ANIMATION IMAGE CODING

Toshinori MIYAZAWA
Waseda University Dept. of Electronics, Information and Communication Engineering
3–4–1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555 JAPAN

Wataru KAMEYAMA,
Hiroshi WATANABE
Global Information and Telecommunication Institute, Waseda University
29-7 Bldg., 1–3–10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051 JAPAN

Toru SAKATANI
NTT Cyberspace Labs.
1–1 Hikari–no–oka Yokosuka-shi, Kanagawa 239–0847 JAPAN

and

Hideyoshi TOMINAGA
Waseda University Dept. of Electronics, Information and Communication Engineering
3–4–1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555 JAPAN

ABSTRACT

In this paper, we propose a new coding scheme specifically designed for animation images. This is one of edge-based coding schemes adapted to the signal characteristics of animation images. Images are decomposed into three components, these are lines, regions surrounded by closed contours and background regions. Each component is encoded separately depending on its signal characteristics. Images reconstructed in the proposed scheme can preserve edges even at high compression ratio in comparison with waveform coding. In addition to introducing the basic scheme, we propose three refining algorithms to improve image quality. Experimental results show that the proposed coding scheme can provide good performance to animation images.

Keywords: Image coding, animation image, spline curve, line-drawing, homogeneous color region, background region

1. INTRODUCTION

Discrete Cosine Transform (DCT) based image coding [1], such as JPEG [2] and MPEG [3], has been widely used for still and video coding when the targets are natural images. However, DCT-based image coding may not provide good picture quality for synthesized images, such as animation images and Computer Graphics (CG), since their signal characteristics are quite different from natural images. Luminance signals of animation images contain more impulsive changes around edges than the signals of natural images. When animation images are compressed by DCT, energy of DCT coefficient disperses from low to high frequency component. This phenomenon makes coding efficiency worse at DCT-based image coding. In most cases, it causes undesired noise, called mosquito noise. It appears near edges when the DCT coefficients in high frequency component are roughly quantized.

Edge-based coding [4, 5] can provide solutions to the problem of DCT-based image coding. The scheme described in [4] focuses on grey level images. Therefore, some extensions may be necessary to cope with colors. The scheme described in [5] decomposes images into three components: strong edges, texture, and smooth components. Locations of the strong edges are encoded by chain coding, and texture and smooth components are encoded by DCT. Different from the previous work [5], our approach uses a representative color and a contour.

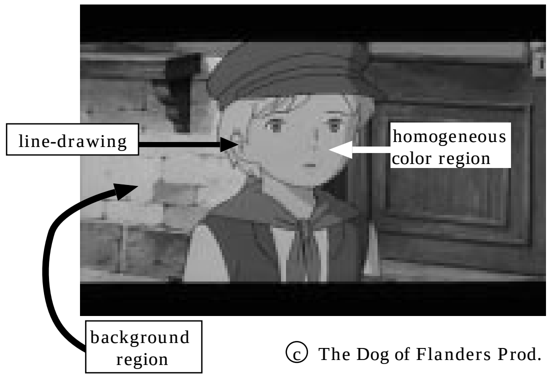

Figure 1: Three Components of An Animation Image

We propose a new coding scheme specifically designed for animation images. Our proposed scheme follows the concept of the schemes described in [4] and [5]. The differences between our proposed scheme and two conventional schemes are described as follows. We define an animation image comprises of three components:(1) line-drawings, (2) homogeneous color regions and (3) the remaining background regions (Fig.1).

Lines, which an animation cartoonist creates (hereafter called "line-drawings"), have usually dark colors and are drawn as the contour of homogeneous color regions. Homogeneous color regions can be defined as an area where pixel values of luminance and chrominance are almost identical (Fig.1). Background regions of animation are similar to natural images since they usually have many colors and details without obvious edges. These regions are counterparts of texture used in [5].

