# Statistical Model Based Video Segmentation and Its Application to Very Low Bit Rate Video Coding

Huitao Luo[†], Jack Kouloheris[‡] and Alexandros Eleftheriadis[†]
[†]Dept. of Electrical Engineering, Columbia University
[‡]Video and Image System Group, IBM T.J.Watson Research Center

**Abstract**

*We present a statistical model-based video segmentation algorithm for typical videophone and videoconference applications. This algorithm makes use of online information and tries to build statistical models for both background and foreground and update the models on the fly. A hierarchical system structure is designed and segmentation is combined with tracking. Two possible applications are discussed: to generate VOP for MPEG-4 and to introduce subjective rate control for DCT-based algorithms. A R-D based rate control algorithm for H.263 is proposed and implemented as an example.*

## 1 Introduction

In current multimedia and coding research, more and more focus is directed to the area of computer vision, especially segmentation. Efforts are made to get semantic or partial semantic information from video signals and combine them with traditional coder designs. Such examples include "region based coding" [5], "model assisted coding" [8], "model based coding" [4], and most of all, the upcoming MPEG-4 standard. Obviously, general-purpose image segmentation has long been a difficult problem. How to generate semantically meaningful VOPs is a critical problem for MPEG4.

In this paper, we propose a simple but interesting and effective online video segmentation algorithm, which uses an inexpensive videoconference quality camera and runs on a Pentium PC in real time. The application we have in mind is mainly "head-and-shoulders" type videoconference, but the same principle may also apply to other applications. This work is motivated by the human-body tracking work of "Pfinder" [3]. We use similar statistical models but develop their idea from tracking to video object segmentation. A hierarchical architecture is designed to enable real time performance on a PC and spatial and temporal filters are designed to improve the boundary quality.

Applications of this segmentation system include creating MPEG-4 VOPs and improving the quantization and rate control mechanism for traditional coder like H.263. It may also be applied to model based coding. The factor here is how much semantic knowledge is used in the encoder. In this work we use segmentation knowledge to guide the rate control for H.263. This part is in some degree motivated by the work of [8],[9],[10], but as our segmentation is improved compared with theirs, we can design better spatial and temporal adaptivity in our algorithm. In addition, we design an integrated distortion model which includes both subjective and objective factors and try to optimize the rate control in a rate-distortion sense.

## 2 Statistical Model Based Video Segmentation

### 2.1 Statistical Modeling

We model the segmentation as a MAP classification problem as in [3]. The straightforward idea is to create a statistical model for the background when there is no foreground in the scene. Then, when the foreground enters, create another model for the foreground and classify the pixels between the statistical models. The only assumption is that the camera is static and the background is not changing rapidly.

In the videophone case, the foreground is "head-and-shoulders". We use two "blob"s to represent the head and shoulders separately. Here we use the same definition of "blob" as [3], i.e., each blob has a spatial $(x, y)$ and chromatic $(Y, U, V)$ gaussian distribution and a support map which indicates whether a pixel is a member of a blob. This is showed in Figure 1: the left image illustrates two blobs, the middle image shows a support map and the right image is a foreground map. In this model, each pixel is represented by a vector $(x, y, Y, U, V)$ and all the pixels in blob $k$ have gaussian distribution with mean $\mathbf{m}_k$ and covariance matrix $\mathbf{C}_k$. The idea behind blob modeling is that it represents an object that has chromatic and spatial similarity.

We model the background as a texture map that varies over time. In common videophone cases, we assume the camera is static and there is no major background change. We model each pixel in the background model as a gaussian distribution in a vector space $(Y, U, V)$ with mean $\mathbf{m}_0$ and covariance $\mathbf{C}_0$. During the segmentation loop, both foreground and background models are updated on the fly. Note that unlike the foreground model, each pixel of background is modeled individually. This can accommodate all kinds of complex backgrounds without limiting them to fit to a structure like "head- and-shoulders" for foreground.

