Mathematical and Computational Applications, Vol. 18, No. 3, pp. 373-382, 2013

REAL-TIME MULTIPLE OBJECT TRACKING USING VIRTUAL SHELLS

Taygun Kekec, Mustafa Unel and Hakan Erdogan

Faculty of Engineering and Natural Sciences, Sabanci University, 34956
Orhanlı-Tuzla, Istanbul
ikekec@sabanciuniv.edu, munel@sabanciuniv.edu, haerdogan@sabanciuniv.edu

Abstract- This paper presents a new multiple object tracking framework where arbitrarily shaped complex objects having severe occlusions are successfully tracked in real-time using stationary cameras. The proposed method utilizes the concept of a virtual shell that encloses each moving object in the scene and simplifies analysis of object interactions. Object's state transitions are handled by an event resolution analysis. Such a regional analysis boosts the tracking process. Finally, a pixel level evaluation is employed for all objects to establish successful pixel memberships. The proposed tracking algorithm is experimentally tested on some public databases and our own challenging aquarium setup that contains many different fish interactions.

Key Words- Tracking, Background Subtraction, Kalman Filter, Virtual Shell

1. INTRODUCTION

Object tracking has been a focus of research for decades due to its promising potential for real-time applications such as intelligent user interfaces, navigation systems and surveillance. A successful tracking algorithm is expected to be robust against environmental changes. Main difficulties of the single object tracking problem are appearance changes, abrupt motion and object non-rigidity. One can view multiple object tracking as the problem of running multiple tracker instances on each object. However such an approach is likely to fail because of not exploiting the dependence between tracker instances and object models, running each tracker independently, causing tracker to get stuck on one object and lose the others (see MacCormick and Blake [1]). One needs a more reliable tracker which uses holistic information exploiting the tracking correlation between objects.

Main problem of the multi object tracking systems is object interactions which give rise to occlusions (see Papadourakis *et al.* [2] and Bose *et al.* [3]). While simple scenarios usually include interaction of two objects, real world examples may include several objects interacting with each other that give rise to complex events and cause the tracker to fail. This necessitates development of more robust, reliable and easily scalable object tracking formulations. Argyros *et al.* [4] propose online learning of color for tracking skin. Khan *et al.* [5] employ particle filtering for tracking of multiple objects. Sullivan *et al.* [6] exploit continuity of depth and motion direction for object labeling problem. Yu and Medioni [7] propose a Markov Chain Monte Carlo formulation for multiple object tracking.

Most of the multi object tracking methods, regardless of the number of cameras installed, requires a background subtraction algorithm to detect motion, a prerequisite step for object representation. The occlusion problem is mostly handled in two different ways: merge-split approach and straight-through approach as noted in work of Gabriel *et al*. [8]. In the merge-split based methods (McKenna *et al*. [9] and Bremond *et al*. [10] use appearance cues, Haritaoglu's W4 system [11] uses appearance and motion cues), separate blobs are updated as long as no occlusion is detected. In the case of occlusion, a joint blob is formed and tracked until the objects are splitted. The detection of beginning and termination of occlusion requires an occlusion and split predicate. Straight-through approaches as in Khan's method [12] and Haritaoglu's Hydra system [13] do not handle split and merge cases separately. These approaches continue to track each individual object in the presence of occlusions using various image features. No joint blob, or hypothesis, is formed during the occlusion phase. These methods require an occlusion predicate which acts as a trigger for ongoing occlusion event.

Regardless of the approach that may be taken, a robust and reliable multiple object tracking framework necessitates formal definition of the interaction problem that is boosted by merge and split predicates. Motivated by these observations, in this work, we propose a novel method for solving multiple object tracking problem under severe occlusions. First, occlusion and split predicates are implemented using virtual shells and blobs. Second, a split-merge level analysis is performed using an event resolution step. This step provides information about object interactions with the help of temporal consistence. Third, a straight-through approach is adopted and pixel level analysis is carried out on shells by considering updated events without forming joint objects.

