芸術科学会論文誌 Vol. 10, No. 4, pp. 263-275

投影マッピングを利用した多視点カメラ映像からのリアルタイムな仮想視点映像生成法

竹中 史雄（非会員）* **藤本 忠博**（正会員）**
原美 オサマ（正会員）** **千葉 則茂**（正会員）**

* 岩手大学大学院工学研究科 ** 岩手大学工学部

Real-time Rendering of Virtual-viewpoint Images from Multi-view Camera Images Using Projection Mapping

Fumio TAKENAKA(Non-member) * **Tadahiro FUJIMOTO**(Member) **
Osama HALABI(Member) ** **Norishige CHIBA**(Member) **

* Graduate School of Engineering, Iwate University
** Faculty of Engineering, Iwate University

{h18j71, fujimoto, ohalabi, nchiba}@cis.iwate-u.ac.jp

アブストラクト

本研究では，複数のカメラから取得した多視点カメラ映像をもとに，投影マッピングを利用した視体積交差法により，リアルタイムで対象物体の３次元形状を復元して描画する手法を提案する．本手法では，３次元空間上で複数の平板を平行かつ等間隔に並べた平行平板群を３軸方向に組み合わせたモデルによりボクセルモデルを疑似表現することで，ボクセルベースの視体積交差法を模擬する．各カメラ映像のフレーム画像ごとに，まず，前景（対象物体）ピクセルと背景ピクセルをα値で区別した投影用テクスチャを生成する．そして，グラフィックライブラリの投影マッピングの機能を利用して，各カメラの投影用テクスチャを各カメラ視点から平行平板群に対して投影し，全カメラの投影領域の積をα値の演算により求めることで，３次元空間上での物体の形状をビジュアルハルにより色情報とともに高速に復元して描画する．本手法は，通常のボクセルベース視体積交差法に比べて非常に高速であり，動きを伴う対象物体に対してもリアルタイムで仮想視点映像を生成することが可能である．

Abstract

This paper proposes a method, which is based on a volume intersection method by utilizing projection mappings, to reconstruct and render the 3D shape of an object in real time from images captured by multiple video cameras. The proposed method imitates a voxel-based volume intersection method on a pseudo-voxel space, which is constructed by placing orthogonally three sets of parallel equally-spaced planes in the 3D space. In each frame time, first, the frame image of each camera is processed to generate a projection texture having different alpha values on foreground (object) pixels and background ones. Then, the projection texture of every camera is projected onto the parallel planes by projection mapping functions of a graphic library. Finally, the visual hull of the object is efficiently reconstructed as the intersection of all cameras' projection regions in the 3D space by alpha operations of the graphic library, and is rendered using color values. The proposed method is more efficient than a conventional voxel-based volume intersection method, and is able to generate virtual viewpoint images with a moving object in real time.

１.はじめに

対象物体を異なる複数のカメラ視点から撮影して得られた多視点映像をもとに，その物体の３次元形状を復元する技術や任意の仮想視点位置から見た仮想視点映像を生成する技術など，イメージベースＣＧの研究が盛んに行われている．撮影された多視点映像は「参照映像」や「参照画像」と呼ばれる．多視点映像から仮想視点映像を生成する代表的な方法を以下に示す.

・参照画像中の各ピクセルに映る物体上の点までのカメラ視点からのデプス値を推定して求めた３次元点を利用する方法[1-4]
・大量の参照画像からライトフィールドを構築する方法[5-7]
・参照画像中の物体の輪郭（シルエット）を利用する方法[8-18]

これらのうち，物体のシルエットを利用する方法では，ビジュアルハル[8]と呼ばれる３次元形状により実際の物体形状を近似的に表現する（２章参照）．この方法は，ステレオマッチング法[19]などで必要とされる異なるカメラの参照画像間でのピクセル単位の対応付けのための探索処理が不要であるため，高速に３次元形状を復元できるという利点があり，リアルタイム性が重視されるアプリケーションなどに適する．ビジュアルハルを扱う方法としては，これまでに，画像ベースで仮想視点映像のピクセル色を高速に計算する方法[9]，ポリゴンモデルにより形状復元する方法[10]，ボクセルモデルにより形状復元する方法[11]などが提案されている．さらに，異なる参照画像間の対応ピクセルの色の同一性（photo consistency）を用いてボクセルモデルの復元精度を高める方法[12]がある．また，視体積交差法（後述）で生成されたビジュアルハルに対して色情報を利用したステレオマッチング法で補正することで復元形状の近似精度を高める方法[13]，その補正に対して元のビジュアルハルが持つ局所的な形状特徴を維持するような拘束をかけて安定した形状を得る方法[14]が提案されている．ただ，この方法は，任意の仮想視点位置から見た映像をリアルタイムで生成できるものの，あらかじめ撮影しておいたビデオ映像を対象として前処理に多くの時間を要するものであり，撮影した映像を取り込みながらリアルタイムで処理することはできない．また，通常，ビジュアルハルを構築する前処理として，背景差分処理などにより参照画像上の背景が映る「背景領域」から対象物体が映る「前景領域」を抽出する必要がある．このとき，その抽出精度が低い場合や，対象物体の一部が他の物体に隠されてオクルージョン領域が発生する場合などには，構築されるビジュアルハルの品質は大きく低下する．これを解決するため，抽出された前景領域と背景領域の信頼度を定義し，安定して高品質なビジュアルハルを構築する方法が提案されている[15]．また，ビジュアルハルを利用して人体の動きのアニメーションを生成する手法も提案されている[16,17]．

ビジュアルハルを生成する代表的な手法の一つとして視体積交差法がある．その中でも，特に，ボクセルモデルによりビジュアルハルを表現するボクセルベース視体積交差法は，原理がシンプルであり，よく用いられる[11-15,18]．しかし，この方法では，ビジュアルハルの精度がボクセル空間の解像度に依存する一方で，その解像度が増加するにつれて高速な処理が難しくなるという問題がある．そこで，本研究では，ボクセルベース視体積交差法の原理に基づき，グラフィックライブラリ OpenGL の投影マッピングの機能を効率的に利用することで，高解像度のボクセル空間に対しても高速にビジュアルハルの仮想視点映像を生成する手法を提案する．本手法では，３次元空間上で複数の平板（ポリゴン）を平行かつ等間隔に並べた平行平板群を３軸方向に組み合わせたモデルによりボクセル空間を疑似表現する．これに類似する復元空間の表現を用いた視体積交差法の研究としては，ボクセル空間を平行平面群で表現し，参照画像上の物体のシルエットをボクセル空間に投影する幾何計算を簡易化することで，高速な処理を実現する手法が提案されている[18]．また，本手法では，復元したビジュアルハルに適切な色を与える方法として，仮想視点の位置に依存した視点依存型のカラーマッピングを行なう[10,14,20,21]．

これ以降，２章で視体積交差法の基本原理を説明し，３章で本研究で提案する手法の具体的な説明を行なう．４章では実験による結果を示し，提案手法の評価を行なう．５章で全体のまとめと今後の課題を述べる．

２．視体積交差法の基本原理

本章では，視体積交差法の基本原理について説明を行う[11]．視体積交差法は，図１に示すように，複数のカメラから物体を撮影して得られる各参照画像上の物体の２次元シルエットを利用して３次元形状を復元する．まず，カメラごとに，カメラ視点から３次元空間に向けて物体のシルエットを逆投影した視錐体を考える．そして，すべてのカメラの視錐体の共通領域（積領域）を求める．この共通領域はビジュアルハルと呼ばれ，実際の物体形状を内包する近似形状となる．ビジュアルハルの近似の精度は参照画像の個数に依存し，より多くの物体の特徴的な輪郭部分をシルエットとして含む参照画像ができるだけ多く与えられるほど，より実際の物体形状に近いものとなる．逆に，

