TRACKING OF EXTENDED SIZE TARGETS IN H.264 COMPRESSED VIDEO USING THE PROBABILISTIC DATA ASSOCIATION FILTER

Vimal Thilak and Charles D. Creusere

Klipsch School of Electrical and Computer Engineering
New Mexico State University
Las Cruces NM 88001 USA
{vimal, ccreuser}@nmsu.edu

ABSTRACT

Object detection and tracking play a significant role in critical applications such as video monitoring and remote surveillance. These systems employ compression to efficiently utilize the available bandwidth. An example of an efficient compression solution to low bit rate video applications is the recently proposed H.264/AVC video coding standard. In particular, H.264/AVC has been optimized for transmission over wireless channels making it an attractive candidate for use in remote surveillance systems. In this paper, we propose an algorithm that exploits motion vectors generated by the H.264 encoder for object detection and tracking. Experimental results demonstrate the effectiveness of the proposed method to detect and track objects in real video sequences.

1. INTRODUCTION

The detection and tracking of objects from video sequences is an important task for several applications including remote video surveillance and data retrieval from multimedia databases [6]. Compression is an important and common component of any surveillance or multimedia database system due to the voluminous nature of video data. An example of an effective compression system is the latest H.264/AVC standard [7]. In systems that employ compression, a direct approach for object tracking is to first decompress the encoded video and then perform the task of tracking. However, decompression is computationally intensive making the above approach unattractive for applications of interest. A practical alterative to tracking is to use the information provided by the compressed bit stream. Thus, the problem of detecting and tracking an object from compressed video is of great interest to the video processing community.

A few authors have attempted to design algorithms for detecting and tracking objects from compressed video. Wang et. al [6] proposed an algorithm based on the Kalman filter that tracks faces from compressed MPEG-1 bit stream. Schonfeld and Lelescu [5] propose an algorithm that utilizes motion vectors to detect objects from multimedia data. In this paper, we propose an algorithm based on motion vector magnitudes to detect and track objects from H.264 bit stream. A novel feature of our detection algorithm is its ability to *optimally* differentiate between the object and the background by computing threshold values that minimize Bayes' risk [2]. Precision tracking is achieved by employing a probabilistic data association filter (PDAF) [1] which is a suboptimal Bayesian algorithm.

This work was supported in part by Sandia National Laboratories under grant SURP 25363

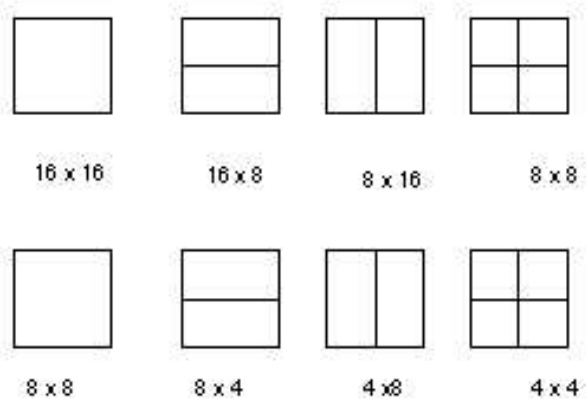

Figure 1: Macroblock types in H.264.

2. THE H.264 VIDEO CODING STANDARD

The H.264 standard [7] is the latest video coding standard proposed jointly by the ITU-T and ISO-IEC committees. The key features of the standard include a bit rate reduction of 50% given fixed fidelity compared to other standards and a provision for a network friendly syntax. H.264 is able to achieve high compression efficiency by exploiting the temporal and spatial redundancies in video by employing respectively, inter-frame predictive coding and intra-frame predictive coding.

In intra-frame coding, the spatial correlation between adjacent blocks of pixels, usually referred to as macroblocks (MB) are exploited by means of predictive coding. Further gains are obtained by encoding the residual frame with a hierarchial block transform [8]. The resulting frame is called an I frame [7].

