Mean Shift Object Tracking with Occlusion Handling

B.Z. de Villiers
HyperVision Research Lab
University of Johannesburg
South Africa
Email: wheres.brett.dev@gmail.com

W.A. Clarke
HyperVision Research Lab
University of Johannesburg
South Africa
Email: willemc@uj.ac.za

P.E. Robinson
HyperVision Research Lab
University of Johannesburg
South Africa
Email: philipr@uj.ac.za

***Abstract*—An object tracking algorithm using the Mean Shift framework is presented which is largely invariant to both partial and full occlusions, complex backgrounds and change in scale. Multiple features are used to gain a descriptive representation of the target object. Image moments are used to determine the scale of the target object. A kalman filter is used to successfully track the target object through partial and full occlusions, the Bhattacharyya coefficient is used to determine the measurement noise estimation.**

I. Introduction

Object tracking is of great importance in computer vision and is used in many applications such as visual surveillance, perceptual user interfaces, augmented reality and intelligent transport systems. Mean Shift [1] is a popular method used in object tracking which is also used in commercial applications due its simple implementation, efficient and robust performance. The Mean Shift method is a non-parametric, variable step-size, statistical density estimator which iteratively determines the nearest mode of a point sample distribution using gradient ascent. The Mean Shift method has been used in a number of computer vision problems, these include line fitting [2], image segmentation [3] and object tracking [4].

A number of improvements to the traditional formulation of the Mean Shift method for object tracking have been investigated [4]. Multiple features have been investigated to gain a more descriptive representation of the target object [5,6]. In [5] various colour spaces and edge directions are used as descriptive features, feature localization weights are determined according to the similarity between background features and features present in the target model. In [6] the RGB colour space, edge directions and textural information (obtained using the discrete wavelet transform) are used as descriptive features, feature localization weights are determined according to the similarity between target candidate features and features present in the target model. Scale space theory was adopted in order to successfully determine the target object's scale during tracking [7]. The Mean Shift method was applied to Gaussian kernels at various scales to determine the target object's scale. Image moments have been used with the similarity weights (between the target model and candidate) to determine the scale and orientation of the target object [8]. Multiple ellipsoidal, asymmetric kernels with asymmetric centres have been used to effectively track target position, scale and orientation simultaneously [9]. In order to remove background features from the target model and candidate a level set function has been used along the contour of the target object [10]. The level set function defines an asymmetric kernel over the target region which does not contain any background features. Mean Shift is used to track the target object's position, scale and orientation.

This paper proposes a tracking algorithm using the Mean Shift framework which is largely invariant to both partial and full occlusions, complex backgrounds and change in scale. Multiple features are used to gain a more descriptive representation of the target object, these features include colour, edges and texture. An adaptive feature weighting method is used to maximize the feature weights of features which better localize the target object. Image moments are used in conjunction with the similarity weights (between the target model and candidate) to determine the scale of the target object. A kalman filter is used to improve the tracking performance during partial and full occlusions, a measurement noise estimation is determined using the Bhattacharyya coefficient [11].

The paper is arranged as follows. Section II provides an overview of the Mean Shift tracking algorithm [4]. Section III provides a description of the various features used to describe the target object. Section IV provides details on the tracking algorithm including scale selection, kalman filter implementation and a brief overview of the tracking algorithm. Section V provides experimental results which describe the performance of the tracking algorithm. Section VI concludes the paper.

II. Mean Shift Tracking Algorithm

A. Target Representation

A target is typically defined by an ellipsoidal region or patch surrounding a region of interest in an image. A feature space is chosen (typically the RGB feature space is used) to determine a histogram of the pixel distribution in the target region. The histogram is represented by target model q. The target model is used to describe the appearance of the object located in the target region. The target model q is comprised of m normalized bins [4].

Target model:

$$\hat{q} = \{\hat{q}_u\}_{u=1...m} \tag{1}$$

$$\sum_{u=1}^{m} \hat{q}_u = 1 \tag{2}$$

Let $\{x_i^*\}_{i=1...n}$ denote the n normalized pixel locations in the target region which are centred around 0. Let $k(x)$ denote a convex, monotonically decreasing, isotropic kernel. Let b: $R^2 \rightarrow \{1...m\}$ be a function which determines the histogram bin $b(x_i^*)$ associated with the pixel location x_i^*. The probability of the feature $u = 1...m$ in the target models histogram is determined by

$$\hat{q}_u = C \sum_{i=1}^{n} k(\| x_i^* \|^2)\delta[b(x_i^*) - u] \tag{3}$$

Where δ is the Kronecker delta function. The normalization constant C is derived by imposing the condition (2), normalization constant C can therefore be represented by

$$C = \frac{1}{\sum_{i=1}^{n} k(\| x_i^* \|^2)} \tag{4}$$

B. Candidate Representation

Typically the target model is formed from the target region in the first frame of a video sequence. The target model is compared to candidate regions in the current frame to determine the location and scale of the target in the current frame. A target candidate $p(y)$ is defined by a histogram of the pixel distribution of a region in the current frame. The target candidate $p(y)$ is comprised of m normalized bins [4].