参照画像が少ない場合には近似の精度が低く，また，原理的に，参照画像上のシルエットに現れない凹形状は復元できない．しかし，ステレオマッチング法などで必要とされる異なるカメラの参照画像間でのピクセル単位の対応付けが不要であるため，高速に３次元形状を復元できるという利点から，特にリアルタイム性が重視される場合に利用されることが多い．

視体積交差法の具体的な実現方法の一つであるボクセルベースの手法は，ボクセルモデルによりビジュアルハルを表現する．ボクセルベース視体積交差法の中でよく用いられる手法である Space Carving Method (SCM)[11]は，ボクセル空間内の各ボクセルを各参照画像に投影することで，ビジュアルハルのボクセルモデルを生成する．具体的には，すべてのカメラの参照画像上のシルエット内部に投影されたボクセルはビジュアルハル内部に位置するものとしてボクセルモデルに残し，一方，一つでもシルエット外部に外れて投影された参照画像があるボクセルはビジュアルハル外部に位置するものとしてボクセルモデルから削除する．ボクセルベース視体積交差法では，ビジュアルハルの精度がボクセル空間の解像度，すなわち，ボクセルの個数に依存し，その解像度が増加するにつれて高速な処理が難しくなってくる．

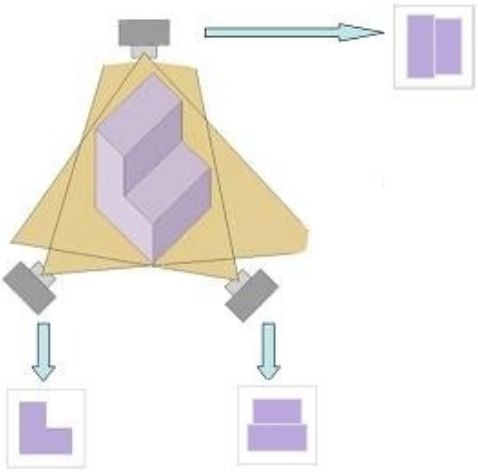


図１：視体積交差法の基本原理

カメラ映像中の物体シルエットを高速に投影することで，リアルタイムで物体形状のビジュアルハルを復元して描画する．図２に本手法の処理の流れを示す．本手法では，各カメラから一定の時刻（フレーム時刻）ごとに取得されるフレーム画像は全カメラで同期が取れているものとし，同一時刻に取得される全カメラのフレーム画像から，その時刻の仮想視点フレーム画像を生成する．まず，カメラごとに，フレーム画像に背景差分処理を適用し，ピクセルごとに，後の α 値演算に必要な α 値，ならびに，仮想視点位置に依存した視点依存型カラーマッピングに必要な RGB 値を設定することで，背景差分画像を生成する．そして，その背景差分画像を投影用テクスチャとしてテクスチャメモリ領域に割り当てる．その後，すべてのカメラの投影用テクスチャを用いて，ボクセル空間を疑似表現した平行平板モデルに対して投影マッピングを行う．そして，各投影用テクスチャの α 値による積演算を実行してビジュアルハルを求め，視点依存型カラーマッピングによる仮想視点からのレンダリングにより仮想視点フレーム画像を得る．以下，本手法の各処理について述べる．

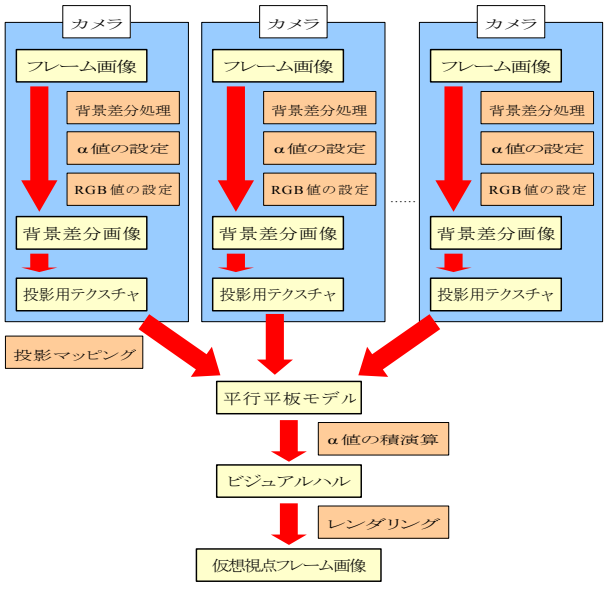


図２：本手法の処理の流れ

３．投影マッピングを用いた視体積交差法

本章では，本研究で提案する投影マッピングを利用した視体積交差法について説明する．本手法は，ボクセルベースの視体積交差法の原理を基本とし，３軸方向に組み合わせた平行平板群でボクセル空間を疑似表現し，グラフィックライブラリ OpenGL の投影マッピング[22-26]の機能を利用して

３．１　背景差分と投影用テクスチャの生成

各カメラからフレーム時刻ごとに取得されるフレーム画像に対して，前景領域と背景領域を区別するために背景差分処理を行い，背景差分画像を生成する．背景差分画像の各ピクセルには，３．４節で述べる α 値演算のため，前景ピクセルに

1.0，背景ピクセルに0.5の α 値を登録しておく．また，同時に，３．５節で述べる視点依存型カラーマッピングのため，フレーム画像のRGB値を用いて，仮想視点と各カメラ視点の位置関係に依存して決定したRGB値を与える．そして，その背景差分画像をテクスチャメモリ領域に割り当てることで，RGBα 値を持つOpenGLのテクスチャデータとして投影用テクスチャを生成する．これにより，フレーム時刻ごとに，各カメラからのフレーム画像による投影用テクスチャが用意されることになる．

３．２　平行平板モデル

ボクセルベース視体積交差法では，物体の３次元形状を復元するための空間としてボクセル空間を用いる．本論文では，ボクセル空間で定義される３次元空間上の直方体領域のことを復元空間と呼ぶ．本手法では，３．１節で述べた方法で生成した投影用テクスチャによる投影を行なうため，図３に示すように，直交x-y-z座標系であらわされる３次元空間上で複数の平板を平行に等間隔に並べた平行平板群をx，y，zの３軸方向に組み合わせたモデル，すなわち，x-y平面，y-z平面，z-x平面のそれぞれに平行に配置した３組の平行平板群によりボクセル空間を疑似表現する．以降，この平行平板群を平行平板モデルと呼ぶ．なお，各平板は，実際には，OpenGLの処理対象としてポリゴンによって定義される．

