

2 眼式立体画像のための ステレオクリッピング

安居院 猛 山崎 龍次 中嶋 正之

東京工業大学 像情報工学研究施設

2 眼式立体画像を CRT ディスプレイなどに表示する場合、画面の大きさが限られているため、右目からは画面内に見えても左目からは画面の外に出てしまっていて見えない、或いはその逆の現象が生じ、画面の両端で立体感を損ねていることがある。そこで、左右どちらの目から見ても画面内に見える領域に 3 次元クリッピングを行なう必要がある。この 3 次元クリッピングのことをステレオクリッピングと呼ぶことにする。

本報告では、ステレオクリッピングを行なう手法として Cohen-Sutherland アルゴリズムを拡張した手法、Sutherland-Hodgman アルゴリズムによる手法等のいくつかの手法について述べる。更に、ステレオクリッピングを行なった画像に対して視覚心理学的な実験を行ない、その有効性を確認した。

STEREO CLIPPING FOR BINOCULAR STEREO IMAGES

Takeshi AGUI Ryuji YAMAZAKI Masayuki NAKAJIMA

Imaging Science and Engineering Laboratory, Tokyo Institute of Technology.
4259, Nagatuta, Midori-ku, Yokohama-city, Kanagawa, Japan

When a man sees a stereo image in a small field of view, such as displayed on a CRT, stereoscopic effect is not made at the both sides of the screen, because at these places, a part of an object which can be seen by the right eye cannot be seen by the left one, or vice versa. We propose a method to confine the displayed object within the view range of both eyes. We called this three dimensional clipping STEREO-CLIPPING.

Stereo-clipping is done by a modified Cohen-Sutherland algorithm, Sutherland-Hodgman algorithm and others. We verified that stereo-clipping had visual effect by psychological experiments.

1. はじめに

近年、ステレオ画像の表示に用いる電子式シャッタ装置の低価格化などによって、様々な分野でステレオ画像が用いられるようになってきている⁽¹⁾。しかし、CRTディスプレイなどに2眼式の立体画像を表示する場合、画面の大きさが限られているため、表示物体の一部が右目からは画面内に見えても左目からは画面の外に出てしまっていて見えない、或いはその逆の現象が生じる。このため、画面の両端で立体感を損ねたり、違和感を覚えたりすることがある。また、右目と左目の対応点が正確に把握できないため、誤った立体認識をしてしまうことも考えられる。この現象を避けるためには、右目と左目のどちらからみても画面内に見える領域だけを表示し、それ以外の領域を切り捨てることが考えられる。以後、右目及び左目から画面内に見える領域をそれぞれ右目視界、左目視界と呼び、更にこれらの重なった領域を両目視界と呼ぶことにする。また、この両目視界に対して3次元クリッピングを行なうことをステレオクリッピングと呼ぶ。

ステレオクリッピングとは、クリッピングボリュームとして両目視界を用いることを指すので、実際にステレオクリッピングを行なうためには様々な方法が考えられる。本報告では以下の3つの手法について述べる。

- (i) Cohen-Sutherlandアルゴリズムの拡張
- (ii) Sutherland-Hodgmanアルゴリズム
- (iii) Zバッファアルゴリズムの応用

また、単純に対象物体に対してステレオクリッピングを行なった場合、壁の向こう側など本来見えないはずの部分が見えるようになってしまい逆に違和感が生じることがある。そこで、ステレオクリッピングを行なった画像に対して適正な隠面消去を施すためのアルゴリズムの改良についても述べる。

実際に作成した立体画像に対して評価実験を行った結果、少なくとも、静止画像に対してはステレオクリッピングによる立体感の向上が確認できた。

2. クリッピングボリュームの形状

先に述べたように、クリッピングボリュームとしては両目視界を用いる。右目視界、左目視界及び両目視界の様子を図1に示す。クリッピングボリュームを定義するクリッピング平面は、図2に示す様に、 xz 平面に垂直なFR, FL, BR, BLの4枚に yz 平面に垂直なTp, Btの2枚を加えた計6枚である。ここで、右目ER及び左目ELはそれぞれ x 軸上の $(d, 0, 0)$, $(-d, 0, 0)$ にあるとする。また、画面は $z=S$ にある xy 平面に平行な平面で、 x 方向には $-X$ から X まで、 y 方向には $-Y$ から Y までを表示領域とする。