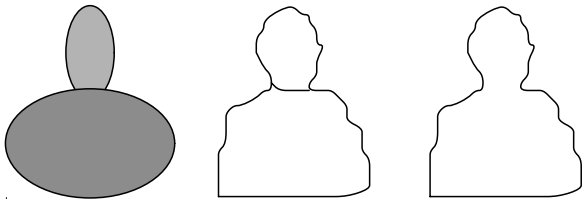

Figure 1: Left:blob representation, middle:support map, right:foreground map

With above modeling work, it is straightforward to classify pixels into different regions by their probability. In our work, we have two foreground classes ($head\ and\ shoulders, k = 1,2$) and one background class ($k = 0$). This can be expressed as:

$$d_k = -(\hat{\mathbf{y}}-\mathbf{m}_k)^T \mathbf{C}_k^{-1}(\hat{\mathbf{y}}-\mathbf{m}_k) - \ln(\det(\mathbf{C})),\ (k = 0,1,2) \quad (1)$$

Each pixel is labeled in support map as:

$$l(x,y) = \text{argmax}_k(d_k(x,y)) \quad (2)$$

### 2.2 Segmentation System

In Pfinder[3], the purpose of the system is to track the spatial positions of the blobs. The requirement for segmentation is not so strict and some noise is acceptable because segmentation is used as an "observation input" for a Kalman filter to track the blobs. But in our system the emphasis is segmentation, while tracking is used as a means to improve the segmentation quality. To get an accurate and refined segmentation boundary is much more computationally expensive than [3].

Based on this consideration, we design a hierarchical system. Each input frame is first subsampled. Blob tracking is carried out in the lower resolution frame. Boundary refinement is then done in the full resolution frame. In this way, tracking and segmentation requirements are addressed differently and efficiently (when spatial position of each blob is tracked in the lower resolution, only the boundary blocks in the full resolution are further processed to get refined boundary). In addition with a hierarchical structure it is easy to offer scalable output for different applications.

The processing steps in the lower resolution frame include:

1. The spatial position of each blob is predicted from current system status with a Kalman filter.
2. For each pixel, a MAP criterion is used to classify it into different regions.
3. Morphology filters are used to convert pixel-by-pixel classification into a simple connected support map for each blob.
4. The statistical model for each blob is updated according to the new segmentation result.

Figure 2: Segmentation results at different resolutions

The processing steps in the full resolution frame include:

1. Pixel-by-pixel classification of those pixels located in the boundary blocks derived from the low resolution segmentation.
2. Morphology filters are used to convert the pixel-by-pixel classification into a simple connected support map.
3. Spatial relaxation is used to refine the boundary.
4. A temporal median filter is used to suppress the temporal high frequency on the boundary.
5. The statistical model for each blob is updated according to the new segmentation result.

Due to space limitations, we do not discuss these steps in detail.

This system is implemented on a 150MHz Pentium with an Intel videoconference camera and capture card. The performance is 9 frames per second for a full resolution ($160 \times 120$) and 25 frames per second for a subsampled resolution ($40 \times 30$) only. Figure 2 shows two segmentation results in the lower(left) and the full(right) resolution. Notice that in the right image the gray region labels the tracked head region while in the left image gray region is not used to enable better observation.

From the experimented results, we believe this segmentation approach can be used in some cases to generate MPEG-4 VOPs with a feasible limitation. In addition, it may also be applied to H.263 type encoder rate control so as to improve the subjective quality of videoconference, i.e., we can allocate more bits to head region macroblock(MB)s. Because of its real time feature, it is feasible to have a combined real time segmentation and encoding system to be used in a videophone application.

## 3 Region based bit allocation and rate control

In this work, we try to incorporate subjective factors obtained by the previous segmentation, into traditional R-D model and design an optimal H.263 compatible encoder in this sense.

Early relevant research can be found in [8],[9] and [10]. Our approach improves over theirs by introducing an integrated bit allocation algorithm that combines the subjective factor and rate distortion criterion. In this paper we call the new encoder "region based encoder".

In general this work is based on Telenor's implementation of H.263[11]. But its rate control part is improved to be region-adaptive. Our contribution can be summed in three aspects: bit allocation, temporal adaptivity and adaptive quantization.

### 3.1 Bit allocation

There are two steps in bit allocation, one is to allocate bits to frames and the other is to allocate bits to macroblocks. In H.263, there are only I and P frames and no B frames. Because of H.263's real time feature, P frames are the major frame in an H.263 stream and I frames are only inserted to compensate channel error. It is not possible to segment an input video beforehand into frame groups and assign bits to each I and P frames according to their relative complexity measures as in [6]. We allocate bits uniformly to each frame as TM5 does while concentrating on the bit allocation to different macroblocks.

**Problem:** The problem can be expressed as follows. Given the bit budget $\bar{B}$ for each frame, find the optimal bit allocation $\hat{B_i}$ to $M$ macroblocks ($i = 1, 2, \cdots, M$) that minimizes the frame distortion in the R- D sense.