The rest of the paper is organized as follows: In section 2, we present our proposed multiple object tracking framework. In section 3, we provide experimental results on well-known video databases and videos generated from our aquarium setup. The paper is finally concluded with several remarks in section 4 and future directions are indicated.

2. TRACKING USING VIRTUAL SHELLS

Our tracking approach has three important ingredients: a virtual shell model, an event resolution analysis and a pixel membership evaluation. By a virtual shell it is meant a closed-bounded curve or surface that encloses each moving object and handles complex interactions between objects that may have arbitrary shapes and motion. An event resolution analysis based on state transitions is performed using geometric relationships between object shells. This resolution step provides a macro level evaluation for multi-object tracking process. Finally, a micro level analysis, namely a pixel membership evaluation is carried out where all interesting pixels are evaluated and assigned to corresponding objects using a probabilistic approach that utilizes virtual shells and event resolutions.

2.1. Background Subtraction

In most tracking systems, background subtraction is a preprocessing step for motion detection. In the first step of our algorithm, the background subtraction technique detailed in Sheikh's work [14] is applied to the input frame to obtain background and foreground pixels. In the cited work, a three step algorithm is employed to obtain pixel labels. First, a background model is created and maintained. Second, a foreground model is created. Third, both models are used under a MAP-MRF

(Maximum A Posteriori- Markov Random Fields) decision framework. Inspired by the performance evaluations in Benezeth's comparison [15], we have replaced the third step with morphological filtering to achieve real-time performance.

After probabilistic evaluation of background and foreground pixels, each pixel is labeled using a weighted median filter. Outputs of Parzen classifiers for each pixel, indicated by τ in the original algorithm, are fed into the filtering process as weights. Using such a filter guarantees reconstruction of the disconnected parts of an object. Finally, a connected component labeling algorithm is employed to obtain labeled blobs in the input image. This procedure is then continued with a geometric filtering step which is obligatory to avoid false positive regions. Geometric filtering can be utilized in many different ways. One can create a filter based on angle, shape or size constraints. In a scenario where objects with arbitrary and highly complex shapes that may undergo abrupt motion, a size filter will be adequate to remove false positives of background subtraction process.

Enhanced blobs are main regions to create, track and update object hypothesis. However, due to high shape variations in temporal domain, blobs themselves are insufficient for object representation. To overcome this problem, we introduce virtual shells for further processing.

2.2. Virtual Shell Model

In shell model, each object possesses a unique enclosing shell (Fig. 1(a)). Shells provide spatial relaxation, allow prediction of object interactions and possible merge and split events. Object interactions can be interpreted as shell interactions. Object merge events are described as merging of multiple shells. Shell radius defines the interaction range for an object. An object's interaction range is highly correlated to its speed. Thus, rapid objects will have bigger shells whereas slowing down objects' shells will shrink in time as shown in Fig. 1(b). Dynamically sized shells facilitate detection of multiple object interactions.

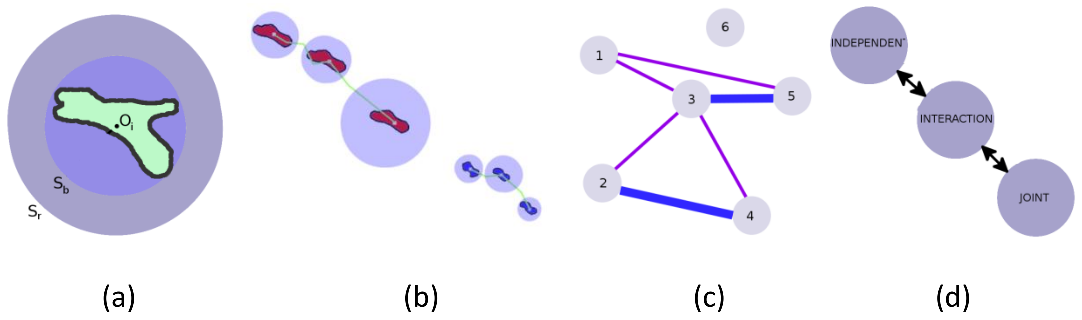


Figure 1. a) An object O_i with its minimum shell S_b and instantaneous shell S_r is depicted. b) Shells dynamically grow and shrink with time c) State transitions of object to object relations d) State graph where thin edges represent INTERACTION and thick edges represent JOINT relationship between objects.