Target candidate:

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1...m} \tag{5}$$

$$\sum_{u=1}^{m} \hat{p}_u(y) = 1 \tag{6}$$

Let $\{x_i\}_{i=1...n_h}$ denote the n_h normalized pixel locations in the candidate region which are centred around y. Let $k(x)$ denote the same convex, monotonically decreasing, isotropic kernel used with the target model only with a different size (based on the scale of the target object) specified by bandwidth h. The probability of the feature $u = 1...m$ in the target candidates histogram is determined by

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k(\| \frac{y - x_i}{h} \|^2)\delta[b(x_i) - u] \tag{7}$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\| \frac{y - x_i}{h} \|^2)} \tag{8}$$

C. Similarity Model

In order to determine the similarity between the target model and the target candidate a similarity function is determined. The similarity function used is the sample estimate of the Bhattacharyya coefficient [11] between the distributions $\hat{q}$ and $\hat{p}(y)$. The similarity function is defined by

$$\hat{\rho}(y) = \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \tag{9}$$

Due to the conditions imposed by (2) and (6) the similarity function has a minimum value of 0 (distributions are orthogonal) and a maximum value of 1 (distributions are equal).

D. Mean Shift Vector

The Mean Shift algorithm iteratively samples target candidate locations in an effort to find the local maximum of the similarity function $\hat{\rho}(y)$. By taking the Taylor expansion around the target candidate probability values $\hat{p}_u(\hat{y}_0)$ (where the target candidate $\hat{p}(\hat{y}_0)$ is centred around $\hat{y}_0$) the estimated linear approximation of the Bhattacharyya coefficient [4] can be described by

$$\rho[\hat{p}(y), \hat{q}] = \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u} + \frac{1}{2} \sum_{u=1}^{m} \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \tag{10}$$

The first term of (10) is independent of position y, therefore to maximize $\rho[\hat{p}(y), \hat{q}]$ it is necessary to maximize the second term of (10), using (7) the second term of (10) denoted by $\rho[\hat{p}(y), \hat{q}]_2$ can be described by

$$\rho[\hat{p}(y), \hat{q}]_2 = \frac{C_h}{2} \sum_{i=1}^{n_h} \omega_i k(\| \frac{y - x_i}{h} \|^2) \tag{11}$$

where

$$\omega_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \tag{12}$$

The Mean Shift vector is determined in order to maximize the similarity function $\hat{\rho}(y)$ by maximizing (11). The Mean Shift vector is determined by

$$Y_1 = \frac{\sum_{i=1}^{n_h} (x_i - \hat{y}_0)\omega_i g(\| \frac{\hat{y}_0 - x_i}{h} \|^2)}{\sum_{i=1}^{n_h} \omega_i g(\| \frac{\hat{y}_0 - x_i}{h} \|^2)} \tag{13}$$

Where $g(x) = k'(x)$. If we choose $k(x)$ to use the Epanechnikov profile [12] described by

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1}(d + 2)(1 - x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

the computation of (13) can be simplified as $g(x)$ becomes a constant. Different kernel profiles may be used, they however have little impact on the localization accuracy of the Mean Shift algorithm. These kernel profiles have a higher computational cost as the kernel derivative $g(x)$ must be determined for each computation of the Mean Shift vector. Using the Epanechnikov profile the Mean Shift vector can be described by

$$Y_1 = \frac{\sum_{i=1}^{n_h}(x_i - \hat{y}_0)\omega_i}{\sum_{i=1}^{n_h}\omega_i} \tag{15}$$

The updated position of the target candidate position $\hat{y}_1$ is simply described by

$$\hat{y}_1 = \hat{y}_0 + Y_1 \tag{16}$$

The Mean Shift algorithm is run recursively until convergence, convergence occurs when the Mean Shift vector is lower than a tolerance ϵ. The tolerance is usually chosen to be the width of a single pixel.

III. Image Features

Multiple Image features were used during tracking in order to better describe the appearance of the target object.

A. Local Binary Pattern Features

The local binary pattern [13,14] is an image operator which transforms an image into an array of integer labels which describe the small scale appearance of the image [14]. The LBP (local binary pattern) is an efficient texture classification method which is invariant to monotonic grey level changes. The local binary pattern was used to provide useful textural descriptive information of the target object.

The basic LBP [13] was initially designed for texture description. The basic LBP operator assigns a label to each pixel in the image. Let $z(x, y)$ describe the 3 × 3 neighbourhood surrounding a pixel. $z(x, y)$ is described by

$$z(x, y) = I(x, y) - I(x_c, y_c) \tag{17}$$

Where $I(x, y)$ represents the pixel values in the 3 × 3 neighbourhood and $I(x_c, y_c)$ represents the centre pixel in the 3 × 3 neighbourhood. Let $s(z(x, y))$ be the thresholding step function where

$$s(z(x,y)) = \begin{cases} 1 & \text{if } z(x,y) \geq 0 \\ 0 & \text{if } z(x,y) < 0 \end{cases} \tag{18}$$

The pixels surrounding the centre pixel in $s(z(x, y))$ form a binary number which is used as a label to describe the pixel. Fig. 1 shows an illustration of the basic LBP operator. A histogram of these labels can be used to describe the image.