**Distortion Model:** In H.263, all the DCT coefficients within one macroblock are quantized with one quantizer $Q$. Assuming DCT coefficients have a uniform distribution, the quantization error for macroblock $i$ is

$$D_i = \sum \Delta_i = \sum \Delta_f = k * Q_i^2 \tag{3}$$

As in [7], we use $X_i = Q_i * B_i$ to represent the macroblock complexity, we then have

$$D_i = k \frac{X_i^2}{B_i^2} \tag{4}$$

We further include a subjective importance factor $\beta_{l(i)}$ where $l(i)$ is the region label of macroblock i,($l(i) = background, head, or\ shoulders$). The final distortion equation is

$$D_i = \frac{\beta_{l(i)} X_i^2}{B_i^2} \tag{5}$$

**R-D optimization:** With distortion model eq.(5), it is easy to solve the R-D optimization problem with the Lagrange multipliers method:

$$S = D + \lambda B = \sum_{i=1}^{M} D_i + \lambda \sum_{i=1}^{M} B_i \tag{6}$$

Setting $\frac{\partial S}{\partial B_i} = 0$, we have

$$k \frac{\beta_{l(i)} X_i^2}{B_i^3} = \lambda \tag{7}$$

Now we use the quantization model for MPEG encoder in [6]:

$$B_i = a_{P,i} mad_i Q_i^{b_{P,i}} \tag{8}$$

$$B_i = a_{I,i} \Delta_i Q_i^{b_{I,i}} \tag{9}$$

where $mad_i$ is the motion predicted mean absolute difference for P frame MB $i$, and $\Delta_i$ is the mean absolute difference for I frame MB $i$. For H.263, we adapt its P frame equation and assume $mad_i$ is linear to block complexity factor $X_i$:

$$B_i = a_i mad_i Q_i^{b_i} \tag{10}$$

$$mad_i = k * X_i = k * Q_i * B_i \tag{11}$$

where the parameters $a_i$ and $b_i$ are MB dependent. Substituting (10),(11) into (7) yields

$$k a_i^{\frac{-2}{b_i}} \beta_{l(i)} B_i^{\frac{2}{b_i}-1} mad^{\frac{-2}{b_i}} = \lambda \tag{12}$$

According to [6], the empirical value for $b_i$ is $-1.5$, then the equation becomes

$$\frac{a_i^{\frac{4}{7}} \beta_{l(i)}^{\frac{3}{7}} mad^{\frac{4}{7}}}{B_i} = \lambda / k^{\frac{3}{7}} \tag{13}$$

This equation shows that the bit allocation $B_i$ is proportional (general sense) to the subjective importance $\beta_{l(i)}$ and the objective complexity $mad_i$. Let

$$w_i = a_i^{\frac{4}{7}} \beta_{l(i)}^{\frac{3}{7}} mad^{\frac{4}{7}} \tag{14}$$

the optimal bit allocation to each MB can be expressed

$$\hat{B_i} = \frac{w_i}{\sum_{k=1}^{M} w_k} \bar{B} \tag{15}$$

where $\bar{B}$ is the average bits for each frame and $\hat{B_i}$ is the adaptive allocated bits to MB $i$. Note this optimization is based on the empirical quantization equation (10) and (11). So it is not an ultimate optimal solution. Different approach based on operational R-D optimization is discussed extensively in [14]. Their solution is computationally expensive and is intended for benchmarking rather than real time implementation. In our case, efforts are made to fit the empirical equations (10) and (11) to actual video source in an adaptive way (see eq.(19),(20)). Under this empirical model assumption, we claim that equation (15) approximates the optimal solution.

### 3.2 Temporal adaptivity

In the bit allocation weight equation (14), a subjective factor $\beta_{l(i)}$ is used to control the bit allocation so as to introduce spatial adaptivity. In most videoconference cases, the background is relative static and it is reasonable to reduce its temporal refresh rate while maintain good visual quality. Under this consideration, we classify the P frames in H.263 into PO and PF frames. In PO frames only the foreground object

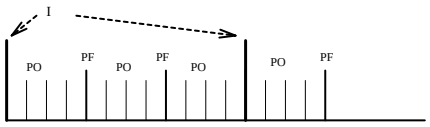


Figure 3: PF and PO frames illustration

MBs are coded, while in PF frames all the MBs in the frame are coded. PF frames work as anchors for PO frames and their temporal relation is illustrated in Figure 3 where possible I frames are also included. The parameter $T_{PF}$ (time between two PF frames) should be adjusted according to an estimate of background motion. In this work we set $T_{PF}$ to be the number of PO frames between two PF frames. Also notice that PO and PF control can be easily integrated into the bit allocation equation (15) by assigning $\beta_{background} = 0$ for PO and $\beta_{background} \neq 0$ for PF.

