Occlusion Detection Using Motion-Position Analysis

Priyanka Mekala,  
Rodrigo J. Salmeron, Jeffrey Fan  
Dept. of Electrical and Computer Eng.,  
Florida International University,  
Miami, Florida 33174, USA

Asad Davari  
Dept. of Electrical and Computer Eng.,  
West Virginia University Tech,  
Montgomery, West Virginia 25136

Jichang Tan  
Dept. of Computer Science and Information Eng.,  
St. Johns University,  
Taipei, Taiwan

Abstract — Occlusion is a complex problem which can cause the loss of the target in the tracking process. One of the solutions for the occlusion problem is to detect the occlusion before it occurs or partially occurs and to track the moving objects using an additional camera separated by a known distance and known camera parameters, which can give a better view. Thus, the loss of a target can be prevented. This paper uses algorithms to perform target tracking under occlusion using motion-position analysis by calculating the shortest distances between the targets using a safe distance factor.

I. INTRODUCTION

VIDEO motion tracking is the process of locating either a single object or multiple moving objects in time using a camera sensor [1]. Finding the location of the moving targets within the video frame can be realized by an algorithm analysis [2].

Occlusion is a complex problem, which can cause the loss of the target in the tracking process. Many efficient methods have been found to solve the occlusion problem [3-6]. Optimizing multiple object tracking and best-view video synthesis is essential for many multimedia applications, such as surveillance, smart rooms, sports analysis and video presentation enhancement. In this paper, we propose an efficient multi-object tracking method based on multiple shortest path searching and a recursive dynamic programming approach for best-view selection.

The contribution of this paper can be summarized as the following: (1) occlusion detection for multiple moving objects using a novel threshold approach, (2) introduction of safe distance factor in order to detect occlusion before a certain time interval needed for correspondence of cameras, and (3) study of linear and non linear motion and influence of the algorithm on both.

This paper is organized as follows: Section II presents tracking implementation and occlusion consideration. Section III will present linear and non-linear motion of objects under consideration. In Section IV, a novel threshold approach for occlusion detection will be presented. Experimental results are shown in Section V. Finally, Section VI concludes the paper.

II. MODELING OF MOTION

Before applying the proposed algorithm to the multimedia input data, the signal requires the following preliminary steps:

- Segmentation of non-background objects from still background
- Detection of target/object
- Tracking of the position, motion, and velocity of the target

A. H. 264 Tracking Implementation

The introduction of H.264 encoding chips provides a far better hardware solution by developing motion estimation blocks to detect Macro-Block movement up to 10 reference frames in streaming imaging data. By adding a vector bank with the proposed framework, a high-resolution H.264-based vision sensor could be easily developed and implemented for the purpose of target tracking in a real-time manner for aviation systems. [9]. Thus all the positions, velocities, and size attributes are known using the above scheme of H.264 with the included vector bank.

B. Occlusion considerations

As seen in [10-12], tracking is a challenging task when there are complex interactions between targets. It is important to be able to track multiple objects simultaneously to obtain favorable results. We categorize object interactions into two classes. The first class of interactions constrain an object’s relative location, i.e., objects tend to keep relative positions or spatial lay out during a short period of time. The second type of interaction is object mutual occlusion. Figure 1 displays these two cases.

Tracking with multiple cameras not only increases the monitored area, but also helps to disambiguate in matching when subjects are occluded from a certain viewing angle [11]. Figure 2 displays such a case where Camera 2 distinguished Object 2 and Camera 3 distinguished Object 1.
Fig. 1. Four objects in motion taken from one camera at different focal lengths. (a) First class of object interaction model, without occlusion. (b) Second class of object interaction model, with partial occlusion [13].

The attributes POSITION, VELOCITY and SIZE describe the STATE of the object.

In Figure 2, Object 1 and Object 2 are partially occluded and the view from Camera 1 cannot capture the actual information of Object 1 after some time as it may be totally occluded by Object 2. In such case, an additional camera is needed to change the view perspective into a different angle. Since the objects are all in 3D non-linear motion, as described on a 2D plane, it may not be effective to change the angle of a single camera in order to view all the multiple objects at the same time without occlusion.

The block diagram for the proposed algorithm to activate another camera is as shown:

Fig. 2. Target tracking using multiple cameras

III. MOTION OF OBJECTS

We consider only the center of gravity (CoG) of the objects and hence the multiple objects are all represented as point sources. The points are moving independently of each other. In Figures 5, there are 3 objects (shown as point sources) that are moving independently. The motion vectors of the objects found using H.264 is used. Hence, the motion vectors would allow a prediction to be made that the state is continued until a few frames have passed. Thus, the speed and direction of each object is considered known.

