Object detection and tracking involves determination of position and size of the target object in each frame of a video sequence. Difficulties in tracking include changes in object appearance, background variations, occlusions, rotation, scale and impulse noise [8]. Occlusions in particular have been heavily researched due to their common occurrence in many tracking applications [9]. Re-acquisition is also problematic once the object position is lost. In most cases, system requirements are not well defined due to a lack of prior knowledge.

Given the initial state of the target object (position and size) in the first video frame, the goal is to accurately estimate states of the target in subsequent frames. The camera may be either stationary or non-stationary. To model the object, a feature detector is used. For HOG, features are described with a gradient vector. This vector is then used in subsequent frames to scan a detection window (patch) searching for the position providing the highest correlation

with the reference vector in a maximum likelihood sense. In this process the gradient vector senses the environmental information not only for the object of interest, but also for other objects and the general background throughout the detection window. This is important when trackers can be distracted by regions containing objects of similar appearance to the target object. A robust application must therefore sense activity surrounding the target and be invariant to constant background changes due to a moving camera and a target in motion [10].

Information obtained by the HOG measurement response is used to inform a Kalman filter tracker about abrupt changes in position, occlusions, deformations and the probability that the measured object position is in fact correct. This information is used to adaptively update the Kalman model estimation.

Simulations are conducted showing the impact on tracking accuracy when the Kalman model diverges from object motion predictions. Accuracy here is defined as the Euclidian distance between object position estimates and ground truth. An in-depth analysis is then performed showing the degree to which object detection accuracy varies as a function of feature detector measurement confidence.

Our method of adaptively updating the Kalman model estimates is tested experimentally using a fully annotated online benchmark video dataset library VOT14 [12] . This collection includes fully-annotated videos with noise impediments including low resolution (LR), illumination variation (IV), occlusion (OCC), deformation (DEF), in-pane rotation (IPR), out-of-plane rotation (OPR), background clutter (BC), motion blur (MB) and fast motion (FM). Ground truth coordinates are also provided for all frames in each of the videos.

Related Work

In [13], a robust Kalman filter is introduced by multiplying the Kalman gain by a constant derived from previous tracking events. The constant is used when the correction departs from the model prediction. However, this approach is limited by proper setting of the constant when noise is present. Heuristic approaches have also been used for determining Kalman gain [14] and [15] but suffer when future events are independent of past experience. The tuning and performance of these estimators are often obtained from simulation or empirical results [16]. However, these methods require a priori knowledge of state measurement and process models [17].

The use of accelerometer sensors have been introduced to define Kalman model parameters such as gain in order to minimize model error [18]. Other sensors have been employed using additional Kalman filters to minimize sensor perturbations, using the output of one KF into an Extended Kalman Filter for more precise model estimations [19],[20],[21].

Support Vector Machines have been used for supervised learning classification. However, SVMs cannot be used for objects which may be unknown until the tracking process begins.

SVMs also require extensive labelling and offline training, making them difficult to use for real-time applications.

Self-training algorithms have been introduced which first use a trained classifier to discriminate between object and background [22], [23], [24]. The classifier in then updated over time to account for object appearance changes. However, when errors occur in the updating process, the errors may be propagated thus causing the process to become unstable. This error drifting may then eventually cause loss of track. Self-training is also subject to outlier errors.

For nonlinear systems, various modifications of the Kalman filter have been proposed. The Extended Kalman filter (EKF) is a nonlinear approximation of the Kalman filter that assumes small estimation errors and approximates them to calculate its covariance matrix. EKF is vulnerable to linearization errors, which can cause not only poor performance but also divergence [25], [26],[27]. The EKF is also difficult to evaluate in a complex system [28]. The Unscented Kalman filter (UKF) performs better than EKF in many situations, but is not applicable for real-time tracking due to its computational complexity (cubic in the size of the state vector) [29].

Our methods for tracking overcome the problems in the various approaches outlined in this section. The confidence in how well the object is tracked is derived from the HOG analysis. This information is available before the Kalman filter makes its object location estimate in each video frame. This overcomes the problem with modifying Kalman gain based only on prior history, allowing improved tracking when objects make unanticipated position changes.

The importance of a priori knowledge of the object and its environment is also reduced with our method. In many real-time situations, the Kalman filter may quickly become unstable when the model diverges from the actual environment. A real-time adaptive method is used here to overcome this problem by constantly updating the filter according to actual measurements by the discriminator. The HOG response in essence provides the Kalman filter with an improved estimate of the environment.

Our method allows for real-time tracking of an object presented in the first frame. This overcomes the problem inherent with machine learning algorithms that use offline or a combination of offline/online learning algorithms to develop a model for the object to be tracked.

Methods of Classification – A Brief Description for Real-time Analysis

The ability to successfully track an object is contingent on how well the object differs from its background. With objects whose appearance is similar to the background, a successful track may require prior knowledge of expected motion. Use of online adaptive feature selection has

been studied whereby features used are selected based on what best distinguishes between object and background [30] .

Unfortunately, object appearance changes and movements are difficult to model and may be highly non-linear. Various methods have been used to overcome appearance changes including Eigenfiltering [31] and Condensation [32]. These are generative trackers that use object information only, ignoring the background.

A large variety of descriptors have been described for object modelling [33],[34],[35]. Robust classifiers such as SIFT (Shape Invariant Feature Transform), SURF (Speeded Up Robust Features) , Harris Corners and HOG (Histogram Oriented Gradients) are a few descriptors that have been successfully used for object recognition and tracking.

Both SIFT [36] and SURF [37] detect interest points and define a method for creating an invariant descriptor. The descriptors match these interest points, comparing a large number of features in images whose objects may be subject to scale changes, rotation and illumination. Keypoints (or local features) are selected based on regions showing larger gradients. These include corners, edges and intersecting lines where brightness changes are apparent. Harris Corners uses feature descriptor methods based on edge and corner detection [38].

Various improvements to the SIFT descriptor include shifting Gradient Location-Orientation Histogram (sGLOH) [39] ,shape context [40] and Space-Time Interest Points (STIP) [41]. Improvements to the Harris detector include GFTT [42], which is often used in tracking applications [43].

The probability of a correct match between an object in two images is determined given the accuracy of the fit, number of feature points and an assessment of the possibilities for false matches. Ability to probe the relative accuracy of a match makes these detectors a good fit for the innovation for creating confidence functions for adaptive Kalman model updates.

HOG is used in this report to illustrate advantages in using detection confidence to inform a Kalman filter for object tracking. However, this innovation is extensible to other feature description methods as outlined above. A key determinant is the ability to quantify the probability that an object match is correct.

Histogram of Oriented Gradients (HOG) is a feature detector used predominately for object characterization. HOG is similar to SIFT, edge detectors and shape contexts; however, HOG differs in that histograms are normalized and overlap a detection area in order to enhance accuracy. Changes in illumination are also mitigated via the normalization process. Object features are extracted and characterized according to their gradient distribution. For object tracking, these distributions are then compared to a region of interest in subsequent frames to determine a best fit for match. The probability of the match being correct is determined by how well the object's gradients correlate between images.

Proposed Work

In this report a HOG classifier is used to scan the patch area in each video frame captured by a non-static camera. Targets include object classes such as pedestrians, drones, projectiles and moving vehicles. From this process gradient models are developed for both target and surrounding environment. The scanning process used by the HOG classifier provides a response array of correlation results detailing the relative likelihood of a match to the target object. The HOG response is analyzed to determine the maximum response which in turn correlates to object position. Information from the HOG response matrix is also used to inform the confidence level in the accuracy of the HOG measurement. A confidence factor is then developed from analysis of the response matrix.

Object position measurements and confidence factor are combined into a cost function which is then used by a Kalman filter tracker to adaptively update its model in real-time. The cost function takes into account various unforeseen noise impediments. By probabilistically estimating motion state and measurement error, the resulting tracking-by-detection algorithm robustly tracks objects in complex backgrounds with occlusions and other noise impediments.

This report is organized into several sections.

1. Detailed analysis of HOG methodology for object measurements and development of a confidence factor method to determine confidence in measurement
2. Kalman filter analysis for object tracking. Discussion of problems using Kalman filter for object tracking.
3. Analysis of results and discussion of stability and performance.
4. Conclusion

Histogram of Oriented Gradients Classifier (HOG)

The Histogram Oriented Gradient detector was introduced by Dalal and Triggs at the CVPR conference in 2005 [44]. HOG remains a popular feature detector due to its robustness to various impediments such as low lighting, blur, scale and illumination variations [45].

The series of steps used for creating the detector are shown in Figure 1 below.

