LOW - BITRATE VIDEO CODING WITH THIRD ORDER GEOMETRIC TRANSFORMATIONS

Cornelis H. Slump, Marcel A.J.A. van Veen and Frederik J. de Bruijn
Dept. of Electrical Engineering, University of Twente,
PO Box 217, 7500 AE Enschede, The NETHERLANDS
Tel: +31 53 4892094; fax: +31 53 4891060
e-mail: c.h.slump@el.utwente.nl

ABSTRACT

This paper describes low-bitrate video compression based upon the characterization of the new frame as a set of geometric transformations of objects of the previous frame. Objects with motion are detected and the motion is estimated. The estimated motion (motion field) is used to obtain the parameters for the geometric transformations. The pertinent geometric transformations are rotation, translation, zooming and isotropic and anisotropic distortion. The motivation for choosing this set of third-order transformations is that we have at our disposal special ASICS for real-time video processing. We only want to transform moving objects and therefore the boundaries of the moving objects must be known. The boundaries of the objects are represented by closed contours.

1 INTRODUCTION

Digital video communication is a rapidly evolving field, especially because of the progress made in video coding techniques and *very-large-scale integration* (VLSI) technology. This progress leads to a high number of video applications, like *high-definition television* (HDTV), interactive visual communication, and the videophone. Of our special interest are the very-low-bitrate applications, like the videophone, videoconferencing and real-time video transmissions over ATM networks or multi media. These very-narrow-bandwidth applications normally have a bit-rate lower than 64 kbit/s and have sometimes even an upper limitation of 20 kbit/s. Compression ratios higher than a factor 1000 are required to achieve these very-low-bitrate conditions. The high compression ratios achieved by the current compression methods are accompanied by a low frame rate, modest video quality, a tiny viewing window and disturbing effects like blocking and blurring. There is enough work to be done to improve the quality of these very-low-bitrate applications, with respect of the low-bitrate aspect.

The H.320 standard has been developed between 1983 and 1990 for visual telephony over ISDN. The H.261 standard describes the compression of the video component of the audiovisual signal [2]. The two possible frame sizes in H.261 are 352×288 and 172×144, both at a frame rate of 29.97 Hz and with the possibility to skip up to 3 consecutive frames. Both sizes are standardized in the *Common Interface Format* (CIF) and the *Quarter CIF* (QCIF) video format respectively. The H.261 standard aims at a transmission at a rate of $p \times 64$ kbit/s, where $p = 1, 2, \ldots, 30$. With values of $p = 1$ or 2, the QCIF-frames are transmitted at about 10 frames per second. This mode is normally employed for video-telephony applications. With values of $p \geq 6$, the CIF format is used at 15 or more frames per second. This mode is intended for videoconference applications.

Between 1991 and 1996 the H.324 standard was developed to extend the use of visual telephony to conventional analog telephone lines with the use of the V.34 modem. The H.324 standard is one of the standards used in the MPEG-4 compression standard [4, 5]. The available transmission rate on an analog telephone line is 28.8 kb/s for both speech and video, which leave about 20 kb/s for video. The H.263 describes the compression of the video component and is designed for transmission at rates *below* 64 kbit/s. The encoded signal can be sub-QCIF (with a frame size of 128×94), QCIF, CIF or any larger input format with a minimum number of consecutive skipped frames. H.263 is capable of providing the same quality as H.261 with less than half the number of bits. H.263 uses second generation video coding techniques, which try to overcome the limitations of the first generation video coding techniques. Still, the block-wise processing of the frames as applied in the H.263 standard causes the block-border to become visible when the transmission capacity is insufficient.

This paper proposes an alternative approach to perform the necessary motion compensation. Existing compression methods compensate the motion by means of macroblock compensation. The disadvantage of this method is the blocking effect and the high number of required transformation parameters. The proposed *geometric object transformation* method compensates the motion of the total object. The advantage of this approach is the lower number of bits required to characterize the motion and there is no blocking artifact. The proposed approach also proceeds from the in-house made ASICS, capable of performing third-order object

transformations [1].

Our first objective is the implementation of the proposed approach to get a better understanding concerning obtained image quality and compression ratios.

2 VIDEO CODING BASED UPON GEOMETRIC OBJECT TRANSFORMATIONS

Figure 1 shows an image from a typical videophone scene.