Our proposed animation coding scheme has a possibility to provide better coding performance than a DCT-based approach [6]. However, there are still three major issues to be solved First, anti-aliasing processing on edges causes difficulty in the extraction of homogeneous color regions. Secondly, approximation of straight lines or short lines needs a large amount of data and causes more noises when they are approximated by spline curves. Finally, there is not an optimum method for approximating line-drawings. We describe proposed animation image coding scheme in Section 2. We propose three algorithms to solve the three major problems: (1) clockwise searching mask algorithm described in Section 3, (2) hybrid coding algorithm by straight lines and spline curves described in Section 4, (3) Utilization of Ramer algorithm described in Section 5 and followed by conclusion.

2. ANIMATION IMAGE CODING SCHEME

We proposed animation image-coding scheme [6]. The three components, which are line-drawings, homogeneous color regions and background regions, are encoded in a different manner (Fig.2).

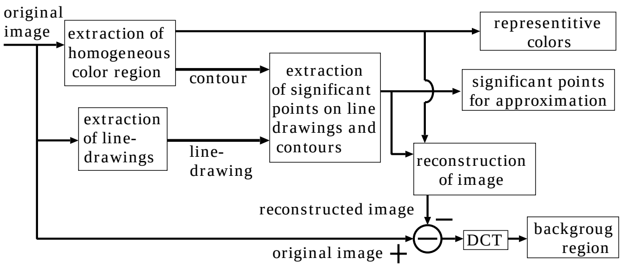

Figure 2: Animation image coding scheme

We extract line-drawings from the original animation image by using combinations of spatial filters. We identify homogeneous color regions in the original animation image, and then, obtain the representative color and contours of them. Detected lines are line-drawings and contours of homogeneous color regions. Significant points along lines are extracted and used to approximate the original lines by spline curves. Location of the significant points can be encoded by variable length code (VLC) or fixed length code (FLC). We construct a partially coded image only consists of spline curves and representative colors. We derive a background region by subtracting the partially coded image from the original image. This background region can be regarded as a similar type of natural image suitable for DCT-based image coding.

3. EXTRACTION OF HOMOGENEOUS COLOR REGION CONSIDERING ANTI-ALIASING

3.1. Influence of anti-aliasing

Anti-aliasing processing gets rid of jaggy effects on the line-drawings. Owing to this processing in digitized animation images, homogeneous color pixels near line-drawings may be modified to different colors. Thus, the extraction of the homogeneous color regions often fails around edges because of such color-changed pixels (Fig.3).

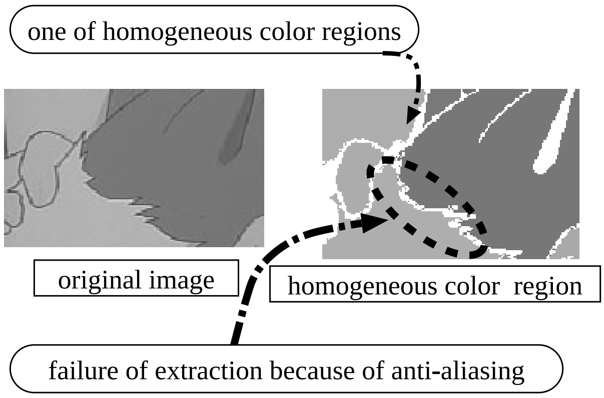

Figure 3: Influence of anti-aliasing on edges

The failure of the extraction causes two problems. One is double contours shown in Fig.4. They are mistakenly detected near edges and cause an erroneous coded image. The other is failure of background extraction. This prevents from achieving high compression.