Traditionally the histogram describing a texture or image is determined by separating uniforms patterns (such as 00000000 or 11001111) into bins. Where each unique uniform pattern has a preallocated bin and all non-uniform patterns are grouped in a single bin. There are 58 unique uniform patterns in the basic LBP and 198 non-uniform patterns [14]. In order to improve the rotational invariance of the LBP, the binary label for each pixel is circularly bit-shifted to find a minimum binary value which describes the pixel for eight possible orientations of the LBP operator. This is shown by

$$LBP_{P,R}^{r,i} = \min_i ROR(LBP_{P,R}, i) \tag{19}$$

Where $LBP_{P,R}^{r,i}$ denotes the output rotationally invariant binary label, $ROR(x, i)$ denotes the circular bitwise right rotation of bit sequence x by i steps and $LBP_{P,R}$ denotes the original basic LBP binary label.

Performing this rotation invariance step is useful in that it allows the LBP to perform robustly when rotation occurs as well as limiting the number of possible unique uniform patterns. The unique uniform patterns are reduced to the following 9 patterns 00000000, 00000001, 00000011, 00000111, 00001111, 00011111, 00111111, 01111111, 11111111 after the rotation invariance step.

The basic LBP operator with the rotation invariance step was used for each channel in the RGB colour space. A 3-dimensional RGB-LBP histogram with 10 bins per channel was formed from the 3 channels R, G and B.

B. Edge Features

Edges describe the structure of an image, edges provide beneficial descriptive information in object tracking when objects in a scene have similar colour yet different structure. A 2-dimensional edge histogram of size $N_e \times N_e$ with one channel for edge magnitude and the other for edge direction is used to describe the edge features in the target object. The simple Scharr operator [15] was used to find edges in the image as it provides efficient, robust and rotational invariant edge detection. The gradients $D_x(x, y)$ and $D_y(x, y)$ are represented by

$$D_x(x, y) = S_x \bigotimes I(x, y) \tag{20}$$

$$D_y(x, y) = S_y \bigotimes I(x, y) \tag{21}$$

Where $D_x(x, y)$ is the gradient in the x direction, $D_y(x, y)$ is the gradient in the y direction, S_x is the simple Scharr gradient operator in the x direction and S_y is the simple Scharr gradient operator in the y direction, $\bigotimes$ is the convolution operator and I(x,y) represents the intensity values in the image. The edge magnitude denoted by $D(x, y)$ and the gradient direction denoted by $\theta(x, y)$ are represented by

$$D(x, y) = \sqrt{D_x(x, y)^2 + D_y(x, y)^2} \tag{22}$$

$$\theta(x, y) = \arctan(\frac{D_y(x, y)}{D_x(x, y)}) \tag{23}$$

Where $\theta(x, y)$ is determined between edges directions $0^\circ \leq \theta(x, y) < 360^\circ$. Edges were filtered such that only edges with magnitudes above a threshold t_e were considered in the edge feature histogram.

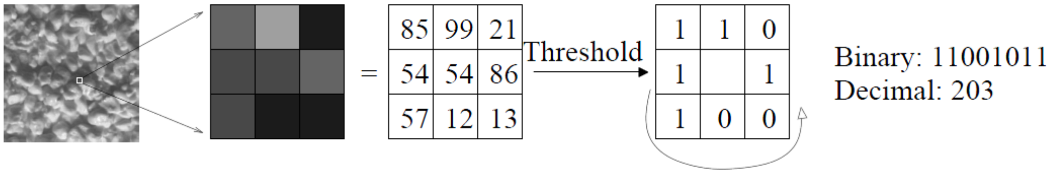


Fig. 1: Local Binary Pattern Operator

C. Colour Features

Colour histograms are most commonly used in conjunction with the Mean Shift algorithm as they are robust to partial occlusion and change in scale and rotation. They perform well under non-rigid deformations of the target object and changing complex backgrounds [4,12]. Colour histograms do however fail when other objects or background features have the same or similar colour. A 3-dimensional RGB colour histogram of size $N_c \times N_c \times N_c$ was used to describe the RGB colour distribution of the target object. A 1-dimensional Hue (from the HSV colour space) colour histogram of size N_h was used to describe the Hue colour distribution of the target object. The Hue histogram is useful as it is largely illumination invariant.