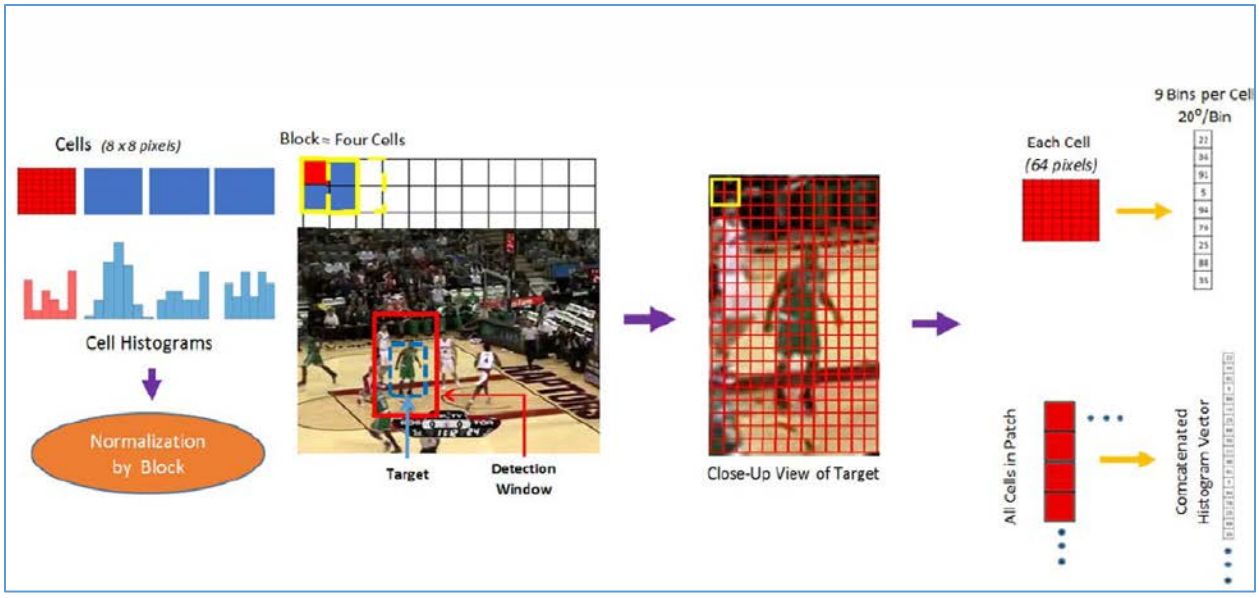


FIGURE 1 - HOG CALCULATION ANALYSIS

To form the histograms, each image is divided into a series of 8x8 pixel cells. Each cell is grouped with its adjacent cells to form blocks consisting of 4 cells each. The histograms contain nine bins, each bin representing 20 degrees. Gradients are computed by performing calculations at each cell window as it slides across the image. For purposes of illustration, an image size of 64x128 pixels is used. There is a 50% overlap (stride) as the cell moves through each row and column. Figure 2 illustrates this process.

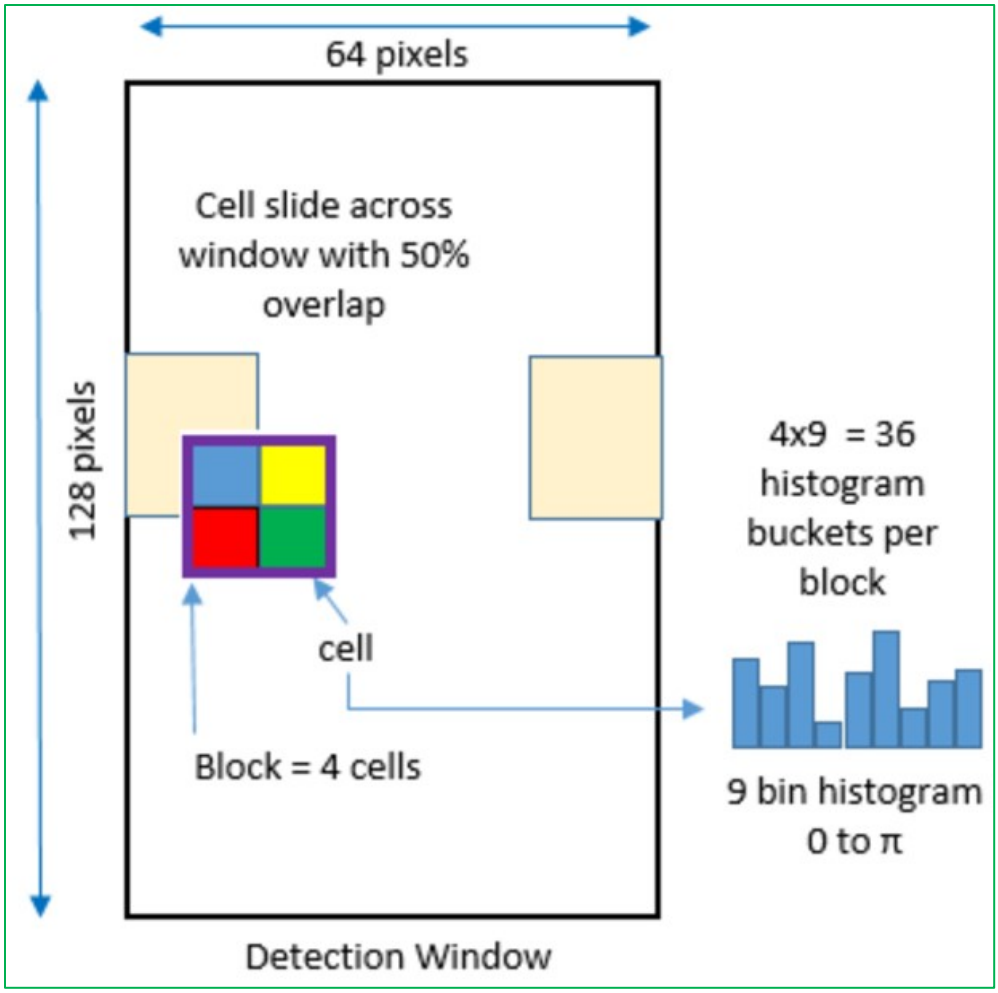


FIGURE 2 - HOG HISTOGRAM DESIGN

Histograms are calculated for each block, mapping the 64 pixels in each cell to one of nine bins in the histogram. The magnitude of each gradient is reflected by the contribution to each bin. Depending on orientation, the magnitude values are added to the corresponding gradient bin. Calculations for the magnitude and gradient values are shown below. (m,n) refers to the pixel position in the cell corresponding to row and column positions respectively.

Gradient Vector Calculations

$$I_h(m,n) = I_k(m+1,n) - I_k(m-1,n) \ \forall(m,n) \quad (1)$$

$$I_v(m,n) = I_k(m,n+1) - I_k(m,n-1) \ \forall(m,n) \quad (2)$$

Gradient Magnitude

$$M_k(m,n) = \sqrt{I_h(m,n)^2 + I_v(m,n)^2} \quad \forall(m,n) \tag{3}$$

Gradient Direction

$$\theta_k(m,n) = tan^{-1}\left(\frac{I_v(m,n)}{I_h(m,n)}\right) \quad \forall(m,n) \tag{4}$$

An example for gradient values for each of the cell positions is shown below. Magnitude values are entered into each histogram bin based on orientation (gradient direction). Magnitudes which do not fall directly into the center of a bin are added into the two adjacent bins. In the example shown here, the gradient magnitude of 22 falls half way between the 60 degree and 80 degree bins. Therefore, one-half the magnitude is entered into each bin. See Figure 3.

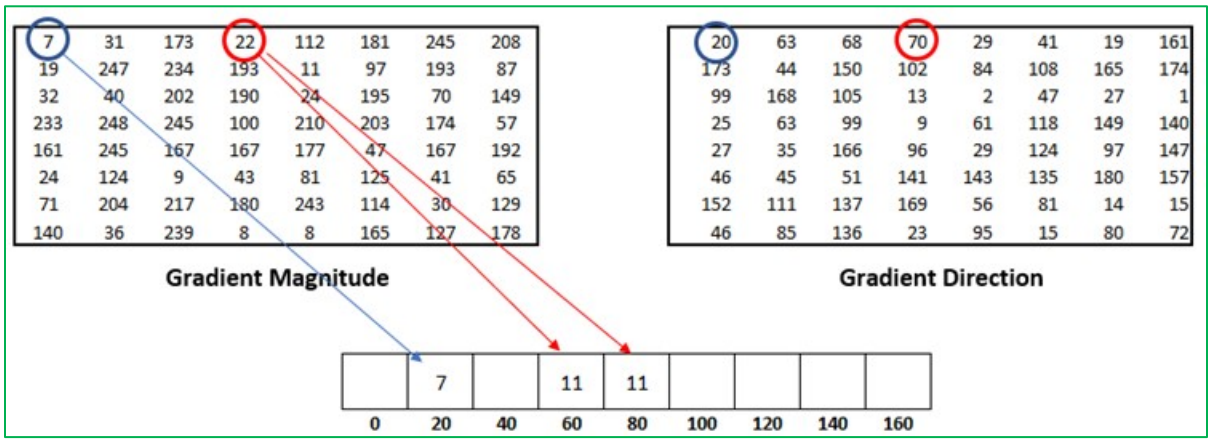


FIGURE 3 - HOG GRADIENT CALCULATION

Normalization

L2-norm is used for normalization. Each set of four cells is formed into blocks by concatenating the histograms of the four cells within the block. This block normalization is performed by concatenating the histograms of four cells within the block into a vector with 36 components (4 histograms x 9 bins per histogram). Divide this vector by its magnitude to normalize it.

This process is outlined by Dalal and Triggs for each block and is illustrated in equation 5.

$$v_i^* = {v_i} \Big/ \sqrt{\sum_{i=1}^{k} v_i^2 + \epsilon}$$ (5)

where *i* is the individual pixel in the block, *k* equals the number of pixels in the block and ε is an insignificant constant which prevents division by zero. v_i and v_i^* are the normalized gradient and the original gradient magnitude respectively for each pixel *i*.

Normalization Detail

Normalization provides the HOG classifier with increased robustness to illumination and contrast changes. An example is shown below for additive and multiplicative intensity changes.

Given the orientation of pixel values shown below,

	122	
150		58
	30	

the magnitude in the x and y directions is calculated.
magnitude- X direction gradient: 150 - 58 = 92
magnitude- Y direction gradient: 122 - 30 = 92

$$M(x,y) = 130.1 = \sqrt{92^2 + 92^2}$$

If a value of 10 is added to each pixel value (suggesting an intensity change),

	132	
160		68
	40	

the same magnitude in the X direction and magnitude in the Y direction gradients are obtained.

