STATIONARY BACKGROUND GENERATION IN MPEG COMPRESSED VIDEO SEQUENCES

Ramazan Savas Aygun and Aidong Zhang

Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260 USA

ABSTRACT

The development of the new video coding standard, MPEG-4, has triggered many video segmentation algorithms that address the generation of the video object planes (VOPs). The background of a video scene is one kind of VOPs where all other video objects are layered on. In this paper, we propose a method for the generation of the stationary background in a MPEG compressed video sequence. If the objects move frequently and all the components of the background are visible in the video sequence, the background macroblocks can be constructed by using Discrete Cosine Transform (DCT) DC coefficients of the blocks. After the generation of the stationary background, the moving objects can be extracted by taking the difference between the frames and the background.

1. INTRODUCTION

The MPEG-4 [5] video standard introduces the notion of Video Object Plane (VOP) to support content-based access of video streams. Each video object is represented as a video object plane (VOP). The segmentation of video objects is one of the core parts of MPEG-4 video encoding. The background of the objects is represented with VOP_0. All other VOPs of objects are layered on the background VOP_0. The VOPs are often not known beforehand and they need to be extracted from image sequences.

Most of the video segmentation algorithms that address VOP extraction have emerged due to the new video coding standard, MPEG-4. Most methods that are proposed investigate the segmentation of video objects and are computation intensive. Kim et al. proposed a method where VOPs are extracted by first generating moving edge maps [2]. The method proposed in [1] consists of a motion detection phase employing higher order statistics and a regularization phase to achieve spatial continuity. VOPs are generated from an estimated change detection mask (CDM) in [3]. A buffer is used to increase temporal stability by labeling each pixel as changed if it belonged to an object at least once in the last L change detection masks. The model in [4] uses the edge pixels in an edge image detection. An object tracker matches this binary model against subsequent frames. The model is updated every frame to accommodate for rotation and changes in shape of the object. After video objects are extracted, the remaining part of the image is considered as the background, VOP_0.

Our focus is different from the previous work done in this field. We concentrate on the generation of the stationary background, which will provide flexibility in MPEG-4 video construction and editing. In this paper, we propose an algorithm which generates the background from a video sequence and then detects the moving objects on it. We consider the background as the part of the image without the moving objects which are replaced with the real background. So, this is not just the separation of the background from moving objects. We also generate the background hidden behind the moving objects from the video sequence. The background can be constructed if all the components appear in the video sequence. Because of the moving objects, not all parts will appear in the same frame. We perform our operations on the macroblock level using only DCT DC coefficients in an MPEG-1 video stream. This approach fits well to applications where the background does not change often such as lectures or halls recorded by a static camera. In these applications, the background remains the same and objects move enough so that all parts of the background are visible in the video sequence.

2. STATIONARY BACKGROUND GENERATION

2.1. MPEG Video Stream

An MPEG-1 video stream is composed of I, P and B frames, where P and B frames exploit the similarity between the frames. The detection of the background can be accomplished if the moving object displaces its location. So that the background that is hidden behind the object will be visible. In consecutive frames, there is usually very slight change. It is time consuming to process every frame. Therefore, the process of the background generation at intervals of frames will provide faster speed. P and B frames assume very little change with respect to their dependent frames and their macroblocks are decoded using macroblocks in

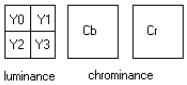

Figure 1: Blocks of a macroblock in MPEG frames.

the I frames. So, I frames are better candidates for the background construction since there will be enough object displacement in video sequence and they do not depend on any other frame. But, if the mobile objects are small (i.e. that can fit in a macroblock) and their displacement is also small, then further processing is needed. In that case, P and B frames need to be processed. In this work, we address large mobile objects (i.e. that cannot fit into single macroblock).

