社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－CVIM－153（11）
2006／3／16

# 形状復元による実時間自由視点画像生成 ー高精度化にむけてー

上田　恵† 鍋嶋　累†
有田　大作‡ 谷口　倫一郎‡

†九州大学大学院システム情報科学府
‡九州大学大学院システム情報科学研究院

本稿では，3 次元形状復元による自然な自由視点画像生成を PC クラスタを利用してオンラインで行う手法について述べる．提案するシステムでは，まず，対象形状を復元し，復元した形状を三角パッチ表現形式に変換する．次に，復元した対象形状表面の色情報を獲得し，最後に仮想視点位置に応じて自由視点画像を生成する．また，形状獲得の際には，形状の平滑化，対象形状に応じたボクセル空間の変形によりオンラインでの生成画像の高精度化を図る．実験により，提案手法によって実時間でより高精度な自由視点画像の生成が可能となることを確認した．

# Real-Time Free-viewpoint Video Generation Based on 3D Shape Reconstruction: Toward High-Fidelity Video Generation

Megumu Ueda†, Rui Nabeshima†, Daisaku Arita‡
and Rin-ichiro Taniguchi‡

†Department of Intelligent Systems, Kyushu University
‡Department of Intelligent Systems, Kyushu University

In this paper, we present a system generating a free-viewpoint image effectively in real-time using multiple cameras and a PC-cluster. Our system firstly reconstructs a shape model of objects by the visual cone intersection method, secondly transforms the shape model represented in terms of a voxel form into a triangular patch form and smooths it, thirdly colors vertexes of triangular patches, and finally displays the shape-color model from the virtual viewpoint directed by a user. The key issue is 3D shape refinement to represent 3D object shapes more accurately without increase of processing time and memory space, which is effective to generate images with higher reality. Here, we describe implementation details of our system and show some experimental results.

## 1. はじめに

現在ではテレビ放送により，実世界の様子を離れた場所で，そして実時間で見ることが可能である．しかし，テレビ放送ではカメラが撮影した場所からの視点でしか見ることができず，さらにその中からテレビ局が選択した映像のみしか視聴者は見ることができない．そこで，近年では次世代の映像メディアとして視聴者が任意の視点から映像を見ることができる自由視点画像の研究が盛んに行われている．しかし，自由視点画像の生成には，対象の三次元形状や色情報などが必要であり，非常に高い計算コストを必要とする．本研究では，実時間で自由視点画像を生成し，多数のユーザに対して生成された自由視点画像を配信するシステムの構築を目的としており，本稿では，実時間で自由視点画像を生成する手法を提案する．

金出らが “Virtualized Reality” のコンセプトを提案して以来[1)]，様々な自由視点画像に関する研究が行なわれてきた．自由視点画像生成のアプローチは大きく二つに分類することができる．一つは三次元形状を復元するアプローチであり，もう一つは三次元形状を復元しないアプローチである．本研究では，実時間性や映像の配信を考慮し，一つ目の三次元形状を復元するアプローチを選択した．

一つ目のアプローチでは，多視点のカメラ画像から対象の三次元形状を復元し，復元した対象形状の色情報を獲得し，コンピュータグラフィックスの技術を用いて自由視点画像を生成する．三次元形状を実時間で復元する手法は提案されているが，実時間で正確な色を付け，映像を生成することは難しかった[2)3)]．また，三次元情報の復元精度を上げることにより，生成される自由視点画像を高精度にする手法も提案されているが[4)5)]，その処理には時間がかかるため実時間処理は難しかった．

本研究では，高速な色情報獲得手法を提案し，PC クラスタを用いることにより，実時間で三次元形状復元から正確な色情報取得，自由視点画像生成まで可能なシステムを提案した[6)]．しかし，高速な自由視点画像生成が可能ではあるが，生成される映像の精度は高くはなかった．そこで，本稿では実時間での高精度化手法である，形状の平滑化と変形ボクセル空間を提案する．また，実験を行い，提案手法によって実時間でより高精度な自由視点画像の生成が可能となることを確認した．