However, if each value is multiplied by 2, a different value is calculated.

	244	
300		116
	60	

magnitude- X direction gradient: 300 - 116 = 184
magnitude- Y direction gradient: 244 - 60 = 184

$$M(x, y) = 260.2 = \sqrt{184^2 + 184^2}$$

If you divide all vectors by their respective magnitudes, then each magnitude- X direction and magnitude- Y direction gradient is equal to 0.707. Normalization in effect makes the detector more invariant to contrast changes.

Our Approach

Using a HOG feature detector, a detection window (patch) is scanned for each video frame. This removes the need for scanning an entire image, thus minimizing computation time for real-time applications. Accurate positioning of the object in each frame thus becomes critical to the success of the tracking process. Loss of track occurs when objects fall outside the patch region. Unless alternative methods are used, tracking may be lost forever. Targets include object classes such as pedestrians, drones, projectiles and moving vehicles.

From this scanning process gradient models are developed for both target and surrounding environment in the patch area. The scanning process used by the HOG classifier provides a response array of correlation results detailing how well each section of the patch matches the object of interest. Figure 4 shows the patch area and target location. Video frame shows target and detection window. In this example, the detection window is 2.5 times larger than the target object.

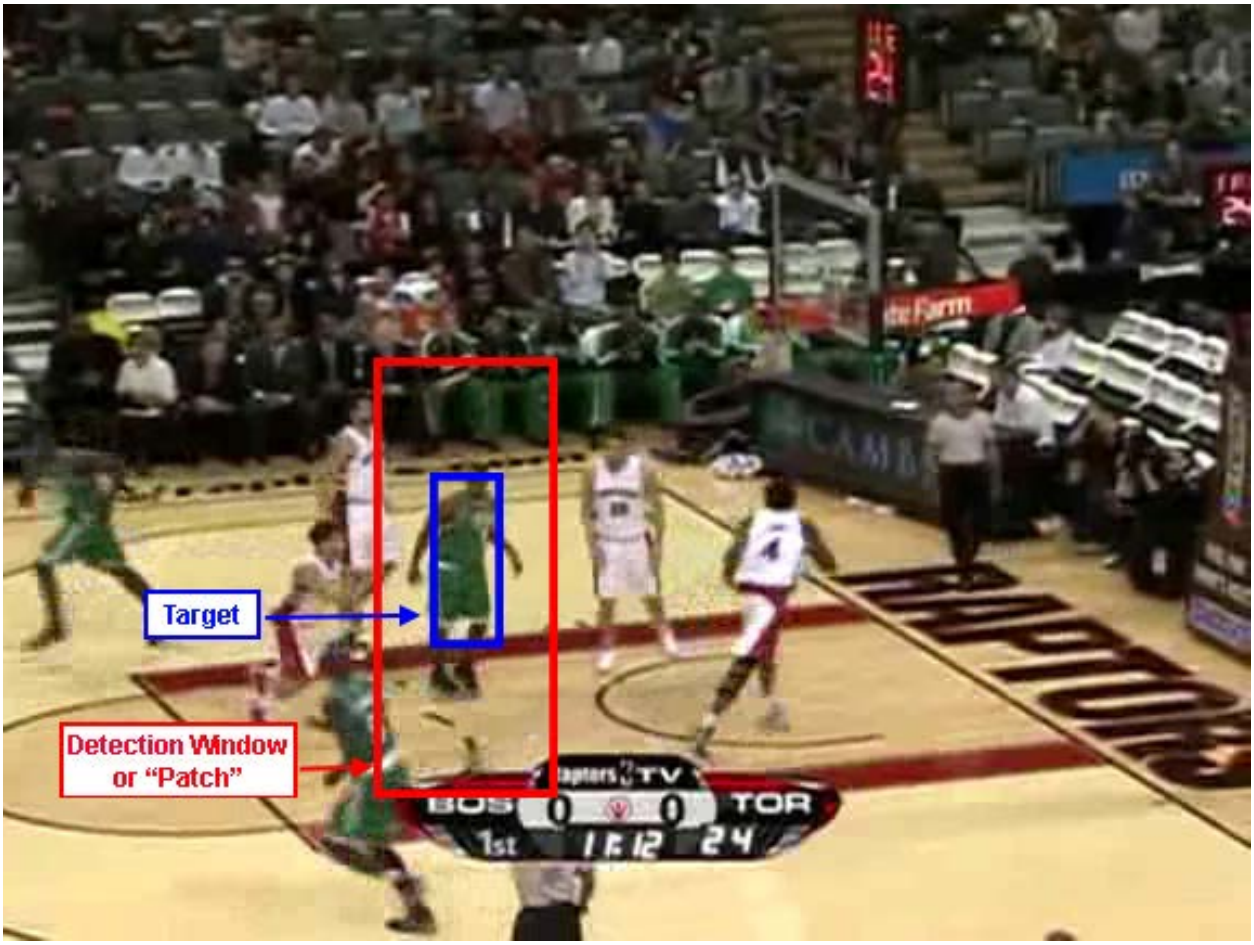


FIGURE 4 - TARGET AND PATCH POSITIONS ILLUSTRATED

A reference or ground-truth target is provided in the initial frame. An appearance model is then created defining the object by a vector of gradients. For subsequent frames, this reference model is scanned across multiple sections of the patch area to search for the object whose appearance is most similar to the reference. For each measurement, correlation values are obtained by multiplying the reference vector against each candidate vector. The resulting vector is summed, providing a quantitative indication of how well the sample appearance compares to the reference. This number represents the confidence that the computed target position is accurate from a Euclidean distance perspective.

Scanning the entire image is computationally prohibitive for real-time tracking. Only the detection window or patch is used. In this example, an 8x8 cell (64 pixels) is reduced to a string of nine values (magnitude and direction). This significantly reduces the dimensionality of its feature vector.

Kalman Filter Analysis

The Kalman Filter (KF) [46] is a linear optimum state estimator for Gaussian dynamic systems which yields the smallest expected mean-squared error. The performance of the KF degrades in the presence of non-Gaussian noise that nevertheless is present in many real-world scenarios. In traditional applications using the Kalman filter, state variables are estimated with assumed values of process and measurement noise. However, the utility of the standard Kalman filter degrades when the statistical noise model diverges from the environment. In real-time object tracking, a priori models are difficult to estimate for real systems. Over time, the errors between Kalman estimates and actual values can diverge, becoming unstable when the rate of convergence is slow. Performance of the Kalman filter is highly dependent on its predefined model.

For use of the Kalman filter to track moving objects, a dynamic model of the object motion is required. We use a model that assumes constant velocity (CV) during the sampling interval. This model is used extensively due to its simplicity and effectiveness. Disturbances to a constant velocity motion model occur when the object unexpectedly is subject to random impulse noise. This modelled noise (υ_k) affects performance of the Kalman filter when it varies from the system state estimation.

Noise statistics for a tracked object may also be time variant. In real environments, objects may have different trajectories with various degrees of acceleration. This novel proposal does not restrict the level of noise to being Gaussian or constant. To ensure convergence and minimize the impact of random non-Gaussian noise used in the estimates, a real-time adaptive Kalman filter is proposed.

Review of Kalman Filter

Kalman equations are shown below. The Kalman Model indicates the update state equation for the next object position based on the object's prior state, prior measurement and process noise. This state-space model contains a pair of equations for state and observation. The model parameters used to estimate object position are generally not known in advance, nor are they necessarily static.

Kalman Model

The state equations for the environments are described in equations 6-7.

$$x_k = Ax_{k-1} + \omega_k \qquad 6$$

where ω_k *is the Gaussian process noise with Normal distribution* $N(0, Q)$.

$Q = E[\omega_k \, \omega_k^T]$, the estimation error covariance matrix.

$$Z_k = Hx_k + v_k \quad (7)$$

Where v_k *is the Gaussian measurement noise with Normal distribution* $N(0, R)$

$R = E[v_k\, v_k^T]$, the measurement noise covariance matrix

x_k is the state vector

A is the state transition matrix from the state time *k-1* to the state at time *k*

In the model Z_k is the actual measurement of the object at the discreet time index *k*, H is an *m x n* matrix called the observation transition matrix. H is the connection between state vector and measurement vector. In this instance, velocity is not measured so is removed from the measurement model. H is assumed to be time invariant.

Kalman Equations

$$x_k = Ax_{k-1} \quad (8)$$

$$P_k = A \times P_{k-1} \times A^T + Q \quad (9)$$

$$K_k = P_k H^T (HP_k H^T + R)^{-1} \quad (10)$$

$$x_k = x_{k-1} + K_k(Z_k - Hx_{k-1}) \quad (11)$$

$$P_k = (I - K_k H)P_k \quad (12)$$

P_k is the covariance error matrix at time *k*. P_k is defined as the estimated error $E[e_k\, e_k^T]$ where $e_k = x_k - x_{k\_actual}$

The state transition matrix **A**, process noise covariance **Q** and measurement noise covariance **R** are each initialized prior to tracking.

Kalman Filter

The Kalman filter is an optimal estimator whereby it utilizes uncertain observations to infer parameters of interest. A basic assumption in using the filter is that environmental noise is

Gaussian. The filter then minimizes the mean square error of parameter estimates. For linear systems where there is uncertainty in the simulation model and measurements, the Kalman filter minimizes the error by a continuous update process.

The algorithm works in a two-step process that includes both prediction and measurement update. A weighted average is used for measurement estimates with weighting being higher based on estimates that have a higher certainty. As new information arrives, each prediction is updated.

The Kalman filter is readily used for real-time processing, an essential feature for our tracking application. The recursive nature of the Kalman filter minimizes computational overhead. There is no need to store all previous measurements to compute future estimates.

