A STUDY ON CONTENT-ORIENTED CODING SCHEME AND DECODER DOWNLOADABLE SYSTEM

Ouji NAKAGAMI†, *Naoto SHIMIZU*†, *Toshinori MIYAZAWA*‡,
Wataru KAMEYAMA†, Hiroshi WATANABE†, Hideyoshi TOMINAGA†,‡

†Waseda Univ.
Graduate School of GITS
1-3-10, Nishiwaseda, Shinjuku-ku, Tokyo, Japan

‡Waseda Univ.
Dept. of Elec., Info. and Comm. Eng.
3-4-1, Okubo, Shinjuku-ku, Tokyo, Japan

1. INTRODUCTION

Content-oriented coding is the one that takes the characteristics of the image content into account. Natural and synthetic images have different characteristics. However, JPEG/ MPEG employs DCT regardless of the image content. Thus, their coding schemes are not optimized for synthetic images; such as animation images and computer graphics. As a promising application of the content-oriented coding, we have been studying animation image coding that is mainly based on line and region extraction.

Currently, a fixed coding algorithm has been used for all kind of images in the broadcasting application. An approach to switch coding algorithm for image contents has a chance to provide better coding efficiency as a whole than the conventional one. Thus, it is important to establish a certain mechanism distributing different decoders depending on the contents for real time applications.

In this paper, we propose a scheme to improve coding efficiency of animation images and mechanism to download necessary decoders in real time.

Keywords: Image coding, Animation image, Smoothing operation, Decoder downloadable, Multimedia system, Middleware

2. ANIMATION IMAGE CODING

2.1. Coding scheme

Animation images have many impulsive changes in the luminance signal around line drawings. At the same time, many homogeneous color regions also exist. When signals of an animation image are transformed to frequency domain using DCT, the energy of DCT coefficients disperses from low to high frequency. In order to achieve high compression ratio, many coefficients are regarded as zero by quantization. Thus, undesired noises often appear arround line drawings by DCT-based image coding.

Therefore, we have been proposing a coding scheme specifically designed for animation images. In the animation image coding method [1], line drawings and homogeneous color regions are extracted from an image. Then, all lines and contours of homogeneous color regions are approximated by some significant points. An advantage of this coding scheme is to achieve high compression ratio by allowing some distortion of lines and edges. Computational complexity required for decoding process can be decreased since few points are enough to represent lines. Furthermore, size of decoded images can be changed in any resolution by the same number of significant points.

However, the convensional method cannot encode animation images having complicate background regions efficiently. In addition, high approximation accuracy to the contours is required to reduce the encoding error. Thus, more robust coding scheme to overcome these problems is proposed in the following.

2.2. Vector-DCT Hybrid Coding

In the animation image coding [1], the rest of line drawings and homogeneous color regions was coded by DCT since background regions are close to natural images. A homogeneous color region is detected by Region-growing algorithm [2]. However, it fails in some cases because of the mixture of the homogeneous regions and the background. Thus, it is necessary to improve such extraction process.

We propose a smoothing operator and region-growing algorithm jointly for this purpose. A flowchart of the proposed animation image coding is shown in Fig.1. Animation images are coded in two layers; Base layer and additional layer. Base layer is constructed from the significant points extracted from line image and region image. Connected fil-

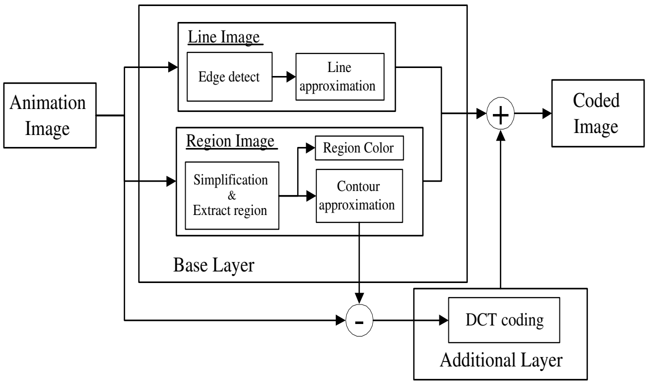


Figure 1: Flowchart of animation image coding

ter [3], which is used in this method, is one of smoothing filters for simplification of images. Since this filter does not remove high frequency components unlike linear low-pass filters, the sharpness of edges in the filtered image is kept. Then, region-growing algorithm can extract homogeneous color regions from whole part of the image.