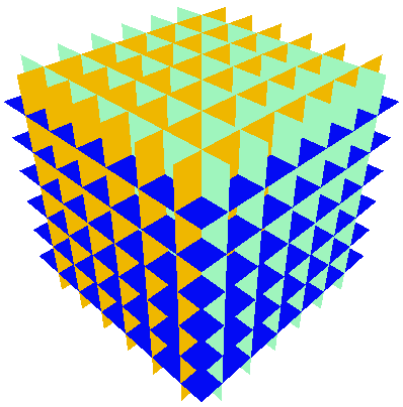


図３：復元空間上の平行平板モデル

３．３　投影マッピング

本手法では，視体積交差法におけるボクセル空間に対するカメラ視点からの視錐体の投影を模擬するため，３．２節で述べた平行平板モデルに対して３．１節で述べたカメラごとの投影用テクスチャを投影マッピング[23]する．投影マッピングとは，指定した投影中心や投影方向に従って，３次元空間上のポリゴンに対して投影用テクスチャをマッピングする処理である．これにより，あたかもプロジェクタの映像がスクリーンに投影されるかのようにテクスチャをポリゴンにマッピングすることができる．このとき，投影方向に複数のポリゴンがある場合でも，図４に示すように，投影が前方のポリゴンによって遮られることなく，投影範囲に位置するすべてのポリゴンに対して投影用テクスチャがマッピングされる．

OpenGLでは，通常，ポリゴンにテクスチャをマッピングする際には，ポリゴンの各頂点の３次元座標に対してテクスチャのテクスチャ座標を割り当てる必要がある．テクスチャ座標は，物体の座標と同じように，内部的には３次元座標（４次元同次座標）を持ち，３次元の座標変換を適用することができる．投影マッピングはこれを利用しており，テクスチャ座標に対して回転と平行移動を与えるモデルビュー行列を適用することによって投影位置や投影方向を変更し，射影行列によって透視投影を行うことができる．また，ポリゴン頂点へのテクスチャ座標の割り当てを自動的に計算する機能[24]が用意されている．

本手法では，平行平板モデル中のすべての平板に対して，すべてのカメラの投影用テクスチャを各カメラ視点から投影マッピングする．このとき，各平板に複数の投影用テクスチャを投影マッピングするために，個々の投影用テクスチャの情報をそれぞれ別のテクスチャユニット[25]に設定しておく．これにより，マルチテクスチャ[25]の機能を用いて３．４節で述べる α 値を用いた全カメラの視錐体の共通領域の算出が可能となる．

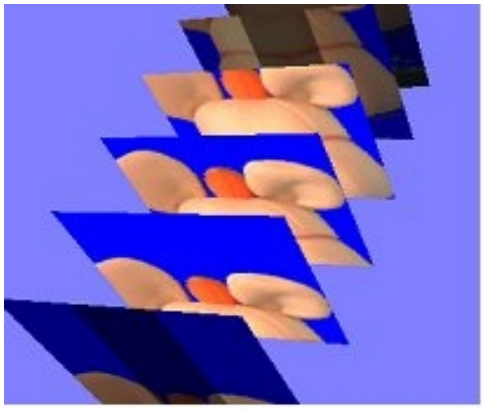


図４：投影マッピング．１組の平行平板群に対して，下方から斜め上方に向かって投影用テクスチャを投影している．

３．４　α値演算によるビジュアルハルの生成

視体積交差法では，２章で述べたように，各カメラのフレーム画像中の物体シルエットをカメラ視点から３次元空間に向けて投影した視錐体について，全カメラの視錐体の共通領域，すなわち，全視錐体の積領域を求めてビジュアルハルを得る．本手法では，３．３節で述べた投影マッピングにより平行平板モデルの各平板に投影された全カメラの投影用テクスチャが持つ α 値を用いて，積領域を求める．具体的には，マルチテクスチャ[25]の機能を用いて，３．１節で述べた各投影用テクスチャに与えた α 値により，各平板上で全カメラの投影用テクスチャに関する α 値の積演算を実行する．そして，α テスト[26]と呼ばれる処理により，すべての投影用テクスチャの前景ピクセルが投影される共通領域のみを残して表示する．これをすべての平板について行うことで，３次元空間上で全視錐体の積領域に相当する部分だけが残され，ビジュアルハルが表示される．ここで，α テストとは，ポリゴン上の各点が持つ α 値と設定した閾値との大小関係により，そのポリゴンの表示部分を決定する処理である．３．１節で述べたように，各投影用テクスチャ上で，前景ピクセル，すなわち，対象物体のシルエット内のピクセルの α 値を 1.0，背景ピクセルの α 値を 0.5 とした場合，平板上で全投影用テクスチャに関する α 値の積を求めると，全カメラの視錐体内に含まれる共通領域の積は 1.0 となり，一つでも視錐体外となるカメラがある領域の積は 0.5 以下となる．よって，α テストの閾値 α_s として $0.5 < \alpha_s < 1.0$ となる値を設定し，平板上で閾値 α_s より大きな α 値を持つ部分だけを表示することとする．これをすべての平板について行うことで，ビジュアルハルが表示されることになる．図５は，２つのカメラの投影用テクスチャを投影マッピングした例を上から見た図を示す．各カメラの投影用テクスチャの３次元空間上での投影領域のうち，緑色領域が物体シルエットの投影領域，水色領域が背景の投影領域を示す．また，赤色の四角が復元空間を示す．復元空間内の各部分の α 値の積を考えた場合，２つのカメラの物体シルエット投影領域の共通領域の α 値だけが 1.0 となり，他の領域の α 値は 0.5 以下となる．α テストを行うことで，その共通領域のみを表示することができる．

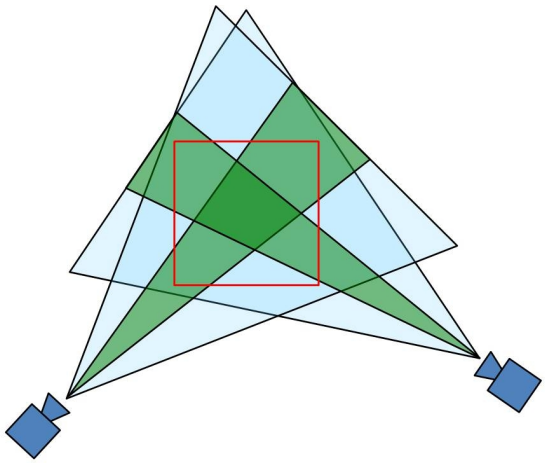


図５：投影マッピングとα値の積

３．５　視点依存型カラーマッピング

３．４節の方法でビジュアルハルを表示した場合，平板上で投影用テクスチャの α 値の積を求める際に，実際には，RGB 値についても積演算が行われる．ここで，もし，各カメラの投影用テクスチャにフレーム画像の RGB 値をそのまま与えた場合には，平板の表示される部分の表示色は，仮想視点の位置による見え方の違いを考慮することなく，すべてのカメラのフレーム画像の RGB 値の積による色となってしまう．そこで，本手法では，仮想視点の位置による視点依存型のカラーマッピングにより，仮想視点から見た適切な色をビジュアルハルに与える．

本手法のカラーマッピング処理では，物体を見る角度が仮想視点と最も近いカメラのフレーム画像が持つ RGB 値を平行平板モデルに与えて表示を行う．まず，M 台のカメラに対して，現在の仮想視点 e の視線方向の単位ベクトル Ve と各カメラ i(=1,...,M)の視線方向の単位ベクトル Vi の内積を計算し，その内積の値が最も大きなカメラ k を求める．カメラ k は，仮想視点の視線方向とのなす角度が最も小さい視線方向を持つ．そこで，カメラ k の投影用テクスチャ T_k にはカメラ k のフレーム画像 F_k の RGB 値を与える．一方，その他のカメラの投影用テクスチャの RGB 値には白色をあらわす値を与える．ここで，RGB 値として正規化した 0.0 <= R, G, B <= 1.0 を用いた場合，白色をあらわす RGB 値は［1.0, 1.0, 1.0］である．つまり，これにより，全カメラの投影用テクスチャの RGB 値の積はカメラ k のフレーム画像 F_k の RGB 値となり，結果として，平行平板モデルには仮想視点と最も近い角度で物体を見るカメラ k による色がマッピングされることになる．