In the inter-frame coding process, temporal redundancy between successive frames are exploited using motion estimation and motion compensation [7]. A video frame is referred to as a P frame if the motion vectors for the current frame are derived from previous frames, i.e, its causal neighbors. A frame is called B frame in case the motion vectors are derived from both its causal and non-causal neighbors.

H.264 provides a high degree of flexibility for choosing

the size of blocks to be used in the motion compensation process. Figure 1 illustrates the various block sizes that may be chosen by the encoder during motion estimation. The macroblock can be partitioned into various block sizes including 16×16, 8×16, 16×8 and 8×8 samples. In case the 8×8 block is chosen, the macroblock can be further partitioned into 8×4, 4×8 and 4×4 samples. Thus, up to sixteen motion vectors may be encoded for a 16×16 macroblock.

3. THE DETECTION ALGORITHM

The proposed algorithm consists of two modules namely the detection module and the tracking module. The detection algorithm employs motion vectors extracted from compressed H.264 bitstream to detect an object of interest. The results from the detection module are passed on to the tracking module to track the object of interest in a video sequence.

In this paper, we assume that the video sequence is encoded using the IPPP... format. This format is supported by the H.264/AVC Baseline profile [7] which targets low latency applications like remote video surveillance. However, our work can be extended to other formats such as IBBPBBP... by adopting the method proposed by Wang et. al [6].

3.1 The Detection Module

The object detection or the segmentation module employs motion vector magnitudes to detect an object of interest in a video sequence. The segmentation consists of three steps namely: (i) binary image formation (ii) optimal pixel classification and (iii) clustering. These steps are described in detail below.

3.2 Binary Image Formation

The purpose of this step is to transform a greyscale image (frame) into a binary image. The transformation is performed by means of the following threshold operation

$$\beta_i = \begin{cases} 1 & M_L^{'} \leq M_i \leq M_H^{'} \\ 0 & otherwise \end{cases} \quad (1)$$

where β_i is the pixel in location i, M_i is the motion vector magnitude and $M_L^{'}$ and $M_H^{'}$ are the initial choice of lower and upper threshold limits for the motion vector magnitude. $M_L^{'}$ and $M_H^{'}$ are based on prior information about the target and background motion vector magnitudes. All pixels whose motion vector magnitudes are within $(M_L^{'}, M_H^{'})$ are classified as target pixels while all the others are classified as background pixels.

3.3 Optimal Pixel Classification

The objective of this step is to improve the classification results of the previous steps by obtaining optimal threshold values (M_L, M_H). We derive the optimal threshold values by adopting the Bayesian approach as explained in the following. The pixels can be classified into two classes namely the target and the background since we are dealing with the case of detecting one target in a video sequence. The probability densities for the two classes of pixels $p(M/1)$ and $p(M/2)$ and the prior probabilities of the classes π_1 and π_2 where class 1 represents the target and class 2 represents the background are assumed available. If the pixels are classified according to the threshold operation given by (1), then the probability of misclassifying a class 2 (background) pixel as a target pixel is

$$P_{e1} = p(M_t/2) = \int_{M_L}^{M_H} p(M/2)dM$$

where M_t is the event that $I \in [M_L, M_H]$. Similarly, the probability of misclassifying a class 1 pixel as class 2 pixel is

$$P_{e2} = 1 - p(M_t/1) = 1 - \int_{M_L}^{M_H} p(M/1)dM$$

The *a priori* probabilities π_1 and π_2 can be calculated assuming that the size of the target in pixels is known. These probabilities are given by

$$\pi_1 = \frac{N_t p_t^{'}}{N_t p_t^{'} + (S - N_t) p_b^{'}}$$

$$\pi_2 = 1 - \pi_1$$

where $p_t^{'}$ and $p_b^{'}$ are respectively, the pixel detection probabilities in the target and background regions given by (3), S is the size of the video frame in pixels and N_t is the size of the target in pixels.