Figure 1: *Frame 1068 from sequence "Anouk".*

Evidently, the shape of the objects in the image should be analyzed [6]. Both the HVS-properties as well as these particular image characteristics justify the development of a *contour-texture approach* for coding videophone sequences as an alternative to blockwise processing. To obtain a contour-texture representation, the image sequence has to be segmented. The result of the segmentation process gives a set of connected regions. Each region can be characterized by its shape and its evolution in the time domain. The evolution in time can be described as a geometric transformation. Both the outline of each region as well as its time progression needs to be transmitted, in addition to the interior, the *texture*, of the region. Because the regions rather stationar higher data compression ratios may be accomplished, while nonstationarities in the image are accounted for by a parameterized description of the temporal progression of the regions.

The approach to perform motion compensation by means of geometric transformations follows from rapid growth in technical possibilities. The hardware solution is a spatial image transformer (AELT512), capable of transforming images of 512×512 pixels at 25 or 30 frames per second. The design of this image transformer chip was initiated at the *Laboratory of Network Theory & VLSI-Design* at the University of Twente and has been further developed by the company *AEMICS* in Hengelo. This spatial image transformer performs a 2-dimensional 3rd-order transformation and, therefore, any motion which approximates a 3rd-order polygon description can be motion compensated. A 3rd-order transformation makes it possible for objects to perform primary operations like translation, rotation, zooming, skewing, but also more complex operations like 3-dimensional warping operations and can be expressed as follows:

$$\begin{aligned} \Delta x_i &= r_o^2(Dx_o - dy_o) \\ \Delta y_i &= r_o^2(Dy_o + dx_o) \end{aligned} \tag{1}$$

With x, y the coordinates of the transformed image and x_o, y_o the coordinates in the object plane, r_o is the distance in the motion compensated image from the optical axis. Motion compensation can be divided into two categories, isotropic and anisotropic transformation. In Figure 2 three kinds of transformation can be seen: pincushion, barrel (both isotropic) and pocket-handkerchief (anisotropic) motion.

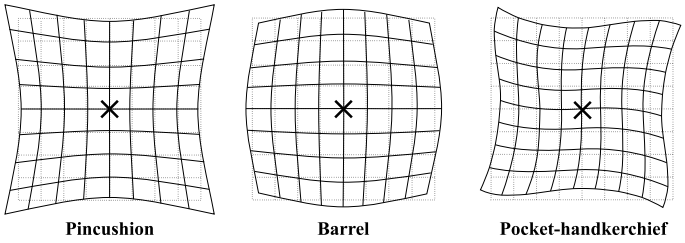

Figure 2: *Three kinds of distortion.*

We characterize motion transformation by the two dimensional third order polynomial Eq. 1. Correction can be made by the operator $\mathbf{T}$, see Figure 3 which is the inverse of this polynomial, also a two dimensional third order polynomial [1, 3]:

$$\begin{aligned} x(u,v) \quad = \quad & (a_x + e_x v + i_x v^2 + m_x v^3) + \\ & (b_x + f_x v + j_x v^2 + n_x v^3)u + \\ & (c_x + g_x v + k_x v^2 + o_x v^3)u^2 + \\ & (d_x + h_x v + l_x v^2 + p_x v^3)u^3 \end{aligned} \tag{2}$$

and a similar equation for $y(u,v)$.

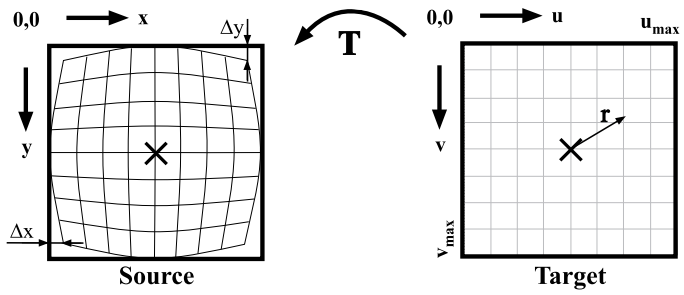

Figure 3: *Image transformation.*

Figure 3 shows the reverse address computation procedure. The target image is the corrected output image, the source image is the acquired motion-distorted image. For every pixel and thus u, v coordinate in the target image, the corresponding coordinates in the input image are computed. The source image is characterized by a matrix of numbers representing the pixel intensity in the center of the pixel, the so-called gridpoint. The pixel intensities of computed coordinates in the source image not identical to gridpoints are interpolated by cubic spline interpolation.

The transformation viz. Equation 2 is easily executed off-line by a computer but we have also implemented it in hardware for real time applications by forward difference accumulation [1]. The source coordinates x and y are expressed by a 16 terms equation as a third order polynomial of the target address. Each of the 16 terms is given by the partial derivative:

$$\frac{\delta^{k+l}}{\delta u^k \delta v^l} \quad k,l = 0,\cdots,3 \tag{3}$$

2.1 The Encoder

The Encoder is displayed in Figure 5. The encoder generates an image sequence of *intra-frame* encoded images (I) with a sub-sequence of consecutive *predictive frames* (P) in between, which can be depicted as IPP ... PIPP ... P. *Block 1*, determines the geometric-transformation parameters of the objects which must be motion compensated, also the contours are defined for all of the moving objects. Contour extraction is performed using the *Four-Triangle Method* [7]. The obtained contours are encoded using the *Freeman Chain Code* [7, 8]. The obtained regulated optical flow field, shown in Figure 4, matches the geometric transformation of the object.