D. Colour and Edge Features

Colour and Edge features where combined in an effort to combine structural and colour information in a single histogram. Edges were found using the simple Scharr operator. The greyscale gradient magnitude $D(x,y)$ was determined for each pixel in the target object region. The pixel value $I_i(x,y)$ for each RGB channel is determined by.

$$I_i(x,y) = \begin{cases} I_i(x,y) + D_i(x,y) & \text{if } D(x,y) \leq t_e \\ I_i(x,y) - D_i(x,y) & \text{if } D(x,y) > t_e \end{cases} \tag{24}$$

Where $I_i(x,y)$ is i'th RGB channel value for the pixel $I(x,y)$ and $D_i(x,y)$ is the gradient magnitude for the RGB channel i. A 3-dimensional colour-edge histogram of size $N_c \times N_c \times N_c$ was used to describe $I_i(x,y)$.

Let σ denote the scale of the target object. Due to the elliptical shape of the target region, typically both background and object features are present in the target region of scale σ [10]. Background features in the target model can have an effect on the localization accuracy of the tracking algorithm. In order to minimize this effect 3 colour-edge histograms were used to describe the target object. The 3 colour-edge histograms were determined for target regions of scales σ, 0.8σ and 0.6σ. Histograms formed from target regions smaller than the scale of the object are less likely to contain background features.

E. Background Weighted Colour Features

If some background features are present in the target model and candidate, the localization performance would be improved if the background feature information in the target model and target candidate was suppressed. This is done by weighting the target model and target candidate with a background model at each frame such that the target object has a more salient description relative to the background [4].

Let $\hat{o}(y)$ denote the background model centred around y. Let $\{x_i\}_{i=1\ldots n_h}$ denote the n_h normalized pixel locations in the background model region which are centred around y. Let $a(x)$ denote a concave, monotonically increasing, isotropic kernel with a size (based on the scale of the target object) specified by bandwidth h. The probability of the feature $u = 1 \ldots m$ in the background model histogram is determined by

$$\hat{o}_u(y) = C_h \sum_{i=1}^{n_h} a(\| \frac{y - x_i}{h} \|^2)\delta[b(x_i) - u] \tag{25}$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} a(\| \frac{y-x_i}{h} \|^2)} \tag{26}$$

The background kernel used is described by

$$a(x) = \begin{cases} 0 & \text{if } x \leq 1 \\ x - 1 & \text{if } 1 < x \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Where 1 represents the boundary of the target model or candidate region and 2 represents the boundary of the background model region. The kernel $a(x)$ assigns weights to pixels such that features further from the object boundary have a higher weighting. Let $\hat{o}^*$ denote the smallest non-zero histogram bin in the background histogram $\hat{o}(y)$. The scaling array v_u [4] used to minimize similar features between the background model and the target model and candidate is described by

$$\{v_u = \min(\frac{\hat{o}^*}{\hat{o}_u}, 1)\}_{u=1\ldots m} \tag{28}$$

The background weighted target model $\hat{q}_u$ and target candidate $\hat{p}_u(y)$ are represented by

$$\hat{q}_u = C v_u \sum_{i=1}^{n} k(\| x_i^* \|^2)\delta[b(x_i^*) - u] \tag{29}$$

where

$$C = \frac{1}{\sum_{i=1}^{n} k(\| x_i^* \|^2) \sum_{u=1}^{m} v_u \delta[b(x_i^*) - u]} \tag{30}$$

$$\hat{p}_u(y) = C_h v_u \sum_{i=1}^{n_h} k(\| \frac{y - x_i}{h} \|^2)\delta[b(x_i) - u] \tag{31}$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\parallel \frac{y-x_i}{h} \parallel^2) \sum_{u=1}^{m} v_u \delta[b(x_i^*) - u]} \tag{32}$$

A 3-dimensional background weighted colour histogram of size $N_c \times N_c \times N_c$ was used to describe a more salient RGB colour representation of the target object.

IV. Target Object Localization

A. Feature Localization Weights

Each feature determines an updated target object position $\hat{y}_1$ using the Mean Shift localization algorithm. To determine the best estimation of the target object's updated position, a weighted average is determined of the updated target object positions determined by the various features. The updated target object position $\hat{y}_1$ is determined by

$$\hat{y}_1 = \sum_{j=1}^{K_f} \omega_j \hat{y}_{1_j} \tag{33}$$

Where ω_j denotes the localization weight for feature j, $\hat{y}_{1_j}$ denotes the updated target object position for feature j and K_f denotes the number of features. The feature weights are determined from 3 global weights, The global weights consist of predetermined feature weights, model-candidate similarity feature weights and model-background similarity feature weights.