With the above definitions we can define the Bayesian risk function as follows

$$C = \pi_2 P_{e1} + \pi_1 P_{e2} \quad (2)$$

The Bayesian risk C will be minimized by differentiating (2) with respect to M_L and M_H and setting the result to zero yielding the following relationship

$$\pi_2 p(M_t/2) = \pi_1 p(M_t/1)$$

which has two solutions which are the desired optimal threshold values.

In order to simplify the analysis we assume that the probability densities $p(M/1)$ and $p(M/2)$ are Gaussian, i.e., $p(M/1) = N(\mu, \sigma^2)$ and $p(M/2) = N(\eta, \kappa^2)$. Defining the target layer thresholds as $M_L = \mu - \delta_1$ and $M_H = \eta + \delta_2$, the optimal δ_1 and δ_2 are obtained by solving the following equations

$$\frac{\pi_2}{\kappa} \exp(-\frac{(\mu - \delta_1 - -\eta)^2}{2\kappa^2}) = \frac{\pi_1}{\sigma} \exp(-\frac{-\delta_1^2}{2\sigma^2})$$

$$\frac{\pi_2}{\kappa} \exp(-\frac{(\mu - \delta_2 - -\eta)^2}{2\kappa^2}) = \frac{\pi_1}{\sigma} \exp(-\frac{-\delta_2^2}{2\sigma^2})$$

The optimal values M_L and M_H are easily obtained once we know δ_1 and δ_2.

The parameters for the densities $p(M/1)$ and $p(M/2)$, i.e, the sample mean and the sample variance are estimated from the data obtained during the initial classification step. The probability $p_t^{'}$ required for calculating the *a priori* probabilities is computed using the following

$$p_t^{'} = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{I_L^{'}}^{I_H^{'}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \quad (3)$$

The probability $p_b^{'}$ is calculated in a similar manner.

3.4 Clustering

The clustering method employed in the algorithm is a simple nearest neighbor algorithm [2]. In this method, a pixel belongs to the cluster if it is linked to at least one other pixel by a distance less than a certain distance. The centroids of the resulting clusters are computed and passed to the tracking module.

4. THE TRACKING MODULE

We assume that the motion of the target can be modelled by the following equations

$$x_{k+1} = Fx_k + \Gamma v_k \tag{4}$$

$$z_k = Hx_k + w_k \tag{5}$$

where x_k is the state of the system at time k, v_k is a zero mean, white, Gaussian noise sequence with covariance matrix $\Gamma Q_k \Gamma^{'}$, w_k is a sequence of zero mean, white, Gaussian noise with covariance matrix R_k, F and H are matrices that are assumed to be independent of time.

A classical method to track objects in video based on the model in (4) and (5) is the Kalman filter [6]. This is a linear estimation technique for tracking the state of the target such as position (usually the centroid) and velocity. However, the Kalman filter assumes that the detection algorithm classifies only one cluster as target. Hence this method cannot handle cases in which the target splits into many clusters or multiple clusters are detected (classified) as targets. This problem is also referred to as the data association problem [1] in target tracking literature. The probabilistic data association filter (PDAF) [1] is employed to overcome the above shortcoming of the Kalman filter. The PDAF handles the case of fragmented targets by probabilistic weighting of all the "validated neighbors" [3] of the target.

4.1 The PDAF

We adopt the notations in [1] to describe the PDAF algorithm. The symbol $'$ denotes the matrix or vector transposition operator.

Define the elliptical validation region $\tilde{V}_k(\gamma)$ by

$$\tilde{V}_k(\gamma) = \{z : \nu^{'}(k+1)S^{-1}\nu(k+1) \leq \gamma\} \tag{6}$$

where γ is a value obtained from tables of chi-square distribution, since the weighted norm of the innovation in (6) is described by a chi-square distribution and ν is the innovation given by

$$\nu(k) = z(k) - H\hat{x}(k|k-1)$$

Define the events
$\theta_i(k) = \{z_i(k)$ is the target oriented measurement$\}$ $i = 1,...,m_k$
$\theta_0(k) = \{$none of the measurements at time k is target oriented$\}$
with probabilities

$$\beta_i(k) = P\{\theta_i(k)|Z^k\}, i = 1,...,m_k$$

With the above definitions, the PDAF algorithm is presented below.

