Computer Graphics

Theory and Applications

Edited by Tosiyasu L. Kunii

Computer Graphics

Theory and Applications

Edited by Tosiyasu L. Kunii

List of Contributors

The page numbers given below refer to the page on which contribution begins.

Alexander Ames 337
Kimio Arai 212
Michael J. Bailey 128
Brian A. Barsky 75
Michael L. Barton 426
Alice Bernhard 328
Ramachandran Bharath 414
Lewis D. Brentano 222
Thomas H. Bruggere 247
Y. K. Chan 144
Gene Chao 194
Camila Chaves Cortes 261
Richard Chuang 350
Thomas DeFanti 380
Glenn Entis 350
Akira Fujimoto 2
K. Fujimura 96
David B. Hoffman 414
Takashi Hoshi 460, 500
Ilhan A. Ince 426
Yoshioki Ishii 500
Mitsuru Ishizuka 475
Kansai Iwata 2
Willem Jansen 162
Irwin M. Jarett 310
Yu-Huei Jea 16
Howard W. Johnson 310
Atsumi Kimura 212
Donald J. Kostuch 320
T. Kozawa 182
T. L. Kunii 96
David Kushner 374
Harry Lehrhaupt 337
Stephen R. Levine 445
Tsung-Che Lin 16
N. Magnenat-Thalmann 360
Aaron Marcus 405
N. Nakajima 11
Fumio Nakamura 212
Shohei Nakamura 514
Kazuhiko Nakayama 460, 500
Masayuki Numao 475
Y. Ohta 111
James J. Oravec 426
R. Rus Rodgers 137
Francisco A. Rodriguez 34
David F. Rogers 34
Judson Rosebush 374
Steven G. Satterfield 34
J. Sakemi 182
Allan H. Schmidt 388
Tsu Yuan Shen 50
Howard A. Spielman 300
Yasuyuki Suzuki 67
Tadashi Takanishi 212
Daniel Thalmann 360
S. Tokumasu 111
H. Toriya 96
Yoshio Tozawa 483
A. Tsukizoe 182
James R. Warner 60
Michael J. Wozny 203
K. Yamaguchi 96
Yasuhiko Yasuda 475

Preface

This book is an extensive treatise on the most up-to-date advances in computer graphics technology and its applications. Both in business and industrial areas as well as in research and development, you will see in this book an incredible development of new methods and tools for computer graphics. They play essential roles in enhancing the productivity and quality of human work through computer graphics and applications.

Extensive coverage of the diverse world of computer graphics is the privilege of this book, which is the Proceedings of InterGraphics '83. This was a truly international computer graphics conference and exhibit, held in Tokyo, April 11-14, 1983, sponsored by the World Computer Graphics Association (WCGA) and organized by the Japan Management Association (JMA) in cooperation with ACM-SIGGRAPH. InterGraphics has over 15 thousands participants.

This book consists of seven Chapters. The first two chapters are on the basics of computer graphics, and the remaining five chapters are dedicated to typical application areas of computer graphics. Chapter 1 contains four papers on "graphics techniques". Techniques to generate jag free images, to simulate digital logic, to display free surfaces and to interact with 3 dimensional (3D) shaded graphics are presented. Chapter 2 covers "graphics standards and 3D models" in five papers. Two papers discuss the CORE standard and the GKS standard. Three papers describe various 3D models and their evaluations.

Chapter 3 attacks one of the major application areas "CAD/CAM (computer-aided design and manufacturing)" with 11 papers. Four papers cover mechanical CAD/-CAM, two papers CAD/CAM for VLSI, and the remaining five papers report on important topics such as CAD/CAM education, documentation, communication, computer-aided engineering and CAD trends in the 1980s.

In-depth studies of key issues in another important area, "office automation (OA)", are given in Chapter 4 in five papers. It covers management, financial and word processing applications as well as general areas in OA. In Chapter 5 the fascinating area of "computer animation" is presented in four papers emphasizing 3D techniques and dedicated systems design. Diverse "graphic applications" such as automated cartography, graphic design, scientific applications and hard copy are covered in five papers in Chapter 6.

The last chapter, Chapter 7, contains five papers on "image processing", which is the reverse of computer graphics. Computer graphics is a mechanism to generate images from their description, and image processing is the reverse mechanism to generate image description from images.