## 2. 自由視点画像生成システムの概要

実時間での自由視点画像生成には上述のように多くの処理を必要とする．そこで分散並列計算機の一種である PC クラスタを用いて，オンラインで並列画像処理を行う．ここで採用した並列処理の基本的な手法は，三次元空間の分割処理ではなく，カメラ毎の並列処理である．すなわち，前段では各ノードはそれぞれ別のカメラについての処理を行ない，後段で各カメラから得られた情報の統合を行う．これは，空間を分割して並列処理を行なうと各 PC での処理量に偏りが生じやすいためである．カメラ間の並列処理であれば，三次元形状復元の処理量は同一であり，また，カメラから見える対象の表面積は基本的にはあまり大きな差がないので，色情報取得の処理量のばらつきも少ないという利点がある．

処理の概要は以下のとおりである．

(1) カメラ画像の取得
(2) カメラ画像からの対象物体抽出
(3) 各カメラ画像から抽出した対象物体領域を用いた三次元形状の復元
(4) 三次元形状の高精度化
(5) カメラ画像から三次元形状の色情報を取得
(6) 各カメラ画像から取得した色情報の統合
(7) 自由視点画像の表示

これらの処理を PC クラスタを用いて並列に行なう

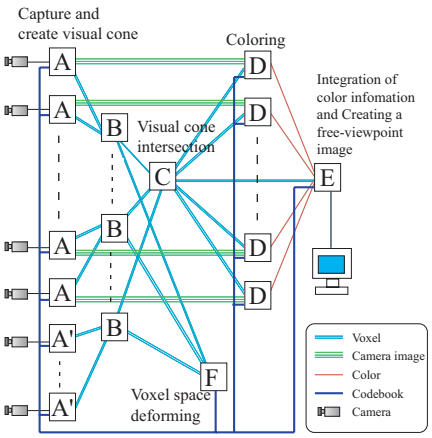



**図 1** PC クラスタの構成

(図 1)．以下に各ノード PC の処理を述べる．

### ノード A

まず，ノード **A** はカメラ画像を取得し，背景差分によって対象物体のシルエットを抽出する．次に，抽出した対象シルエットから視体積を構築する．視体積とは，視点を頂点，シルエットを断面とする錐体のことであり，対象物体は必ずこの内部に存在する．最後にノード **A** は視体積をノード **B** に，色付きシルエット画像をノード **D** に送る．いくつかのノード **A** はノード **A'** として選択することができる．ノード **A'** は視体積のみを送信し，色付きシルエット画像は送信しない．色付けに多くのカメラを使いすぎてもその寄与は少ないと思われるので，処理時間やネットワークの負荷と精度とのバランスを考慮して，カメラ配置から事前にノード A とノード A' の割り振りを決めておく．

### ノード B，ノード C

ノード **A** から送られてきた視体積に対して，ノード **B** とノード **C** の二段階で視体積交差法を行なう[7)]．視体積交差法とは，各視体積の共通部分を求めることにより対象の形状を復元する手法である．つまり，最終的な対象形状はノード **C** で復元される．次に，ノード **C** は復元した形状をボクセル表現形式から三角パッチ表現形式へ変換する．この変換には離散マーチング・キューブ法を用いる[8)]．最後にノード **C** は復元した対象形状データを送信するが，三角パッチ表現形式は膨大なデータ量となってしまうそこで，各ノードの処理時間やネットワーク負荷のバランスを考慮し，三角パッチ表現形式のまま送信はせずに，三角パッチへ変換されるボクセルの座標と対応する離散マーチング・キューブ法のパタンのみを送信する．

**ノード D**

まず，ノード **D** はノード **C** から送られてきた情報から三角パッチ表現形式の対象形状を再構築し，復元した形状表面に平滑化を施す．平滑化の詳細は 3 章で述べる．次に，ノード **A** から送られてきた色付きシルエット画像を用いて対象形状の色情報を求める．色情報取得の詳細は 4.1 節で述べる．最後に，取得した色情報をノード **E** に送る．

