Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter

Oscar Efrain Ramos Ponce, Mohammad Ali Mirzaei, Frédéric Merienne

Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter

Oscar E. Ramos, Mohammad Ali Mirzaei, Frédéric Merienne

***Abstract*— The classical mean shift algorithm for tracking in perfectly arranged conditions constitutes a good object tracking method. However, in the real environment it presents some limitations, especially under the presence of noise, objects with varying size, or occlusions. In order to deal with these problems, this paper proposes a reliable object tracking algorithm using mean shift and the Kalman filter, which was added to the traditional algorithm as a predictor when no reliable model of the object being tracked is found. Experimental work demonstrates that the proposed mean shift Kalman filter algorithm improves the tracking performance of the classical algorithms in complicated real scenarios. The results involve the tracking of an object in a gray level and in a color sequence, with varying size and in presence of total occlusion.**

I. Introduction

Tracking is an important technique for a variety of applications, ranging from military such as anti-aircraft and missiles shelter defense systems to very commercial cases like unmanned autopilot navigation, monitoring and security. The main concern in all the applications is to find the object of interest, which is generally called the target, and then to follow it using vision systems. The goal of object tracking in a video stream is to continuously and reliably determine the position of an object against dynamic scenes with presence of noise [1]. To this sake, numerous sophisticated algorithms have been proposed and implemented. For example, considering Gaussian and linear problems, Welch and Bishop [2] proposed a Kalman filter for tracking a user's pose for interactive interface with virtual environments. The proposed single-constraint-at-a-time (SCAAT) tracking method fused the measurements of different optical sensors in order to improve the tracking accuracy and stability. Proposed schemes in [3][4] recruited the posterior probability distribution over some scene properties of interest, based on image observations to improve the functionality of object tracking under real working conditions. Other tracking strategies can also be found as Multiple Hypothesis Tracking [5][6], kernel-based tracking [7][8], and tracking based on optical flow [9].

A scale invariant feature transform, known as SIFT [10], based on the mean-shift algorithm was presented for object tracking in real scenarios. SIFT features were widely used in this method to find correspondences in the regions of interest across the frames. Meanwhile, mean-shift was used to find regions of interest via color histograms [11]. To improve the performance of this technique in complex scenes, [12] proposed a new algorithm for optimally adapting the ellipse outlining the objects of interest. The target is represented by a Gaussian mixture model in a joint feature-spatial space, with each ellipsoid corresponding to a different fragment. These fragments are automatically adapted to the image data, then selected by a region-growing procedure and updated according to a weighted average of the past and present image statistics. The target and background are modeled in a Chan-Vese manner using the framework of level sets to preserve accurate boundaries of the target [13]. The Active Shape Model (ASM) has been widely used to recognize and track a face from a video sequence. Since this method is computationally heavy, another complementary research [14] proposed an enhanced ASM and predicted mean-shift algorithm to meet these challenges, which combines the context information and predicts mean-shift to obtain multi-angle start shapes for ASM searching and the best result shape is chosen based on a matching evaluation.

In this paper, the target object is identified with a bounding box in the first frame using image processing techniques. After that, the histogram model is used as a feature to estimate the motion vector. By having this motion vector, the bounding box is moved to a new place to indicate the motion of the target and to localize its position. For color images, color model is applied to extract this feature but the method remains similar. In order to cope with the different size of the target object, the scale needs to be found and the tracking system rescaled. Moreover, to deal with noisy situations and total occlusion, the Kalman filter was added to the mean-shift approach improving the reliability of the current tracking algorithm.

This paper is organized as follows. Section II describes the region of interest selection using image processing techniques. The mean-shift algorithm is briefly recalled in section III. Section IV adapts the technique to handle tracking when the object has different scales. Color information is added to the algorithm in section V. Finally, section VI combines the Kalman filter with the algorithm to deal with total occlusion.