- State estimation

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\nu(k)$$

where $K(k)$ represents the Kalman gain and ν is given by

$$\nu(k) = \sum_{i=1}^{m_k} \beta_i(k)\nu_i(k)$$
$$\nu_i(k) = z_i(k) - H\hat{x}(k|k-1)$$

- Error covariance update

$$P(k|k) = \beta_0 P(k|k-1) + (1-\beta_0)P^c(k|k) + \tilde{P}(k)$$

where

$$\tilde{P}(k) = K(k)(\sum_{i=1}^{m_k} \beta_i(k)\nu_i(k)\nu_i^{'}(k) - \nu_i(k)\nu_i^{'}(k))K^{'}(k)$$
$$and\ P^c(k|k) = (1-K(k)H)P(k|k-1)$$

- One step prediction

$$\hat{x}(k+1|k) = F\hat{x}(k|k)$$

- Covariance of the predicted state

$$P(k+1|k) = FP(k|k)F^{'} + \Gamma Q(k)\Gamma^{'}$$

- Kalman gain

$$K(k+1) = P(k+1|k)H^{'}S^{-1}(k)$$

and

$$S(k) = HP(k+1|k)H^{'} + R(k)$$

The probabilities $\beta_i(k)$ are computed according to the following expressions.

$$\beta_i(k) = \frac{e_i}{b + \sum_{j=1}^{m_k} e_j}$$

$$\beta_0(k) = \frac{b}{b + \sum_{j=1}^{m_k} e_j}$$

$$e_j = \exp(-\frac{1}{2}\nu_i^{'}(k)S^{-1}(k)\nu_i(k))$$

$$b = \frac{(\frac{2\pi}{\gamma})^{\frac{n1}{2}} m_k C_{n1}(1 - P_D P_G)}{P_D}$$

where $n1$ is the dimension of the measurement z, C_{n1} is the volume of the $n1$-dimensional unit hypersphere ($C_2 = \pi$ for example), P_D is the probability that the true measurement is detected and P_G is the probability that the true measurement will fall in the validation region.

5. EXPERIMENTS

The proposed algorithm is evaluated using a 70 frame QCIF video sequence recorded at a rate of 18 frames/sec. The H.264 reference software JM 6.2 [4] is used to encode and decode the video sequences in our experiments. The parameters for the detection algorithm are $M_L^{'} = 4$ and $M_L^{'} = 32$.

Figure 2: Image illustrating the detection and tracking of the target.

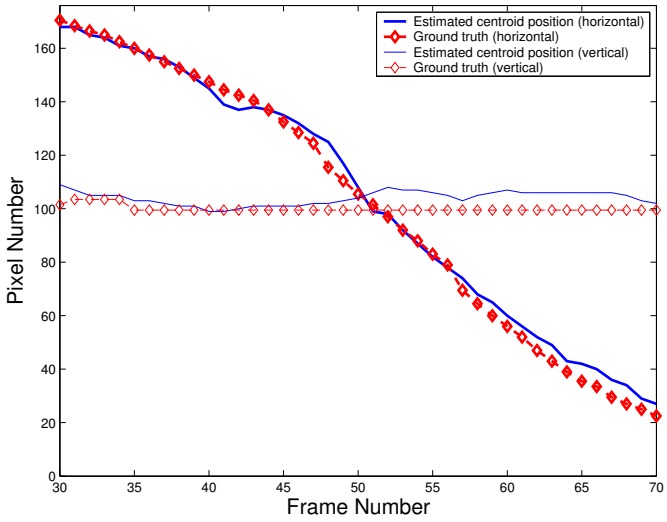

Figure 3: Tracking of centroid along the horizontal and vertical directions

The size of the target is 32 x 76 pixels. The proximity distance used in the nearest neighbor algorithm is three pixels.

