A multiple hypothesis tracking method with fragmentation handling

Atousa Torabi
École Polytechnique de Montréal
P.O. Box 6079, Station Centre-ville
Montréal (Québec), Canada, H3C 3A7
atousa.torabi@polymtl.ca

Guillaume-Alexandre Bilodeau
École Polytechnique de Montréal
P.O. Box 6079, Station Centre-ville
Montréal (Québec), Canada, H3C 3A7
guillaume-alexandre.bilodeau@polymtl.ca

Abstract

In this paper, we present a new multiple hypotheses tracking (MHT) approach. Our tracking method is suitable for online applications, because it labels objects at every frame and estimates the best computed trajectories up to the current frame. In this work we address the problems of object merging and splitting (occlusions) and object fragmentations. Object fragmentation resulting from imperfect background subtraction can easily be confused with splitting objects in a scene, especially in close range surveillance applications. This subject is not addressed in most MHT methods. In this work, we propose a framework for MHT which distinguishes fragmentation and splitting using their spatial and temporal characteristics and by generating hypotheses only for splitting cases using observation in later frames. This approach results in a more accurate data association and a reduced size of the hypothesis graph. Our tracking method is evaluated with various indoor videos.

1 Introduction

Multiple object tracking (MOT) consists of two main parts; the first part involves localizing the targets in an image (object detection) and the second part involves associating detected regions or observations to tracked targets (data association). The critical part of MOT is data association. Data association deals with two main issues: 1) merging and splitting objects in the scene (occlusion), and 2) shortcomings of object detection methods such as misdetection of some parts of an interesting object (fragmentation). In this work, a MOT method, which is able to deal with object occlusion and object fragmentation, is proposed and the shortcomings of previous works are addressed.

The first issue concerning MOT is occlusion resulting from interactions between objects in the scene. Because of dramatic changes in the appearance of occluding targets, it is extremely challenginh to maintain target's identity using object appearance or motion information. Therefore, an object re-identification is required when occluding targets split. Graph representation of tracked targets enables us to integrate spatial and temporal information and propagate the object identity both forward and backward in time before merging and after splitting. Graph representation has been used previously in several studies [5, 3, 2]. One of the classical graph-based MOT methods is multiple hypothesis tracking (MHT) introduced by Reid [9]. The MHT approach has been also applied in many works such as [4, 8, 1]. The basic idea of MHT is considering information of several frames, keeping all possible hypotheses about the trajectories of objects, and finally choosing the most probable hypothesis for the trajectory of objects. The main drawback of this method is exponential growth of hypotheses. In [8], a real-time MHT is proposed that controls the exponential growth of the hypotheses by pruning unlikely hypotheses to meet criteria for complexity and memory usage. In our work, hypothesis generation is only performed for splitting cases. In other cases when there is no identity ambiguity, sequential data association is applied. Our data association strategy reduces the number of hypotheses tremendously without graph pruning (see further information in section 5).

In most MHT studies, data association is only based on previous observation and is not guided by later frames. In the cases where in few frames the object appearance changes considerably because of illumination variations or erroneous foreground segmentation, the observations may not provide adequate information for data association in later frames and consequently, the errors from these wrongly generated hypotheses propagate in later frames. In [1], a MHT is proposed to have a reliable tracking system by exploiting later frames and relating them to previous observations. This approach is limited, as it requires the whole video before producing the tracking results because of an offline upward analysis that relates later observations to previous ones. In our work, this data association strategy is adopted for continuous stream of video and upwards analysis is performed at each frame. Our MOT is able to correct erroneous data associations in previous frames by using the observations in the current frame (see section 6 for details).

The second issue that MOT should deal with is imperfections of the object detection outputs such as misdetection of some parts of an interesting objects [10] which results in an object region fragmenting into few sub-regions. In close-range videos, object fragmentation may visually be confused with splitting of a group of objects in the scene. In Reid's MHT [9] data associations are assumed to be one-to-one. This means a target is associated with only one measurement (detected region in the scene) and vice versa. This method cannot associate all fragmented regions related to one object to a corresponding tracked target. In a recent work [7], a MHT is proposed which distinguishes splitting and fragmentation using constant object size information. Constant object size is a strong constraint that does not take into account the effect of shadows on the size of blobs and camera position. Our MHT approach handles object fragmentation before hypothesis generation. In our work, splitting and fragmentation are distinguished by monitoring the average velocity of the blobs (observations) for a few frames. This is based on the fact that fragmented blobs show coherent behaviour and move with the same average velocity.