2.3. Event Resolution

Object to object relationships in the scene can be in one of three possible states: INDEPENDENT, INTERACTION and JOINT as shown in Fig. 1(c)-(d). Object state transitions follow a simple but crucial assumption: no two objects can make transition directly from INDEPENDENT to JOINT state or vice versa. An object in JOINT state must first move to the INTERACTION state before becoming an INDEPENDENT object. Thus, the tracking problem can be expressed as keeping track of the states and the positions of each object.

To be more specific, let an object O_i has an event list denoted by $L_{j,t}$ at time t, that keeps record of INTERACTION ($I_{i,j,t}$) and JOINT ($J_{i,j,t}$) relationships with object O_j. An object having no relationship with other objects has an empty list $L_{j,t} = \emptyset$. Proposed shell model provides an infrastructure for the determination of possible state transitions. Let S_i and S_j be the shells of the objects O_i and O_j, respectively. Also let B_i and B_j denote the blobs corresponding to O_i and O_j. In what follows, we shall consider several possible scenarios and analyze each in detail.

- Case I.

$$S_i \cap S_j = \emptyset,\ I_{i,j,t-1},\ J_{i,j,t-1} \notin L_{i,t-1} \tag{1}$$

This condition says that if object shells do not intersect and they had no INTERACTION or JOINT relationship at time $t-1$, then each object is INDEPENDENT of each other.

- Case II.

$$\begin{aligned} S_i \cap S_j \neq \emptyset,\ I_{i,j,t-1},\ J_{i,j,t-1} &\notin L_{i,t-1} \\ L_{j,t} = L_{i,t-1} &\cup \{I_{i,j,t}\} \end{aligned} \tag{2}$$

When two object shells has a non-empty intersection and had no INTERACTION and JOINT relation in their list, one can conclude that this is a start of an INTERACTION event between objects O_i and O_j. Volume of the intersection set accumulates when objects move toward each other. The INTERACTION event must be added to both objects' event list $I_{j,t}$ and $L_{j,t}$.

- Case III

$$\begin{aligned} B_i = B_j,\ S_i \cap S_j \neq \emptyset,\ I_{i,j,t-1} &\in L_{i,t-1} \\ L_{j,t} = L_{i,t-1} \setminus \{I_{i,j,t-1}\} &\cup \{J_{i,j,t}\} \end{aligned} \tag{3}$$

This equation models an incoming JOINT event. If two blobs are identical (single blob), their shells are in interaction, and they were in interaction at time $t-1$, then they have joined into one object. While most of the time object to object relationships can be expressed with either INDEPENDENT or INTERACTION states, JOINT state provides a meaningful and necessary stage when one object fully occludes another one. At the end of occlusion, an occluded object can reappear anywhere but limited to the shell of the occluder object.

- Case IV

$$B_i \neq B_j,\ S_i \cap S_j \neq \emptyset,\ J_{i,j,t-1} \in L_{i,t-1}$$
$$L_{j,t} = L_{i,t-1} \setminus \{J_{i,j,t-1}\} \cup \{I_{i,j,t}\} \tag{4}$$

Two objects in JOINT state can split in time (see Fig. 2). On split event, their shells will have a non-empty intersection set and their blobs will be spatially distinct. The event list must be updated to hold incoming INTERACTION event and discard previous JOINT event between objects.

- Case V

$$S_i \cap S_j = \emptyset,\ I_{i,j,t-1} \in L_{i,t-1}$$
$$L_{j,t} = L_{i,t-1} \setminus \{I_{i,j,t-1}\} \tag{5}$$

Two interacting objects end their interaction which is detected by an empty intersection set between shells. In this case both objects will go to an INDEPENDENT state and isolate themselves from each other as shown in Fig. 2. Previous interaction record must be erased from the event list.

