7 Cross Scan Buffer for Interactive Photorealistic Rendering

We propose the cross scan buffer which preserves the result of hidden surface removal as performed by the cross scanline algorithm. The cross scan buffer reduces image re-generation time and eliminates aliasing artifacts even if the image is arbitrarily scaled. Perfect anti-aliasing is achieved because the cross scanline algorithm analytically determines visible polygonal surfaces and divides them into sets of triangles and trapezia. The cross scan buffer supports the various applications that currently use the conventional buffering methods for anti-aliasing. This chapter introduces and tests three applications: image scaling, shadow creation, and texture mapping. Experimental results verify that the cross scan buffer is very powerful yet efficient.

7.1 Introduction

The standard rendering pipeline consists of the following operations [Foley90]; viewing conversion, perspective conversion, clipping, hidden surface removal, and shading. Image synthesis is usually iterated until the synthesized images satisfy the user. The user may modify object layout, view point, lighting parameters, and object surface parameters inclusive of mapping. Since the iteration is mainly required to set those parameters, overall rendering time can be shortened by saving the results of hidden surface removal in a buffer. Such a buffer would also improve the interactive environment for image synthesis. Several rendering buffers have been proposed including the span buffer [Whitted81], [Nakamae89], ray-tree, and G-buffers [Saito90]. They lie between hidden surface removal and shading operations, and store the results of hidden surface removal. However, conventional buffers cause aliasing artifacts because they employ scanline, ray-tracing, or z-buffer, and so digitize the object surfaces as pixels or scanlines.

In this chapter, we propose the cross scan buffer which preserves the results of the hidden surface removal by the cross scanline algorithm [Tanaka90]. The cross scanline algorithm can determine the exact geometric shapes of visible surfaces and can accurately prevent the occurrence of aliasing artifacts; exact shapes are stored in the buffer. This characteristic of the cross scan buffer is very useful when arbitrarily scaling a rendered image; there is no need to repeat the rendering pipeline. This new algorithm is very powerful because it yields rapid image synthesis. In addition, the cross scan buffer offers:

- more accurate shadow creation than is possible with the conventional z-buffer shadowing [Reeves87],
- exactly anti-aliased texture mapping,
- aliasing-free image composition,
- more accurate primary ray tracing than Area Sampling Buffer [Sung92], and
- accurate form factor computation.

In the fields targeted, the cross scan buffer can efficiently prevent aliasing artifacts unlike the conventional digital buffers.

This chapter presents the concept of the cross scan buffer and introduces its applications.

7.2 Concept of the Cross Scan Buffer

The cross scan buffer preserves the hidden surface removal results of the cross scanline algorithm (see Appendix **A**). The cross scanline algorithm analytically determines visible polyhedral objects by scanning the object space first horizontally then vertically. In Fig.7.1, horizontal and vertical scanlines are drawn as solid and dotted lines, respectively. Horizontal scanlines are located at every horizontal pixel boundary. Vertical scanlines are located at

(1) intersections of the horizontal scanlines and polygon edges,

(2) polygon vertices, and

(3) edge crossing points, and are scanned between two adjoining horizontal

scanlines.

As a result of horizontal and vertical scanning, the polygons are divided to triangles and/or trapezia. Thus, the cross scanline algorithm efficiently executes accurate hidden surface removal without aliasing artifacts.

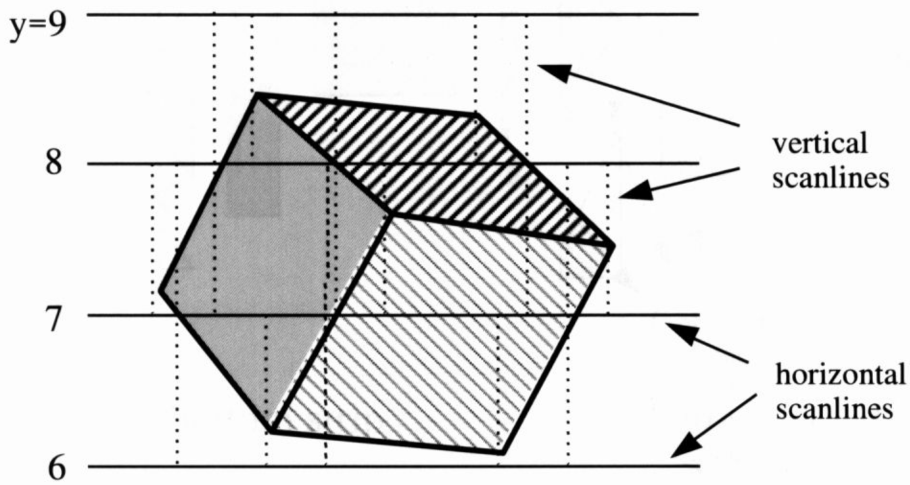


Fig.7.1 Result of the Cross Scanline Algorithm.

The cross scan buffer stores visible polygons divided into triangles and trapezia as shown in Fig.7.2, which corresponds to the section between the 7th and 8th horizontal scanlines of Fig.7.1. Object surfaces are segmented into strips by adjoining horizontal scanlines, then each strip is segmented by several vertical scanlines. Finally, each segment is divided into a set of triangles and trapezia.

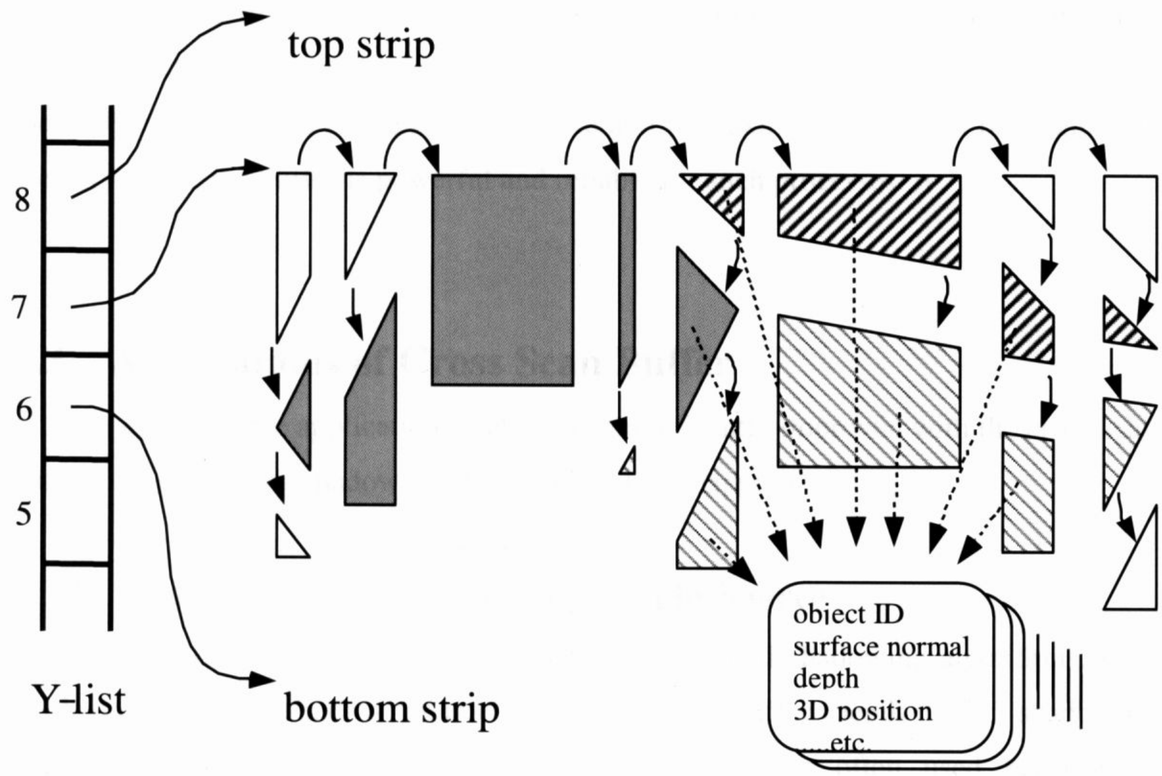


Fig.7.2 Cross scan buffer description of the middle strip in Fig.7.1.

The triangles and trapezia are connected by pointers as shown in Fig. 7.2. Here, the 7th cell of the Y-list points to the trapezium located at the most left side of its corresponding strip. In the representation used by the CSB, the background, regions wherein no objects are visible, are also described as a surface. The white triangles and trapezia in Fig. 7.2 represent the background.

Shapes of the triangles and trapezia are described by the positions of their vertices on the image plane. Parameters such as object IDs, surface normal vectors, and distances from the view port, are also included if they are needed by the application. The structure of the cross scan buffer seems rather complicated, however it can save all visible objects and their shapes even if they are very small and thin.

Storing exact object shape is one great advantage of the cross scan buffer. Aliasing artifacts are prevented if the values of each pixel are calculated by accumulating colors of surfaces projected onto the pixel in proportion to their area. In this way, the CSB achieves exact anti-aliasing easily and efficiently. Generally speaking, users often modify the rendering and lighting parameters to obtain higher quality rendering result. Since correctly highlighted images can be generated from the surface normals saved in the CSB by using the Precise Rendering Method [Tanaka92] even when the highlights are very sharp, the buffer is powerful and reliable for such processes.

7.3 Applications of Cross Scan Buffer

We discuss three applications that can be significantly improved with the cross scan buffer: image scaling, shadow creation, and texture mapping.

7.3.1 Shadow Creation due to Point Light Sources

Since shadows improve image photo realism, several shadowing algorithms, such as shadow polygon, shadow volume [Crow77], and z-buffer (depth-buffer) shadowing [Reeves87], have been proposed. Z-buffer shadowing is often used because its algorithm is simple and its computing cost is lowest of the known alternatives. Instead of z-buffer shadowing, we propose cross scan buffer shadowing which replaces the shadow z-buffer by the cross scan buffer. First we explain the conventional z-buffer shadowing algorithm as well as its advantages and disadvantages, and then the cross scan buffer shadowing is introduced.

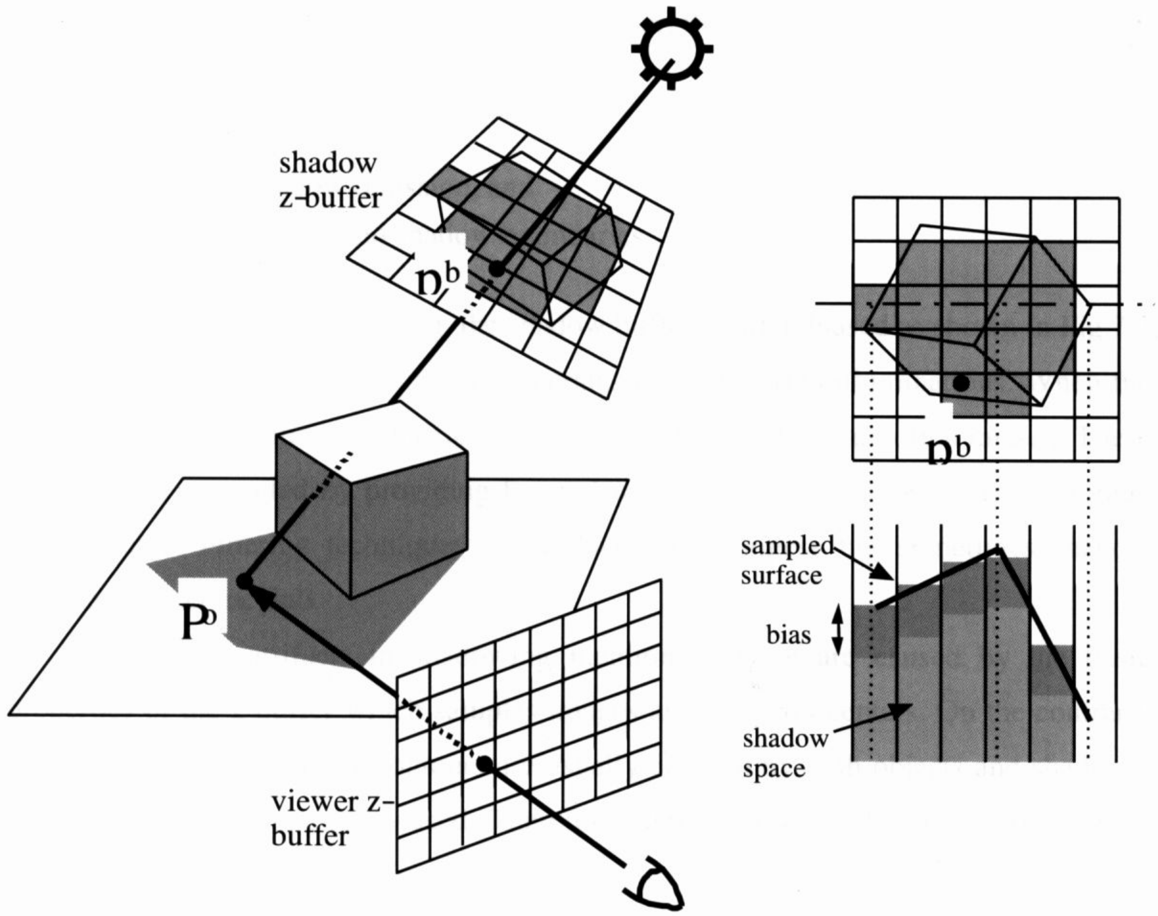


Fig.7.3 Z-buffer Shadowing.

As shown in Fig. 7.3, the z-buffer algorithm is applied to all light sources where the source positions are the view point positions. Distances from each point light source to illuminated surfaces are stored in separate z-buffers. Here, the z-buffers are called shadow buffers. By referring to the shadow buffers, it is easy to determine whether an object surface is illuminated or shadowed as follows:

Let $\mathbf{P}_o$ be a 3D point on the object surface as shown in Fig. 7.3, where the point is recovered from a pixel of the z-buffer rendered from the actual view point. The 3D point $\mathbf{P}_o$ is projected onto the shadow buffer, and the distance between $\mathbf{P}_o$ and the light source is also calculated simultaneously. As shown in Fig. 7.3, $\mathbf{p}_b$ is the projection of $\mathbf{P}_o$ onto the shadow buffer. The corresponding pixel to $\mathbf{p}_b$ stores the distance from the light source to the illuminated surface. If the pixel value is smaller than the distance from the light source to $\mathbf{P}_o$, $\mathbf{P}_o$ is shadowed. Z-buffer shadowing is a fast and simple algorithm, and is independent of object complexity. However, the digital errors of the z-buffer can cause:

(1) erroneous shadowing,
(2) small shadow omission, and
(3) jagginess around shadow boundaries.

To prevent wrong shadowing, the shadow buffer is often biased as shown in Fig.7.3, however, an unnecessarily large bias generates more distorted illuminations. When the shadow buffer resolution is insufficient, small and thin shadows are often missed. These mistakes are prevented by providing large shadow buffers at the cost of large storage requirements. Filtering techniques can reduce the jagginess but sometimes shadow boundaries become dull.

The aliasing artifacts in shadowing mentioned above are caused by the basic properties of the z-buffer which samples images at their pixel centers. On the contrary, most pixels around shadow contours are usually covered by both objects and shadows, thus, jagged boundaries are caused. Therefore, digital errors in shadowing and hidden surface removal must be removed.

We propose the cross scan buffer (CSB) shadowing algorithm in which the CBS is employed in place of the z-buffers in the z-buffer shadowing algorithm. Let us call the CSBs obtained for the light sources the shadow CSBs, and the CSBs obtained for the view point the viewer CSBs. As already explained in Section **7.2**, 3D shapes are divided into a set of triangles and trapezia, and then stored in a CSB. The following algorithm judges whether a piece of an object surface is shadowed or not.

(1) First, a triangle or trapezium is read from the viewer CSB. In Fig. 7.4, the gray trapezium is read.
(2) The trapezium is clipped by vertical pixel boundaries. The trapezium in Fig. 7.4 is divided to 3 trapezia, here we choose the hatched trapezium as an example. Positions of the new vertices created by the clipping are easily calculated by interpolating those of the original vertices stored in the viewer CSB.
(3) The 3D shape of each clipped trapezium, which is defined by its four vertices values in 3D space, is recovered from the vertices positions in the

viewer CSB and their distances form the view point.

(4) The recovered 3D shape is then projected onto the shadow CSB.

(5) The projected shape is clipped by triangles and/or trapezia of the shadow CSB and decomposed into several pieces. Each piece is examined to determine whether it is included in the shadow CSB or not. If not, the piece is shadowed. The reflection intensity of the piece is calculated, and then, accumulated in its related image pixel.

