Temporal Segmentation of Video Sequences for Content-Based Coding

JULIANA F. CAMAPUM WANDERLEY AND DANIEL C. DANNA

UnB/ENE – Universidade de Brasília, Departamento de Engenharia Elétrica, 70919-970, Brasília, DF, Brasil
juliana@cic.unb.br; dcdanna@terra.com.br

Abstract. In this paper, we present an overview of the state of the art of segmentation based video coding aimed at low-bit rate application. The most significant contribution is the algorithm for the segmentation of videoconference image sequences for content-based coding. The segmentation is based on motion estimation trough the computation of the optical flow field and motion segmentation by applying a Graph-Theoretical clustering. The algorithm will be added to the video codecs of the OpenH323 project which is based on H.323 ITU-T Recommendation. This visual collaboration environment for Desktop, composed of terminals, gatekeepers and MCU (Multipoint Control Unit), has already been implemented and tested. It is particularly convenient for academics implementation, the software is freely distributed, and its code can be modified according to the needs of the user. However, the first tests using the H.261 codec showed a video delay of 3 seconds, confirming the need to optimize the video codec.

1. Introduction

Many strategies have been proposed for video coding. They can be divided in block-based and region-based techniques. Most algorithms use block based motion compensation strategies. Although widely adopted, unfortunately these techniques, for low bit rate applications perform rather poorly and suffer from severe drawbacks.

Studies of the human visual system emphasised the importance of image edges in the interpretation of images leading to a new generation of image coding algorithm, called region-based coding. They are characterized by segmentation based motion compensation schemes.

2. Block-based coding

Conventional coding methods such as the standards H.261, MPEG1 and H.263 belong to the first generation coding techniques. The standard implementation is based on the transform block-based coding DCT to remove intraframe (spatial) correlation and motion estimation to solve the temporal redundancy reduction problem. Motion estimation is calculated by matching individual blocks with blocks in a neighborhood and then the block displacement estimation is transmitted as the motion vector. The decoder predicts the current frame using motion information from previously decoded frame. The residual, difference between the actual video frame and the frame resulting from motion compensation, is partially transmitted according to bit-rate restrictions. An intra-coded frame, called refresher frame, is also periodically transmitted.

The aim of these standards is to optimize the compression performance. The H.263 standard made it possible the implementation of video conferencing at bit rates less than 64 kbit/s.

However, some of the problems like noisy motion fields, poor prediction of edges and block artifacts, were not solved yet. For example, when a block contains more than one object moving in different directions, motion estimation is inaccurate because pixels in a block have the same motion vector. They are also computationally intensive because pixels considered for block matching covers entire image domain.

In the last years, many researchers proposed several improvements to the standard block-based coding. Some of them are variable block size through quad-tree partitioning, pre or post filtering to reduce block artifacts and split the image into 16x16 blocks and find the best backward motion vector for each block. However, block based methods results in a peaky distribution of residual error at block edges and image edges and blurring of image edges.

In [1], a new algorithm is proposed improving image quality by reducing the blocking artifacts. Optical flow vectors are estimated and divided into blocks of 8x8 and then DCT coding is applied to the motion vectors and also to the predictive errors.

Another approach was suggested by Cafforio [2]. He combined block matching with polynomial approximation to derive a new motion estimation technique. Measured block matching functions are input to a region-growing algorithm that searches the polynomial motion fields that best match the data. Though the prediction error is close to the one obtained in classical block matching techniques, results are more akin to a "physically" meaningful one.

Recently, the efforts aim to provide greater flexibility for "content-based" access and manipulation of data.

Therefore, the bulk of the very low bit-rate coding research has centered on the second-generation techniques [3], the region-based coding.

3. Region-based coding

A natural alternative to the block-based standards is object based coding, first proposed by Musmann *et al.* [4].