In real-world applications, noise impediments in the environment may not be Gaussian. Performance of the Kalman filter can also be negatively affected by sudden changes in movement, occlusions and other changes in the environment. In some cases, the Kalman filter can diverge when estimates sufficiently diverge from the expected model.

In using the information gained from the HOG analysis, we inform the Kalman filter about changes in the environment in a statistical sense. Confidence derived from the HOG object estimation is used to refine the Kalman model to update the measurement noise estimation. This increases the probability that the Kalman tracker error is minimized.

The effects of divergence on the Kalman filter when noise variances are incorrectly estimated has been explored by Guo in the 2018 paper "Tracking Analysis and Improvement of Broadband Kalman Filter using the Two-Echo-Path Model as a Rapid Tunnel". Simulation results are demonstrated where the model is misaligned from the environment.

HOG Descriptor Analysis

In this section the HOG response matrix is computed for each frame for various videos in the VOT14 database. The maximum HOG response is used as the estimate for the most probable object position. The confidence that the estimate is correct is determined by the magnitude of maximum HOG response.

In the video *Doll* below, the target is moved in both in-plane and out-of-plane rotations. The target is also subject to scale variations and occlusions. Two sample frame images are shown to illustrate the image where tracking is accurate and again when tracking errors are present. See Figure 5 and Figure 6.

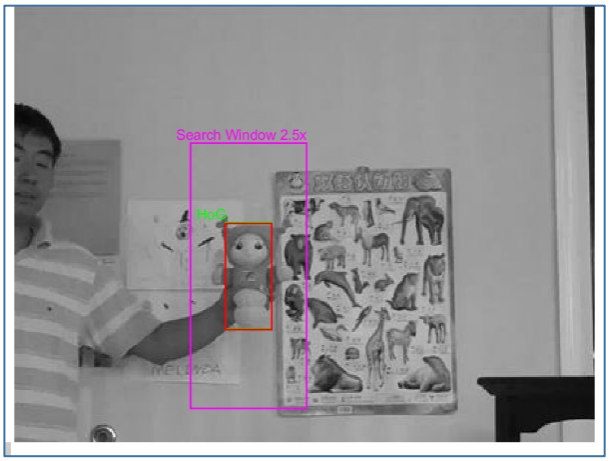


FIGURE 5 - DOLL VIDEO FRAME 1

FIGURE 6 - DOLL VIDEO FRAME 3760

The HOG maximum response across all frames in the video is plotted (see Figure 7) . The maximum HOG response (shown in blue) deteriorates rapidly near frame 3750. This corresponds to a significant mismatch between the HOG position estimate and ground truth (labelled HOG Position Error in the plot). The position errors are shown in red while the Y-axis is located to the right of the plot. These errors are calculated by computing the RMS distance between the HOG positon estimate and the position provided by ground truth. Errors in this instance stem from scale variation and blur. These impediments can be clearly seen at frame 3750. Figure 8 is a zoomed-in version of Figure 7, so that the change in maximum HOG response can be more clearly displayed.

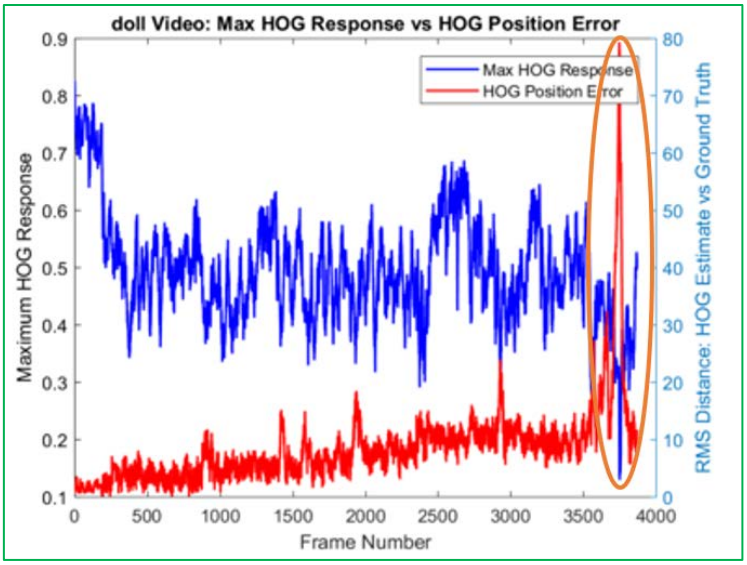


FIGURE 7 - DOLL VIDEO HOG RESPONSE VS RMS POSITION ERROR (ALL FRAMES).

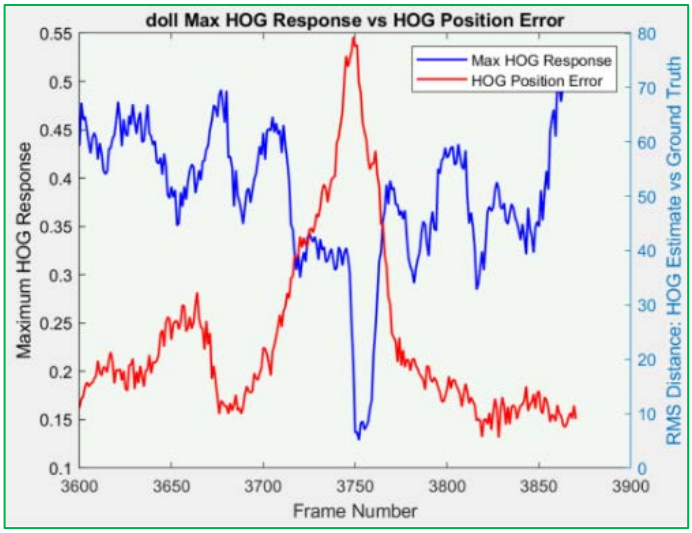


FIGURE 8 - ZOOMED-IN VERSION SHOWING FRAMES 3600 THROUGH 3800 ONLY.

The relationship between the maximum HOG response and position estimate error is further illustrated by viewing scatter plots (maximum HOG response vs RMS error). See Figure 9. Errors in the individual X and Y directions are shown in the middle and right plots. Here the Doll video is used again to examine the relationship between the maximum HOG response error looking at position errors in the horizontal (X) and vertical (Y) directions individually as well for the RSM error (shown to the left).

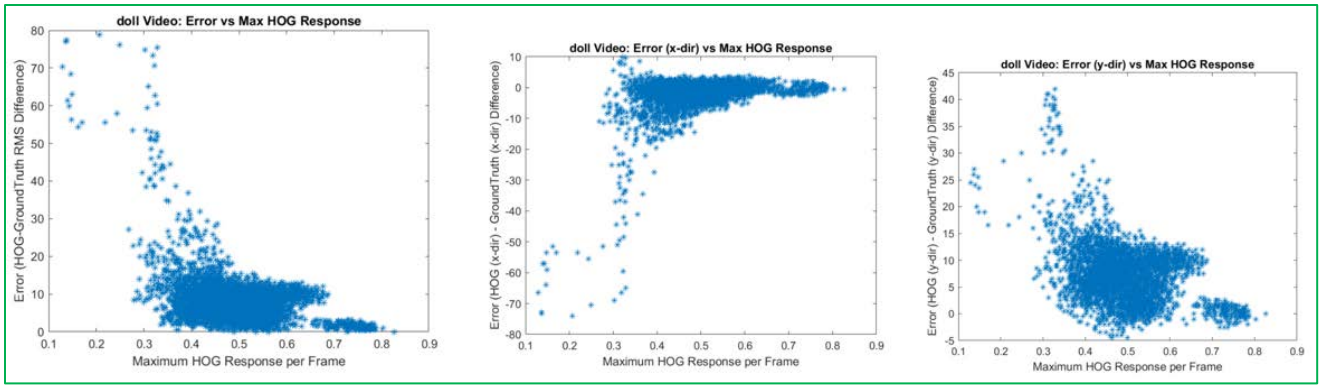


FIGURE 9 - DOLL VIDEO - POSITION ESTIMATE ERRORS VS HOG MAXIMUM RESPONSE.

The scatter plots clearly illustrate the probability of large errors when the maximum HOG response is less than 0.3, while having a low probability of error when above 0.5. Between these two values, the error is more variable but well behaved (< 20-25 pixels).

Scatter plots using other video examples are shown. For reference, the first frame of the liquor video is shown in Figure 10. Scatter plots for maximum HOG response vs X-direction, Y-direction and RMS errors are shown Figure 11, Figure 12 and Figure 13 in respectively.