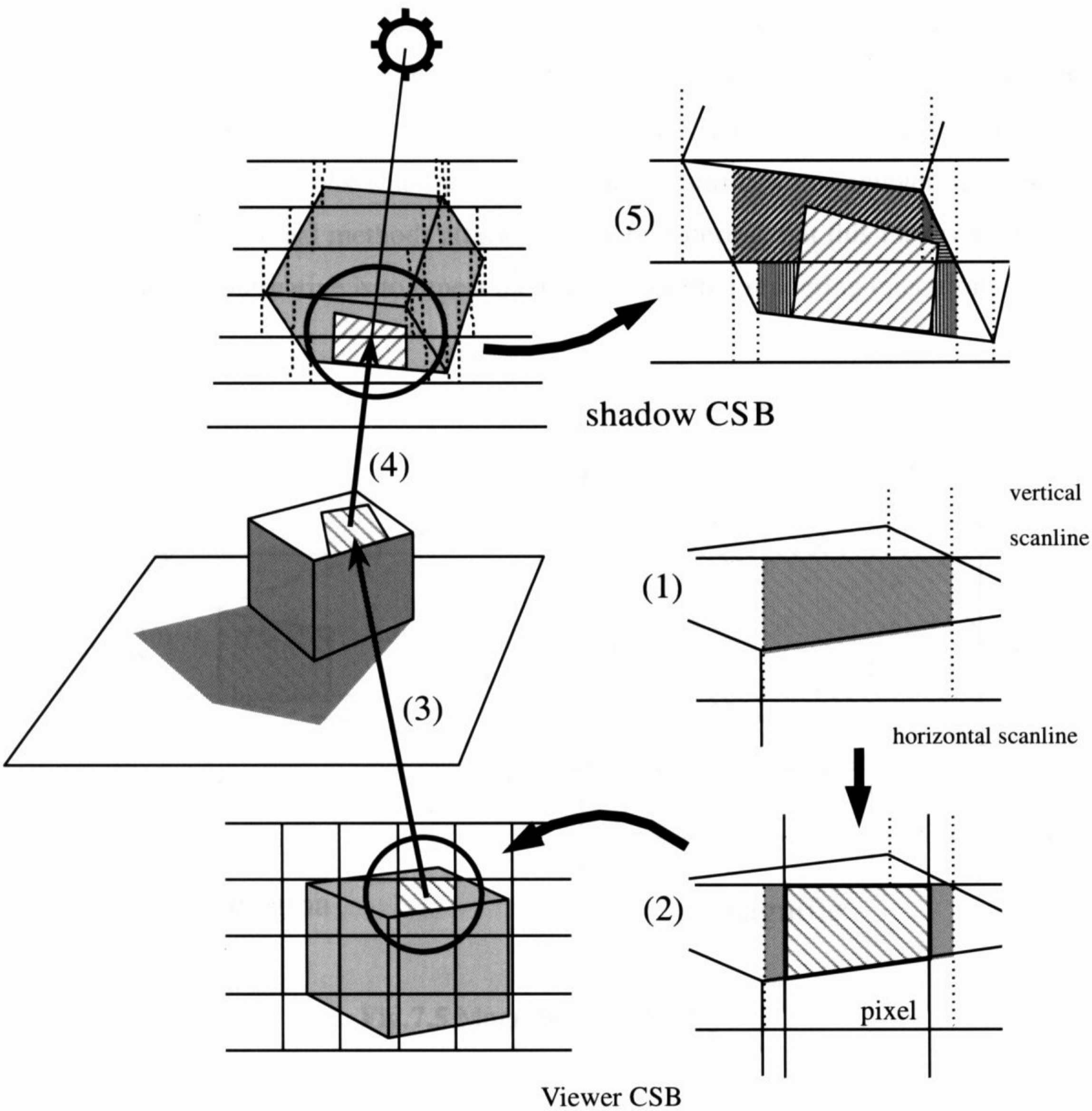


Fig.7.4 CSB-shadowing.

Since the shadow CSBs contain exact surface geometric information, accurate and aliasing-free shadowing is possible. The biasing techniques required with z-buffer shadowing are not needed.

7.3.2 Image Scaling to Arbitrary Sizes

Quick image re-scaling is not only useful, but also very practical in the trial and error process common in image synthesis. Users often want to enlarge the images in order to check rendering results in detail. Enlarged images are helpful for fine adjustment and verification of mapping coordinates. Unfortunately, aliasing artifacts become much more obvious when the magnification ratio is a non-integer such as 1.3 and 2.7. The conventional methods attack aliasing by filtering, but this makes the image blurry. The other alternative is to repeat the time consuming rendering operations.

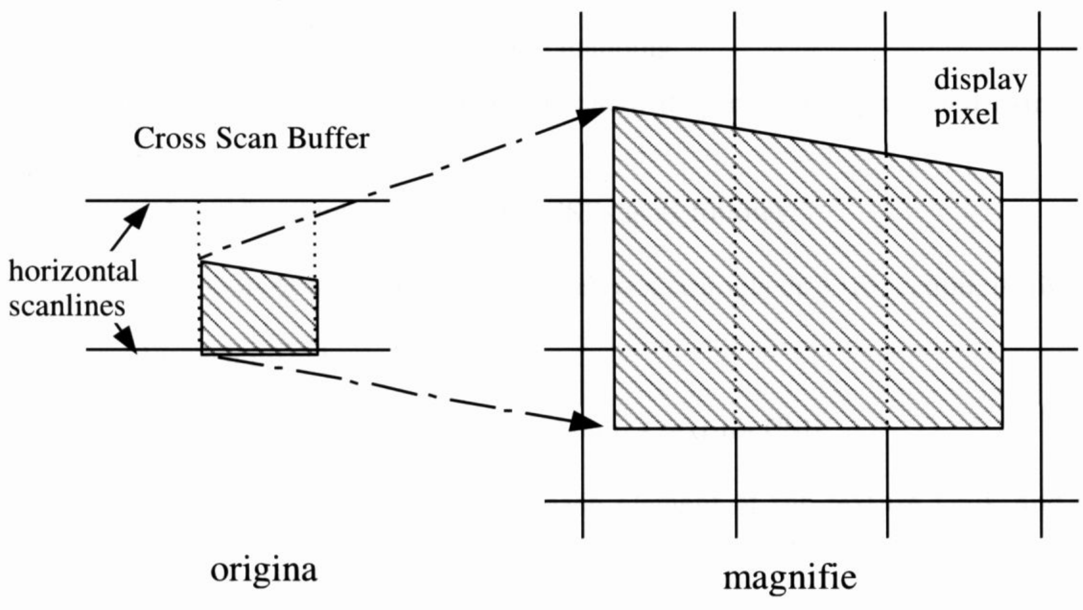


Fig.7.5 Magnified for display.

Images stored in the cross scan buffer can be quickly magnified to any degree without causing aliasing artifacts. When images are scaled up or down, most triangles and trapezia described in the CSB intersect new horizontal and vertical pixel boundaries.

As shown in Fig. 7.5, the hatched trapezium in the buffer is magnified. Since it edges cross new horizontal and vertical pixel boundaries, the magnified trapezium should be divided again by both boundaries.

Fortunately, the computing cost for this is very cheap, because the new vertical edges can be obtained by interpolation between the left and right sides of the trapezium parallel to the vertical pixel boundaries. Of course, computing time for hidden surface removal is not needed. The cross scan buffer achieves aliasing-free quick image scaling as is proved in Section **7.4**.

7.3.3 Texture Mapping

When images for texture mapping are digitized at low resolution, aliasing artifacts arise, especially when the mapping texture is magnified. High resolution texture images can reduce aliasing but extra memory costs are incurred. Aliasing in texture mapping is caused for the same reason as in z-buffer shadowing as explained before. Thus, the cross scan buffer is also applicable to texture mapping.

When texture images are computer-generated, they can be represented using the cross scan buffers named texture CSBs.

The procedure of texture mapping with CSBs is as follows:

Each element of the viewer CSB, which is a triangle or a trapezium, is divided by each pixel boundary.

Then, each divided piece is projected onto a texture CSB.

The projected piece is clipped by each element of the texture CSB, and the clipped area size is calculated.

Intensity of the texture element is multiplied by its area size.

Finally, the color value is accumulated in the image pixel related to the viewer CSB.

When a texture is defined as a numerical function or a procedural function, exact pixel values can be calculated by integrating the function in each pixel.

Texture images are often extracted from real photographs, so they are divided by pixel boundaries. They can be described in the CSB style, however, each texture CSB requires a large amount of memory. Since all texture CSB elements are the same square, we can define the texture CSB with a step function which determines the reference address of the texture image. The functional texture is partially and temporarily converted to the CSB style in the CSB mapping algorithm for clipping the viewer CSB, this significantly reduces the memory storage cost of CSB textures.

7.4 Experimental Results

We will show experimental results of the three applications introduced in Section 7.3. They demonstrate the effect and performance of the cross scan buffer.

7.4.1 Shadow Creation due to Point Light Sources

The CSB shadow algorithm effectively creates shadows with no aliasing artifacts as shown in Fig.7.6. Image resolution is 640×480 pixels, and the shadow CSB has 129 horizontal scanlines.

In this experiment, super sampling was employed to approximate the clipping process; visible surfaces stored in the viewer CSB are clipped by elements in the shadow CSB. There were 5×5 super sampled points per pixel.

Figure 7.7, a magnified image of Fig.7.6, shows that the sample points are sufficient to approximate the exact clipping process between CSBs.

Figure 7.8 is the image generated by simulating the z-buffer shadow algorithm, and then magnified to the same size as Fig.7.7. The resolution of the shadow z-buffer was set at 830×830 pixels because the shadow z-buffer occupied the same size memory of the shadow CSB. This resolution seemed sufficient to prevent aliasing artifacts, however, noticeable jaggies appear around shadow boundaries. Other artifacts are due to the self shadowing problem. That is, visible points closer to shadow boundaries cast shadows on themselves. To eliminate such artifacts, the bias technique [Reeves87] is sometimes effective, but it very difficult to adjust the bias appropriately.

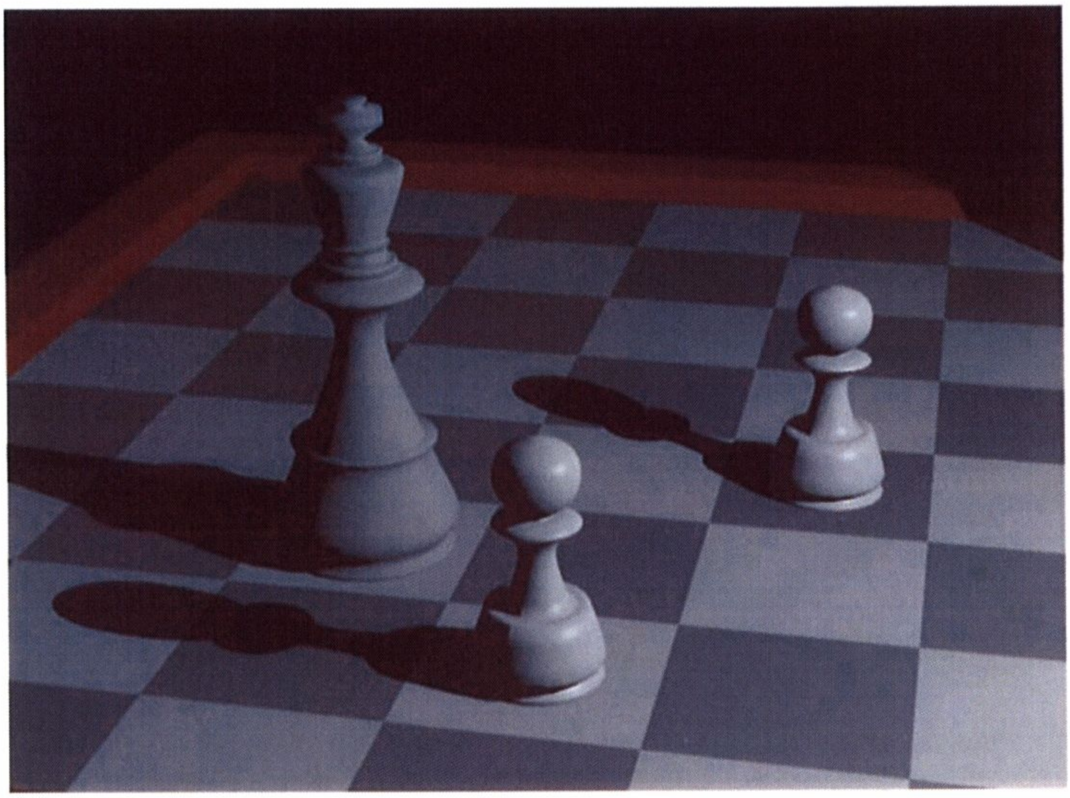


Fig7.6 CSB shadowing image.

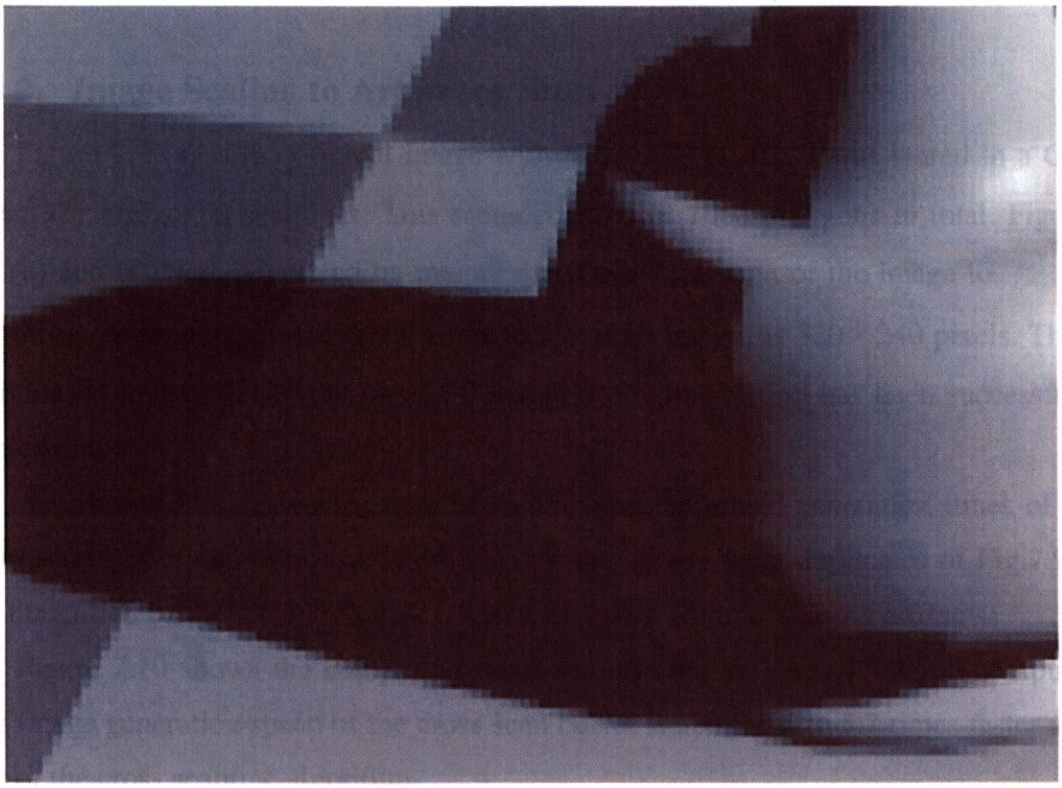


Fig7.7 CSB shadowing image magnified.

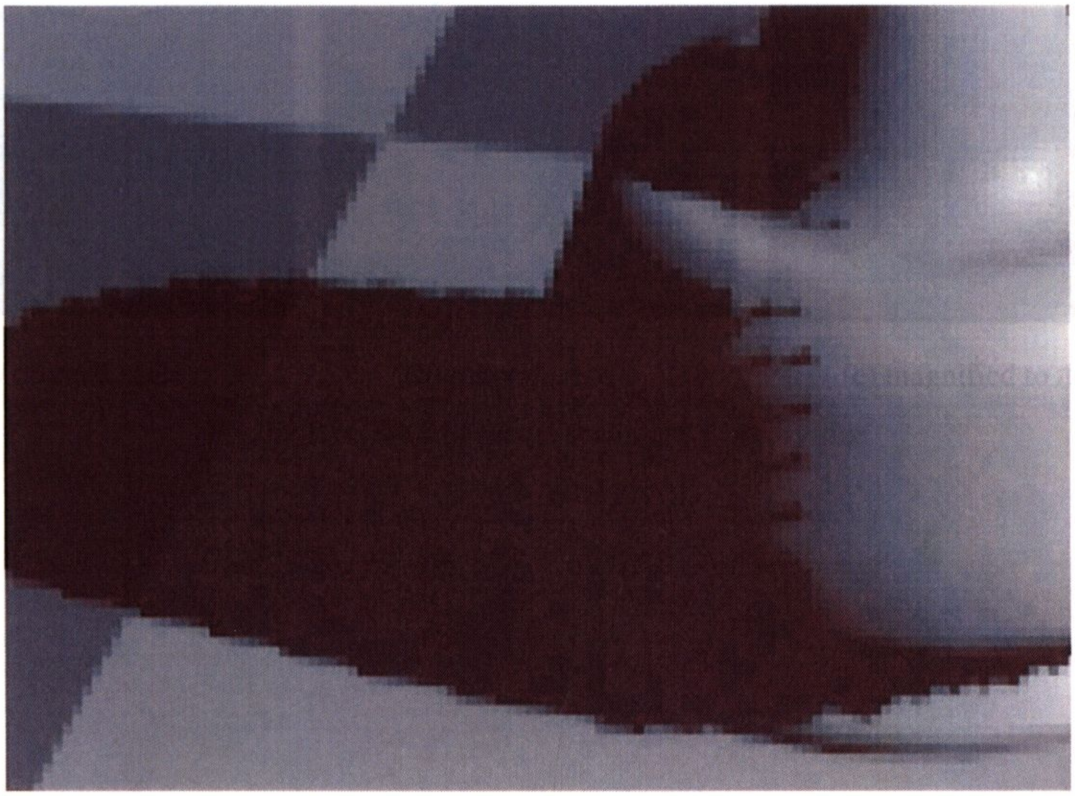


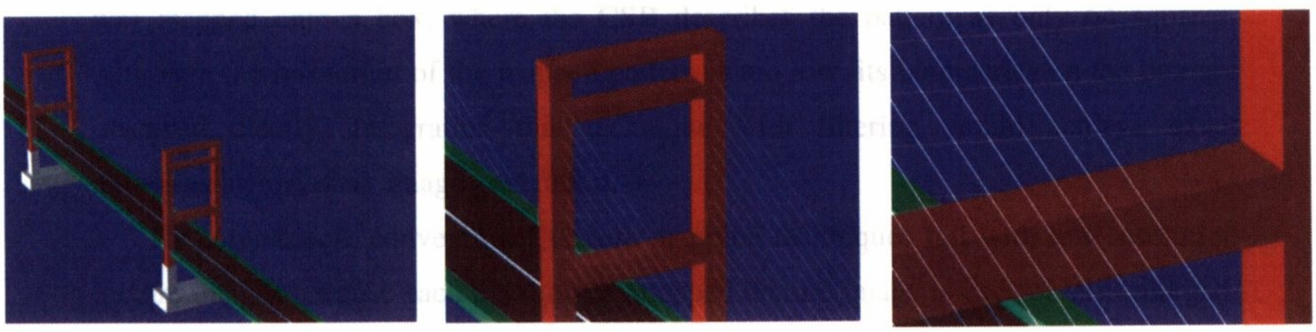
Fig7.8 Z-buffer shadowing image magnified.

