

# 半空間集合のCSG表現形状データに対する 光線追跡アルゴリズムの検討

A Study of the Ray Tracing Algorithm for the Shape Description Data  
by Using CSG-Expression of Half Space Primitives

金 環 塚      岡 田 稔      横 井 茂 樹      鳥 脇 純 一 郎  
Kyung Eun KIM    Minoru OKADA    Shigeki YOKOI    Jun-ichiro TORIWAKI

名古屋大学 工学部 情報工学科  
Department of Information Eng., Faculty of Eng., Nagoya Univ.

**あらまし**      光線追跡法は物体の陰影付けの手段として非常に有用である。一方、無限平面によって定義される半空間をプリミティブとしたCSG表現により、凹、凸を含む一般の多面体を記述する方法は形状定義の方法として理解し易い。本文ではこのような形状表現データを光線追跡法によって表示するアルゴリズムについて検討した。本方法では、従来用いられていた2値による空間の集合演算に対してI（内部）、B（境界）、O（外部）を真理値とする3値演算、及び、従来の補集合の定義を修正した準補集合（Semi-Complement）を導入した。これにより、実数の数値計算を用いず Symbolic に交点判定を行うことが可能となり、交点の位置関係に関する誤差が生じにくいという特徴を持つ。さらにこの考え方に基づく形状リストに対する光線追跡システムを作成し、本方法が一般の多面体の表示に有用であることを確認した。

**Abstract**      The Ray-tracing algorithm is very useful as a tool for generating shaded pictures. In this paper, we studied the ray tracing algorithm for shapes defined by the Constructive Solid Geometry (CSG) using primitives of half spaces. In the algorithm, it is important to evaluate effectivity of ray-object intersection. We simplified the procedure to find effective intersection points by introducing a three-valued operation amended by the concept of semi-complement instead of the binary set operation. It is possible to discriminate intersections symbolically without numerical computation. In addition, we developed the system to display general objects using ray-tracing based on the above operation and confirmed the usefulness of the method proposed here.

## 1. ま え が き

光線追跡法はリアリティの高い映像を生成する手段として注目を浴びている。光線追跡法においては光線と対象形状との交点を求めることが基本となるため、形状記述の方法に応じて計算法が工夫されている<sup>3,4,5)</sup>。

本研究では多面体形状を記述する方法として考え方が

容易な半空間集合のCSG表現をとりあげ、その場合の光線追跡のアルゴリズムについて検討を加えた。これまでCSG表現データに対する光線追跡法の計算法については、いくつかの研究が報告されている<sup>1,5)</sup>。これらの研究において、光線と物体の交点の検出のための交点の有効性判定にはブール演算（2値演算）が用いられてい

たが、判定法に問題を含んでいた。本研究では、空間の論理演算を I (内部), B (境界), O (外部) を真理値とする 3 値演算に拡張するとともに、準補集合の考え方を導入することにより問題点を解消した。また、平面により定義される半空間をプリミティブとして 3 値演算に基づく CSG 表現の一方法を示し、この CSG リストで直接駆動される光線追跡システムを作成した。これにより、本方法が一般の凹、凸を含む多面体の表示に有用であることを実験的に示す。

## 2. 半空間をプリミティブとする CSG 表現による形状記述法

### 2-1 プリミティブの定義

3 次元の全空間を  $\Omega$  で表すと、 $\Omega$  はある無限平面  $S$  によって 2 つの領域で分けられる。この平面上と平面のどちらかの側の領域、つまり境界面 (平面  $S$ ) を含む半空間を形状記述のプリミティブと考える。

これは一般的に次式で表される。

$$OBJ_k = \{P \mid f_k(P) \geq 0\} \quad (1)$$

但し、 $f_k(P) = f_k(x, y, z) = a_k x + b_k y + c_k z + d_k$

( $a_k, b_k, c_k, d_k$  は平面を定めるパラメータ)

