

TRACKING PARTIALLY OCCLUDED OBJECTS WITH CENTRIPETAL ACTIVE CONTOUR

PRAVINKUMAR G. KANDHARE, ABDULLAH N. ARSLAN
AND NIKOLAY METODIEV SIRAKOV

Abstract. This study develops a new approach for tracking single object through a video in case of partial occlusion. The approach uses an active contour (AC) to determine the mass center of the target in the present frame. The mass center is utilized by a Modified Kalman Filter which estimates its position, in the next frame, as the center of the area, the target is most likely to appear in. This area is bounded by the AC, which evolves toward the target. If multiple objects are enveloped, a shell algorithm splits the AC to multiple contours utilizing the mass center position estimated in the previous frame. Every shell is constructed by a part of the AC which envelops the visible, from the contour, parts of objects and ends at the mass center of the AC. In every shell, an AC extracts the object's boundary and generates a finite numerical sequence as a signature for this boundary. A sequence alignment approach matches the signature of every object, in the present frame, with the signature of the target. To validate the method, experiments are performed on 14 videos. The paper ends with discussion on the results and the contributions comparing them with contemporary works.

1. INTRODUCTION

Object tracking in video sequence has become an increasingly important ongoing research topic, with applications including surveillance, security, computer vision, animation, sports, and autonomous navigation. It holds because the present society accumulates millions of video records, which potentially need analyses [1, 7].

A number of tracking algorithms are available in the literature [1, 6, 8, 11, 13, 14, 20, 24]. Generally, tracking systems face two basic obstacles: motion detection, and matching. Motion problem consists of: predicting the position of a target object in the next frame, (locating the search area in which the object can be found). Matching problem consists of: identifying the target object in the next frame within the search area. In some real time applications, detecting the search region in the next consecutive frame is a challenging task due to a random jump of the target (cusps in the motion function). Also, the accuracy of tracking can be affected due to loss of information caused by projection of a 3D object onto 2D images, scene illumination changes, shadows, complex object motion, non-grid or articulated nature of objects, partial and full object occlusions, complex shapes and camouflage in color [1].

MSC (2010): primary 68U05, 68U10; secondary 68W32, 68W99.

Keywords: shrinking active contour, Kalman Filter, tracking, cyclic sequence alignment, matching.

As stated in [9], the tracking methods can be separated to four classes: a) Active contour (AC); b) Region; c) Feature; d) Model based tracking. In the AC tracking, the target description is simpler with respect to the region tracking. However, the latter approach is sensitive to initialization [14]. A challenge for every tracking approach is to detect the target after occlusion. A number of occlusion-handling methods have been proposed in the literature. The Structural Kalman Filter [11] utilizes the relational information among the sub-regions of a moving object to handle occlusion. The relational information and actual measurements are used to obtain a priori estimate of the next state of the sub-regions.

Yang et al. [24] introduced a tracking model to handle complete and long duration occlusion without considering prior knowledge about the target's shape and motion. In [8], Han et al. have proposed method for tracking multiple objects under partial occlusion. In their work the object is represented with multiple features. The paper applies the assumption that the object texture will be different in the case of an occlusion, compared to the original object. In each frame, the new features selected from the current frame are updated automatically by an updating mechanism. To avoid faulty decision, weighting functions eliminate the tracking error produced by the scene occlusions.

To deal with object tracking in such an environment, we suggest using a Modified Kalman Filter integrated with a shrinking active contour model (S-ACES) [20, 21], shell algorithm for objects separation [21], and sequence alignment approach to detect shape similarity [2, 3]. The alignment approach utilizes finite numerical sequences, generated on the basis of the S-ACES extracted boundaries.

This paper is organized as follows: Section 2 formulates the problem; Section 3 describes S-ACES and boundaries extraction. Section 4 presents the Modified Kalman Filter. In Section 5, we describe the contour splitting approach that separates objects and Section 6 discusses a target recognition algorithm based on sequence alignment. Section 7 shows experimental results, while 8 concludes the paper outlining the contributions, the advantages and the bottlenecks.

2. PROBLEM DEFINITION

As mentioned in the introduction, S-ACES [3, 20, 21] will be used to measure the present object's position. For this purpose we adapt the existing Kalman Filter [11, 13] to make it work with S-ACES [20].

In object tracking, we extrapolate the future position of the target object through a set of previous frames. This is a challenging task if the target is occluded (completely or partially) [6], or the motion function is not smooth.

In the present study we assume that the motion is a smooth function: the rate of change of the motion is continuous, along with the rate of change in the target's shape. A single target with partial occlusions is the subject of tracking. No prior information is necessary. The user clicks on the target at its first appearance.

In the case of occlusions, multiple objects fall in the capture range of S-ACES. To separate the objects we split the AC applying a shell algorithm introduced in [20, 21]. A sequence alignment approach [18, 22, 23] determines the target object subject of tracking among others.

In the present work, the boundaries extracted by S-ACES are represented as finite numerical sequence of boundary points. Each of these sequences is matched [2, 3] to the finite numerical sequence generated from the boundary of the target. The closest match defines the target and it's tracking continues from the present position. The mass center of the target is called measured position and is applied to calculate the estimated position of the target in the next frame [13].