7.4.2 Image Scaling to Arbitrary Sizes

Figure 7.9 (a) is the original image with 320×240 pixels. It was stored in a CSB using 321 horizontal scanlines. This scene consists of 1068 polygons in total. Figures 7.9 (b) and (c) were generated by magnifying the central area of the image to π and π^2 times, respectively. Note that the resolution of all images is 320×240 pixels. These anti-aliased images verify that the CSB image scaling algorithm has been successfully implemented.

A series of images were generated to compare the image generation times of the cross scan buffer and the cross scanline algorithm. They were duplicates of Fig.7.9(a) but magnified 0.1, 0.4, 0.7, 1.0, 1.3, 1.6, and 1.9 times with no aliasing artifacts.

Figure 7.10 shows the computing times on a 24 MIPS VAX 4000/500 computer. The image generation speed of the cross scan buffer is about 3.5 to 5.7 times faster than that of the cross scanline algorithm.



(a) original scale (b) magnified to π (c) magnified to π^2

Fig.7.9 Examples of image scaling to arbitrary size.

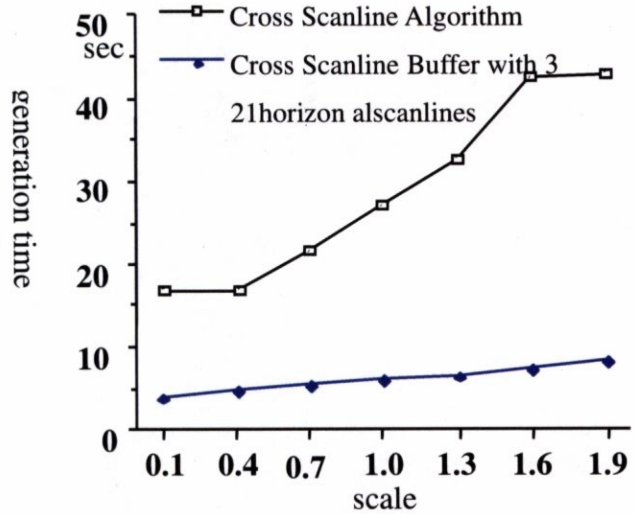


Fig.7.10 Computing speed.

7.4.3 Texture Mapping

A simple example, a checker pattern, was employed to demonstrate the effectiveness of CSB texture mapping. As shown in Fig. 7.11, an 8×8 checker pattern was mapped onto a box, where the CSB describes the pattern as only 64 squares. Although the resolution of the mapped pattern is too low, its boundaries on the box are observed clearly. Integrating this technique with filtering would achieve exact anti-aliasing and clear image generation.

By comparison, conventional texture mapping techniques fail with low resolution texture images because each pixel of the mapped texture image is excessively enlarged. This results in generated images that are very dim. Even when the checker pattern is defined as a procedural function, traditional techniques allow aliasing artifacts caused by image sampling to appear on the boundaries of the pattern.

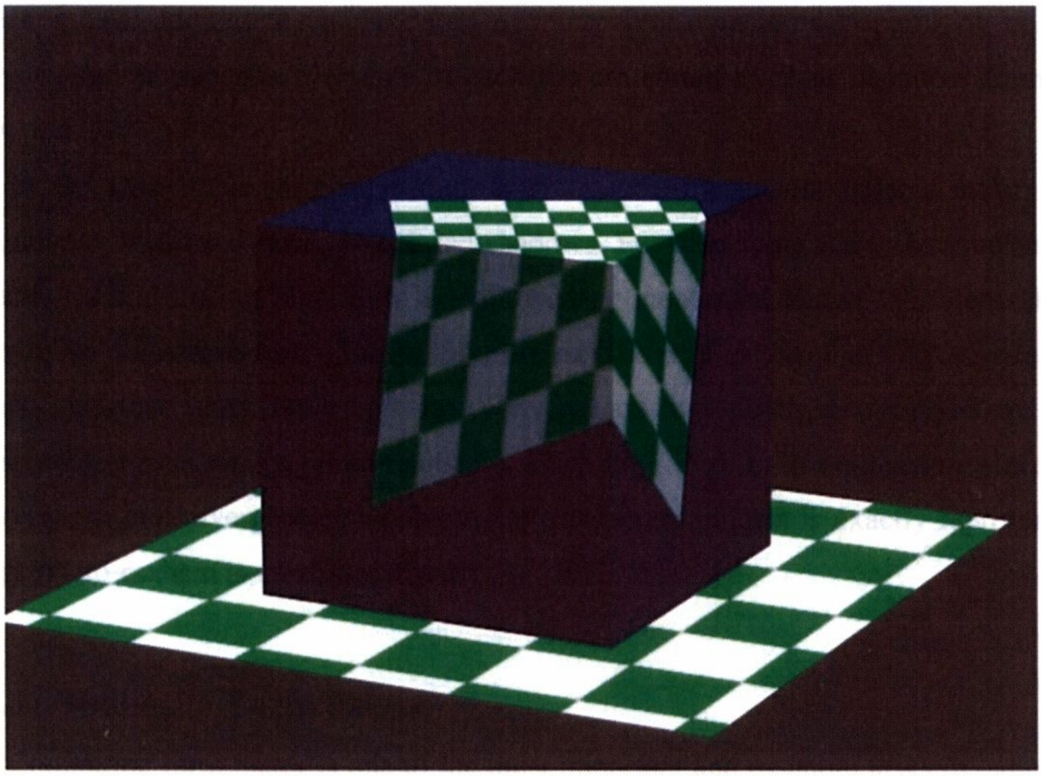


Fig.7.11 Checker pattern by CSB mapping.

7.5 Discussion

7.5.1 Ray Tracing with CSB

Several methods have been proposed to quickly determine visible surfaces and then to trace the secondary rays at the pixels where reflective and transparent objects are located. The item buffer [Weghorst84] and the ZZ-buffer [Salesin90] employ the z-buffer; the local ray tracing method [Nakamae86] employs the scanline algorithm. However, these sampled buffers cause aliasing. Our idea is to employ the cross scan buffer since the CSB inherently prevents aliasing artifacts. Surface normal vectors stored in the CSB might solve the important problem in distributed ray tracing [Cook84]; how to control ray number and the tracing directions.

7.5.2 Transparent Objects

Although the present CSB assumes opaque objects, it can be readily adapted to nonrefractive transparency, the same as scanline and list-property algorithms. The extended CSB additionally include lists of transparent objects overlapping on opaque triangles and/or trapezia. Elements of each list are sorted by their distances from the view port.

In the CSB shadowing algorithm, nonrefractive transparent objects make light illumination weak and generate colored shadows. For displaying refractive transparent objects, the ray-tracing algorithm is employed as mentioned above. The generalized perspective transformation [Shinya88] also can be applied to the CSB. Since the transformation exactly represents ray convergence and divergence caused by real transparent objects, image photo-realism is much improved. In this extension, since ray convergence and divergence caused by real transparent objects is exactly represented, image photo-realism is much improved.

7.5.3 Radiosity Form Factor

Computing form factor is one of the most important and essential tasks in radiosity, and the hemi-cube method [Cohen85] is most commonly used. Although the use of a fast z-buffer algorithm makes the hemi-cube method an efficient algorithm, it often causes aliasing artifacts due to its point sampling characteristic. Replacing the z-buffer in the hemi-cube method with the CSB would certainly prevent aliasing. Form factor of a CSB element is obtained by integrating light energy coming from the element. The integration is accurately and rapidly calculated using the table integration method for polygonal light illumination [Tanaka93].

7.5.4 Image Composition

Combining real photographs with computer generated pictures is needed to generate realistic images. Jaggies appearing around superimposed objects can be eliminated by the α blending method [Porter84], if accurate contributions of both images are given. However, its exact computation with point sampling is very difficult and expensive. The CSB gives exact the contribution ratio; it is proportional to object areas in each pixel. Thus the CSB can prevent aliasing artifacts occurring in composite images.

7.6 Conclusion

We have proposed the concept of the cross scan buffer, which preserves accurate geometric shapes of visible surfaces in a list structure. The surfaces are described using the triangles and trapezia determined by the cross scanline algorithm. We have also introduced three applications of the cross scan buffer; image re-scaling to arbitrary sizes, two-pass shadow creation, and texture mapping. The applications were realized and tested.

In image scaling, no aliasing artifacts were generated even if the magnification ratios were arbitrary real numbers. Image re-generation speed with the cross scan buffer was about 3.5 to 5.7 times faster than that with the cross scanline algorithm. Experimental results verified that the cross scan buffer is very effective yet efficient. CSB will significantly enhance the trial and error operations now used to optimize the rendering parameters.

The CSB shadowing algorithm uses the cross scan buffers to describe exact shadow spaces. Experimental results confirmed that the algorithm succeeded in preventing the jaggies that normally appear at shadow boundaries as well as the strong aliasing artifacts caused by the self shadowing problem. Quality of the images generated by the CSB shadowing algorithm was much better than those by the conventional z-buffer shadowing. The Binary Space-Partitioning (BSP) tree also can describe shadow volume analytical. Which is most efficient depends on what rendering algorithm we use. Their comparison is a future work.

The CSB mapping describes texture images with CSBs to avoid the aliasing artifacts caused by digital textures. Anti-aliasing in texture mapping was also confirmed in the experiment. The checked pattern was accurately mapped, so no jaggies appeared.

In addition, we mentioned several applications of the cross scan buffers such as ray tracing, radiosity, and image composition. Its use eliminates the aliasing artifacts caused by conventional methods.

8 Conclusions

The major achievement of this thesis is summarized into two parts:

(1) Fast and accurate shading and shadowing technologies have been developed for extended light sources including point, linear and area light sources.

The ray-oriented buffer was introduced that segments 3D space to many radial sub-space sections. The segmentation guarantees that if a point is included in a section, all light rays falling on the point are also contained in the section. Each cell of the buffer is assigned to a section and saves a list of polygons that lie within or intersect the section. Therefore, candidate polygons that may cast shadows onto a point are obtained by determining the cell where the point is located. Therefore, the ray-oriented buffer reduces the number of the candidates that occlude the light sources.

In addition, five formulas were also introduced to define the bounding-volume of shadow space caused by a polygon. By applying 5 formulas, the shadowing algorithm drastically reduces the number of candidate polygons.

Moreover, polygons are subdivided to trapezia by the cylindrical scan-conversion algorithm before they are stored in the ray-oriented buffer. The cylindrical scan-conversion algorithm reduces the cost for generating the ray-oriented buffer. The cost was sufficiently small compared to the whole image generation cost.

(2) Intermediate buffering technologies have also impacted on not only photorealistic also non-photorealistic rendering.

Especially, the concept of G-buffer has pioneered a new paradigm in computer graphics, non-photorealistic rendering. Before publishing the concept of non-photorealistic rendering paradigm by the G-buffer paper [Saito90], computer graphics researchers had been fumbling a new paradigm beyond the radiosity, *i.e.*, highly advanced photorealistic rendering technologies. The non-photorealistic rendering research has been active and attractive, so that annual international workshops on non-photorealistic rendering have been held.

The G-buffer has a powerful impact on graphics industries. ATI Technologies, Inc., one of the major graphics accelerator board company, has adopted the G-buffer as its architecture.

The G-buffer was also applied to NC machining of 3D objects. By preparing G-buffers from a parallel projection, the various functions required for a NC system were realized with image processing operations. This allows any surface description and any tool shape to be used. NTT developed a rapid prototyping system for industrial design based on the G-buffer technologies. The system was introduced to several product design companies.

The concept of the cross scan buffer was proposed, which preserves accurate geometric shapes of visible surfaces in a list structure. The surfaces are described using the triangles and trapezia determined by the cross scanline algorithm. Image re-generation speed was greatly improved so that interactive photorealistic rendering becomes available. Various kinds of applications of the cross scan buffer were also possible such as image re-scaling to arbitrary sizes, two-pass shadow creation, and texture mapping.

In the followings, the achievement of each chapter is summarized.

8.1 Ray-oriented Buffer for Linear Light Sources

A fast shadowing algorithm for linear light sources was proposed. The algorithm accomplishes the following improvements.

(1) To reduce the computing cost, the algorithm employs the ray-oriented buffer to reduce the number of candidates that occlude the light sources. The buffer segments 3D space to many sections with 2 parameters. The segmentation guarantees that if a point is included in a section, all light rays falling on the point are also contained in the section. Each cell of the buffer is assigned to a section and saves a list of polygons that lie within or intersect the section. Therefore, candidate polygons that may cast shadows onto a point are obtained by determining the cell where the point is located.

(2) In addition, our shadowing algorithm drastically reduces the number of candidate polygons by applying 5 formulas. The formulas define the bounding-volume of shadow space caused by a polygon. If even one of the formulas is false at a given

point, the polygon never casts shadows onto the point, so the polygon can be removed. Experimental results confirmed that false polygons, which are included in the candidates but do not cast shadows, were nearly completely removed.

(3) Moreover, polygons are subdivided to trapezia before they are stored in the ray-oriented buffer. This is because intersections between trapezia and the light triangle can be quickly calculated. In the experiment, light clipping cost for a trapezium was one third of the cost for the original polygon.

(4) To reduce the cost for generating the ray-oriented buffer, the cylindrical scan-conversion algorithm was introduced. The cylindrical scan-conversion processed 10,000 polygons in about 5 seconds. The cost was sufficiently small compared to the whole image generation cost.

Due to the above improvements, the proposed shadowing algorithm was over 10 times faster than the conventional algorithm. Our shadowing algorithm successfully generated pictures with realistic half shadows. Since, their generation took only a few minutes, the generation speed is acceptable.

8.2 Extended Ray-oriented Buffer for Area Light Sources

The rendering algorithm and spatial data structure for linear light sources has been entended to generate photo-realistic images containing penumbras and highlights from area light sources. For accurate shading and shadowing, exact determination of unoccluded light portions is essential. The extended method achieves fast and exact light segmentation by combining a ray-oriented buffer and cross scanline clipping.

(1) A ray-oriented buffer subdivides 3D space by following light rays and saves polygonal objects intersecting each subspace. Bounding volumes of shadow spaces caused by each polygon are also saved in the buffer. The space subdivision and the volumes together reduce the number of polygons subjected to clipping by eliminating polygons that not occlude the light sources. Experimental results confirmed that less than 1 % of all polygons were subjected to clipping. The buffer was efficiently constructed by using the Coupled Cross Ssan Conversion algorithm presented in this chapter.

(2) Cross scanline clipping splits an area light source with horizontal and vertical

scanlines in order to determine unoccluded light segments efficiently. We also proposed a fast silhouette generation algorithm that suits cross scanline clipping.

We employed our previous rendering equation for Lambertian distributive area light sources. It accurately integrates both diffuse and specular reflection using Phong's model. A sophisticated image can be created in a few minutes by the combination of the ray-oriented buffer, the cross scanline clipping, and the rendering equation even if objects cast shadows onto curved or rugged surfaces. Its speed is very practical for fine rendering.

However, there have been many fields to apply these two algorithms. For examples,

(1) The algorithms and techniques proposed here can be extended to global illumination. Cross scanline clipping is applicable to analytic form factor solutions in radiosity [Cohen93]. A combination of clipping and ray-oriented buffers would suit fast discontinuity mesh generation.