**ノード E**

まず，ノード **E** はユーザから仮想視点位置の入力を取得する．次に，ノード **C** から送られてきた情報から三角パッチ表現形式の対象形状を再構築し，復元した形状表面に平滑化を施す．平滑化はノード **D** と同じ処理である．これは，三角パッチ表現形式で対象形状データの送受信を行わないので，受信側（ノード **D**，**E**）のそれぞれのノードで同じ平滑化の処理を行う必要があるからである．次に，ノード **E** はノード **D** から送られてきた色情報を，仮想視点位置に応じて統合する．色情報統合の詳細は 4.2 節で述べる．最後に，自由視点画像を生成する．

**ノード F**

まず，ノード **F** はノード **B** から送られてきた視体積に対して視体積交差法を施し，対象形状を復元する．次に，ノード **F** は復元した形状を基により高精度な形状復元のためにボクセル空間を変形させる．この詳細は 5 章で述べる．最後に，変形したボクセル空間の情報をノード **A**，**A'**，**D**，**E** に送信する．

## 3. 形状表面の平滑化

視体積交差法で復元した形状に対して離散マーチング・キューブ法で形状表面を生成するだけでは滑らかな形状表面は獲得できない．そのために，誤った色付けが行われたり，生成される自由視点画像が不自然になってしまったりする可能性がある．よって，より高精度な自由視点画像生成のために形状表面の平滑化を行う．

### 3.1 ラプラシアンスムージングによる平滑化

本研究では，実時間で平滑化を行う必要があるため，高速な手法であるラプラシアンスムージングによって平滑化を行う．ラプラシアンスムージングでは，平滑化後の頂点の位置は隣接する頂点の平均で求められ，

$$v_i^{new} = v_i^{old} + \sum_{j=0}^{n}(\frac{v_j^{old} - v_i^{old}}{n}) \qquad (1)$$

Original　One iteration　Two iterations

**図 2**　形状表面の平滑化

という式で表される．ここで，$v_i$ は平滑化される対象の頂点の三次元位置，$v_j$ は $v_i$ と三角パッチを共有する頂点の三次元位置であり，$n$ は $v_j$ の数である．平滑化を繰り返し適用するほど，形状はより滑らかになるが，特徴となる形状を失う可能性もある．本研究では，実時間で行う必要があり，処理時間との兼ね合いから，1 回のみ平滑化を適用する (図 2)．

### 3.2 ルックアップテーブルの更新

本システムでは，形状復元や色付けを高速に行うために，三次元空間上の点とそれを二次元カメラ画像上に投影した点の対応関係を事前に求めてルックアップテーブルに保持している．しかし，平滑化を行うと頂点位置が任意になってしまうので，事前にルックアップテーブルを作成することができなくなってしまい，二次元カメラ画像上の投影位置を求める計算をオンラインで行わねばならず，形状復元や色付けに時間がかかってしまう．

そこで本研究では，これまでと同様に作成された平滑化前のルックアップテーブルを用いて平滑化後の頂点の投影位置を高速に求め，ルックアップテーブルを更新する．この計算は

$$t_i^{new} = t_i^{old} + \sum_{j=0}^{n}(\frac{t_j^{old} - t_i^{old}}{n}) \qquad (2)$$

という式により求めることができる．これは式 1 の $v$ が $t$ に変わるだけのものであり，$t$ はルックアップテーブルに保存されている二次元カメラ画像上の座標値である．この計算は線形計算であるため，二次以上のレンズ歪みが発生している場合には正確な座標値を求めることはできない．しかし，この計算では十分に小さな領域を対象としており，その領域内では局所的に歪みも線形であると考えることができるので，十分に正確な値を求めることができると考えられる．これにより，平滑化後もルックアップテーブルを利用することが可能となり，投影座標を高速に求めることが可能となる．