3. S-ACES ACTIVE CONTOUR MODEL

An important step for tracking is to locate the target in the current frame, and segment the target from the background of the image. Different kinds of models are available in the literature for boundary extraction [15, 17]. We select the parametric shrinking active contour (S-ACES) developed in [21], because of its large capture range, accuracy, and efficient numerical implementation [20]. In our tracking method, S-ACES is used as a measurement tool, which is formulated with an evolution equation (EE), initializing conditions (IC) and a boundary condition (BC). The EE is described with

$$r(s, t) = e^{(s \frac{|ds|}{2} - |ds|^2 t)} \left[C_1 \cos \left(c \frac{|ds|}{2} s \sqrt{3} \right) i + C_2 \sin \left(c \frac{|ds|}{2} s \sqrt{3} \right) j \right]. \quad (3.1)$$

In Equation (3.1), s denotes arc length parameter, t is the time parameter, C_1 , C_2 and c are real valued coefficients. The IC, formulated below, places the initial contour in the present frame at the position estimated in the previous frame

$$\begin{aligned} r(s, t)_{t=0.001, \frac{|ds|}{2}=\alpha, C_1=C_2=R, c=1000} \\ = R e^{s\alpha - 4\alpha^2 \cdot 0.001} [\cos(1000\sqrt{3}\alpha s) i + \sin(1000s\sqrt{3}\alpha s) j]. \end{aligned} \quad (3.2)$$

In the above equation, $R = \frac{1}{2}(n_r^2 + n_c^2)^{\frac{1}{2}}$, while n_r and n_c represent, respectively, the number of rows and number of columns of the current video frame, $a = \frac{|ds|}{2}$, $|ds|$ is the arc's s length approximation. There is a parameter, in Equation (3.2), set to 1000 in order to keep the evolving curve closed. The BC used to halt the active contour on the target boundary is

$$r(s^*, t^*) = r(s^*, t^* + \partial t) \quad \text{if} \quad \frac{\partial f(s^*, t)}{\partial t} > \varepsilon \quad \text{for} \quad s = s^* \quad \text{and} \quad t^* > 0.001. \quad (3.3)$$

In Equation (3.3) $f(x, y)$ denotes the image (frame) function, whereas ε is a threshold chosen by the user with respect to the nature of the target. The boundary points extracted by S-ACES are used to calculate the mass center of the object in the present frame. This mass center is the measured position of the target, and is denoted hereafter by X_t^{meas} for frame F_t .

Assume multiple objects are enveloped by the initial contour defined with Equation 3.2 in the present frame. Since S-ACES is a centripetal model the active contour evolved by Equation 3.1 produces a shell, which envelops an object or group of objects as shown in Figure 1 (b). A shell of an object is created by the centripetal point and the object's boundary points, visible from the initial AC. Further, the mass center of every shell is calculated and S-ACES runs in every shell using its mass center. The process of shell generation halts if a single object

remains in every shell, which occurs when the mass center of a shell belongs to a single object (Figure 1 (b)). The method for shell definition is described in more detail in [21].

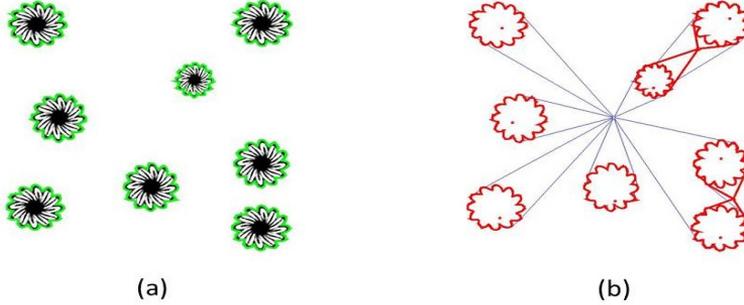


Figure 1. (a) Flowers and their boundary extracted by S-ACES; (b) The shells used for extraction.

4. MODIFIED KALMAN FILTER

The main concept of the traditional Kalman Filter (KF) is that the underlying system is a linear dynamic system and all measurements and error terms have a Gaussian distribution [25]. The KF is an optimal recursive data processing algorithm, which uses measurement data, prior knowledge about the objects and measurement device to produce an estimation of the desired variables and dynamically minimize the error.

In our tracking approach, we modified the KF to optimize its fusion with S-ACES. Thus, three previous frames are taken into account to estimate the mass center position of the target in the next frame. The algorithm functionality is divided into the following stages: 1. S-ACES; 2. Prediction; 3. Estimation; 4. Splitting-Recognition.

In S-ACES stage, the algorithm extracts the boundary of the object (target) subject of tracking. This extracted boundary helps in determining the mass center of the target in the current frame. In prediction stage, the algorithm calculates the future location of the target in the next consecutive frame. The error in the predicted position is minimized during the estimation stage. In each frame, the mass center of the target is measured and the two remaining terms are calculated. The predicted position X_{t+1}^{Cpred} for frame F_{t+1} is calculated in F_t using information from F_t , F_{t-1} and F_{t-2} .