(2) The proposed algorithms were designed for polyhedral objects, however our ray-oriented buffer has the ability to handle curved objects. In the case of curved objects, the buffer description of sectors and bands will be retained. The cylindrical scan-conversion can be adopted with a small modification if bounding boxes of curved objects are given.

(3) However, a fast clipping algorithm is a remaining problem. We think Bézier clipping [Nishita90] satisfies our demand but it must be confirmed.

(4) Light sources with complex distribution properties can be rendered by adding an advanced general illumination model such as Arvo's [Arvo95], although reflection integration would become much more complex.

They are future works.

8.3 G-buffers for Non Photo-realistic Rendering

A new rendering technique has been developed that produces 3D images with enhanced visual comprehensibility. The concept and techniques proposed here can be widely applied for various purposes, not only photorealistic but also NON-photorealistic rendering. This is the first paper and pioneered a new rendering paradigm, "non-photorealistic rendering". ATI adopted G-buffer technology as the architecture of their graphics accelerator board. Several of these, edge enhancement, line drawing illustrations, topographical maps, medical imaging, and surface analysis, were presented.

Shape features can be readily understood if certain geometric properties are enhanced. To achieve this, we develop drawing algorithms for discontinuities, edges, contour lines, and curved hatching. All of them are realized with 2D image processing operations instead of line tracking processes, so that they can be efficiently combined with conventional surface rendering algorithms.

Data about the geometric properties of the surfaces are preserved as Geometric Buffers (G-buffers). Each G-buffer contains one geometric property such as the depth or the normal vector of each pixel. By using G-buffers as intermediate results, artificial enhancement processes are separated from geometric processes (projection and hidden surface removal) and physical processes (shading and texture mapping), and performed as post-processes. This permits a user to rapidly examine various combinations of enhancement techniques without excessive re-computation, and easily obtain the most comprehensible image.

8.4 G-buffer Application for NC Machining of 3D Models

The G-buffer method was applied to NC machining. By preparing G-buffers from a parallel projection, the various functions required for a NC system were realized with image processing operations. This allows any surface description and any tool shape to be used. Conventional hardware and software for computer graphics and image processing can be employed with this method. This makes NC system development much easier. Experimental results show that our method can be effectively used in an actual machining process. The method should be a great tool, not only for CAD/CAM, but also for scientific visualization.

NTT has been developed an NC machining software prototype as a rapid prototyping system for industrial design. The system was introduced to several product design companies.

8.5 Cross Scan Buffer for Interactive Photorealistic Rendering

We have proposed the concept of the cross scan buffer, which preserves accurate geometric shapes of visible surfaces in a list structure. The surfaces are described using the triangles and trapezia determined by the cross scanline algorithm. We have also introduced three applications of the cross scan buffer; image re-scaling to arbitrary sizes, two-pass shadow creation, and texture mapping. The applications were realized and tested. Image re-generation speed was greatly improved by the cross scan buffer so that interactive photorealistic rendering becomes available.

(1) In image scaling, no aliasing artifacts were generated even if the magnification ratios were arbitrary real numbers. Image re-generation speed with the cross scan buffer was about 3.5 to 5.7 times faster than that with the cross scanline algorithm. Experimental results verified that the cross scan buffer is very effective yet efficient. CSB will significantly enhance the trial and error operations now used to optimize the rendering parameters.

(2) The CSB shadowing algorithm uses the cross scan buffers to describe exact shadow spaces for point light sources. Experimental results confirmed that the algorithm succeeded in preventing the jaggies that normally appear at shadow boundaries as well as the strong aliasing artifacts caused by the self shadowing problem. Quality of the images generated by the CSB shadowing algorithm was much better than those by the conventional z-buffer shadowing.

In addition, we mentioned several applications of the cross scan buffers such as ray tracing, radiosity, and image composition. Its use eliminates the aliasing artifacts caused by conventional methods.

This thesis, especially the G-buffer, impacts on graphics industries.

ATI Technologies, Inc. [Mitchell02a], [Mitchell02b], [ATI03], one of the major computer graphics company, has adopted the G-buffer as its architecture. Their non-photorealistic rendering tools based on G-buffer have been widely utilized around the world.

Based on the G-buffer NC machining technologies, NTT has developed a rapid prototyping system using NC milling machine for industrial design. The system was introduced to several product design companies as well as institutions.

The concept of G-buffer has pioneered a new paradigm in computer graphics, non-photorealistic rendering (NPR). Since publishing the G-buffer paper [Saito-Toki90] pioneering the concept of non-photorealistic rendering paradigm, computer graphics researchers have been rushing into the new paradigm beyond highly advanced photorealistic rendering technologies such as the radiosity. International workshops on non-photorealistic rendering have been held every year since 2000 [NPRXX], and some books on NPR have already been published [] [] [].

The G-buffer has also impacted on graphics industries. ATI Technologies, Inc. [ATI03], one of the major computer graphics company, has adopted the G-buffer as its architecture.

The pixel shading unit of the RADEON□ 9700 can output up to four colors to different render targets. The ability to output to multiple render targets simultaneously allows multiple intermediate values to be saved out between rendering passes and allows for implementation of G-buffer techniques [Saito and Takahashi 1990]. An image-space outlining technique, as described by Saito and Takahashi, using multiple simultaneous pixel shader outputs is shown later in these notes and in the SIGGRAPH 2002 sketch *Real-Time Image-Space Outlining for Non-Photorealistic Rendering.*
SIGGRAPH 2002 - *State of the Art in Hardware Shading* Course Notes

NTT developed a rapid prototyping system for industrial design based on the G-buffer technologies. The system was introduced to several product design companies.

For examples,

- non-photorealistic rendering, and
- NC machining.

These two applications widely extend the possibilities of computer graphics.

Acknowledgement

I would like to thank the many people who have helped me on the path towards this dissertation during my time at NTT Laboratories and Tokyo Denki University.

I am extremely grateful for the opportunity to have had Dr. Kiyoharu Aizawa, Professor of the University of Tokyo, as a principle supervisor for this thesis. He has been an inspiration to me since I first read his distinguished paper on model-based coding technology in the 1980s.

I next thank Dr. Hiroshi Yasuda, Professor of the University of Tokyo, used to be my boss at NTT Human Interface Laboratories. I was greatly encouraged by his positive response to our inchoate research results on '*non*' photo-realistic rendering. Moreover, he pointed out that we were going to pioneer a new paradigm instead of photo-realistic rendering.

I thank the members of my reviewing committee for their time and energy in helping me improve this document: Dr. Hiroshi Harashima, Dr. Tomoyuki Nishita, Dr. Yoichi Sato, and Dr. Kenji Naemura. I have had the pleasure of learning from them in other ways as well. Dr. Nishita's thorough knowledge of the field of photo-realistic rendering, has been a valuable resource, and I appreciate his comments on my research. His insightful commentary on drafts of this entire dissertation and many previous papers has been immensely useful in helping me clearly communicate my sometimes inchoate thoughts.

My deepest gratitude is due to Dr. Mikio Shinya, Dr. Takafumi Saito and Dr. Toshimitsu Tanaka for being co-researchers, friends, and intellectual sparring partners. Our innumerable years at NTT Laboratories (sometimes called NTT *Seafood Division*) spent discussing the issues of photo-realistic rendering, visualization and related topics have been a major influence in my vision of the entire field. The new paradigm on 'non' photo-realistic rendering including the concept G-buffer was one of the most delightful results in collaboration with Dr. Takafumi Saito. The fast and analytic rendering algorithm with area light sources was the fruit of Dr. Toshimitsu Tanaka's quite unique idea to subdivide 3D space into radial sub-sectors.

Dr. Mikio Shinya has been an inspiration to me since 1984 he and I started the computer graphics project at NTT Basic Research Laboratories. What I value most about these years at NTT Laboratories was the opportunity to absorb not only his insights into the computer graphics, but his fundamental approaches to research that emphasizes rigor and first principles.

Dr. Tadashi Naruase, Mr. Masaharu Yoshida, and Dr. Seiichiro Naito have been both intellectual and personal comrades, and have also spent many years thinking on new graphics computer architecture, 'SIGHT'.

I have had the great fortune of benefiting from the wisdom, encouragement, and friendship of many former leaders at NTT Basic Research Laboratories. Dr. Shinichiro Hashimoto and Dr. Isao Masuda encouraged and initiated me into the basic research on pattern recognition with utmost care and kindness.

Thanks to many current and former people in NTT Laboratories for their friendship and supports over the years, including Keiichi Tsukada, Takahiko Kamae, Ryuichi Yasuhara, Kazuyoshi Tateishi, and Susumu Ichinose. Masahiko Tsujimoto has in particular been my comrade in arms for the last several years at NTT Cyber Solutions Laboratories.

I am thankful that many people in my field have offered both sound professional advice and welcome friendship, including Dr. Eihachiro Nakamae, Dr. Ken Anjyo, Dr. Tsuyoshi Yamamoto, Dr. Turner Whitted, Dr. Pat Hanrahan, Dr. Nelson Max, Dr. Pierre Poulin, and the late Dr. Alain Fournier.

I acknowledge, appreciate, and return the love and support of my family. My wife Michiko and my son Atsushi always share a part of my soul, and I dedicate this thesis to them.

Bibliography

[Amanatides84] J. Amanatides, "Ray tracing with cones," *Computer Graphics* (Proc. *ACM SIGGRAPH'84*), Vol.18, No.3, pp.129-135 (1984).

[Arvo87] J. Arvo, and D. Kirk, "Fast Ray Tracing by Ray Classification," *Computer Graphics* (Proc. *ACM SIGGRAPH'87*), Vol.21, No.3, pp.55-64 (1987).

[Arvo95] J. Arvo, "Applications of Irradiance Tensors to the Simulation of Non-Lambertian Phenomena," *Computer Graphics* (Proc. *ACM SIGGRAPH `95*), Vol.29, No.4, pp.335-342 (1995).

[ATI03] http://www.ati.com/developer/gdc/GDC2003_ScenePostprocessing.pdf

[Atherton87] P. Atherton, C. Earl, and C. Fred, "A Graphical Simulation System for Dynamic Five-Axis NC Verification," Proc. *Autofact '87*, pp.2-1 - 2-12 (1987).

[Bao93] H. Bao and Q. Peng, "Shading Models for Linear and Area Light Sources," *Computer & Graphics*, Vol.17, No.2, pp.137-145 (1993).

[Barros79] J. Barros, "Generating smooth 2-D monocolor line drawings on video displays," *Computer Graphics* (Proc. *ACM SIGGRAPH'79*), Vol.13, No.3, pp.260-269 (1979).

[Baum90] D. R. Baum, and J. M. Winget, "Real Time Radiosity through Parallel Processing and Hardware Acceleration," *Computer Graphics* (Proc. *1990 Symposium on Interactive 3D Graphics*), Vol.24, No.2, pp.67-75 (1990).

[Beck86] Beck, J. M., Farouki, R. T., and Hinds, J. K. "Surface Analysis Methods," *IEEE Computer Graphics and Applications*, Vol.6, No.12, pp.18-36 (1986).

[Beyond3D] http://www.beyond3d.com/articles/deflight/

[Blinn76] J. Blinn and M. Newel, "Texture and Reflection in Computer Graphics Images," *CACM*, Vol.19, No.10, pp.542-547 (1976).

[Blinn77] J. Blinn, "Methods of light reflection for computer synthesized pictures," Proc. *ACM SIGGRAPH'77*, pp.192-198 (1977).

[Blinn78] J. Blinn, "Simulation of Wrinkled Surfaces," Proc. *ACM SIGGRAPH'78*, pp.245-251 (1978).

[Carpenter84] L. Carpenter, "The A-buffer, an Antialiased Hidden Surface Method," *Computer Graphics* (Proc. *ACM SIGGRAPH '84*), Vol.18, No.3, pp.103-108 (1984).

[Catmull78] E. Catmull, "A hidden-surface algorithm with anti-aliasing," *Computer Graphics* (Proc. *ACM SIGGRAPH'78*), Vol.12, No.3, pp.6-11 (1978).

[Chappel83] I. T. Chappel, "The Use of Vectors to Simulate Material Removal by Numerically Controlled Milling," *Computer Aided Design*, Vol.15, No.3, pp.156-158 (1983).

[Chin89] N. Chin and S. Feiner, "Near Real-Time Shadow Generation using BSP Trees," *Computer Graphics* (Proc. *ACM SIGGRAPH'89*), Vol.23, No.3, pp.99-106 (1989).

[Chin92] N. Chin and S. Feiner, "Fat Object-Precision Shadow Generation for Area Light Sources using BSP Trees," *Computer Graphics* (Proc. *1992 Symposium on Interactive 3D Graphics*), Vol.26, No.2, pp.21-30 (1992).

[Cohen85] M. Cohen and D. Greenberg, "The Hemi-cube: A Radiosity Solution for Complex Environments," *Computer Graphics* (Proc. *ACM SIGGRAPH'85*), Vol.19, No.3, pp.31-40 (1985).

[Cook81] R. L. Cook and K. Torrance, "A reflectance model for computer graphics," *Computer Graphics* (Proc. *ACM SIGGRAPH'81*), Vol.15, No.3, pp.307-316 (1981).

[Cook84] R. L. Cook, "Shade Trees," *Computer Graphics* (Proc. *ACM SIGGRAPH '84*), Vol.18, No.3, 223-231 (1984).

[Cook84] R. L. Cook, T. Porter and L. Carpenter, "Distributed Ray Tracing," *Computer Graphics* (Proc. *ACM SIGGRAPH'84*), Vol.18, No.3, pp.134-145 (1984).

[Crow77] F. Crow, "Shadow Algorithm for Computer Graphics," Proc. *ACM SIGGRAPH'77*, pp.242-248 (1977).

[Crow78] F. C. Crow, "The use of grayscale for improved raster display of vectors and characters," *Computer Graphics* (Proc. *ACM SIGGRAPH'78*), Vol.12, No.3, pp.1-5 (1978).

[Crow82] F. C. Crow, "A More Flexible Image Generation Environment". *Computer Graphics* (Proc. *ACM SIGGRAPH '82*), Vol.16, No.3, pp.9-18 (1982).

[Crow84] F. C. Crow, "Summed-Area Tables for Texture Mapping," *Computer Graphics* (Proc. *ACM SIGGRAPH'84*), Vol.18, No.3, pp.207-212 (1984).

[Dickinson89] R. R. Dickinson, R. H. Bartels, and A. H. Vermeulen, "The Interactive Editing and Contouring of Empirical Fields," *IEEE Computer Graphics and Applications*, Vol.9, No.3, pp.34-43 (1989).

[Drebin88] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume Rendering," *Computer Graphics* (Proc. *ACM SIGGRAPH '88*), Vol.22, No.4 pp.65-74 (1988).

[Drettakis94] G. Drettakis and E. Fiume, "A Fast Shadow Algorithm for Area Light Sources Using Backprojection," *Computer Graphics* (Proc. *ACM SIGGRAPH '94*), Vol.28, No.4, pp.223 - 230 (1994).

[Drysdale87] R. L. Drysdale, and R. B. Jerard, "Discrete Simulation of NC Machining," Proc. *3rd Annaal ACM Conf. Computational Geometry*, pp.126-135 (1987).

[Duff85] T. Duff, "Composing 3D Rendered Images," *Computer Graphics* (Proc. *ACM SIGGRAPH'85*), Vol.19, No.3, pp.41-44 (1985).

[Feibush80] E. A. Feibush, M. Levoy and R. L. Cook, "Synthetic texturing using digital filters," *Computer Graphics* (Proc. *ACM SIGGRAPH'80*), Vo.14, No.3, pp.294-301 (1980).

[Foley90] J. Foley, A. van Dan, S. Feiner, and J. Hughes, "Computer Graphics (2nd Edition)," Addison-Wesley Publishing Company, Addition-Wesley (1990).

[Fujimoto86] A. Fujimoto, T. Tanaka, and K. Iwata, "ARTS: Accelerated Ray Tracing System," IEEE Computer Graphics and Applications, Vol.6, No.4, pp.16-26 (1986).

[Ghazanfarpour87] D. Ghazanfarpour and B. Peroche, "A Fast Antialiasing Method with A-Buffer," Proc. *Eurographics '87*, pp.503-512 (1987).

[Glassner86] A. Glassner, "Adaptive Precision of Texture Mapping," *Computer Graphics* (Proc. *ACM SIGGRAPH'86*), Vol.20, No.3, pp.297-306 (1986).

[Gooch01] B. Gooch and A. Cooch, "Non-Photorealistic Rendering", A. K. Peters, Ltd. (2001). ISBN 1-56881-133-0

[Goral84] C. Goral, K. Torrence, and D. Greenberg, "Modeling and Interaction of Light Between Diffuse Surfaces," *Computer Graphics* (Proc. *ACM SIGGRAPH '84*), Vol.18, No.3, pp.213-222

(1984).

[Heckbert86] P. Heckbert, "Filtering by Repeated Integration," *Computer Graphics* (Proc. *ACM SIGGRAPH'86*), Vol.20, No.4, pp.315-321 (1986).

[Heckbert92] P. S. Heckbert, "Discontinuity Meshing for Radiosity," Proc. 3rd Eurographics Workshop on Rendering, pp.203-216 (1992).