3. ステレオクリッピングの実現方法

3. 1 Cohen-Sutherland アルゴリズムの拡張

まず、従来のCohen-Sutherlandクリッピングアル

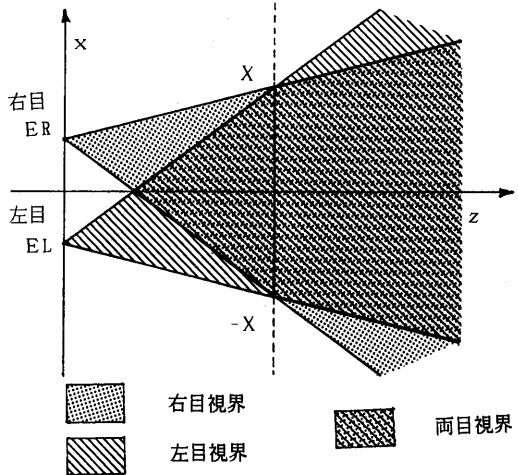
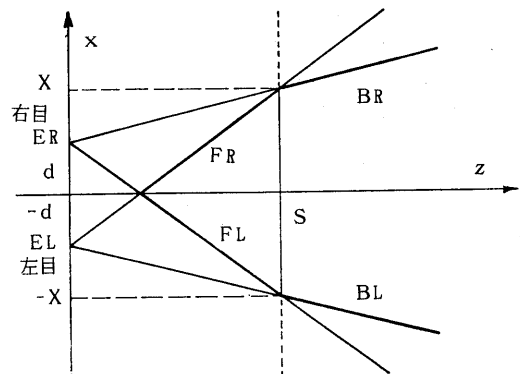
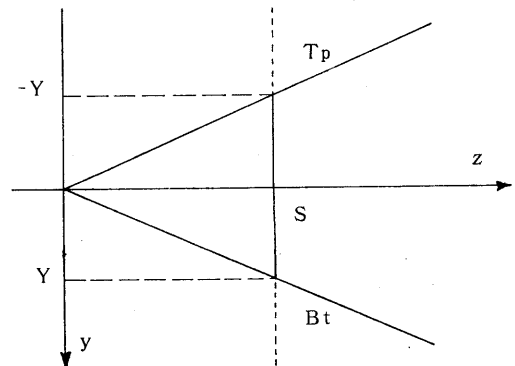


図1 右目視界、左目視界、両目視界



FR, FL, BR, BLは、
 xz 平面に垂直なクリッピング平面
(a) 平面図



Tp, Btは、 yz 平面に垂直なクリッピング平面
(b) 側面図

図2 クリッピング平面

ゴリズムについて説明する⁽²⁾。Cohen-Sutherlandアルゴリズムでは、まず、直線の両端点がそれぞれクリッピングウインドウに対してどの様な位置にあるかを表す4ビットのコードを生成する。各ビットは次の条件によって1にセットされる。

- 第1ビット-端点がウインドウの左側にある
- 第2ビット-端点がウインドウの右側にある
- 第3ビット-端点がウインドウの下側にある
- 第4ビット-端点がウインドウの上側にある

こうして生成される端点コードは図3に示すようになる。この端点コードを用いることによって、直線が完全に可視であるか完全に不可視であるかを速やかに決定することができる。即ち、両端点のコードが共に0000であれば明らかに可視である。また、両端点のコードの論理積をとった結果が0000でないならば完全に不可視だということができる。このどちらでもない場合には、可視とも不可視とも決定できないので、左右上下のウインドウエッジに対して、直線とウインドウエッジとの関係を順に調べて行かなければならない。そして、直線がウインドウエッジのどれかをまたいでいる場合には、そのウインドウエッジとの交点を求め、外側だった端点とこの交点とを交換する。このようにして一つのウインドウエッジによってクリップされたら、再び端点コードを生成し直し、完全に可視か完全に不可視だといえるまでこの処理を繰り返す。

Cohen-Sutherlandクリッピングアルゴリズムの3次元への拡張は、従来、直方体かビュー切頭体のクリッピングボリュームに対して考えられていた⁽³⁾。即ち、4ビットであった端点コードにクリッピングボリュームの前後を表す2ビットを加えて6ビットの端点コードを生成する。以下の処理は2次元の場合と同様である。