The global model-candidate similarity feature weight determines a weight based on the similarity function between the target model and target candidate. The higher the similarity, the higher the weight associated with the feature. The model-candidate similarity feature weight ω_c is described by

$$\omega_{c_j} = \frac{1}{(1 - \rho[\hat{p}_j(y), \hat{q}_j])(C_c)} \tag{34}$$

where

$$C_c = \sum_{j=1}^{K_f} \frac{1}{(1 - \rho[\hat{p}_j(y), \hat{q}_j])} \tag{35}$$

Where ω_{c_j} denotes the model-candidate similarity feature weight for feature j, $\hat{q}_j$ denotes the target model for feature j and $\hat{p}_j(y)$ denotes the target candidate for feature j. The global model-background similarity feature weight determines a weight based on the similarity function between the target model and background model. The higher the similarity, the lower the weight associated with the feature. The model-background similarity feature weight ω_b is described by

$$\omega_{b_j} = \frac{\omega_{p_j}}{(\rho[\hat{o}_j(y), \hat{q}_j])(C_b)} \tag{36}$$

where

$$C_b = \sum_{j=1}^{K_f} \frac{\omega_{p_j}}{(\rho[\hat{o}_j(y), \hat{q}_j])} \tag{37}$$

Where ω_{b_j} denotes the model-background similarity feature weight for feature j, $\hat{o}_j(y)$ denotes the background model for feature j and ω_{p_j} denotes predetermined feature weight for feature j. The localization weight ω_j for the feature j is determined by

$$\omega_j = \alpha\omega_{c_j} + \beta\omega_{b_j} + \gamma\omega_{p_j} \tag{38}$$

where

$$\alpha + \beta + \gamma = 1 \tag{39}$$

Where α, β and γ are constants which specify the relationship between the various global weights and the feature weights. The features weights are normalized such that $\sum_{j=1}^{K_f} \omega_j = 1$.

B. Scale Selection

It is necessary to determine the scale of the target object to effectively track it through out a video sequence. Image moments [16,17] are used to determine the scale of the target object in this algorithm, a similar approach is used by [8] and [18]. In [18] (CAMSHIFT) the scale and orientation is determined using image moments on a skin probability back projection. In [8] (SOAMST) the traditional kernel-based Mean Shift object tracking algorithm is used, the similarity weights ω_i (12) are used as a probability back projection. Image moments are used with the similarity weights to determine the scale and orientation of the target object. A similarity area estimation is used to correctly determine the target object's scale.

In this algorithm the similarity weights ω_i are determined for each pixel in the target region with a scale 1.2σ. Image moments are then used in conjunction with the similarity weights to determine the scale of target object in the current frame. The similarity weights ω_i are determined by

$$\omega_i = \sum_{j=1}^{K_f} \omega_j \omega_{i_j} \tag{40}$$

Where ω_{i_j} denotes the similarity weight determined by (12) for feature j. The zeroth order moment denoted by M_{00} is determined by

$$M_{00} = \sum_{i=1}^{n_h} \omega_i \tag{41}$$

Where n_h is the number of pixels in the target region with a scale 1.2σ. The second order moments denoted by M_{20}, M_{02} and M_{11} are determined by

$$M_{20} = \sum_{i=1}^{n_h} \omega_i x_{i,1}^2 \tag{42}$$

$$M_{02} = \sum_{i=1}^{n_h} \omega_i x_{i,2}^2 \tag{43}$$

$$M_{11} = \sum_{i=1}^{n_h} \omega_i x_{i,1} x_{i,2} \tag{44}$$

Where $x_{i,1}$ denotes the i'th x value in the target region with a scale 1.2σ and $x_{i,2}$ denotes the i'th y value in the target region with a scale 1.2σ. The second order central moments denoted by μ_{20}, μ_{02} and μ_{11} are determined by

$$\mu_{20} = \frac{M_{20}}{M_{00}} - \bar{x}_1^2 \tag{45}$$

$$\mu_{02} = \frac{M_{02}}{M_{00}} - \bar{x}_2^2 \tag{46}$$

$$\mu_{11} = \frac{M_{11}}{M_{00}} - \bar{x}_1\bar{x}_2 \tag{47}$$

Where $\bar{x}_1$ is the target object's centre x position and $\bar{x}_2$ is the target object's centre y position. The second order central moment covariance matrix donated by Cov is represented by

$$Cov = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \tag{48}$$

The eigenvalues of the covariance matrix represent the size of the axis a and b of the target object region. Half the height of the target object is determined by b and half the width is determined by a, they are represented by

$$a = \frac{\mu_{20} + \mu_{02}}{2} - \frac{\sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2} \tag{49}$$

$$b = \frac{\mu_{20} + \mu_{02}}{2} + \frac{\sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2} \tag{50}$$

It is assumed that the scale change between frames is relatively small, to get a more accurate and smooth scale change between frames the target height and width is determined by

$$a = (\zeta)a_p + (1 - \zeta)a_n \tag{51}$$

$$b = (\zeta)b_p + (1 - \zeta)b_n \tag{52}$$

Where (ζ) denotes a constant which determines the rate at which the target object's scale should change, a_p denotes half the object width determined in the previous frame, a_n denotes half the object width determined in the current frame, b_p denotes half the object height determined in the previous frame and b_n denotes half the object height determined in the current frame.

C. State Estimation

The Mean Shift algorithm is not well suited for tracking objects in the presence of full occlusions. In order to improve the performance of the Mean Shift tracking algorithm in the presence of partial and full occlusions a kalman filter [19,20] is used. A kalman filter is a state estimation algorithm which compares state prediction against state measurements to get an accurate estimation of the true state.