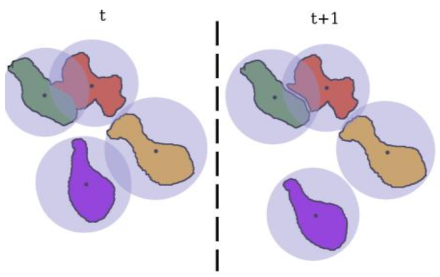


Figure 2. Demonstration of Case IV (green and red) and Case V (purple and orange).

2.4. Pixel Membership Evaluation

After performing an event resolution analysis at object level, each object's nearby pixels in a region of interest must be evaluated for pixel membership. This region of interest is selected as object's shell. As in Fig. 2, there can be both the object pixels and the interacted objects' pixels inside the shell radius of an object. Each pixel is then evaluated for possible membership with each object O_i using the following optimization problem.

$$\hat{c}(p_x) = \arg\max_c P(c|p_x),\ \ \forall p_x \in S_i,\ \ c \in \aleph(L_{j,t}) \tag{6}$$

where $\aleph(.)$ is defined as:

$$\aleph(L_{j,t}) = \{O_i\} \cup \{O_j | O_j \in L_{j,t}\} \tag{7}$$

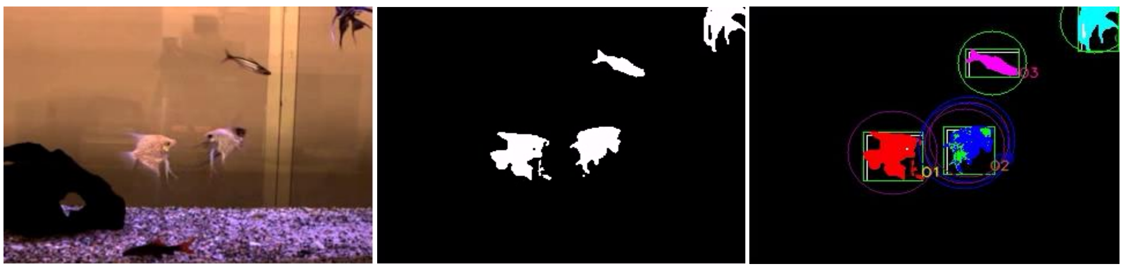
Note that the set $\aleph$ for an object O_i is defined as the union of the object and other objects O_j that are in relations with O_i. All elements of $L_{j,t}$ are checked to determine for possible ownership of the pixel p_x. Then the probability term $P(c|p_x)$ can be computed by utilizing Bayes theorem; i.e.

$$P(c|p_x) = P(p_x|c)\, P(c) \tag{8}$$

where the likelihood term $P(p_x|c)$ is calculated using the color histogram of pixels, and the prior term P_c is modeled as a kernel function corresponding to the shell S_i. In a scenario where frequent occlusions take place and objects undergo abrupt motion, the prior term provides robustness to the pixel evaluation process by imposing the constraint of spatial proximity (to the shell center) on pixels. Pixels far away from the center will have less attraction to belong to that shell.

This kernel function of each object is centered at C_{S_i}, their shell center. Intuitively this saturation introduces some constraint where an object O_j cannot enter membership voting for a pixel if the pixel is outside of its own shell, even if it is in interaction with O_i. Selection of the kernel is a question of accuracy and computational constraints. In our implementation, Epanechnikov kernel [16] has been selected as the kernel function. The pixels that are far away from shell center have lower probabilities to belong to that object.

As an example, in Fig. 3, we consider an occlusion scenario between two objects. Purple colored shells indicate interaction where blue color indicate JOINT event between objects. After utilizing pixel evaluation step, green colored pixels indicate pixels of the occluded object while the blue pixels belong to the occluder.



(a) (b) (c)

Figure 3. a) Aquarium image b) Background subtraction c) Output of the event resolution and pixel membership evaluation where pixels of each object are colored differently for illustration purposes.

2.5. Position Update

After assigning each pixel to the corresponding object, each object's center is updated using member pixels; i.e.