４．実験

本章では，提案手法の有効性を検証するために行った実験について述べる．

４．１　実験環境

実験環境を図６に示す．４台の IEEE1394 カメラをそれぞれ三脚に固定し，延長ケーブルを介してすべてのカメラを１台のＰＣに接続した．各カメラは，対象物体をほぼ等距離から取り囲むように対象物体の手前側に設置し，それぞれ，上から見下ろすように対象物体の中心付近の一点に向けた．各カメラには，左から右に向かって順にカメラ①，②，③，④と名付けた．対象物体としては，図７と８に示すように，小型扇風機，ならびに，電動で動くカエルのぬいぐるみを用意した．なお，撮影時には，動きを伴う対象物体をリアルタイムで処理していることを確認するため，扇風機はゆっくりと首を回し，また，カエルのぬいぐるみは口，目，手などを動かす動作を行わせた．

図６：実験環境

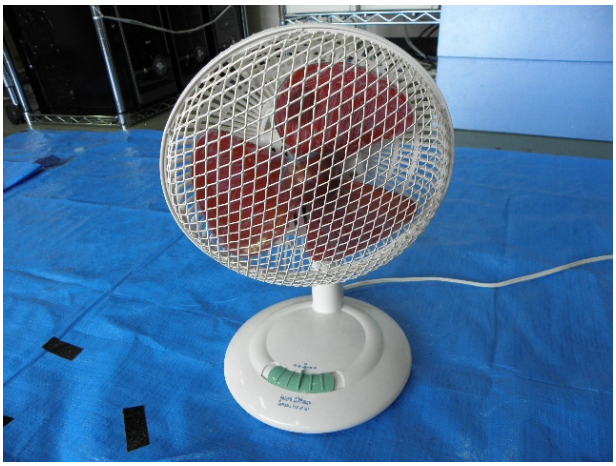


図７：対象物体１（小型扇風機）

図８：対象物体２（カエルのぬいぐるみ）

表１に実行環境を示す．フレーム画像解像度については，カメラからのフレーム画像の取得時には 640*480(ピクセル)であるが，立方体状の平行平板モデルへの投影マッピングのしやすさ，ならびに，ピクセル数を減らすことによる投影用テクスチャの生成と投影マッピングの処理の効率化を考え，カメラからの取得直後にフレーム画像の両端を削除することで正方形状の 480*480(ピクセル)にリサイズした．また，本実験で用いたＰＣの CPU は４コアを持つが，本実験で実装したプログラムでは並列化処理を全く行っていないため，１コアでの実行となっている．さらに，Cg 等のシェーダ言語による GPU に向けた実装も行っていない．

表１：実行環境

CPU	Intel(R)Core(TM)i7 2.80GHz
メモリ	4.00GB
ビデオカード	NVIDIA GeForce GTX 260
カメラ（４台）	Point Grey Research 社製 Firefly MV FFMV-03M2C-CS
フレーム画像解像度	640*480 (480*480)
描画ウィンドウ解像度	320*320

４．２ カメラ校正

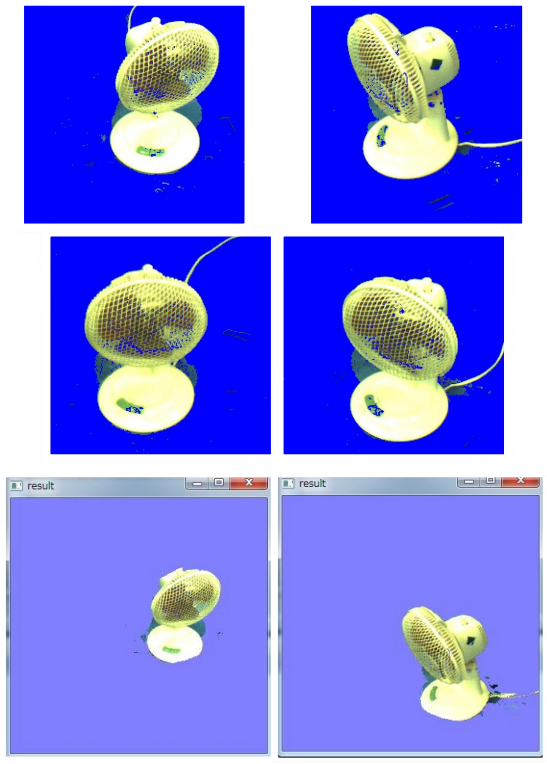
本実験では，コンピュータビジョン用ライブラリ OpenCV に実装されている Zhang によるカメラ校正法[27]のライブラリ関数を用いて，４台のカメラの外部パラメータと内部パラメータを求めた．

外部パラメータは３次元空間上でのカメラ間の相対的な位置と姿勢をあらわし，ワールド座標系における回転と平行移動の座標変換で与えられる．一方，内部パラメータは各カメラに固有のパラメータであり，本実験では，画像平面上の直交する２軸の歪みは０とし，焦点距離と主点座標値のみを求めた．Zhang による校正法では，幾何特性が既知である平面パターンを多視点から撮影し，撮影画像から抽出した特徴点をもとにパラメータを計算する．実際には，平面パターンとして白黒の格子パターンを持つチェッカーボードを用い，チェッカーボード上の特徴点の１つをワールド座標系の原点とし，地面と平行な平面を x-z 平面，鉛直方向を y 軸とした．はじめに，カメラごとにチェッカーボードを多視点から撮影して内部パラメータを求めた．続いて，対象物体を撮影する位置に各カメラを設置し，すべてのカメラから撮影できる位置にチェッカーボードを置いて撮影することで外部パラメータを求めた．

４．３　実験結果

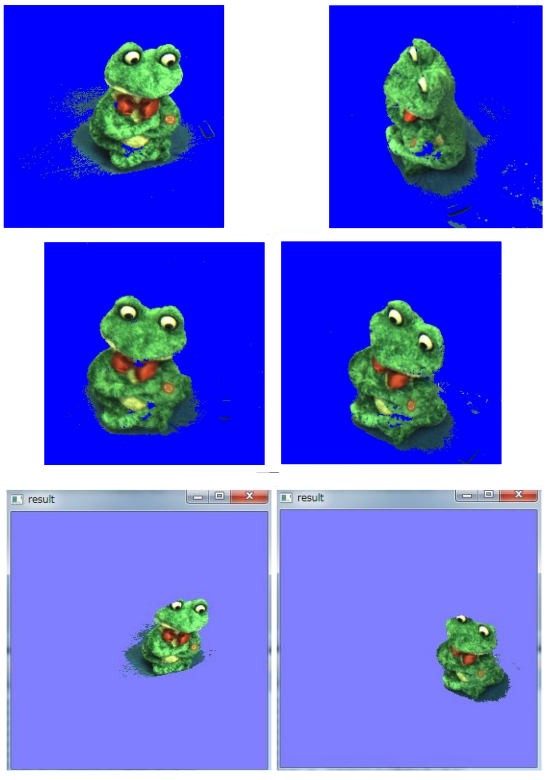
４．１節で述べた実験環境の下で，４．２節で述べた方法で４台のカメラを校正した後，提案手法による仮想視点映像の生成実験を行なった．図９と１０は，それぞれ，図７と８の対象物体に対する画像を示す．それぞれの図で，上の４つの画像は４台のカメラにより取得されたフレーム画像の例であり，左上がカメラ①，左下がカメラ②，右下がカメラ③，右上がカメラ④によるフレーム画像である．そして，下の２つの画像は，提案手法によって復元されたビジュアルハルをカメラ①～④とは異なる位置から描画した仮想視点フレーム画像の例である．図９と１０とも，(a)はカメラ①とカメラ②の間に仮想視点を置いた場合であり，(b)はカメラ③とカメラ④の間に仮想視点を置いた場合である．

