LOW BIT RATE VIDEO CODING FOR MOBILE MULTIMEDIA COMMUNICATIONS♦

Reginald L. Lagendijk, Jan Biemond and Cor P. Quist
Delft University of Technology, Department of Electrical Engineering, Information Theory Group
P.O. Box 5031, 2600 GA Delft, The Netherlands
Tel: +31 15 278 3731; Fax: +31 15 278 1843
e-mail: {lagendijk,biemond}@et.tudelft.nl; WWW: http://www-it.et.tudelft.nl

ABSTRACT

In this paper we first describe the objectives of the Delft Mobile Multimedia Communications project. Next, the subject of lossy contour compression is considered in more detail as it is an essential component of most object or region-based compression techniques for low bit rate video coding. We propose an optimized B-splines approximation approach, which results in a 40 percent higher compression than the lossless conditional chain code method. Achieved rates are, depending on the tolerable deviation between original and coded contour, in the order of 0.70 to 0.90 bit per contour pixel.

1. MOBILE COMMUNICATIONS PROJECT

The information superhighway has given access to multimedia communications for many users worldwide. However, communication of integrated video, audio and data streams is currently limited mostly to wired computer systems. Multimedia communications may very well serve the interests of mobile users, especially those for which mobile communication systems greatly enhance the quality of professional tasks to be carried out. Especially in the Netherlands, where a significant portion of the gross national product in earned by mobile workers (transportation sector), early experiences with and evaluation of mobile multimedia communication systems is crucial.

The most recently realized mobile communication system in Europe – GSM – is, however, oriented too much on a call duration of a few minutes and does not provide flexible bandwidth allocation. Both aspects are considered essential in realizing efficient and user friendly multimedia communications.

There are many projects and standardization efforts worldwide that concentrate on the technical issues involved in mobile multimedia communications, such as transmission and modulation techniques, transmission protocols and compression techniques [1]. The user's perspective, however, is as important to take into account as solving the technical challenges. The user's needs, expectations, perception, requirements and working environments of mobile multimedia systems should eventually determine the definition of the overall communication system, and, therefore, what technical solutions are really called for.

For these reasons, the Delft MMC research project [2] is investigating potential mobile services, the use of these services, and its organizational aspects in direct relation to transmission and compression techniques. Objectives are to find answers to the following questions:

- What is the impact of MMC on work efficiency, arrangements and communication in clusters of tasks, and which tasks are mostly in need of mobile multimedia communications;
- What combination of modalities offered by multimedia modalities (e.g. sound and video) is best fit for user interfaces in mobile environments;
- In which way should dynamic space-time-frequency allocation, based on the virtual cell transmission concept, be carried out for mobile users and services;
- Which efficient and transmission-error resistant compression techniques are best suited to the mobile user's needs and the properties of the mobile channel.

The research is conducted by research groups from various disciplines and is both theoretical and practical in nature including field trials.

2. VIDEO COMPRESSION

Current technical developments in the field of low bit rate video compression all incorporate some sort of segmentation-based coding. After decomposing an image sequence into spatio-temporal regions that are either relevant to the application or that are homogeneous in their spatio-temporal properties (texture, motion), the different regions are compressed. Irrespective of the actual compression techniques applied to the spatio-temporal regions, the region's shape information needs to be available to the video decoder. Especially the coding of object shapes may induce a significant amount of additional side information that is absent in conventional coding schemes. Much like

♦ This work is funded in part by the Netherlands' Foundation for Technical Sciences (STW) under the project DEL44.3461: Mobile Multimedia Communications: Wireless Access to the Information Superhighway.

the choices that can be made in transmitting motion information, there are three possibilities for compressing contour information:

- Intraframe coding of contour information, disregarding temporal redundancy in this representation;
- Predictive coding, such that only contour differences (shape prediction errors) have to be transmitted;
- Spatio-temporally recursive coding such that no shape information needs to be transmitted at all because the decoder can reproduce the required shape information from previously received information.

In principle the use of temporal information may further reduce the bit rate for the contour shape information, but the prediction of digital contours is not a trivial problem, requiring contour matching [3]. Within a multimedia setting, however, the possibility to directly decode an object's shape may constitute a marked advantage for manipulation of compressed visual information (e.g., editing of picture contents, automatic annotations) and browsing purposes [4]. For these reasons we concentrate on *lossy intraframe* compression of contours in this paper. We present a contour coding scheme that approximates object shapes using B-splines [5].

