実画像変形による科学技術計算の可視化*

小 木 哲 朗^{*1}, 廣 瀬 通 孝^{*2}

Scientific Visualization Based on Warping of Photograph Image

Tetsuro OGI and Michitaka HIROSE

In scientific visualization, displaying numerical data using realistic image is essential to recognize physical phenomena effectively. In this paper, a visualization method using photograph image was proposed. By relating the analysis model that is used in the numerical simulation to the shape model and the photograph image, the warped image is generated in real-time according to the results of the numerical simulation. This method was applied to the visualization of scientific data such as an earthquake response analysis and a deformation analysis. As results, realistic images were generated to visualize data in real-time. In addition, the distortion of the generated image that depends on the rotation or the deformation of the displayed objects was evaluated. This distortion can be decreased by increasing the number of the control points for the warping of the photograph image.

Key Words: Image Processing, Human Interface, Computer Aided Analysis, Scientific Visualization, Virtual Reality, Image-based Rendering, Numerical Simulation, Supercomputing

1. 緒 言

科学技術計算の結果を可視化するサイエンティフィック・ビジュアライゼーションにおいて、バーチャルリアリティは有効な技術として期待されている⁽¹⁾。バーチャルリアリティを利用した科学技術データの可視化方法としては、これまでに BOOM などの視点追従形のディスプレイ装置を用いる方法⁽²⁾や、CAVE などの没入型仮想空間を用いる方法が研究されている⁽³⁾⁽⁴⁾。また視覚化ではないが、力覚フィードバックや振動感覚、風覚等の触覚情報を利用したデータ提示に関してもいくつかの研究が行われている^{(5)~(7)}。これらの方法では、立体視や多感覚情報を利用することで、数値データを直観的に理解させることができる程度可能になってきた。しかしながら、提示されている情報は計算で用いられたデータそのものであるため、対象としている物理現象を理解できるとはいえない場合が多い。

サイエンティフィック・ビジュアライゼーションの目的としては、研究者が現象を探索する手段として使

用する場合と、第三者へのプレゼンテーションとして使用する場合がある。特に、後者のプレゼンテーションを目的とした場合には、単に計算結果を提示するのではなく、現実の物理現象と結びつけた表現を行うことが要求される。つまり、映像としてもリアリティの高い可視化方法を実現することが望まれる。

一方、バーチャルリアリティの研究分野では、近年リアリティの高い映像を生成する技術として、イメージベースト・レンダリングと呼ばれる方法が注目を集めている^{(8)~(10)}。写真等の実画像を直接利用することで、実画像のリアリティ品質を保った三次元世界を合成する方法である。この方法では、従来の三次元モデル+テクスチャという方法に比べて、細かいモデリングを行わなくてもリアリティの高い自然な画像を生成できるという利点がある。

本研究では、この種の技術をサイエンティフィック・ビジュアライゼーションの手段として適用する方法について検討を行っている⁽¹¹⁾。すなわち、科学技術計算の結果を実画像と結び付けることによって、映像として実写性の高い提示方法の開発を行うこと、またその有効性について検討を行うことが目的である。以下本論文では、提案する可視化方法の基本原則を説明し、具体的な適用例、および生成画像に含まれる誤差

* 原稿受付 1998 年 2 月 19 日。

*¹ 正員, 東京大学インテリジェント・モデリング・ラボラトリー (〒113-8656 東京都文京区弥生 2-11-16)。*² 正員, 東京大学工学系研究科 (〒113-8656 東京都文京区本郷 7-3-1)。

E-mail: tetsu@iml.u-tokyo.ac.jp

評価について論じる。

2. 実画像を利用した可視化方法

2.1 形状モデルと実画像の対応 本研究ではおもに、変形解析、振動解析等の物体の移動や変形を伴う数値シミュレーションを問題として取上げ、実画像を利用して計算結果を可視化する方法について考える。この際、物体の移動や変形に対する画像をリアルタイムで合成することが要求される。

物体の移動や変形を扱うためには、対象物の形状に関して何らかの情報を持っていることが必要であるが、数値シミュレーションでは解析対象の形状はあらかじめわかっていることが前提となる。そのため、対象物に関しては3次元の簡易形状モデルを定義し、これを画像合成に利用することができる。