このCohen-Sutherlandアルゴリズムをステレオクリッピングに適用することは一見困難に考えられる。しかし、端点コードの各ビットが持っている意味を、クリッピングボリュームの上側や下側という意味から、図2に示す各クリッピング平面の内側外側とい

1001	1000	1010
0001	0000 ウインドウ	0010
0101	0100	0110

図3 端点コード

う意味に考え直すことによって、ステレオクリッピングへの適用が可能になる。即ち、次のような条件によって6ビットの端点コードを生成する。

- 第1ビット-端点がFRの外側にある。
- 第2ビット-端点がFLの外側にある。
- 第3ビット-端点がBRの外側にある。
- 第4ビット-端点がBLの外側にある。
- 第5ビット-端点がTpの外側にある。
- 第6ビット-端点がBtの外側にある。

このようにして生成された端点コードは図4のようになる。この場合にも、完全に可視、或いは完全に不可視の決定は従来のコードと同様に行うことができる。ここで、点P(x, y, z)がそれぞれのクリッピング平面の外側にある条件は、次式の様になる。

$$FR: z < \frac{S}{(x+d)} (x+d) \quad (1)$$

$$FL: z < -\frac{S}{(x+d)} (x-d) \quad (2)$$

$$BR: z < \frac{S}{(x-d)} (x-d) \quad (3)$$

$$BL: z < -\frac{S}{(x-d)} (x+d) \quad (4)$$

$$Tp: z < -\frac{S}{Y} y \quad (5)$$

$$Bt: z < \frac{S}{Y} y \quad (6)$$

ここで、これらの式の不等号を等号に置き換えると各クリッピング平面の方程式となる。したがって、FR, FL, BR, BLの方程式は次のように一般化できる。

$$z = \alpha (x + \beta) \quad (7)$$

また、Tp, Btは次のように一般化できる。

$$z = \alpha y \quad (8)$$

一方、点P1(x1, y1, z1)から点P2(x2, y2, z2)への直線のパラメトリック表示は次式で表される。

$$P(t) = P1 + (P2 - P1)t \quad (0 \leq t \leq 1) \quad (9)$$

従って、(7)(8)(9)式より、直線P1P2と各クリッピング平面との交点を与えるt0を求めると次のようになる。

$$FR, FL, BR, BL \text{ については } t0 = \frac{z1 - \alpha(x1 + \beta)}{z1 - z2 + \alpha(x2 - x1)} \quad (10)$$

$$Tp, Bt \text{ については } t0 = \frac{z1 - \alpha y1}{z1 - z2 + \alpha(y2 - y1)} \quad (11)$$

したがって、このt0を(9)式のtに代入することによってクリッピング平面と直線との交点を求める

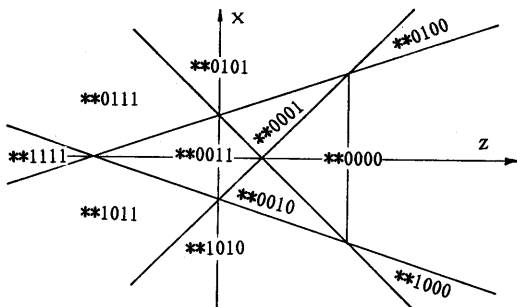
ことができる。

このようにして得られた交点を用いて、従来のCohen-Sutherlandアルゴリズムと同様の処理を行えば、直線に対するステレオクリッピングを実現することができる。

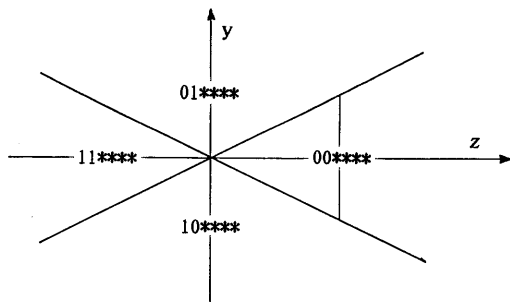
3. 2 Sutherland-Hodgmanアルゴリズムの適用

これまでに述べたアルゴリズムがラインクリッピングのアルゴリズムであったのに対して、このSutherland-Hodgmanアルゴリズム⁽⁴⁾は多角形クリッピングのアルゴリズムである。このアルゴリズムの基本的な考え方は、図5に示す様に、元の多角形を1つのウィンドウエッジでクリップし、これによって得られた中間多角形を各ウィンドウエッジで順次クリッピングしていくというものである。このアルゴリズムは、任意の凸多角形のクリッピングウィンドウ、或いは任意の凸多面体クリッピングボリュームに対して有効なので、ステレオクリッピングにそのまま適用することができる。