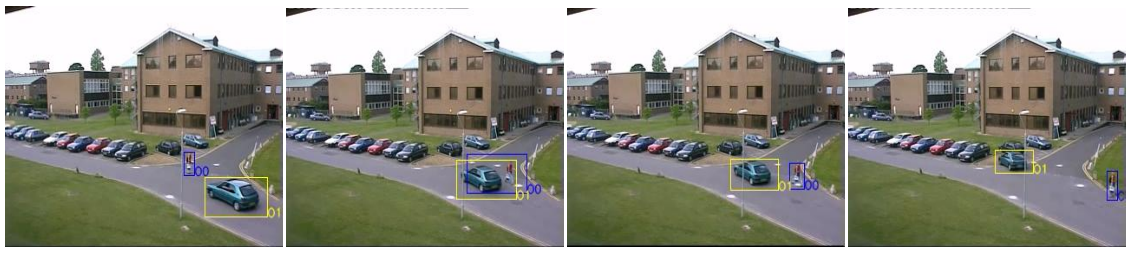
$$C_{i,t+1} = \alpha E[P_i] + (1-\alpha)\hat{P}_{C_{i,t+1}\,|\,C_{i,t}} \tag{9}$$

where P_i represents all member pixels of O_i. The updated position is a linear combination of the current first order moment of P_i, i.e. $E[P_i]$, and the Kalman filter prediction, i.e. $\hat{P}_{C_{i,t+1}|C_{i,t}}$. Integration of filtering into the process helps smooth transitions of object centers in temporal domain avoiding instantaneous noisy estimations.

3. EXPERIMENTS

We present experiments on PETS, Caviar and aquarium sequences. Aquarium images has been acquired using Imaging Source DFK21BF04H Firewire CCD cameras and resized to 320x240 resolution under RGB color space. The ground level of aquarium contains shiny colorful pebbles which can act as false positives to many detection algorithms based on appearance variation. They are suppressed successfully using our background subtraction step as shown in Fig. 3(b). First aquarium sequence contains rapid motion and self-occlusion examples. Second aquarium sequence focuses on several objects' interaction.

In Fig.4 tracker's results on PETS video is shown. A car and a person is moving towards each other, merges and splits after their interaction ends. In our quantitive analysis, we have used the MOTA metric described in Bernardin's work [17] for evaluation of multiple object tracking accuracy. In Fig. 5, quantitative results from PETS sequence are shown where three objects (shown by blue, green, red bars) reside occasionally on the scene. We defined the tracking accuracy for each object as the distance of computed object center to ground truth object center.



(a) (b) (c) (d)

Figure 4. Sequence from PETS database.

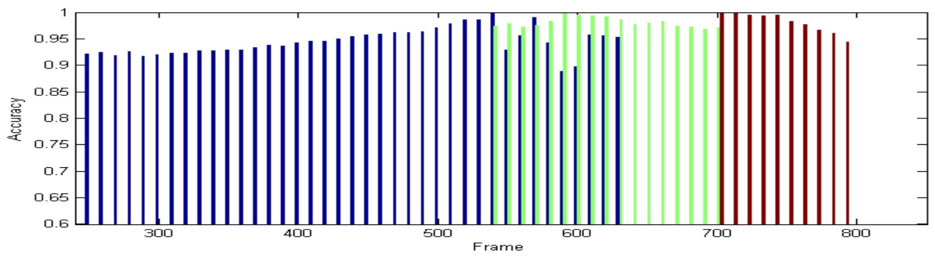
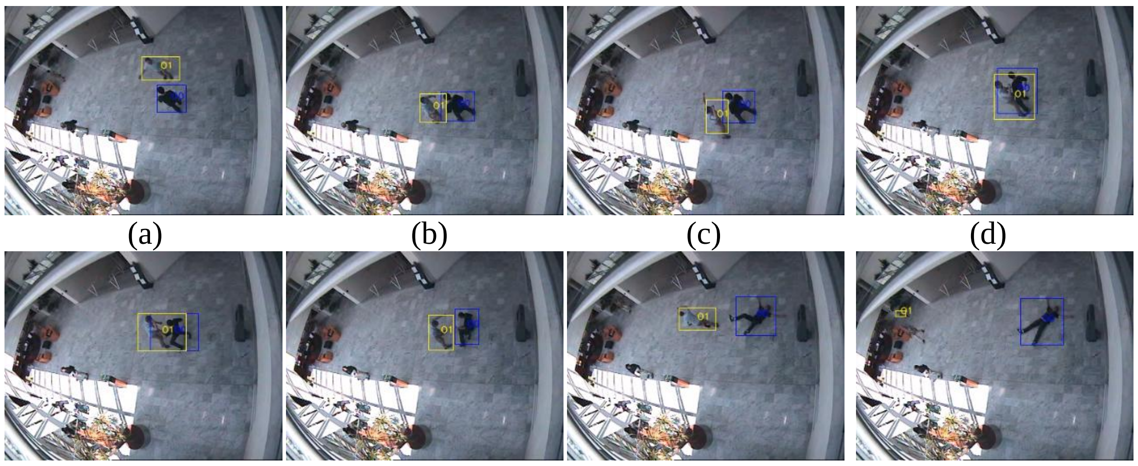