一般に写真として撮影された画像は、図1に示すように形状モデルをカメラの撮影条件に従って透視変換することによって決定される。今、対象としている物体の形状モデル上の点を $P(x, y, z)$ 、これを撮影した画像上の点を $Q(u, v)$ とすると、両者の関係は透視変換行列 M により次式で表される。

$$Q = MP \quad \dots\dots\dots (1)$$

この関係式をもとに、カメラの撮影条件、すなわち透視変換行列 M を求めることを考える⁽¹²⁾。式(1)は三次元空間から二次元平面への変換であるため、同次座標 (x^*, y^*, z^*, w^*) を用いると、次のような行列式で表現することができる。

$$\begin{pmatrix} x^* \\ y^* \\ - \\ w^* \end{pmatrix} = \begin{pmatrix} a1 & a2 & a3 & a4 \\ a5 & a6 & a7 & a8 \\ - & - & - & - \\ a9 & a10 & a11 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \dots\dots\dots (2)$$

$$(u, v) = \left(\frac{x^*}{w^*}, \frac{y^*}{w^*} \right) \quad \dots\dots\dots (3)$$

上式において、 $a1 \sim a4$ と $a5 \sim a8$ はそれぞれ独立に

Fig.1 Analysis model, shape model and photograph image

計算することが可能なため、 $P(x, y, z)$ と $Q(u, v)$ の対応点の組が与えられると、 $a1 \sim a11$ は2組の七元連立一次方程式に分けられる。そのため、形状モデル上の点 $P(x, y, z)$ と実画像上の点 $Q(u, v)$ に関し、7点以上の座標の対応を取ることができれば、透視変換行列 M は計算できることになる。

変換行列 M がいったん求められると、対象物の移動や変形に対しても、形状モデル上の点 P' に対する実画像上の対応点 Q' は、

$$Q' = MP' \quad \dots\dots\dots (4)$$

によって容易に計算することができる。

2.2 実画像変形 変形後の実画像上の対応点が計算されると、本手法では元の画像を変形させることで新しい画像を生成する。例えば形状モデル上の点 P_1, P_2, P_3 が P'_1, P'_2, P'_3 に移動し、対応する実画像上の点が Q_1, Q_2, Q_3 から Q'_1, Q'_2, Q'_3 に移動したとする。このとき、変形後の画像は P'_1, P'_2, P'_3 から幾何モデルとしてレンダリングするのではなく、画像上で Q_1, Q_2, Q_3 で囲まれた画像領域を Q'_1, Q'_2, Q'_3 へワーピングの変換を行うことによって新たな画像を生成する⁽¹³⁾。

この際、内部領域の画像は制御点の移動に従って均等に伸縮される。図2は画像の変形方法を示したものであるが、図2のように三角形の内部の点 R は、各辺に対する距離の比が等しくなるような点 R' へ移動される。すなわち、点 R と点 R' に関しては次式が成り立つことになる。

$$\frac{r1}{q1} = \frac{r1'}{q1'} \quad \dots\dots\dots (5)$$

$$\frac{r2}{q2} = \frac{r2'}{q2'} \quad \dots\dots\dots (6)$$

$$\frac{r3}{q3} = \frac{r3'}{q3'} \quad \dots\dots\dots (7)$$

ここで $r1, r2, r3$ はそれぞれ点 R から辺 $Q_2-Q_3, Q_3-Q_1, Q_1-Q_2$ へ垂線を下ろした距離、 $q1, q2, q3$ はそれぞれ頂点 Q_1, Q_2, Q_3 と辺 $Q_2-Q_3, Q_3-Q_1, Q_1-Q_2$

Fig.2 Method of generating warped image

との距離である。この変換は、実際には OpenGL グラフィックス・ライブラリーの二次元テクスチャの機能を用いてインプリメントが行われている。

またこの方法では、形状モデルは実画像変形のための制御点の位置計算にしか使わないため、詳細なモデルは必要ではなく、形状の輪郭を表す程度の簡単なモデルで十分に表現することができる。

2・3 解析モデルと形状モデルの対応 一方、数値シミュレーションでは、解析計算のためのモデルを使用する。この解析モデルと前述の形状モデルの間の対応付けを行うことによって、数値シミュレーションの結果と実画像の変形を結び付けることができる(図1)。しかしながら、この両者のモデルは、一般に座標系やモデルの持ち方がかなり異なる場合が多い。例えば、形状モデルとしては対象物の輪郭を表す三次元形状を定義しながら、解析モデルとしては対称性から二次元メッシュ、あるいは一次元の質点系モデル等で近似する場合がある。