II. Preparation of the data for the algorithm

Each image will be considered as a 3 dimensional matrix $\mathbb{N}^3 \rightarrow \mathbb{R}$ with the form $I(x, y, t)$. The two first components of the domain $x \in [0, H]$, $y \in [0, W]$ correspond to the pixel location (conventionally starting from the top left corner), and the third one $t \in [1, N_{img}]$ to the image sequence number (associated with the time), where H represents the height of the image, W its width and N_{img} the total number

O. Ramos is with LAAS-CNRS, University of Toulouse III (UPS), 7 av. du Colonel Roche, 31077 Toulouse, France

A. Mirzaei is with ParisTech, Image Institute ENSAM, CNRS, Le2i Lab, 2 rue Thomas Dumorey, 71100 Chalon-sur-Saône, France

F. Merienne is with the Image Institute, ENSAM, Le2i Lab, 2 rue Thomas Dumorey, 71100 Chalon-sur-Saône, France

of images in the sequence. In this work, the range is reduced to the set of gray levels $I(x, y, t) \in [0, 255]$, but it is evident that any other representation would be easily applied.

A. Background subtraction

In order to obtain the background from the set of given images, the median of each pixel in time is performed. The assumptions are that there is no motion in the camera or background from image at time t to image at time $t+1$, and that the objects not constituting the background are moving along the scene and appear in different positions in time, that is, the moving object appears only for a short period of time in a given location. Then, the background image for a certain (x, y) pixel position, $I_{back}(x, y)$, is given by

$$I_{back}(x, y) = med\{I(x, y, 1), I(x, y, 2), ..., I(x, y, N_{img})\}$$

where $med\{.\}$ represents the median operation.

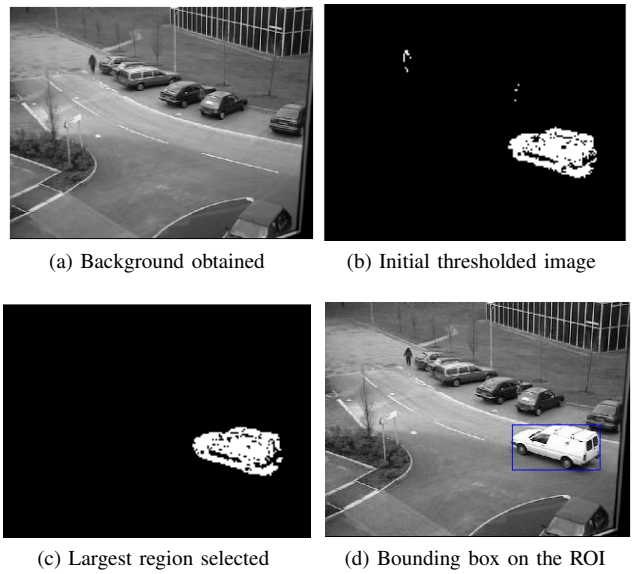
B. Region of Interest

We assume that the first image presents the object that has to be tracked. Then, the absolute value of the difference between the first image and the background, computed as

$$I_{diff}(x, y) = |I(x, y, 1) - I_{back}(x, y)| \tag{1}$$

highlights the objects that are different to the background. To find the region of interest (ROI), I_{diff} can be binarized applying a threshold to it. However, due to noise or to the existence of minor objects in first image (which were not present in the background), some small undesired regions not constituting the ROI might appear. In order to eliminate them, morphological operations can be applied.

To illustrate this, a sequence of images of a street with a car and some people occasionally walking by will be used. The objective is to track the car. The background obtained is shown in Figure 1a. The result after thresholding is shown in Figure 1b. However, there are some undesired elements to be eliminated, which have a small number of constituting pixels. Then, a criteria for their elimination was to first label the connected elements using 8-connectivity, leading to several connected regions, and then keeping only the region with most connected elements, that is, the largest region. This region constitutes the desired ROI and corresponds to the car. To improve the result, the thresholded image can be first closed applying successively a dilation and then an erosion. Figure 1c shows the result after keeping only the largest region. The maximum and minimum positions of this region were determined and used as the limits of the bounding box of the ROI. The width w_r and height h_r of the region were computed, as well as its center (x_c, y_c), which is the mean of the extremes in x and in y. The bounding box is shown in Figure 1d.