In Figure 6, the frames 1-4 show three moving objects O1, O2 and O3 at four different time intervals. It can be observed that in frame 3, occlusion has occurred and it may seem that only two objects are present at that particular time interval. Using our algorithm it can be detected at Frame 1 or 2 that occlusion is to occur at next frame and, subsequently, another camera is activated. The views from an additional camera are seen in Figure 6.

The algorithm for the decision of occurrence of Occlusion:

![Algorithm Diagram]

Fig. 3. Target tracking algorithm using 2 cameras

![Tracking Diagram 1]

Fig. 4. Occlusion occurrence detection algorithm

![Tracking Diagram 2]

Fig. 5. Tracking of four objects from Camera 1

![Tracking Diagram 3]

Fig. 6. Tracking of four objects from Camera 2
The motion of objects is shown in linear and non-linear case as follows:

A. Two-Dimensional Linear Motion:
Suppose that there exists a linear relationship for velocity:

\[ V = U + a\Delta t \]  \hspace{1cm} (2)

For a time interval,

\[ \Delta t = (t - t_i) \]  \hspace{1cm} (3)

where, \( V \) is velocity, \( U \) is an initial velocity, \( a \) is acceleration, \( t_i \) is initial time, and \( t \) is a point in time after. Let \( P_{ik}(r_{ik}, V_{ik}, R_{ik}) \) represent the object parameters, where \( r_{ik} \) the position of the object in 2-D space is, \( V_{ik} \) object’s velocity, and \( R_{ik} \) is the object’s attributes, for all \( i \) objects in all \( k \) time intervals.

Let

\[ r_{ik} = (x_{ik}, y_{ik}) \]  \hspace{1cm} (4)

be the position vector for a moving object in the two dimensional case. Therefore,

\[ d_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \]  \hspace{1cm} (5)

B. Two-Dimensional Nonlinear Motion: Equation for governing general motion using Kinematics System Model

Two cases can be considered for the kinematics model of the system depending on the dimension of the environment. In the first case it is considered a number of vehicles or systems are moving in a 2D environment is considered. Systems are modeled as a point mass in 2D, or planar motion.

In the second case, the system is modeled as a point mass in a 3D environment. The discrete matrix form for the equation governing the motion of both systems is:

\[
\begin{bmatrix}
X_{k+1} \\
V_{k+1}
\end{bmatrix} = \begin{bmatrix}
A & B
\end{bmatrix}
\begin{bmatrix}
X_k \\
V_k
\end{bmatrix} + B_{a_k}
\]  \hspace{1cm} (6)

where,

\[ X = [x \ y]^T, \ V = [v_x \ v_y]^T, \ a = [v_x \ v_y] \]

\[ A = \begin{bmatrix}
I_2 & \Delta t I_2 \\
O_2 & I_2
\end{bmatrix}, \text{ and } B = \begin{bmatrix}
O_2 \\
\Delta t I_2
\end{bmatrix} \]

in a 2-D environment. The subscript \( k \) represents the discrete time step, \( I_2 \) represents an identity matrix of size 2×2, \( T \) represents the transpose of a matrix, and \( O_2 \) is a zero matrix of size 2×2. Vectors \( x, v, \) and \( a \), respectively, represent the position, velocity, and acceleration input in the inertial frame. [14] In this paper the 2D motion equations are considered for the occlusion detection.

IV. CALCULATION OF THRESHOLD

The object can be of any rough shape, and one must therefore be able to take into consideration about the size in limiting the threshold value. Any object could be completely bounded inside a sphere (for 3D) or a circle (for 2D). Let the objects be bounded in sphere of radius ‘R’ and for objects of different sizes considered here as \( R_1, R_2, R_3, \ldots, R_n \) correspond to the radius of the bounding sphere.

\[ \text{Threshold} = 2 \times \text{Min} \{R_1, R_2, R_3, \ldots, R_n\} + \eta \]  \hspace{1cm} (6)

\[ 2 \times \text{Max} \{R_1, R_2, R_3, \ldots, R_n\} + \eta \]  \hspace{1cm} (7)

\[ 2 \times \text{Avg} \{R_1, R_2, R_3, \ldots, R_n\} + \eta \]  \hspace{1cm} (8)

\[ d_{ik} \geq R_i + R_k + \eta \]  \hspace{1cm} (9)

\( \eta \) is known as the safe distance factor which is a function of velocities of the moving objects and time correspondence factor. The time correspondence factor is the time needed for the second camera to correlate and identify the objects in the frame with the first camera. Also, it is required to consider the focal lengths of the camera in order to maintain the spatial coordinate ratio with the frame ratio.

\[ \eta = f(V_{ij}, \tau) * \rho \]  \hspace{1cm} (10)

where \( V_{ij} \) is the velocity of \( i \)th object at \( j \)th time interval, \( \tau \) is the time correspondence factor needed for the second camera to correspond with first camera, \( \rho \) is the ratio of spatial coordinates ratio to frame ratio.