そのため、解析モデルを用いた計算結果を形状モデルに反映するためには、解析モデル座標系から形状モデル座標系に変換し、寸法や対称性等の形状に関する知識をもとに必要な情報を補う等の対応付けの処理が必要になる。このように、解析モデル、形状モデル、実画像上の制御点の間の対応付けが行われることによって、数値シミュレーションの結果を実画像変形として描画することが可能になる。

2・4 二次元画像の合成 本可視化方法では、二次元画像の変形として処理を行うため、他の画像とも実写性を保った自然な合成を行えることが特徴である。

画像間の合成として最も単純な方法は、切り貼り合成である。この方法は、図3に示すように、一方の画像のある領域を二次元画像として切り抜き、他方の画像上にはり付けるといった方法である。この方法では、

Fig. 3 Image composition by cut and paste of original images

それぞれの画像の撮影条件が異なる場合、厳密に正しい画像が合成される訳ではないが、解析対象ではない背景等に用いる上では十分な場合が多い。

合成する画像に関して厳密な整合性を取りたい場合には、解析対象と同様に合成する画像部分に対しても、形状モデルを定義する必要がある。与えられた画像から透視変換行列を求める方法に関しては前述したが、この変換行列からさらにカメラの視点位置、視軸、画角等の撮影条件を明示的に求めることができる。

今、カメラモデルの視点位置を c_x, c_y, c_z 、参照点を r_x, r_y, r_z 、カメラのツイスト角を γ 、視体積のクリップ面の左右、上下、遠近の座標をそれぞれ l, r, t, b, f, n とすると、変換行列 M は次式で表すことができる⁽¹⁴⁾。

$$\begin{aligned}
 M &= \begin{pmatrix} a1 & a2 & a3 & a4 \\ a5 & a6 & a7 & a8 \\ - & - & - & - \\ a9 & a10 & a11 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-f-n}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \\
 &\times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &\times \begin{pmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &\times \begin{pmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -r_x \\ 0 & 1 & 0 & -r_y \\ 0 & 0 & 1 & -r_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots\dots\dots (8)
 \end{aligned}$$

ここで、

$$\alpha = \tan^{-1} \left(\frac{r_y - c_y}{\sqrt{(r_z - c_z)^2 + (r_x - c_x)^2}} \right) \quad \dots\dots\dots (9)$$

$$\beta = \tan^{-1} \left(\frac{r_x - c_x}{r_z - c_z} \right) \quad \dots\dots\dots (10)$$

$$d = \sqrt{(r_x - c_x)^2 + (r_y - c_y)^2 + (r_z - c_z)^2} \quad \dots\dots\dots (11)$$

である。式(8)の第1項は射影変換、第2項以降は視界変換を示している。

ここで式(2), (3)から求められた変換行列 $M(a1 \sim a11)$ の値が与えられると, 式(8)~(11)は連立方程式として解くことができ, これから視点位置, 参照点, ツイスト角, 視体積のクリップ面等のカメラ情報に関する変数を求めることができる. このようにして, 二つの画像に関する視点情報がそれぞれ求められると, 一方の視点を他方に一致させるように, 視点移動の変換を行うことによって, 二つの画像を正確に合成することができる. 図4は以上の視点位置を考慮した画像合成の方法を図示したものである. 例えば, 画像1の Q_1 部分を視点移動 V によって, 画像2の Q_2 の視点に合わせる場合, 合成画像上の Q_1, Q_2 はそれぞれ次式に従って生成されることになる.

$$Q_1 = M_1 VP_1 \dots\dots\dots (12)$$

$$Q_2 = M_2 P_2 \dots\dots\dots (13)$$

ここで M_1, M_2 は画像1, 2に対する透視変換行列である.

3. 数値シミュレーションの可視化例

3.1 システム構成 実画像を利用した数値シミュレーションの可視化方法の有効性を確認するため

Fig. 4 Image composition using three dimensional model

に, 具体的な問題への適用を行った. 数値計算を行うための計算機としては並列スーパーコンピュータ IBM SP 2 (50 CPU) を用い, レンダリングを行うためのグラフィックス・ワークステーションとしては SGI i-Station を用いた. 両者の計算機間は FDDI の専用ネットワークで接続され, SP 2 で計算した数値データを i-Station へ送りながら, リアルタイムで可視化を行うことができる.