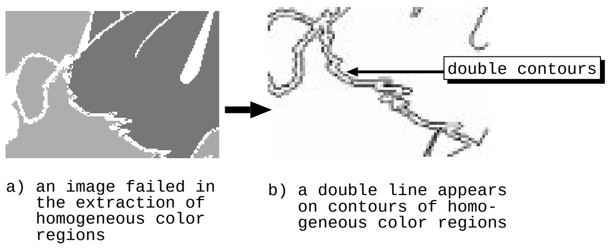

Figure 4: An example of double contours

3.2. Clockwise searching mask

We extrapolate the area of homogeneous color regions towards lines. To detect homogeneous color regions correctly, we focus on two points. (1) It is difficult to extract homogeneous colors only by values of pixels in luminance and chrominance signals. The location of pixels on line-drawings and color boundaries should be considered as the destinations for extrapolation of such homogeneous color regions. (2) There can be multiple homogeneous color regions around a line-drawing. Thus, it is necessary to use a mask whose center locates on a line-drawing and count each weight of homogeneous color pixels in the mask. In this case, the color that gives the maximum weight is chosen in the extrapolation.

We propose to use a clockwise searching mask (Fig.5). This mask is used to search target pixels that have a

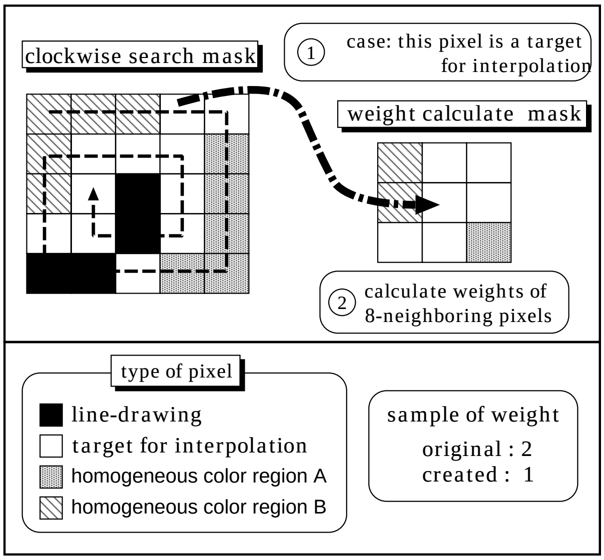

Figure 5: Extrapolation Of Homogeneous Color Regions

similar color to homogeneous color region's by tracing along a line drawing. If the target pixel is found, the number of homogeneous color pixels in eight neighboring pixels is counted and the color that has the largest weight is chosen.

3.3. Result

Fig.6 illustrates the comparison of the results of the conventional method and the proposed method [6]. The proposed method enables the more precise extraction of homogeneous color regions.

4. HYBRID CODING BY STRAIGHT LINES AND SPLINE CURVES

4.1. Problem of the spline approximation

It is the simplest way to approximate all lines by spline curves. However, there are three types of lines that are difficult to be encoded by the simplest way. They are (1) straight lines, (2) short lines close to 4 pixel length, and (3) lines that do not exists on contours of homogeneous color regions. Straight lines should be approximated by the location of two ends since these lines cannot be approximated by spline curves. Significant points on

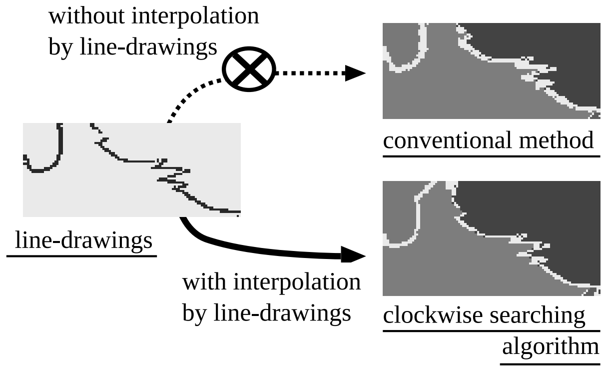

Figure 6: Result of Extrapolation of Homogeneous Color Regions

short lines should be sampled and each line between two adjacent significant points should be approximated by a straight line. Because such line requires at least 4 points for spline curve approximation. Isolated lines should be treated as independent one from such contours.