In the remainder of this paper, the architecture and the main steps of the tracking algorithm are presented. We also report experimental results of indoor videos.

2 System overview

Our visual tracking system is designed for a single static monocular camera. Fig. 1 shows the main steps of our tracking system. Before tracking, object detection is performed using Gaussian mixture background subtraction and shadow elimination proposed in [6].

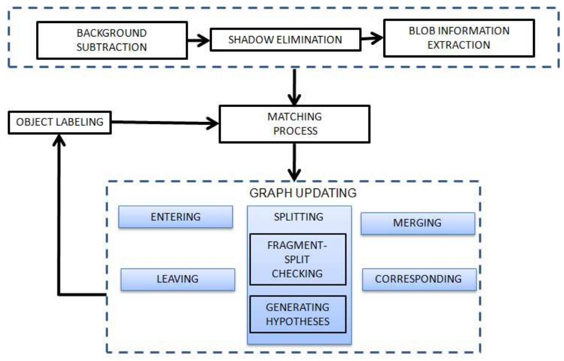

Figure 1. Overview of the tracking system.

Our algorithm has three main steps. In the first step, matching is performed between detected blobs in previous frame $t-1$ and current frame t to detect entering, leaving, correspondence, merging, and splitting events, and to determine the state of blobs in current frame as *single object*, *group of objects*, and *possible fragment*. We use two graphs for tracking. An event graph is built to record all the splitting and merging between blobs and store their appearance information while they are tracked. A hypothesis graph is built to generate hypothesis for handling data association of split objects. In the second step, named graph updating, the necessary operations are performed to update the information of both graphs after an event. In the last step, object labelling is done for all objects in the scene and the trajectory is estimated for objects which have left the scene in the current frame.

3 Event graph and hypothesis graph

Fig. 2 shows an event graph with its corresponding hypothesis graph. The event graph represents all blobs with their merging and splitting events during tracking. Each vertex of this graph (track node) stores appearance information of a blob including centroid positions, adaptive color histogram (see section 5), blob state, and the frame number of the last update in the node. Edges represent merging and splitting events among the blobs.

The hypothesis graph is a directed weighted graph. The vertices of this graph (hypotheses nodes) simply correspond to track nodes of the event graph belonging to entering blobs (blobs which appear in scene) and split blobs (blobs which come apart from a group blob or a single blob). Identified group blob do not have hypothesis nodes. This is because hypothesis nodes are used to solve the data association problem before and after object interactions. The weight of each edge $n_i n_j$ which represents a hypothesis is defined as

$$\omega\left(n_i n_j\right) = \left|AH\left(n_i\right) - AH\left(n_j\right)\right|, \tag{1}$$

where $\omega\left(n_i n_j\right)$ is the Euclidean distance between two adaptive color histograms of two blobs belonging to hypothesis nodes n_i and n_j.

In practice, the edge information are recorded in the nodes. Hence, for each hypothesis node n_i, three sets of nodes called S (Source), E (End) and BH (Best Hypotheses) are defined as

$$S\left(n_i\right) = \left\{\forall n_j | \exists n_j n_i\right\}, \tag{2}$$

$$E\left(n_i\right) = \left\{\forall n_k | \exists n_i n_k\right\} \text{ and} \tag{3}$$

$$BH\left(n_i\right) = \left\{\forall n_j \in S\left(n_i\right) | E_1\left(n_j\right) = n_i\right\}. \tag{4}$$

The sets of Eq. 2 and Eq. 3 are ordered increasingly based on the weights of their common edges with n_i. In Eq. 4, BH can be empty or contain one or more elements. E_1 is

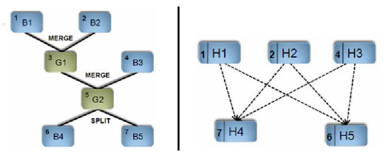

Figure 2. Event graph (left), hypothesis graph (right). In the hypothesis graph, the number at the left of each hypothesis node corresponds to a track node in event graph with the same number in the upper left corner.

the first element of E. S, E, and BH sets are used for object labelling and for finding trajectories. It is important to notice that event graph and hypothesis graph may be composed of more than one component (subgraph) since the connections between nodes reflect the interactions that happened between the blobs during tracking (Two blobs that do not interact are not connected).