Figure 5. Tracking accuracy for PETS Sequence where each color represents a tracked object.

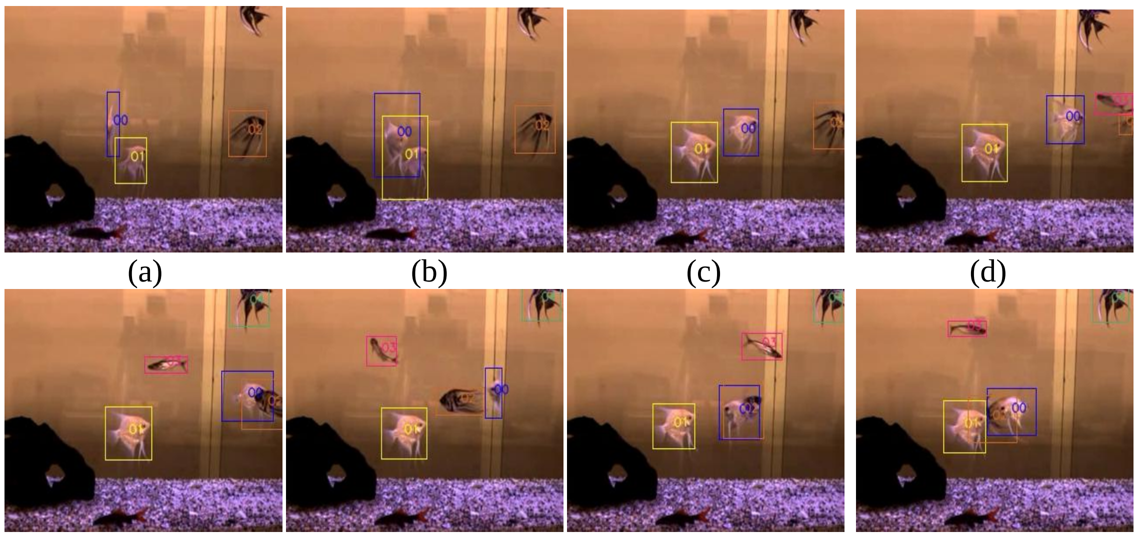
A fight scene from CAVIAR database is shown in Fig. 6. In this scenario, two people move towards each other and start fighting. During fight as in (Fig. 6b-e), severe occlusions are observed in close contact. One man gets down and lies on the floor while other runs away with high speed (Fig. 6(f)-(h)).


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6. Sequence from Caviar database.

In Fig. 7, we show a tracking sequence from our experimental setup; an aquarium containing several fish. While in many scenarios split and merge events are solved with the help of linear motion assumptions, this assumption is not valid in this environment due to abrupt and rapid motion. This is observable in Fig. 7(e) and (f) which shows that small fish in pink bounded box performs a sudden turn. Moreover, self-occlusion and object interactions are very frequent. In Fig. 7(a) fish in blue bounded box severely self-occludes which reduces number of pixels associated with it and suffers from a noticeable shape variation. Interaction of two fish is observable in Fig. 7(g) and (h). Fish in blue box appears in front and with maneuver of neighbor fish they change roles, it appears behind of its neighbor in a very short time.