ここで、プリミティブ  $OBJ_k$  の内部、境界、外部を以下のように定義する。

内部:  $(OBJ_k)_I = \{X \mid f_k(X) > 0\}$

境界:  $(OBJ_k)_B = \{X \mid f_k(X) = 0\}$

外部:  $(OBJ_k)_O = \{X \mid f_k(X) < 0\}$

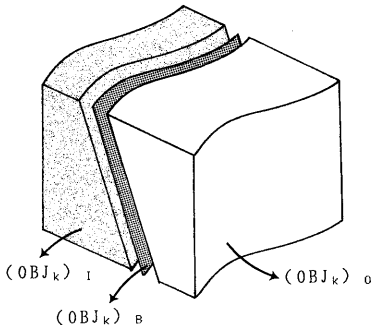


図1。無限平面で分けられる全空間  $\Omega$

### 2-2 集合演算の定義

上で定めたプリミティブの集合演算によって物体形状を記述する集合演算としては、和集合、積集合 (これは一般の集合演算と同じ) とともに補集合の定義を少し修正した準補集合を用いる。すなわち、

・和集合  $(OBJ_k \cup OBJ_l)$  :

(OR  $OBJ_k$   $OBJ_l$ ) で表す。

・積集合  $(OBJ_k \cap OBJ_l)$  :

(AND  $OBJ_k$   $OBJ_l$ ) で表す。

・準補集合  $(OBJ_k)^{SC} \equiv (OBJ_k)^C \cup (OBJ_k)_B$  :

(SC  $OBJ_k$ ) で表す。

ここで、準補集合の記述は物体の形状表現における境界上の点の所属を明確に表すという重要な意味を持つ。

### 2-3 形状表現の方法

物体の形状は、上記プリミティブとその間の演算の形で記述される。(この記述を形状リストと呼ぶ)

形状に対する集合演算の表現方式をそのままデータの情報として用いた形状リストは少量のデータで分かりやすいデータ構造を持っている。形状リストを構成する領域  $R_i$  を決定する各平面  $S_i$  は、平面のパラメータの値とともに不透明物体の面か透明物体の面かを示す情報を持っている。この集合演算を用いた形状リストのシンタックスを図2に示す。

```
<CSG expr.> ::= <left-p> <operator> [ <CSG expr.> |
                                     <region-var.> ] ... <right-p>
<operator> ::= AND | OR | SC
<region-var.> ::= <半空間プリミティブ名>
<left-p> ::= (
<right-p> ::= )
```

図2。集合演算による定義する形状リストのシンタックス

図3のような簡単な2次元形状の形状リストの一例を上で定義したシンタックスによって書くと、<形状リスト1>のようになる。

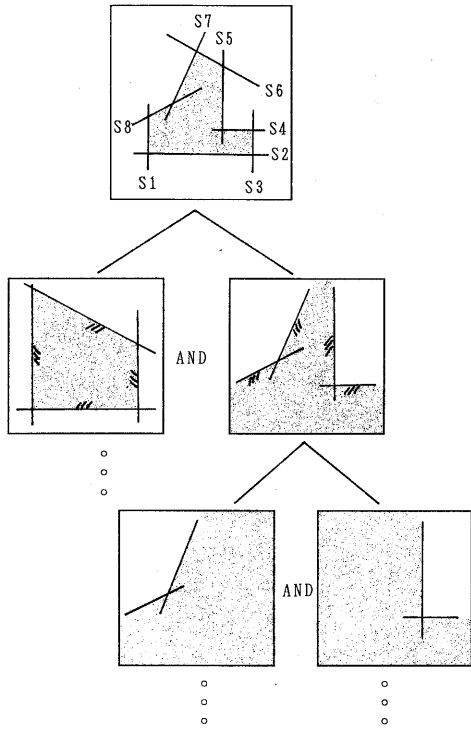


図3. 2次元形状の半空間プリミティブによる構成の例

<形状リスト1>

```
OBJECT=(AND (OR R1 R2 R3 R6 )
             (OR R4 R5 )
             (OR R7 R8 ))
        =(AND (OR R1 R2 R3 R6 )
             (SC (AND R4 R5 ))
             (SC (AND R7 R8 )))
```

### 3. 点の内外判定のための3値演算

#### 3-1 3値演算の定義と性質

半空間の集合  $OBJ_k$  に対して集合演算を施して構成されたある集合(物体)を光線追跡法で表示する場合、直線(光線)と物体との交点を求めることが重要である。これは全ての平面との交点を求めることにより行われ

るが、その際、各交点が物体の表面を構成する面の上にあるか否かを分類することが必要になる。これは以下のように簡単な演算によって求められることが知られる。まず、式(1)で決まる半空間Rに対する点Pの状態値を次の関数で定義する。

$$F(P;R) = \begin{cases} I; & \text{点Pが集合Rの内部にある} \\ B; & \text{点Pが集合Rの境界上にある} \\ O; & \text{点Pが集合Rの外部にある} \end{cases}$$