A nearly constant velocity model is assumed for the object [1]. The matrices H, F and Γ required for implementing the PDAF algorithm are presented in [1]. We use the centroid and the velocity of the target in both x and y-direction to describe the state of the system. The observation vector consists of the centroid measurement provided by the detection algorithm. Figures 3 and 4 illustrates the results obtained using the example sequence. Figure 3 and compares the estimated centroid position with the ground truth. The ground truth is obtained by manually recording the centroid position from the video sequence. Figure 4 illustrates the estimated position and its $\pm\sigma$ accuracy. Finally, we include a frame of video in Figure 2 showing the tracked object to illustrate the effectiveness of the tracking algorithm.

6. CONCLUSION

This paper proposes a system for detecting and tracking an object from compressed H.264 bit stream. The detection algorithm is a simple threshold operation that classifies a pixel

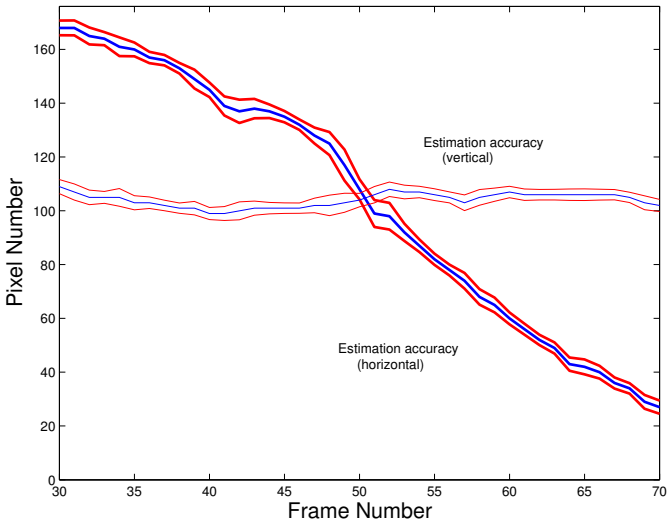

Figure 4: Estimated position and its $\pm\sigma$ accuracy

either as object or as background. An simple solution is proposed for determining the optimal threshold values. It is observed that the detection algorithm may cause the object to be fragmented in some video frames. The probabilistic data association filter [1] is employed in our algorithm to overcome this deficiency of the detection algorithm. Experimental results demonstrate the effectiveness of the proposed solution.

7. REFERENCES

[1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association* . Address: Academic Press, 1987.

[2] R. O. Duda and P. E. hart and D. G. Stork, *Pattern Classification* . Address: Wiley-Interscience, 2001

[3] A. Kumar and Y. Bar-Shalom and E. Oron, "Precision Tracking Based on Segmentation with Optimal Layering for Image Sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp 182–188, February 1995.

[4] K. Suehring: Editor, *H.264/AVC Software Coordination*. Address: http://bs.hhi.de/ suehring/tml/, 2003.

[5] D. Schonfeld and D. Lelescu, "Vortex: Video Retrieval and Tracking from Compressed Multimedia Databases–Multiple Object Tracking from MPEG-2 Bit Stream," *Journal of Visual Communication and Image Representation* , vol II, pp 154–182, 2000.

[6] H. Wang, H. S. Stone, and S.-F. Chang, "FaceTrack: Tracking and Summarizing Faces from Compressed Video," in *Proc. SPIE Multimedia Storage and Archiving Systems IV*, Boston, MA, Sept. 1999.

[7] T. Wiegand and G. J. Sullivan and G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp 560–576, July 2003.

[8] M. Wien, "Variable block-size transforms for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp 604–613, July 2003.