4.1. S-ACES stage

As stated in Section 3 the S-ACES is applied on each individual frame to extract the boundary points of the target object subject of tracking. These boundary points are used to calculate the mass center of the target, which represents the measured position in the present frame F_t , and is denoted by X_t^{Cmeas} . In the

first triplet of frames where the target appears, S-ACES is applied using radius R and center C , selected by the user. In F_t , for $t = 3, 4, \dots$, $C = X_t^{Cest}$ and $R = \max_{X_{i(t-1)} \in A_{t-1}} \{d(X_{t-1}^{Cest}, X_{i(t-1)})\}$, where A_{t-1} is the vector column of the target's boundary coordinates determined by S-ACES in F_{t-1} , and $X_{i(t-1)}$ is the i^{th} entry of A_{t-1} .

4.2. Prediction stage

This stage predicts the position of the mass center in F_{t+1} using the information from frames F_t, F_{t-1} , and F_{t-2} [11, 13]. In Equation (4.1), V_t and a_t denote the velocity and acceleration of the target. T is the time between two frames. Velocity and acceleration are calculated using the following formulas

$$V_t = \frac{X_t^{Cmeas} - X_{t-1}^{Cmeas}}{T} \quad \text{and} \quad a_t = \frac{V_t - V_{t-1}}{T}, \quad (4.1)$$

where X_t^{Cmeas} and X_{t-1}^{Cmeas} are the measured positions in F_t and F_{t-1} , respectively. a_t is the acceleration for the time T that elapsed between F_{t-1} and F_t ,

$$X_{t+1}^{Cpred} = X_t^{Cest} + V_t T + \frac{1}{2} a_t T^2 + E_{X_t}. \quad (4.2)$$

Error in prediction is calculated using the equation, $E_{X_t} = X_t^{Cmeas} - X_t^{Cpred}$. In Equation (4.2), X_{t+1}^{Cpred} represents future location of the target in the $(t+1)$ th frame, known as "predicted position". X_t^{Cest} is the estimated position at F_t , but calculations are performed at F_{t-1} , as shown in Section 4.3. Equations (4.1) and (4.2) suggest that in order to start the recursive Equation (4.2) initial data is necessary from frames F_1, F_2 , and F_3 . Thus, the user manually picks a point at the target object in frame F_1 and apply S-ACES to define the target boundary A_1 and it's mass center X_1^{Cmeas} . The boundary A_1 is used as a template for matching, if necessary to detect the target in $F_t, t = 4, 5, \dots$. Further, we consider $X_t^{Cmeas} = X_t^{Cpred} = X_t^{Cest}$ and $E_{X_t} = E_{Z_t} = 0$ for $t = 1, 2, 3$ to generate the data necessary to calculate the acceleration a_t to be used in X_{t+1}^{Cpred} .

4.3. Estimation stage

The predicted position may contain an error in the result. To minimize this error, estimated position X_{t+1}^{Cest} is introduced and calculated as follows

$$X_{t+1}^{Cest} = X_{t+1}^{Cpred} + E_{Z_t}. \quad (4.3)$$

In Equation (4.3), X_{t+1}^{Cest} is the estimated position for F_{t+1} . The calculation is performed in F_t using the Measurement Error (E_{Z_t}) determined by

$$E_{Z_t} = X_t^{Cmeas} - X_t^{Cest}. \quad (4.4)$$

In the original Kalman Filter [11, 13], E_{Z_t} is calculated through the covariance matrix, whereas our measurement error calculation is simpler and less time consuming as shown in Equation in (4.4). But its accuracy is lower compared to the original one. The large S-ACES capture range helps our method compensate the lower accuracy [21].

5. CONTOUR SPLITTING

There are possible scenes in a video frame F_{t+1} where the X_t^{Cest} may be mapped out of the target object O_T . Two cases are possible: a single object is enveloped by the S-ACES contour (Figure 2 a)); and multiple objects are enveloped (Figure 2 b)). In either case every single object, visible by S-ACES, is enveloped by a Shell

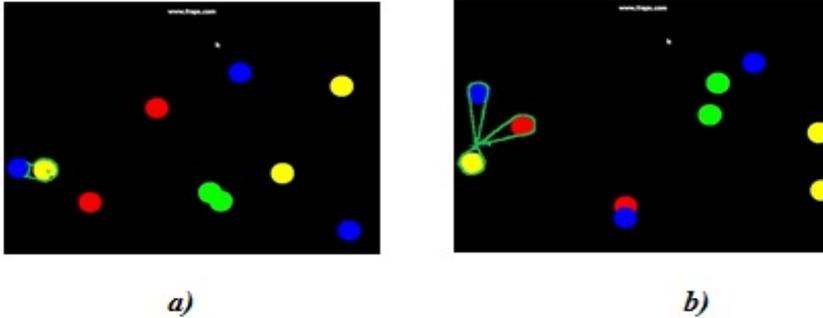


Figure 2. a) The F_3 of a video of multiple moving balls. Only the target is enveloped by S-ACES but X_3^{Cest} lies out of O_T ; b) in F_{16} three objects are enveloped by S-ACES [12].