At this stage, the animation image is represented by only lines and homogeneous color regions, and the lines and the contours of the regions are approximated by significant points.

A compensation method for the result of smoothing and approximation operation is needed as the additional layer to acquire a high quality coded image.

First, the difference between each approximated homogeneous color region and an original image is calculated. It is divided into 8×8 pixels blocks and coded by DCT. Some DCT blocks include edges when they locate on the contours of homogeneous color regions. For avoiding mosquito noise caused by DCT coding, dummy data is padded to the pixels. The sum of the difference value in the each blocks is calculated by Eq.(1)

$$D(i,j) = P_o(i,j) - P_h(i,j) \tag{1}$$

where, $P_o(i,j)$ and $P_h(i,j)$ is the value of the original image and extracted homoegneous color region. In Eq.(2), D_s is the sum of $D(i,j)$ of the block.

$$D_s = \sum_{i=0}^{7}\sum_{j=0}^{7} D(i,j) \tag{2}$$

At the block on the background regions, D is bigger than the one on the homogeneous color regions because of the smoothing operation. Therefore, it is suitable to prioritize the block in order of D value in encoding.

Fig.2 shows the effect of the additional layer. In the conventional method, the regions that cannot be extracted as homogeneous color regions or line drawings were encoded by DCT. The homogeneous color regions detected in failure caused much error. On the contrary, the proposed method enables to control the number of coding DCT blocks. The result of Fig.2 shows that the necessity of the additional layer when animation images include complicated background regions.

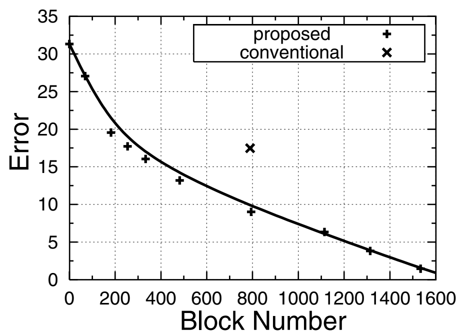


Figure 2: The effect of additional layer

2.3. Dynamic Programming with feedback algorithm

According to region-based image coding [3][4], all parts of an image should be regarded as the set of homogeneous color regions. This approach is more suitable for animation image coding. In animation images, homogeneous color region is usually larger than that of natural images. Therefore, its contours are efficiently vectorized by extracting the significant points.

Dynamic Programming (DP) algorithm is effective for fitting various shapes to contours of the regions [5]. Ramer algorithm [6] has been used in the conventional approach. It was developed for straight-line approximation. Thus, significant points determined by the Ramer algorithm are not always proper for spline curve function.

Dynamic Programming can examine all points on the curve since they may be chosen as significant points. It can provide the set of significant points that minimize the total error between original and approximated line. Although this method requires high computational complexity, the optimized approximation can be obtained.

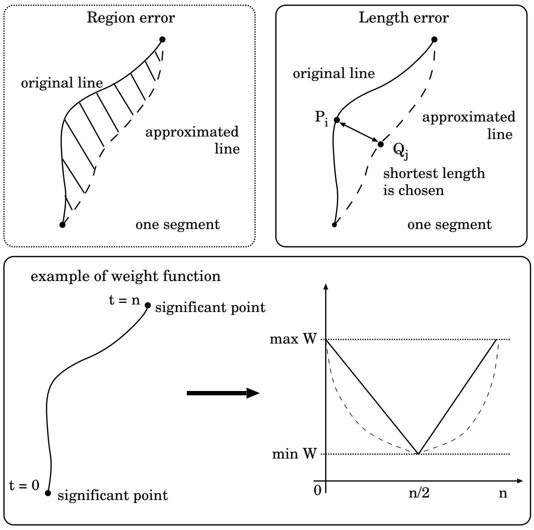


Figure 3: estimation of the approximation error

There are two ways of estimating the total error in the conventional study; region and length error algorithm. They are shown in Fig.3. The former estimates an error area between an original line and an approximation line, and the latter calculates a sum of a length from points on an original line to the nearest points on an approximation line. The latter can add weight by

$$E = \frac{1}{l} \sum_{i=1} w_i |P_i - Q_i|^2 \qquad (3)$$

where, E is an approximation error, l is the number of contour points, and w_i is the functional weight about point i on an original line. P_i are the coordinates of point i, and Q_j are the coordinates of point j, which is the nearest from point i. The set of significant points is calculated by the following steps.