In MPEG-1, each macroblock of I-Frame is composed of 6 blocks: 4 luminance blocks and 2 color components (Figure 1). Blocks are compressed using Discrete Cosine Transform(DCT) and each block is represented with DC and AC coefficients. Our algorithm works at the level of macroblocks rather than pixels since the compression is performed at the level of macroblocks. Let $MB^{pq}(\alpha)$ be the macroblock at the p^{th} row and q^{th} column of frame α. Each DC coefficient of $MB^{pq}(\alpha)$ can be denoted as $DC^{pq}(\alpha)$. $MB^{pq}(\alpha)$ may be represented with its DC coefficients as follows:

$$\{Y_0^{pq}(\alpha), Y_1^{pq}(\alpha), Y_2^{pq}(\alpha), Y_3^{pq}(\alpha), Cb^{pq}(\alpha), Cr^{pq}(\alpha)\}$$

where Y_0, Y_1, Y_2, Y_3, Cb and Cr are DC coefficients of the blocks that are shown in Figure 1.

2.2. The Algorithm

The background generation has three steps:

- The generation of the background from a video shot
- Merging of the same backgrounds that are generated from different video shots which belong to the same video scene
- Merging of the same backgrounds that appear in different video scenes

In this paper, we will focus on the generation of the background from a video shot. Our algorithm has three phases: the clustering of the macroblocks, the selection of the cluster which may contain the background macroblock and the selection of the background macroblock from the cluster. The algorithm is thus as follows:

/* $ClusterList(p,q)$ keeps all clusters of macroblocks that appeared at p^{th} row and q^{th} column*/
/* $cluster(p,q)$ is the selected cluster (which will probably have the background macroblock) from the $ClusterList(p,q)$.*/

For each frame α of the video sequence
  For each macroblock $MB^{pq}(\alpha)$
    Cluster $MB^{pq}(\alpha)$ into $ClusterList(p,q)$
    Select the $cluster(p,q)$ from $ClusterList(p,q)$
    Select the background macroblock from $cluster(p,q)$
  Combine all macroblocks selected

2.2.1. Clustering

Two methodologies may be used for clustering: non-incremental clustering and incremental clustering. In a non-incremental clustering method, all the macroblocks must be stored until a shot change occurs. The macroblocks can be clustered using a non-incremental clustering method for a video shot. This method is good if the video shot length is short (less than 5 seconds). But if the shot length is long and if it is possible to generate the background macroblock earlier, there is no need to first process and then cluster all the macroblocks in a video shot. In this case, it is better to use incremental clustering.

Let the background macroblock at location (p,q) need to be generated. All macroblocks that showed up at this location are clustered. The feature vector for a macroblock is the DC coefficients of the blocks. In our case, we map macroblocks to a one-dimensional space and order them according to the distance from a specific point and then cluster them incrementally as macroblocks arrive. The macroblocks are clustered using Nearest Neighbor Rule (1-NNR). If the distance from the existing clusters is more than a specific threshold (τ), a new cluster is created for the macroblock. Most clustering algorithms will be satisfactory to cluster the macroblocks which have 6 elements in their feature vector. Since, the DC coefficients are already approximation to the macroblock, the distance function and how features are evaluated gain significance in clustering.

Distance Function. In our work, two types of distance measures are considered: additive and selective. Additive distance measures accumulate the difference at each feature (e.g. Manhattan, Euclidean). Selective distance measures depend on the selection of one of the difference of features (e.g. maximum, minimum). As an example for additive distance measure, Euclidean distance measure will be used. Both $maximum$ and $minimum$ distance measures will be considered for the selective distance measure.

The features may be evaluated in several ways. four methods will be stated here. First method assigns equal weights to each DC coefficient. Let $MB^{pq}(\alpha)$ and $MB^{pq}(\beta)$ be the macroblocks that are compared. The absolute difference between two DC coefficients of $MB^{pq}(\alpha)$ and $MB^{pq}(\beta)$ can be represented as $\triangle DC^{pq}(\alpha,\beta)$. Then the Euclidean distance between $MB^{pq}(\alpha)$ and $MB^{pq}(\beta)$ will be computed as

$$\sqrt{\sum_{i=0}^{3} \triangle Y_i^{pq}(\alpha,\beta)^2 + \triangle Cb^{pq}(\alpha,\beta)^2 + \triangle Cr^{pq}(\alpha,\beta)^2}$$