## 4. 色 付 け

本章では，ノード D における復元した形状表面の

色情報の取得，およびノード E における色情報の統合について述べる[6]．

### 4.1 単一カメラ画像からの色情報の取得

色情報は対象形状表面の各三角パッチ頂点について求め，描画の際は頂点間を滑らかに補間した色を三角パッチの面に塗る．ノード **D** では各三角パッチ頂点について色情報を取得する．色情報は，各三角パッチ頂点をカメラ画像に投影し，対応する画素の値を頂点の色情報として取得する．この際，カメラから可視な頂点の色情報のみを取得せねばならない．通常は各頂点について他の三角パッチによって遮蔽されていないかを判定する必要があるので膨大な計算を必要とするが，本研究では不可視な面の二つの特徴，

**特徴 1:** カメラの方向を向いていない三角パッチは不可視である．

**特徴 2:** 上述のような三角パッチによって遮断される三角パッチの頂点も不可視である．

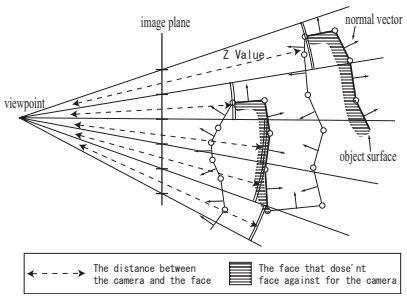
という二つの特徴を用いて高速に可視判定を行う．

可視判定手法は 2 段階の処理で構成される（図 3）．まず，ノード **D** はカメラの方向を向いていない面を求める（特徴１）．その際，それらの面の中で各画素に対して最も近い面の距離値を保存しておく．次に，カメラの方向を向いている面のうち，前の処理で保存しておいた距離値よりも遠い面を求める（特徴２）．これらの処理の後に残った面は，特徴１にも特徴２にも当てはまらない面であるので可視と判定する．この方法により高速に可視判定を行うことができ，実時間で正確な色情報を取得できる．

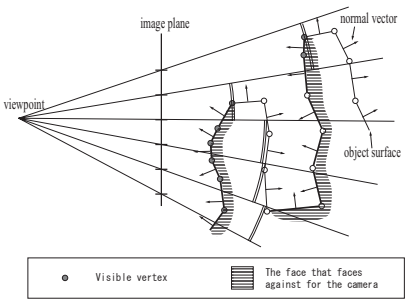
また，色付けをより高精度にするため，1 個の三角パッチを 12 個に分割する（図 4）．ボクセル空間解像度を上げることでも同様の効果は得られるが，処理時間が劇的に増加してしまう．しかし，このようにすることで処理時間を劇的に増加させることなく，より高精細な自由視点画像を生成することが可能となる．分割数はカメラ解像度や形状復元の精度に応じて経験的に決めている．さらに，画像からの色情報の取得を画素値の重み付き平均を取ることによりサブピクセルレベルで行い，生成される自由視点画像の高精度化を図っている．

### 4.2 色情報の統合

ノード **E** では取得した各カメラからの色情報を重み付き平均を用いて統合する．各カメラの重みは，仮想視点位置，実カメラの位置，対象形状に応じて動的に決定される（図 5）．あるカメラ $n$ の重み $W_n$ は




(a) ステップ 1




(b) ステップ 2

**図 3** 可視判定

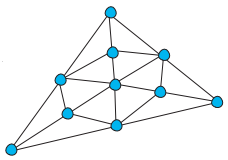


**図 4** 三角パッチの分割

$$W_n = \frac{(cos\theta_n cos\phi_n + 1)^\alpha}{\sum_{k=0}^{N}(cos\theta_k cos\phi_k + 1)^\alpha} \qquad (3)$$

として計算される．ここで，$N$ は対象の頂点を見ることができるカメラの数，$\theta_n$ は仮想視点から頂点へのベクトルと，カメラから頂点へのベクトルとがなす角度である．$\phi_n$ は頂点からカメラへのベクトルと，面の法線ベクトルとがなす角度である．これにより，仮想視点と近いカメラの色情報の重みが大きくなる．さらに，面の法線方向も考慮されているので，面に対して正面に存在するカメラの色情報の重みが大きくなる．