The state prediction matrix F in $X_k = FX_{k-1} + v_k$ is determined using simple equations of motion for position, velocity and acceleration. The state prediction matrix also called the system matrix is represented by

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{53}$$

For the kalman filter to perform accurately the measurement noise n_k [20] needs to be estimated. The measurement noise n_k is used to estimate how reliable the measurements are in $z_k = HX_k + n_k$. An accurate estimation of the measurement noise is necessary in order to minimize the effect of inaccurate target object localization during occlusion. The measurement noise n_k is determined relative to the similarity between the target model and the target candidate, the more similar the target model and candidate, the more accurate the measurement. The measurement noise n_k is described by

$$\{n_k = 10^l(1 - \rho[\hat{p}(y), \hat{q}]_c) \quad \text{if } \epsilon_{l+1} < \rho[\hat{p}(y), \hat{q}]_c \leq \epsilon_l\}_{l=0\ldots3} \tag{54}$$

where

$$\rho[\hat{p}(y), \hat{q}]_c = \sum_{j=1}^{K_f} \rho[\hat{p}_j(y), \hat{q}_j]\omega_j \tag{55}$$

Where $\epsilon_{l\{l=0\ldots3\}}$ are constants which specify the bounds of the piecewise measurement noise estimation function. It is assumed that a target object's velocity is constant during occlusion. Using this assumption in order to improve the tracking performance during occlusion, the current state matrix velocity is updated every frame with the target object's weighted average velocity V_{a_k} represented by

$$V_{a_k} = 0.85V_{a_{k-1}} + 0.15((1 - V_{n_k})V_k + V_{n_k}V_{a_{k-1}}) \tag{56}$$

Where V_{n_k} is the velocity noise at frame k determined by

$$\{V_{n_k} = 0.2l \quad \text{if } \epsilon_{l+1} < \rho[\hat{p}(y), \hat{q}]_c \leq \epsilon_l\}_{l=0\ldots3} \tag{57}$$

The state matrix velocity X_{k_v} is updated with the weighted average velocity such that $X_{k_v} = 0.85V_{a_k} + 0.15X_{k_v}$.

D. Tracking Algorithm Overview

Using the methods described in sections II, III and IV, the tracking algorithm can be summarized as follows

1) Determine target model $\hat{q_j}$ for features $1 \ldots j$
2) Initialize iteration number $k_i \leftarrow 0$
3) Initialize position y_0 of candidate target in current frame
4) Determine candidate target $\hat{p_j}(y_0)$ for features $1 \ldots j$
5) Calculate feature localization weights w_j for features $1 \ldots j$
6) Calculate similarity weights ω_{i_j} for features $1 \ldots j$
7) Calculate combined similarity weights ω_i

8) Determine updated target object position y_1
9) If $\parallel y_1 - y_0 \parallel < \epsilon$ (where $\epsilon < 1$) or if $k \geq N$ (where N is chosen to be 20) stop. Go to step 10)
Otherwise $k_i \leftarrow k_i + 1$ and $y_0 \leftarrow y_1$. Go to step 4)
10) Determine height $2b$ and width $2a$ of target object
11) Update target object states using kalman filter, this includes updating object position. Determine y_0 for the next frame using state prediction matrix F
12) Load next frame, go to step 2)

V. Experimental Results

The proposed algorithm's performance is compared to the original Mean Shift tracking algorithm with variable scale selection in [4] and the SOAMST algorithm in [8]. These algorithms were selected to use $64 \times 64 \times 64$ RGB colour histograms, the algorithms in [4] and [8] were implemented using the same kalman filter implementation used in the proposed tracking algorithm. The algorithms were tested on a complex scene (video sequence: motinas_multi_face_frontal.avi, frames: 1 - 300, target: Emilio) [21]. A persons face (Target: Emilio) was tracked in a complex environment with partial and full occlusions, change in scale, change in illumination and slight change in the appearance of the target object. During the video sequence the target's face is fully occluded by the face of a person (target: Joe, frames: 88 - 95). There is a rapid change in scale of the target (frames: 250 - 300) and a change in illumination experienced by the target (frames: 196 - 275).

The tracking performance of the algorithms can be observed from Fig. 2 (visual description of tracking performance for proposed algorithm, original Mean Shift tracking algorithm and SOAMST algorithm) and Fig. 3 (graphs describing position and scale selection error from ground truth). The original Mean Shift object tracking algorithm shows good performance in tracking the target object, however once occlusion occurs the tracker diverges, the algorithm does not benefit greatly from the kalman filter implementation. The SOAMST algorithm shows good performance in tracking the target object, however the algorithm selects the scale of the target object abruptly and inaccurately. Like the original Mean Shift algorithm the SOAMST algorithm diverges when occlusion occurs and does not benefit greatly from the kalman filter implementation. The proposed algorithm shows good performance in tracking the target object through out the video sequence. The algorithm localizes the target object inaccurately during occlusion, however the algorithm does not diverge during occlusion. The proposed tracking algorithm benefits greatly from the kalman filter implementation in minimizing the effect of object occlusion.