The second method assigns the same weight to the chrominance and the luminance and takes the average of the luminance coefficients. The average of the luminance DC coefficients of $MB^{pq}(\alpha)$ is denoted with $Y^{pq}_{avg}(\alpha)$ which is $\frac{1}{4}\sum_{i=0}^{3} Y_i^{pq}(\alpha)$. In this case, the distance will be computed as

$$\sqrt{(\triangle Y^{pq}_{avg}(\alpha,\beta)^2 + Cb^{pq}(\alpha,\beta)^2 + \triangle Cr^{pq}(\alpha,\beta)^2}$$

The DC coefficient is 8 times the average of the values in the block. So, a DC coefficient is the smoothing of the values in the block. It may be better to keep the differences as much as possible. Instead of averaging luminance values, the maximum luminance difference, $\triangle Y^{pq}_{max}(\alpha,\beta)$, which is $max_{i=0,3} Y_i^{pq}(\alpha,\beta)$, may be evaluated. Then the distance will be

$$\sqrt{\triangle Y^{pq}_{max}(\alpha,\beta)^2 + \triangle Cb^{pq}(\alpha,\beta)^2 + \triangle Cr^{pq}(\alpha,\beta)^2}$$

Sometimes, it may only be necessary to consider sharp changes in the sequence. Instead of computing the maximum of luminance difference, the minimum luminance difference can be taken, $\triangle Y^{pq}_{min}(\alpha,\beta)$, which is $min_{i=0,3}\triangle Y_i^{pq}(\alpha,\beta)$. Then the distance will be

$$\sqrt{\triangle Y^{pq}_{min}(\alpha,\beta)^2 + \triangle Cb^{pq}(\alpha,\beta)^2 + \triangle Cr^{pq}(\alpha,\beta)^2}.$$

Maximum selective measure is used if the difference between two macroblocks needs to be emphasized as much as possible. The following function is used for $maximum$ distance:

$$max(\triangle Y^{pq}_{max}(\alpha,\beta), \triangle Cb^{pq}(\alpha,\beta), \triangle Cr^{pq}(\alpha,\beta)).$$

The sharp changes at a macroblock can be detected using the $minimum$ distance measure as follows:

$$min(\triangle Y^{pq}_{min}(\alpha,\beta), \triangle Cb^{pq}(\alpha,\beta), \triangle Cr^{pq}(\alpha,\beta)).$$

2.2.2. The Cluster Selection

If the number of clusters are more than one for a macroblock location, there is a moving object at that macroblock. If the number of clusters is one, then there is no movement at that macroblock and the cluster is the only candidate which contains the background macroblock.

There are two basic factors that will be used for the selection of the cluster: *frequency* and *continuity*. The frequency of a cluster is the number of elements in that cluster. The continuity of a cluster denotes the maximum length of the sequence of macroblocks of the cluster that appeared sequentially. The clusters are first chosen according to their frequency. If there is a tie, the cluster which has a higher continuity is selected. Sometimes, there may be no or very little motion in succeeding I-Frames. All these frames are considered as the same frame and they have the effect of a single frame on the frequency and the continuity of the clusters.

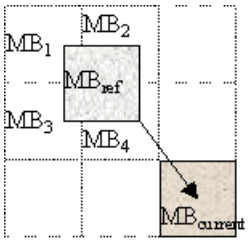

Figure 2: Estimation of DC coefficient of the reference macroblock.

2.2.3. The Background Macroblock Selection

The background are chosen from the cluster in four ways. Three of them are about luminance and the other one considers both luminance and chrominance. In the first case, if the moving object absorbs light (non-shining), it will cause darkness on the background. It is better to choose the candidate which has a higher luminance. Low luminance is due to the object in the environment. In the second case, if the moving object is a light source or reflects light, it will cause the background to shine. So, it is better to choose the candidate which has a lower luminance. High luminance is caused by the object in the environment. In the third case, if the type of the object is unknown or it may have the properties of both shining and non-shining object, then it may be better to be neutral. So, the background block which is closer to the mean is chosen. Finally, if the color is considered, then the background macroblock which is closer to the mean of all the block coefficients is the candidate.