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. A tracking sequence from aquarium where three fish interact.

In Fig. 8, we show a second sequence from aquarium environment. This sequence has a total of 350 frames. In this sequence we track six fish. Four of them go into an interaction eventually (Fig.8(i)-(m)). Note that objects' interaction complexity is high as many shells collide with each other and pixel classification task extends to several objects. At the end of the interaction, classification succeeds and each fish's location is preserved (Fig. 8(n)-(p)). This sequence shows that arbitrary number of objects' interaction can be coped with using our framework.

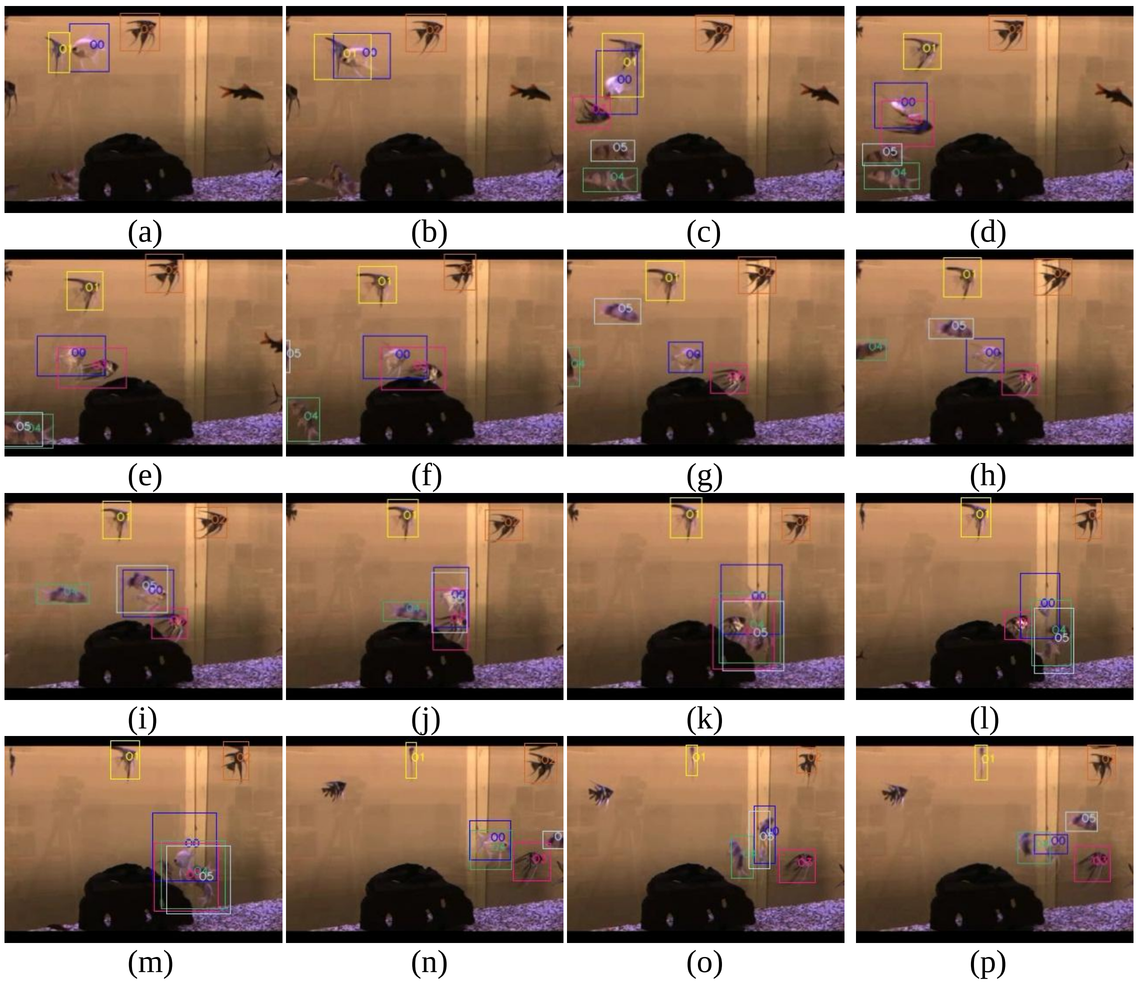


Figure 8. A tracking sequence from aquarium where several fish interact.