The object based coding is called second generation technique and is based on image content. They are divided in high bit-rate standards such as MPEG2 which are applied to data storage in compact discs and low bit rate such as MPEG4 which are applied to video teleconferencing, video telephony, video phones. The last one is considered more challenging and it is the scope of this research.

In the object-based approach, a prior decomposition of sequences into semantically meaningful, physical objects is required. The moving objects in the video scene are extracted, and each object is represented by its shape, motion, and texture. Parameters representing the three components are encoded and transmitted, and the reconstruction is performed by synthesizing each object. The most important cue exploited by a majority of techniques is motion.

Physical objects are often characterized by a coherent motion that is different from that of the background. So-called change detection masks (CDM's) and estimated flow fields are the most common forms of motion information incorporated into the segmentation process.

The segmentation of the scene is very important in the object-based approach. The segmentation step can be divided into motion field segmentation and intensity based segmentation. Motion field segmentation is based on a similarity in motion and intensity based is related to gray scale homogeneity. Intensity segmentation is being more common though they are often suboptimal in terms of coding efficiency. A motion-based segmentation is more desirable in representing moving objects.

When segmentation information needs to be encoded, this may result in an unacceptable amount of overhead for the low bit rate applications.

Yokoyama *et al.* [5] proposed a region-based scheme using intensity based segmentation. The contour information need not be encoded at all since the regions are obtained by segmenting the previously decoded frame. The drawback is that holes and overlapping regions cannot be predicted. Yoon [6] generated a dense motion field as a strategy for filling in the holes and overlapping regions and a multiscale segmentation as a rate control strategy to regulate the amount of motion information transmitted. The scale parameter controls gray level homogeneity and consequently, the number of regions. The motion estimation is done as an *afterthought* to fit the segmentation that has already been found.

In [7], the segmentation process applies a spatiotemporal filter based on intensity differences. Next, 3-D motion is estimated and only the first frame and motion parameters for consequent frames are transmitted. The algorithm separate and track regions analyzing a number of frames simultaneously.

An effective and high performance video coding algorithm for low bit-rate system can be achieved by combining region segmentation with optical flow. Chen [8] developed an algorithm that first calculates the interframe motion flow through a modified optical flow algorithm, then segment the motion vectors and code the information (shape and motion). This method can be used for both, image reconstruction and image analysis. Besides, there are no residual errors to transmit.

Some algorithms treat motion estimation and segmentation together. In [9], a region growing motion segmentation and estimation is developed. The motion model parameters of labeled regions are tested for unlabeled pixels so that regions get larger while satisfying the error criterion. The process involves reestimation of the motion parameters as the labeled regions grow. Motion parameters are estimated by constraining the optical flow by a quadratic transform. This coding scheme assume object patches with quadratic surface and 3-D rigid motion and depend on the variance of the scene image. In [10], a first coarse segmentation is obtained by computing the optical flow vector for each image pixel, then a merging process is applied by modeling by an affine function the optic flow field on each region and joining regions having similar motion models.

Yang [11] developed a clustering algorithm based on morphological operations for motion field estimation and segmentation in both the encoder and decoder. An initial motion vector is estimated for each cluster core and motion information is used to continue growing the whole cluster. The motion vector of the cluster is estimated and used to guide the morphological growing. Motion compensation reveals performance gain through cluster matching instead of using block matching. Moreover, coding efficiency is increased at the expense of higher complexity in the decoder since shape information is not transmitted. The prediction error is coded by adaptive scalar quantization method. Experimental tests on standard video sequences lead to better results over the MPEG-1.

A joint motion estimation and segmentation algorithm based on the Markov random field (MRF) model was developed by Han [12]. An extended MRF model with constraints such as spatiotemporal smoothness and consistency of motion vectors with segmentation is implemented. The clustering is based on both the motion

and intensity information. The object motion and contour are encoded efficiently with temporal updating. Simulations showed comparable performance to the H.263 coder with less blurriness and devoid of block artifacts.