4 Matching process

In the first step of our algorithm, a distance matrix is computed to find the possible corresponding blobs $B_i(t-1)$ and $B_j(t)$ in two consecutive frames along with their appearance dissimilarities. It is defined as

$$D_{t-1}^{t}(i,j) = \begin{cases} d(h_{B_i(t-1)}, h_{B_j(t)}) & \text{if overlapped} \\ -1 & \text{otherwise} \end{cases}, \tag{5}$$

where $D_{t-1}^{t}(i,j)$ is the color histogram intersection distance between the ith blob in frame $t-1$ and the jth blob in frame t, if the two blobs bounding boxes overlap. Otherwise, these two blobs cannot match each other and their corresponding element in the matrix is -1. This assumption is based on the fact that a blob should move on a short distance in two successive frames because of the frame rate of the camera. Therefore, its bounding boxes in previous and current frames should overlap. The size of distance matrix is $N \times M$, where N is the number of blobs in frame $t-1$ and M is the number of blobs in frame t. The color histogram intersection is defined as

$$d(h_{B_i(t-1)}, h_{B_j(t)}) = \frac{\sum_{k=1}^{K} min(h_{B_i(t-1)}(k), h_{B_j(t)}(k))}{\sum_{k=1}^{K} h_{B_i(t-1)}(k)}, \tag{6}$$

where $h_{B_i(t-1)}$ and $h_{B_j(t)}$ are the color histogram of ith blob in frame $t-1$ and the jth blob in frame t, and K is the number of color histogram bins.

A blob in frame $t-1$ matches with a blob in frame t if the dissimilarity is not -1. The events such as entering, leaving, merging, and splitting are detected by finding the matching blobs in two consecutive frames using the distance matrix. The state of each blob is determined. If entering is detected, the state is *single object*, if merging is detected (many blobs in frame $t-1$ match with one in frame t), the state of merged blob is *group of object*, if a splitting is detected (one blob in frame $t-1$ matches with many blobs in frame t), the states of splitting blobs are *possible fragment*, and if correspondence is detected (one-to-one match), the state remains unchanged.

5 Graph updating

Event graph and hypothesis graph are updated depending on each detected event in matching process:

- If a blob in current frame t is an appearing object, a track node in event graph and a hypothesis node in hypothesis graph are added.

- If correspondence is detected between two blobs in frame $t-1$ and t, the track node in event graph belonging to the object is updated by adding its blob centroid in current frame t, adding current frame number, and updating its adaptive color histogram using

$$AH_{B(t)} = \sum_{k=1}^{K} \alpha AH_{B(t-1)}(k) + (1-\alpha) h_{B(t)}(k). \tag{7}$$

In Eq. 7, $AH_{B(t-1)}$ is the adaptive color histogram of blob B at frame $t-1$, K is the number of color histogram bins, and $h_{B(t)}$ is the color histogram of blob B at frame t, and α (varying between 0 and 1) is an adaptation parameter. Updating track node for correspondence event is equivalent to sequential data association for blobs which are not in situation of possible identification uncertainty. This choice is based on the fact that if two blob regions in two consecutive frames are found to be similar with a one-to-one matching, it is very likely that they are associated with the same object.

- If some blobs in frame $t-1$ are merged into a single blob in the current frame t, tracking of merging blobs is stopped and a new track node in event graph for group blob is initiated.

- If a blob in frame $t-1$ has disappeared from the field of view of camera, its track node in event graph is deactivated and its trajectory is calculated (section 6).

- If splitting is detected, more processing is required as described in the following subsections.

5.1 Splitting event

The most crucial event for data association in a merge-split tracking approach is splitting, because the uncertainties about identities of splitting blobs need to be handled. Fig. 3 shows the inter-object occlusion handling steps for a splitting event. These steps are detailed in the following subsections.

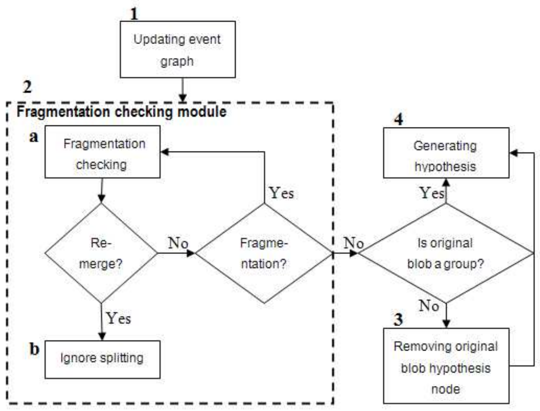

Figure 3. The steps of inter-object occlusion handling.