図９と１０のカメラによるフレーム画像は，背景差分処理を行った結果として背景を青色で表示している．本実験の背景差分処理では，はじめに対象物体を置かずに背景のみを撮影した背景フレーム画像を取得した．そして，対象物体の撮影時には，その背景フレーム画像とフレーム時刻ごとに取得したフレーム画像の間で各ピクセルの色を比較し，フレーム画像上の前景ピクセルと背景ピクセルの判定を行った．



(a)　(b)

図９：対象物体１（図７）のカメラ撮影したフレーム画像と仮想視点フレーム画像



(a)　(b)

図１０：対象物体２（図８）のカメラ撮影したフレーム画像と仮想視点フレーム画像

補足資料の２つの動画ファイルは，それぞれ，図９と１０に対応し，リアルタイムでの仮想視点映像生成の実験の様子をビデオカメラで撮影した実験映像である．これらの実験映像中でディスプレイ上に６個のウィンドウが表示されているが，このうち，上の４つのウィンドウがカメラから取得された映像であり，一方，右下のウィンドウが生成された仮想視点映像である．マウス操作により仮想視点を３次元空間上でインタラクティブに動かすことで，リアルタイムで仮想視点映像が生成されていることがわかる．また，ディスプレイ上で仮想視点映像ウィンドウの左側のウィンドウには，４台のカメラと仮想視点の視線方向をあらわす５本の線分が表示されている．仮想視点の動きに従い，仮想視点の視線方向をあらわす黒い線分が動いている．

上記の２つの対象物体に関する仮想視点映像生成の結果から，リアルタイムの処理速度（後に詳述）で，仮想視点から見た対象物体の形状と色がある程度正しく描画されていることがわかる．しかし，生成された映像の品質はあまり高くはない．この大きな原因の一つとして，３．５節で述べた視点依存型カラーマッピングがある．この方法では，仮想視点に最も近い視線方向を持つカメラｋのフレーム画像 F_k の色のみが平行平板モデルによるビジュアルハルに与えられる．しかし，仮想視点はカメラｋの視点とは異なるため，ビジュアルハル上で仮想視点から見える面のうち，カメラｋの視点からは見えない不可視面が生じる．そして，原理上，この不可視面には，フレーム画像 F_k 上でその不可視面を境界とする物体輪郭上に位置するピクセルの色が奥行き方向に引き伸ばされてマッピングされる．例えば，図１１の場合，中央の赤いカメラで示す仮想視点に対して，両側の青い実際のカメラのうち，右側のカメラｋが最も仮想視点に近いものとして選ばれる．このとき，そのフレーム画像 F_k 上の前景ピクセルのうち，青色で示すピクセル群の色はビジュアルハルのカメラｋから可視の面にマッピングされる．一方，ピンク色の点で示すピクセルは，カメラｋから不可視の面を境界とする物体輪郭上に位置し，その色はその不可視面に奥行き方向に引き伸ばされてマッピングされる．この現象の結果として，本手法で生成される仮想視点映像では，ビジュアルハルのいずれかの輪郭側に奥行き方向に伸びる筋状のパターンが発生してしまう．これは，カメラ台数を増やし，カメラの間隔を狭めて配置することである程度は改善されるが，各カメラとビジュアルハルを構成する各面との可視性判定に基づき，複数のカメラのフレーム画像の色をマッピングに用いるなど，本質的な改善策が望まれる．

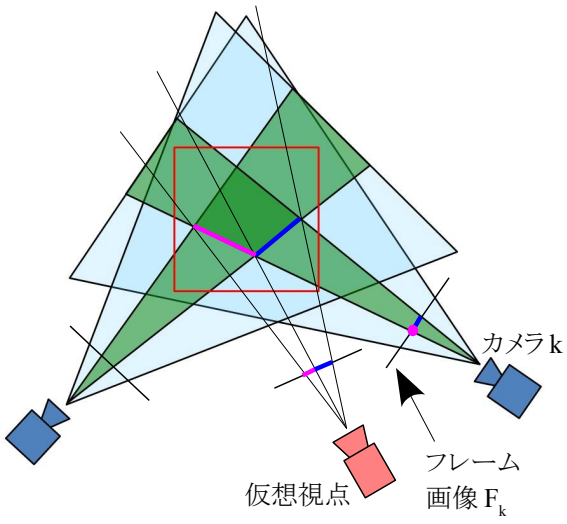


図１１：視点依存型カラーマッピングの欠点

本手法による仮想視点映像の品質を高める要素として，カメラから取得するフレーム画像の解像度，ならびに，平行平板モデルの解像度（平板の枚数）が挙げられる．ここで，平行平板モデルの解像度が仮想視点映像の品質に与える影響を図１２に示す．各画像の下の N*3 の表記は，N が１組の平行平板群中の平板の枚数であり，それが x，y，z 軸方向に３組あることを意味する（３．２節）．図１２より，平板の枚数が多いほど，ビジュアルハルの精度が向上し，仮想視点映像の品質が高まることがわかる．

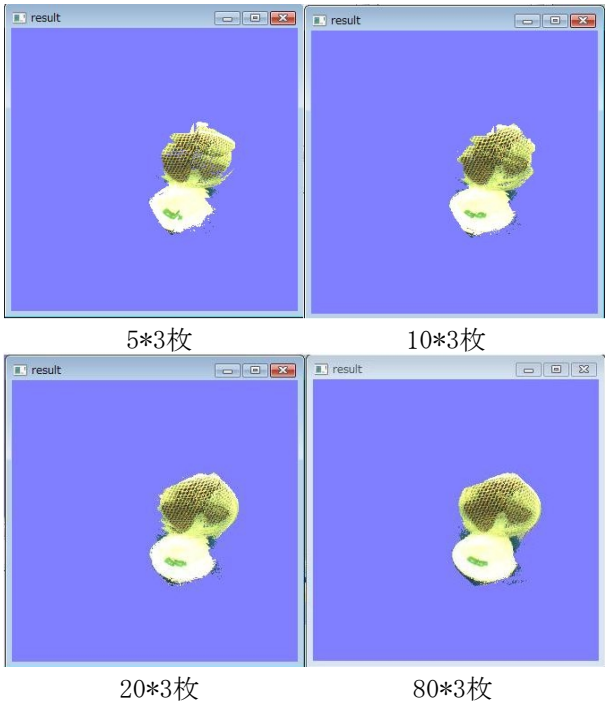


図１２：平板の枚数に対する仮想視点映像の品質の違い

補足資料の動画ファイルからも明らかなように，今回の実験ではカメラを対象物体の手前側にしか設置していないため，仮想視点が対象物体の上方や後方に位置した際には妥当な仮想視点映像を生成できなかった．これは，対象物体の周囲を取り囲むように多数のカメラを配置することで改善できる．その他にも，仮想視点映像の品質を向上させるためには，カメラ校正や背景差分処理の精度を高めることなどが必要である．