3.2 地震応答解析の可視化 応用例としてはまず, 地震発生時に建物がどのように揺れるかを計算する地震応答解析の問題を取上げた. この種の問題では, 一般に質点系等の非常に単純なモデルを用いて計算されることが多いが, この計算結果を実際の建物の画像と結び付けることで, 映像として実写性の高い可視化を行うことがここでの課題である.

図5は対象として取上げた建物の画像上に, 本可視化手法で用いる解析モデル, 形状モデルを示したものである. 70階建ての建物であるため, 解析モデルとしては70自由度の質点系モデルを用い, これに地震動を与えた時の応答を線形加速度法によりリアルタイムで計算する. 画像変形に利用する形状モデルとしては, 各階ごとの建物の輪郭形状でモデル化を行っている

Fig. 5 Analysis model and shape model of the building

Fig. 6 Visualization of earthquake response analysis using photograph image

る。この形状モデルの各頂点と画像上の座標を対応付けることによって、透視変換行列が求められる。

地震応答解析の計算により、解析モデルの各質点の動きが計算されると、これに従って形状モデルの各階の動きが決定される。この形状モデルの動きに透視変換を与えることで、実画像上の制御点の動きが計算され、変形画像が生成される。

図6は以上の方法で描かれた、建物の振動のようすを示したものである。ただし図では現象をわかりやすく表示するため、変形量を1000倍に拡大して描画を行っている。背景に関しては単調な空であるため、建物の周りの画像を引き伸ばす方法で建物に隠れていた背景画像の生成を行っている。本シミュレーションでは、 512×512 pixelsの画像を用い、55~57 Hzの画像更新レートでリアルタイムの可視化を実行することができた。

3.3 変形解析の可視化 次の応用例としては、列車が鉄橋を通り過ぎる際に橋がどのように変形するかという構造物の変形解析問題を取上げた。すなわち、荷重条件の変化に従って鉄橋の変形をリアルタイムに計算し、実画像を用いて表現するという課題である。

図7は、ここで用いた画像と解析モデル、形状モデルを示したものである。解析モデルは二次元トラス構造でモデル化を行い、有限要素法により変形解析を実行した。有限要素モデルの節点数は34である。また形状モデルは鉄橋の外形を形取り、各頂点は実画像上の座標点と対応付けされている。

図8は、以上の方法で合成された鉄橋の変形映像を示したものである。図8では変形量を1000倍に拡大して表示している。この例では、列車に対しても別の画像から形状モデルを定義し、移動に応じて鉄橋画像との合成を行っている。また、鉄橋の変形時に現れる背景の木々に関しては、別の領域から切り取り、切り

ばり合成を行っている。鉄橋の手前にある木々は、いったん前景として切り出し、変形画像の生成後に切りばり合成を行っているため、鉄橋や列車の動きに影響を受けずに前後関係を保っている。本シミュレーションの画像更新レートは、 512×512 pixelsの画像に対し57~58 Hzであった。

4. 生成画像の誤差評価

実画像を利用した可視化手法では、制御点の移動に関しては解析モデルに従って正しく計算されるが、それ以外の点は二次元画像の変形によって合成されるため、画像中に多少の誤差が含まれる。サイエンティフィック・ビジュアライゼーションという目的で可視化を行う場合には、映像に含まれる誤差によって現象に対する誤った理解を与えてしまう恐れがあるため、誤差の問題に関しては十分な検討を行っておく必要がある。本章では、以下の二つの点から生成画像に含まれる誤差の評価を行い、科学技術計算における数値データを扱う方法として適当であるかどうかの検討を行った。

4.1 対象の移動に伴う誤差 まず対象の移動に伴い、制御点で囲まれた内部の画像領域に含まれる誤

Fig.7 Analysis model and shape model of the bridge

Fig.8 Visualization of deformation analysis using photograph image

Fig. 9 Warping by translation and rotation of the object

差について考える。図9に示すようにA, Bで囲まれた面が平行移動する場合には、内部領域もそのまま平行移動するため、投影画像に誤差は含まれない。しかしながら面が回転移動する場合には、画像上の各点の投影面に対する位置関係が変化するため、内部領域には誤差が含まれることになる。