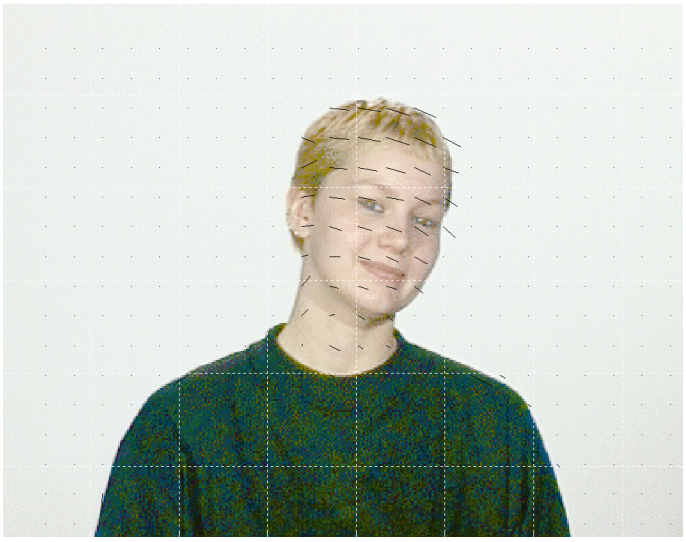

Figure 4: *Example of computed optical flow.*

The transformation parameters are obtained using a 2-dimensional least-squares fit of a 2-dimensional 3rd-order polynomial. The result of block 1 is a set of geometric transformation parameters and contour descriptions of the spatial areas associated with the moving objects. This set of transformation parameters are passed on to block number 2.

Block 2, creates the reconstructed P-frame $N+1$, to check if the reconstructed (motion compensated) frame satisfies a predefined error criterion.

Block 3 checks the error between the reconstructed P-frame $N+1$ and the original frame $N+1$. Above a certain error threshold, supplementary data is DCT-encoded, quantized and transmitted.

2.2 The Decoder

The proposed concept for the Decoder is displayed in Figure 6. The bit-stream from the Encoder to Decoder can contain I-frame data or else data for reconstruction of the P-frames. The data for P-frame reconstruction are both the quantized transformation parameters and the contour information. These two parts are withdrawn from the bit stream in *block 1*. *Block 2* receives the extracted transformation-parameter information and contour information from block 1. The geometric transformations are performed for all moving objects, this with use of the contours as boundaries. The transformations are performed on the previous I- or P-frame. The transformed objects are detached from the transformation region by making use of a polygon filling algorithm and the created gaps are filled in. The complexity of the Decoder seems much lower than the Encoder, but that is not quite true. The complexity difference lies mainly in the extraction of the contours and the extraction of the transformation parameters. The P-frame reconstruction is the same in the Encoder and the Decoder.