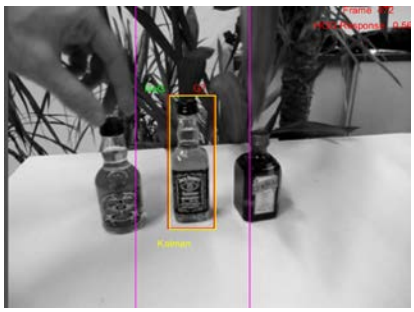


FIGURE 10 - LIQUOR VIDEO FRAME

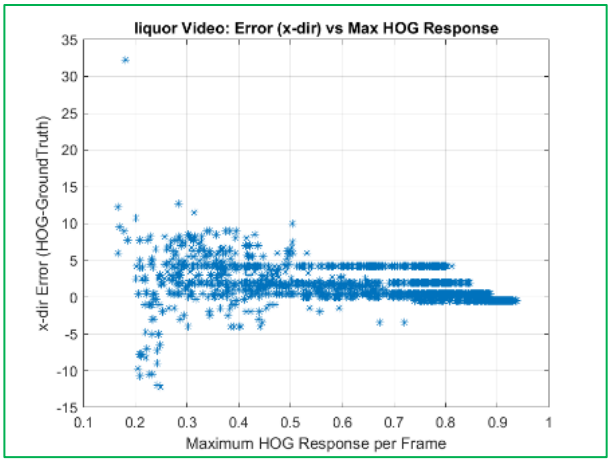


FIGURE 11 - LIQUOR VIDEO POSITION ERROR PLOT (X-DIRECTION)

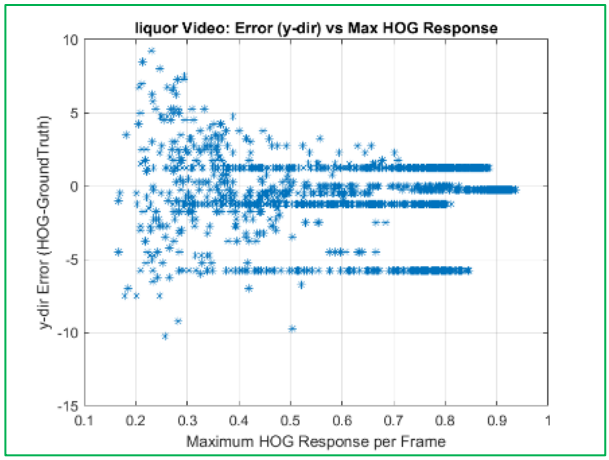


FIGURE 12 - LIQUOR VIDEO POSITION ERROR PLOT (Y-DIRECTION)

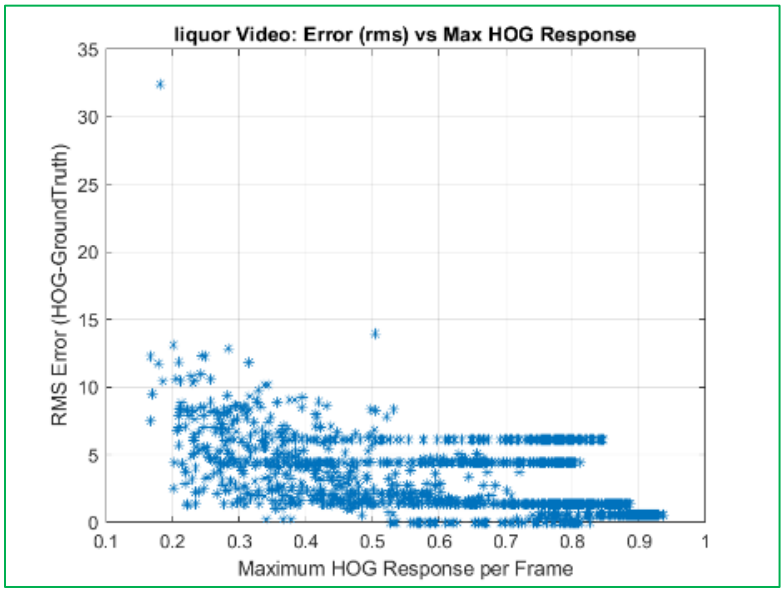


FIGURE 13 - LIQUOR VIDEO POSITION ERROR PLOT (RMS)

It can be seen that errors tend to increase significantly as the maximum HOG response measures less than 0.3.

An additional example is shown below in the 'Soccer' video. Positioning errors again increase rapidly as the HOG maximum response falls below 0.3. Figure 14 is the first frame from the soccer video. Figure 15, Figure 16 and Figure 17 are scatter plots for the maximum HOG response vs error in the X direction, Y direction and RMS respectively.

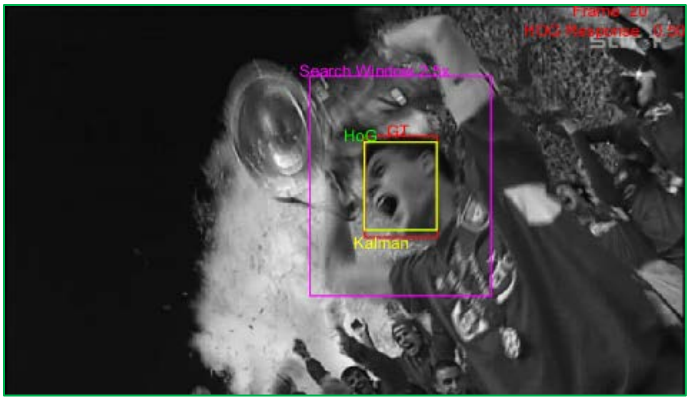


FIGURE 14 - SOCCER VIDEO IMAGE FRAME

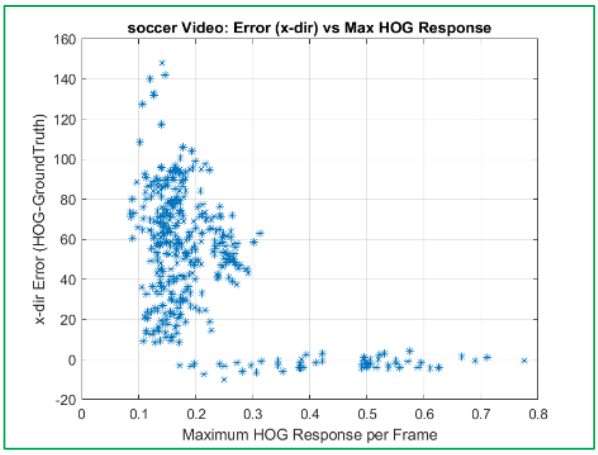


FIGURE 15 - X-DIRECTION HOG RESPONSE VS POSITION ERROR

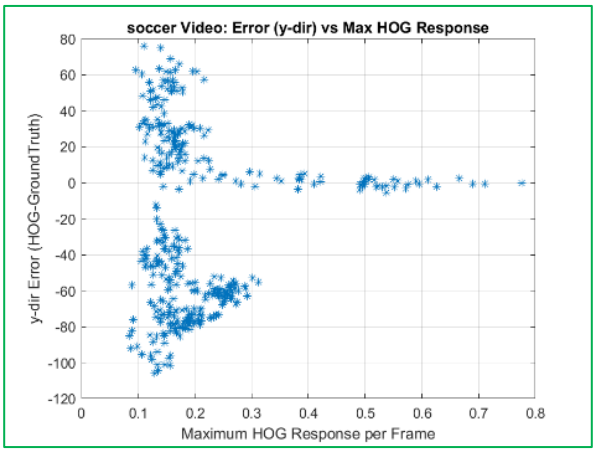


FIGURE 16 - Y-DIRECTION HOG RESPONSE VS POSITION ERROR

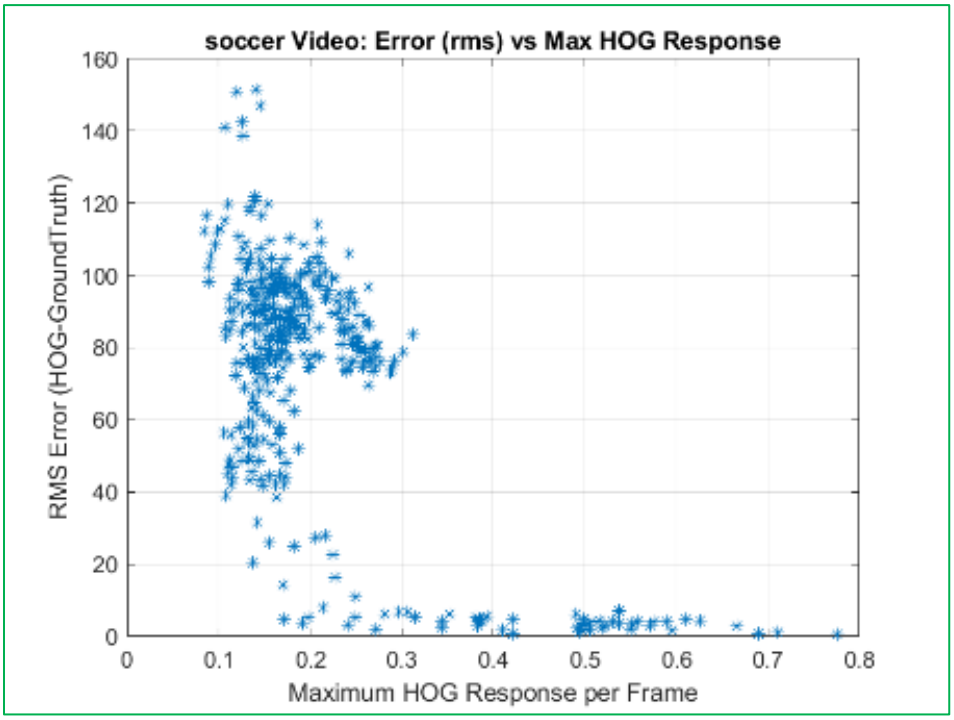


FIGURE 17 - RMS HOG RESPONSE VS POSITION ERROR

To further validate the relationship between position error and maximum HOG response, simulations for 30 videos selected from the VOT14 database and then compiled into histograms comparing the maximum HOG response to the RSM error. Altogether about 20,000 video frames were analyzed. The results of this experiment are shown below showing analysis in the

X and Y directions as well as RMS, ie (sqrt($X^2 + Y^2$)). See Figure 18, Figure 19 and Figure 20.