[21] constructed from the object's boundary, visible by the initial contour S-ACES, and two straight lines. All the shells meet at the X_{t+1}^{Cest} calculated at F_t (Figure 2 b)). This point is the center C for S-ACES [21] at F_{t+1} . Recall that A_t denotes the vector column formed by the coordinates of the target boundary determined by S-ACES in F_t .

Statement 5.1. *Since S-ACES is a centripetal model [21], a criterion that tells X_{t+1}^{Cest} is located out of O_T (Figure 2) is*

$$X_{t+1}^{Cest} \in A_{t+1}. \quad (5.1)$$

Statement 5.2. *If Equation (5.1) is satisfied, the number of appearances of X_{t+1}^{Cest} in A_{t+1} equals the number of objects enveloped by S-ACES (Figure 2 b)). Every two appearances of X_{t+1}^{Cest} in the vector A_{t+1} enclose the set of boundary points $S_{(t+1)i}$ that belong to a single object.*

Statement 5.1 determines the frames to be processed by the splitting and matching algorithms. In such frames, Statement 5.2 defines and separates the sequences of boundary points $S_{(t+1)i}$, which belong to a single object in the present frame. Then, every sequence $S_{(t+1)i}$ for $i = 1, 2, \dots, k$, where i denotes the consecutive number of the set, k is the number of the enveloped sets (objects), is compared with A_1 applying a *cyclic sequence alignment* based target recognition method. We consider the set $S_{(t+1)i}$ with the highest score of matching with A_1 belonging to the target. Then we find the mass center and the maximal distance of the $S_{(t+1)i}$ points to the mass center and use both of them to employ S-ACES and find the boundary of the target. Then, we calculate X_{t+1}^{Cpred} , and X_{t+1}^{Cest} employing Equations 4.2 and 4.3. The above method was applied to track the yellow ball (the

lightest one in gray scale paper) in the video clip from [16]. Statements 5.1 and 5.2 are employed along with the Shell algorithm introduced in [21] to separate the two enveloped objects in Figure 3 a). The separated objects are shown in Figure 3 b), where each of them is bounded by S-ACES. The localized target is shown in c).

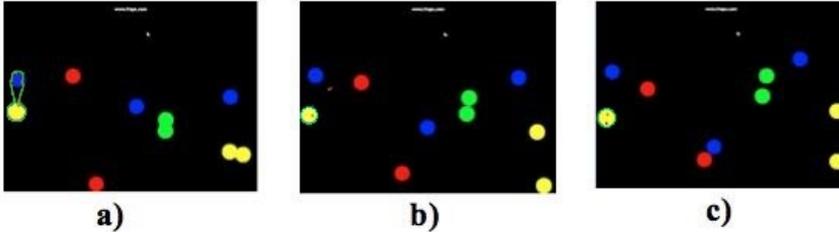


Figure 3. a) F_{11} - two objects enveloped by S-ACES; b) F_{14} - every object is enveloped by a contour; c) the target is detected using the sequence alignment approach [23].

6. TARGET RECOGNITION

Cyclic sequence alignment is used for target recognition. Sequence alignment definition has its roots in string edit distance [22], and it has been successfully applied to shape matching applications [2, 3]. Let $A = a_1 a_2 \dots a_n$ and $B = b_1 b_2 \dots b_m$ be two sequences defined over an alphabet Σ . Special symbol $'-'$ (not in Σ) can be inserted in A and B in any selected positions to yield sequences $A' = a'_1 a'_2 \dots a'_{n'}$ and $B' = b'_1 b'_2 \dots b'_{m'}$ of the same length, and with the condition that there does not exist any position i where $a'_i = b'_i = '-'$. The result is a $2 \times n'$ matrix (called an alignment matrix) which has A' as row 1, and B' as row 2 as shown in Equation (6.1). This matrix cannot include a column containing only $'-'$ s. It can be seen that an alignment matrix for A and B is not unique. Let δ be a function that assigns a score to every possible column of two symbols in $\Sigma \cup \{'-\'}$. Sequence alignment is the problem of finding an alignment with the maximum score over the set M_{AB} of all possible alignment matrices

$$S(A, B) = \max_{\left[\begin{array}{cccc} a'_1 & a'_2 & \dots & a'_{n'} \\ b'_1 & b'_2 & \dots & b'_{m'} \end{array} \right] \in M_{AB}} \left(\sum_{i=1}^{n'} \delta \left(\left[\begin{array}{c} a'_i \\ b'_i \end{array} \right] \right) \right). \quad (6.1)$$

This problem can be solved in time $O(nm)$ using the following dynamic programming formulation [23]: for all i, j , $0 \leq i \leq n$, $0 \leq j \leq m$,

$$D[i, -1] = D[-1, j] = -\infty; D[0, 0] = 0; \text{ and}$$

$$D[i, j] = \max \left\{ \begin{array}{l} D[i-1, j] + \delta \left(\left[\begin{array}{c} - \\ b_j \end{array} \right] \right), D[i, j-1] + \delta \left(\left[\begin{array}{c} a_i \\ - \end{array} \right] \right), \\ D[i-1, j-1] + \delta \left(\left[\begin{array}{c} a_i \\ b_j \end{array} \right] \right) \end{array} \right\}, \quad (6.2)$$