元の多角形をそれぞれのウィンドウエッジで順にクリッピングしていくことによって多角形のクリッピングが行えることは、明らかである。そこで、一つのウィンドウエッジによるクリッピングについて説明する。



ただし、* はyの値によって
値が変わることを示す
(a) 平面図



ただし、* はxの値によって
値が変わることを示す
(b) 側面図

図4 ステレオクリッピングのための端点コード

まず、入力された多角形の頂点列の最初の点は、可視か不可視かだけを調べて多角形の頂点列の一番後ろにコピーされる。この後、順に入力多角形の各頂点について調べて行くわけだが、ウィンドウエッジと隣あった2つの頂点 P_i , P_{i+1} との関係は、次の4通りしかおこり得ない。

- a) 両端点とも可視
- b) 両端点とも不可視
- c) P_i が可視, P_{i+1} が不可視
- d) P_i が不可視, P_{i+1} が可視

この4つの状態に対してそれぞれ次のように頂点を出力すればよい。ただし、ウィンドウエッジと辺 $P_i P_{i+1}$ との交点をJとする。

- a) P_{i+1} を出力
- b) 出力無し
- c) Jだけを出力
- d) Jを出力しさらに P_{i+1} を出力

ここで、ウィンドウエッジと辺 $P_i P_{i+1}$ との交点Jを求める方法はどのようなものであってもよい。

以上のアルゴリズムにおいて、各頂点の可視不可視の判定、及び交点Jの算出に、3. 1で述べたCohen-Sutherlandアルゴリズムを用いることによって、Sutherland-Hodgmanアルゴリズムは、そのまま、ステレオクリッピングに適用することができる。

3. 3 Zバッファアルゴリズムの応用

右目に着目して両目視界を捕らえると、FL, BR, Tp, 及びBtは実際に画面の端になる。即ち、スクリーン座標系に変換したこれらのクリッピング平面の方程式はそれぞれ、 $x_s = -X$, $x_s = X$, $y_s = Y$, $y_s = -Y$ となる。但し、スクリーン座標系を(x_s , y_s , z_s)とする。したがって、これらの面に対しては、スクリーン座標系でクリッピングを行った方が容易であると考えられる。一方、残ったFR及び

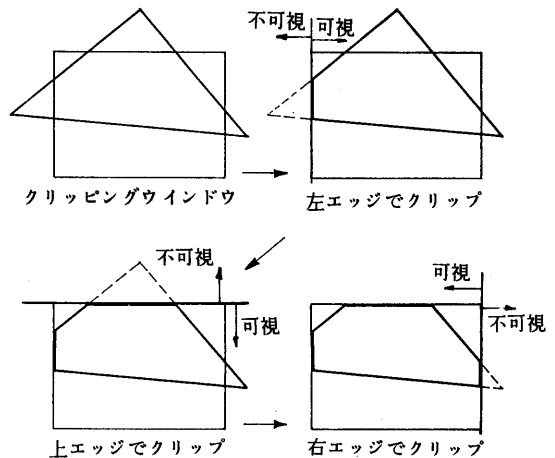


図5 Sutherland-Hodgman多角形
クリッピングアルゴリズム

BLをスクリーン座標系に変換すると次式の様になる。
FRの方程式

$$zs = \frac{-1}{2d \cdot S} xs + \frac{X+2d}{2d \cdot S} \quad (12)$$

$$= F(xs)$$

BLの方程式

$$zs = \frac{-1}{2d \cdot S} xs + \frac{2d-X}{2d \cdot S} \quad (13)$$

$$= B(xs)$$

したがって、あるxs に対して、zs が可視である条件は、次式が成り立つことである。

$$B(xs) \leq zs \leq F(xs) \quad (14)$$

そこで、隠面消去にZバッファアルゴリズムを用いている場合には、各画素毎にzsの値をZバッファと比較するときに、同時にB(xs)、F(xs)との比較も行うことによって、結果としてステレオクリッピングを実現することができる。具体的には、B(xs)との比較は、ZバッファをB(xs)の値で初期化するだけでよい。F(xs)との比較には実際にF(xs)の値を持った1次元配列を用意し、描画しようとする点Ps(xs, ys, zs)のzsの値をF(xs)の値と比較する。そして、Psのzsの値が、Zバッファの値よりも大きく、F(xs)の値よりも小さい場合のみZバッファの値を更新し、その点の描画を行うようにすることによってFR及びBLに対するクリッピングが行われたことになる。