4.2. Hybrid coding by straight lines and spline curves

In order to classify three types of these lines, we employ the following four steps. First, representative lines of line-drawings should be chosen by thinning line-drawings because line-drawings have their widths. Next, each line is traced until an intersection appears. If an intersection is found, this segment is regarded as a line. Then, each line is traced to check whether it is straight or not. If straight lines are found, each line is labeled as a "straight line". Finally, all lines are sorted in the length and labeled as "straight" or "spline". Each line is encoded by the most optimal way as follows: (1) straight lines are encoded by the location of two ends, (2) short lines are approximated by straight lines and encoded by sampled points, and (3) curves, longer than some threshold length, are approximated by spline and encoded by significant points that are used for spline curve approximation.

We explain details of hybrid coding using Fig.7. Line A and line B overlap each other at the acute angle (Fig.7(a)). By line thinning, the overlapped part of the line-drawings becomes a line and an intersection appears on it (Fig.7(b)). Then, lines are divided at this intersection (Fig.7(c)). These lines can be encoded precisely in the manner optimal for each type of line (Fig.7(d)), without aforementioned spline side effect. Furthermore, the information of the width of line-drawings is added to each line, so that each divided line can be decoded by using this information.

4.3. Simulation result and consideration

We tested the performance of hybrid coding by straight lines and spline curves. In the experiment, lines whose

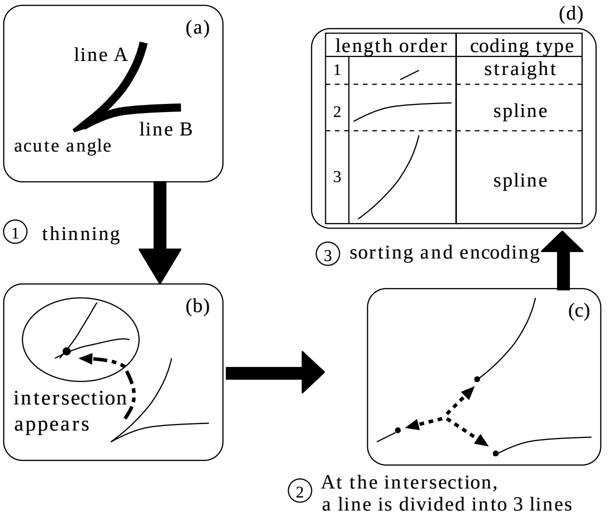

Figure 7: Example of hybrid coding by straight lines and spline curves

lengths are less than 20-pixel lengths were sampled at the interval of 8-pixel length. In addition, a line between two adjacent sampled points was approximated by a straight line. The input image (thinned line-drawings) and approximated lines are shown in Fig.8. Rate distortion characteristics is shown in Fig.9. The error value is the summation of each area surrounded by the original line and the approximated line. From Fig.9, the output image shown in Fig.8 is regarded as the image that has the lowest error value comparing to other images of the same data size.

5. EXTRACTION OF SIGNIFICANT POINTS

5.1. Problem of the conventional method

In the conventional method [6], significant points for approximation of spline curves and straight lines are extracted by using two consecutive vectors. These vectors are traced along line-drawings or contours and the angle (or length) created by two vectors is calculated. If the angle is larger than the threshold, the point between two vectors is chosen as a significant point. However, significant points on curves are not always detected from smooth curves.

5.2. Ramer Algorithm

5.2.1. Outline of Ramer Algorithm

Ramer algorithm is a well-known method for curve approximation [7]. In the algorithm, a straight line is drawn between two ends on the original curve, and the point most distant from the straight line is chosen as a significant point for the approximation (Fig.10).