(a) Background obtained (b) Initial thresholded image

(c) Largest region selected (d) Bounding box on the ROI

Fig. 1: Example of preparation of the data

III. Basic Mean-Shift algorithm

The algorithm described in this section corresponds basically to the one in [15], [16]. First, the grey levels for each one of the pixels in the ROI have to be quantized into m bins. Let the number of pixels in the ROI be n. For a certain pixel $X_i = (x_i, y_i)$ in the ROI, where $i \in [1, n]$, its quantized value or bin number, will be $b(X_i)$. A new image containing only this values for the ROI, at any time t, is created as:

$$b(X_i) = b(x_i, y_i) = \left\lfloor \frac{I(x_i, y_i, t)}{256/m} \right\rfloor \tag{2}$$

where $\lfloor . \rfloor$ represents the floor function (largest integer not greater than f). Using this formulation, $b(X_i)$ will have values ranging from 0 to $m-1$. The target model q is a discrete m-bin histogram obtained from the ROI in the first image and containing m elements: $q = [q_1, q_2, ..., q_m]$. Each element is given by [7]:

$$q_{u+1} = C \sum_{i=1}^{n} k \left\| \frac{X_i - X_c}{h} \right\|^2 \delta(b(X_i) - u) \tag{3}$$

where $u \in [0, m-1]$, X_c are the coordinates of the center of the ROI, C is a normalization factor so that q sums up to 1, h is half of the size of the window, and k is the Epanechnikov kernel defined by

$$k(x) = \begin{cases} 0.5 C_d^{-1}(d+2)(1-x) & , x < 1 \\ 0 & , x > 1 \end{cases} \tag{4}$$

Since the image lies in a 2 dimensional space, $d = 2$ and $C_d = \pi$, the volume of a unitary circle. Note that the subindex of q is $u+1$ instead of u. This change in the notation was necessary since the values of the histogram $b(X_i)$ start with 0, whereas the elements of q, q_i start with 1.

For the computation, first the center of the ROI, X_c, is calculated with the values of q initialized to zero, and then the pixels belonging to the ROI are read in order. For each pixel, the square of the centered and normalized magnitude is computed in this case as:

$$\left\| \frac{X_i - X_c}{h} \right\|^2 = \left(\frac{x_i - x_c}{h_r/2} \right)^2 + \left(\frac{y_i - y_c}{w_r/2} \right)^2 \tag{5}$$

where w_r and h_r are the width and height of the region, as described previously. This value is the argument of the k function. Then, the value of k is added to the element of the target model q_{u+1} considering that $b(X_i) = u$. Finally, after the whole computation, q is normalized to sum up to 1, like a probability distribution.

For the ROI in the next frame (the target candidate), the color model is computed exactly in the same way, but now it is named $p(Y_c)$ instead of q. The center of the new ROI is Y_c referring to the fact that now the central point of the patch is going to be changed in order to find a good similarity with the target model. Each one of the pixels in the new frame is called $Y_i = (x_i, y_i)$. To evaluate how this new color model matches the previous one, the Brattacharyya coefficient ρ is calculated as:

$$\rho(q, p(Y_c)) = \sum_{u=1}^{m} \sqrt{q_u p_u(Y_c)}. \tag{6}$$

The weight for each pixel Y_i in the ROI is then:

$$w_i = w(Y_i) = \sum_{u=1}^{m} \delta(b(Y_i) - u) \sqrt{\frac{q_u}{p_u(Y_c)}} \tag{7}$$

which states that for each pixel, the root square is taken only for the component of p and q that corresponds to the value of $b(Y_i)$; all the other other components are neglected by the δ function. The new location Z is:

$$Z = \frac{\sum_{i=1}^{n} Y_i w_i g(\|(Y - Y_i)/h\|^2)}{\sum_{i=1}^{n} w_i g(\|(Y - Y_i)/h\|^2)} \ . \tag{8}$$

The function $g(x) = -k'(x)$ and in this case, since the kernel $k(x)$ depends linearly on x, its derivative is a constant and can be omitted from the previous equation as it appears in both the numerator and denominator and will be then canceled after the division. However, the fact that it is different to zero only when $x < 1$ has to be taken into account. After obtaining the new position of Z inside the patch, with reference to its center, it has to be translated to the position in the image, which can be easily done by adding it to the coordinates of the center of the patch in the image frame. Doing this process iteratively yields to the new center of the patch, which corresponds to the desired center of the tracked object.

Summarizing, the basic algorithm is as follows:

1) Compute the target model using (3)
2) Initialize the patch (ROI) for the new frame in the position of the previous frame, Y, and obtain its model $p(Y)$.
3) Compute the distance ρ given by (6).
4) Calculate the new location Z applying the mean shift, using (8).
5) Compute the new values of $p(Z)$ and ρ.
6) If the new distance is smaller than the previous one, that is $\rho[p(Z), q] < \rho[p(Y), q]$, change the value of Z to $0.5(Y + Z)$.
7) If $\|Z - Y\|$ is small, stop; otherwise assign Z to Y and go to step 2.