左目用の画像については、これと左右対称に行えばよいので、説明は省略する。

4. 適正な隠面消去手法

4. 1 隠面消去における異常の発生理由

前節で述べた手法によってステレオクリッピングを行なった場合、物体の前部を削除してしまうために、図6に示すように、本来見えないはずの部分が見えるようになってしまい、逆に違和感が生じることがあった。

ここでは、先に述べたSutherland-Hodgmanアルゴリズムによる手法とZバッファアルゴリズムの応用による手法に対して、隠面消去が適正に行なわれるようにする方法について述べる。

4. 2 Sutherland-Hodgman アルゴリズムの応用

ここでは、右目の場合について説明する。まず大まかな手順を以下に示す。

- (1) クリッピング平面FRを除いた全てのクリッピング平面で、描画しようとするポリゴンをクリッピングする。
- (2) クリッピングされたポリゴンをクリッピング平面FRによって、図7に示す様にクリッピングポリゴンの内側のポリゴンと、外側のポリゴンとに分割する。以下、内側のポリゴンを内部ポ

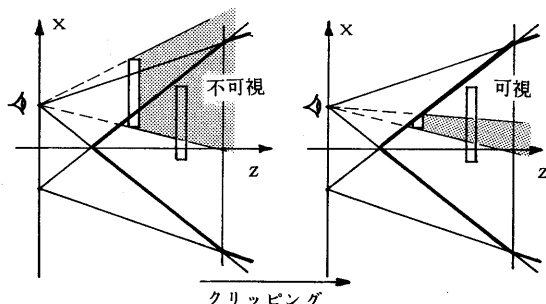


図6 隠面消去の異常の発生理由

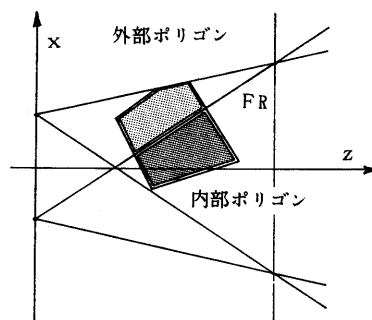


図7 内部ポリゴン 及び 外部ポリゴン

リゴン、外側のポリゴンを外部ポリゴンと呼ぶ。

(3) 内部ポリゴンは別に求めた色で描画し、外部ポリゴンはストックする。

(4) 全てのポリゴンの処理が終わった後、ストックされた外部ポリゴンを背景色で描画する。

ここで、ポリゴンの切断は、3. 2で述べたクリッピングの際の頂点出力を内部ポリゴンと外部ポリゴンに対して並列的に行うことによって実現される。即ち、内部ポリゴン、外部ポリゴンそれぞれに対して次のように頂点を出力する。

- a) 両端点とも可視の場合
内部ポリゴン: P_{i+1} を出力
外部ポリゴン: 出力無し
- b) 両端点とも不可視の場合
内部ポリゴン: 出力無し
外部ポリゴン: P_{i+1} を出力
- c) P_i が可視, P_{i+1} が不可視の場合
内部ポリゴン: Jだけを出力
外部ポリゴン: Jを出力し更に P_{i+1} を出力
- d) P_i が不可視, P_{i+1} が可視の場合
内部ポリゴン: Jを出力し更に P_{i+1} を出力
外部ポリゴン: Jだけを出力

この様にして得られた内部ポリゴンと外部ポリゴンを用いて上述の処理を行えば、ステレオクリッピン

グを行った画像の隠面消去を適正化することができ
る。左目に関しては、これと対称のことを行なえば
よい。

4. 3 Zバッファの応用による手法

Zバッファアルゴリズムの応用による手法に関し
ては、隠面消去を適正化するための修正は極めて容
易である。3. 3で述べた、点の表示に関する処理
を次のように変更すればよい。

まず、Zバッファの更新だけを先に行なう。即ち、
表示しようとする点の z_s の値をZバッファの値と比
較し、Zバッファよりも手前ならばZバッファの値
を更新、そうでなければ何もせず次の点に移る。