$$S(A, B) = D[n, m].$$

Let B^k denote the cyclic shift of $B = b_1 b_2 \dots b_m$ for $0 \leq k \leq m - 1$. That is, $B^k = b_{1+k} b_{2+k} \dots b_m \dots b_{m+k}$, where $i+k$ means $i+k-m$ if $i+k > m$. The cyclic sequence alignment score of A and B is the maximum score obtained by aligning A to every possible cyclic shift of B [18]. It can be defined by the following equation

$$C(A, B) = \max_{0 \leq k \leq m-1} S(A, B^k). \quad (6.3)$$

We use cyclic sequence alignment to identify the object we track (O_T) among possible candidate objects (B). We do this by computing a cyclic alignment score for O_T and each candidate B , and by choosing B that has the highest alignment score with O_T . For cyclic sequence alignment we align O_T to every cyclic shift of B as shown in Equation (6.3). Each ordinary sequence alignment is done by using the dynamic programming formulation in Equation (6.2). In this case, each symbol in O_T and B is a (x, y) coordinate on 2D plane. The δ function is defined by

$$\delta \left(\begin{bmatrix} - \\ (x_j^b, y_j^b) \end{bmatrix} \right) = \delta \left(\begin{bmatrix} (x_i^a, y_i^a) \\ - \end{bmatrix} \right) = 0,$$

$$\delta \left(\begin{bmatrix} (x_i^a, y_i^a) \\ (x_j^b, y_j^b) \end{bmatrix} \right) = \begin{cases} 1, & \text{if the Euclidian distance between} \\ & (x_i^a + dx, y_i^a + dy) \text{ and } (x_j^b, y_j^b) \text{ is at most } \tau; \\ 0, & \text{otherwise,} \end{cases}$$

where dx and dy are given displacements along x and y axes, and τ is the tolerance distance. In other words, assuming that the target object moves to a new position which is at distance dx on the x axis and dy on the y axis from the old positions, we cyclically compare coordinates to count the number of pairs within distance τ . The cyclic comparison considers that the target object may have been rotated during this movement. We implement a straightforward cyclic sequence alignment algorithm which performs $O(m)$ ordinary sequence alignments (one for each cyclic shift of the second sequence). The total time for cyclic sequence alignment is $O(nm^2)$. Since the sequences are short (m and n are in the order of hundreds), the actual run time for comparison is in the range of 10 – 30 milliseconds using a desktop with 2.3 GHz Intel core i5. This properly makes the present algorithm suitable for online matching. The algorithm was applied on F_{14} shown in Figure 3 b). The target object (the yellow one on the left) was successfully determined and shown in Figure 3 c). The present experiment shows that our matching algorithm is capable of distinguishing objects with visually similar shapes. The reasoning holds because the matching algorithm uses finite numerical sequences generated by the S-ACES Shell algorithm. This algorithm employs different boundary approximation in every shell with respect to its size. The different approximations provide different scores in the matching. Sequence alignment works well because it returns matching score when the border of the tracked object is only a part of the merged boundary (in the case of presence of other nearby objects and occlusion). Cyclic sequence alignment enables rotation-invariant comparison. This holds because the tolerance distance in coordinates-matching helps with cyclic matching by a bounded rotation amount. To obtain the invariance, the sequence describing

the boundary should be generated as a sequence of angles where an angle is formed at each boundary point using two reference points (e.g. its cyclic predecessor and successor on the boundary). Examples to this can be found in [2, 3].

7. EXPERIMENTAL RESULTS

Experiments with 14 video clips were performed to validate the Modified Kalman Filter with centripetal S-ACES and matching. We used Netbeans 7.0.1 on standard Mac OS X (2.3 GHz Intel Core i5). Simulation model was implemented using Java language to evaluate tracking performance [21]. We present tracking results in Figures 3–7 from five different video clips. Figure 3 shows an example of tracking a yellow colored ball which is moving along with other balls. The splitting and matching algorithms successfully determined in F_{14} the target which was lost in F_{11} . The tracking software tool processes all frames at once, rather than with manual execution of each individual frame, using batch processing technique. We evaluated performance of our tracking algorithm using five video clips that have different frame sizes and number of frames. We considered two kinds of image sequences to perform the experiments as shown in Figure 4. In the first row (Figure 4, Aircraft image sequence), the target aircraft is in a very close range with another one. Second row (Figure 4, Soccer image sequence) is used to analyze performance of tracking in overlapping and close interaction environment. In Figure 4 upper row, the target aircraft is tracked successfully from F_1 to F_{24} . From F_{49} to F_{74} , three objects are enveloped by S-ACES and tracked in a group/cluster without splitting and recognition. Similarly in Soccer image sequence, a soccer player is tracked from F_1 to F_{23} . The AC enveloped other soccer players along with the target object from F_{39} to F_{46} . As our goal is to track a single target object, we need to separate the target from other objects and recognize it for further tracking.



Figure 4. Tracking results obtained using Modified Kalman Filter without Shell and Matching sequence [10, 16].