The next section will describe the visual collaboration environment where the region-based coder will be introduced and tested. Next, the segmentation algorithm which will identify objects in the scene for low bit-rate coding is presented.

4. Definition and test of the Visual Collaboration Environment OpenH323

The visual collaboration environment was developed based on H.323 ITU-T Recommendation [13] and realized in the context of the OpenH323 project [14]. The decision to use open software decreased the cost of the system and allowed the manipulation of the source code of the entities. This possibility is very interesting when there is a need to optimize the system, in our case, the video codec.

The entities H.323 OhPhone (H.323 terminal), OpenGatekeeper and OpenMCU were implemented and tested [15]. The main objective was to establish a communication channel. The next step will be the introduction of the new video codec based on motion segmentation that is being developed.

Several testes were realized. One of them consisted of three terminals and one MCU (*Multipoint Control Unit*) as can be seen in Figure 1. The test consisted of realizing calls from the terminals to the MCU in order to verify the division of the client screen by the MCU, the exchanged packets and the quality of the established session. The registration of the terminals to the MCU used the H.225 protocol and information like connectionRequest, connectionConfirm, terminalAlias, conferenceAlias and terminalType. The exchanged messages followed the H.323 Recommendation.

When a terminal is registered to the MCU, the audio and video coded are negotiated. In this test the H.261 and G.711 codecs were chosen automatically.

The audio and video quality was low. The audio showed a delay of 2s and the video around 3s and the image was not consistent, making it difficult the interaction. Further tests were realized using just audio terminals and hardly any problem was detected in the session. All the clients could listen to each other with a small delay of 1s.

Therefore, the major problem detected was the video codec. In order to minimize this problem by optimizing the performance of the video codec, an algorithm based on motion segmentation which takes advantage of temporal redundancies is being developed. In the next section, the motion segmentation algorithm is presented.

5. The motion segmentation algorithm

The segmentation algorithm is divided in two parts. Firstly, the motion of the object is detected, then a clustering algorithm is used to group pixels based on similar motion and spatial features.

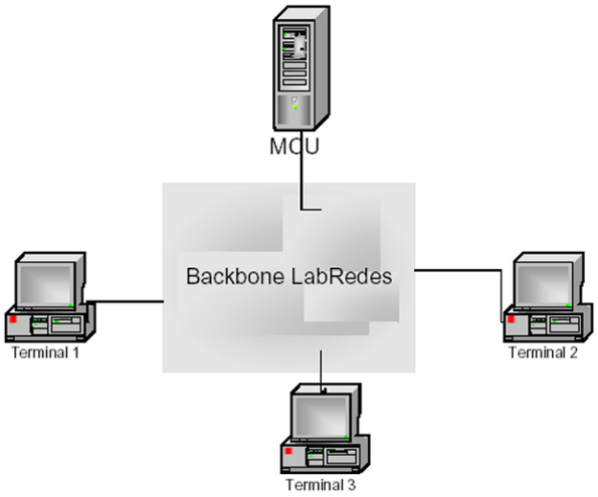


Figure 1 Visual Collaboration Environment with three terminals and one MCU.

5.1 Motion Detection

The first step in segmenting a moving object of interest from a complex background is the computation of the normal component of the optical flow. As accurate estimation of visual motion is computationally expensive, a filtering technique is applied which provides an output zero-crossing image $S(x,y,t)$ by convolving the intensity history of the image $I(x,y,t)$ with a d'Alembertian of a spatial-temporal Gaussian filter [16]. This process is defined by Eq. (1) and (2). The width of the filter is given by $w = 2\sigma\sqrt{2}$ where σ is the standard deviation of the Gaussian function.

$$S(x,y,t) = -\left(\nabla^2 + \frac{1}{u^2}\frac{\partial^2}{\partial t^2}\right) F(x,y,t) \otimes I(x,y,t) \quad (1)$$

$$F(x,y,t) = u\left(\frac{1}{2\sigma^2\pi}\right)^{3/2} \exp\left[-\frac{1}{2\sigma^2}\left(x^2 + y^2 + u^2t^2\right)\right] \quad (2)$$

u is a time scaling factor and $\otimes$ represents the convolution process.