VI. Conclusion

A tracking algorithm using the Mean Shift framework is presented which performs robustly in complex scenes where occlusion occurs. The algorithm uses multiple features to uniquely describe objects, image moments to effectively determine the target object's scale and a kalman filter to aid the localization algorithm during occlusion. The algorithm has shown superior tracking performance in complex scenes when compared to the original Mean Shift tracking algorithm and the scale adaptive SOAMST algorithm.

References

[1] K. Fukunaga and L. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," in *IEEE IT*, vol. 21, no. 1, pp. 32 - 40, 1975.

[2] Y. Cheng "Mean Shift, Mode Seeking, and Clustering," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790 - 799, 1995.

[3] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," in *International Conference on Computer Vision*, vol. 2, pp. 1197 - 1203, 1999.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564 - 577, May 2003

[5] J. Wang and Y. Yagi, "Integrating Shape and Color Features for Adaptive Real-time Object Tracking," in *IEEE International Conference on Robotics and Biomimetics*, pp. 1 - 6, 2006

[6] A. Babaeian, S. Rastegar, M. Bandarabadi and M. Rezaei, "Mean Shift-Based Object Tracking with Multiple Features," in *41st Southeastern Symposium on System Theory*, pp. 68 - 72, March 2009

[7] R. T. Collins, "Mean-shift blob tracking through scale space," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 234 - 240 2003

[8] J. Ning, L. Zhang1, D. Zhang and C. Wu, "Scale and Orientation Adaptive Mean Shift Tracking," in *Computer Vision, IET*, vol. 6, iss. 1, pp. 52 - 61, 2012

[9] S. Zhang and Y. Bar-Shalom, "Robust Kernel-Based Object Tracking with Multiple Kernel Centers," in *12th International Conference on Information Fusion*, pp. 1014 - 1021, July 2009

[10] A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 - 6, 2007

[11] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data," in *Kybernetika*, vol. 34, no. 4, pp 363 - 368, 1998.

[12] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603 - 619, May 2002.

[13] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," in *Pattern Recognition*, vol. 29, no. 1, pp. 51 - 59, 1996.

[14] M. Pietikinen, A. Hadid, A. Zhao and T. Ahonen "Local Binary Patterns for Still Images" in *Computer Vision using Local Binary Patterns*, 2011, 2011, XV, 209 p. 87 illus., 56 in color, pp 13 - 43

[15] B. Jhne, H. Scharr, and S. Krkel, "Principles of filter design," in B. Jhne, H. Hauecker, and P. Geiler, editors *Handbook of Computer Vision and Applications,*, vol. 2, pp 125 - 151. Academic Press, 1999.

[16] F. Chaumette, "Image Moments: A General and Useful Set of Features for Visual Servoing," in *IEEE Transactions on Robotics*, vol. 20, no. 4, pp 713 - 723. August 2004

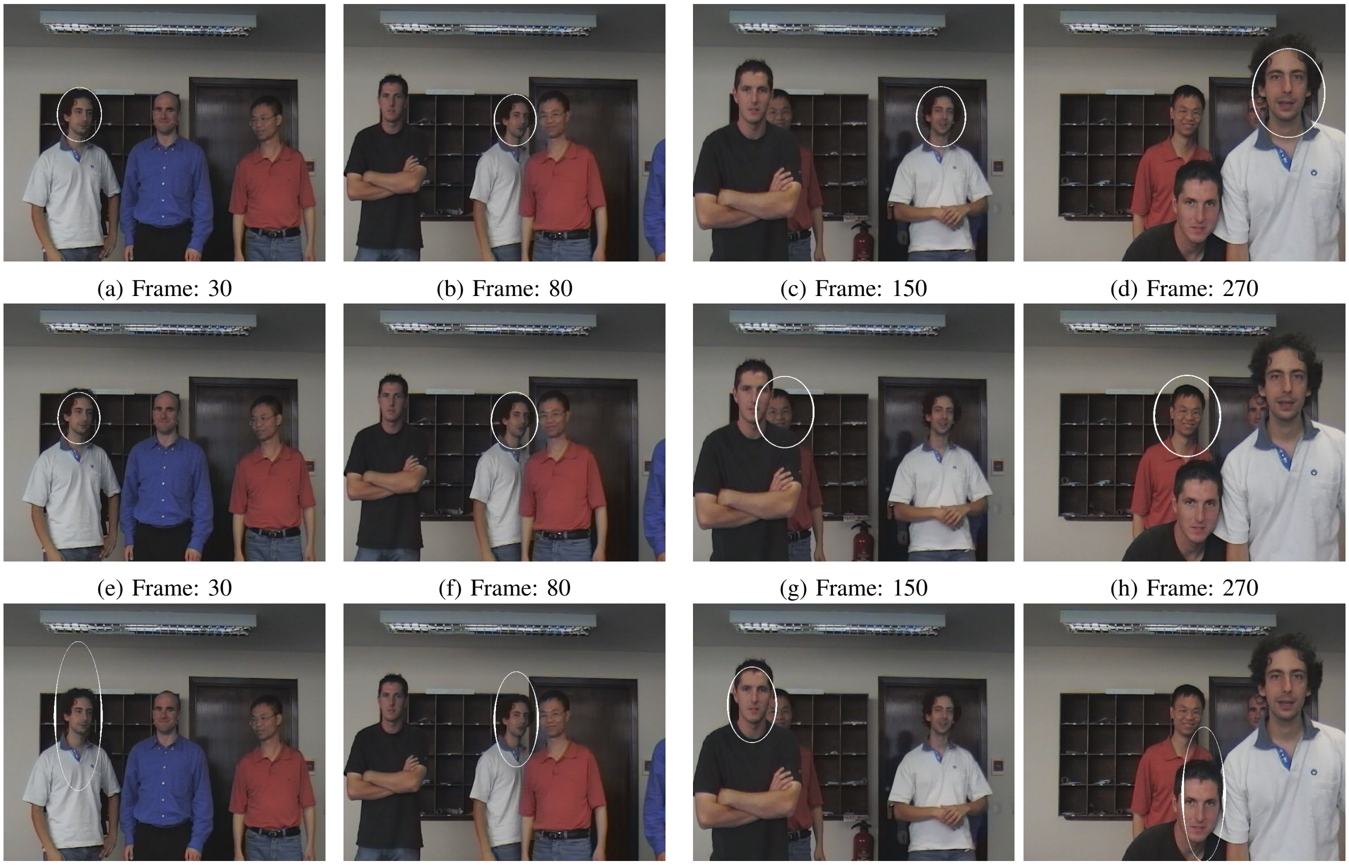
[17] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis: Theory and Applications*, World Scientific, Singapore, 1996.

