Object Tracking Using Adaptive Block Matching
Karthik Hariharakrishnan*, Dan Schonfeld*, Philippe Raffy**, Fathy Yassa**
*Multimedia Communications Lab, UIC, IL60607, **NeoMagic Corporation, CA95054
e-mail: {kharihar,ds}@ece.uic.edu, {praffy,fyassa}@neomagic.com

ABSTRACT
We propose an object-tracking algorithm that predicts the object contour using motion vector information. Tracking is achieved by predicting the object boundary using motion vectors, followed by contour update, using occlusions/disocclusion detection. An adaptive block-based approach has been used for estimating motion between frames. An efficient modulation scheme is used to control the gap between frames used for object tracking. The algorithm for detecting occlusions proceeds in two steps. First, covered regions are estimated from the displaced frame difference. These covered regions are classified into actual occlusions and false alarms using motion characteristics. Disocclusion detection is also performed in a similar manner. The immediate applications of the proposed tracking algorithm are video compression using MPEG-4 and content retrieval based on standards like H.26L. Preliminary simulation results demonstrate the performance of the proposed algorithm.

Keywords
Object tracking, Video compression, Content retrieval, Motion estimation, Occlusion detection, K-Means clustering.

1. INTRODUCTION
Visual tracking has been an area of intensive research in the field of computer vision. With the advent of emerging multimedia standards like MPEG-4 it has become essential to develop a system that performs visual tracking in a computationally efficient manner. Applications of object tracking range from video compression, video retrieval, interactive video and scene composition etc. A variety of techniques have been employed for segmenting a semantically meaningful object out of a video scene. The most common approaches that have been proposed fall into the following categories: Region-based tracking, active contours (a.k.a snakes) and mesh-based tracking.

In region-based approaches, the user initially defines the video object. Video sequences are then segmented using a segmentation tool like the watershed transformation. Correspondence between the segmented regions in consecutive frames is established and this enables tracking of the video object in subsequent frames [Salembier]. The proposed method is similar to region-based methods and hence a comparison is given. Contour-based approaches usually do not make use of the spatial and motion information of the entire object and rely only on the information closer to the boundary of the video object. Snakes [Terzopoulos] was proposed as a method for tracking the boundary of the video objects using a parametric planar curve (active contours). Mesh-based approaches [Altunbasak] define an initial set of nodes on the boundary and interior of the object using gradient and motion information. These set of nodes are tracked by sampling the node motion vectors from the estimated optical flow, to generate the object mask in the next frame. Motion artifacts commonly present in block-based matching are overcome by using a content-based mesh because patch boundaries typically correspond to motion boundaries and hence motion estimation can be done more accurately.

In this paper, we propose a new method that employs block-based tracking for generating object masks in a video sequence. Adaptive-blocks are used in the motion estimation and object-mask generation phase. Adaptive block sizes are equivalent to mesh-based approaches and hence motion estimation is accurate. Blocking artifacts commonly present in “fixed size” block approaches are reduced. A modulation scheme that calculates the frequency of object tracking step is proposed. Occlusions are handled by combining motion as well as spatial criteria. Disocclusions are also detected and merged with the object automatically. Tracking has been performed on coherent scenes and an algorithm such as [Schonfeld] can be employed to detect scene changes.

Section 2 gives information on the proposed approach. Section 3 discusses the algorithm used for identifying occlusion/disocclusions in video sequences. Section 4 includes experimental results to demonstrate the proposed method.

2. TRACKING ALGORITHM
2.1 General Approach
The overall algorithm has been outlined in Fig 1. Motion estimation is performed between successive frames using a block-based approach. Taking advantage of the fact that motion between consecutive frames is usually very small, the estimation has been performed once in Nv (default value = 3) frames. A modulation scheme has been provided that varies Nv depending on the motion between frames. The modulation scheme enforces motion estimation between consecutive frames (\(N_v=1\)), if substantial motion has been observed. Occlusions are detected using a criterion that combines both motion and color information. The duality between occlusion and disocclusion is exploited to develop an efficient disocclusion algorithm. New regions that appear are detected using the displaced frame difference and they are automatically merged with the object if they exhibit similar motion characteristics.