3. CONTOUR COMPRESSION

A digital contour is defined as an ordered set of 8-connected object pixels located on the border of an object. This implies that contours are always closed and not self-intersecting. To keep the contour coding approach as generic as possible, we do not make use of any specific knowledge about the segmentation algorithm that generated the object's mask.

The straightforward encoding of an 8-connected contour takes 3 bit per contour pixel (BPCP). Earlier work on contour coding indicated that efficient lossless coding can be achieved using conditional chain codes at approximately 1.5 BPCP [6,7]. Since in a compression environment a rate allocation will typically distribute the available bit rate over the various information types to be encoded (contours, texture, motion) [8], lossless encoding of the contours may not always be possible or desirable.

Lossy (approximating) contour coding algorithms should take into account the following requirements:

- The performance of the algorithm should not be dependent of the location and orientation of the contour, nor should the arbitrary starting point be of influence;
- The quality of the contour should be specified in terms of maximal deviation of the approximating contour from the original contour. The mean-squared error approximation of a given contour leads to artifacts especially near perceptually important features (corners). Asymmetry should be possible for approximation errors on the inside and outside of the object;
- The method should be flexible enough for a wide variety of contours encountered in coding "natural" objects but also objects known as incorrectly predicted or "model-failure" areas.

The lossy multiple grid chain codes contour coding technique proposed in [9] reaches approximately 1.1 BPCP for a maximum error of 1 pixel. The approximation of contours using straight line segments yields 1.6 BPCP for a maximum error of 1 pixel and 1.3 BPCP for a maximum error of 2 pixels. Within the MPEG-4 standardization, quadtree coding of contours is considered. This method, however, suffers strongly from position and orientation dependency.

4. B-SPLINES CONTOUR APPROXIMATION

Following [10], we approximate a discrete contour by a continuous two-dimensional polynomial $P(t)=[x(t),y(t)]$. The polynomials $x(t)$ and $y(t)$ are piecewise, uniform cubic B-splines, which are continuous everywhere in their first and second derivatives. A similar approximation methods was recently proposed in [8]. The total spline approximation is built up of s segments, requiring a total of s control points $(C_{i,x},C_{i,y})$ $(i=1,\ldots,s)$. Every segment shares three control points with neighboring segments, which eases local adaptations of a spline. The (x-component of the) i^{th} segment of the cubic B-spline is given by:

$$x(u)=\frac{1}{6}[C_{i-1}(1-u)^3+C_i\,(3u^3-6u^2+4)+ C_{i+1}(-3u^3+3u^2+3u+1)+C_{i+2}u^3]. \qquad (1)$$

Given the number of segments that we need to use to approximate a given contour, and given a relation between (a subset of) samples on the original contour and the values of the u parameter in (1), we can analytically fit a piecewise uniform B-spline onto the given contour by minimizing the mean-squared distance between original and approximating contour. This overdetermined problem requires the solving of:

$$\mathbf{C}=\left[\mathbf{Q}^t\mathbf{Q}\right]^{-1}\mathbf{Q}^t\mathbf{R} \qquad (2)$$

Here $\mathbf{Q}$ is an $m\times s$ matrix containing the spline basis (or blending) functions evaluated at m selected positions (determined by the u values) on the approximating curve. $\mathbf{R}$ is the vector containing the m selected samples on the original contour, and $\mathbf{C}$ is the vector with the required s control points. The control points are rounded to the nearest integer grid point and are transmitted to the decoder. Note that the approximating spline does not necessarily pass through points on the original contour (e.g., begin or end points of a segment).

5. ITERATIVE OPTIMIZATION

The set of control points is optimized such that the mean-squared distance between the original and approximating

B-spline is minimal. A typical spline approximation resulting from minimizing the MSE criterion is shown in Figure 1 (upper left). A more desirable approach, however, is to limit the maximal distance between original and approximating contour to, say, maximally 2 pixels on the outside of the object and 1 pixel on the inside. Since this criterion does not allow analytical optimization, iterative tuning of the following contour parameters is required:

- The number of segments in the spline;
- The location of the breakpoints of the spline;
- The selection of sample points on the original contour and the relation between these samples and a *u*-value.

A fairly robust initial guess can be made for the number of segments using the following estimate:

$$s = \min\left\{35\,,\,4 + \frac{\text{contour length}}{25}\right\} \qquad (3)$$

However, the optimization of the number and length of the spline segments is still of crucial importance in minimizing the bit rate and the maximum deviation between original and approximating contour.