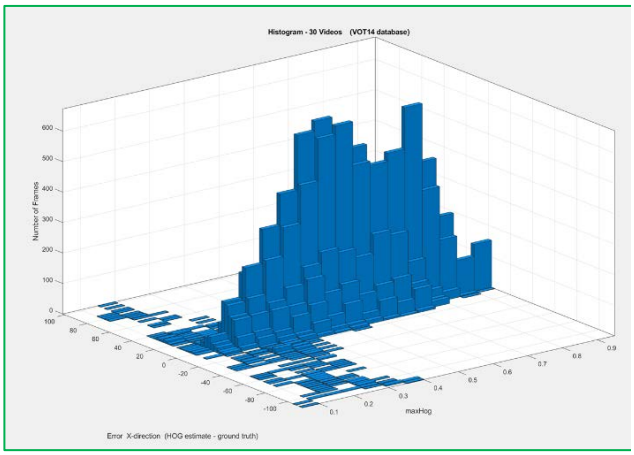


FIGURE 18 - MAX HOG VS X-DIRECTION ERROR

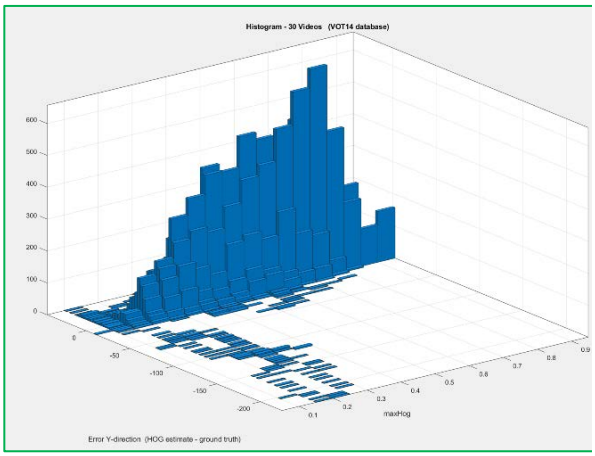


FIGURE 19 - MAX HOG VS Y-DIRECTION ERROR

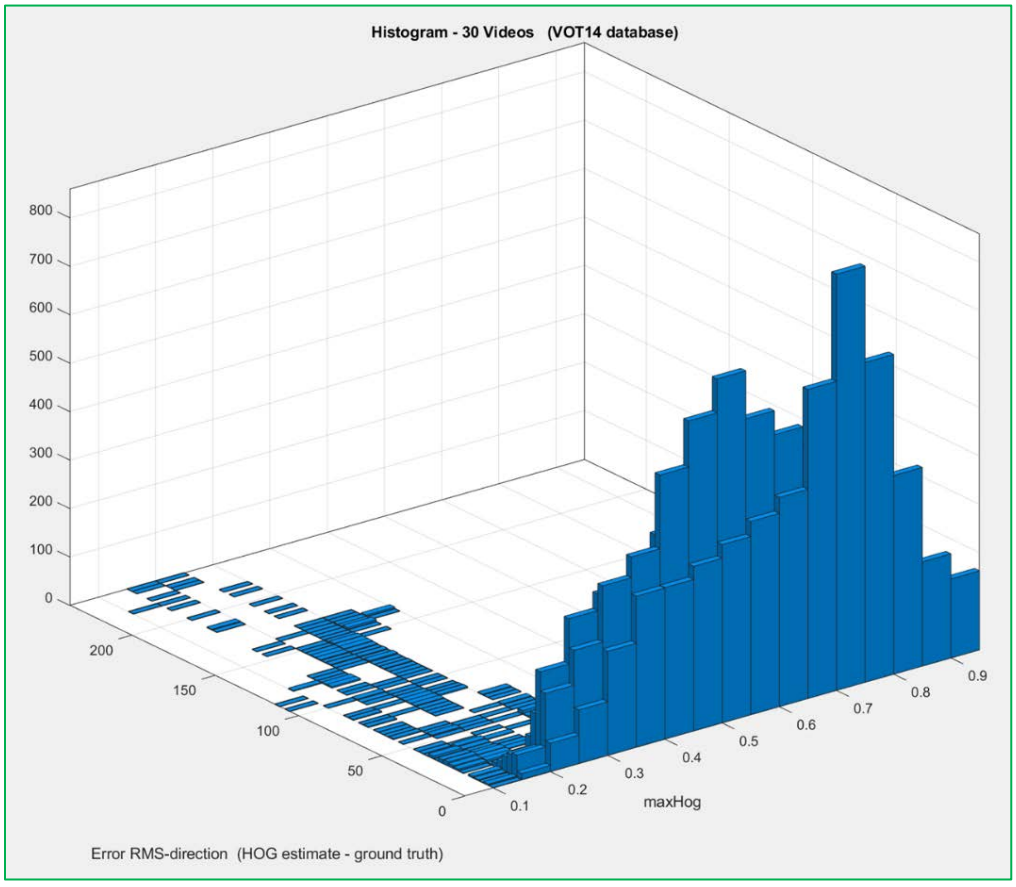


FIGURE 20 - MAX HOG VS RMS ERROR

Results and Discussion

Within the Kalman system, the model used for the measurement analysis is viewed in a statistical sense. It is probabilistically estimating motion state and predicting object position. The statistical information used for determining object location calculated by HOG can be interpreted as a priori confidence in object position. System state is simultaneously updated to account for perturbations and non-linear outliers that may occur in a random fashion.

To develop an optimal correction factor for the Kalman measurement noise estimate, measurements from the 30 videos in the VOT14 database were compiled. In total, 40,000 measurements were used (20,000 measurements in each of the X and Y directions). Twenty bins were used for each range of maximum HOG response. The standard deviations corresponding to each bin were then plotted against the bin range. The RMS values for each standard deviation bin are shown in Figure 21. A curve fitting algorithm (5th order polynomial fit) for each standard deviation measurement was calculated. This is shown in the line plot in Figure 21.

bin	XYmean	XYstd	SamplesPerBin
0.05	NaN	60	0
0.1	-55.313	53.007	4
0.15	4.8096	38.275	130
0.2	-5.7932	39.043	474
0.25	-3.3257	20.962	1378
0.3	-3.3352	21.912	2184
0.35	-0.033105	14.293	3066
0.4	0.58429	10.098	3316
0.45	1.2825	6.7513	4448
0.5	1.3718	5.7745	4356
0.55	0.53974	6.1103	4114
0.6	0.27538	5.1636	3526
0.65	-0.60608	4.5513	3118
0.7	-0.65668	4.1728	3022
0.75	-0.86225	3.309	3118
0.8	0.0259	2.7966	2056
0.85	-0.04017	1.9017	1002
0.9	0.056054	0.99814	446
0.95	-0.3689	0.22709	328
1	0	0	2

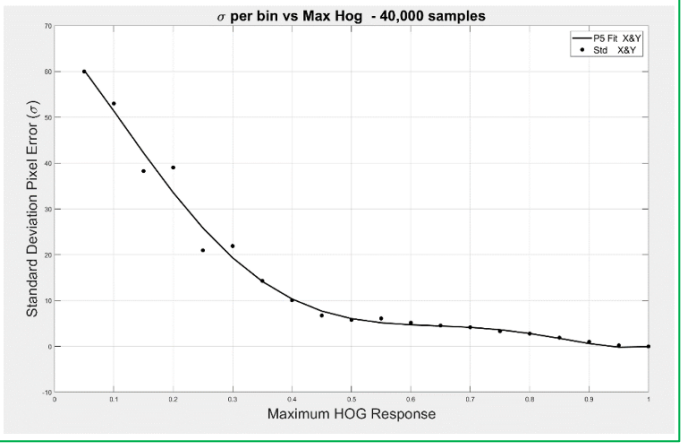


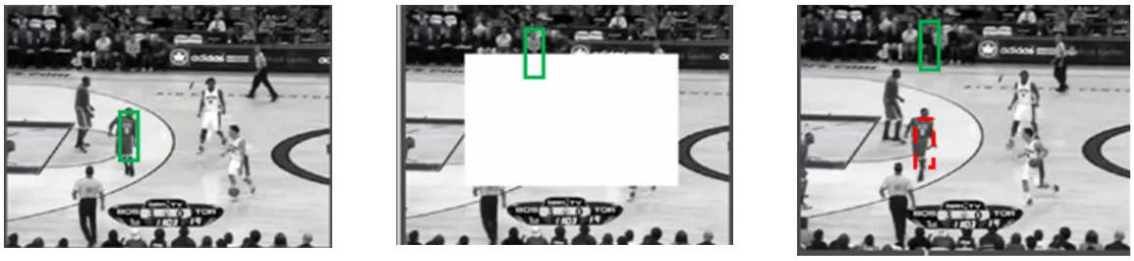
FIGURE 21 - ANALYSIS OF STANDARD DEVIATIONS BASED ON 20,000 FRAMES

Tracking Improvements

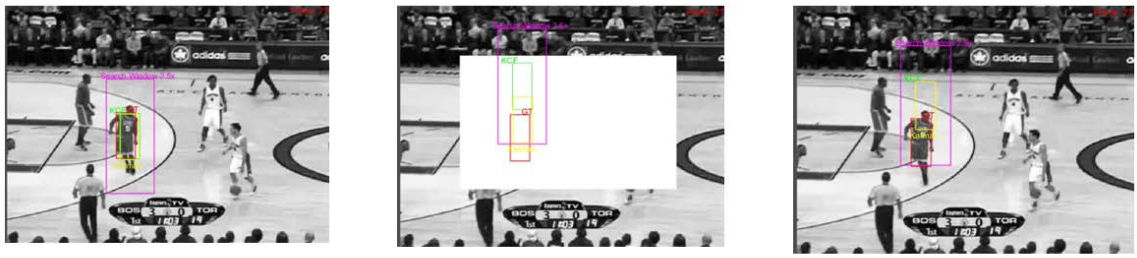
The values in the polynomial curve are then used to adjust the Kalman filter estimated measurement error based on the maximum HOG response calculated for each frame.