It is the great pleasure of the editor to acknowledge the following key people who made this InterGraphics '83 possible: Mr. Caby C. Smith, President of WCGA, Mr. Akira Totoki, President of JMA, Mr. Michiya Ishii, Former Secretary General of JCGA, and Prof. Thomas A. DeFanti, Chairman of ACM-SIGGRAPH.

I would like to thank Springer-Verlag Tokyo, especially Mr. M. Tsuchida, Ms. C. Sato and Mr. H. Matthies for their help to publish this beautiful volume.

Dr. Tosiyasu L. Kunii

Chairman, Program Committee,
InterGraphics '83, and
Professor of Computer Science,
The University of Tokyo

List of Contents

Chapter 1
Graphics Techniques 1

Jag Free Images on a Raster CRT
(A. Fujimoto, K. Iwata) 2

An Interactive Graphic Processor for Digital Logic Simulation System
(Y. H. Jea, T. C. Lin) 16

Shiphulls, B-Spline Surfaces and CADCAM
(D. F. Rogers, S. G. Satterfield, F. A. Rodriguez) 34

Interactive Techniques for 3-D Shaded Graphics
(T. Y. Shen) 50

Chapter 2
Graphics Standards and 3D Models 59

Device Independence and Intelligence in Graphics Software
(J. R. Warner) 60

Implementation of the Graphical Kernel System (GKS)
(Y. Suzuki) 67

A Description and Evaluation of Various 3D Models
(B. A. Barsky) 75

Oct-tree Algorithms for Solid Modeling
(K. Fujimura, K. Toriya, K. Yamaguchi, T. L. Kunii) 96

Implementation of Geometric Modelling System: HICAD
(S. Tokumasu, Y. Ohta, N. Nakajima) 111

Chapter 3
CAD/CAM 127

Computer Graphics in Mechanical Engineering
(M. J. Bailey) 128

Computer-Aided Design, Analysis, Manufacturing and Testing Techniques as Applied to Plastic Part and Mold Development
(R. R. Rodgers) 137

Computer Aided Manufacturing of Machine Parts by Volume Building
(Y. K. Chan) 144

Computer-Aided Design of Pumps
(W. Jansen) 162

Geometrical Analysis of Mask Pattern for VLSI and Its Application
(A. Tsukizoe, J. Sakemi, T. Kozawa) 182

High-Performance Computer Graphics for VLSI Systems Design
(G. Chao) 194

Education and Training Requirements for CAD/CAM
(M. J. Wozny) 203

Design Document Generation from Engineering Databases
(F. Nakamura, A. Kimura, T. Takanishi, K. Arai) 212

Data Communications, CAD/CAM Graphics and Distributor Design Concepts
(L. D. Brentano) 222

Computer Aided Engineering
(T. H. Bruggere) 247

Computer Aided Design Trends in the 80's
(C. C. Cortes) 261

Chapter 4
Office Automation 299

Computer Graphics for Management: Prerequisites for an Effective Introduction
(H. A. Spielman) 300

Using Computer Generated Graphics to Present Financial Statements
(I. M. Jarett, H. W. Johnson) 310

The Value of Computer Graphics
(D. J. Kostuch) 320

Computer Generated Graphics... User, Designer, Programmer
(A. Bernhard) 328

Integration of Word Processing and Graphics
(A. Ames, H. Lehrhaupt) 337

Chapter 5
Computer Animation 349

3D Shaded Computer Animation, Step-by-Step
(R. Chuang, G. Entis) 350

The Use of 3D Abstract Graphical Types in Computer Graphics and Animation
(N. Magnenat-Thalmann, D. Thalmann) 360

3 Dimensional Reconstruction: A Case Study of a Perspective Problem
(J. Rosebush, D. Kushner) 374

Extended Memory Use in the Zgrass Graphics System
(T. A. DeFanti) 380

Chapter 6
Graphic Applications 387

Automated Cartography
(A. H. Schmidt) 388

Graphic Design for Computer Graphics
(A. Marcus) 405

Computer Graphics to Show Optimal Smoothing and Trend Adjustments for Exponential Forecasts
(D. B. Hoffman, R. Bharath, C. M. Carlson) 414

Application of Interactive Computer Graphics to the Results of Electromagnetic Field Calculations
(I. A. Ince, M. L. Barton, J. J. Oravec) 426