[Higashi83] M. Higashi, I. Kohzen, and J. Nagasaka, "An Interactive CAD System for Construction of Shapes with High-quality Surface," *Computer Applications in Production and Engineering* (Proc. *CAPE '83*), pp.371-390 (1983).

[Higashi90] M. Higashi, T. Kushimoto, and M. Hosaka, "On Formulation and Display for Visualizing Features and Evaluating Quality of Free-form Surfaces," Proc. *Eurographics '90* (1990).

[Hook86] T. V. Hook, "Real-Time Shaded NC Milling Display," *Computer Graphics* (Proc. *ACM SIGGRAPH '86*), Vol.20, No.4, pp.15-20 (1986).

[Jerard89a] R. B. Jerard, R. L. Drysdale, K. Hauck, B. Schaudt, and J. Magewick, "Methods for Detecting Errors in Numerical Controlled Machining of Sculptured Surfaces," *IEEE Computer Graphics and Applications*, Vol.9, No.1, pp.26-39 (1989).

[Jerard89b] R. B. Jerard, S. Z. Hussaini, R. L. Drysdale, and B. Schaudt, "Approximate Methods for Simulation and Verification of Numerical Controlled Machining Programs," The *Visual Computer*, Vol.5, No.6, pp.329-348 (1989).

[Kajiya86] J. T. Kajiya, "The Rendering Equation," *Computer Graphics* (Proc. *ACM SIGGRAPH'86*), Vol.20, No.3, pp.143-150 (1986).

[Kawashima89] Y. Kawashima, K. Itoh, S. Nonaka, and K. Ejiri, "A Flexible, Quantitative Method for NC Machining Verification Using a Space Division Based Solid Model," in "*New Advances in Computer Graphics*" (Proc. *CG International '89*), pp.421-437 (1989).

[Kishinami87] T. Kishinami, T. Kondo and K. Saito, "Inverse Offset Method for Cutter Path Generation", Proc. 6th *Int'l Conf. Production Engineering*, pp.807-812, (1987).

[Kondo88] K. Kondo, F. Kimura and T. Tajima, "An Interactive Rendering System with Shading," in "*Japan Annual Reviews in Electronics.*

Computers & Telecommunications," **18**, "Computer Science and Technologies," Kitagawa. T. (Ed.), Ohmsha and North-Holland, Tokyo, pp.255-271 (1988).

[Lorensen87] W. E. Lorensen, and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithms," *Computer Graphics* (Proc. *ACM SIGGRAPH '87*), Vol.21, No.4, pp.163-169 (1987).

[Mammem89] A. Mammem, "Transparency and Antialiasing Algorithms Implemented with the Virtual Pixel Maps Techniques," *IEEE Computer Graphics and Applications*, Vol.9, No.4 pp.43-55 (1989).

[Max86] N. Max, "Atmospheric Illumination and Shadows," *Computer Graphics* (Proc. *ACM SIGGRAPH '86*), Vol.20, No.4, pp.117-124 (1986).

[Mitchell02a] J. L. Mitchell, C. Brennan, and D. Card, "Real-Time Image-Space Outlining for Non-Photorealistic Rendering," ACM SIGGRAPH2002 Sketch (2002).

[Mitchell02b] J. L. Mitchell, "RADEON™ 9700 Shading," ACM SIGGRAPH 2002 Course Notes "*State of the Art in Hardware Shading*" (2002).

[Murakami90] K. Murakami and K. Hirota, "Incremental Ray Tracing," Proc. *Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pp.15-29 (1990).

[Nadas87] T. Nadas, and A. Fournier, "GRAPE: An Environment to Build Display Processes," *Computer Graphics* (Proc. *ACM SIGGRAPH '87*), Vol.21, No.4 pp.75-83 (1987).

[Nakamae86] E. Nakamae, K. Tadamura, and T. Nishita, "Half-Tone Representation Using Local Ray Tracing," *J. Information Processing Society of Japan*, Vol.27, No.11, pp.1077-1085 (1986) (in Japanese).

[Nakamae86b] E. Nakamae, K. Harada and T. Ishizaki, "A mountage method: The overlaying of the computer generated images onto a background photograph," *Computer Graphics* (Proc. *ACM SIGGRAPH '86*), Vol.20, No.4, pp.207-214 (1986).

[Nakamae89] E. Nakamae, T. Ishizaki, T. Nishita, and S. Takita, "Composing 3D Images with Anti-Aliasing and Various Shading Effects," *IEEE Computer Graphics and Applications*, Vol.9, No.2, pp.21-29

(1989).

[Newman79] W. M. Newman, and R. F. Sproull, "Principles of Interactive Computer Graphics (2nd Ed.)," McGraw-Hill (1979).

[Nishita81] T. Nishita and E. Nakamae, "An algorithm for the generation of smooth line drawings on color display," *Trans. IPS Japan*, Vol.22, No.6, pp.505-511 (1981) (in Japanese).

[Nishita83] T. Nishita, and E. Nakamae, "Half-Tone Representation of 3-D Objects Illuminated by Area Sources or Polyhedron Sources," *IEEE Computer Society's 7th International Computer Software & Applications Conference* (Proc. *COMPSAC*), pp.237-242, (1983).

[Nishita84] T. Nishita, K. Harada and E. Nakamae, "Half-tone representation of 3-D objects with smooth edges by using a multi-scanning method," *Trans. IPS Japan*, Vol.25, No.5, pp.703-711 (1984) (in Japanese).

[Nishita85a] T. Nishita, I. Okumura, and E. Nakamae, "Shading Models for Point and Linear Sources," *ACM Transaction on Graphics*, Vol.4, No.2, pp.124-146 (1985).

[Nishita85b] T. Nishita and E. Nakamae, "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection," *Computer Graphics* (Proc. *ACM SIGGRAPH'85*), Vol.19, No.3, pp.23-30 (1985).

[Nishita90] T. Nishita, T. Sederberg, and M. Kakimoto, "Ray Tracing Rational Trimmed Surface Patches," *Computer Graphics* (Proc. *ACM SIGGRAPH'90*), Vol.24, No.4, pp.1337-345 (1990).

[Perlin85] K. Perlin, "An Image Synthesizer", *Computer Graphics* (Proc. *ACM SIGGRAPH '85*), Vol.19, No.3, pp.287-296 (1985).

[Porter84] T. Porter, and T. Duff, "Compositing Digital Images," *Computer Graphics* (Proc. *ACM SIGGRAPH '84*), Vol.18, No.3, pp.253-259 (1984).

[Poulin90] P. Poulin, and J. Amanatides, "Shading and Shadowing with Linear Light Sources," *Proc. Eurpgraphics '90*, pp.377-386 (1990).

[Reeves87] W. Reeves, D. Salesin, and R. L. Cook, "Rendering Anti-aliased Shadows with Depth Maps," *Computer Graphics* (Proc. *ACM SIGGRAPH'87*), Vol.21, No.3, pp.283-291 (1987).

[Rosenfeld82] A. Rosenfeld and A. C. Kak. "Digital Picture Processing (2nd

Ed.)," Academic Press, (1982).

[Saito89] T. Saito, M. Shinya, and T. Takahashi, "Highlighting Rounded Edges," In "*New Advances in Computer Graphics*" (Proc. *CG International '89*), pp.613-629 (1989).

[Saito90] T. Saito and T. Takahashi, "Comprehensible Rendering of 3-D Shapes," *Computer Graphics* (Proc. *ACM SIGGRAPH '90*), Vol.24, No.4, pp.197-206 (1990).

[Sakuta87] T. Sakuta, M. Kawai, and Y. Amano, "Development of an NC Machining System for Stamping Dies of Offset Surface Method", Proc. *Autofact '87*, pp.2-13 - 2-27 (1987).

[Salesin90] D. Salesin and J. Stolfi, "Rendering CSG Models with a ZZ-buffer," *Computer Graphics* (Proc. *ACM SIGGRAPH '90*), Vol.24, No.4, pp.67-76 (1990).

[Shinya87] M. Shinya, T. Takahashi, and S. Naito, "Principal and Applications of Pencil Tracing," *Computer Graphics*, (Proc. *ACM SIGGRAPH'87*), Vol.21, No.3, pp.45-54 (1987).

[Shinya89] M. Shinya, T. Saito, and T. Takahashi, "Rendering Techniques for Transparent Objects," Proc. *Graphics Interface'89*, pp.173-182 (1989).

[Stewart94] A. J. Stewart and S. Ghali, "Fast Computation of Shadow Boundaries Using Spatial Coherence and Backprojections," *Computer Graphics* (Proc. *ACM SIGGRAPH'94*), Vol.28, No.4, pp.231-238 (1994).

[Strothtte98] T. Strothtte, "Computational Visualization – Graphics, Abstraction, and Interactivity," Springer-Verlag (1998). ISBN 3-540-63737-0

[Strothtte02] T. Strotholle, and S. Schlechtweg, "Non-Photorealistic Computer Graphics – Modeling, Rendering, and Animation," Morgan Kaufmann Publishers (2002). ISBN 1-55860-787-0

[Sung92] K. Sung, "Area sampling buffer: tracing rays with z-buffer hardware," Proc. *Eurographics'92*, pp.C-299-C-310 (1992).

[Tanaka90] T. Tanaka and T. Takahashi, "Cross Scanline Algorithm," Proc. *Eurographics'90*, pp.63-74 (1990).

[Tanaka92] T. Tanaka and T. Takahashi, "Precise Rendering Method for Exact Anti-aliasing and Highlighting," *The Visual Computer*, Vol.8, No.4, pp.315-326 (1992).

[Tanaka93] T. Tanaka and T. Takahashi, "Fast Analytic Rendering with

Accurate Table Integration," Proc. *Computer Graphics International'93*, pp.127-138 (1993).

[Udupa89] J. K. Udupa, "Display of Medical Objects and their Interactive Manipulation". Proc. *Graphics Interface '89*, (1989), 40-46.

[Verbeck84] C. P. Verbeck and D. P. Greenberg, "A Comprehensive Light Source Description For Computer Graphics," *IEEE Computer Graphics and Applications*, Vol.4, No. 7, pp. 66-75 (1984).

[Wallace87] J. R. Wallace, M. F. Cohen, and D. P. Greenberg, "A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods," Computer Graphics (Proc. ACM SIGGRAPH'87), Vol.21, No.3, pp.311-320 (1987).

[Wang86a] W. P. Wang, , and K. K. Wang, "Real-Time Verification of Multiaxis NC Programs with Raster Graphics", Proc. *IEEE Int'l Conf. Robotics and Automation*, pp.166-171 (1986).

[Wang86b] W. P. Wang, , and K. K. Wang, "Geometric Modeling for Swept Volume of Moving Solids", *IEEE Computer Graphics and Applications*, Vol.6, No.12, pp.8-17 (1986).

[Ward94] G. J. Word, "The RADIANCE Lighting Simulation and Rendering System," *Computer Graphics* (Proc. *ACM SIGGRAPH'94*), Vol.28, No.3, pp.459-472 (1994).

[Weghorst84] H. Weghorst, G. Hooper, and D. Greenberg, "Improved Computational Methods for Ray Tracing," *ACM Transaction on Graphics*, Vol.3, No.1, pp.52-69 (1984).

[Whitted80] T. Whitted, "An Improved Illumination Model for Shaded Display," *CACM*, Vol.23, No.6, pp.343-349 (1980).

[Whitted81] T. Whitted and D. Weimer, "A Software Test-Bed for the Development of 3D Raster Graphics Systems," *Computer Graphics* (Proc. *ACM SIGGRAPH'81*), Vol.15, No.3, pp.271-277 (1981).

[Whitted82] T. Whitted and D. M. Weimer, "A Software Testbed for the Development of 3D Raster Graphics Systems". *ACM Trans. Graphics*, Vol.1, No.1, pp.43-58 (1982).

[Williams78] L. Williams, "Casting Curved Shadows on Curved Surfaces," *Computer Graphics* (Proc. *ACM SIGGRAPH'78*), Vol.12, No.3, pp.270-274 (1978).

[Woo90] A. Woo, P. Poulin, and A. Fournier, "A Survey of Shadow

Algorithms," *IEEE Computer Graphics and Applications*, Vol.10, No.11, pp.13-32 (1990).

[Wylie67] C. Wylie, G. Romney, D. Evans, and A. Erdahl, Half-tone Perspective Drawings by Computer, *AFIPS* Conf. Proc., Vol.31, pp.49-54 (1967).

[Zhang86] D. Zhang, and A. Bowyer, "CSG Set-Theoretic Solid Modeling and NC Machining of Blend Surfaces", Proc. *2nd Annual ACM Conf. Computational Geometry*, pp.236-245, (1986).

Appendix A:
Analytic Hidden Surface Removal

This chapter proposes a new hidden surface removal algorithm which is based on the scanline algorithm but scans in two directions, horizontally and vertically. Named the cross scanline algorithm, it can efficiently detect all polygons and calculate their exact projected areas in each pixel even if the polygons are much smaller than the pixel. Perfect anti-aliasing is achieved because the the cross scanline algorithm analytically determines visible polygonal surfaces and divides them into sets of triangles and trapezia. Comparisons with the regular sub-scanlines algorithm show that high quality anti-aliased images can be generated.

A.1 Introduction

Computers can easily create extremely complicated images, but there are many problems in displaying them accurately. One of the most important problems is aliasing which significantly degrades image quality. Thus, many anti-aliasing methods have been proposed [Nishita81], [Nishita84], [Nakamae86], [Crow78], [Catmull78], [Barros79], [Feibush80]. Aliasing effects are commonly called jaggies.

Jaggies appear at pixels forming polygon edges because several polygons, including the background, can fall on boundary pixels. If pixel values are determined in proportion to the regions of the covering polygons, jaggies can be corrected. Therefore, to generate highly anti-aliased images, a hidden surface removal algorithm which can calculate exact polygon regions is necessary.

The region which a polygon occupies in a pixel can be calculated approximately by multi-sampling algorithms [Nishita81], [Nishita84], [Nakamae86], [Crow78]. The algorithms search the nearest polygon at some points or several scanlines per pixel. Their accuracy can be improved by increasing the number of sampling points or lines. However, the computing time would also increase, and smaller polygons than the sampling pitch are often missed.

To generate highly anti-aliased images, we propose the cross scanline algorithm which uses horizontal and vertical scanlines. The algorithm detects all polygons and accurately calculates their areas even if they are much smaller than the pixel.

This chapter discusses standard anti-aliasing algorithms using scanlines, details of the cross scanline algorithm, and computer experiments that confirm the accuracy and speed of the cross scanline algorithm.

A.2 Anti-aliasing for scanline algorithm

There are two kinds of anti-aliasing methods for the standard scanline algorithm.

The first includes the clipping algorithms [Catmull78], [Barros79], that determine correct polygon regions. The quality of their images is inherently good, however, their computing times are too long because of the recursive clipping needed for each pixel.

The second method is the algorithm which locates scanlines on horizontal pixel boundaries and calculates polygon areas by estimating' the polygons as trapezoids or triangles clipped by the scanlines. The algorithm calculates the areas exactly, but only if the polygon edges do not cross each other or polygon vertices do not exist between adjacent scanlines. The estimation errors caused by vertices and edge crossing points are shown in Fig.A.1(a) with hatching.

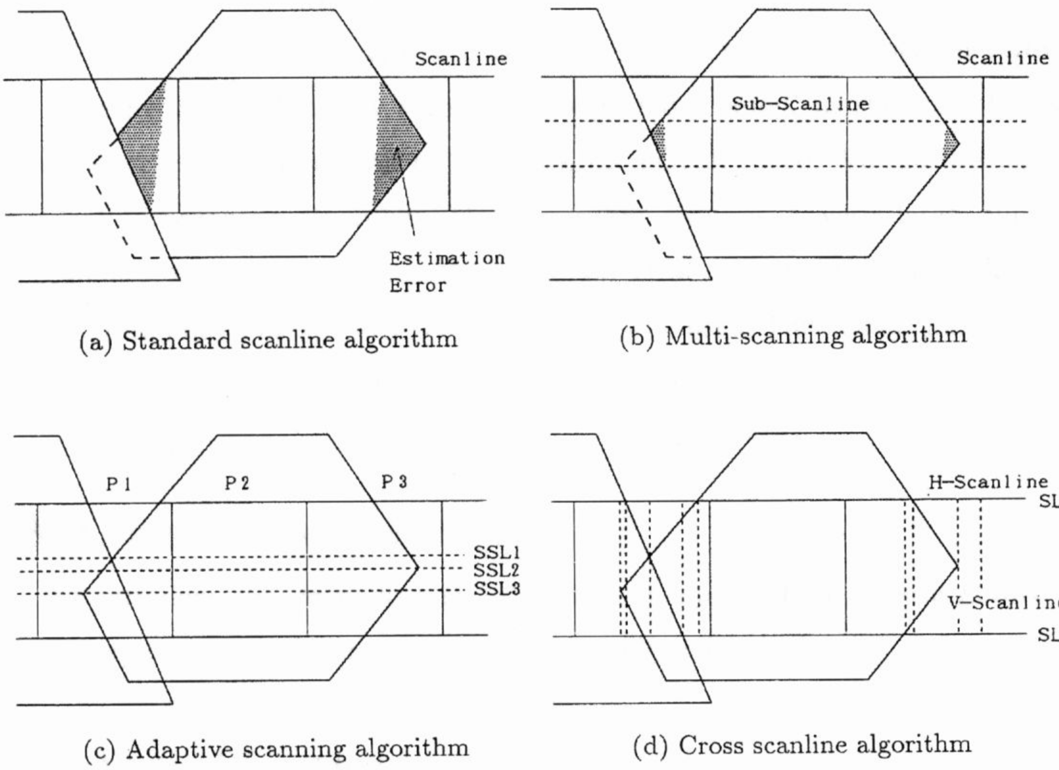


Fig.A.1 Scanline algorithms and their estimation errors.