We have developed an iterative optimization algorithm that first makes a rough guess of the B-spline approximation using (3) and the algorithm (2). Next, the following modifications of this result are considered in an iterative fashion until the maximum distance criterion is satisfied:

1. Usage of more samples on the original contour to influence the shape of the spline locally more strongly. In other words, where the deviation of the approximating contour is too large, a denser sampling of the original contour is used;
2. Splitting a segment into two new segments. This may be needed if the deviation between original and approximating contour remains too large;
3. Merging two neighboring segments or removing an intermediate segment. This data reduction operation can be carried out if the initial approximation is better than actually required.

Notice that the result of step 2 and 3 is that the approximating contour is no longer a uniform B-splines function.

After each modification, the new control points (and approximation error) have to be calculated by (2). In this way the contour approximation locally adapts itself to specific object characteristics, such as sharp bends and long straight lines. In Figure 1 we give an illustration of how the iterations progress on an highly irregularly shaped region. The final result also shows the location of the spline control points (black dots).

6. COMPRESSION OF CONTROL POINTS

The control points of the approximating contour need to be sent to the decoder. If a PCM representation is used, this takes two 8 bit code words per control point.

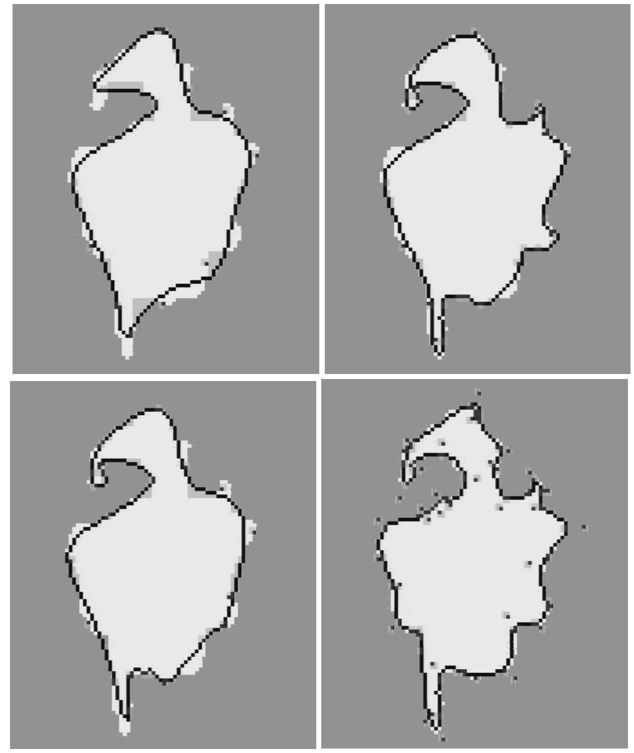

Figure 1: Illustration of approximating iterations for region with complex shape.

Since there is a significant correlation between the individual components of consecutive control points (correlation coefficient in the order of 0.85), the *x*-coordinates and *y*-coordinates are encoded differentially. The expected prediction gain is about 2 bit per control point. Figure 2 shows the histogram of the control point differences. For the objects that we considered, the resulting differences appear to be (two-sided) Laplacian distributed with λ=0.1 and a range of [-64,+64]. Entropy encoding was applied to the control point differences using Huffman variable length coding. The expected entropy is a total of about 11.2 bit per control point.

7. COMPRESSION RESULTS

The algorithm has been tested on various artificial regular and irregular test objects, and on model-failure areas generated by an Object-Based Codec (SIMOC). The results are fairly robust with respect to the various parameters of the algorithm, but all results indicate that the proposed iterative optimization of the number of contour segments is unavoidable. For the set of test objects we considered, an average of 0.82 BPCP was obtained for a maximum deviation of 1 pixel on the object's inside and 2 pixels on the outside. Table 1 lists the results for a few of the objects. It shows that a spread of about 10% in bit rate is found as a result varying parameters and the order in which the iterative tuning steps are carried out. The en-

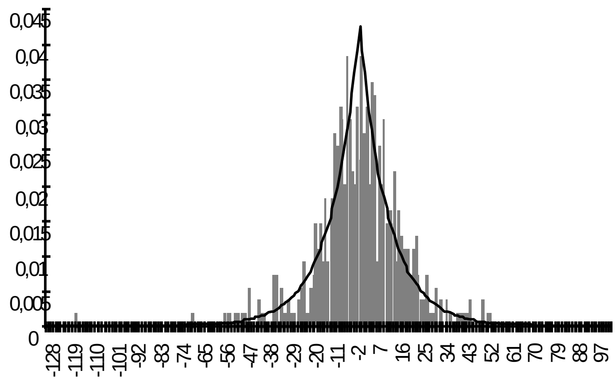

Figure 2: Histogram of control point differences.

tropy coding of the control point differences yields a compression of approximately 20%. If the maximum allowed deviation on the object's inside is increased to 2 pixels, the average goes down to 0.70 BPCP (Table 2), while for a maximum deviation of 1 pixel the average is 0.88 BPCP.