The loop can be controlled for a number of fixed iterations before it reaches a small difference in the last step in order to make the process computationally less expensive. This approach, although faster, sacrifices the accuracy leading to bounding boxes slightly less well localized in the image, but the tracking still works.

The results of the tracking process for different positions of the object is shown in Figure 2. The size of the patch does not change from frame to frame, but the size of the car decreases as the car goes further in the image. Nevertheless, the algorithm is still able to track the car, even though the size of the new car is smaller than the initial one.

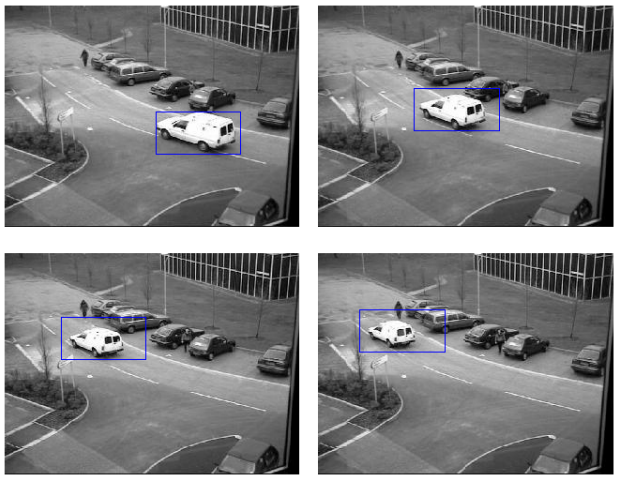


Fig. 2: Tracking using the same ROI size

IV. Handling the varying size of the object

The simplest way to handle the tracking of an object whose size varies in time is to define different scales and then compare the size of the current bounding box with the previous one to find the scale down/up between two consecutive windows. The usage of the γ (gamma) distribution function, called γ-normalization method, was introduced by [17]. The γ-normalization function is

$$f_\gamma(\theta, V) = |V|^{\gamma/2} f(\theta, V) \tag{9}$$

where θ represents the mean and γ the variance. This function is directly linked with the position of the local patches which are defined by the mean.

The method followed here is the one proposed in [7]. It consists in pre-computing the color distribution for the target candidate at three different scales in every frame. Then, the one that gives the highest Bhattacharyya coefficient for the whole iteration procedure in that frame is selected keeping that scale as the initial one for the next frame. Let H_{old} be the original size (height or width) of the ROI. The three test sizes are computed as H_{old}, $(1 + \Delta)H_{old}$ and $(1 - \Delta)H_{old}$. From them, the one that gives the largest coefficient, H_M, is selected, and the new size of the window is given by:

$$H_{new} = \Delta H_M + (1 - \Delta) H_{old} \tag{10}$$

where Δ is the incremental size of the window, typically 0.1. This new size is used for the computation inside the iteration, and it is also used as ROI input size for the next frame.

The results obtained using the scale adaptation are shown in Figure 3. Comparing these results with Figure 2, the improvement in the scale selection can be appreciated, and is particularly noticeable for the last images.

Fig. 3: Tracking using an adaptive scale for the ROI

V. Tracking with color images

The procedure described in the previous sections can be also applied to color images. However, some modifications are necessary to handle the color information. The results will also depend on how the color distribution is modeled. There are many possibilities and this section will present two examples.

The first sequence used consists on the top view of a toy car moving in a static background. In this case, for background extraction, the median was taken for each plane independently of the others, and then the results were "added" in order to generate again the color background. The segmentation is based in morphological operations in a similar way as for the gray level images: thresholding the image and then keeping the largest connected region. But, the difference of the first image and the background was computed in each of the R, G and B planes independently. This led to three "segmented" images for each image plane, which were then combined by simple addition.