A.2.1 Multi-scanning algorithm

The estimation errors can be reduced by sampling more lines per pixel. This algorithm is called the multi-scanning algorithm [Nishita84], [Nakamae86], and the lines inserted between the original scanlines are called sub-scanlines. The estimation errors of the multi-scanning algorithm are decreased by increasing the number of sub-scanlines. For example, the reduction in error by three times over scanning, which means scanning three evenly spaced lines per pixel, is evident in Fig.A.1(b).

The multi-scanning algorithm still has the problem that jaggies on horizontal edges are much stronger and more frequent than on vertical edges. This is because the vertical resolution, which depends on the number of sub-scanlines, is much lower than the horizontal resolution. Of cause, using a lot of sub-scanlines strengthens anti-aliasing, but the computing cost is excessive.

A.2.2 Adaptive scanning algorithm

By analyzing estimation errors, it is shown that they occur at polygon vertices or edge crossing points. Thus, by scanning through those points, as shown in Fig.A.l(c), errors are prevented, and correct polygon regions are calculated. We call this algorithm the adaptive scanning algorithm. The algorithm requires a very large number of sub-scanlines to completely remove estimation errors. If an image includes 10,000 vertices, a minimum of 10,000 sub-scanlines must be used. This means that an average of about 20 lines must be scanned per pixel, if the image is 512×512.

A.3 Cross scanline algorithm

We introduce the cross scanline algorithm in this section for calculating exact polygon regions with a minimal increase in computing time.

A.3.1 Vertical Scanning

The problem of the adaptive scanning algorithm is that extremely many sub-scanlines are required to split polygons. This separation is only necessary for pixels which contain a vertex or edge crossing point. For example sub-scanline $\mathbf{SSL}_1$ in Fig.A.1(c) is essential for pixel $\mathbf{P}_1$, but not for $\mathbf{P}_2$ or $\mathbf{P}_3$. Since the algorithm creates sub-scanlines that extend across the image, many pixels are examined unnecessarily.

Even if the length of sub-scanlines is limited to one pixel, for example, sub-scanline $\mathbf{SSL}_h$ in Fig.A.2(a) is scanned from point **E** to **F**, the computing cost is not significantly reduced. This is because, (1) all polygons (include hidden polygons) existing at **E**, and (2) all edges intersecting to the line segment **EF**, must be given to determine polygons falling on **EF**. The polygons and edges are obtained by searching all polygons and edges on $\mathbf{SSL}_h$. Since this searching is necessary on each horizontal sub-scanline, the computing cost is not greatly reduced.

If vertical sub-scanlines are drawn across a pixel as shown in Fig.A.2(b), the previous operations (1) and (2) are replaced with operation (3) all polygons at point **G** and operation (4) all edges intersecting **GH**. The polygons (operation 3) will have already been detected in $\mathbf{SL}_1$ scanning. Thus, determining them requires no extra computing, if they are stored.

The cost for operation (4) is also small, because the edges can be easily gained from the edges falling on the previous vertical sub-scanline **G'H'** (if they are included), or adding them to the list (if not included);

i) $\mathbf{E_1}$, edges intersecting the line segment **G'G**.

ii) $\mathbf{E_2}$, edges intersecting the line segment **H'H**.

iii) E3, edges whose vertices fall in the box **G'H'HG**.

$\mathbf{E_1}$ and $\mathbf{E_2}$ can be determined from the edges sorted by the x values of their intersection with $\mathbf{SL_1}$ and $\mathbf{SL_2}$, respectively. $\mathbf{E_1}$ and $\mathbf{E_2}$ will have been detected in scanning $\mathbf{SL_1}$ and $\mathbf{SL_2}$. $\mathbf{E_3}$ are determined from the list of vertices lying between $\mathbf{SL_1}$ and $\mathbf{SL_2}$ as sorted by their x values. $\mathbf{E_1}$, $\mathbf{E_2}$, and $\mathbf{E_3}$ for all vertical sub-scanlines are determined by searching the lists just once. Thus, the computing cost is minimized. However, this method does not work well for limited horizontal sub-scanlines, since not only x but also the y values of their end points vary.

The cross scanline algorithm uses the horizontal pixel boundaries as horizontal scanlines or **H**-scanlines. The vertical sub-scanlines are one pixel in height and are called **V**-scanlines.

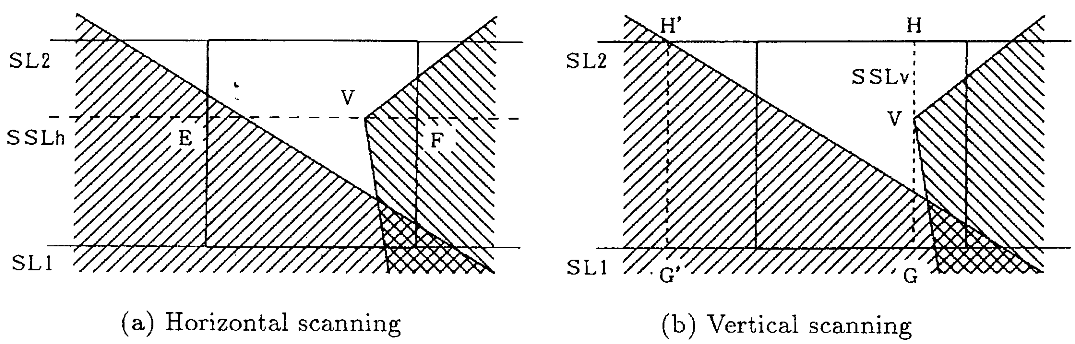


Fig.A.2 Limited sub-scanlines in one pixel.

A.3.2. Algorithm

The **H**-scanlines are located at each horizontal pixel boundary. The **V**-scanlines are scanned between adjoining **H**-scanlines. As shown in Fig.A.1(d), they are located at;

(1) intersection of edges and the upper **H**-scanline (**SL_1**),

(2) intersection of edges and the lower **H**-scanline (**SL_2**),

(3) vertices existing between the **H**-scanlines,

(4) intersecting points between edges, and

(5) right and left sides of the image.

The cross scanline algorithm has four operations.

[Begin]

(1-1) Make the edge list for **H**-scanning.

(1-2) Set **H**-scanning position to the bottom of an image.

[H-Loop]

(2-1) On the **H**-scanline, remove hidden surfaces by using the standard scanline algorithm.

(2-2) Save the sorted edge list for the **H**-scanline.

(2-3) Locate **V**-scanlines at the previous (1)-(5) points.

(2-4) Start **V**-scanning position at the left side of the image.

[V-Loop]

(3-1) On each **V**-scanline, remove hidden surface.

(3-2) Calculate polygon areas between the adjoining **V**-scanlines.

(3-3) Move the **V**-scanning position to the next **V**-scanline.

[Next H-position]

(4-1) Move the **H**-scanning position to the next upper horizontal pixel boundary.

[End]

In the cross scanline algorithm, the method of removing hidden polygons is the same as in the standard scanline algorithm.

A.4 Anti-aliasing for polygon intersections

The cross scanline algorithm can handle intersecting polygons, and the intersections are also completely anti-aliased. The V-scanlines are located on the assumption that each polygon region on an image is bounded by polygon edges. Non-intersecting polygons satisfy the assumption even if they overlap each other.

However, the assumption fails for intersecting polygons, because their projections onto the image are often bounded by their intersections. To solve this problem, operations (2-1a) and (3-1a) are added after (2-1) and (3-1) in Section A.3.2.

(2-1a) If polygon intersections are detected in (2-1), the intersection lines ale added to the edge list as temporary edges.

(3-1a) If polygon intersections are detected in (3-1) and the intersection lines have not been included in the edge list yet, they are added to the list and new V-scanlines are located at the intersections of the added edges and existing edges.

By these operations, the assumption becomes correct, and exact polygon regions can be calculated. The temporary edges are considered original edges except for the property that they are polygon outlines.

A.5 Conclusion

The cross scanline algorithm offers a powerful anti-aliasing effect. The algorithm uses horizontal and vertical scanlines and rapidly calculates exact polygon regions in each pixel. Since it is not necessary to chose the optimum number of sub-scanlines, recursive image generation caused by under estimating aliasing is prevented.

Computer experiments show that anti-aliasing with the cross scanline algorithm does not depend on edge direction. Even if an edge is nearly horizontal, it is extremely anti-aliased. This virtually impossible with the multi-scanning algorithm. The cross scanline algorithm smoothly displays lines, which are much thinner than the pixel size. The images created by the cross scanline algorithm are much better than the images from the multi-scanning algorithm. The computing speed of the cross scanline algorithm is higher than that of the multi-scanning algorithm to create highly anti-aliased images.

It is easy to combine the cross scanline algorithm with filtering operations. Moreover, the close scanline algorithm is useful in texture mapping and smooth shading of thin or small polygons. These subjects will be discussed in future works.

Appendix B:
Extended Rendering Equation

B.1 Introduction

This appendix derives a shading model for area light sources which covers both diffuse and specular reflection. The shading model assumes ideally diffuse polygonal light sources and uses Phong's reflection model. The model can accurately integrate the intensities of diffuse and specular reflection without simulating an area light source as an array of point light sources.

B.2 Background

Enhanced shading models are extremely important in computer graphics, especially for photo-realistic rendering. It is commonly accepted that separating reflection into diffuse and specular terms is a reasonable approximation. Several shading models based on this approximation have been developed such as Phong [Phong75], Blinn [Blinn77] and Cook-Torrance [Cook81]. When object surfaces are illuminated by point sources, both diffuse and specular reflections can be calculated with these methods. Several shading models expressing linear and area sources have been developed.

Nishita *et al.* [Nishita85] developed a system to handle linear light sources such as fluorescent tubes. Their method derives an analytical solution of diffuse reflection. For fast computation, each linear light source is classified into one of three cases: parallel or perpendicular to the illuminated surface, or skewed. For the first and second cases, two simple solutions are introduced. For skewed sources, the integration of diffuse reflection is approximated as a polynomial. However, their system can not handle specular reflection. To generate specular reflection, the linear light source must be converted to a series of point light sources. Therefore, this system can not escape from sampling problems.

Poulin *et al.* [Poulin90] employ Phong's model to integrate specular reflection from linear light sources. They simplified the integration by ingeniously transforming the coordinate system, and the integration is approximated with Chebyshev polynomials. Since this method can calculate the intensity of specular reflection accurately, sampling problems can be prevented.

Nishita *et al.* [Nishita83] introduced a method for shading and shadowing from area light sources. Light sources are defined as light emitting polygons which follow Lambert's law. Illuminated object surfaces are assumed to exhibit ideal diffuse reflectivity. They use the contour integration method for calculating the direct illuminance. Their method yields an analytical equation, however specular reflection is disregarded.

In Verbeck's system [Verbeck84], an area light source is simulated as an array of point light sources. Area source illumination is calculated by summing the contributions of each point source. This system accepts several reflection models, since illumination by point sources is simple. However, too few point light sources causes highlight roughness. Since accurate rendering usually requires dozens of point sources, its computing cost is very high.

Another approach samples the direction of incident light. Radiosity[Goral84] schemes using hemi-cube[Immel86] expansion techniques can calculate the specular reflection caused by area light sources[Shao88][Sillion89]. However, rendering sharp reflection surfaces requires a long computing time because hemi-cube resolution must be high. Computing cost of distributed ray tracing [Cook84] is also expensive. Although cone tracing[l] and pencil tracing [Shinya87] accept area light sources, their direct application is very complex.

For accurate photo-realistic rendering, an analytic solution to calculate the reflected intensity of surfaces with various roughness coefficients is necessary.

B.3 Shading with Polygonal Light Sources

In this section, we introduce an improved shading model that accurately integrates both diffuse and specular reflection with area light sources. Our model assumes Lambertian distributed light sources and Phong's reflection model.

B.3.1 Lambertian Distributed Light Source

A polygonal light source S is defined by a string of vertices V_1, V_2, ..., V_m as shown in Fig.A.1. It radiates light to the side where the vertices are viewed in a clockwise direction. If the light source is Lambertian distributive, flux dF_i coming from the emitting area dS to point **P** is calculated by *Equation* B.1. Symbols in the following discussion are defined in Terminology B.1.

$$dF_i = (\mathbf{N}_Q \cdot \mathbf{L}_Q)/\, r^2 I_o\, dS = \cos \Psi / r^2 I_o\, dS \qquad \text{(B.1)}$$

If the solid angle of dS viewed from **P** is described as $d\omega$, dS is related to y in *Equation* B.2. Thus, flux dF_i is given by *Equation* B.3.

$$\cos \Psi\ dS = r^2\, d\omega \qquad \text{(B.2)}$$

$$dF_i = I_o\, d\omega \qquad \text{(B.3)}$$

B.3.2 Modification of Phong's Reflection Model

Reflection models have been proposed by Phong[Phong75], Blinn[Blinn77], and Cook-Torrance[Cook81]. All of them can handle specular reflection, however, we employed Phong's model because its specular reflectivity is rotational symmetrical to vector $\mathbf{E}_r$ in Fig.B.1. This nature simplifies the integration of specular reflection. Phong's model is defined by *Equations* B.4 – B.6. Referenced symbols are defined in Terminology B.1.

$$Is = k_d I_d + ksIs \qquad \text{(B.4)}$$

$$I_d = R_d\, (\mathbf{N}_p \cdot \mathbf{L}_p)\, I_i \qquad \text{(B.5)}$$

$$I_s = R_s\,(\mathbf{E}_r \cdot \mathbf{L}_p)^n\, I_i \tag{B.6}$$

Phong's reflection model relates incident and reflection intensity. However, to integrate reflection from area light sources, the relation between incident flux and reflection intensity is necessary. Thus, we modified Phong's model. Assume that a surface is illuminated from its surface normal direction with constant flux F_i If the specular reflection of flux also obeys Phong's model, the total flux of specular reflection F_s can be calculated by Equation B.7.

$$F_s = \int_0^{2\pi}\int_0^{\pi/2} I_s \sin\theta\, d\theta\, d\phi = R_s F_i \int_0^{2\pi}\int_0^{\pi/2} \cos^n\theta \sin\theta\, d\theta\, d\phi = R_s F_i \frac{2\pi}{n+1} \tag{B.7}$$

From energy conservation, F_s must be equal to F_i when $R_s = 1$. Thus, Phong's model was normalized. We employ *Equation* B.8 to calculate the intensity of specular reflection with small incident flux dF_i.

$$dI_s = R_s\,(n+1)/2\pi\,\, (\mathbf{E}_r \cdot \mathbf{L}_p)^n\, dF_i \tag{B.8}$$

B.4 Remarks

A computation method has also proposed for the integration of specular reflection [Tanaka91]. Each light source is projected onto a unit sphere and triangulated by great circles of the unit sphere. Then each triangulated source is transformed from the Cartesian to the polar coordinate system to simplify the reflection integration. Finally, specular reflection is integrated by using the Chebyshev polynomial approximation. Several test images have shown that the proposed method can generate higher quality photo-realistic images than conventional methods. The method completely prevents highlight roughness.