次に、表示しようとする点の z_s の値を $F(x_s)$
の値と比較し、 $F(x_s)$ よりも手前ならば背景色で
その点を表示し、そうでなければ本来の表示すべき
色で表示する。この様にして、適正な隠面消去を行
うことができる。

5. ステレオクリッピングの処理例

5. 1 作成した画像例

ステレオクリッピングを行っていない画像の例
を図8に、単純にステレオクリッピングを行なった
画像の例を図9に、適正な隠面消去を行なった画像
の例を図10に示す。この結果から、図8でははつ
きりとわからなかった手前の壁の向きが、図9では
かなり正確にわかることがわかる。また、図9で見
えていた手前の壁の内側が、図10ではきれいに消
去されていることがわかる。

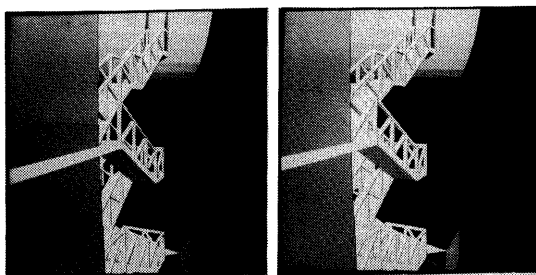
5. 2 処理速度の比較

Cohen-Sutherlandアルゴリズムはラインクリッ
ピングのアルゴリズムであり、Sutherland-Hodgmanア
ルゴリズムとZバッファアルゴリズムを応用した手
法は多角形クリッピングのアルゴリズムであるので、
別々に評価を行った。

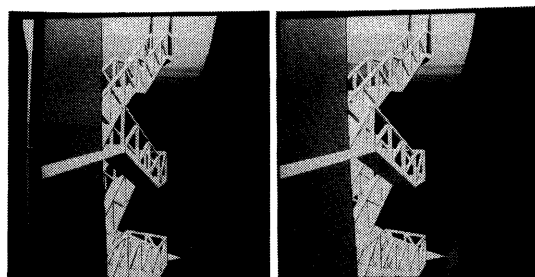
まず、Cohen-Sutherlandアルゴリズムを用いた場
合とステレオクリッピングを行わない場合とで処理
速度を比較した。実験に用いたデータは、ワイヤー
フレームで描かれたキューブの動画像であり、大体
全体の1/3程度が画面外に出ている。キューブは
36本の線からできており、72枚の動画像を作成す
るのに要した時間を比較した。使用したコンピュータ
はPC9801F2である。実験結果を以下に示す。

Cohen-Sutherlandアルゴリズム	163秒
ステレオクリッピングなし	131秒

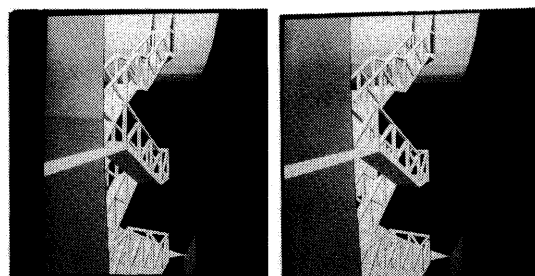
ただし、この時間は座標変換なども含めた描画全体
の処理時間である。また、ステレオクリッピングを
行わなかった場合のクリッピングは透視変換を施し
た後、右目用と左目用それぞれの画像に対して、従
来のCohen-Sutherlandの2次元クリッピングアルゴ
リズムを用いた。



(a) 右画像 (b) 左画像
図8 ステレオクリッピングを
行っていない画像の例



(a) 右画像 (b) 左画像
図9 単純にステレオクリッピングを
行なった画像の例



(a) 右画像 (b) 左画像
図10 隠面消去を適正化して
ステレオクリッピングを行った画像の例

次に、以下の5つの場合の処理速度を比較する。

- ステレオクリッピングを行わない場合
- Sutherland-Hodgmanアルゴリズムを用いて、隠面消去を適正化しないステレオクリッピングを行った場合
- Zバッファアルゴリズムを用いて、隠面消去を適正化しないステレオクリッピングを行った場合
- Sutherland-Hodgmanアルゴリズムを応用して隠面消去を適正化したステレオクリッピングを行った場合
- Zバッファアルゴリズムを応用して隠面消去を適正化したステレオクリッピングを行った場合