1. choose several points as significant ones at high curvature points on original line.
2. divide an original line into some segments between significant points. The value of E is estimated by normalization of the length. Threfore, if the segment length is too longer, DP algorithm can not acquire a precise approximation result.
3. check the value of E for each segment. If larger the T_e of the error, try other set of significant points on the segment.
4. determine the points which minimize E of each segments as the significant points.

Since DP algorithm requires the error T_e to determine the best set of significant points, we should set suitable T_e for each contours. If T_e is too small, many determined significant points concentrate at high curvature points on an original line. On the other hand, the large T_e increases the approximation error. Therefore, we propose feedback algorithm to eliminate the useless significant points. First, sufficiently small T_e is set, and DP acquires enough significant points. Next, DP is applied to eliminate significant points that do not reduce approximation error. Fig.4 shows the result of DP algorithm with and without feedback algorithm. Fig.4(a) illustrates the relation between the number of significant points and approximation error. DP with feedback algorithm enables more precise approximation in spite of the small number of significant points than the conventional method (Fig.4(b)).

3. DECODER DOWNLOADABLE SYSTEM

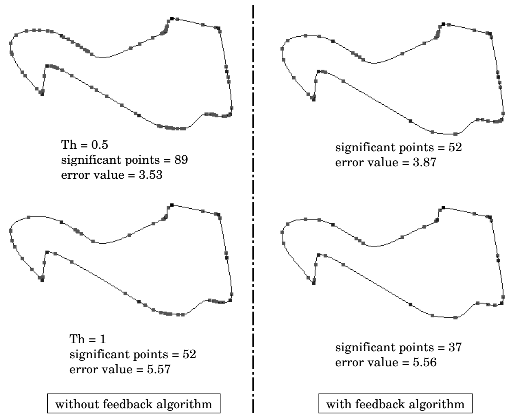
3.1. Requirements

Decoder downloadable system is a system architecture to distribute various decoders. It is necessary not only to develop image coding algorithms but also to prevail decoders implementing them.

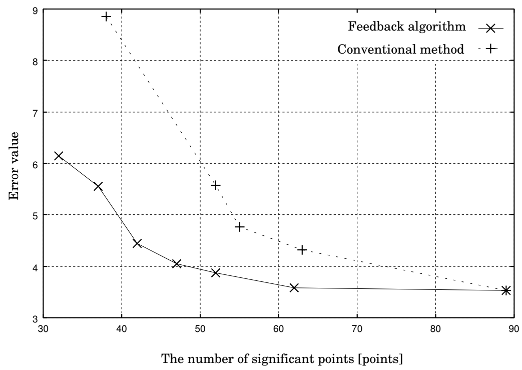
Multimedia systems on the Internet, such as Microsoft Media Player, RealPlayer and so on, provide functions to play back new types of media by installing appropriate decoder software, which is called "Plug-in," via network when required. However, Plug-in has problems as follows:

1. Time delay

   The time delay happens when a new content starts and a client does not have the suitable decoder. This interrupts seamless playback of contents and becomes a serious problem especially in case of TV broadcasting.



(a) approximation of a contour by DP algorithm



(b) approximation error

Figure 4: the result of DP with feedback algorithm

2. Installation problem

   Plug-in requires viewers of contents a little knowledge about image coding. TV broadcasting has been universal service, and Internet streaming will be the same situation, too. The system in the universal service should not expect clients having service installation knowledge. This point is very important for applying Content-oriented coding scheme, because it will require more knowledge about image coding.