To overcome the above problem we show in Figure 5 tracking results obtained using Modified Kalman Filter with Shell and Sequence Matching. In the upper row, S-ACES envelops target aircraft and the part of other object. In F_5 , Shell algorithm separates these objects. The Shell approach provided two objects but it is still not known which is the target. Here, we applied matching sequence algorithm to recognize the target object. A template of the target boundary is (extracted in F_1 by S-ACES) compared with all objects separated by the Shell algorithm. The object with which the maximum match score (via *cyclic sequence alignment*) is obtained identifies the target. The target was detected in F_6 . Similar situation arises in F_{13} and it is resolved in F_{15} using Modified Kalman Filter with Shell and match score (Figure 5 bottom row).

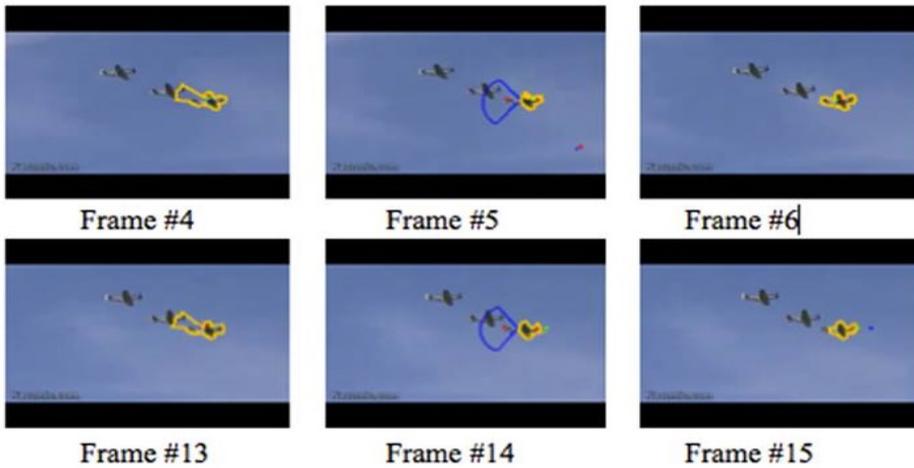


Figure 5. Tracking with the Modified Kalman Filter with Shell and Matching sequence [16].

Another example of tracking with splitting and recognition is given in Figure 6, where a neutrophil is chasing an intruder. The latter is the target which was lost in a frame and found in the second consecutive frame.

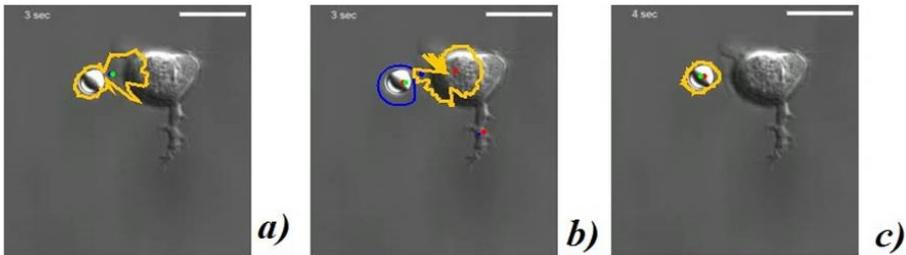


Figure 6. a) Neutrophil and the intruder in close interaction. They are enveloped together by S-ACES; b) Both objects are split by shells algorithm; c) The subject of tracking is identified.

Further, experiment with occlusion was performed for tracking a car in a video clip with 150 frames. Every frame is of size of 480×360 pixels. The new tracking algorithm, using modified Kalman, centripetal AC with shells and matching, successfully tracked the car through the video. When the car emerged behind the tree, the matching algorithm compared its boundary and the boundary of the tree, extracted by S-ACES, with the template of the car. The latter was successfully recognized as seen in Figure 7.

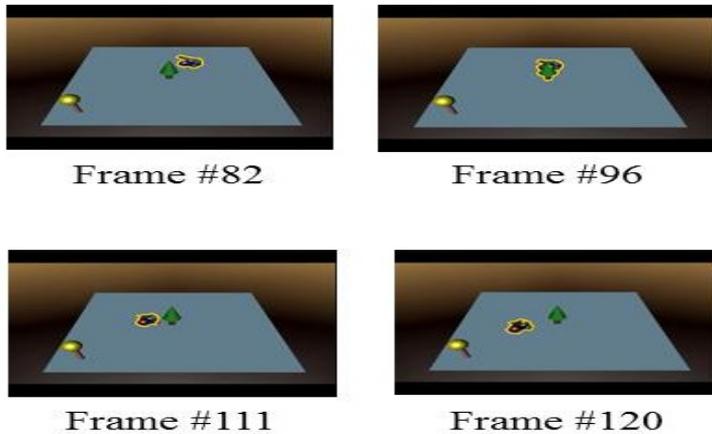


Figure 7. Tracking object with partial occlusion in a video [5] using the centripetal AC with Shell and Matching. Frame #82 A car under tracking by the AC; Frame #96 the car and occluding tree bounded by the AC; Frame #111 The car was separated from the tree, and recognized as a subject of tracking; Frame #120 Tracking after the occlusion.