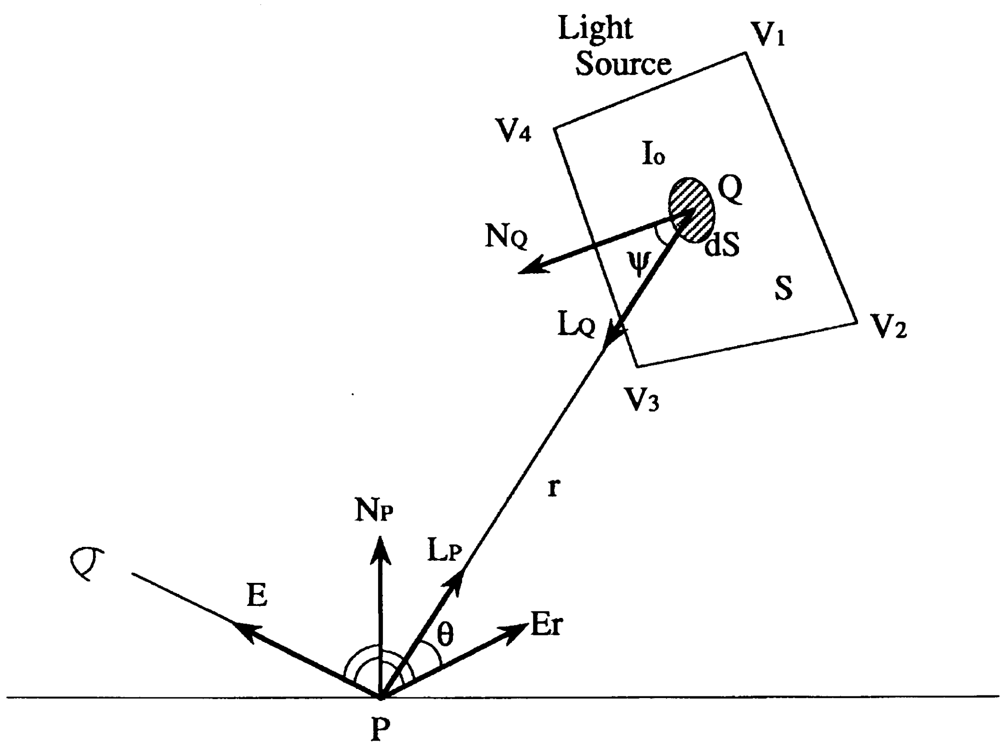


Fig.B.1 Reflection Model

Terminology 1

E: unit vector in the direction of the viewer

E_r: unit vector of mirror reflection of E

I_i: intensity of the incident light

I_o: radiation intensity of S

I_r: intensity of the reflected light

I_d: intensity of diffuse reflection

I_s: intensity of specular reflection

k_d: fraction of reflectance that is diffuse

k_s: fraction of reflectance that is specular ($k_d + k_s = 1$)

L_P: unit vector in the direction of light

L_Q: unit vector from Q to P ($= -L_p$)

n: index of reflection

N_P: unit surface normal at P

N_Q: unit surface normal at Q

r: distance between P and Q

R_d: diffuse bidirectional reflectance

R_s: specular bidirectional reflectance

S: polygonal light source

dS: small area on S

ψ: angle between N_Q and L_Q

dF_i: flux coming from dS

Academic Achievements

Refereed Papers

Computer Graphics

(1) Mikio Shinya, Tokiichiro Takahashi and Seiichiro Naito, "Principles and Applications of Pencil Tracing", *Computer Graphics*, Vol.21, No.4, (Proc. *ACM SIGGRAPH '87*) pp.45-54 (1987).

(2) Mikio Shinya, and Tokiichiro Takahashi, "Theories and Error Analysis of Pencil Tracing", The IEICE Transactions on Information and Systems, PT2 (Japanese Edition), Vol.J72-D-Ⅱ, No.12, pp.2000-2011 (1989) (in Japanese).

(3) Mikio Shinya, Takafumi Saito and Tokiichiro Takahashi, "Rendering Techniques for Transparent Objects", *Graphics Interface'89*, pp.173-182 (1989).

(4) Mikio Shinya, Hitoshi Takakuwa, Takafumi Saito, and Tokiichiro Takahashi, "Cross Filtering for Rendering Bright Objects", Journal of the Institute of Television Engineers of Japan, Vol.42, No.8, pp.860-861 (1988) (in Japanese).

(5) Toshimitsu Tanaka and Tokiichiro Takahashi, "Cross Scanline Algorithm", *Eurographics'90*, pp.63-74 (1990).

(6) Toshimitsu Tanaka and Tokiichiro Takahashi, "Cross Scanline Algorithm for Anti-aliasing", Transactions of Information Processing Society of Japan, Vol.32, No.2, pp.197-205 (1991) (in Japanese).

(7) Toshimitsu Tanaka and Tokiichiro Takahashi, "Shading for Area Light Sources", *Eurographics'91*, pp.235-246 (1991).

(8) Toshimitsu Tanaka and Tokiichiro Takahashi, "Shading Method for Objects Illuminated by Area Light Sources", Transactions of Information Processing Society of Japan, Vol.32, No.11, pp.1383-1391 (1991) (in Japanese).

(9) Toshimitsu Tanaka and Tokiichiro Takahashi, "Precise Rendering Method for Edge Highlighting", *Computer Graphics International '91*, pp.235-298 (1991).

(10) Toshimitsu Tanaka, Tokiichiro Takahashi, "Precise Rendering Method and its Application for Highlight Generation", Transactions of Information Processing Society of Japan, Vol.33, No.4, pp.471-480 (1992) (in Japanese).

(11) Toshimitsu Tanaka and Tokiichiro Takahashi, "Fast Analytic Rendering with Accurate Table Integration", *Computer Graphics International '93*, pp.127-138 (1993).

(12) Toshimitsu Tanaka and Tokiichiro Takahashi, "Fast and Accurate Image Generation with Adaptive Tables", The IEICE Transactions on Information and Systems, PT2 (Japanese Edition), Vol.J78-D-II, No.1, pp.19-28 (1995) (in Japanese).

(13) Toshimitsu Tanaka and Tokiichiro Takahashi, "Precise Rendering Method for Exact Anti-Aliasing and High-Lighting", *The Visual Computer*, Vol.8, No.4, pp.315-326 (1994).

(14) Toshimitsu Tanaka and Tokiichiro Takahashi, "Cross scan buffer and its applications", *Computer Graphics Forum*, Vol.13, No.3, (Proc. *Eurographics '94*) pp.467-476 (1994).

(15) Toshimitsu Tanaka and Tokiichiro Takahashi, "Fast Shadowing Algorithm for Linear Light Sources", *Computer Graphics Forum*, Vol.14, No.3, (Proc. *Eurographics '96*) pp.205-216 (1995).

(16) Toshimitsu Tanaka and Tokiichiro Takahashi, "Fast Analytic Shading and Shadowing for Area Light Sources", *Computer Graphics Forum*, Vol.16, No.3, (Proc. *Eurographics '97*) pp.231-240 (1997).

(17) Takafumi Saito, Mikio Shinya, and Tokiichiro Takahashi, "Highlighting Rounded Edges", *Computer Graphics International'89*, pp.613-629 (1989).

(18) Takafumi Saito, and Tokiichiro Takahashi, "Comprehensible Rendering of 3-D Shapes", *Computer Graphics*, Vol.24, No. 4, (Proc. *ACM SIGGRAPHG '90*) pp. 197-206 (1990).

(19) Takafumi Saito, and Tokiichiro Takahashi, "Edge Enhanced Drawing in Computer Graphics", Transactions of Information Processing Society of Japan, Vol.31, No.8, pp.1205-1212 (1990) (in Japanese).

(20) Saito, T., and Takahashi, T., "Edge-Enhanced Drawing in Computer Graphics", *Journal of Information Processing*, Vol.15, No.2 pp.234-242 (1992).

(21) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, Seiichiro Naito, "SIGHT --- A Dedicated Computer Graphics Machine", *Computer Graphics Forum*, Vol.6, No.4, p327-334 (1987).

(22) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, and Hiroshi Kaneko, "An Architecture of a Graphics Computer---SIGHT", The IEICE Transactions on Information and Systems, (Japanese Edition), Vol.J69-D, No.3, pp.474-476 (1986) (in Japanese).

(23) Tokiichiro Takahashi, Masaharu Yoshida, N. Naruse, Seiichiro Naito, "Architecture and Performance Evaluation of the Dedicated Graphics Computer SIGHT", *IEEE MONTECH'87*, pp.153-160 (1987).

(24) Masaharu Yoshida, Tadashi Naruse, and Tokiichiro Takahashi, "A Dedicated Graphics Processor SIGHT-2", 4^{th} *Eurographics Workshop on Hardware*, (R. L. Grimsdale and W. Strasser (eds.), "*Advances in Computer Graphics Hardware IV*"), pp.151-169 Springer-Verlag (1991).

(25) Tadashi Naruse, Masaharu Yoshida, and Tokiichiro Takahashi, "Performance Analysis of the CG Computer SIGHT", The IEICE Transactions on Information and Systems, PT1 (Japanese Edition), Vol.J74-D-I, No.2, pp.166-174 (1991) (in Japanese).

(26) Tokiichiro Takahashi, Masaharu Yoshida, N. Naruse, K. Takikawa, "An HDTV Animation System using a Dedicated Graphics Computer SGHT-2", *Computer Animation'92*, (N. M. Thalmann and D. Thalmann (eds.) "*Models and Techniques in Computer Animation*"), pp.162-176 (1993).

(27) Toshimitsu Tanaka, Tokiichiro Takahashi, and Seiichiro Naito, "Three-Dimensional Shape Generation based on Generalized Symmetry", *Computer Graphics International '88*, pp.364-378 (1988).

(28) Toshimitsu Tanaka, Tokiichiro Takahashi, and Seiichiro Naito, "Recovery of 3D Shape Based on Generalized Symmetry", The IEICE Transactions on Information and Systems (Japanese Edition), Vol.J71-D, No.1, pp.84-91 (1988) (in Japanese).

(29) Takafumi Saito, and Tokiichiro Takahashi, "NC Machining with G-buffer Method", *Computer Graphics*, Vol.25, No.4, (Proc. *ACM SIGGRAPH '91*) pp.207-216 (1991).

Pattern Recognition

(30) Tokiichiro Takahashi, and Isao Masuda, "A Basic Study on Handprinted Chinese Character Recognition by the Stroke Extraction Methods Using Pen Point Sequence", The IECE Transactions on Information and Systems (Japanese Edition), Vol.J65-D, No.10, pp.1294-1301 (1982) (in Japanese).

(31) Tokiichiro Takahashi, Seiichiro Naito, and Isao Masuda, "Handprinted Chinese Character Recognition Using Stroke Extraction Method with Referral to Peripheral Structural Information", The IECE Transactions on Information and Systems (Japanese Edition), Vol.J67-D, No.9, pp.1052-1059 (1984) (in Japanese).

(32) Tokiichiro Takahashi, Seiichiro Naito and Isao Masuda, "Hand-printed Chinese Character Recognition Using Stroke Extraction Method with Referral to Peripheral Structural Information", *ICTP'84* (*International Conference on Text Processing with a Large Character Set*), pp.163-168 (1984).

(33) Isao Masuda, Norihiro Hagita, Teruo Akiyama, Tokiichiro Takahashi and Seiichiro Naito, "Approach to Smart Document Reader System", *IEEE Computer Vision and Pattern Recognition* (Proc. *IEEE CVPR'85*), pp.550-557 (1985).

Learning Science and e-Learning Systems

(34) Tomoyuki Nishita, Kunio Kondo, Yoshio Ohno, Kenjiro L. Miura, Yoshiaki Takai, Yoshinori Dobashi, Satomi Ishiuchi, Tokiichiro Takahashi, Ayumi Kimura, and Ayumi Miyai, "Development of Web Based Training System and Courseware for Advanced Computer Graphics Courses Enhanced by Interactive Java Applets", *ICGG 2002*, (*International Conference on Geometry and Graphics*), pp.123-128 (2002).

(35) Tsuneko Kura, Tokiichiro Takahashi, Tsuyoshi Fujimoto, Naoyuki Kakehi, and Hana Tsuchiya, "CyberPedia: an Internet-based Content-growing Encyclopedia and Evaluation of a Pilot Project", *WebNet 2001*, pp.730-731 (2001).

(36) Naoyuki Kakehi, Tsuneko Kura, R. Kusaka, Tsuyoshi Fujimoto, and Tokiichiro Takahashi, "Development of eTrip System: Collaborative Learning Platform for a Field Trip", *WebNet* 2001, pp.643-648 (2001).

(37) Takeshi Sugiyama, and Tokiichiro Takahashi, "CollaBoard: Web-based Collaborative Learning System Using SVG", *WebNet* 2001, pp.1190-1191 (2001).

(38) Naoyuki Kakehi, Tsuneko Kura, Tokiichiro Takahashi, Tsuyoshi Fujimoto, Hana Tsuchiya, and Sadayuki Hongo, "Concept and Applications of CyberPedia: a New Platform for Internet-based Collaborative Encyclopedias for K-12 Education", *ED-MEDIA 2002*, (*World Conference on Educational Multimedia, Hypermedia & Telecommunications*), pp.900-901 (2002).

(39) Takeshi Sugiyama, Naoyuki Kakehi, Tsuneko Kura, and Tokiichiro Takahashi, "CoCoFolio: A Web-based Electronic Portfolio for Enriching Students Learning by Collaboration", *e-Learn 2002*, (*World Conference on e-Learning in Corporate, Government, Healthcare, and Higher Education*), pp.916-921 (2002).

(40) Takeshi Sugiyama, Tsuneko Kura, Naoyuki Kakehi, Tokiichiro Takahashi, Keiko Takemiya, "Manga University: Web-based Correction System for Artistic Design Education", *ICCE 2002*, (*International Conference on Computers in Education*), pp.1404-1405 (2002).

(41) Takahiro Noda, H. Seshimo, Tetsuya Ura, Mina Maruyama, Akimichi Tanaka, and Tokiichiro Takahashi, "Development of WebAngel: Proxy-based Navigation-Support System for Learning from the Web", *ED-MEDIA 2003, (World Conference on Educational Multimedia, Hypermedia & Telecommunications)*, pp409-414 (2003).

(42) Takahiro Noda, H. Seshimo, Mina Maruyama, and Tokiichiro Takahashi, "Guiding Students around Personalized Information Sources: An Intermediary Approach to Utilizing Existing Web Resources", *3rd IEEE International Conference on Advanced Learning Technologies (ICALT2003)*, p.483 (2003).

(43) M. Katsura, Masaki Hisada, Akimichi Tanaka, and Tokiichiro Takahashi, "The development of the learning history visualization system WebFolio in Web-Based Learning", *Proc. of FIT 2003 (Forum on Information Technology 2003)*, LN-005 (2003) (in Japanese).

Books

(1) Tomoyuki Nishita, Yoshio Ohno, Kunio Kondo, Kenjiro L. Miura, Yoshiaki Takai, Tokiichiro Takahashi, Yoshinori Dobashi, "e-Learning 'Master Course on CG Techniques'", The Computer Graphic Arts Society (2001) (in Japanese).

(2) Tomoyuki Nishita, Yoshio Ohno, and Tokiichiro Takahashi, "Master Course on CG Techniques Part I Sub-Textbook", The Computer Graphic Arts Society (2001) (in Japanese).

(3) Tomoyuki Nishita, Yoshio Ohno, and Tokiichiro Takahashi, "Master Course on CG Techniques Part II Sub-Textbook", The Computer Graphic Arts Society (2001) (in Japanese).

(4) Tomoyuki Nishita, Yoshio Ohno, and Tokiichiro Takahashi, "CG Educational CD-ROM 'Comprehensible CG Techniques' ", The Computer Graphic Arts Society (2002) (in Japanese).

(5) Y. Kawaguchi (ed.) (Y. Kawaguchi, M. Kusahara, S. Sasada, T. Takada, Tokiichiro Takahashi, K. Ishigaki, N. Shichijo, S. Moroi, T. Kimura, T. Kawano, K. Kushiyama, Y. Wada, H. Yoshida, C. Honda, H. Tamegaya, R. Kijima, M. Koide, E. Terashima, M. Yokota, H. Ikeda, N. Yoshikawa, K. Uehara, and H. Yoshimura), "Introduction to CG", Maruzen (2003) (in Japanese).

Art Work

(1) Takafumi Saito, Tokiichiro Takahashi, and Jin Tamai, “Rainbow Skull”, Senior High School “Art” Textbook I, (Approved by the Ministry of Education), Nihon Bunkyo Shuppan (1994) (in Japanese).

Award

(1) **Best Papers Award**, Information Processing Society of Japan 1992, “Cross Scanline Algorithm for Anti-aliasing”, (May, 1992).

(2) **Best Paper Award**, Eurographics’94/Computer Graphics Forum 1994, “Cross Scan Buffer and its Applications”, (September, 1994).

(3) **Best International Papers Award**, Japan Information-Culturology Society 1996, “Research on Photorealistic and Non-photorealistic Rendering Technologies”, (November, 1996).

(4) **First Prize**, Eurographics’91 Proceedings Cover Image Contest, (September, 1991).

(5) **Selected Paper Award**, Computer Graphics Internaitonal’88, “Three-Dimensional Shape Generation based on Generalized Symmetry”, (June, 1988).

(6) **Selected Paper Award**, Computer Graphics Internaitonal’93, “Fast Analytic Rendering with Accurate Table Integration”, (June, 1993).

Technical Reports and National Congresses

Computer Graphics

(1) Mikio Shinya, Tokiichiro Takahashi, Seiichiro Naito, "Pencil Tracing Algorithm for Computer Graphics", Proc. of the 1986 IEICE Society Conference on Information and Systems, Vol.1, #1-90, p.1-90 (1986), (in Japanese).

(2) Tokiichiro Takahashi, Mikio Shinya, Seiichiro Naito, "An Analytical Illumination Calculation using Pencil Tracing Algorithm", Proc. of the 1987 IEICE General Conference, Vol.6, #1641, p.6-283 (1987), (in Japanese).

(3) Mikio Shinya, Tokiichiro Takahashi, Seiichiro Naito, "A Theoretical Study of Pencil Tracing for Image Synthesis", Proc. of The 33rd National Convention of IPSJ, #4Q-4, pp.2111-2112 (1987), (in Japanese).

(4) Mikio Shinya, Tokiichiro Takahashi, "An Image Synthesis Method using Pencil Tracing: Ray Interpolation", Proc. of the 1987 IEICE Society Conference on Information and Systems, Vol.1, #98, p.1-98 (1987), (in Japanese).

(5) Mikio Shinya, Tokiichiro Takahashi, "A Shadowing Technique for Transparent Objects", Proc. of the 1987 IEICE General Conference, Vol.D-1, #D-521, p.1-264 (1988), (in Japanese).