3. Closed specification

   The specification of Plug-in and the way to transfer Plug-in are closed in each multimedia software vendor. There is no compatibility among the software vendors. Developers of multimedia system are urged to create the architecture to download the decoders.

In case, we will apply Conent-oriented coding to not only Interenet streaming but also TV broadcasting, there is no integrated system to distribute them for both TV broadcasting and Interenet streaming. We implement Decoder downloadable system to achieve such a system and to provide developers of multimedia system the functions to download the decoders.

3.2. Proposed system architecture

In this section, we propose the system architecture which enables to download decoder dynamically and to play back without delay before a new content starts.

The proposed client's system architecture is shown in Fig.5.

On a server side, there are Content server, Decoder server and Information server. Content server stores actual bitstreams of the contents. Decoder server stores the binary codes of the decoders. Information server stores the information shown in Fig.6. Clients can seek where the suitable decoders for each content is stored. It is done by exchanging this information between clients and Information server, and clients can download decoders dynamically. We implement the client system by using Java. This is because the proposed system should work in the various environments.

We develop the part of media processing with JMF (Java Media Framework) [7] as the first step. A question is often raised about the processing performance of Java. Therefore, we designed the system in the such a way that the module for media processing depending on JMF can be detached from other parts entirely.

In this implementation, CORBA (Common Object Request Broker) [8] is used to define its interface between server and client being independent from the hardware specification, because of the independency of CORBA GIOP (General Inter-ORB Protocol) from any transport protocol.

The procedure in the system is executed in order the following:

(1) The scheduler in the client system starts to parse the information shown in Fig.6. The system confirms whether the decoder exists in the client's side or not.

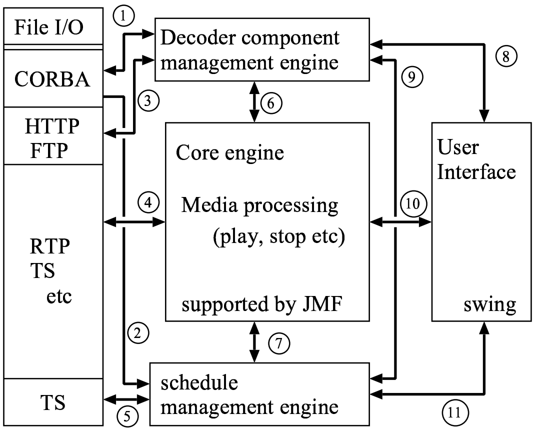


Figure 5: Decoder downloadable system architecture

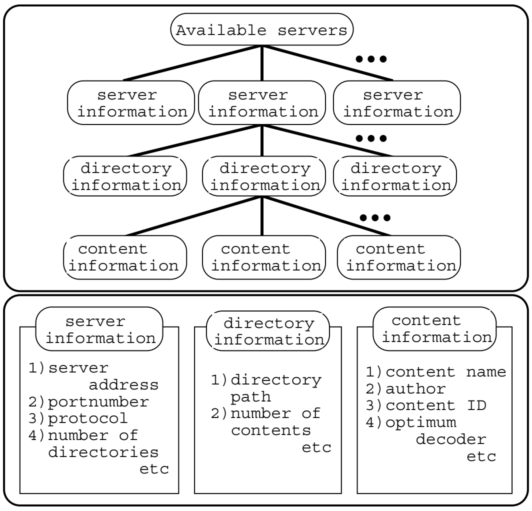


Figure 6: Content information

(2) Sending the information about the content that the client does not have a suitable decoder to Information server.

(3) Information server starts to seek where the suitable decoder is stored using the information sent in (2). It returns this seeking information to the client.

(4) Decoder information has the IP address of Decoder server, the filename of the decoder and so forth. The scheduler of the client decides when to start downloading the decoder from Decoder server by that information.

The role of each part in the system is as follows:

- Core engine

  This part provides the implementation of media play, the support of trick mode and so forth. The decoders are actually installed in this part.

- Decoder component management engine

  This part manages the decoders in the client's system, queries about what decoder is needed, download the decoders, deletes the decoders in the stock.

- Schedule management engine

  This part controls when to start media play, downloading the decoders, deleting the decoders. It indicates to each part that provides actual above functions.