5.1.1 Event graph updating

In the frame where a splitting event is detected, for all split blobs with the state of $possible fragment$, track nodes are added with related edges in the event graph (step 1, Fig. 3). Then splitting is distinguished from fragmentation in the next step.

5.1.2 Fragmentation checking

Because of the nature of background subtraction methods, whenever a moving object in the scene has a similar color as the background, some parts of the moving object may not be detected as foreground and cause fragmentation. The goal of fragmentation checking module (step 2, Fig. 3) is distinguishing between splitting and fragmentations. To do this, for N frames ($N = 4$ based on experiments), the system tracks the original blob (the blob before splitting) as a whole, using the summation of the feature vectors of all its possible fragmented blobs. It also tracks all possible fragmented blobs individually. If within N consecutive frames after splitting all possible fragments are merged to one blob, it will be assumed that the event was fragmentation. In this case, splitting is ignored (step 2b, Fig. 3) and the tracking of the original object will be continued. Otherwise, after N frames, fragmentation checking (step 2a, Fig. 3) distinguishes between fragmentation and splitting based on the fact that fragmented blobs: 1) belong to one target, 2) are close in the image, and 3) exhibit coherent motion over a few frames. In other words, the maximum differences in the average velocity between any pair of possible fragmented blobs from the same object must be less than a fixed threshold, and the distance moved by the centroids of these blobs relative to each other is much smaller than the overall distance moved by the centroid of the original object. If a fragmentation is detected, we continue tracking the original object and its fragmented blobs. Fragmentation checking will be repeated. But if a splitting is detected, we stop tracking the original object and we continue tracking the split blobs as individual objects. The states of split blobs become *single object* and we exit the fragmentation checking module.

5.1.3 Removing hypothesis node

Our algorithm has hypothesis nodes only for blobs appearing in the scene and split blobs, but it does not have hypothesis nodes for group blobs (blobs for which it is known that they include more than one tracking targets). This means that if a blob (with the state *single object*) associated to a hypothesis node splits in later frames, the blob state will be corrected to *group of objects* and its node will be deleted along with its related edges from the hypothesis graph (step 3, Fig. 3). In this way, we keep hypothesis nodes only for individual objects to have smaller search space in hypothesis graph and more accurate data association.

5.1.4 Generating hypothesis

To generate the hypotheses (step 4, Fig. 3), first for determined split blobs, hypothesis nodes are initiated. Then S, E, and BH sets of all the nodes in the same subgraph as the newly initiated nodes are updated. Generating hypothesis only for nodes in the corresponding subgraph and not for the other nodes in the hypothesis graph is part of our strategy to reduce the number of hypotheses. To do the update, newly initiated nodes are added to the E sets of the nodes from previous frames in the subgraph, and the previous nodes in the subgraph are added in the S sets of the newly initiated nodes. Also, the BH sets of newly initiated hypothesis nodes are created according to their S sets. In other words, all the nodes in the subgraph are connected together with directed edges from the past hypothesis nodes to new hypothesis nodes. Weight of each directed edge is the likelihood that the source node and the end node have the same appearance and it is calculated using Eq. 1. After updating E sets of the previous hypothesis nodes, if the first elements of these sets are changed (S sets and E sets are always ordered increasingly), consecutively for some other nodes in the same subgraph, BH sets are updated. This is based on the fact that the intersection of two BH sets for two different nodes should be empty. The adaptive color histograms is used for generating hypothesis (likelihood between two nodes), because it gives the global color information of the blob during several frames and it helps reducing the effect of dramatic changes in color histograms of blobs caused by short-time variations of lighting or shadows.

6 Object labelling and finding trajectory

The goal of object labelling is identifying each tracked blob with a label in the current frame. For correspondence event, the blob's label in frame t is the same as the blob's label in frame $t-1$. For merging, the merged blob's label in frame t is the labels of all merging blobs in frame $t-1$. For entering blob in frame t, the label is a new one.