これらに対して図8～10を作成するのに要した時

間を比較した。使用したコンピュータはVAX11/750である。実験結果を以下に示す。

a)	166秒
b)	165秒
c)	184秒
d)	171秒
e)	186秒

ただし、この値も描画全体の処理時間である。また、a)、c)、e)のスクリーン座標系でのクリッピングには従来のCohen-Sutherlandアルゴリズムを用いた。

この結果から、b)が最も速いことがわかるが、これは、全体の処理時間のほとんどが各画素毎のz値の計算とzバッファとの比較に費やされるために、できるだけ早い段階で描画しない部分を切り捨てているためだと考えられる。

6. 視覚的效果に対する評価

6. 1 動画像に対する評価実験

2眼式の立体映画などにおいて物体が画面の外へ動いて行くときに生じる違和感が、ステレオクリッピングを行うことによって低減されるかどうかを調べるために、次の実験を行った。

ワイヤーフレームで描かれたキューブの2眼式立体アニメーションに対して、ステレオクリッピングを行ったものと行っていないものを用意した。これをアナグリフ方式を用いて被検者に見せ、“どちらの方がより自然な感じがするか”とたずねた。この結果を表1に示す。

この結果から、立体映画などで物体が画面の外に動いていく場合の違和感の低減に対しては、ステレオクリッピングは効果がないと言える。

6. 2 静止画像に対する評価実験

ステレオクリッピングが2眼式立体画像の立体感の向上に効果があったかどうかを調べるために次の実験を行った。

図11、図12に示す2種類の画像に対して、ステレオクリッピングを行ったものと行っていないものを用意して、これらを被検者に見せ、“どちらの方がより立体的に見えるか”とたずねた。この結果を表2に示す。

この実験結果から、静止画像については動画像とは違い、ステレオクリッピングを行うことによって、2眼式立体画像の立体感が向上したといえる。

そこで更に、6.1で用いたワイヤーフレームの動画像の内の1枚を取り出して、静止画像に対する評価実験を行った。この結果を表3に示す。

この結果から、ステレオクリッピングは、同じ画像を用いても、動画像の場合よりも静止画像の場合に効果的であることがわかる。このことから、動画像の場合には立体形状を正確に把握する時間的余裕

がないため、ステレオクリッピングが行われていてもいなくてもさほど差が感じられないものと考えられる。

6. 3 立体形状把握の正確さに関する実験

ステレオクリッピングを行わない場合には、画面の両端において、左右の画像の対応点が正確に見つけることが難しいため、立体形状の把握が困難になったり、立体形状を誤解してしまうことが考えられる。これに対して、ステレオクリッピングを行った場合には、画面の両端でも左右の対応点が比較的容易に見つけられるため、立体形状の把握が行い易くなると考えられる。このことについて調べるために、次の実験を行った。

図13に示す様な静止パターン9種類に対して、ステレオクリッピングを行ったものと行っていないものを用意する。これらのパターンは全て、パターンの右端は視点からの距離を同じに揃えてあり、パターンの左端、即ち、画面よりわずかに外にある部分では前後に距離が取ってある。これらのパターンをランダムに並べ変えてから、アナグリフ方式を用いて被検者に見せ、上の図形と下の図形のどちらが手前に出ている様に見えるかを答えてもらった。

この実験を10人に対して行い、正しく答えた回数、誤って答えた回数、及び解らないと答えた回数の合計を表4に示す。

この結果から、ステレオクリッピングを行うことによって、画面の端における立体形状の把握が正確になっていることがわかる。

表1 動画像に対する評価実験の結果

より自然に見えると答えた人数	
クリッピングあり	クリッピングなし
11	9

表2 静止画像に対する評価実験の結果

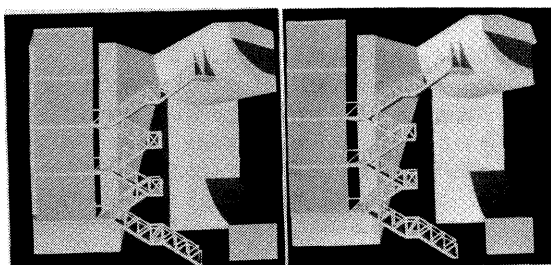
	より立体的に見えると答えた人数	
	クリッピングあり	クリッピングなし
画像A	18	2
画像B	17	3