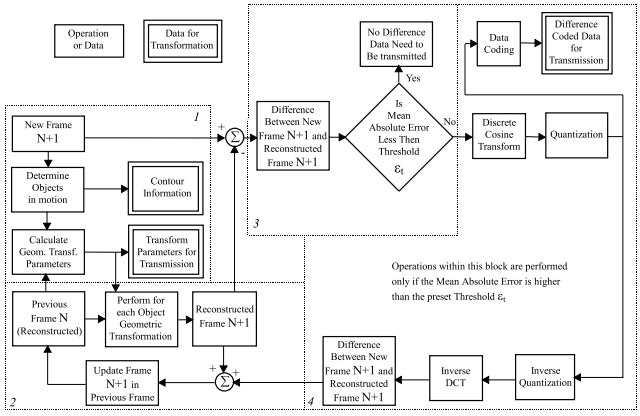

Figure 5: *Encoder Block Diagram.*

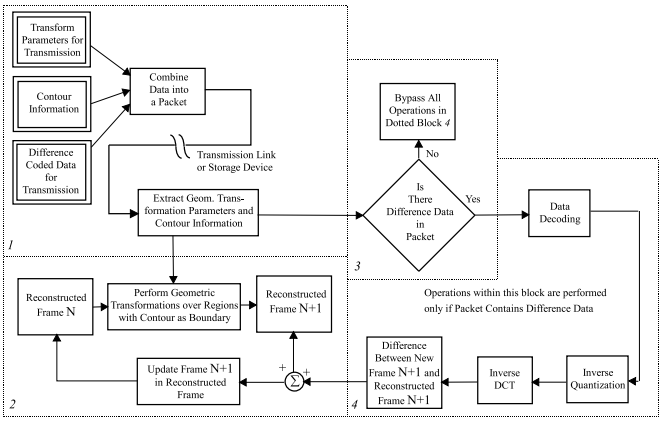

Figure 6: *Decoder Block Diagram.*

3 RESULTS

The software algorithm was applied to sequence of 368 colour frames of 288×384×24 bits (Quarter PAL), with a total duration of 14.32 seconds. The sequence was encoded according to the pattern IPPPPPIPPPPP, resulting in 61 I-frames and 307 P-frames. Of the 307 P-frames 363 different objects were motion compensated and 53 frames contained no significant motion. The average contour length is 283 connected coordinates.

A 3rd-order 2-dimensional transformation requires 16 parameters, which require 12 bits each. The Freeman-coded contour elements require 0.5 bits per element on average. The initial point of the chain code requires 17 bits. To match the results of H.263, the assumption is made that one out of three frames is displayed (i.e., a frame rate of 8.33 Hz).

The average compression ratio R can be determined as

$$R = \frac{B}{(IP + CL \times CP + TP \times PAR) \times FR \times AMC}$$

with

B	=	number of bits/frame	=	$288 \times 384 \times 24$
FR	=	frame ratio	=	1/3
IP	=	initial point	=	17 bits
CL	=	av. contour length	=	283 bits
CP	=	av. nr. of bits/contour elem.	=	0.5 bits
TP	=	nr. of transf. param.	=	16
PAR	=	number of bits per param.	=	12 bits
AMC	=	av. nr. of objs. transformed	=	363/307 = 1.182

where $CL \times CP$ is the average number of bits to describe a contour, and $TP \times PAR$ is the number of bits needed to describe a transformation. The resulting compression ratio of the observed sequence is $R = 19213.21$. The compressed sequence is virtually indistinguishable from the original.

4 DISCUSSION AND CONCLUSIONS

An alternative method has been developed for very-low-bitrate compression methods. The motion compensation is performed by geometric object transformations instead of macroblock translation. A software implementation of the geometric object transformation method is developed, providing information concerning reconstruction quality and compression ratio. The approach indicates very promising results:

1. The possibility to perform the object transformations by a 2-Dimensional 3rd order transformation.
2. Objects can be identified by their contours and their optical flow.
3. In comparison to other very-low-bitrate compression methods no effect like blocking and a reduced blurring effect.
4. A very high compression ratio (19200) for the reconstructed frames, this in comparison to a very-low-bitrate compression standard H.263 (compression ratio between 6000-8000).

Very-low-bitrate compression methods are only intended for sequences with relative low number of objects in motion, no quickly appearing and disappearing of objects, no frequent sequence changes and a stationary camera position. The same conditions hold for the geometric object transformation method.

Real-time implementation of this approach seems to be technically possible and is currently in progress.

References

[1] A.G.J. Nijmeijer, M.A. Boer, C.H. Slump, M.M. Samsom, M.J. Bentum, G.J. Laanstra, H. Snijders, J. Smit and O.E. Herrmann, "Correction of Lens-Distortion for Real-Time Image Processing Systems," in *Proceedings of the 1993 IEEE Workshop on VLSI Signal Processing*, vol. 6, Veldhoven, The Netherlands, 1993.

[2] *ITU-T Recommendations H.261: Video Coding for Audiovisual Services at p×64 kbit/s*, Rev. 2, 1993.

[3] M.J. Bentum, *Interactive Visualization of Volume Data*, Ph.D. Thesis, University of Twente, Enschede, The Netherlands, 1995.

[4] *Draft ITU-T Recommendation H.263: Video Coding for Narrow Telecommunication Channels at < 64 kbit/s*, 1995.

[5] L. Torres and M. Kunt, *Video Coding: The second Generation Approach*, Kluwer Academic Publishers, Boston, 1996.

[6] P.J. van Otterloo, *A Contour-Oriented Approach to Digital Shape Analysis*, PhD-thesis, Delft University of Technology, 1988.

[7] M. Eden and M. Kocher, "On the Performance of a Contour Coding Algorithm in the Context of Image Coding – Part I: Contour Segment Coding," *Signal Processing*, pp. 121–178, 1986.

[8] F. Freeman, " On the Encoding of Arbitrary Geometric Configurations," *IEEE Trans. Elec. Computers*, vol. 10, pp. 234–259, 1961.