## 5. 変形ボクセル空間

ボクセル空間解像度を上げれば視体積交差法の精度

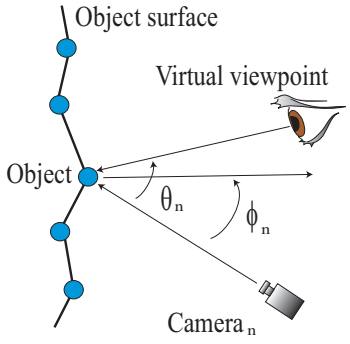



**図 5** 色情報の重みの付け方

が向上するが，$O(N^3)$ で計算コストが上昇してしまうためあまり現実的でない．そこで，計算コストをあまり上昇させずに復元精度を向上させる手法として変形ボクセル空間を提案する．

### 5.1 ボクセル空間の変形方法

ボクセル空間の変形方法は，復元した対象形状の外側でかつ，その対象形状の近傍に存在するサンプリング点を対象形状の方向に移動させるというものである（図 6）．この変形は

$$s_i^t = s_i^{ini} + \sum_{s_j \in Neighbor(s_i)} \lambda_{ij}^t (s_j^{ini} - s_i^{ini}) \quad (4)$$

という式で表せる．ここで，$s^t$ は時刻 $t$ におけるサンプリング点の座標であり，$s^{ini}$ は初期状態のサンプリング点の座標である．また，$s_i$ は非占有のサンプリング点である．$Neighbor(s_i)$ はサンプリング点 $s_i$ の 6 近傍に存在する占有のサンプリング点の集合である．$\lambda_{ij}^t$ はサンプリング点の移動可能範囲により値が決定される．つまり，$\lambda_{ij}^t$ はサンプリング点の現在位置に応じて動的に変化し，サンプリング点毎に値が異なっても良い．サンプリング点の移動可能範囲というのは

- 隣り合うサンプリング点が入れ替わってはならない
- 隣り合うサンプリング点同士が離れすぎてはならない

という制約により決められる．後述する実験では，

$$\lambda_{ij}^t = \begin{cases} \lambda_{ij}^{t-1} + a & \lambda_{ij}^{t-1} + \lambda_{ji}^{t-1} + a < 1 \\ \lambda_{ij}^{t-1} & \text{その他} \end{cases} \quad (5)$$

という関数で定義している．$a$ は 1 未満の正数である．ただし，$Neighbor(s_i)$ が空の場合は全ての $\lambda_{ij}^t$ は 0 になる．また，$s_j$ と，$s_i$ をはさんで $s_j$ の反対側のサンプリング点 $s_{j'}$ が共に占有の場合は，$\lambda_{ij}^t$ と $\lambda_{ij'}^t$ は共に 0 となる．それ以外の場合では，サンプリング点は各軸方向に 1 フレームごとにサンプリング間隔の $a$ 倍だけ移動し，サンプリング点が占有になるか隣のサンプリング点と入れ替わってしまう直前になると停止する．こうすることで，上記の制約を満たすボクセル空間の変形を容易に実現できる．

### 5.2 コードブックの送信

ノード **F** は変形ボクセル空間の変形情報 (コードブック) を多くのノードに送信しなければならないため，コードブックのデータサイズはできるだけ小さいほうが望ましい．そこで，コードブックとして移動するサンプリング点とその方向のみを送信することによって，データサイズを小さくしている．

変形ボクセル空間が適用されると，空間の解像度が部分的に高まり，形状復元の精度が向上する．しかし，ボクセル空間の変形情報は，ノード **F** から各ノードにコードブックが届いたあとに反映されるため，本システムではパイプライン並列処理を行っていることから，現在のフレームのコードブックをボクセル空間の変形に利用することは困難である．そのため，1 フレーム前のコードブックを利用してボクセル空間を変形させている．したがって，対象の変形や移動の速度が大きい場合は形状復元の精度は向上しない．このような遅延の生じない高速な変形ボクセル空間の実現は今後の課題としたい．