この時、 $F(P;R1 \cup R2)$ 、 $F(P;R1 \cap R2)$ 、 $F(P;R1^{SC})$ は、 $F(P;R1)$ と $F(P;R2)$ の値を用いて表1に基づいて求められる。但し、3-2で述べるように、BとBの演算では各々が同じ平面上にある場合は正しくない結果を得ることがあるから形状定義の時この条件を避けるようにする。

表1. 状態値に関する3値演算の真理値表

	$F(P;R2)$	$F(P;R2)$	$F(P;R1^{SC})$	
OR operation	AND operation	operation	Semi-Comp. operation	
	$F(P;R2)$	$F(P;R2)$	$F(P;R1)$	
	I B O	I B O	I O	
$F(P;R1)$	I	I I I	I	O
	B	I B B	B	B
	O	I B O	O	I
$F(P;R1)$	I	I B O	I	O
	B	B B O	B	B
	O	O O O	O	I

従って、複数個の半空間に対して上記の集合演算を任意回、任意順序で施して得られる集合Rに対する $F(P;R)$ は、表1を利用して順次計算していくことにより求められる。実際に形状が2-3で述べた形状リスト表現で与えられたとき、ある点の3値演算評価(内外判定)はインタプリタ方式を用いて次の手順で形状リスト列そのものを直接解釈、評価することにより得られる。(以下リスト評価という)

- i) <region-var.>に3値{I, B, O}の状態値を代入する。
- ii) <left-p>によってスタックに対応する左括弧の位置をPUSHする。

- iii) <right-p>によってスタックに対応する右括弧の位置を POP する。
- iv) <left-p>から<right-p>の<CSG expr.>を<operator>で演算する。
- v) ii)からiv)をスタックが空になるまで繰り返す。

### 3-2 CSGモデリングにおける相反問題

境界面を含む半空間で定めたプリミティブで形状を記述するとき、境界面が重複して存在する場合に問題が生じることがある。例えば、ガラスのテーブル(T)の上に置かれた取っ手(H)が付いているガラスのコップ(C)を表示する場合、和集合の形状表現 (T U H U C) をリスト評価すると次のような問題が生じる(図4)。

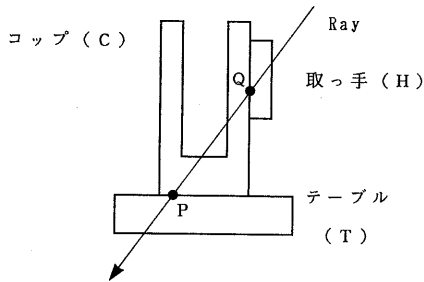
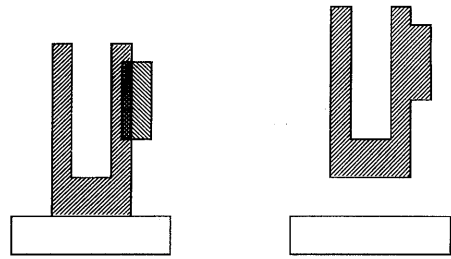


図4. 形状表現の例

テーブルとコップの接面上の点Pでは、その状態値はB(境界)として評価され、テーブルとコップに関しては別の物体と評価される。一方、取っ手とコップの接面上の点Qでも同様に状態値Bとなり、CとHを同一の連続した物体として扱いたい場合には問題が残る。一方、このような問題を避けるために、状態値Bも内部の状態値Iとして扱うと、点PとQでの状態値はIとして評価され(これはプリミティブの定義が $OBJ_k = \{P | f_k(P) > 0\}$  (2)の時考えられる。)、テーブルとコップと取っ手が全て連続物体と評価される。これらの相反問題を避けるには他の方法もあると思うが簡単な解決方法をそれぞれ図5の(a)、(b)に示し、実験結果の表示例を図9の(a)、(b)に示す。