An artificial occlusion is first introduced in the 'basketball' video in Figure 22. The occlusion is constructed using a solid color which in effect reduces the gradient changes to zero. Two sets

of frames are shown, one using only the HOG descriptor for tracking, while the second set shows the effect by using an informed Kalman filter with measurement noise correction factor for tracking assist.



Track Lost due to Occlusion (HOG only)



Track Maintained after Occlusion (Kalman)

FIGURE 22 - TRACKING ANALYSIS COMPARING HOG ONLY WITH ADAPTIVE KALMAN FILTERING

Several bounding boxes are shown. The bounding boxes include the following.

- Ground truth (red) – provided by the VOT14 database
- Patch (maroon) – detection patch window covers the area used for searching for the target.
- HOG estimation (green) – most probable object location without using Kalman filter
- Kalman estimation (yellow dashed) – object position estimated provide by Kalman filter using HOG measurement information

These windows with corresponding colors will be used again in subsequent video frames.

It is shown in the third frame set for each video sequence from Figure 22 that using the Kalman filter enables the detector to maintain track. Surface plots were also constructed comparing the HOG response both before and during the occlusion. The relatively sharp peak for the signal is indicative of a higher confidence level that the tracking error is minimal. See Figure 23 and Figure 24. Conversely, the errors encountered during the occlusion are much larger.

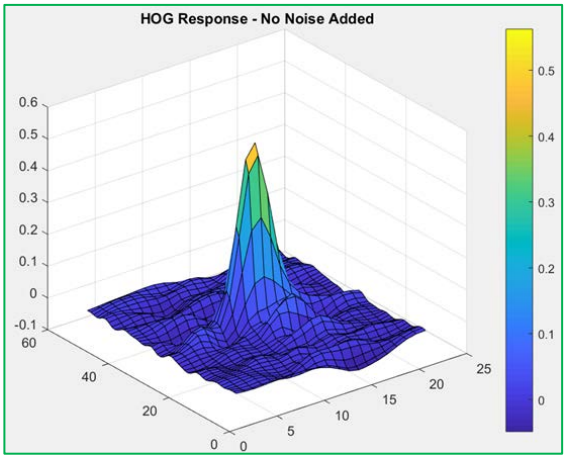


FIGURE 23 - SURFACE PLOT SHOWING HIGH MEASUREMENT CONFIDENCE

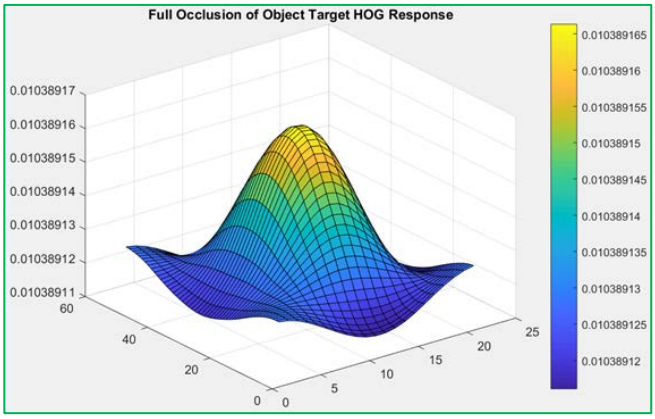


FIGURE 24 - SURFACE PLOT OF OCCLUDED FRAME

The adaptive Kalman approach is demonstrated used several video clips provided by the VOT14 database. Results are shown comparing the use of HOG-only for object track with the use of a Kalman filter. In the first sequence below, several frames of the *Jogging* video are shown. Noise impediments outlined in VOT14 include deformation, occlusion and out-of-plane rotation. The occlusion event shown by the jogger running behind a light pole is examined here.

Figure 25 below shows a video sequence using HOG only for object tracking. The occlusion causes loss of track starting at frame 73. Once the object leaves the patch window, tracking cannot be recovered without using additional methods for maintaining track. Figure 26 shows the same video sequence, but this time using a Kalman filter to assist in tracking. No adaptive scaling of the Kalman model is implemented. Track is again lost as shown in Figure 26.

FIGURE 25 – JOGGING: USING HOG ONLY FOR OBJECT TRACKING (FRAMES 68, 71, 73, 86)

FIGURE 26 - JOGGING: HOG+NON-ADAPTIVE KALMAN FILTER TRACKING (FRAMES 68, 71, 73, 86).

In this next sequence (Figure 27), HOG is assisted again using a Kalman filter. However, this time the confidence factor is employed to update the Kalman model with information obtained by the HOG gradient measurements. Additional frames are shown to better illustrate the changes in the Kalman position estimation. As shown here, the Kalman filter is now able to maintain track during and after the occlusion event.

FIGURE 27 - JOGGING SEQUENCE USING HOG TO INFORM AN ADAPTIVE KALMAN FILTER.

A second video *'football1'* was evaluated. Noise impediments in this video series include partial occlusions, in-plane rotations, out-of-plane rotations, background clutter and blur. In Figure 28 a frame sequence using HOG-only for tracking is shown. In the final frame of the video (frame 174), track is again lost when HOG-only is used. Using a Kalman filter without a scaling factor also resulted in a loss of track. This is shown in Figure 29. In each case, the ground truth position (red rectangle) is located outside the patch rectangle (maroon). In this scenario, track cannot be re-established.

In Figure 30, the adaptive Kalman filter is used. In the second row, the Kalman estimation aided with the knowledge of the HOG maximum response is still located within the patch, thus able to maintain track. Additional frames are shown in the sequence to provide a better view of the tracking process.

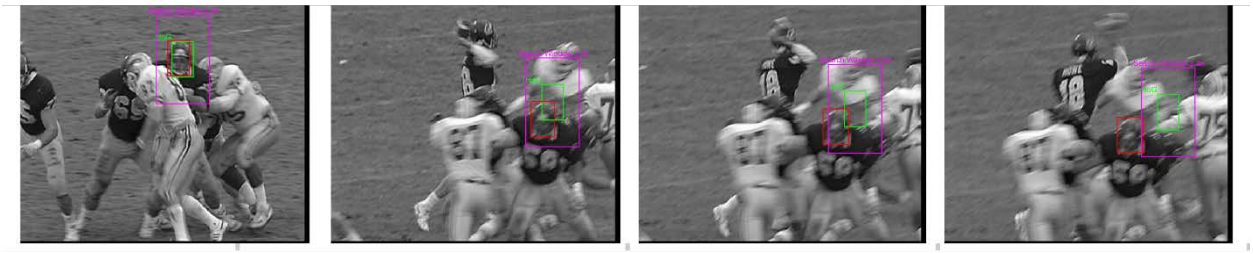


FIGURE 28 - 'FOOTBALL' FRAME SEQUENCE USING HOG ONLY (FRAMES 154, 172, 173, 174).

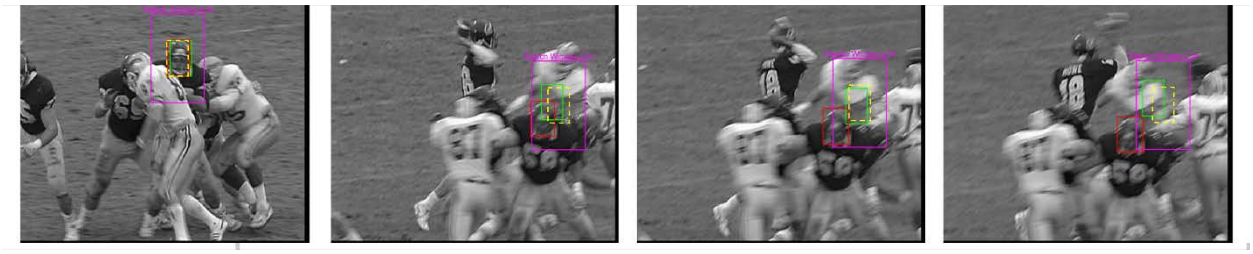


FIGURE 29 - 'FOOTBALL' FRAMES WITH HOG+ NON-ADAPTIVE KALMAN FILTER (FRAMES 154, 172, 173, 174).

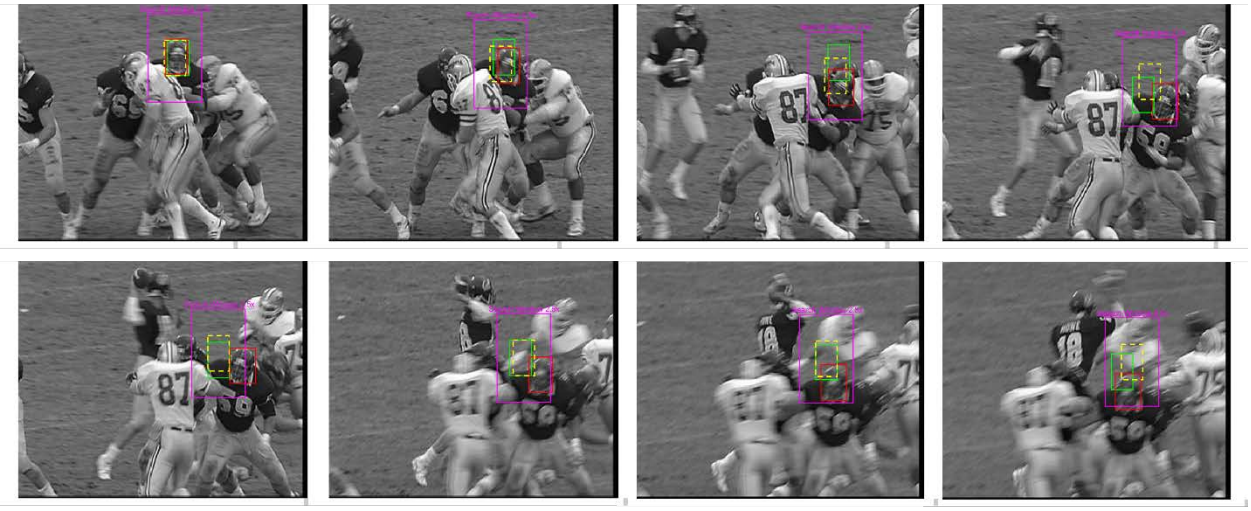


FIGURE 30 - 'FOOTBALL' FRAMES: HOG WITH ADAPTIVE KALMAN FILTER (ADDITIONAL FRAMES INCLUDED).