表3 動画像の一部を用いた、静止画像に対する評価実験

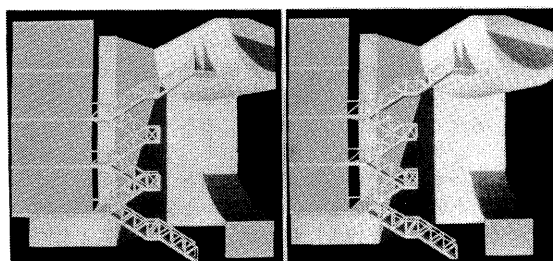
より立体的に見えると答えた人数	
クリッピングあり	クリッピングなし
14	6

表4 形状把握実験の結果

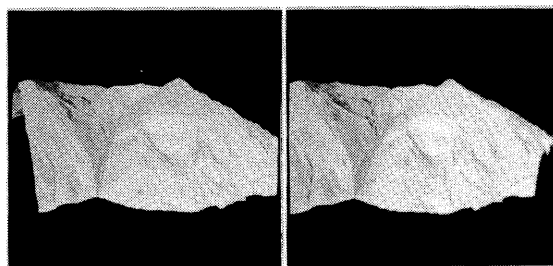
	正答	誤答	解らない
クリッピングあり	71	3	16
クリッピングなし	37	15	38



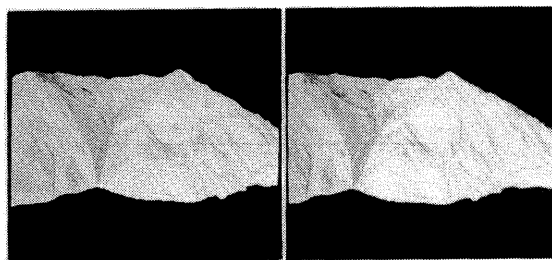
(a) ステレオクリッピングを行った画像



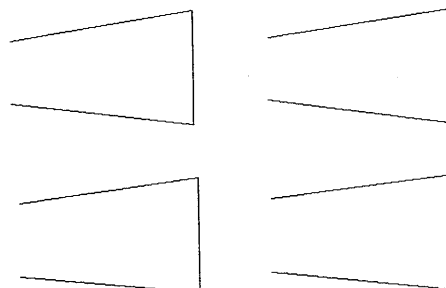
(b) ステレオクリッピングを行っていない画像
図 1.1 評価実験に用いた2眼式立体画像 A



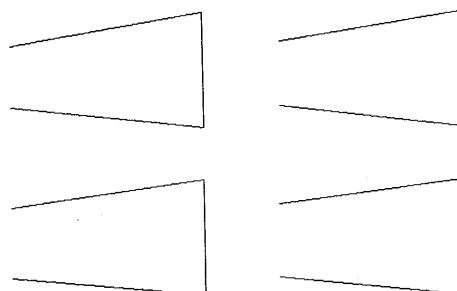
(a) ステレオクリッピングを行った画像



(b) ステレオクリッピングを行っていない画像
図 1.2 評価実験に用いた2眼式立体画像 B



(a) ステレオクリッピングを行った画像



(b) ステレオクリッピングを行っていない画像
図 1.3 実験に用いたテストパターン例

7. まとめ

ステレオクリッピングには立体形状の把握を正確にする効果があることが、6. 3で行った実験の結果からわかる。したがって、6. 2で行ったように静止画像をじっくりと見る場合には、ステレオクリッピングを行ってある方が立体形状を正確にとらえられるために、より立体的だと感じるのだと考えられる。

また、5. 2で行った処理速度の比較の結果から、Sutherland-Hodgmanアルゴリズムを用いれば、ステレオクリッピングによって描画速度を上げることができることも明らかとなった。

参考文献

- (1) 安居院・中嶋：“ステレオグラフィックス&ホログラフィ”，産業報知センタ，pp.192-194（1985）
- (2) W. M. Newman, R. F. Sproull, "Principles of Interactive Computer Graphics 2nd ed.", McGraw-Hill(1979).
- (3) D. F. Rogers, "Procedural Elements for Computer Graphics", McGraw-Hill(1985).
- (4) I. E. Sutherland, and G. W. Hodgman, "Re-entrant Polygon Clipping", Comm. ACM, Vol. 17, pp. 32-42(1974)