本手法によりフレーム時刻ごとに１つの仮想視点フレーム画像を生成するのに要する処理は以下となる．

（A）カメラによりフレーム画像を取得する．
（B）フレーム画像をリサイズする．具体的には，画像の両端を削除することで解像度を 640*480 から 480*480 に変更する．
（C）リサイズしたフレーム画像に対する背景差分処理を行い，前景ピクセルと背景ピクセルを区別した α 値，ならびに，視点依存型カラーマッピングのための RGB 値を与えた背景差分画像を生成する．
（D）背景差分画像を投影用テクスチャとしてテクスチャメモリ領域に割り当てる．
（E）投影用テクスチャの投影マッピングを行う（テクスチャの投影行列の設定，平行平板モデルの生成，マルチテクスチャ処理，α テスト処理，レンダリング処理）．

上記の処理の中で，（A）～（D）は平行平板モデルを構成する平板の枚数に依存せず，一方で，（E）は平板の枚数に依存している．そこで，まず，平板の枚数に依存しない（A）～（D）の処理時間を計測し，その後，平板の枚数を変えて（E）の処理時間を計測した．その結果を表２に示す．表中の処理時間は，１０回の計測を行い，その平均の値とした．また，平板枚数の N＊3 という表記は，図１２と同様，N が１組の平行平板群中の平板の枚数であり，それが３組あることを意味する．なお，N が小さな場合に（E）に同じ値が並んでいるが，これは計測可能な時間の最小単位が 0.1 ミリ秒であったことが原因と考えられる．（A）～（E）の合計時間は，カメラによるフレーム画像の取得から仮想視点フレーム画像を生成するまでの総処理時間である．ここで，（B）の正方形状の解像度（480*480 ピクセル）へのリサイズは，４．１節で述べたように，立方体状の平行平板モデルへの投影マッピングのしやすさ，ならびに，ピクセル数を減らすことによる投影用テクスチャの生成と投影マッピングの処理の効率化を目的としている．実際，実験により，リサイズを行わずに元の解像度（640*480 ピクセル）のままで以降の処理を行った場合，総処理時間が増加することがわかった．ではあるが，一方で，表２から，このリサイズの処理時間は総処理時間に対して大きな割合を占めていることがわかる．理想的には，カメラ撮影の時点で，直接，目的とする解像度のフレーム画像が取得できれば，リサイズは不要となる．表２の（C）～（E）の合計時間は，リサイズにより目的とする解像度のフレーム画像が得られた以降，背景差分処理から仮想視点フレーム画像を生成するまでの処理時間となる．

表２から，平板枚数が N = 10 から 31622 までは処理時間の増加が非常に緩やかであり，平板枚数が増加した場合でも高速な描画ができていることがわかる．例えば，N = 10000 の場合でも，「A＋B＋C＋D＋E」に示すフレーム画像取得から仮想視点フレーム画像生成までの総処理時間は 78.1(ミリ秒/フレーム)であり，約 12.8(fps)の処理速度が実現できている．また，「C＋D＋E」に示す目的とする解像度のフレーム画像が得られた以降の処理時間は 49.3(ミリ秒/フレーム)で約 20.3(fps)となる．この平板枚数はボクセル空間の解像度が 10000^3 の場合に相当し，通常のボクセル空間を用いたボクセルベース視体積交差法では実現できない処理速度である．一方，N = 56234 から以降は処理時間が急激に増加している．これは，ビデオカードがテクスチャマッピングを施したポリゴンを扱う性能等に起因するものと考えられる．

ここで，計算量の見積もりについて考える．本手法で用いる N*3 = 3N 枚の平板を持つ平行平板モデルによる復元空間は，N^3 個のボクセルを持つボクセル空間に相当する．通常のボクセルベース視体積交差法（２章）では，M 台のカメラのフレーム画像上の物体シルエットに対して個々のボクセルを投影してビジュアルハルの内外判定を行うため，全体の計算オーダは $O(M N^3)$ となる．一方，本手法では，M 個の投影用テクスチャを 3N 枚の平板に投影マッピングして α 値の積演算を行う．このとき，１回の投影マッピングと α 値演算の計算量が投影用テクスチャの解像度 L^2 に比例すると考えると，全体の計算オーダは $O(M N L^2)$ となる．つまり，もし復元空間と投影用テクスチャの一辺の解像度が等しく N = L が成り立つとみなすことができれば，両者の計算オーダは等しいと言える．そして，本手法は，投影用テクスチャの解像度 L^2 に比例する部分の処理（投影マッピングと α 値演算）を OpenGL の機能により高速に実行するものであると言える．

また，メモリ使用量の観点から次のことが言える．通常のボクセル空間を用いた方法では，一般に，N^3 個のボクセルをメモリ上に保持する必要があり，実際にプログラムが実行可能であるボクセル空間の大きさに限界があるという問題がある．一方，本手法では，これを 3N 枚の平板の処理で実現できるため，その問題を回避することができる．

表２：提案手法の処理時間

	処理時間（ミリ秒）		
A	4.4		
B	24.4		
C	40.4		
D	4.5		
A＋B＋C＋D	73.7		
平板枚数	E	C＋D＋E	A＋B＋C＋D＋E
10*3	0.1	45.0	73.8
31*3	0.1	45.0	73.8
100*3	0.2	45.1	73.9
200*3	0.2	45.1	73.9
316*3	0.2	45.1	73.9
1000*3	0.6	45.5	74.3
3162*3	1.4	46.3	75.1
10000*3	4.4	49.3	78.1
31622*3	14.5	59.4	88.2
56234*3	1328.0	1372.9	1401.7
100000*3	3872.5	3917.4	3946.2
177827*3	7548.1	7593.0	7621.8
316227*3	16400.6	16445.5	16474.3

※ 表中の平板枚数は，200*3以外は，対数を取った際に均等な間隔となるように選択した．200*3は比較手法による表３との比較のために選択した．

本論文での提案手法の有効性を検証するため，２章で紹介した通常のボクセル空間を用いたボクセルベース視体積交差法としてよく用いられるSpace Carving Method (SCM)を比較手法として実装し，処理速度について比較実験を行った．比較手法によりフレーム時刻ごとに１つの仮想視点フレーム画像を生成するのに要する処理は以下となる．

（＊）ボクセルごとに，各カメラのフレーム画像上で投影されるピクセル位置を求め，登録する．
（A）カメラによりフレーム画像を取得する．
（B）フレーム画像に対する背景差分処理を行い，前景ピクセルと背景ピクセルを２値化して区別した背景差分画像を生成する．
（C）（＊）の登録情報と（B）の２値情報を用いて視体積交差法の計算を行う．具体的には，ボクセルごとに，各カメラのフレーム画像（背景差分画像）上で投影されるピクセルが前景か背景かにより，ビジュアルハルの内部か外部かを判定する．
（D）ビジュアルハルの内部のボクセルについて，フレーム画像を参照してRGB値を与え，仮想視点フレーム画像を生成する．