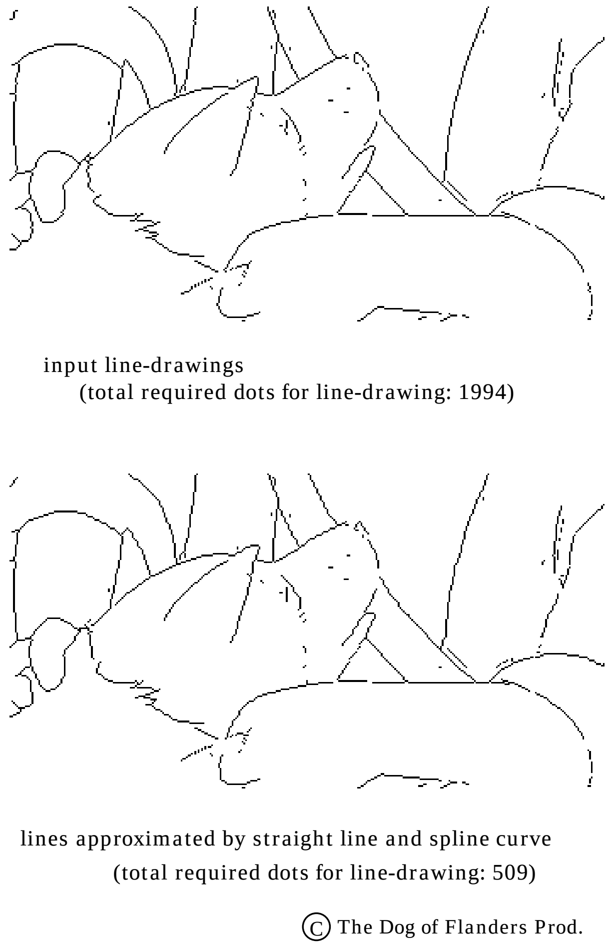

Figure 8: output image created by hybrid coding algorithm by straight lines and spline curves

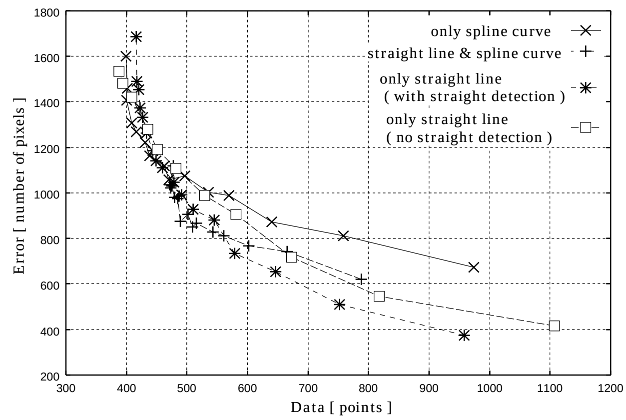

Figure 9: An result of hybrid coding algorithm by straight lines and spline curves

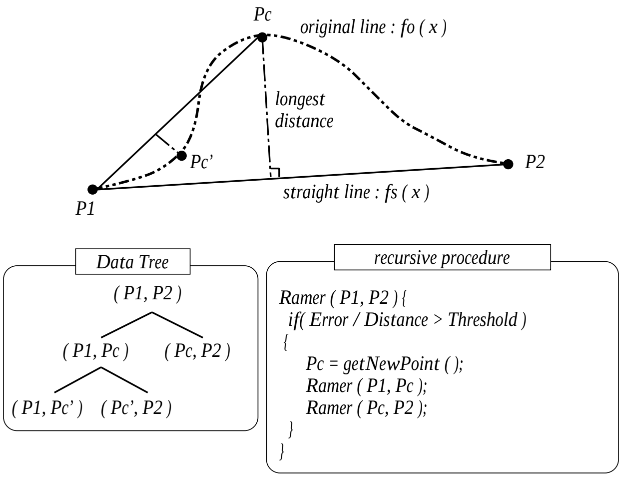

Figure 10: Ramer Algorithm

5.2.2. Performance Evaluation of Ramer Algorithm

In order to compare the conventional method for extraction of significant points and Ramer algorithm, we use a sample image shown in Fig.11. It comprises of a straight line and a curve. In this case, approximation by straight lines is only employed.