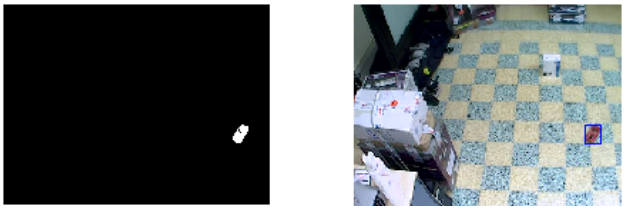


Fig. 4: Segmented region for the toy car image

Here, the approach to mean-shift algorithm is to compute the histogram in each image channel independently and then increase the size of the q and p vectors to $3m$, where m is the number of bins. This approach simplifies the problem, without using explicitly the color histogram, but profiting somehow the color information. The rest is kept as for the gray scale case. Some of the results are shown in Figure 5. The first images show that mean shift is robust enough to deal with partial occlusions, which were present when the toy car passed partially behind the box, making its lower part disappear from the field of view of the camera. However, when there is complete occlusion (toy car completely behind the box), the tracking system fails and the bounding box gets "lost": it remains in the place where the toy disappeared as a consequence of the complete occlusion.

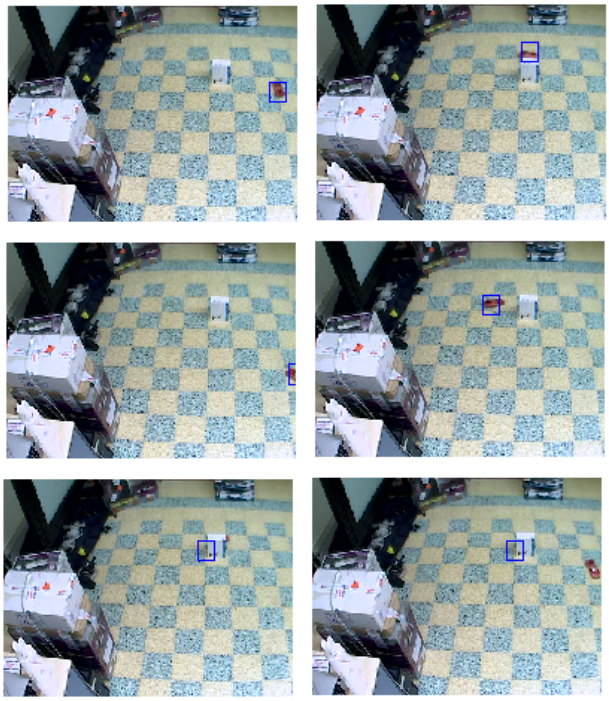


Fig. 5: Tracking a toy car. The time evolution is left to right, and top to down. The car is tracked even after partial occlusion, but it fails after complete occlusion (bottom images)

The second color sequence is the top view of a plane with the camera moving with a similar velocity of the plane, so that it almost always remains in the bottom part of the image. The background composed of clouds changes as the plane moves, and then, it cannot be correctly estimated using the median. Then, background subtraction cannot be applied. The approach here was to use the *region growing* method for segmentation. That is, the initial image was segmented without considering the sequence background at all. After this, it was binarized, and the extremes of the biggest region (corresponding to the plane and discarding the sky) were obtained. The next steps were exactly as in the previous toy car case. Using the RGB color space, acceptable results were obtained, but the bounded region was not very stable. Then, the HSV space was intended as a solution. Some images obtained using the HSV space are shown in Figure 6.

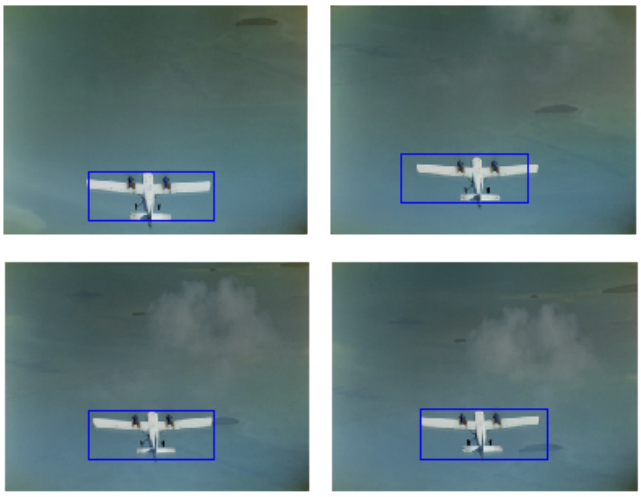


Fig. 6: Plane sequence tracking in the HSV space

One of the problems encountered with this sequence was that the bounding box containing the whole plane also contains a large part of sky, which is constantly changing. Then, for some images, the box with the ROI tends to get slightly lost, but still with most of it covering the plane, due to the large amount of changing sky. One solution is to restrict the tracked area to the central part of the plane where no sky is present. This yields accurate results since there is no "background" noise. To obtain this central part, the threshold value was modified, when subtracting the first image from the "background" (no region growing segmentation was used for this slightly different approach). For displaying the results, only a bounding box around the tracked part can be shown, or a bounding box extended to the whole plane can be used, being the last one more visually attractive. To emphasize this approach, only this region is shown in Figure 7.