2.2 Object-Mask Initialization
The initial object mask is given as an input to the tracking algorithm. The object contour can be initialized using a computer-assisted method such as the efficient and accurate live-wire approach [Mortensen]. In this paper, the segmentation method proposed by [Gatica] was used as it has better boundary localization properties than traditional watershed segmentation techniques.
2.3 Motion Estimation

Let \( I(x,y,k) \) denote the \( k \)th frame of a video sequence with \((x,y)\) denoting \( x \) and \( y \) co-ordinates of a pixel. Motion estimation starts with an initial block size of 16*16. Every reference block from \( I(x,y,k+N_0) \) is matched with the corresponding block in \( I(x,y,k) \). If the matched block is completely within the object or background, the corresponding block in \( I(x,y,k+N_0) \) is labeled as a seed block (Fig 2). If the matched block lies in the boundary, the corresponding block in \( I(x,y,k+N_0) \) is labeled as an uncertain block.

2.4 Modulation Scheme

In this algorithm, tracking is performed once every three \( N_0 \) frames and this leads to a speed improvement. A modulation scheme is provided that calculates the amount of object motion between frames separated by \( N_0 \) and reduces \( N_0 \) if the estimated motion is relatively high. This enables the tracking component to be applied infrequently if the motion in the video sequence is low.

2.5 Object Mask Generation

The partition of the previous frame that corresponds to the object is denoted by \( P(k) \). \( P(k) \) is a binary image representing the object in frame \( k \). The goal of this component is to generate the current object mask \( P(k+N_0) \), given the motion vectors. Every block in the current frame \((k+N_0)\) is motion compensated to find out the portion of the block that lies within \( P(k) \). This gives us the object support for the current frame \( P(k+N_0) \). This object mask is refined using occlusion/disocclusion detection.

3. OCCLUSIONS AND DISOCCLUSIONS

Occlusion and disocclusion detection is performed in two stages. The first stage corresponds to the detection of covered and uncovered region masks. In the second phase, the covered and uncovered regions are classified into actual occlusions and disocclusions using motion data. Ideally, the covered regions should correspond to occlusions and the new regions to disocclusions. In many instances, object regions that are not actually occluded are declared as covered regions. The above phenomena can be observed if there is a highly textured region or when there are illumination changes. It is clear that a further classification is required to find actual occlusions (disocclusions) from the detected covered (uncovered) regions. This can be achieved by using a motion-based criterion to classify the covered/uncovered regions detected using the displaced frame difference.

3.1 Occlusion Detection

3.1.1 Covered Regions

To enforce the occlusion detection step, regions that will be covered in future frames need to be estimated. The object contour has already been predicted as \( P(k+N_0) \). Due to partial occlusions, some part of the object visible in the current frame \((k+N_0)\) might disappear in the next frame \((k+2N_0)\). To estimate these regions, the current frame \( P(k+N_0) \) is motion compensated using frame \( k+2N_0 \). The motion compensated frame is subtracted from the original frame and all differences greater than a threshold are labeled as covered regions. A binary mask is formed that represents the covered regions. Post-processing operations are applied on the mask to remove noise and small clusters. Connected regions in the mask are labeled. The covered regions are represented by \( P_{cov} \). The goal of the next section is to characterize the motion of the object using appropriate number of clusters. The motion of covered regions can now be compared.
with the motion clusters to classify the covered regions as actual occlusions and false alarms. The false alarms correspond to the regions in $P_{cov}$ that are not actually occluded in the future frames.

### 3.1.2 Motion Clustering

For the computation of covered regions, forward motion between frames $(k+N_0)$ and $(k+2N_0)$ was estimated in the previous step. Using these forward motion vectors, $P(k+N_0)$ and $P_{cov}$, the object motion has to be characterized using a few clusters. Removing the covered regions $P_{cov}$ from $P(k+N_0)$ forms a new mask $P_{uncov}(k+N_0)$. The motion vectors in $P_{uncov}(k+N_0)$ are clustered using an adaptive K-means algorithm. The number of clusters is adaptively estimated. The algorithm starts by fitting a single cluster to the motion vectors in $P_{uncov}(k+2N_0)$. Initial cluster centers are chosen by randomly selecting motion vector samples from $P_{uncov}(k+N_0)$. After K-Means clustering has converged, the final distance between the samples and cluster centroids is calculated. If a large fraction (β) of the samples have a distance value greater than a threshold (α), the number of clusters is increased by one and the clustering algorithm is repeated. The number of clusters is increased until the fraction β becomes less than a value (typically 10%).