(6) Mikio Shinya, Takafumi Saito, Tokiichiro Takahashi, "Pencil Tracing Dispersive Objects", Proc. of The 37th National Convention of IPSJ, #3T-6, pp.1691-1692 (1988), (in Japanese).

(7) Mikio Shinya, Tokiichiro Takahashi, Takafumi Saito, Hitoshi Takakuwa, "Cross Filtering for Rendering Shining Objects", Proc. of 1988 General Conference of the Institute of Television Engineers of Japan, #20-1, p.425 (1988), (in Japanese).

1 Takafumi Saito, Mikio Shinya, Tokiichiro Takahashi, "Displaying Highlights on Polyhedral Edges", Proc. of the 37th National Convention of IPSJ, #3T-1, pp.1681-1682 (1988), (in Japanese).

(8) Takafumi Saito, Mikio Shinya, Tokiichiro Takahashi, "An Edge Highlighting Method and its Extension", IPSJ Symposium on Graphics and CAD'88, pp.9-16 (1988), (in Japanese).

(9) Toshimitsu Tanaka, Tokiichiro Takahashi, "A Fast Illumination Method by Referring Non-Linear Tables", Proc. of the 43rd National Convention of IPSJ, #5U-06 (1988), (in Japanese).

(10) Toshimitsu Tanaka, Tokiichiro Takahashi, “Cross Scan Buffer: its Concept and Applications”, Proc. of the 1993 IEICE General Conference, Vol.7, #D-615, p.7-376 (1993), (in Japanese).

(11) Toshimitsu Tanaka, Tokiichiro Takahashi, “Cross Scan Buffer Shadowing”, Proc. of the 1993 IEICE General Conference, Vol.7, #D-614, p.7-375 (1993), (in Japanese).

(12) Toshimitsu Tanaka, Tokiichiro Takahashi, “Cross Scan Buffer for Multi-size Images”, Proc. of the 47th National Convention of IPSJ, #2V-9, pp.2-329−2-330 (1993), (in Japanese).

(13) Toshimitsu Tanaka, Tokiichiro Takahashi, “An Analytical Shadowing Algorithm for Linear Light Sources”, Proc. of IIEEJ Symposium on Visual Computing ’95, #V-9, pp.104-105 (1995), (in Japanese).

(14) Toshimitsu Tanaka, Tokiichiro Takahashi, “A Shadowing Algorithm Using Ray-Oriented Buffer for Area Light Sources”, Proc. of Joint Symposium ’96 on IIEEJ Visual Computing and IPSJ Graphics and CAD, #V-8, pp.108-109 (1996), (in Japanese).

(15) Toshimitsu Tanaka, Seiichiro Naito, Tokiichiro Takahashi, Isao Masuda, “Recovery of 3D curved surface based on Generalized Symmetry”, IEICE Technical Report, Vol.PRU86-5, pp.35-44 (1986), (in Japanese).

(16) Toshimitsu Tanaka, Tokiichiro Takahashi, Isao Masuda, “Recovery of Three-Dimensional Shape based on Generalized Symmetry”, Proc. of The 33rd National Convention of IPSJ, #4P-9, pp.1621-1622 (1987), (in Japanese).

(17) Toshimitsu Tanaka, Tokiichiro Takahashi, Isao Masuda, “Recovery of 3D Generalized Symmetrical Object Shape”, Proc. of the 198 IEICE General Conference, Vol.7, #S17-4, pp.287-288 (1987), (in Japanese).

(18) Takafumi Saito, Tokiichiro Takahashi, “Extended Quadric Patch --- A Geometric Curved-Surface Representation without Undesirable Effects of the Parameters”, Proc. of IPSJ Symposium on Graphics and CAD, pp.67-73 (1987), (in Japanese).

(19) Hiroaki Yashiro, Tokiichiro Takahashi, Kei Takikawa, “An O(n(1+to) Algorithm for A Simple Polygon Triangulation and its Evaluation”, IEICE Technical Report, Vol.PRU89-41, pp.9-16 (1989), (in Japanese).

(20) Masaharu Yoshida, Tadashi Naruse, Tokiichiro Takahashi, Seiichiro Naito, "Design of a Graphics Computer – SIGHT –", IPSJ Technical Report, Computer Architecture, Vol.60-4, pp.1-8 (1985), (in Japanese).
(21) Tokiichiro Takahashi, Tadashi Naruse, Masaharu Yoshida, Seiichiro Naito, "An Architecture of a Graphics Processor: SIGHT -A Firmware Configuration-", IEICE Technical Report PRL, Vol.PRL85-53, pp.79-88 (1985), (in Japanese).
(22) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, Hiroshi Kaneko, "An Architecture of a Graphics Computer SIGHT", Proc. of the 1985 IEICE Society Conference on Information and Systems, Vol.1, #93, p.1-93 (1985), (in Japanese).
(23) Masaharu Yoshida, Tadashi Naruse, Tokiichiro Takahashi, "Design of a Graphics Computer---SIGHT---", IPSJ Technical Report, Computer Architecture, Vol.60-4, pp.1-8 (1985), (in Japanese).
(24) Tokiichiro Takahashi, Tadashi Naruse, Masaharu Yoshida, "An Architecture of a Graphics Processor: SIGHT ---A Firmware Configuration---", IECIE Technical Report, Vol.AL85-54/Vol.PRU85-53, pp.79-88 (1985), (in Japanese).
(25) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "A Dedicated Graphics Computer SIGHT", Proc. of The 17th Image Engineering Joint Conference, #12-4, pp.217-218 (1986), (in Japanese).
(26) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "A Performance Evaluation of the CG Computer SIGHT", Proc. of The 33rd National Convention of IPSJ, #1P-1, pp.1551-1552 (1987), (in Japanese).
(27) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "SIGHT/C – A High-level Language for CG Computer SIGHT", Proc. of the 1987 IEICE Society Conference on Information and Systems, Vol.1, #95, p.1-95 (1987), (in Japanese).
(28) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "SIGHT---A Dedicated Computer Graphics Machine", Proc. of 1987 Sapporo International Computer Graphics Symposium, CG-1, pp.25-30 (1987).
(29) Masaharu Yoshida, Tokiichiro Takahashi, Tadashi Naruse, "Performance Evaluation of the SIGHT Prototype System", Proc. of the 1988 IEICE General Conference, Vol.5-6, #1638, p.6-316 (1988), (in Japanese).
(30) Tokiichiro Takahashi, Tadashi Naruse, Masaharu Yoshida, "Performance Evaluation of the SIGHT Prototype System (II)", Proc. of The 35th National Convention of IPSJ, #6G-9, pp.2383-2384 (1988), (in Japanese).
(31) Masaharu Yoshida, Tadashi Naruse, Tokiichiro Takahashi, "A Dedicated CG Computer SIGHT", Proc. of CG Osaka '88, pp.C3-1-C3-9 (1988), (in Japanese).

(32) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "A Dedicated Graphics Computer SIGHT---Parallel Operations of Ray Tracing---", IPSJ Symposium on Computer Architecture, pp.217-225 (1988), (in Japanese).
(33) Tokiichiro Takahashi, Tadashi Naruse, Masaharu Yoshida, Reiko Minamikawa, Tsuguchika Kaminuma, "Cell Division Animation of C.elegans by Dedicated Graphics Computer SIGHT", IEICE Technical Report, Vol.IE88-14, pp.23-30 (1988), (in Japanese).
(34) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "A Ray Tracing Processor SIGHT", Technical Report of the Institute of Image Electronics Engineers of Japan, Vol.88-01-3, pp.13-18 (1988), (in Japanese).
(35) Masaharu Yoshida, Tadashi Naruse, Tokiichiro Takahashi, "A Dedicated Graphics Processor SIGHT-2", IPSJ Technical Report, Computer Architecture, Vol.77-6, pp.43-50 (1989), (in Japanese).
(36) Tadashi Naruse, Masaharu Yoshida, Tokiichiro Takahashi, "On a 3 Dimensional Vector Arithmetic on the Graphics Computer SIGHT", IPSJ Technical Report, Computer Architecture, Vol.77-7, pp. 51-58 (1989), (in Japanese).
(37) Tokiichiro Takahashi, Masaharu Yoshida, Tadashi Naruse, "Static Load Balancing for Multiprocessor Ray Tracing on the Dedicated Graphics Computer SIGHT2", Proc. of IIEEJ Symposium on Visual Computing '94, #V-1, pp.91-92 (1994), (in Japanese).

(38) Tokiichiro Takahashi, "A Study on 'Visual Computing'", Proc. of IIEEJ Symposium on Visual Computing '93, #V-1, pp.129-132 (1993), (in Japanese).
(39) Yuichi Koiso, Tomoyuki Nishita, Naoyuki Kakehi, Tsuneko Kura, Tokiichiro Takahashi, "Development of Matching System between Standard Map and Deformed Map using FFD", Proc. of the 62nd National Convention of IPSJ, #3V-7, pp.4-135－4-136 (2001), (in Japanese).

Pattern Recognition (all in Japanese)

(40) Tokiichiro Takahashi, Michio Umeda, "Sub-pattern extraction of Handprinted Chinese Characters based on List Processing", Proc. of the 1980 IEICE General Conference, #1325, p.5-314 (1980).

(41) Tokiichiro Takahashi, Michio Umeda, Isao Masuda, "Sub-pattern extraction of Handprinted Chinese Characters based on Stroke Structure Matching", Proc. of the 1981 IEICE General Conference, #1323, p.5-314 (1981).

(42) Tokiichiro Takahashi, Kazumi Odaka, Isao Masuda, "Classification of Handprinted Chinese Characters by Stroke-Order Information in Reference Pattern", IEICE Technical Report, Vol.PRL81-12, pp.87-94 (1981).

(43) Tokiichiro Takahashi, Isao Masuda, "Classification of Handprinted Chinese Characters by Stroke Shape Extraction Method using Stroke-Order Information in Reference Pattern", Proc. of the 1982 IEICE General Conference, #1395, p.5-373 (1982).

(44) Tokiichiro Takahashi, Seiichiro Naito, Isao Masuda, "Handprinted Chinese Character Recognition by the Stroke Extraction Method using Peripheral Structural Information", IEICE Technical Report PRL, Vol.PRL82-44, pp.47-54 (1982).

(45) Tokiichiro Takahashi, Seiichiro Naito, Isao Masuda, "Comparative studies on characteristics of discriminant function between the stroke extraction method and the pattern matching method based on peripheral contributivity function features", Proc. of the 1983 IEICE General Conference, p.5-341 (1983).

(46) Tokiichiro Takahashi, Seiichiro Naito, Isao Masuda, "Software Design for Handprinted Chinese Character Recognition based on Structural Analysis", Proc. of the 1983 IEICE Society Conference on Information and Systems, Vol.1, #102, p.1-102 (1983).

(47) Tokiichiro Takahashi, Seiichiro Naito, Isao Masuda, "Evaluation of Stroke Extraction Method by Stroke-Order Information in Reference Pattern", Proc. of the 1984 IEICE General Conference, Vol.5, #S10-6, pp.5-509-5-410 (1984).

(48) Tokiichiro Takahashi, Seiichiro Naito, Isao Masuda, "", Technical Journal of NTT Electrical Communications Laboratories, Vol.34, No.1, pp.13-24 (1985).

(49) Tokiichiro Takahashi, Masaharu Yoshida, Seiichiro Naito, "Systolic Binary Processor-Tree Architecture for Pattern Recognition", IEICE Technical Report, Vol.PRU86-13, pp.21-30 (1986).

Learning Science and e-Learning Systems (all in Japanese)

(50) Tomoyuki Nishita, Kunio Kondo, Yoshio Ohno, Kenjiro L. Miura, Yoshiaki Takai, Yoshinori Dobashi, Satomi Ishiuchi, Tokiichiro Takahashi, Ayumi Kimura, and Ayumi Miyai, "Development of Web Based Training System and Courseware for Advanced Computer Graphics Courses Enhanced by Interactive Java Applets", Proc. of the 62^{nd} National Convention of IPSJ, #5D-4, pp.SP2-97-100 (2001).

(51) Tokiichiro Takahashi, "CyberPedia, an Internet-based Contents-Growing Encyclopedia", Proc. of the 1998 IEICE General Conference, Vol.D-1 (Information and Systems), #SD-2-4-4, pp.391-392/Vol.D-2 (Information and Systems), #SD-2-4-4, pp.515-516 (1998).

(52) Tokiichiro Takahashi, Junji Horiuchi, Tsuyoshi Fujimoto, Sadayuki Hongo, Hiroaki Yashiro, Gunei Sato, Shigeki Iwabuchi, Kazuyuki Mikami, "Development of a CyberPedia, an Internet-based Contents-Growing Encyclopedia", Proc. of the 1998 IEICE Society Conference on Information and Systems, #D-15-9, p.348 (1998).

(53) Kunihiro Takiuchi, Takuya Tarama, Ritsuko Maezono, Tsutomu Shimomura, Shinichi Hirokawa, Junichi Matsui, Naozumi Tsuchida, Tsuyoshi Fujimoto, and Tokiichiro Takahashi, "A trial class making use of "The Internet Cyber Zukan" authoring system", Proc. of The Internet and Education Symposium 2000 in Okinawa, pp.112-116 (2000).

(54) Makiko Katsura, Tokiichiro Takahashi, Gunei Sato, "Reconstruction of Teaching Materials of English Employing WebAngel", Proc. of The 63^{rd} National Convention of IPSJ, Vol.4, #3S-06, pp171-172 (2001).

(55) Toshifumi Sano, Takahiro Noda, Tetsuya Ura, Hitoshi Seshimo, Akimichi Tanaka, Mina Maruyama, Tokiichiro Takahashi, "Development and Evaluation of Proxy-based WebAngel", Proc. of the 1993 IEICE General Conference, Vol.2002, Information and Systems Part 1, #D-15-27, p.216, (2002)

(56) Tsuneko Kura, Takeshi Sugiyama, Tokiichiro Takahashi, Masahiko Tsujimoto, Keiko Takemiya, "Development of Web-based Correction System for Design", Proc. of the 1993 IEICE General Conference, Vol.2002, Information and Systems Part 1, #D-15-8, p.197, (2002)

(57) Hitoshi Seshimo, Takahiro Noda, Mina Maruyama, Akimichi Tanaka, Tokiichiro Takahashi, "Inquiry-based Learning Support System on the Web: 'WebAngel'", IEICE Technical Report, Vol.102, No.388 (ET2002 44 - 58), pp.7 - 12, (2002).

(58) Hitoshi Seshimo, Takahiro Noda, Mina Maruyama, Akimichi Tanaka, Tokiichiro Takahashi, “WWW Navigation Support System for Inquiry-based Learning: ‘WebAngel’---Its Concept and System Development”, Preprint of the 39th Annual Meeting on Information Science and Technology, Vol.39, pp.41-47 (2002).
(59) Takahiro Noda, Hitoshi Seshimo, Tetsuya Ura, Toshifumi Sano, Mina Maruyama, Akimichi Tanaka, Tokiichiro Takahashi, “Development of Proxy-based WebAngel and its Evaluation”, Preprint of the 39th Annual Meeting on Information Science and Technology, Vol.39, pp.49-54 (2002).
(60) Naoyuki Kakehi, Takeshi Sugiyama, Tsuneko Kura, Tokiichiro Takahashi, CoCoFolio: An Electronic Portfolio for Enriching Students Learning through Contents Collaboration”, Preprint of the 39th Annual Meeting on Information Science and Technology, Vol.39, pp.55-59, (2002).
(61) Masaki Hisada, Makiko Katsura, Hitoshi Seshimo, Akimichi Tanaka, Tokiichiro Takahashi, “A Reflection Support System for Internet-based Learning---Concept and Development of Prototype”, IEICE Technical Report, Vol.102, No.697 (ET2002 95 - 119), pp.133 - 138 (2003).
(62) Takahiro Noda, Hitoshi Seshimo, Mina Maruyama, Tokiichiro Takahashi, “Development of WebAngel Proxy Server to Utilize Existing Web Resources as Educational Material”, IEICE Technical Report, Vol.103, No.60 (ET2003 1 - 9), pp.19 - 24 (2003).
(63) Tetsuya Ura, Satoru Matsuo, Tsuyoshi Fujimoto, Tokiichiro Takahashi, Hideo Suzuki, “An Enhanced Distance Education System using White Board Sharing System”, *Proc. of FIT 2003* (*Forum on Information Technology 2003*), Vol.FIT 2003, Part 3, #K-089, pp.617 - 618 (2003).
(64) Tsuneko Kura, Naoyuki Kakehi, Hideo Suzuki, Takeshi Sugiyama, Tokiichiro Takahashi, Keiko Takemiya, “Development of Web-based Correction System for Artistic Design”, The Second International Conference on Creating, Connecting and Collaborating through Computing, pp.40–47 (2003).
(65) Hitoshi Seshimo, Akimichi Tanaka, Mina Maruyama, Hideo Suzuki, Tokiichiro Takahashi, “Visualization and Analysis of Learning Styles in Inquiry-based Learning using Web”, IEICE Technical Report, Vol.103, No.697 (ET2003 98-136), pp.137 - 142 (2004).