Hard Copy and Computer Graphics
(S. R. Levine) 445

Chapter 7
Image Processing 459

A Landsat Image Databank Using Mass Storage System —"TSUKUSYS" Image-Bank—
(T. Hoshi, K. Nakayama) 460

A Rule-Based Interpretation of Contour Patterns with Curves
(M. Ishizuka, M. Numao, Y. Yasuda) 475

A Specification of Image Processing for Remote Sensing
(Y. Tozawa) 483

The Analysis of LANDSAT (M2S) Image Data Using Interactive Information Retrieval Language SOAR
(Y. Ishii, K. Nakayama, T. Hoshi) 500

3 Dimensional Image Processing of Ultrasonotomograms
(S. Nakamura) 514

Author Index 525

Chapter 1

Graphics Techniques

JAG FREE IMAGES ON A RASTER CRT

Akira Fujimoto*
Kansei Iwata
Graphica Computer Corporation
505 Kaitori, Tama-shi
Tokyo 206, JAPAN

ABSTRACT

This paper presents an algorithm for removing all kinds of jags (staircase-like effects) due to the aliasing phenomenon which is intrinsic in all synthesized wire frame and continuous-tone images displayed on raster CRT's.

This algorithm produces images whose quality is virtually equal to the results obtainable by the use of far more elaborate techniques but nevertheless the implementation and the computational costs of the algorithm are extremely low. This and other important features of the algorithm have been proven by numerous experimental results and succesfull practical implementations.

The actual implementation of the algorithm was carried out for two types of aliasing. First as a smooth vector generator for wire frames and second as an anti-aliasing post processor for continuous-tone image outputs, both being implemented locally as intelligent features on the display sites.

Because of its universality, speed, accuracy and low implementation cost, the algorithm can be inexpensively hardwarized, providing an effective anti-aliasing tool for practical applications.

1. INTRODUCTION

In recent years, sharply decreasing memory costs have resulted in expanding popularity of raster scan displays, making them increasingly competitive with random-scan vector displays. The special properties of the raster-scan display have facilitated greater realism in pictures and posed new challenges for graphic applications. On the other hand the raster-scan display can be and is actually more often used for many of the purposes for which the random-scan display was developed. In raster-scan applications, however, we inevitably face the well known problem which is characteristic to this device: aliasing or rastering. Picture degradation caused by aliasing is often considered as an "Achilles' heel" of the raster display devices. Considerable effort has been expended in attempting to overcome the "jaggy" or "staircase" appearance of lines and edges. All

*former name: Wieslaw Romanowski

solutions are naturally limited to displays with a frame buffer which can handle more than two intensities per pixel. Any algorithm, no matter what sort of approach has been adapted, finally fixes the gray scale intensity level for each pixel. Many different approaches of different levels of complexity have been proposed. Most of the existing solutions, however, in spite of being able to produce high quality anti-aliased images, lack universality or involve a considerable amount of calculation which makes them often prohibitively expensive or just too slow from the view point of practical applications. The decisive factor for succesfull application of any particular anti-aliasing technique is, along with quality, the speed of the anti-aliasing process. In the present paper, the author presents an algorithm which, while producing high quality anti-aliased images, succesfully resolves a problem of speed.

2. SPATIAL FILTERING

As regards the "jaggy line" produced on the raster CRT, one must clearly realize that there exists no medicament which can treat the aliasing problem without side effects. This stems directly from the so called "sampling theorem", which states that a sampled signal cannot reproduce a frequency component higher than half the sampling frequency.

To avoid aliasing, different filtering techniques are applied before actual sampling takes place. The purpose of filtering is to remove spatial frequences of the image which are too high to be representable on the given raster. Obviously, a filtering procedure removes "jag" at the cost of spatial frequencies Images produced on the screen will look more or less blurred, depending on the amount of high frequencies "sacrified" during filtering. In other words, the application of low pass filtering before sampling merely changes the degradation of the image from one from to another. Realizing this fact is especially important when treating vectors on raster CRT's. Preserving the high spatial frequencies for vectors which are relatively thick even without any filtering can be no less important than removing its jags. The filtering procedure involves choosing a filter function which is mathematically tractable. The vector is assumed to have a finite thickness and the actual intensity of each pixel of the vector is obtained by integrating the filter function over the region it covers.