上記の処理の中で，（＊）はプログラムの起動後に最初に一回だけ行われるもので，フレーム時刻ごとの視体積交差法の計算を高速に実行するための処理である．一方，（A）～（D）がフレーム時刻ごとの処理である．（B）で生成される背景差分画像は，先に述べた提案手法で生成される背景差分画像とは異なり，ピクセルごとに α 値とRGB値を持たず，それに代わり，前景と背景を区別する２値を持つ．これは，比較手法では投影用テクスチャの生成が不要なため，α 値が不要なだけでなく，RGB値をフレーム画像から直接参照できるからである．また，比較手法でも視点依存型カラーマッピングを用い，（D）で参照するフレーム画像は仮想視点と視線方向が最も近いカメラのものを用いた．比較手法では，（A）と（B）はボクセル数に依存せず，一方，（C）と（D）がボクセル数に依存する処理である．そこで，まず，ボクセル数に依存しない（A）と（B）の処理時間を計測し，その後，ボクセル数を変えて（C）と（D）を合わせた処理時間を計測した．その結果を表３に示す．（A）～（D）の合計時間は，カメラによるフレーム画像の取得から仮想視点フレーム画像を生成するまでの総処理時間である．また，（B）～（D）の合計時間は，目的とするフレーム画像が得られた以降，背景差分処理から仮想視点フレーム画像を生成するまでの時間である．なお，提案手法で行ったフレーム画像のリサイズは，比較手法では（B）の背景差分処理を行うピクセル数を減らす効果はあるが，表２の（B）のリサイズの時間（24.4ミリ秒）と表３の（B）の背景差分処理の時間（7.8ミリ秒）を考えた場合，むしろ全体的な処理時間を増やす結果となるため，比較手法ではリサイズを行っていない．また，比較手法では，プログラムを実行できたボクセルの最大個数は200^3であり，それ以上のボクセル数を扱うためのメモリ領域は確保できなかった．

図１３は，表２の「A＋B＋C＋D＋E」と表３の「A＋B＋C＋D」を用いて，提案手法と比較手法について，フレーム画像取得から仮想視点フレーム画像生成までの総処理時間をグラフにしたものである．横軸を復元空間の一辺の解像度Nとし，縦軸を処理時間としている．なお，両軸とも対数としている．表２と表３，および，図１３から，両手法の総処理時間を比べると，比較手法ではNの増加に伴う処理時間の増加が急激であるのに対して，提案手法では前述したようにN = 10から31622までは処理時間の増加が非常に緩やかであることがわかる．そして，N = 100までは比

較手法のほうが処理時間が短いが，N = 200 では両手法の処理時間の大小が逆転し，提案手法のほうが処理時間が短くなっている．一方，これも前に述べたが，提案手法ではN = 56234 から以降に処理時間が急激に増加しており，これはビデオカードがテクスチャマッピングを施したポリゴンを扱う性能等に起因するものと考えられる．また，図１４は，表２の「Ｃ＋Ｄ＋Ｅ」と表３の「Ｂ＋Ｃ＋Ｄ」を用いて，両手法について，目的とするフレーム画像が得られた以降，背景差分処理から仮想視点フレーム画像生成までの処理時間をグラフにしたものである．このグラフからも図１３のグラフと同様の傾向が見られる．本実験では，メモリ確保の問題から，復元空間の一辺の解像度がN = 200 よりも大きな場合に対して比較手法を実行することができなかったが，実験の結果から，Nが極端に大きな値をとる場合のビデオカードの性能等に起因するであろう処理速度の低下はあるものの，Nがその値に達するまでは，Nが増加するにつれて処理速度に関する比較手法に対する提案手法の優位性は増すものと考えられる．また，メモリ使用量の観点についても，通常のボクセル空間を用いた比較手法ではプログラムが実行可能であるボクセル空間の大きさに限界が生じた一方で，提案手法ではその問題を回避できており，提案手法の優位性が示されたと言える．

表３　比較手法の処理時間

	処理時間（ミリ秒）		
A	4.4		
B	7.8		
A＋B	12.2		
ボクセル数	C＋D	B＋C＋D	A＋B＋C＋D
10^3	1.0	8.8	13.2
31^3	1.4	9.2	13.6
100^3	30.2	38.0	42.4
200^3	236.2	244.0	248.4

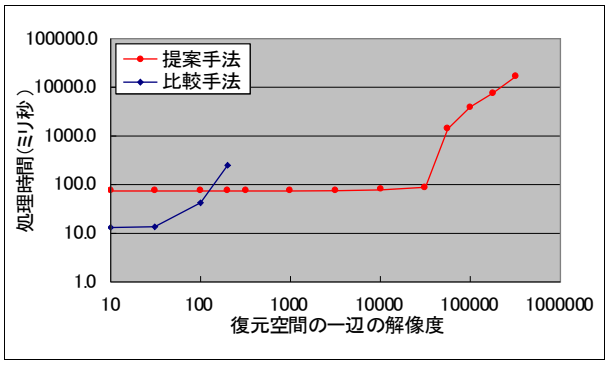


図１３：提案手法と比較手法の処理時間（フレーム画像取得から仮想視点フレーム画像生成までの総処理時間）

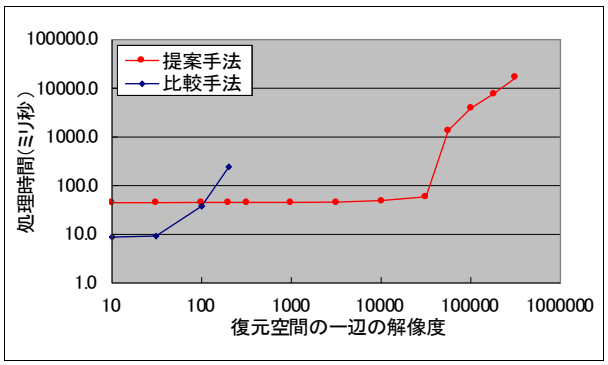


図１４：提案手法と比較手法の処理時間（背景差分処理から仮想視点フレーム画像生成までの処理時間）

５．おわりに

本研究では，多視点カメラ映像に対する投影マッピングを用いた視体積交差法による高速な仮想視点映像の生成法を提案し，実験により有効性を検証した．

今後の課題としては，まずは，現在の視点依存型カラーマッピングの問題の解決を目指し，各カメラとビジュアルハルを構成する各面との可視性判定に基づき，複数のカメラのフレーム画像の色を有効に利用する方法の提案が挙げられる．また，今回は４台のカメラを用いた実験を行なったが，より高品質な仮想視点映像を生成するためには，カメラの台数を増やす必要がある．さらに，カメラの性能に依存する点ではあるが，リサイズをすることなく目的とする解像度のフレーム画像を直接に取得することで全体の処理時間を短縮することや，より高解像度のフレーム画像を取得することで仮想視点映像の品質を高めることも挙げられる．これらを実現するために，今後，多数のカメラを複数のＰＣに分けて接続し，高解像度のフレーム画像に対する背景差分等のカメラごとの処理を並列化して高速に実行するような，ＰＣクラスタ

等による分散環境の構築を予定している．さらに，それに関連して，提案手法自体をＰＣクラスタやマルチコアプログラムの仕組みの上で並列化し，さらなる高速化をはかることも今後の課題である．現在，OpenGL の投影マッピングの機能を利用しているが，GPU 上での高速な実行に向けたシェーダ言語による実装も課題である．さらに，より精度の高い仮想視点映像の生成を目指し，異なるカメラ映像間の対応ピクセルの色の同一性（photo consistency）を用いてビジュアルハルを補正することによる凹形状の復元も課題として挙げられる．

謝辞

本研究の実験の一部をサポートしてくれた岩手大学大学院工学研究科博士前期課程の樋口拓馬君に感謝する．本研究の一部は科学研究費補助金（基盤研究(C) 21500090）の援助を受けている．

注記

本論文は，文献[28]の第 26 回 NICOGRAPH 論文コンテストにて発表した内容に基づき，記述内容や実験等に改善を加えて投稿したものである．

参考文献

[1] S. E. Chen, L. Williams, View Interpolation for Image Synthesis, Proceedings of SIGGRAPH 1993, pp.279-288, 1993.