### 5.3 ルックアップテーブルの更新

ボクセル空間を変形させるとサンプリング点が移動してしまうため，事前にルックアップテーブルを作成することができなくなる．そこで式 4 に示すように，変形後のサンプリング点の位置を初期状態の位置の重み付き平均で表すことにより，3.2 節における形状の平滑化と同様に，初期状態のルックアップテーブルを用いて変形後のルックアップテーブルを作成することができるようにしている．

## 6. 実験結果

本手法を用いてオンラインで自由視点画像を生成し，その生成結果や処理時間，遅延時間，通信量に対する考察を行った．

本実験では合計 21 台の PC を利用した．その内訳は，ノード **A** が 6 台，ノード **A'** が 3 台，ノード **B** が 3 台，ノード **C** が 1 台，ノード **D** が 6 台，ノード **E** が 1 台，ノード **F** が 1 台である．実験で使用した PC の性能を表 1 に示す．各 PC はスイッチ型ギガビット LAN の一つである Myrinet によって相互に結合されており，1Gb/sec の通信が可能である．PC クラスタ上での並列画像処理のために，我々の研究室で開発した実時間並列画像処理環境 RPV[9] を利用して

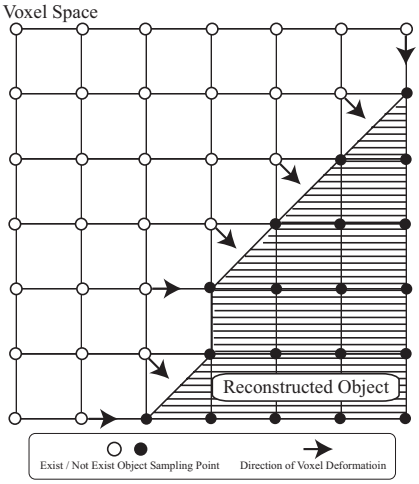



**図 6**　変形ボクセル空間の概要

**表 1**　PC の性能

| | |
|---|---|
| OS | Red Hat Linux9 |
| CPU | Intel Pentium4 3GHz |
| メモリ | 1GB |
| コンパイラ | gcc-3.2.2 |

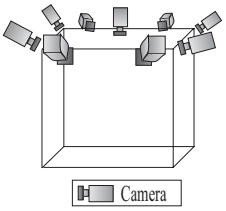


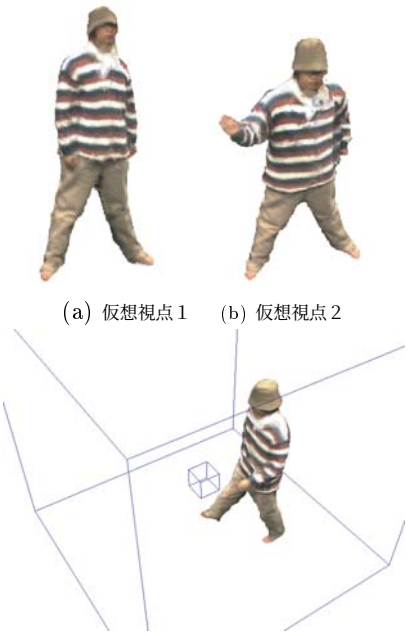

**図 7**　本実験のカメラ配置

いる．さらに 9 台の IEEE1394 ディジタルカメラ[10] が接続されており，全てのカメラは同期信号発生装置により同期がとられている．カメラは図 7 のように配置されている．また，カメラは予めキャリブレーションされている．カメラキャリブレーション手法としては，レンズ歪みを考慮した Tsai の手法[11] を利用した．カメラ画像の解像度は 640 × 480 で，空間解像度は 128 × 128 × 128，ボクセルの一辺を 2cm として実験を行った．また変形ボクセル空間における a の値は 0.25 とした．つまり，各サンプリング点は各軸方向に 5mm ずつ 3 回まで移動することができることになる．

**生成画像に関する考察**

図 8 に入力したカメラ画像，図 9 に生成された自由視点画像，図 10 に生成された自由視点画像列を示す．いずれも原画像と同程度の画質が得られていることがわかる．図 11 に平滑化の効果を示す．図 11 から，平滑化の適用により，生成される画像がより自然になっていることがわかる．図 12 に変形ボクセル空間による効果を示す．図 12 から，変形ボクセル空間を使用することにより，人差し指のような細い形状がより正確に復元されていることがわかる．