Nishida and Nakamae(2) proposed making the pixel intensity proportional to the area it contributes to the vector. This corresponds to the implied adoption of the Fourier window as the filter function. This function intersects the line in various patterns. The computational effort involved in calculating the area covered by Fourier window makes the proposed algorithm practically unacceptable. On the other hand, S.Gupta and R.F. Sproul(3) adopted a conical filtering function. Because of its circular symmetry, for the given thickness of the line, pixel intensity depends only on the distance from the pixel center to the center line, and it can be precalculated for specific distant values. This requires creating and storing the look up table for each particular gray scale. The table must be referred to in each generation of the particular pixel. Moreover, the algorithm requires separate, time-consuming treatment for the end points of the vector. Also, the problem of intersecting lines is left open. On the other hand, no single experimental result justifying the claimed practical importance of the proposed algorithm was presented.

3. THE ALGORITHM

The solution proposed in the present paper utilizes the special properties of the vector, namely its slope and thickness, which is geometrically equal to zero. The Fourier window is adopted for the spatial filter. The Fourier window is assumed to have a value of 1.0 in the region ± 0.5 of the pixel and is zero everywhere else[4]. In present implementation, however, the filter function is normalized so that the enclosed volume is 1.(see Fig. 1). Furthermore, it is assumed that the size (d) of the window is the function of the slope of the vector, and that its orientation is always ajusted to the slope of the vector as shown in Fig.1. Each pixel intensity is obtained by convoluting the filter function and the line intensity. The integration of the filter function will be performed over the region $2(d/2-dx)d$ (see Fig.1). Integration of the filter function is thus extremely simple. Each pixel intensity is inversely proportional to the vector, i.e., $C'=C\,(d/2-dx)/d$, where C is the overall vector intensity and C' the intensity of the pixel under consideration.

The pixel intensity distribution in the direction perpendicular to the center line of the vector is shown on Fig.2. The intensity of the pixel as it approaches the center line of the vector will converge to the overall vector intensity. If the thickness of the vector is assumed to be finite, then this property is satisfied only when the size of the filter function is equal to or less than the thickness of the vector.

All further discussion will be restricted to the vectors in the first octant (i.e., $0<y<x$); the extention to other regions is obvious.

First, we try to produce a jag free vector which is as thin as possible. The thickness of the displayed vector is limited by the smallest possible size of the Fourier window (see Fig.1). When generating the pixel on the one side of the vector, we must also ensure the existence of its counterpart on the other side of the vector. This obvious requirement sets a limit on the window size for a vector with given slope (see Fig.2). As was noted above, each pixel intensity is a linear function of its distance from the center line of the vector. The principle of superposition can then be applied and the intensity of each pixel can be calculated incrementally without explicit calculation of this distance. The algorithm, in which incremental d - Fourier window size

dx - distance between pixel and center line of vector

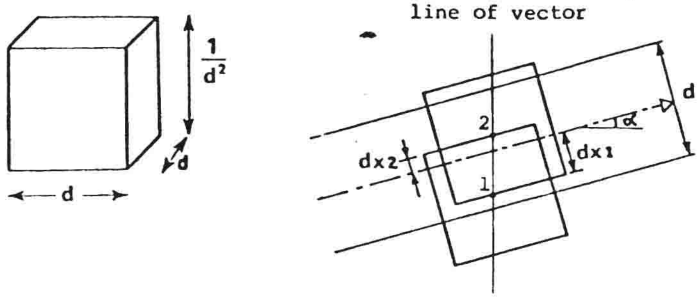
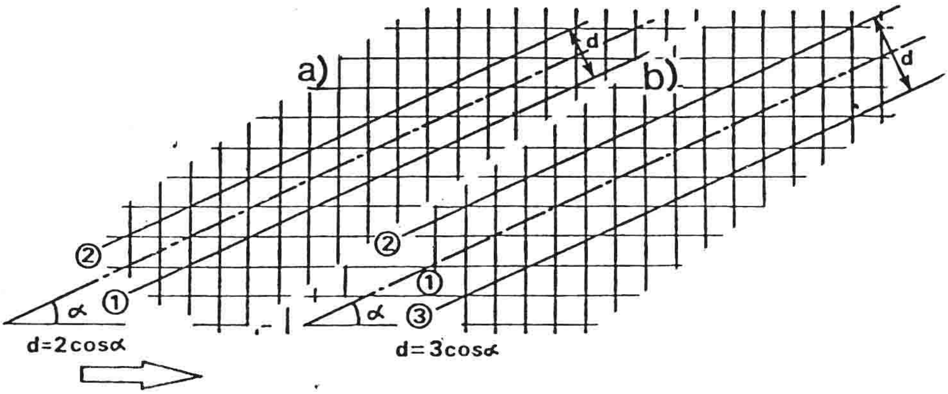
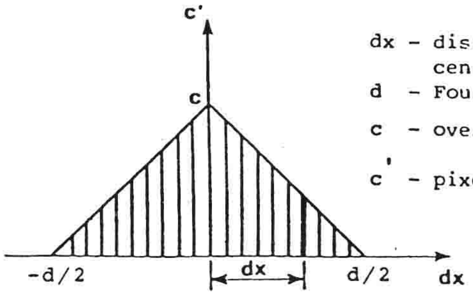