### 3.3 Adaptive quantization

In the above discussion, we derived an optimal bit allocation to each MB but how to choose proper quantizers to achieve this allocation remains a problem. Actually, there is no clear theoretical function like $Q = f(B)$ or $B = f(Q)$. In this work, we use a feed back control to realize the bit allocation budget. In addition, the modeling parameter $a_i$ is also estimated online in a feed back way.

In general, we adapt Telenor's feed back rate control scheme, which is expressed as:

$$Q_{t,i} = \bar{Q}_{t-1,l(i)}(1 + \Delta_G + \Delta_L) \tag{16}$$

where

$$\Delta_G = (B_{t-1} - \bar{B})/(2\bar{B}) \tag{17}$$

$$\Delta_L = 4(\sum_{k=0}^{i-1} B_{t,k} - \sum_{k=0}^{i-1} \hat{B}_{t,k})/R \tag{18}$$

Compared with Telenor's scheme, we maintain different average quantizer $\bar{Q}_{t-1,l(i)}$ for different labeled region $l(i)$. The idea is that a quantizer is similar between temporal neighboring frames for the same region type. In addition, region adaptive cumulative rate target $\sum_{k=0}^{i-1} \hat{B}_{t,k}$ is used to generate local adaptive factor $\Delta_L$ (MB level). In eq. (18), $\sum_{k=0}^{i-1} B_{t,k}$ is the actual bit consumption for the first $i-1$ MBs thus far, $\sum_{k=0}^{i-1} \hat{B}_{t,k}$ is the cumulative target rate. Note $\hat{B}_{t,k}$ is obtained through eq. (15). But the global adaptive factor $\Delta_G$ (frame level) remains the same because we allocate bits uniformly to each frame.

For parameter $a_i$, we try to classify the MBs into different groups according to their *mad*:

$$g(i) = \frac{mad_i}{g_{threshold}} \tag{19}$$

And get the average $a_i$ for each group $\bar{a}_{t-1,g(i)}$ (in frame t-1). So we have

$$a(t,i) = a_{t,g(i)} = \bar{a}_{t-1,g(i)} \tag{20}$$

Note that in quantization model eq.(10) we set $b_i$ to be $-1.5$, and only $a_i$ is used to account for input adaptivity. By grouping the MBs according to their complexity, we can better control the performance of this empirical model.

### 3.4 Experimental results

We implement our algorithm on a PC with the hardware system described in 2.2. Because of the online feature of our segmentation algorithm, we can not use standard video sequences in our testing. Instead we captured a testing sequence with our videoconference system. The sequence is 500 frames in length, 15fps, one person with considerable motion before a static background. Because the hardware system uses a frame format of 160x120, we pad it to QCIF format.

First we compare our algorithm with Telenor's TM5. Their latest software version is TMN2.0 [12]. Both algorithms use the baseline mode. The target frame rate is 10fps, and the bit rate is 32kps. The parameters used in our algorithm are $g_{threshold} = 250$, $T_{PA} = 30$. The subjective factors used are: $\beta_{head} = 4, \beta_{shoulders} = 1, \beta_{background} = 0$ for PF frames and $\beta_{head} = 1, \beta_{shoulders} = 1, \beta_{background} = 1$ for PA frames.

Figure 5 shows the bit allocation of a typical PF frame by our region based algorithm. We number the images from left to right and top to bottom. (a) is the origin frame, (b) is mad, (c) is bit allocation budget and (d) is the actual achieved bit allocation by adaptive quantization. Note that (d) is not strictly equal to (c) due to the model error and the feed back nature of the control mechanism.

In Figure 6 we see that our algorithm improves the head region SNR by about 1 DB at the expense of dropping the overall SNR about the same amount in Figure 7. This change is desirable because comparing Figure 6 and Figure 7, TM5's overall SNR curve is always higher than its head region SNR curve while our algorithm have head region SNR higher than overall SNR. Figure 8 shows the stream bit rate of two algorithms. Because of PO and PF alternation in our region-based algorithm, it exhibits higher bit rate fluctuation than TM5. This fluctuation depends on the changing of background and is difficult to overcome in real time implementations. In general we can see that the feed back rate control scheme works well.

Figure 4 compares two reconstructed frames by the region based algorithm(left) and TM5(right). The region based algorithm exhibits better subjective quality in the head and facial region. In addition, the PO and PF alternation maintains good background quality and the overall trade off yields good subjective quality.