**図 8**　入力したカメラ画像



(a) 仮想視点 1　(b) 仮想視点 2

(c) 仮想視点 3

**図 9**　生成された自由視点画像

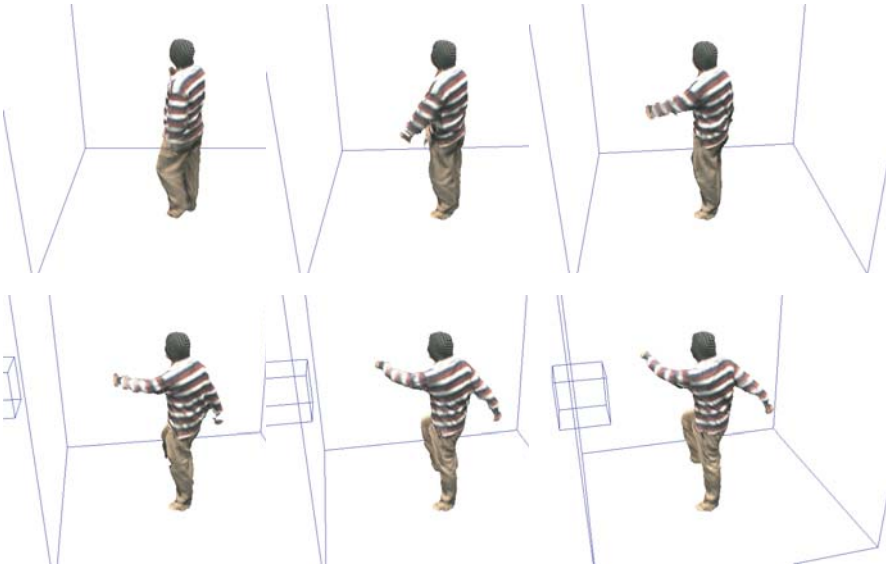


**図 10** 生成された自由視点画像列

**表 2** 各ノードの処理時間と送信量

| Node | Time (msec) | Data (KByte) |
|---|---|---|
| A | 25 | 350 |
| B | 0.5 | 256 |
| C | 12 | 35.4 |
| D | 48 | 205 |
| E | 44 | None |
| F | 65 | 10.8 |

**遅延時間に関する考察**

レイテンシは 350 ミリ秒程度であった．主観的にはまだ遅延を感じるので，本システムを人同士のインタラクションなどに使う場合には，さらなる改善が必要であると考えられる．

**処理時間に関する考察**

システムの平均スループットは 10fps であった．表 2 に各ノードの処理時間と送信量を示す．表 2 から，ノード **F** が他のノードよりも処理時間が大きいことがわかる．ノード **F** はノード **A** にとってはフィードバックとなるため，ノード **F** の処理に時間がかかるとシステムのパフォーマンスに大きく影響を及ぼす．実際に，表 2 の処理時間から計算すると，全てのノードは 15fps 以上で動作しているにもかかわらず，システムの平均スループットは 10fps 程度であり，また，ボクセル空間の変形を毎フレームではなく 2 フレーム毎に行った場合は，スループットは 15fps 程度に上昇した．このことから，ノード **F** の処理速度がシステムに大きく影響を与えていることがわかる．

**送受信データ量に関する考察**

表 2 と図 1 から，ノード **E** が最も多くのデータを受信し，毎フレーム 1276Kbyte のデータを受信していることがわかる．これは前述した Myrinet の性能から計算すると送受信に約 10 ミリ秒時間がかかるが，システムのスループットには影響を与えない．しかし，今後の応用として自由視点画像の配信を考えると，このデータを配信することになるので，ネットワークの負荷を減らすためにデータ圧縮する必要があると思われる．

## 7. おわりに

本稿では，高精度な自由視点画像を実時間で生成する手法を提案した．実験の結果，実時間で高精度な画像を生成できることが確かめられた．今後の課題としては以下のようなものが考えられる．