- User interface

  This part is the interface for user input.

- File I/O

  This part selects each protocol in accordance with the purposes. 1) CORBA : sending and receiving Content Information, 2) HTTP or FTP : downloading the binary data of the decoders, 3) RTP (Real-time Transport Protocol) or TS (Mpeg-2 Transport Stream) : receiving media data

Each interface in the system is described as follows:

interface 1: Receiving the information about the decoders, sending the information of contents.

interface 2: Receiving the information about the contents.

interface 3: Notification of decoder information, download binary data of decoders.

interface 4: Download media data of contents.

interface 5: Download EPG (Electronic Program Guide) data that is used by the client's scheduler.

interface 6: Confirmation for existence of decoders, provision of decoder modules to Core Engine.

interface 7: Media play, indication when decoder's memory load starts, providing the information about what state the memory load is.

interface 8: Indication whether decoder is added or deleted to the system by the users.

interface 9: Providing the information about the contents, indication when downloading the decoder and deleting the decoder in the client system start, confirmation for existence of decoders.

interface 10: Acceptance of the input information by the users.

interface 11: Indicating the information of contents that can be played back, the choice of the contents by the users.

The ability of scheduling for downloading decoders in this system is evaluated in case of Pull Service [9]. The result shows the optimum number of downloading decoders that achieves the minimum time delay is given by the probability that the user switch the content, such as channel hopping. However, this probability may obey a specific distribution. We take an approach to leave place for implementing the function that this distribution is given by storing the history of changing the contents to the scheduler. Furthermore we will implement the remaining part of this system for Push Service, and evaluate it.

4. CONCLUSION

We proposed the animation image coding and the Decoder downloadable system which puts this coding scheme to practical use.

Animation image coding adopts two-layer coding scheme. On the base layer, an animation image can be encoded at low bit rate since line drawings and contours of each homogeneous color region is expressed by few significant points approximating the contour line. DP with feedback algorithm enables precise approximation for the contours by the controllable number of significant points. The lost data caused by smoothing operation is compensated by any number of DCT blocks on additional layer. This enables smaller coding error than the conventional method. Our proposed coding scheme is only designed for still images. Thus, this scheme can be used for the Intra frame coding of video coding.

Decoder downloadable system is composed of Content, Decoder and Information server. This system enables dynamic decoder downloading from Decoder server by negotiation between clients and Information server. Therefore, clients can decode the contents without the knowledge about image coding scheme and can play back media contents seamlessly. This system, developed by Java and CORBA, is superior for the reason of portability. It will be applied to TV broadcasting as well as Internet streaming.

5. REFERENCES

[1] T. Miyazawa, W. Kameyama, H. Watanabe, T. Sakatani and H. Tominaga, "Animation Image Coding," *World Multiconference on Systemics, Cybernetics and Informatics (SCI) 2001, Proc. Vol.XIII*, pp.404-409, July 2001.

[2] R. Adams, L. Bischof, "Seeded Region Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6), pp 641-647, June 1994.

[3] P. Salembier and J. Serra , "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," *IEEE Transactions on image processing*, Vol.4, No.8, pp.1153-1160, Aug. 1995.

[4] C. A. Christopoulos, W. Philips, A. N. Skodras, J. Cornelis, "Segmented image coding: Techniques and experimental results," *Signal Processing:Image Communication*, pp.63-80, Nov. 1997.

[5] F. Lu, E. E. Milios, "Optimal Spline Fitting to Planar Shape," *Signal Processing*, vol.37 pp.129-140, 1994.

[6] U. Ramer, "An iterative procedure for the polygonal approximation of planar curves, " *Computer Graphics and Image Processing Vol.1*, pp.244-256, 1972.

[7] Sun Microsystems, "JavaTM Media Framework API," http://java.sun.com/products/java-media/jmf/

[8] Object Management Group, "CORBA/IIOP Specifications," http://www.omg.org/technology/documents/

[9] T. Miyazawa, N. Shimizu, W. Kameyama, H. Watanabe, H. Tominage, "A study on Architecture of Decoder Downloadable System for pull service," *PCSJ2001*, pp.27-28, Nov. 2001.