8. DISCUSSION

Our work indicates that even with a small tolerated deviation between original and approximating contour, an improvement of 40 percent on the average in bit rate can be achieved compared to the lossless conditional chain codes method. If we consider a typical low bit rate application as studied in the MMC project using QCIF/12.5 Hz format, in which for instance 5 objects need to be represented with average contour length of 400 pixels, then a total contour bit rate of 18 kbit/sec is obtained. More complex interframe coding of contours may be needed to further reduce this number.

ACKNOWLEDGMENT

The authors wish to thank Mr. E. de Boer from KPN Research, Leidschendam (The Netherlands) for his supervision of the M.Sc. project of Mr. Quist, on which this paper is based. Further, the authors wish to thank Dr. Van Beek, currently with University of Rochester, Rochester NY, who provided the conditional chain coding results.

REFERENCES

[1] IEEE Communications Magazine, February 1996 issue, vol 34, no 2, 1996.

[2] The Delft Mobile Multimedia Communications (MMC) project, http://mmc.et.tudelft.nl/mmc.

[3] B. Kamgar-Parsi, A. Margalit and A. Rosenfeld, "Matching general pulynomial arcs", *Image Understanding*, vol. 53, no 2, 1991, pp. 227-234.

[4] A. Nagasaka and Y. Tanaka, "Automaitic Video Indexing and Full-video Search for Object Appeareances", *Visual Database Systems II*, Elsevier 1992.

[5] C.P. Quist, *Intra-frame contour compression with cubic B-splines*, M.Sc. thesis Delft University of Technology and KPN Research, July 1995.

[6] C. Lu and J.G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representations", *IEEE Trans. on Communications*, vol. 39, no. 10, pp. 1511-1514, 1991.

[7] G.C. Gurski and M.T. Orchard, "On the use of orientation information for improved contour difference coding", *Proc. Int. Conference on Circuits and Systems*, pp. 563-566, 1993.

[8] K.W. Stuhlmuller, A. Salai and B. Girod, "Rate-constrained contour representation for region-based motion compensation", *Proc. of SPIE Visual Communication and Image Processing 1996*, pp.344-355, 1996.

[9] A. Gasull, F. Marques and J.A. Garcia, "Lossy image contour coding with multiple grid chain codes", *Proc. Workshop on image analysis and synthesis in image coding*, Berlin, Germany, 1994.

[10] F. Lu and E.E. Milios, "Optimal splines fitting to planar shape", *Signal Processing*, vol. 37, pp. 129-140, 1994.

Table 1: Evaluation of B-splines approximation for maximum deviation of 1 pixel on the object's inside and 2 pixels on the outside, and comparison with conditional chain codes.

Object	Conditional chain codes	Splines worst case (without/with VLC)	Splines best case (without/with VLC)	Splines typical case (without/with VLC)
"Block"	1.06	0.80 / 0.65	0.63 / 0.56	0.74 / 0.61
"Model failure area"	1.63	1.76 / 1.34	1.51 / 1.17	1.61 / 1.23
"Hand"	1.56	0.92 / 0.78	0.78 / 0.73	0.88 / 0.76
"Contour mother-daughter"	1.34	0.88 / 0.76	0.69 / 0.62	0.74 / 0.65
Average over 16 objects	*1.48*	*1.09 / 0.86*	*1.02 / 0.82*	*1.04 / 0.82*

Table 2: Evaluation of B-splines approximation for maximum deviation of 1 and 2 pixels on the object's inside and outside (typical results).

Object	Deviation of 1 pixel (without/with VLC)	Deviation of 2 pixels (without/with VLC)
"Block"	0.74 / 0.60	0.63 / 0.55
"Model failure area"	1.96 / 1.46	1.22 / 0.96
"Hand"	1.05 / 1.87	0.61 / 0.60
"Contour mother-daughter"	0.94 / 0.77	0.55 / 0.52
Average over 16 objects	*1.14 / 0.88*	*0.85 / 0.70*