[18] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," in *Intel Technology Journal*, 2(Q2), pp. 1-15, 1998.

[19] G. Welch and G. Bishop, SIGGRAPH 2001, Course 8, Topic: *An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC 27599-3175, 2001

[20] K. Nickels and S. Hutchinson, "Estimating Uncertainty in SSDBased Feature Tracking," in *Image and Vision Computing*, vol. 20, pp. 47-58, 2002.

[21] E. Maggio, E. Piccardo, C. Regazzoni and A. Cavallaro, "Particle PHD filter for multi-target visual tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*,(ICASSP 2007), Honolulu (USA), April 15-20, 2007

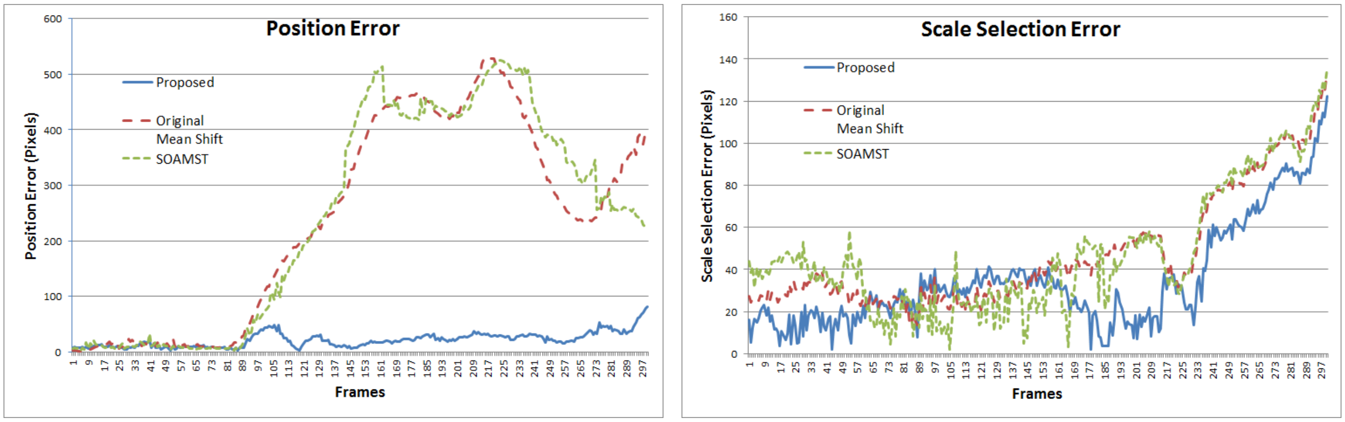


(a) Frame: 30 (b) Frame: 80 (c) Frame: 150 (d) Frame: 270

(e) Frame: 30 (f) Frame: 80 (g) Frame: 150 (h) Frame: 270

(i) Frame: 30 (j) Frame: 80 (k) Frame: 150 (l) Frame: 270

Fig. 2: Proposed algorothm (a - d), Original Mean Shift algorothm (e - h), SOAMST (i - l)



(a) Position Error from Ground Truth

(b) Scale Selection Error from Ground Truth

Fig. 3: Tracking Error from Ground Truth