In Fig11, extracted significant points, inserted passing points and approximated lines are drawn. The conventional method shows that it is difficult to extract significant points because the curvature of the sample line is small. Hence, all points used for the approximation are inserted passing points (Fig.11(a)). Additionally, there is the problem that such useless passing points are inserted into a straight line. However, in the case of Ramer algorithm, extracted points are concentrated on the curve (Fig.11(b)). Furthermore, according to the threshold T used in Ramer algorithm, the number of extracted points on the curve adaptively changes. Comparison of two algorithms for extraction of significant points is listed below.

- the conventional method

  Parameters for control of extracted points are (1) the scalar of a vector, (2) an angle or a length created by two vectors, and (3) intervals of inserted passing points. In this method, it is necessary to employ more parameters than employed in Ramer algorithm. This method is more suitable for the extraction of acute points than points on smooth curves.

- Ramer Algorithm

  An error value regulated by a length of a straight line is the only parameter for the control of extracted points. Control of them in Ramer algorithm is easier than the one in the conventional method. However, the problem is that this algorithm is influenced by notches on line-drawings when threshold is too small.

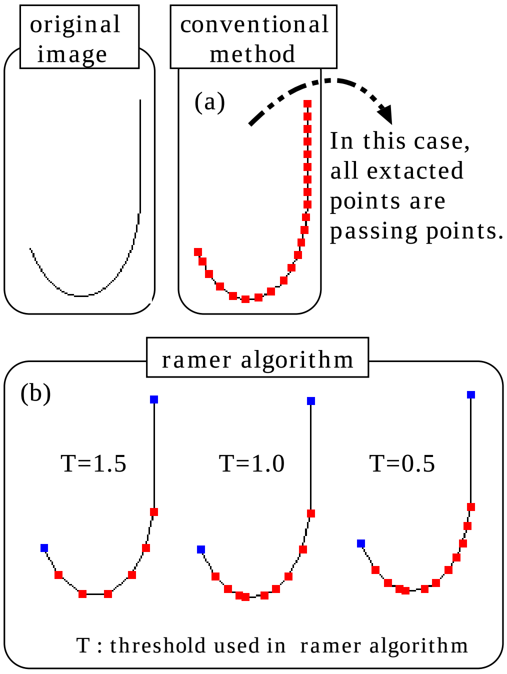

Figure 11: Comparison of methods for extraction of significant points

5.3. Utilization of Ramer Algorithm

5.3.1. Outline

We propose utilization of Ramer algorithm, which is especially designed for animation image coding. In this algorithm, we focus on two points: (1) process hybrid approximation of spline curves and straight lines and (2) control output data sizes (see Fig.12).

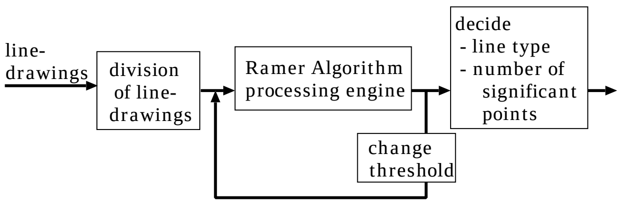

Figure 12: Utilization of Ramer Algorithm

This algorithm takes four steps. First, extracted line-drawings are divided at each intersection of them in the same manner shown in Fig.7. Secondly, Ramer algorithm is executed on each divided line-drawing. Computed significant points and an error value are output. Thirdly, the threshold used in Ramer algorithm is changed and significant points and an error value are recursively output at each time of this change. Finally, from the relationship between the number of significant points (data size) and error values (data quality), the type of lines used in the approximation and the number of significant points are decided. Error values employed in this algorithm are (1) an error value between the original lines and lines approximated by straight lines and (2) an error value between the original lines and lines approximated by spline curves.