For splitting, the label of a split blob in frame t is determined by processing the hypothesis graph. To do this, we traverse the hypothesis graph bottom-up, from the latest frame, starting from split blob's hypothesis node n_i . To do this, TN (Traversing Nodes) set is initialized by

$$TN_0(n_i) = \phi, \tag{8}$$

and it will be updated by

$$TN_t(n_i) = (TN_{t-1}(n_i) \cup BH(n_{current})) - n_{next}. \tag{9}$$

In Eq. 9, $n_{current}$ is the current node during graph traversal (at first $n_{current}$ is n_i and $TN_{t-1}(n_i)$ is ϕ), $TN_t(n_i)$ is a set of possible next destination nodes in current frame t, and n_{next} is the next node to traverse in the graph chosen with two criteria: 1) n_{next} exists in either $BH(n_{current})$ or $TN_{t-1}(n_i)$ 2) n_{next} has the closest temporal relation with $n_{current}$. It is important to notice that if in $BH(n_{current})$ or $TN_{t-1}(n_i)$ more than one node obey n_{next} criteria, we traverse these nodes separately. Traversing graph upward and updating TN set will be continued until we reach a node for which the TN set will become empty (nowhere to go next). The split blob is labelled with the label of the blob that we reach after traversal in hypothesis graph. A hypothesis node belonging to a split blob which has an empty BH set before starting graph traversal is a new appearing object which gets a new label.

When an object has left the scene, we construct its trajectory. To do this, the hypothesis graph is traversed in the same way as for labelling to get its path in the hypothesis graph. However, in the hypothesis graph, some parts of the trajectory (when the object was tracked in a group) are missing, because group blobs have no nodes in hypothesis graph. The missing parts of the path are recovered by completing it with help of the event graph.

6.1 Illustrative example

Fig. 4 illustrates an example for trajectory construction and error correction in data association (object labelling) for two objects that occlude each other two times. In Fig. 4A (left) (event graph), B_1 and B_2 are merged into G_1, and then G_1 splits to B_3 and B_4. In the Fig. 4A (right) (hypothesis graph), edges show the generated hypotheses for B_1, B_2, B_3, and B_4. The thicker edges represent the best hypotheses. We suppose that because of dramatic changes in color histograms, split blobs are erroneously matched. This means B_3 seems more similar to B_2 (smaller dissimilarity) while, in fact, B_3 is the same object as B_1, and B_4 seems more similar to B_1 while, in fact, B_4 is the same object as B_2. In later frames, in Fig. 4B (left) for the second time B_3 and B_4 are merged into G_2, and then G_2 splits to B_5 and B_6. Finally, let us suppose the B_5 has left the scene and we need to compute its trajectory. Fig. 4B (right) shows that for split blobs (B_5 and B_6) hypotheses are generated not only from hypothesis nodes belonging to B_3 and B_4 but also from hypothesis nodes belong to B_1 and B_2. This is the strategy discussed in the introduction to allow data association to be guided by later frames. Indeed, B_5 is re-identified correctly as the same target as B_1, and erroneously identified B_3 and B_4 in Fig. 4A are corrected based on the observation of B_5 and B_6. To find the trajectory of B_5, the hypothesis graph is traversed starting from H_5. First H_5 is the current node and $TN_1(H_5)$ is $\{H_3, H_1\}$, then H_3 is selected as the next node based on the two criteria and $TN_2(H_5)$ updates to $\{H_1\}$. At last the next selected node is H_1 and $TN_3(H_5)$ updates to ϕ because H_1 is not connected to any source node (BH is empty). So the trajectory up to now is $\{B_5, B_3, B_1\}$, which are the corresponding track nodes of $\{H_5, H_3, H_1\}$. The missing parts of the trajectory of B_5 is recovered by searching in event graph. Indeed, to go from B_5 to B_3 and then B_3 to B_1 in the event graph, we have to pass by G_2 and G_1. They are added to obtain the complete trajectory $\{B_5, G_2, B_3, G_1, B_1\}$.

7 Experiments

7.1 Experimental methodology

Our algorithm is written in C++. We use color histogram in the HSV color space with $6 \times 3 \times 3$ bins for matching process and generating hypothesis. Our algorithm uses only one fixed threshold for fragmentation handling (see Section 5.1.2). This threshold is for the average velocity difference between objects. For all experiment, it is fixed at 3 pixels per frame. For background subtraction, we use the method of [11] using the parameters of table 1.