8. CONCLUSION

The present paper develops a new method capable of tracking a single target, which could be partially occluded by other objects in the scene. The method employs multiple approaches including Modified Kalman Filter for tracking without occlusions; shrinking active contour S-ACES to define boundaries, and to calculate the mass center of the target; shell approach to split the active contour if multiple objects are enveloped and extract the boundaries of these objects; cyclic sequence alignment to detect the target among the separated objects. A boundary template generated by S-ACES in the first frame is used for the matching.

An advantage of the present method compared to the Kalman [11] and those present in [8, 24] is the higher speed. Another advantage is that the geometric information extracted from the objects in a frame is useful to determine the semantic of the scene. Also, the present method is capable of distinguishing objects with visually similar shapes while the methods from [8, 24] will miss the target in such a situation. However, a disadvantage of the present approach is that it cannot handle right now a tracking of multiple targets as [8, 24] can do. Also, if multiple objects are enveloped, S-ACES will produce the same shells with the

same approximation for all objects if they have similar shapes and X_t^{Cest} coincides with the mass center of the set of objects.

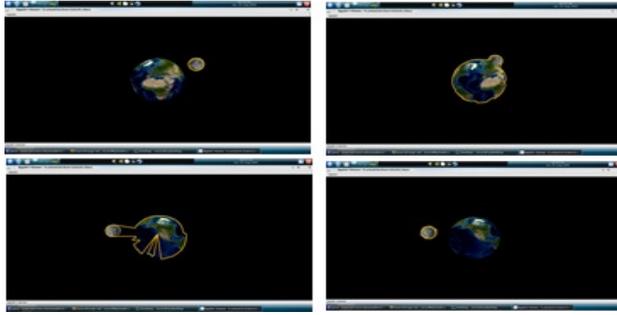


Figure 8. From upper left to down right $F_{225}, F_{260}, F_{366}, F_{368}$.

A target may be missed if one of the objects enveloped by S-ACES appears to have a weak boundary when the objects are close or overlap each other. In this case the AC enters into the object with the weak boundary (Figure 8, F_{366}). As a consequence the mass center of the AC will be incorrect, which may lead the method to miss the target. To avoid cases when the objects boundaries change over time an adjustment of the AC contour parameters or the mass center calculation is necessary. Such an interaction was done in F_{367} in order to recover the target in F_{368} .

Another bottleneck of the new method is its inability to track targets, which changes shape from frame to frame. The methods in [6, 8, 11, 13, 14, 24] are not able to track such objects as well, but an enlarging active contour presented in [20] can track objects with changing shape.

Reference [4] reported a very fast tracking method with a high “percentage of correct classification”. This method employs low level pixel information, which bounds the method from use in high level problems solution like: target and/or activity recognition. In contrast the newly introduced Modified Kalman Filter, using S-ACES and matching extracts and uses geometric and shape information from the target. This makes the two approaches fundamentally different. Also, the new method is suitable for target/activity recognition and handles slowly moving objects, whereas the background subtracting methods similar to the one in [4] experience difficulties when tracking such objects.

The run times for tracking in case of splitting and recognition are higher compare to the tracking without recognition (Table 1). But splitting and recognition algorithms are applied only to frames where partial occlusion or very close interaction between the target and another object is detected applying Statement 5.1.

The calculation complexity of S-ACES and shell algorithm [21] is $O(n_r n_c)$, while the matching approach has a calculation complexity in the order of $O(nm^2)$. In the above estimations n_r and n_c are the sizes of a single video frame, n and m are the sizes of the objects signatures (boundary sequences) to be matched. Follows that the complexity of the entire method is $O(n_r n_c + nm^2)$.

Table 1 shows the run time of the new tracking software measured on five video clips with target occlusion and video clips where the target was never occluded.

Name of video file	Image size	# Frames	Run time per frame	in second
			No Matching	Matching
Java_Bouncing_ball	480x360	299	0.13846	0.464
Aircraft	320x240	219	0.07346	0.920
Soccer	480x360	136	0.10070	0.559
Neutrophil	218x240	50	0.090	0.615
Vehicle	480x360	77	0.2181	0.399

Table 1. Experimental data derived using S-ACES tool.

We have calculated the accuracy of our method for tracking objects with partial occlusions. The accuracy of boundary extraction by S-ACES was calculated in [19] to be 94%. In the present paper the accuracy of tracking is determined using video clips where the tracked object is never occluded and video clips where it is occluded. In the case of no occlusion the accuracy was 97.6%, while in case of occlusion was 98.7%. The accuracy is calculated dividing the number of frames in which the tracked object is missed by the number of frames the tracked object is successfully detected.

One may tell the accuracy of tracking in case of occlusion is higher than if the traced object has never been occluded. This fact is due to the recognition algorithm called to recover the target, which improves the accuracy.

Our work continues with elaboration of the method to make it capable of tracking multiple objects, to decrease the complexity of matching, and hold full occlusions.