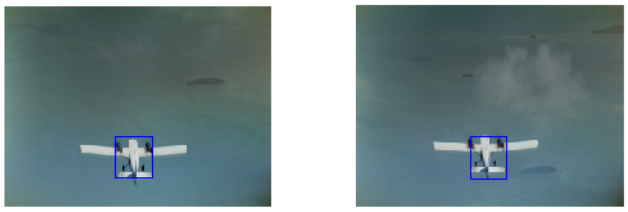


Fig. 7: Plane sequence tracking using a reduced ROI

There is a compromise between speed and color space. If RGB is used, the algorithm will work faster, but for some frames the tracking region will be slightly moved (very small displacement. When using HSV the result is better, but it is slightly slower as compared to RGB, because of the color space modification.

Another approach for dealing with color is to assume that the shape of a non-rigid object is approximated by an ellipsoidal (rectangle or circular or any other kind of shape) region in an image. The region can then be selected exactly in the same way that the very initial steps fo the initial condition were selected. Let x_i denote a pixel location, θ the initial location of the center of the object in the image, V the variance, M the number of bins in the histogram and let $b(x_i)$ be a function that assigns the color value for each pixel to its bin. The value of the m^{th} bin is calculated by

$$o_m = \sum_{i=1}^{N_{v_0}} \mathcal{N}(x_i; \theta_0, V_0)\delta[b(x_i) - m] \tag{11}$$

In this equation $\mathcal{N}$ represents the Gaussian distribution and δ is the Kronecker delta function. The weighting coefficient calculation is exactly the same as the one used in the first part: the farther the lesser weight, and the nearer the larger weight.

VI. Tracking in presence of total occlusion: profiting the Kalman Filter

In order to increase the accuracy and robustness of the tracking, and to deal with total occlusion, the Kalman filter was introduced in the body of the algorithm. The idea is to predict the position of the tracked object in the new frame based on the object's previous motion.

A. Methodology

Let the states of the filter be $X = [x_c, y_c, \dot{x_c}, \dot{y_c}]$, where $\dot{x_c}$ and $\dot{y_c}$ represent the velocity in x_c and y_c, respectively. The discrete-time process model will be given by

$$X_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{t-1} + N(0, Q) \tag{12}$$

where Δt is the sampling time and $N(0, Q)$ represents a normal distribution for the model with a covariance given by Q. For initialization, the initial positions are set to the initial centers of the ROI (computed in the very first frame) and the velocities are arbitrarily initialized (e.g. to 1). The discrete-time measurement model is given by:

$$Y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X_t + N(0, R) \tag{13}$$

where $N(0, R)$ is the zero-centred normal distribution corresponding to the uncertainty in the measurement, with R as the covariance matrix for the measurement.

After the initialization of the values, the basic iteration of the mean-shift algorithm is performed for the second frame, but before assigning the new center of the ROI (the tracked object), the prediction part of the Kalman filter is computed with the very well known Kalman equations. After the prediction step, the value of the distance ρ is checked. If $\rho > \rho_{min}$, the measurement (estimation of the new center) is consistent enough and it is used to update the prediction. The value of ρ_{min} is experimentally determined and usually ranges from 0.8 to 0.95, but other values are possible. If $\rho < \rho_{min}$, then the new center calculated using mean-shift is not reliable, and no update is performed. This happens, for instance, when there is complete occlusion. In this situation, the new center relies completely on the prediction of the Kalman Filter, since no measurement is taken into account.