今、図9に示すようにA, Bで囲まれた面がABの中点を中心に角度 θ だけ回転したとする。A, Bの間隔を2m、視点から投影面、対象物までの距離をそれぞれ l_1 , l_2 とすると、制御点A, Bに対する画像上の位置 u_A , u_B はそれぞれ、

$$u_A = -\frac{l_1 m}{l_2} \quad (14)$$

$$u_B = \frac{l_1 m}{l_2} \quad (15)$$

から、

$$u_{A'} = -\frac{l_1 m \cos \theta}{l_2 + m \sin \theta} \quad (16)$$

$$u_{B'} = \frac{l_1 m \cos \theta}{l_2 - m \sin \theta} \quad (17)$$

へ移動する。このとき三次元的な移動を考慮すると、領域内の点Rに対する画像上の位置 u_R は、

$$u_R = \frac{l_1 r}{l_2} \quad (18)$$

から

$$u_{R'} = \frac{l_1 r \cos \theta}{l_2 - r \sin \theta} \quad (19)$$

に移動するはずである。

ところが二次元画像の変形で画像の合成を行うと、制御点内部の画像は均等に伸縮されるため、

$$u_R'' = \frac{l_1 \cos \theta (m^2 \sin \theta + l_2 r)}{(l_2 - m \sin \theta)(l_2 + m \sin \theta)} \quad (20)$$

へ移動される。そのため点Rに対する画像上の位置は

$$e = u_R'' - u_{R'} \quad (21)$$

だけずれ、誤差となる。

図10は、式(19)～(21)に従い、面の回転角 θ と画

Fig. 10 Image distortion by the rotation

Fig. 11 Image distortion by the rotation for the half model

Fig. 12 Warping by deformation of the object

像位置 r/m に対する誤差をグラフに表したものである。各パラメータの値は、 $l_1=10$, $l_2=10\,000$, $m=1\,000$ とし、 $-1.0 \leq m \leq 1.0$, $0 \leq \theta \leq 80$ に対する誤差 e の大きさを示してある。このグラフから、誤差は制御点間の中央付近で最も大きく、回転角 θ に関しては、 50° 付近で最も大きくなっていることがわかる。

この誤差の値は、式(20)より制御点の間隔 m が大きいほど大きくなる。そのため、画像に含まれる誤差を小さくするためには、制御点の数を多くし、制御点の間隔をできるだけ小さくする方法が考えられる。例えば、前述の例で制御点A, Bの中央に新たな制御点Cを追加し、制御点間隔を $1/2$ の m にすると、回転角 θ と画像位置 r/m に対する誤差の大きさは図11のようになり、最大誤差の大きさは25～28%に減少される。理想的には制御点の数を調整することで、画像に含まれる最大誤差を1ピクセル以下におさえることが

Fig. 13 Image distortion by the deformation

できればよい。ここで $m=1\,000\text{ cm}$ とすると、最大誤差の大きさは画像上で $e=0.014\text{ cm}$ となり、実用上は問題のない大きさになると考えられる。しかし制御点数が多くなると、当然画像の生成にかかる計算時間も大きくなるため、面の大きさや移動量に応じて制御点の分割を適当な数に決定する必要がある。

4.2 対象の変形に伴う誤差 次に対象の変形に伴う誤差について考える。今、図 12 に示すような A, B で囲まれた面を考え、制御点の一つ B が m' だけ移動し、面が変形したとする。面の内部領域は制御点の移動に従って均等に変形するものと仮定すると、画像上の各点と投影面との位置関係が変化するため、この場合も内部領域には誤差が含まれることになる。

制御点 A, B の間隔を $2m$ 、面の角度を θ 、視点から投影面、対象物までの距離を l_1, l_2 とする。このとき、三次元位置を考慮すると領域内の点 R に対する画像上の位置は、

$$u_R = \frac{l_1 r \cos \theta}{l_2 - r \sin \theta} \quad \dots\dots\dots (22)$$

から

$$u_R' = \frac{l_1(2mr + m'm + m'r)\cos \theta}{2ml_2 - (2mr + m'm + m'r)\sin \theta} \quad \dots\dots\dots (23)$$

に移動する。しかしながら、二次元画像の変形では、

$$\begin{aligned} x_R'' = & \frac{-l_1 m \cos \theta}{l_2 + m \sin \theta} \\ & + \frac{l_1 l_2 (2m + m') \cos \theta}{(l_2 - (m + m') \sin \theta)(l_2 + m \sin \theta)} \\ & \times \frac{(l_2 - m \sin \theta)(r + m)}{(l_2 - r \sin \theta)2m} \quad \dots\dots\dots (24) \end{aligned}$$