REFERENCES

- [1] M. A. Ali, S. Indupalli and B. Boufama, *Tracking multiple people for video surveillance*, in: Proc. of CRV06 Workshop on Video Processing for Security (VP4S-06), June 2006.
- [2] A. N. Arslan, C. F. Hempelmann, C. Di Ferrante, S. Attardo and N. M. Sirakov, *From shape to threat: Exploiting the convergence between visual and conceptual organization for weapon identification and threat assessment*, Recipient of the Lockheed-Martin Best Paper Award, in: F. A. Sadjadi and A. Mahalanobis (eds.), *Automatic Target Recognition XXIII*, Proceedings of SPIE **8744**, 2013, 15 pp.
- [3] A. N. Arslan, N. M. Sirakov and S. Attardo, *Weapon ontology annotation using boundary describing sequences*, IEEE SSI AI, Santa Fe, New Mexico, April 22-24, 2012, 101–104.
- [4] O. Barnich and M. V. Droogenbroeck, *ViBe: A universal background subtraction algorithm for video sequences*, IEEE T. Image Process. **20** (2011), 1709–1724.
- [5] <http://www.youtube.com/watch?v=X1Ax77cGA9o>, user: Bodnaar, Obstacle avoidance of a mobile robot (simulation), additional info: Obstacle avoidance of a mobile robot, Mechatronics, 2009, University of Szeged.
- [6] B. Bose, X. Wang and E. Grimson, *Multi-class object tracking algorithm that handles fragmentation and grouping*, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, 1–8.
- [7] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, E. Enomoto, O. Hasegawa, P. Burt and L. Wixson, *A system for video surveillance and monitoring*, VSAM Final Report, Carnegie Mellon University, 2000.

- [8] B. Han, C. Paulson, T. Lu, D. Wu and J. Li, *Tracking of multiple objects under partial occlusion*, in: Proceedings of SPIE Defense & Security Symposium 2009, Orlando, FL, USA, April 13–17, 2009.
- [9] W. Hu, T. Tan, S. Maybank and L. Wang, *A survey on visual surveillance of object motion and Behaviors*, IEEE T. Syst. Man Cy. C **34** (2004), 334–352.
- [10] <http://www.youtube.com/watch?v=0OCJuynUJIo&feature=youtu.be>, user: Jomin Mathew, ITV Sport, Liv. vs Rea.
- [11] D. S. Jang, S. W. Jang and H. I. Choi, *Structured Kalman filter for tracking partially occluded moving objects*, in: S. W. Lee et al. (eds.), Biologically Motivated Computer Vision, Lecture Notes in Computer Science Volume **1811**, 2000, 248–257.
- [12] <http://www.youtube.com/watch?v=P5qP3x-e1Dw>, user: Jdatabean, Java Applet Bouncing Ball3.
- [13] R. E. Kalman, *A new approach to linear filtering and prediction problems*, J. Basic Eng. – T. ASME **82** (1960), 35–45.
- [14] I. A. Karaulova, P. M. Hall, A. D. Marshall, *A hierarchical model of dynamics for tracking people with a single video camera*, in: Proceedings of British Machine Vision Conference, 2000, 262–352.
- [15] M. Kass, A. Witkin and D. Terzopoulos, *Snakes: Active contour models*, Int. J. Comput. Vision **1** (1988), 321–331.
- [16] <http://www.youtube.com/watch?v=Ktxe35uEEDc>, user: Kauzmic, Massive P-51 Takeoff and Flyby!
- [17] C. Li, C. Y. Kao, J. C. Gore and Z. Ding, *Implicit active contour driven by local binary fitting energy*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), Minneapolis, Minn, USA, June 2007, 1–7.
- [18] M. Maes, *On a cyclic string-to-string correction problem*, Inform. Process. Lett. **35** (1990), 73–78.
- [19] M. Mete and N. M. Sirakov, *Lesion detection in dermoscopy images with novel density-based and active contour approaches*, BMC Bioinformatics **11** (2010), 13 pp.
- [20] N. M. Sirakov, H. Kojouharov and N. N. Sirakova, *Neutrophils and bacteria tracking by active contours in video sequences*, IEEE SSI AI 2010, May 23–25, Austin, Texas, Pub. by IEEE Computer Society, 2010, 5–8.
- [21] N. M. Sirakov and K. Ushkala, *An integral active contour model for convex hull and boundary extraction*, in: G. Bebis et al. (eds.), ISVC '09 Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II, Lecture Notes in Computer Science **5876**, Springer, 2009, 1031–1040.
- [22] R. A. Wagner and M. J. Fisher, *The string-to-string correction problem*, J. ACM **21** (1975), 168–173.
- [23] M. S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*, Chapman & Hall, London, 1995.
- [24] T. Yang, J. Li, Q. Pan and S. Z. Li, *Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes*, Computer Vision and Pattern Recognition **1** (2005), 970–975.
- [25] A. Yilmaz, O. Javed and M. Shah, *Object tracking: A survey*, ACM Comput. Surv. **38** (2006), 1–45.

Pravinkumar G. Kandhare, Department of Computer Science and Information Systems, Texas A & M. University, Commerce, TX 75428, USA

e-mail: pkandhare@leomail.tamuc.edu

Abdullah N. Arslan, Department of Computer Science and Information Systems, Texas A & M. University, Commerce, TX 75428, USA

e-mail: abdullah.arslan@tamuc.edu

Nikolay Metodiev Sirakov, Department of Computer Science and Information Systems and Department of Mathematics, Texas A & M. University, Commerce, TX 75428, USA

e-mail: nikolay.sirakov@tamuc.edu