The next step is to detect the zero-crossings in $S(x,y,t)$ and calculate the magnitude of the velocity of each zero-crossing pixel. The velocity is calculated using a $5\times5\times5$ neighborhood so that we need at least five successive image frames to calculate the spatial-temporal zero-crossings. The magnitude of the visual motion v_n

(Eq. 3) in the image plane is estimated for the middle frame of five successive image frames.

$$v_n = \frac{u\frac{\partial S}{\partial t}}{\sqrt{\left(\frac{\partial S}{\partial x}\right)^2 + \left(\frac{\partial S}{\partial y}\right)^2}} \quad (3)$$

where the derivatives $\partial S/\partial x$, $\partial S/\partial y$ and $\partial S/\partial t$ were calculated using numerical finite difference methods on values from Eq. (1). Subsequently, the magnitude of the normal visual motion (velocity) of the middle frame is scaled in the range 0 .. 255. In Figure 2, the scaled velocity image I_v is displayed in grayscale and referred to as the 'velocity grayscaled image'.

(a)

(b)

(c)

Figure 2 (a) original image (b) velocity gray-scaled image (c) segmented image (foreground and background)

5.2 Spatial-Feature (motion) Clustering

Our aim is to cluster pixels in the velocity gray-scaled image into patches having similar velocity and to extract the object of interest from the slowly moving background by applying a simple threshold. This is achieved by adopting the Graph Theoretical (GT) clustering algorithm of Koontz *et al.* [17]. Pixels are grouped together based on their spatial position and velocity. For example, consider Figure 2a showing an image represented by a matrix of $N \times M$ pixels that we would like to segment. The steps are:

1. Calculate the velocity gray-scaled image I_v (Figure 2b) as described in section 5.1 and apply a velocity threshold (e.g. $\overline{vel}$) to the image I_v giving $I_v^{'}$. The velocity threshold is discussed in more detail in [18] [19].
2. Construct a 2-D spatial histogram $C[x][y]$ for the thresholded velocity image $I_v^{'}$ having $Q \times Q$ quantization levels each representing a spatial increment of

$$CS_x = \frac{N}{Q} \quad and \quad CS_y = \frac{M}{Q} \quad (4)$$

where CS_x and CS_y represent the size of the histogram bin with coordinates x and y respectively. The choice of value for Q is discussed at the end of this section but good results were obtained with a typical value of $Q = 32$.

The indices (x, y) of each pixel $I_v^{'}(i, j)$ are calculated as

$$x = f(i) = INT\left\{\frac{i}{CS_x} + 1\right\}$$
$$y = g(j) = INT\left\{\frac{j}{CS_y} + 1\right\} \quad (5)$$
$$i = 1 \ldots N \text{ and } j = 1 \ldots M$$

In Eq. 5, the operation $INT(z)$ gives the largest integer smaller than z.

The pixel $I_v^{'}(i, j)$ belongs to one of the spatial histogram bins if the pixel velocity is above an estimated threshold $\overline{vel}$. Hence, each spatial histogram entry represents a sum of pixel counts (Eq. 6).

$$C[x][y] = C[x][y] + 1 \Leftrightarrow I_v^{'}(i, j) > \overline{vel}$$
$$x = f(i), y = g(j) \quad (6)$$

The velocity threshold (for example, $\overline{vel} = 5.25$) is used to ensure that only pixels with a relatively strong motion are associated with the bin.

3. GT clustering is used to build a tree structure with a single root representing each unimodal cluster in the following way (see Koontz *et al*. [17] for a more detailed description). Given a spatial histogram, select each bin in turn and examine its eight neighbors; choose the one with the biggest pixel count to establish a link if the neighbor is larger than the current bin. If the current bin has the same value as the biggest neighbor, then one of them is chosen arbitrarily to be the root and a link is established. A link between two bins signifies that all the pixels inside the two bins belong to the same cluster.