4. CONCLUSION AND FUTURE WORKS

We have now presented a real-time multiple object tracking algorithm. Based on virtual shell modeling, the algorithm uses event resolution analysis and pixel class evaluation to achieve robust tracking. The algorithm has been tested on some well-known databases and also on our challenging aquarium setup where multiple objects interact with each other and create very complicated occlusion scenarios. Algorithm is fast, repeatable and robust. The experimental results are quite promising.

As a future work, we plan to integrate several robust cues along with color to improve pixel discrimination, devise better handling mechanisms for disputed pixels, and develop soft assignment rules for pixels to increase accuracy.

5. REFERENCES

1. J. MacCormick and A. Blake, A probabilistic exclusion principle for tracking multiple objects, *International Journal of Computer Vision (IJCV)* **1**, 572-578 2000.
2. V. Papadourakis and A. Argyros, Multiple objects tracking in the presence of long-term occlusions, *Computer Vision and Image Understanding (CVIU)* **114(7)**, 835–846, 2010.
3. B. Bose, X. Wang and E. Grimson, Multi-class object tracking algorithm that handles fragmentation and grouping, *Computer Vision and Pattern Recognition (CVPR'07)*, 1-8, 2007.
4. A. A. Argyros, and M. I. A. Lourakis, Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera, *European Conference on Computer Vision (ECCV'04)*, 368–379, 2004.
5. Z. Khan, T. Balch, and F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, *Pattern Analysis and Machine Intelligence (PAMI)* **27(11),** 1805-1819, 2005.
6. J. Sullivan and S. Carlsson, Tracking and labelling of interacting multiple targets, *European Conference on Computer Vision (ECCV'06)*, 619–632, 2006.
7. Q. Yu and G. Medioni, Multiple-Target Tracking by Spatiotemporal Monte Carlo Markov Chain Data Association, *Pattern Analysis and Machine Intelligence (PAMI),* **31(12),** 2196–2210, 2009.
8. P. F. Gabriel, J. G. Verly, J. H. Piater and A. Genon, The state of the art in multiple object tracking under occlusion in video sequences, *Advanced Concepts for Intelligent Vision Systems (ACIVS'03),* 166–173, 2003.
9. S. J. Mckenna, S. Jabri, Z. Duric, H. Wechsler and A. Rosenfeld, Tracking groups of people, *Computer Vision and Image Understanding (CVIU)* **80**, 42–56, 2000.
10. F. Bremond and M. Thonnat, Tracking multiple nonrigid objects in video sequences, *IEEE Transactions on Circuits and Systems (TCAS)* **8(5),** 585–591, 1998.
11. I. Haritaoglu, D. Harwood and L. S. Davis, W4: Real-Time Surveillance of People and Their Activities, *Pattern Analysis and Machine Intelligence (PAMI)* **22(8),** 222–227, 2000.
12. S. Khan and M. Shah, Tracking People in Presence of Occlusion, *Asian Conference on Computer Vision (ACCV'00)*, 1132–1137, 2000.
13. I. Haritaoglu, D. Harwood, and L. S. Davis, Hydra: multiple people detection and tracking using silhouettes, *IEEE Workshop on Visual Surveillance(VS'99)*, 6–13, 1999.
14. Y. Sheikh and M. Shah, Bayesian modeling of dynamic scenes for object detection, *Pattern Analysis and Machine Intelligence (PAMI)* **27**, 1778–1792, 2005.
15. Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent and C. Rosenberger, Comparative study of background subtraction algorithms, *Journal of Electronic Imaging (JEI)* **19**(3), 033003-033003, 2010.
16. B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall/CRC, 1986.
17. K. Bernardin, E. Elbs and R. Stiefelhagen, Multiple object tracking performance metrics and evaluation in a smart room environment, *IEEE Workshop on Visual Surveillance (VS'06)*, 2006.