### 3.1.3 Dissimilarity Test

Regions that are occluded in frame $(k+N_0)$ are not present in $(k+2N_0)$. Hence, occluded regions exhibit motion characteristics similar to the object until frame $(k+N_0)$ and totally different motion characteristics from frame $(k+N_0)$ to $(k+2N_0)$. On the other hand, regions in $P_{uncov}$ that do not disappear have motion characteristics similar to the object until $(k+2N_0)$. This concept is used to classify the regions as occlusions and false alarms. The motion of each region in $P_{uncov}$ is found by fitting a translational model to the forward motion vectors.

Let $M_{ij}$ the $j^{th}$ covered region's forward motion vector,

$$M_{ij} = \text{Centroid of the } i^{th} \text{ motion cluster of the object}$$

for every $i$, calculate,

$$d_{ij} = ||M_{ij} - M_{ji}||$$

Only if $d_{ij} > \varphi_{occ}$ (where $d_{ij,\min} = \min(d_{ij})$) the $j^{th}$ region is treated as an occlusion and removed from $P(k+N_0)$. The covered regions that do not satisfy the above condition are not occluded in the future frame and hence are not removed from the object mask.

### 3.2 Disocclusion Detection

The algorithm employed for detecting occlusions is used to come up with a disocclusion detection algorithm. Thus we view occlusion and disocclusion as dual problems. To detect disocclusions, the new regions (uncovered) appearing in the current frame have to be estimated. To estimate these regions, the current frame $(k+N_0)$ is motion compensated using frame $k$. The motion compensated frame is subtracted from the actual frame and post-processed to give a binary mask containing uncovered regions $P_{uncov}$. Regions that are disoccluded in frame $(k+N_0)$ are present in $(k+2N_0)$ but not in $k$. Hence, disoccluded regions exhibit motion characteristics similar to the object from frame $(k+N_0)$ to $(k+2N_0)$. Thus we have a motion similarity test for the uncovered regions. Regions in $P_{uncov}$ that exhibit similar motion are included in $P(k+N_0)$. The details of the similarity test are omitted for clarity purposes.

### 4. SIMULATIONS

#### 4.1 Tracking Results

The tracking algorithm has been tested on some common MPEG test sequences and real video sequences (Fig 3 & 4). The computation time required for tracking of one frame has been compared with two region-based tracking approaches. The software written for all the algorithms have not been optimized. In the foreman sequence, object motion is not uniform and hence tracking is slowed down when significant motion has been observed. In the bream sequence also, there is significant motion between frames 110 and 122. The effect of the modulation scheme is shown in the next section.

#### 4.2 Modulation Scheme

The modulation scheme described in section 2.4 is used to skip frames when the motion of the object is relatively less. Figure 5 shows the number of frames skipped (vs.) frame number for the foreman and bream sequence. In the bream sequence, the motion of the object is very less until the 110th frame. The modulation scheme detects high motion and slows down motion estimation. Slowing down the tracking process enables accurate tracking for a longer period. The foreman sequence has high motion at various instances as shown in figure 5.

#### 4.3 Occlusion and Disocclusion Detection

Fig 6 shows the effect of the disocclusion detection for the bream sequence. The disocclusion is detected and associated with the object. The movement of the fish cannot be represented by one motion cluster (fish undergoes a swirling motion and requires two motion clusters to represent the overall object.
motion). The adaptive clustering algorithm selects K = 2 in this case. Using K = 1 would not be able to catch the disocclusion. The algorithm has also been applied to real-life videos. Figure 7 illustrates the performance of the occlusion detection algorithm. Another person obstructs the tracked object partially and then moves away. The occlusion detection algorithm modifies the mask accordingly.

Fig 6. Disocclusion (tail of the fish) detected and associated with object.

Fig 7. Occlusion detection for the Shirley sequence.

4.4 Algorithm Comparisons
Figures 8, 9 & 10 show the performance comparison of the proposed algorithm with two region-based methods for object extraction. All the algorithms have been implemented in MATLAB. The computation times observed are 15 sec (our approach), 200 sec [Gatica], 160 sec [Salembier].

Fig 8. Tracking the person with proposed approach

Fig 9. Tracking the person with partition lattice operators

Fig 10. Tracking the person with partition projection

5. REFERENCES