2.2.4. Enhancement with Motion Vectors

The motion vector indicates whether the macroblock moves to other parts of the frame. Although the term "motion vectors" are used in MPEG streams, motion vectors do not exactly express the displacement of a macroblock. They rather give the location of the closest macroblock in the previous or next frame. This is crucial if there is a pattern in the background. Although the macroblock does not move, since its pattern is the same as some other macroblock, the motion vector may point to that macroblock. If the macroblock is previously located in another location, this usually shows the parts where moving objects exist. If there is a pattern in the frame, it is also possible that this macroblock will point to another location sharing the same pattern. We apply two trivial methods to detect this situation.

Since we do our operations on the macroblock level, we do not have the exact DCT coefficients of this block. One way to deal with this is to check all macroblocks that intersect with this macroblock. If all of them have the same characteristics as in the predicted frame, the macroblock has not moved. Another way is to estimate the DC coefficients of the referenced block. It is shown how to estimate the DC coefficients of this block depending on the macroblocks that

Figure 3: Video Sequence and Background Generation.

intersect with this block in [6]. The estimation is done by giving weights according to the macroblock coefficients by the area that share with the reference macroblock MB_{ref} (Figure 2):

$$DC_{ref} = \sum_{i=1}^{4}(w_i \times DC_i)$$

where DC_i gives the DC value of a block in Figure 2 and w_i is the ratio of region covered by MB_i to the region of the whole macroblock (8x8=64 pixels).

Let $MB^{pq}(\alpha)$ and $(MB^{pq}(\beta)$ represent the referenced macroblock in the referenced frame and in the current frame, respectively. If

$$Distance(MB^{pq}(\alpha), MB^{pq}(\beta)) < \tau),$$

where $Distance$ is a distance function and τ is a threshold, this implies that the reference block has not changed in the current frame. This also implies that $MB^{pq}(\alpha)$ was affected by the moving object. So, $MB_{current}(\beta)$ is a candidate for the background macroblock. Otherwise, it would be assumed to be part of a moving object.

2.3. Merging Backgrounds

The problem of merging backgrounds is in fact the problem of merging macroblocks. There are clusters generated for each macroblock location. Merging can be done from scratch as if no clusters existed. Then the upper procedure for generating clusters from a single video shot can be used. A cluster merging approach can also be used. The clusters which share the same characteristics are merged. The merging can be performed by using one of the existing methods in the literature. We merge them according to their closeness of their centroids.

3. RESULTS

We used video streams that are recorded in the lectures. Each lecture is stored as a MPEG-1 video stream. The coding pattern of streams are IBBPBBPBBPBBPBB and the frame rate is 15 frames per second. Figure 3 shows the phases of how the background is detected. The first row displays the frames that are encountered. The second row shows the phases of background generation. Other examples are omitted here due to the space limit.

Our observations showed that if the object moves enough, the background can be constructed in the early frames of the clip. In the given example, the background is generated after processing 13 I frames. 120 B and 48 P frames are skipped. Our results showed that chrominance must be included in the distance computation and the selection of the background macroblock from the cluster. The wrong macroblocks which have color distortions may be selected if only luminance coefficients are considered.

4. CONCLUSION AND FUTURE WORK

An algorithm is presented for the generation of the stationary background from a compressed video sequence. This algorithm is based on the clustering of macroblocks. Experiments showed that backgrounds can be extracted using DC coefficients of MPEG streams. In this paper, we did not consider camera operations. Our next step will be generation of the background in existence of camera motion.

5. REFERENCES

[1] A. Neri et al. Automatic moving object and background separation. *Signal Processing*, 66:119–232, 1998.

[2] C. Kim and J-N. Hwang. An integrated scheme for object-based video abstraction. In *ACM*, 2000.

[3] R. Mech and M. Wollborn. A noise robust method for segmentation of moving objects in video sequences. In *ICASSP97*, pages 2657–2660, April 1997.

[4] T. Meier and K.N. Ngan. Automatic video sequence segmentation using object tracking.

[5] T. Sikora. The mpeg-4 video standard verification model. *IEEE Trans. Circuits Syst. Video Technology*, 7:19–31, February 1997.

[6] B.-L. Yeo and S.-F. Chang. Rapid scene analysis on compressed video. *IEEE Trans. Circuits Syst. Video Technology*, 5(6).