## References

[1] C. Lettera and L. Mastera, "Foreground/ Background segmentation in videotelephony",Signal Processing: Image Comm.(1), 1991, pp. 181-189.

[2] T. Aach, A. Kaup and R. Mester, "Statistical model-based change detection in moving video", Signal Processing(31), 1993, pp.165-180.

Figure 4: Comparison of reconstructed frames


[3] C. Wren,etc., "Pfinder, real-time tracking of the human body", MIT Media Lab technical report No. 353.

[4] K. Aizawa, etc. "Human facial motion analysis and synthesis with application to model-based coding", in Motion Analysis and Image Sequence Processing, Kluwer Academic Publishers,1993.

[5] P. Salembier and M. Pardas, "Hierarchical morphological segmentation for image sequence coding", IEEE Trans. IP , Vol.3, No.5, Sept. 1994.

[6] E. Viscito and C. Gonzales, "A video compression algorithm with adaptive bit- allocation and quantization", SPIE Proc. VCIP, Boston, MA., Nov. 1991, Vol.1605, pp.205-216.

[7] G. Keesman,I. Shah, R. Klein-Gunnewiek, "Bit-rate control for MPEG Encoders", Signal Processing: Image Comm.(6), 1995, PP. 545-560.

[8] A. Eleftheriadis and A. Jacquin, "Model-assisted coding of video telecoferencing sequences at low bit-rates", Proc. ISCAS'94, May-June, 1994.

[9] A. Eleftheriadis and A. Jacquin, "Automatic face location detection for model- assisted rate control in H.261-compatible coding of video", Signal Processing:Image Comm.(7), 1995, pp.435-455.

[10] J.-B. Lee and A. Eleftheriadis, "Motion adaptive model-assisted compatible coding with spatio-temporal scalability", VCIP '97, Feb. 1997.

[11] ITU TSS LBC - 95, Study Group 15, Working Party 15/1, Expert's Group on Very Low Bitrate Visual Telephony, "VIDEO CODEC TEST MODEL, TMN5", Jan.31, 1995.

[12] Software TMN2.0 at ftp://bonde.nta.no/pub/tmn/software/

[13] Y. Lee,etc., "Towards MPEG4: An improved H.263 based video coder", Signal Processing:Image Comm.(10), 1997, pp.143-158.

[14] Antonio Ortega, "Optimization techniques for adaptive quantization of image and video under delay contraints", Ph.D. Thesis, Columbia Unversity, 1994


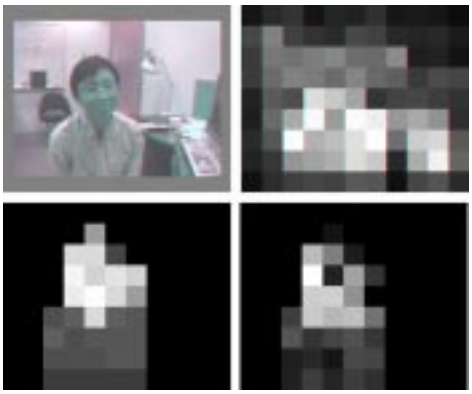

Figure 5: Bit allocation to macro blocks

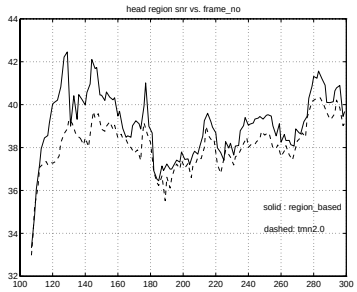


Figure 6: Head region snr vs. frame number

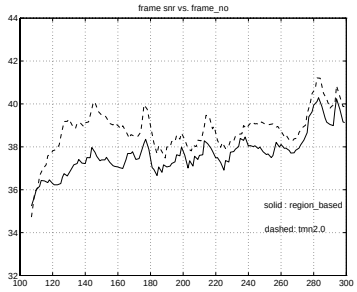


Figure 7: Overall snr vs. frame number

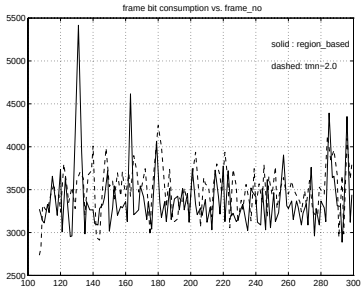


Figure 8: Bits consumption vs. frame number