- **変形ボクセル空間の改良:**
  現在のシステムでは変形ボクセル空間の処理時間が最も大きいため，高速化が必要である．また，形状復元精度を上げるには同一フレーム内で適用可能な変形ボクセル空間が必要である．さらに，移動するサンプリング点の決定法や移動量などを改良することにより精度が向上すると期待される．
- **処理速度の安定化:**
  現在のシステムでは処理速度が安定していないので安定させなければならない．これは形状復元を多重解像度で行うことにより安定させることができる[12]．

(a) 通常の色付け前の復元結果

(b) 平滑化した色付け前の復元結果

(c) 通常の色付け後の復元結果

(d) 平滑化した色付け後の復元結果

**図 11** 平滑化の効果

(a) 通常の結果　(b) 変形ボクセル空間を使用した結果

**図 12** 変形ボクセル空間の効果

- **ノード A' の動的選択：**
  色付けには使用しないノード **A'** を動的に選択する．現在はノード **A'** は事前に決定している．仮想視点位置に応じて動的に選択する方法も考えられるが，自由視点映像の配信を考えた場合，仮想視点位置が複数存在することも考えられ，その方法は難しい．仮想視点の位置に関わらずもっとも色付けの精度が上がるように動的に選択することができれば，より高精度に色付けができると考えられる．

## 参 考 文 献


1) T. Kanade and P. W. Rander and P. J. Narayanan: "Concepts and early results", IEEE Workshop on the Representation of Visual Scenes, pp. 69–76, June, 1995. Θ
2) M. Gross and S. Wurmlin and M. Naef and E. Lamboray and C.Spagno and A.Kunz and E. Koller-Meier and T. Svoboda Luc Van Gool and S. Lang and K. Strehlke and A. Vande Moere and O, Staad: "blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence", Proc. of ACM SIGGRAPH2003, pp. 819–827, 2003.
3) O. Grau and T. Pullen and G. A. Thomas: "A Combined Studio Production System for 3D Capturing of Live Action and Immersive Actor Feedback", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 3, pp. 370–380, March, 2004.
4) 高井 勇志，松山 隆司："3 次元ビデオ映像の高精細表示アルゴリズムと編集システム"，映像情報メディア学会誌，Vol. 56, NO. 4, pp. 593–602, 2002.
5) Joel Carranza and Christian Theobalt and Marcus A. Magnor and Hans-Peter Seidel: "Free-viewpoint video of human actors", ACM Transaction on Graphics, Vol. 22, No. 3, pp. 569–577, July, 2003.
6) 上田 恵，有田 大作，谷口 倫一郎："多視点動画像処理による 3 次元モデル復元に基づく自由視点画像生成のオンライン化ー PC クラスタを用いた実現法ー"，情報処理学会論文誌，pp. 2768–2778，Vol. 46，No. 11，2005.
7) W. N. Martin and J. K. Aggarwal: "Volumetric description of objects from multiple views", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 5, No. 2, pp. 150–158, 1983.
8) 剣持 雪子，小谷 一孔，井宮 淳："点の連結性を考慮したマーチング・キューブ法"，電子情報通信学会技術研究報告，pp. 197–204, Jan, 1999.
9) 有田 大作，花田 武彦，谷口 倫一郎："分散並列計算機による実時間ビジョン"，情報処理学会論文誌，Vol. 143, No. SIG 11(CVIM5), pp. 1–10, 2002.
10) 吉本 廣雅，有田 大作，谷口 倫一郎："1394 カメラを利用した多視点動画像獲得環境"，第 6 回画像センシングシンポジウム講演論文集，pp. 285–290, 2000.
11) Roger Y. Tsai: "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses", IEEE Transaction on Robotics and Automation, Vol. 3, No. 4, pp. 323–344, 1987.
12) 鍋嶋 累，上田 恵，有田 大作，谷口 倫一郎："オンライン自由視点映像生成の可変解像度処理によるフレームレート安定化"，画像の認識・理解シンポジウム，pp. 455–462，2005.