For our experiments, we used nine video excerpts. The first seven video excerpts (LABSEQ1-LABSEQ7, 7.5

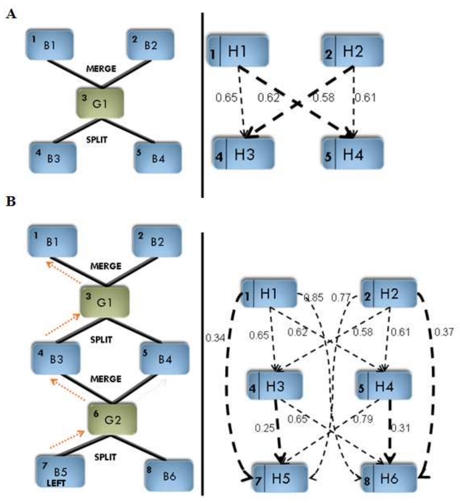

Figure 4. A) Event (left) and hypothesis graph (right) after one merging and splitting. B) The same graph updated after a second merging and splitting. The number at the left of each hypothesis node corresponds to a track node in event graph with the same number in the upper left corner of track node.

fps, and with between 201 to 569 frames of 320 × 240 pixels) are chosen from LABSEQ video dataset captured at our laboratory involving walkers interacting. The hallway video excerpt (25 fps and 1226 frames of 384 × 288 pixels) and the shopping center video excerpt (25 fps and 1675 frames of 384 × 288 pixels) are chosen from the dataset of the CAVIAR project (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/).

To evaluate quantitatively the effectiveness of our tracking system, we used a semi-automatic interactive ground-truth generator to generate ground-truth trajectories of tracking targets in a video sequence and compare them with the computed trajectories of our tracking system. The centroid of blobs for the ground truth is generated after background subtraction which means the centroind position is computed on detected foreground regions and not the complete image. We use four metrics: 1) f_p is the number of false computed trajectories without corresponding ground-truth trajectories; 2) f_n is the number of ground-truth trajectories without corresponding computed trajectories in our tracking system; 3) APE(Average Position Error) is the average of all the position errors belonging to computed trajectories (position error is the Euclidean distance between blob centroid positions of computed trajectory and its corresponding ground-truth trajectory), the unit of this error is pixel; 4) ATI(Average Track Incompleteness) is the average track incompleteness of all computed trajectories (track incompleteness of a computed trajectory is the number of frames that are missing from the computed trajectory plus the number of frames that exist in computed trajectory but not in the ground-truth trajectory divided by the number of frames present in both computed and ground-truth trajectories).

Table 1. Parameters for background subtraction

Video	K	T_g	β	T_b
Hallway	3	2.4	0.0005	0.01
Shopping	5	2.4	0.05	0.01
LABSEQ (all videos)	3	3.4	0.09	0.5

K: Number of Gaussian distributions, T_g: foreground/background threshold (in number of standard deviation), β: Learning rate, T_b: Proportion of background distribution.

7.2 Results and discussion

In this section, first we show qualitative results and then the quantitative results of our experiments.

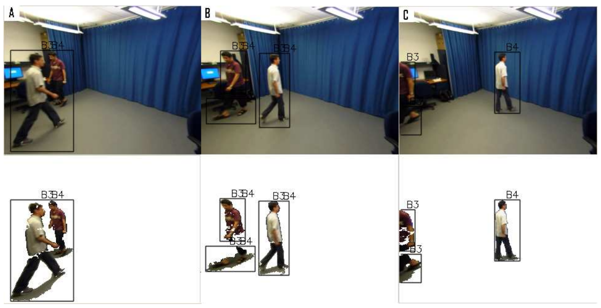

Figure 5. A,B, and C are frames from LABSEQ1 video sequence. First row are images of the original video sequence and the second row are tracking results with our algorithm.

Fig. 5 shows three frames of LABSEQ1 video. This video is chosen to illustrate the robustness of our algorithm to severe blob fragmentation and splitting at the same time. There are also shadows on the floor which change the object size. Fig. 5A shows two persons merged into a group. The label of the group is the labels of both tracked targets in the group. Then, these two persons separate in Fig. 5B. At the same time, there is also fragmentation because a part of the blob's region belonging to one of these persons has the same color as the background. During four frames after splitting (see section 5.1.2), our system does not make decision about labels of blobs in the scene. This is why in Fig. 5B all possible split blobs have the same labels as the group. After four frames, our system handles this challenging situation and differentiates between splitting and fragmentation by using the information of velocities and moving distances between centroids of the possible split blobs (Fig. 5C). Our system also supports objects leaving the scene while they are fragmented (Fig. 5C).