尚、この問題は本方法だけでなく、他のCSGを用いた形状表現においても潜在的に存在する問題であると思われる。



- (a). (1)の定義で同一物体として扱いたい時: 重なりを持った定義をする
- (b). (2)の定義で別の物体として扱いたい時: 隙間を入れて定義する

図5. 形状表現における集合面の回避

## 4. 光線追跡法への適用

### 4-1 交点計算

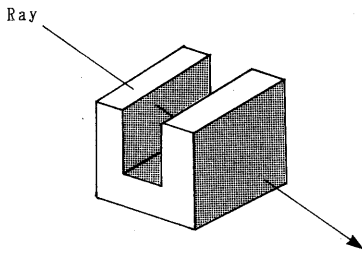
光線追跡法で物体を表示する場合、光線と物体との交点を求めることが基本となるが、本研究では以下のような交点判定、交点計算を行う。(図6参照)

- I) 物体を構成している全ての無限平面と光線との交点  $\{P_i | i=1, \dots, \text{面の数}\}$  を求める。
- II) I) で求めた全ての交点が記述された物体の境界面上にある点  $\{P_i | F(P_i; OBJ_k) = B\}$  か否かを判定する。この判定は3-1で述べた3値演算で容易に求められる。
- III) 抽出された交点の中で視点に最も近い点Pを探す。

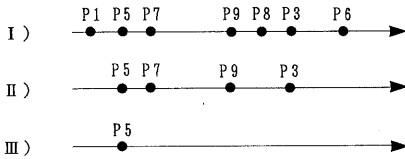
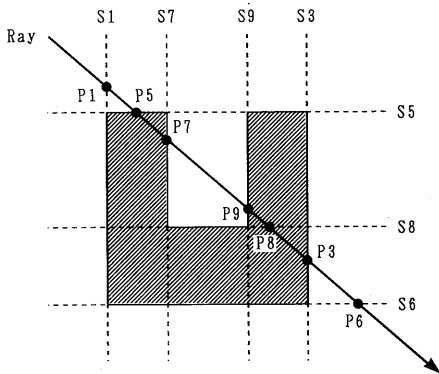
点 $P_i$ が乗っている面に対する論理評価には数値計算の必要が無く、交点相互の位置関係の誤差が生じにくい。これは、従来の2値による集合演算を拡張し、B(境界)というクラスを導入したことによる。ここで、点P5、P8、P6の3値演算を用いた結果は以下の交点判定のためのリスト評価の手順により、各々境界(B)、内部(I)、外部(O)になることが分かる。

<形状リスト2>

```
OBJECT=(AND (SC (OR R1 R2 R3 R4 R5 R6)
              (OR R7 (SC R8) R9))
```



a) <形状リスト 2> で定義された  
3次元の形状のスケッチと光線



b) <形状リスト 2> で定義された 2次元の境界面と光線

図 6. 交点計算の手順

(i) 点 P 5 に対して、

$$\begin{aligned} \text{OBJECT} &= (\text{AND} (\text{SC} (\text{OR} (\text{B} \text{ O} \text{ O} \text{ O} \text{ O} \text{ O}))) \\ &\quad (\text{OR} (\text{I} (\text{SC} (\text{I}) \text{ O}))) \\ &= (\text{AND} (\text{SC} (\text{B})) \\ &\quad (\text{OR} (\text{I} \text{ O} \text{ O}))) \\ &= (\text{AND} (\text{B} \text{ I})) \\ &= \text{B} \end{aligned}$$

(ii) 点 P 8 に対して、

$$\begin{aligned} \text{OBJECT} &= (\text{AND} (\text{SC} (\text{OR} (\text{O} \text{ O} \text{ O} \text{ O} \text{ O} \text{ O}))) \\ &\quad (\text{OR} (\text{O} (\text{SC} (\text{B}) \text{ I}))) \end{aligned}$$

$$\begin{aligned} &= (\text{AND} (\text{SC} \text{ O})) \\ &\quad (\text{OR} \text{ O} \text{ B} \text{ I})) \\ &= (\text{AND} \text{ I} \text{ I}) \\ &= \text{I} \end{aligned}$$

(iii) 点 P 6 に対して、

$$\begin{aligned} \text{OBJECT} &= (\text{AND} (\text{SC} (\text{OR} (\text{O} \text{ O} \text{ O} \text{ I} \text{ O} \text{ O} \text{ B}))) \\ &\quad (\text{OR} (\text{O} (\text{SC} (\text{O}) \text{ I}))) \\ &= (\text{AND} (\text{SC} (\text{I})) \\ &\quad (\text{OR} (\text{O} \text{ I} \text{ I}))) \\ &= (\text{AND} \text{ O} \text{ I}) \\ &= \text{O} \end{aligned}$$