5.3.2. Evaluation

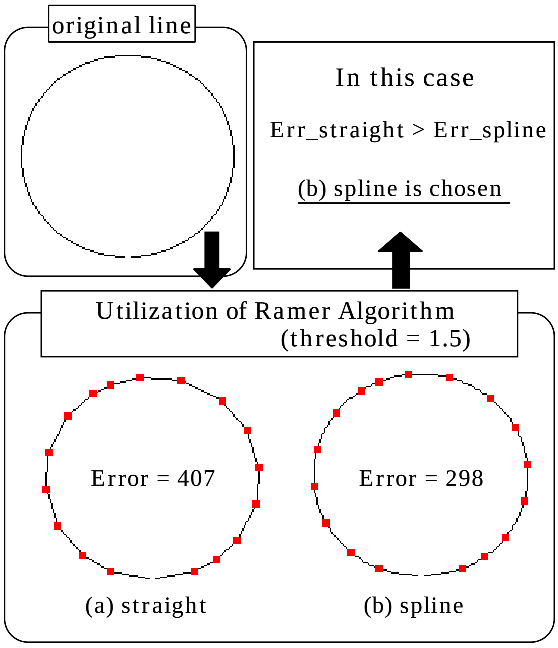

Figure 13: Example of utilization of Ramer Algorithm

A curve in the sample image shown in Fig.13 illustrates that the error value of straight approximation is greater than the one of spline approximation when there are a few significant points extracted by Ramer algorithm. In this case, a spline curve is chosen as the method of the approximation after comparison of the each error value.

The proposed algorithm is executed on the sample image shown in Fig.8 and the result of the evaluation is shown in Fig.14. From this result, the performance of utilization of Ramer algorithm is the best when data is over 560 points.

6. CONCLUSION

In this paper, we have proposed the three-component coding scheme designed for animation images. Line-drawing, homogeneous color regions and background regions are extracted from an animation image and encoded with three refining algorithms. The proposed scheme can provide animation images without mosquito noise which is typically observed by DCT-based coding. Our refined algorithms enable better coding performance than the conventional one.

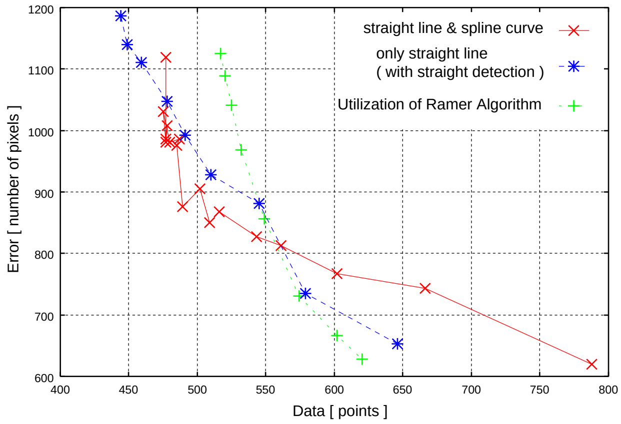

Figure 14: Result of utilization of Ramer Algorithm

Currently, the proposed coding scheme is designed for still images. The extension to video coding will be investigated in the future. It is further study to evaluate the proposed animation coding scheme against conventional DCT-coding, precisely in terms of subjective and objective image quality at same bit rate.

7. REFERENCES

[1] W. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. on Communication*, vol. COM-25,No.11, pp. 1285–1292, 1977.

[2] *ITU-T Rec. T.81 (1992)|ISO/IEC 10918-1:1994 Digital compression and coding of continuous-tone still images: Requirements and guidelines.*

[3] *ISO/IEC 11172-2:1993 Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video.*

[4] S. Carlsson, "Sketch based image coding of grey level image," *Signal Processing*, vol. 15, pp. 57–83, 1988.

[5] X. Ran and N. Farvardin, "A perceptually motivated three-component image model–partII: Applications to image compression," *IEEE Trans. on Image Processing*, vol. 4,NO.4, pp. 430–447, 1995.

[6] K. Furukado, H. Watanabe, and N. Kobayashi, "A Note on Spline/DCT Hybrid Coding for Animation Images (in Japanese)," *IPSJ*, vol. AVM 24-3, pp. 11–16, 1999.

[7] U. E. Ramer, "An iterative procedure for the polygonal approximation of plane curve," *CGIP*, vol. 1, pp. 224–256, 1972.