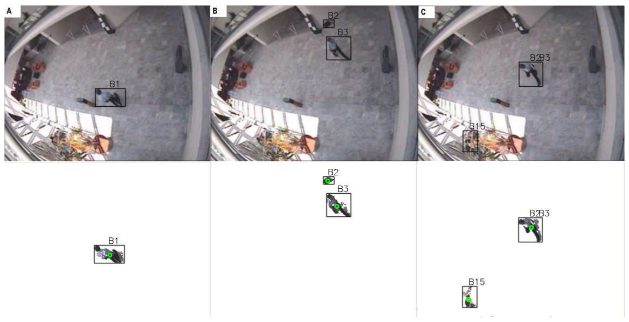

Figure 6. A, B, and C are frames from CAVIAR video sequence (hallway).

Fig. 6 shows three frames of CAVIAR's hallway video. This video illustrates the effectiveness of our algorithm to detect individual tracking targets which initially appear as a group. Fig. 6A shows a person entering the scene with a bag. Our algorithm detects it as one individual target. In Fig. 6B the person puts the bag on the floor. The bag is detected as a new target because it does not correspond to a fragment by the criteria of our algorithm. The hypothesis graph is then updated to take into account this new knowledge by removing the unnecessary group node. Previous individual targets become a group target and first part of the trajectory of bag is reconstructed including the trajectory when it was in a group (Fig.7 shows the trajectories of bag and the person). Classical MHT [9] cannot handle entering blob as a group because the data association approach is one-to-one. Also the recent MHT [7] which can handle entering as a group, is not robust to this video because it only tracks moving targets. The bag is stationary and therefore it will be detected as noise. After some frames, in Fig. 6C the person picks up the bag and leaves the scene.

To evaluate other scenarios, we selected another video (shopping center) from CAVIAR dataset. This video is a challenging video since there are reflections on the floor, blob fragmentations caused by object detection, and object size variations because objects move towards the camera. This video shows the effectiveness of our algorithm for multiple object tracking. Fig. 8A shows three people going out of a store and then walking along the corridor altogether. Later, in Fig. 8B and Fig. 8C, they separate and two of them have an interaction with another person which enters

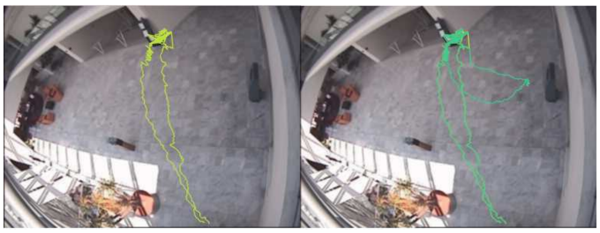

Figure 7. Left image: trajectory of bag, right image: trajectory of person, from CAVIAR video sequence (hallway).

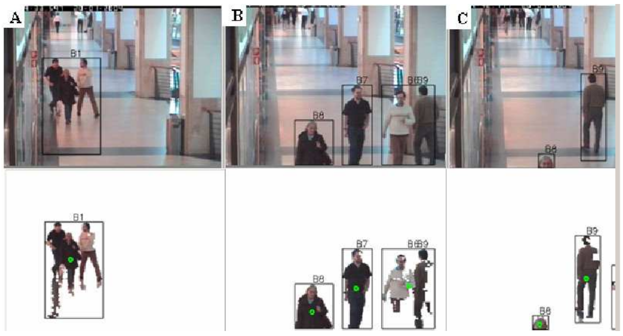

Figure 8. A, B, and C are frames from CAVIAR video sequence (shopping center).

the field of view of the camera while others are leaving the scene. Our algorithm succeeds to handle all the interactions between objects and it was robust to shortcomings of object detection such as fragmentations and shadows on the floor because of using adaptive appearance model and fragmentation handling which are insensitive to object size.