[2] L. McMillan, G. Bishop, Plenoptic Modeling: An Image-Based Rendering System, Proceedings of SIGGRAPH 1995, pp.39-46, 1995.

[3] J. Shade, S. J. Gortler, L.-w. He, R. Szeliski, Layered Depth Images, Proceedings of SIGGRAPH 1998, pp.231-242, 1998.

[4] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality Video View Interpolation Using a Layered Representation, Proceedings of SIGGRAPH 2004, pp.600-608, 2004.

[5] M. Levoy, P. Hanrahan, Light Field Rendering, Proceedings of SIGGRAPH 1996, pp.31-42, 1996.

[6] S. J. Gortler, R. Grzeszczuk, R. Szeliski, M. F. Cohen, The Lumigraph, Proceedings of SIGGRAPH 1996, pp.43-54, 1996.

[7] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, M. F. Cohen, Unstructured Lumigraph Rendering, Proceedings of SIGGRAPH 2001, pp.425-432, 2001.

[8] A. Laurentini, The Visual Hull Concept for Silhouette-based Image Understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.16, No.2, pp.150-162, 1994.

[9] W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan, Image Based Visual Hulls, Proceedings of SIGGRAPH 2000, pp.369-374, 2000.

[10] W. Matusik, C. Buehler, L. McMillan, Polyhedral Visual Hulls for Real-Time Rendering, Proceedings of Eurographics Workshop on Rendering 2001, pp.115-125, 2001.

[11] G. Slabaugh, B. Culbertson, T. Malzbender, R. Shafer, A Survey of Methods for Volumetric Scene Reconstruction from Photographs, Proceedings of the International Workshop on Volume Graphics 2001, pp.81-100, 2001.

[12] S. M. Seitz, C. R. Dyer, Photorealistic Scene Reconstruction by Voxel Coloring, Computer Vision and Pattern Recognition Conf., pp.1067-1073, 1997.

[13] 冨山仁博，片山美和，岩舘祐一，今泉浩幸，視体積交差法とステレオマッチング法を用いた多視点画像からの 3 次元動オブジェクト生成手法，映像情報メディア学会誌，Vol.58，No.6，pp.797-806，2004.

[14] 冨山仁博，片山美和，折原豊，岩舘祐一，局所的形状特徴に拘束された 3 次元形状復元手法とそのリアルタイム動画表示，映像情報メディア学会誌，Vol.61，No.4，pp.471-481，2007.

[15] H. Kim, R. Sakamoto, I. Kitahara, N. Orman, T. Toriyama, K. Kogure, Compensated Visual Hull for Defective Segmentation and Occlusion, Proceedings of 17th International Conference on Artificial Reality and Telexistence (ICAT2007), pp.210-217, 2007.

[16] S. Corazza, L. Mundermann, A. M. Chaudhari, T. Demattio, C. Cobelli, T. P. Andriacchi, A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach, Annals of Biomedical Engineering, Vol.34, No.6, pp.1019-1029, 2006.

[17] D. Vlasic, I. Baran, W. Matusik, J. Popovic, Articulated Mesh Animation from Multi-view Silhouettes, Proceedings of SIGGRAPH 2008, pp.97:1-9, 2008.

[18] ウ小軍，和田俊和，東海彰吾，松山隆司，平面間透視投影を用いた並列視体積交差法，情報処理学会論文誌，Vol.42，No.SIG6(CVIM2)，pp.33-43，2001.

[19] M. Z. Brown, D. Burschka, G. D. Hager, Advances in Computational Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.8, pp.993-1008, 2003.

[20] P. Debevec, Y. Yu, G. Borshukov, Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping, Proceedings of 9th Eurographics Workshop on Rendering, pp.105-116, 1998.

[21] T. Matsuyama, X. Wu, T. Takai, T. Wada, Real-Time Dynamic 3D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3D Video, IEEE Transactions on Circuits and Systems for Video Technology, Vol.CSVT-14, No.3, pp.357-369, 2004.

[22] J. Neider, T. Davis, M. Woo, OpenGL(TM) Programming Guide: The Official Guide to Learning OpenGL, Release 1, Addison-Wesley Publishing Company, 1993.

[23] 床井浩平，床井研究室ＨＰ“第６回投影マッピング”，http://marina.sys.wakayama-u.ac.jp/~tokoi/?date=20040920

[24] 床井浩平，床井研究室ＨＰ“第７回テクスチャ座標の自動生成”，http://marina.sys.wakayama-u.ac.jp/~tokoi/?date=20041228

[25] 床井浩平，床井研究室ＨＰ“第１９回マルチテクスチャ”，http://marina.sys.wakayama-u.ac.jp/~tokoi/?date=20050615

[26] 床井浩平，床井研究室ＨＰ“第４回アルファテスト”，http://marina.sys.wakayama-u.ac.jp/~tokoi/?date=20040916

[27] Z. Zhang, A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pp.1330-1334, 2000.

[28] 竹中史雄，藤本忠博，原美オサマ，千葉則茂，テクスチャ投影を利用した多視点カメラ映像からのリアルタイムな形状復元法，第26回NICOGRAPH論文コンテスト，pp.I4:1-7，2010.

竹中 史雄

2010年岩手大学工学部情報システム工学科卒業．2010年岩手大学大学院工学研究科博士前期課程デザイン・メディア工学専攻入学，現在，在籍中．

藤本 忠博

1990年慶應義塾大学理工学部電気工学科卒業．1992年慶應義塾大学大学院理工学研究科修士課程計算機科学専攻修了．1992年(株)三菱総合研究所入社，1995年同退社．1995年慶應義塾大学大学院理工学研究科博士課程計算機科学専攻入学，1999年同単位取得退学．1999年岩手大学工学部情報工学科助手．2000年博士(工学)．2000年岩手大学工学部情報システム工学科(改組)助手．2002年同講師．2005年同助教授．2007年同准教授．2009年岩手大学工学部電気電子・情報システム工学科(改組)准教授．芸術科学会，電子情報通信学会，情報処理学会，IEEE，ACM会員．

原美 オサマ

1992年ダマスカス大学電気工学科卒業．1998年上海大学大学院修士課程コンピュータサイエンス専攻修了．2001年北陸先端科学技術大学院大学情報科学研究科博士後期課程情報科学専攻修了，博士(情報科学)．2001年同助手．2003年岐阜大学バーチャルシステムラボラトリー研究員．2006年岩手大学工学部情報システム工学科助手．2007年同助教．2009年岩手大学工学部電気電子・情報システム工学科(改組)助教．2010年カタール大学工学部情報工学科助教．芸術科学会，日本バーチャルリアリティ学会，ACM会員．

千葉 則茂

1975年岩手大学工学部電気工学科卒業．1975年日本ビジネスコンサルタント（現，(株)日立情報システムズ）入社，1978年同退社．1984年東北大学大学院博士課程情報工学専攻修了，工学博士．1984年東北大学工学部助手．1986年仙台電波高専情報工学科助教授．1987年岩手大学工学部情報工学科助教授．1991年同教授．2000年岩手大学工学部情報システム工学科(改組)教授．2009年岩手大学工学部電気電子・情報システム工学科(改組)教授．芸術科学会，電子情報通信学会，情報処理学会，IEEE，ACM会員．