となり、画像に含まれる誤差は

$$e = u_R'' - u_R' \quad \dots\dots\dots (25)$$

となる。

図 13 は式 (23)～(25) に従い、角度 θ と位置 r/m に対する誤差 e の値を図示したものである。各パラメータ値は、 $l_1=10$ 、 $l_2=10\,000$ 、 $m=1\,000$ 、 $m'=1\,000$ である。このグラフから、制御点の中央やや変形側の

Fig. 14 Image distortion by the deformation for the half model

点で、角度 50° 付近のときに誤差が最も大きいことがわかる。また、この場合も制御点の間隔を m にすると誤差の値は図 14 に示すように 26～32% に小さくなり、制御点数の調整によって実用的な精度で画像を生成できることが示された。

5. 結 論

本論文では、実画像を利用した科学技術計算の可視化方法を提案し、適用例の提示、および誤差評価を行った。実画像を用いる可視化では、レンダリング用の詳細なモデリングを必要とせず、リアリティが高く、背景にとけこんだ自然な映像を生成できる点が特徴である。

数値計算における解析モデルを実画像変形に用いる形状モデル、および画像上の制御点と結び付けることによって、数値シミュレーションに従った変形画像をリアルタイムで生成することができた。適用例としては、地震による建物の応答解析、荷重変化による鉄橋の変形解析の問題を取上げ、具体的な可視化方法を示した。また、本手法を用いた生成画像に含まれる誤差について、対象物の移動、変形に伴う誤差の評価を行った。この結果、実画像上の制御点の取り方を調整することによってある程度誤差を減少し、科学技術データの可視化として実用的な精度の画像を生成できることがわかった。

以上の結果から、本手法は実写性の高い映像を提示することで、単に計算結果を可視化するだけではなく、物理現象そのものを理解することが可能な可視化方法として、有効であると考えられる。

文 献

- (1) Bryson, S., Virtual Environments in Scientific Visualization, *Proc. VRST94 Conf.*, (1994), 201.
- (2) Bryson, S. and Levit, C., The Virtual Wind Tunnel, *IEEE Computer Graphics Appl.*, (1992), 25.
- (3) Cruz Neira, C., Immersed in Science and Engineering :

- Projection Technology for High-Performance Virtual Reality Environments, *6th Int. Conf. Artificial Reality Tele-Existence* (1996), 77.
- (4) 小木哲朗・渡辺浩志・廣瀬通孝, 没入型仮想環境を利用したインタラクティブ数値シミュレーション, 計算工学講演会論文集, Vol. 2 (1997), 555.
- (5) Brooks, F. P., Ouh-Young, M., Batter, J. J. and Kilpatrick, P. J., Project GROPE-Haptic Display for Scientific Visualization, *Computer Graphics*, **24-4** (1990), 177.
- (6) 矢野博明・廣瀬通孝・小木哲朗・田村善昭, 振動子を用いた科学技術データの可触化, 第13回ヒューマン・インタフェース・シンポジウム論文集 (1997), 17.
- (7) 小木哲朗・廣瀬通孝, 科学技術データ提示における多感覚の統合効果, 機論, **61**-584, C(1995), 322.
- (8) McMillan, L. and Bishop, G., Plenoptic Modeling: An Image-based Rendering System, *SIGGRAPH'95 Conf. Proc.* (1995), 39.
- (9) 廣瀬通孝・宮田亮介, 二次元実写画像を用いた三次元仮想世界の構築, 日本バーチャルリアリティ学会大会論文集, Vol. 1 (1996), 31.
- (10) 苗村健・金子正秀・原島博, 実空間光線情報に基づくリアルな仮想空間の生成, 日本バーチャルリアリティ学会大会論文集, Vol. 1 (1996), 35.
- (11) 小木哲朗・廣瀬通孝, 実画像を利用した科学技術シミュレーションの可視化, 日本バーチャルリアリティ学会大会論文集, Vol. 2 (1997), 51.
- (12) Faugeras, O., *Three-Dimensional Computer Vision*, (1996), 169, The MIT Press.
- (13) Anderson, S., Morphing Magic, *SAMs* (1993), 105.
- (14) Neider, J., Davis, T. and Woo, M., *OpenGL Programming Guide*, (1993), 475, Addison-Wesley.