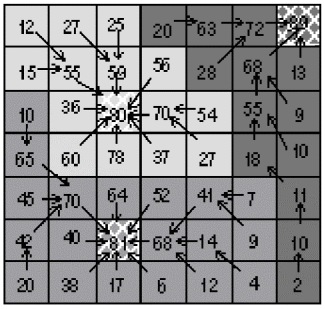
   This process will identify several clusters; each with a unique root. Two adjacent trees can be merged together to form a larger cluster as follows. Find the root of each tree which is the maximal bin in each cluster. Check the position of the roots. If two adjacent roots are found, join the two clusters and choose the maximal bin to be the root. Assign a label for each bin with the number of the bin cluster.

   Figure 3a illustrates a typical 2D spatial histogram and Figure 3b trees built using 8- nearest neighbor GT clustering. In this case three clusters are identified, and the paths from each branch to the root identify a unimodal cluster.

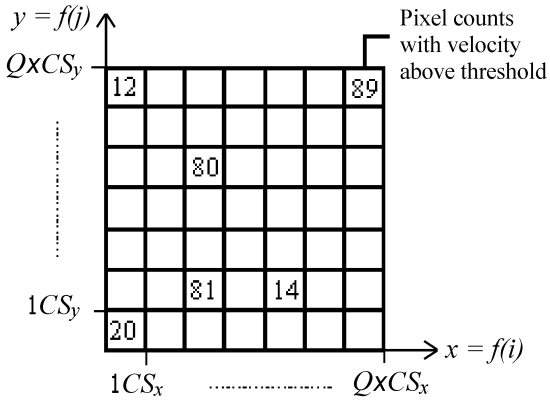
4. Finally, pixels belonging to the segmented clusters are back-projected in the original color frame to give a segmented output color image. Back-projection is performed by sweeping through the original color image in the spatial domain and creating a new image such that only the pixels that belong to one of the segmented clusters are copied from the original image to the new image. To find out if a pixel belongs to one of the clusters, it is necessary to calculate the indices of its bin (x, y) from the spatial coordinates (i, j) of the pixel (Eq. 5). Each bin is labeled with the number of the cluster to which it belongs; this label being assigned during the clustering step.

When an image sequence of N frames is analyzed, we can run the algorithm for each middle frame of 5 creating a segmented color image as described above. By accumulating clustered pixels over several frames in a sequence we create what is called a 'segmented image sequence'. Figure 2c shows the segmented image sequence after running the algorithm for six middle frames computed from an image sequence of ten frames.

An optimization method is used to determine the best histogram partition or, in other words, how many bins the histogram should have. This involves repeating the clustering process (step 3) for different values of Q and choosing the quantization according to a measure of clustering compactness and isolation [19] [20]. The list of pixels belonging to a cluster needs to be larger than a threshold, which is a percentage of the maximum cluster. Otherwise, the cluster is eliminated.



(a)



(b)

Figure 3 (a) 2-D spatial histogram $C[x][y]$ with $Q \times Q$ bins. (b) An example of an 8-nearest neighborhood GT clustering

6. Conclusions and Future Work

In order to achieve an optimized video coder for a video collaboration environment, we have presented an overview of segmentation content-based coding of videoconference image sequences and a new algorithm for motion based segmentation applying optical flow. The key ideas are: the algorithm uses optical flow for motion estimation and Graph-Theoretical clustering for motion segmentation. The motion field is segmented into clusters associated with distinct regions. Besides, we present the implementation and test of the visual collaboration environment based on the OpenH323 project.

The next task should be the transmission of the motion parameters and test the algorithm performance for stationary and non-stationary background. Experiments

should be carried out on several sequences showing a comparison of coding results for several CIF image sequences and different standard and non-standard algorithms.