Figure 1: The Fourier window and its orientation along the vector



a)smallest possible window involving shading of one pixel on each side.
b)expanded Fourier window in which three pixels are shaded per each column.

Figure 2: Fourier window size for given slope of vector.



dx - distance between pixel and vector center line

d - Fourier window size

c - overall vector intensity

c' - pixel intensity: c'=c(d/2-dx)/d

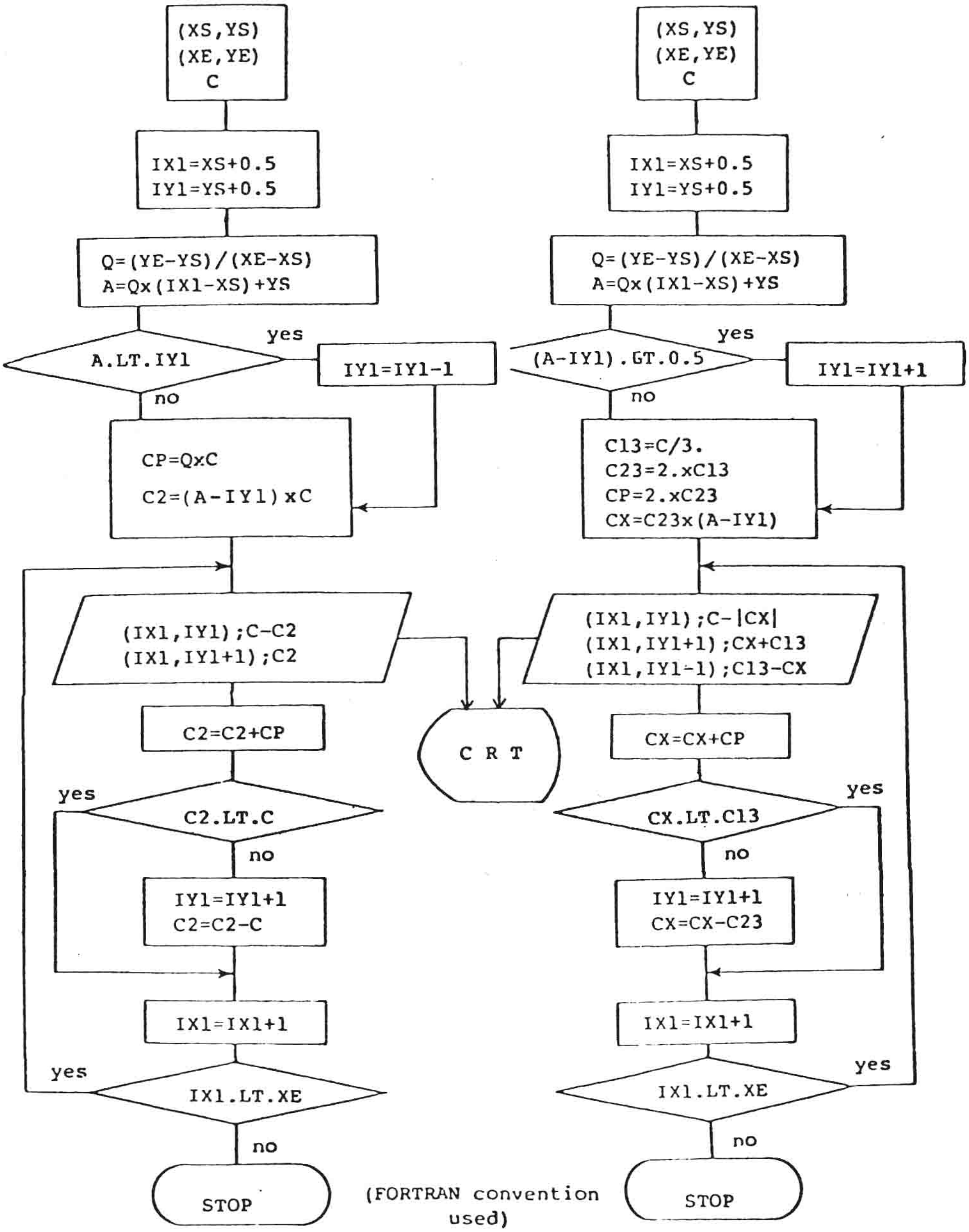
Figure 3: Intensity distribution of pixels across the vector.