B. Example

This methodology was applied to the sequence of the toy car, presented in the previous section. In that case, the mean shift tracking algorithm failed when the toy was completely occluded by the box, as shown in Figure 5. The total occlusion problem was solved using the Kalman Filter. When initializing the algorithm, the value of Δt was assumed to be 2 for a stronger prediction, the values for Q were zeroed everywhere except in the velocities variance where they were established as 0.1^2, and R was set to $diag\{1, 1\}$. For checking the reliability of mean-shift, a value of $\rho_{min} = 0.85$ was used. The results are shown in Figure 8, where the car is moving from left to right passing behind the white box. It can be observed that the target is tracked even after having been completely occluded by the box, thanks to the addition of the Kalman filter prediction.

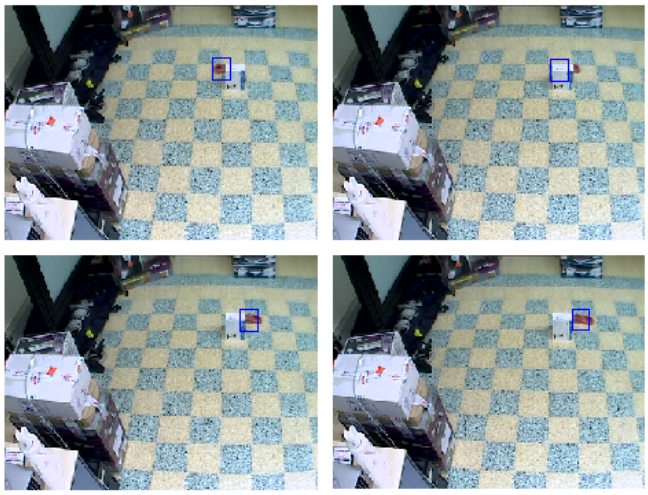


Fig. 8: Kalman Filter with mean-shift algorithm to overcome the total occlusion problem.

VII. Limitations of the algorithm

The experiments showed that this algorithm is quite sensitive to the initial conditions, that is, if the bounding box is initially not located in a good place (or, equivalently, if the initial target is not precisely determined), it will go to a wrong place after some movement of the object. However, this displacement will not be very big and tracking will be still possible, although less accurate. Second, in case that the difference between the background and the object is small in the sense of illumination and contrast, it is hard to track the object with the color model that was defined. One of the possible ways is to try to adapt the color space according to the case of study, if high accuracy is needed. But in case there is a good difference between the object and its background, the algorithm works well due to a lot of redundant information. The main limitation of simple mean-shift tracking is the case of total occlusion. The proposed way to solve this problem in this paper was to use the kalman filter to predict the position of the tracked object when it is totally occluded.

VIII. Conclusions

The mean-shift tracking algorithm works well when there is a lot of information in the frames, a notorious difference in contrast, and illumination without presence of noise. This algorithm might present some problems when facing severe or complete occlusions, or when dealing with varying size objects. The way to solve this problem, proposed in this paper, is the addition of the Kalman filter, which has a good performance in real applications even with presence of noise. A combination of mean-shift tracking with Kalman filter yields to very good results when dealing with total occlusion due to the prediction of the target's motion based on the previous frames measurements. The usage of particle filter is also possible.

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[2] G. Welch and G. Bishop, "Scaat: Incremental tracking with incomplete information," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 333–344.

[3] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.

[4] K. Choo and D. Fleet, "People tracking using hybrid monte carlo filtering," in *Proceedongs of the Eighth IEEE International Conference on Computer Vision*, vol. 2. IEEE, 2001, pp. 321–328.

[5] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.

[6] I. Cox and S. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 2, pp. 138–150, 1996.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 564–575, 2003.

[8] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1296–1311, 2003.

[9] B. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*, vol. 3. Citeseer, 1981, pp. 674–679.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, 2004.

[11] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.

[12] H. Zhou, Y. Yuan, Y. Zhang, and C. Shi, "Non-rigid object tracking in complex scenes," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 98–102, 2009.

[13] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1530–1537.

[14] B. Pu, S. Liang, Y. Xie, Z. Yi, and P. Heng, "Video facial feature tracking with enhanced asm and predicted meanshift," in *2010 Second International Conference on Computer Modeling and Simulation*. IEEE, 2010, pp. 151–155.

[15] D. Comaniciu and V. Ramesh, "Real-time tracking of non-rigid objects using mean shift," Jul. 8 2003, uS Patent 6,590,999.

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[17] Z. Zivkovic and B. Krose, "An EM-like algorithm for color-histogram-based object tracking," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004.