7. Acknowledgements

We thank the Brazilian Research Agency CNPq and FINATEC for their financial support.

8. References

[1] Shu Lin, Y. Q. Shi and Ya-Qin Zhang, "An optical flow based motion compensation algorithm for very low bit-rate video coding, pp. 2869-2872, *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1997.

[2] C. Cafforio, E. Di Sciascio and C. Guaragnella, "Motion estimation and region segmentation via functional optimization", *Proc. of IEEE DSP'97*, pp. 1123-1126, 1997.

[3] C S Choi, K. Aizawa, H. Harashima and T. Takebe, "Analysis and synthesis of facial image sequences in model-based image coding", *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 4, no. 3, pp. 257-275, June 1994.

[4] H. G. Musmann, M. Hotter and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images, " *Signal Process., Image Com*., vol. 1, no. 2, pp. 117-138, Oct. 1989.

[5] Y. Yokoyama, Y. Miyamoto, and M. Ohta, "Very low bit rate video coding using arbitrarily shaped region-based motion compensation", *IEEE Trans. Circuits Syst. for Video Technol*., vol. 5, pp. 500-507, Dec. 1995.

[6] S. C. Yoon, K. Ratakonda and N. Ahuja, Low bit-rate video coding with implicit multiscale segmentation, *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 9, no. 7, pp. 1115-1129, Oct. 1999.

[7] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences", *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 10, no. 8, pp. 1388-1402, Dec. 2000.

[8] L. Chen, Y. Chiu, T. Chiueh and H. Jong, "Object-oriented video coding algorithm for very low bit-rate system", *Proc. of IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 614-618, 1994.

[9] Y. Yemez, B. Sankur and E. Anarim, "Region growing motion segmentation and estimation in object-oriented video coding", *Proc. of IEEE Int. Conf. on Image Processing*, pp. 521-524, 1996.

[10] F. Bartolini, V. Cappellini and L. Tucci, "Simultaneous Optic Flow Estimation and Segmentation by means of LS Techniques", *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, pp. 97-100, 1997.

[11] X. Yang and K. Ramchandran, "A low-complexity region-based video coder using backward morphological motion field segmentation", *IEEE Trans. on Image Processing*, vol. 8, no. 3, pp. 332-345, March 1999.

[12] S. Han and J. Woods, "Adaptive coding of moving objects for very low bit rates", *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp. 56-70, Jan. 1998.

[13] H.323 Recommendation – Packet-Based Multimedia Communications Systems, 07/2000.

[14] http://www.openh323.org, "OpenH323 Project", *Equivalence Pty Ltd*., 1998.

[15] D. C. Danna, "Definição e Implantação de Ambiente de Colaboração Visual com Ferramentas de Software de Código Aberto", *Undergraduation Final Project*, Electrical Engineering Department, University of Brasília, 86 pages, July 2001.

[16] B. F. Buxton and H. Buxton, "Monocular depth perception from optical flow by space time signal processing," *in Proc. Royal Society of London*, UK, B 218, pp. 27-47, 1983.

[17] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A graph-theoretic approach to non-parametric clustering," *IEEE Trans. on Computers*, vol. C-25, no. 9, pp. 936-944, Sep. 1976.

[18] J. F. Camapum Wanderley and M. H. Fisher, "Spatial-Feature Parametric Clustering Applied to Motion-Based Segmentation in Camouflage", *Computer Vision and Image Understanding*, no. 84, pp. 1-14, December 2001.

[19] J. F. Camapum Wanderley, "Colour-based Recognition for Remote Sensing in Environmental Systems," *Ph.D. dissertation*, Coventry University, Coventry, UK, Feb. 1999.

[20] A. Khotanzad and A. Bouarfa, "Image segmentation by a parallel, nonparametric histogram based clustering algorithm," *Pattern Recognition*, pp. 961-973, 1990.