calculation of this pixel intensity was adopted, is shown in Fig.4. The indices 1 and 2 refer to variables relating to pixels lying on the right and left sides, respectively, of the center line of the vector. Such a pair of pixels, however, is placed in the column which is located perpendicular toward the direction of the greatest displacement of the vector. In the first octant, the X axis represents this direction (see Fig.2).

The input data are the coordinates of the initial (XS,YS) and final (XE,YE) points of the vector, together with its intensity C.

The algorithm consists of two parts, i.e., the initialization part and the main part. The initial calculation proceeds as follows. The coordinates of the right pixel (IX1,IY1) are obtained by rounding off the coordinates of the initial points of the vector. Then the slope of the vector (Q) and the value (A) used for checking the y coordinate of the first pixel (IY1) are calculated. After eventual correction of the y coordinate, the increment of the intensity corresponding to one pixel increment (CP) in the direction of greatest displacement is calculated. The initialization process is completed by calculating the value (C2) which corresponds to the initial intensity of the left pixel. In case of a vector where only two pixels are generated per increment, we have C=C1+C2, so the calculation of the right pixel intensity is immediate. The coordinates of the left pixel differ only in the y coordinate: IY2=IY2+1. At this point, the coordinates and intensity of the pair of pixels are sent to frame buffer. The main part of the algorithm then proceeds in the following way. The intensity for the left pixel (C2) is incremented by CP and immediately checked against its maximum possible value, i.e., the overall intensity of the vector (C). If its actual value is still less than (C), the only a move in the direction of the greatest movement is executed, otherwise, an additional increment in the perpendicular direction is made and the actual intensity of the left pixel is corrected by subtracting the overall vector intensity value. The pixel coordinate generation process carried out in the main part of the algorithm happens to resemble Bresenham's one[1]. In the present algorithm, however, it is the pixels' intensity control variable(C2) which directly governs its coordinate generation. This means that the control variable simultanously incorporates two distinct meanings., i.e., physical and geometrical. In Bresenham's algorithm, the controlling error term, which is measured perpendicularly to the axis of greatest displacement, possesses only geometrical meaning.

The variation of the algorithm with an expanded Fourier window size (see Fig.2b) is shown in Fig. 4b. In this case, 3 pixels are shaded in each column. The intensity of the outer pixels will vary between 0 and 2/3 of the overall vector intensity. The intensity of the centrally located pixel oscillates between 2/3 of the overall vector intensity and its maximum value.



a) smallest possible Fourier window; two pixels shaded in each column

b) expanded Fourier window; three pixels shaded in each column

Figure 4: The flow chart of the algorithm.

4. EXPERIMENTAL RESULTS

A display* with the following characteristics was used for the experiment: 512x512 i.e., rather low range of resolution; 280x 280mm of screen size; 8 bits depth buffer (256 intensities) for each RGB per pixel.

a) Smooth vector generator (straight line)

The results of direct application of the algorithm explained in the previous section are shown in Fig.5.

(1) Variable window size.
Variations in the size of the Fourier window (for vectors having different slopes) does not produce any visual difference in vector thickness, in spite of the fact that the actual change in its size is as large as 30% (a case where only two pixels per column are shaded). This is explained by property of the algorithm, which maintains the sum of the intensities of the pair of two pixels in each column constant.