#### 4-2 従来法との比較

ここで述べた方法は 3 値演算を用いて交点の有効性が簡単に調べられるのが特徴であるが比較のためこれまでに提案されている方法とその問題点を以下に示しておく。

##### ・ S. D. Roth の方法<sup>1)</sup>

光線と交わった物体の存在範囲をプリミティブ毎にセグメント化し、各セグメントの境界にはさまれた物体内部に対する毎に内部、外部の 2 値判定を行い、単純化する手順を考慮しているが、物体全てに対して処理を施さなければならないため処理時間がかかる。この手順を図 7 に示す。

##### ・ 村上らの方法<sup>5)</sup>

ある光線と交差したプリミティブの全ての交点を求め、それらの内部、外部判定を行う 2 値演算を行い、結果が真の場合、可視面として判定するという方法を用いている。しかし、この方法を用いると図 8 のような例において物体 A, B に対して  $A \cap B$  の演算の結果は外部として判定される。従って、この条件をチェックする別の判定が新たに必要になると思われる。

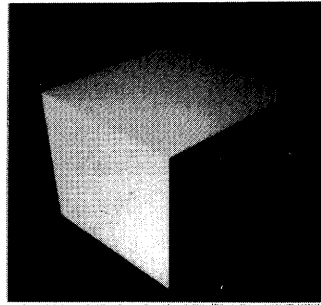
#### 4-3 光線追跡のアルゴリズム

光線追跡は光線が物体表面の交点で反射・屈折して伝搬していくのを視点から逆にたどって輝度計算を行うものである。上で述べた交点計算を用いて交点を計算しながら、反射光線、屈折光線の分岐をそれぞれ計算していくが、この過程は一般的なアルゴリズムと同じ<sup>6,7,8)</sup>となるので説明を省略する。

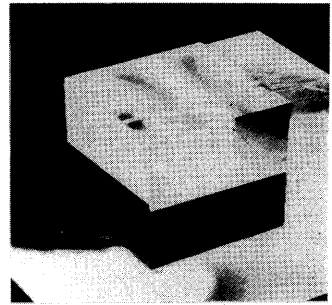


<形状リスト5>

```
OBJECT=(OR(SC(OR R1 R2
             R3 R4 R5 R6))
           R7)
```



(a)

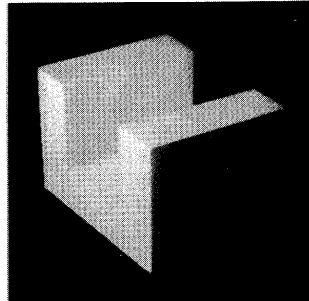


(b)

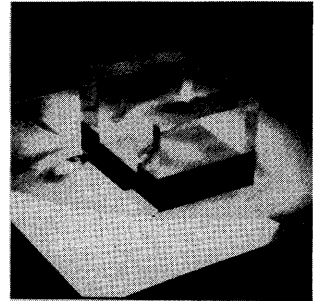
図10。<形状リスト5>で作成した画像

<形状リスト6>

```
OBJECT=(OR(AND(SC(OR R1 R3
                 R4 R5 R8 R9))
            (SC(AND R2 R6 (SC R7))))
          R10)
```



(a)

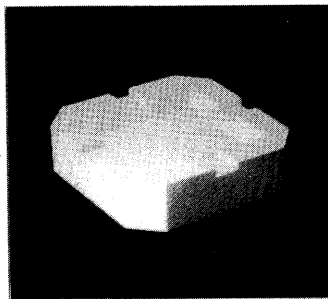


(b)

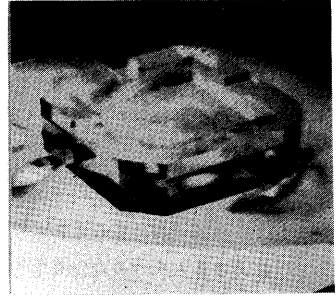
図11。<形状リスト6>で作成した画像

<形状リスト7>

```
OBJECT=(OR(AND(SC(OR R1 R2 R3 R4 R9
                 R10 R11 R12 R15))
            (OR R5 R6 R7 R8 (SC R14))
            (OR(SC R16) R17 R18)
            (OR(SC R16) R19 R20))
          R21)
```



(a)