Conclusion

This paper has described the use of a tracking prediction algorithm for adaptively updating a Kalman filter model based on information derived from the HOG correlation measurements. The method for updating the filter includes a quantitative method for determining confidence in the HOG object position estimation. This information is known a priori to implementing the Kalman filter. This overcomes some of the issues using Bayesian methods where prior measurements (such as Kalman gain) as a means to predicting future events. Also, using our methodology imparts only a minor efficiency penalty with no impact for real-time tracking. Adaptively scaling the Kalman model also is object appearance agnostic. The method works well for not only people tracking, but also for other objects in motion as well.

Areas for future research might include using this method with off-line training models, increasing the patch detection window size when measurement confidence is low and adjusting measurement confidence downwards when multiple objects of similar appearance are located within the patch window.

Using off-line models offer promise given current research into system model updates using methods that would not inhibit real-time tracking. These methods work best when there is a priori knowledge as to the type of object that will be tracked.

References

[1] P. Reinartz, M. Lachaise, E. Schmeer, T. Krauss, and H. Runge, “Traffic monitoring with serial images from airborne cameras,” *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 3–4, pp. 149–158, 2006.

[2] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, “Video sensor network for real-time traffic monitoring and surveillance,” *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 103–112, 2010.

[3] G. Mathur, D. Somwanshi, and M. M. Bundele, “Intelligent Video Surveillance based on Object Tracking,” *3rd Int. Conf. Work. Recent Adv. Innov. Eng. ICRAIE 2018*, vol. 2018, no. November, pp. 1–6, 2019.

[4] C. Kucukkececi and A. Yazici, “Multilevel Object Tracking in Wireless Multimedia Sensor Networks for Surveillance Applications Using Graph-Based Big Data,” *IEEE Access*, vol. 7, pp. 67818–67832, 2019.

[5] J. H. An and K. S. Hong, “Finger gesture-based mobile user interface using a rear-facing camera,” *Dig. Tech. Pap. - IEEE Int. Conf. Consum. Electron.*, pp. 303–304, 2011.

[6] M. Danancher, J. J. Lesage, and L. Litz, “Model-Based Location Tracking of an a priori Unknown Number of Inhabitants in Smart Homes,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 1090–1101, 2016.

[7] S. Queiros, P. Morais, D. Barbosa, J. C. Fonseca, J. L. Vilaca, and J. D’Hooge, “MITT: Medical Image Tracking Toolbox,” *IEEE Trans. Med. Imaging*, vol. 37, no. 11, pp. 2547–2557, 2018.

[8] T. Liu, X. Cao, and J. Jiang, “Visual Object Tracking with Partition Loss Schemes,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 633–642, 2017.

[9] J. Pan, B. Hu, and J. Q. Zhang, “Robust and accurate object tracking under various types of occlusions,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 223–236, 2008.

[10] P. Wang and H. Qiao, “Online appearance model learning and generation for adaptive visual tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 156–169, 2011.

[11] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, “Relative end-effector control using cartesian position based visual servoing,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, 1996.

[12] Y. Wu, J. Lim, and M.-H. Yang, “Online Object Tracking: A Benchmark Supplemental Material,” *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–13, 2013.

[13] I. Bogun and E. Ribeiro, "RobStruck: Improving occlusion handling of structured tracking-by-detection using robust Kalman filter," *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3479-3483.

[14] H. Kaur and J. S. Sahambi, “Vehicle Tracking in Video using Fractional Feedback Kalman Filter,” *IEEE Trans. Comput. Imaging*, vol. 2, no. 4, pp. 1–1, Aug. 2016.

[15] A. Yadav, P. Awasthi, N. Naik, and M. R. Ananthasayanam, “A constant gain Kalman filter approach to track maneuvering targets,” *Proc. IEEE Int. Conf. Control Appl.*, pp. 562–567, 2013.

[16] H. Shu, E. P. Simon, and L. Ros, “Third-order kalman filter: Tuning and steady-state performance,” *IEEE Signal Process. Lett.*, vol. 20, no. 11, pp. 1082–1085, 2013.

[17] R. Dehghannasiri, M. S. Esfahani, and E. R. Dougherty, “Intrinsically Bayesian robust kalman filter: An innovation process approach,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2531–2546, 2017.

[18] H. Rong, C. Peng, Y. Chen, L. Zou, Y. Zhu, and J. Lv, “Adaptive-Gain Regulation of Extended Kalman Filter for Use in Inertial and Magnetic Units Based on Hidden Markov Model,” *IEEE Sens. J.*, vol. 18, no. 7, pp. 3016–3027, 2018.

[19] H. Ren and P. Kazanzides, “Investigation of attitude tracking using an integrated inertial and magnetic navigation system for hand-held surgical instruments,” *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 2, pp. 210–217, 2012.

[20] F. Edrisi and V. J. Majd, “Attitude estimation of an accelerated rigid body with sensor fusion based-on switching extended Kalman filter,” *2015 AI Robot. IRANOPEN 2015 - 5th Conf. Artif. Intell. Robot.*, pp. 1–6, 2015.

[21] A. Makni, H. Fourati, and A. Y. Kibangou, “Energy-Aware Adaptive Attitude Estimation under External Acceleration for Pedestrian Navigation,” *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1366–1375, 2016.

[22] F. Tang, S. Brennan, Q. Zhao, and H. Tao, “Co-Tracking Using Semi-Supervised Support Vector Machines,” 2007.

[23] O. Javed, S. Ali, and M. Shah, “Online detection and classification of moving objects using progressively improving detectors,” *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 696–701, 2005.

[24] P. Dai, K. Liu, Y. Xie, and C. Li, “Online co-training ranking SVM for visual tracking,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 6568–6572, 2014.

[25] L. Khalil and P. Jung, “Scaled Unscented Kalman Filter for RSSI-based Indoor Positioning and Tracking,” *Proc. - NGMAST 2015 9th Int. Conf. Next Gener. Mob. Appl. Serv. Technol.*, pp. 132–137, 2016.

[26] I. Romanovski and P. Caines, “Technical Notes and Correspondence_,” *IEEE Trans. Automat. Contr.*, vol. 51, no. 5, p. 795, 2006.

[27] S. Huang and G. Dissanayake, “Convergence and consistency analysis for extended Kalman filter based SLAM,” *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 1036–1049, 2007.

[28] L. Zhang, S. Li, E. Zhang, Q. Chen, and J. Guo, “Improved square root adaptive cubature Kalman filter,” *IET Signal Process.*, vol. 13, no. 7, pp. 641–649, 2019.

[29] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, “A quadratic-complexity observability-constrained unscented kalman filter for slam,” *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1226–1243, 2013.

[30] R. T. Collins, Y. Liu, and M. Leordeanu, “Online selection of discriminative tracking features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, 2005.

[31] F. De La Torre, J. Vitria, P. Radeva, and J. Melenchon, “Eigenfiltering for flexible eigentracking (EFE),” *Proceedings-International Conf. Pattern Recognit.*, vol. 15, no. 3, pp. 1106–1109, 2000.

[32] Z. Ye and Z. Q. Liu, “Genetic CONDENSATION for motion tracking,” *Soft Comput.*, vol. 11, no. 4, pp. 349–354, 2007.

[33] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 2–9, 2004.

[34] S. L. Al-Khafaji, J. Zhou, A. Zia, and A. W. C. Liew, “Spectral-Spatial Scale Invariant Feature Transform for Hyperspectral Images,” *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 837–850, 2018.

[35] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[36] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vis.*, pp. 1–28, 2004.

[37] H. Bay, “From wide-baseline point and line correspondences to 3D,” *PhD theses, Swiss Fed. Inst. Technol.*, 2006.

[38] N. Hamid, A. Yahya, R. B. Ahmad, and O. M. Al-qershi, “A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 110–116, 2012.

[39] F. Bellavia and C. Colombo, “Rethinking the sGLOH descriptor,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 931–944, 2018.

[40] S. Belongie, J. Malik, J. Puzicha, and 2008, “Shape matching and object recognition using shape contexts. PAMI, 24 (4): 509–522, 2002. 1, …,” *Pami*, vol. 24, no. 24, pp. 509–522, 2002.

[41] I. Laptev and T. Lindeberg, “Space-time interest points,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, pp. 432–439, 2003.

[42] C. Tomasi and J. Shi, “3 Good Features to Track(Shi-Tomasi),” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 593–600, 1994.

[43] K. Matusiak, P. Skulimowski, and P. Strumillo, “Unbiased evaluation of keypoint detectors with respect to rotation invariance,” *IET Comput. Vis.*, vol. 11, no. 7, pp. 507–516, 2017.

[44] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[45] N. Laopracha, K. Sunat, and S. Chiewchanwattana, “A Novel Feature Selection in Vehicle Detection Through the Selection of Dominant Patterns of Histograms of Oriented Gradients (DPHOG),” *IEEE Access*, vol. 7, pp. 20894–20919, 2019.

[46] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Fluids Eng. Trans. ASME*, 1960.