Once the occlusion is detected, the camera 2 is ‘ON’. Now information is forthcoming from both cameras. In order for camera 2 to track the object, correlation algorithm should be executed on the images from camera 1 and camera 2 [15].
\[ C_r(x, y) = \frac{\sum_{x=0}^{n-1} \sum_{y=0}^{n-1} C'(x', y') C(x + x', y + y')}{\sqrt{\sum_{x=0}^{n-1} \sum_{y=0}^{n-1} C'(x', y')^2 \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} C(x + x', y + y')^2}} \]  

(11)

\( C'(x', y') \) is the pixel value from Camera 1 to be correlated to Camera 2. \( C(x, y) \) is the pixel value from Camera 2 of the searching area at \((x, y)\). The correlation algorithm describes the similarity between the frames and identifies the objects corresponding to Camera 1.

V. EXPERIMENTAL RESULTS

A. Result Analysis: Linear Motion

As shown in the Figure 9, three moving objects with linear motion are considered due to the space continuity relation. It is assumed that the objects are moving with similar state attributes and continued velocity vectors for a few set of frames following the state of continuity. Figure 9-(a) shows the motion of three objects moving linearly. Figure 9-(b) is the plot of the calculated shortest distances between every combination of two objects present in the frame considered. Figure 9-(c) shows the various threshold lines calculated, based on the different considerations to obtain threshold values. Four different threshold values are plotted using the formulae (6), (7), (8), and (9) in this paper. Using the minimum value of size attributes in (6) is unsatisfactory due to its lack of detection of the occurrence of occlusion in either advance or the later. The cut-off line is the last line in the plot, and there are no values lying below the threshold indicating an absence of the occurrence of occlusion. Using the maximum of size attributes in (7) is not a sufficient option but, definitely a better option than (6) since it detects the occurrence of occlusion for two points as shown in figure 9-c. The first of the cut-off lines is the line considered, and there are two points below this threshold, implying the occurrence of two points of occlusion. The threshold using (9) is the best option to detect the occlusion but involves more evaluation time and more mathematical complications. Thus using the mean of the size attributes of objects added with the safe distance factor in (8) would yield the best results to detect and prevent occlusion.

It can be observed that from the plot there is the possibility of one occlusion event to take place during the whole consideration of the set of frames. Therefore, at this point another camera is activated, and using correlation, the correspondence between the objects is identified and used to track them from a different view and angle. Again, the algorithm is run in order to detect another occurrence of occlusion.

B. Nonlinear Motion

Figure (10) shows the nonlinear motion of three different objects moving in space. In Figure 10-(b) the shortest distances between all combinations of two objects are considered and plotted. Again, different threshold values are considered and plotted in Figure 10-(c) and (d). It can be observed that in non linear motion there is more probability for occurrence of occlusion within a short interval of time periods compared to linear motion. The threshold depends on the velocities and size attributes of the objects considered, and vary accordingly. Figure 10-(c) indicates that all the threshold calculations lead to cut-off lines which are almost near or coinciding. In the case where the attributes of all objects are very similar, the probability of occurrence of occlusion is very high and using two cameras to track the objects is a better solution. In Figure 10-(d), the thresholds corresponding to objects with different size and velocity attributes are plotted and can be seen that the number of occurrences of occlusion are less that that compared to 10-(c). Hence, between the time frames 4-6, the second camera has to be activated to get the better track of the objects.

![Fig. 9. Three objects of motion with linear motion. (a) Plot the coordinates of objects against time. (b) Plot with the distance between two objects. (c) Plot with distances and different values of threshold found based on the size of the objects.](image-url)
VI. CONCLUSION

Two models of moving objects were studied. This included linear and non-linear models assuming the space continuity relation. The algorithm for detection of occurrence of objects in the 2D model was proposed. Results supporting the algorithm proposed in this paper were presented and the efficiency of the algorithm was also seen in case of the multiple objects model using multiple cameras. Future work will include more complex considerations for the occlusion problem in 3D space.

ACKNOWLEDGMENT

This work was supported by the National Aeronautics and Space Administration through the University of Central Florida’s Florida Space Grant Consortium.

REFERENCES