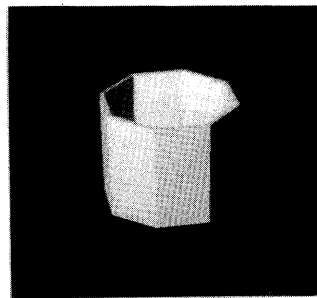


(b)

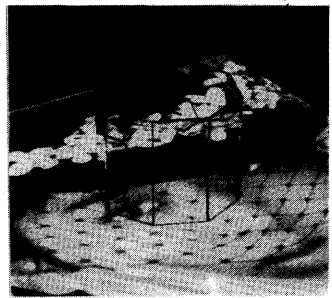
図12。<形状リスト7>で作成した画像

<形状リスト8>

```
OBJECT=(OR(AND(SC(OR R1 R2 R3 R4 R5
                 R6 R7 R8 R19 R17))
            (OR R9 R10 R11 R12 R13
                 R14 R15 R16 R18))
          R20)
```



(a)



(b)

図13。<形状リスト8>で作成した画像

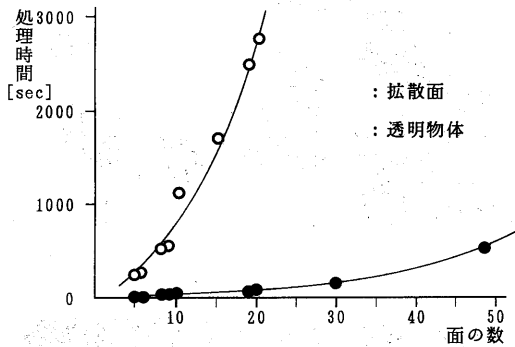


図14. 面の数と処理時間

## 6. まとめ

無限平面によって定義される半空間とその間の集合演算によって一般の多面体を定義する一方法を提案した。そして、この形状リストに基づいて実行される光線追跡システムを作成し、有用性を確認した。本方法の特徴は以下の通りである。

1. 多面体の定義が単純である。
2. 多面体の一つの面を記述する際、その面の辺縁の情報は半空間の3値演算で自動的に生成、評価されるため定義リスト内で陽に記述する必要が無い。
3. 3値演算を用いることにより、注目点が境界面上にあるかどうかの判定を論理演算で行うことができるので数値計算による誤差が生じにくい。

本研究では平面に対する半空間に限定したが、これを拡張すると一般の形状のCSG表現も可能と思われる。しかし、図5で示したように面の数によって処理時間が急増するため処理の高速化を考慮しなければならない。その他の今後の課題として以下の事項を検討する予定である。

1. プリミティブの定義に用いている平面を曲面に拡張する。
2. リスト評価の方法を工夫して高速に評価する方式を検討する。
3. プリミティブの定義による相反する問題点を

さらに検討する。

謝辞.

日頃大変熱心な御指導御討論を頂く鳥脇研究室の皆様  
に感謝する。なお、本研究の一部は文部省科研費(試験  
研究(1)63880008)による。

参考文献

- 1) S.D.Roth: Ray Casting For Modeling Solids, Computer Graphics and Image Processing Vol. 18, pp. 109-144 (1982)
- 2) J.R.Woodward: Elimination Redundant Primitives from Set-Theoretic Solid Models by a Consideration of Constituent, IEEE Computer Graphics & Applications pp. 38-47 (1988)
- 3) 若林, 鶴岡, 木村, 三宅: LISPによるソリッドモデラーの開発, 情処研資, CAD 25-4 (1987)
- 4) 川上, 嘉数, 仲野: CSGモデルにおける可視面分抽出に関する研究, 情処研資, CAD 29-7, pp. 47-54 (1987)
- 5) 村上, 松本: CSG表現からの画像生成方式, 情処全大, pp. 1533-1534 (1984)
- 6) Whitted, T.: An Improved Illumination Model for Shaded Display, Comm. ACM, Vol. 23, No. 6, pp. 343-349 (1980)
- 7) 安田, 横井, 鳥脇, 鶴岡, 三宅: 透明物体表示のための改良光線追跡法, 情処論, Vol. 25, No. 6 pp. 953-959 (1984)
- 8) 安田, 横井, 鶴岡, 三宅: 透明物体表示のための改良光線追跡法(2), 情処論, Vol. 26, No. 4 pp. 591-599 (1985)