We report the quantitative tracking results of our experiments in table 2. In the table 2, NF is the number of video frames and NT is the number of ground-truth trajectories. It is important to notice that the corresponding objects to these trajectories had an overlapping time for being in the field of view of camera. For LABSEQ1 video sequence, our system has a small average position error. For CAVIAR video sequence (shopping center) with four tracking targets, the first three persons have centroid position errors which result to a considerable average position error. This is because they walked for a long time together in the same direction and with the same speed along the corridor. The fragmentation checking module can detect the splitting only after some delay. It results in all three persons being tracked as a group for a while and thus with less precision in their location. The other error for object labelled as B_1 in the group and B_7 as individual (see figure 8), is that it came into the scene in a group with two other peo-

Video	NF	NT	f_p	f_n	APE	ATI
Hallway	1226	2	0	0	10.3678	0
Shopping	1675	4	1	0	18.5803	0.0142
LABSEQ1	201	2	0	0	0.6895	0
LABSEQ2	368	3	2	0	1.5908	0
LABSEQ3	569	3	2	1	0.5007	0
LABSEQ4	244	4	0	0	11.7152	0
LABSEQ5	232	3	1	0	5.3574	0.05769
LABSEQ6	166	4	0	0	0	0
LABSEQ7	508	2	0	0	0	0.1112

Table 2. Tracking results of LABSEQ and CAVIAR video sequences.

ple that appeared 16 frames sooner. So there are 16 false positive tracking frames for that person, because the tracking algorithm cannot tell at which moment this individual object actually entered the scene as it is first detected in the group. This resulted to track incompleteness error. The person with a white shirt had a severe fragmentation because of the similarity between floor color and his shirt. One of the fragmented parts of the person is detected as a false positive independent object, because it was not distinguish from splitting before disappearing from the scene. So in the resulting trajectory, there is one false positive object trajectory f_p. Similar remarks apply to the other videos to explain the results.

Our fragmentation handling method uses temporal and spatial blob's appearance to distinguish between fragmentation and splitting, so it has the limitation that it needs the blob's appearance information for at least four frames after possible fragmentation. In low frame rate videos when a large number of people are existing in the scene, there is a possibility that our fragmentation method fails to detect fragmentation because of several fragmentation, splitting, and merging in the interval less than four frames. Also, if two tracking targets split and then walk toward the camera with approximately the same speed, they can be confused as fragmented regions related to one object.

8 Conclusion and future works

In this paper, a new MHT approach for online applications is presented. Our algorithm is robust to fragmentation caused by misdetection errors of background subtraction and splitting caused by inter-object occlusions. To generate hypotheses, we used adaptive color histograms of blobs which give global color information of blobs in several frames and result in more reliable data association.

Our results show the robustness of our algorithm for multiple object tracking. Our system can handle occlusion and fragmentation well when the targets are first detected individually. If it is not the case, our system do not know the exact moment of appearance of an object. We could improve our algorithm by trying to segment past group blobs after splitting based on the appearance of split targets. This would allow us to establish more precisely the position of the targets and the composition of the groups.

9 Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] C. Alex Yong Sang, H. Weimin, and L. Liyuan. Multiple objects tracking with multiple hypotheses graph representation. In *International Conference on Pattern Recognition*, pages 638–641, 2006. 1

[2] H. Bose, X. Wang, and E. Grimson. Detecting and tracking multiple interacting objects without class-specific models. In *Technical Report: Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, http://hdl.handle.net/1721.1/32536*, 2006. 1

[3] H.-T. Chen, H.-H. Lin, and T.-L. Liu. Multi-object tracking using dynamical graph matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:210–217, 2001. 1

[4] Y. S. Chia and W. Huang. Multiple objects trackingwith multiple hypotheses dynamic updating. *IEEE International Conference on, Image Processing*, pages 569–572, 2006. 1

[5] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:319–325, 1999. 1

[6] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting objects, shadows and ghosts in video streams by exploiting color and motion information. In *International Conference on Image Analysis and Processing*, page 360, 2001. 2

[7] S.-W. Joo and R. Chellappa. A multiple-hypothesis approach for multiobject visual tracking. *IEEE Transactions on Image Processing*, 16(11):2849–2854, 2007. 2, 7

[8] H. Mei, X. Wei, T. Hai, and G. Yihong. An algorithm for multiple object trajectory tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–864–I–871 Vol.1, 2004. 1

[9] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 1, 2, 7

[10] M. Shah, O. Javed, and K. Shafique. Automated visual surveillance in realistic scenarios. *Multimedia, IEEE*, 14:30–39, 2007. 2

[11] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 2000. 5