(2) "Twisted rope" effect.
Lines with slope approaching 0 or infinity still have a somewhat twisted rope-like appearance, even in the case where a generally larger Fourier window was used (upper middle part of Fig.5). This phenomenon is due mainly to the characteristics of the monitor i.e., the light emitted by each pixel is not, in general, distributed uniformly over an idealized square. Actually, however, the twisted rope-like appearance of the smoothed vector cannot be avoided in any anti-aliasing technique unless window size or line thickness is significantly increased[2,3]. The former, however, will result in producing a line which is exessively thick and blurred in comparison with the usual, already thick lines on raster CRT's. The twisted rope effect will obviously diminish significantly or completely disappear in displays with greater resolution.

(3) Bit depth factor.
During the experiments, it became clear that increase of the gray scale above 8 intensities (3 bits per each R,G,B) does not produce visible improvement in the appearance of the vectors.

b) Smooth vector generator (curves)

Any curve can be represented by fitting it with a polygonal line, a standard procedure in graphics. Arcs approximated by using 8,20,30,100,1000 and 10000-sided polygons are shown in Fig.6. In this particular case (radius=320 pixels) no visible change is observed when the number of polygons is increased beyond 30. Experimental results proved that the visually smooth

---

* Graphics and image processing display M508, manufactured by the Graphica Computer Co., Japan.

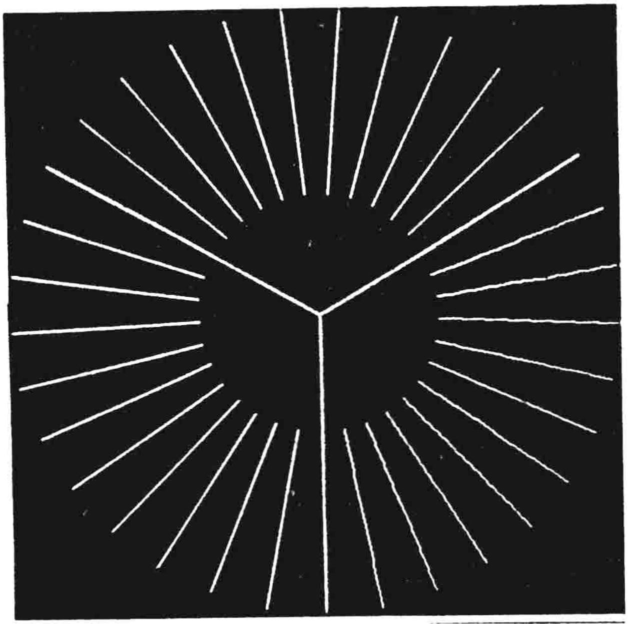


Figure 5: Vectors drawn on a 512 x 512 raster CRT.

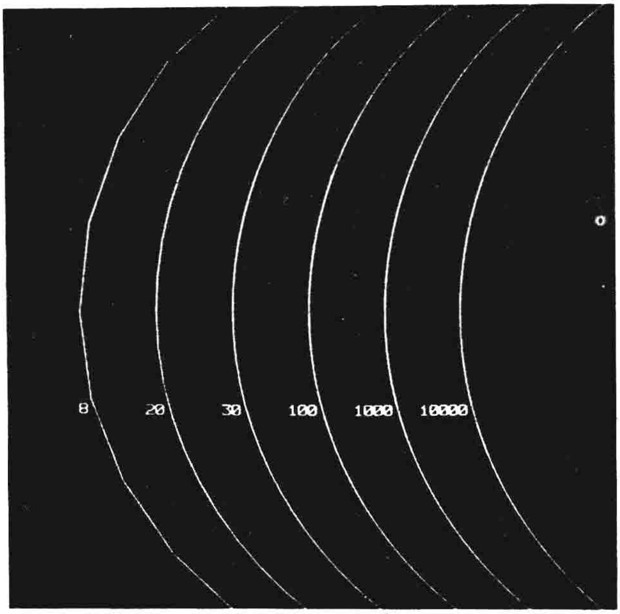


Figure 6: Anti-aliased vectors drawn on a 512 x 512 raster CRT: extention to curve generation.