

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

High Fidelity Visualization Algorithm and 3D Editing System for 3D Video

学生会員 高井 勇志[†], 松山 隆司[†]Takeshi Takai[†] and Takashi Matsuyama[†]

Abstract 3D video is the ultimate medium for recording images of a dynamic visual event in the real world. The recorded behaviors of objects can be observed from any viewpoint, because 3D video records whole 3D shape, motion, and precise surface properties (i.e. color and texture) of objects. In our last paper¹⁾, we presented a method of reconstructing the shape of a dynamic 3D object from multi-view video images, which enables a temporal series of 3D voxel representations of an object's behavior to be obtained in real-time. In this paper, we present (1) an algorithm for generating texture on the surface of a 3D object from multi-view video images, and (2) an editing system for visualizing 3D video with an omnidirectional background image using a versatile 3D camera action. This paper focuses on how we can generate high fidelity object images from arbitrary viewpoints based on 3D object shapes of limited accuracy. We propose a novel texture-mapping algorithm, which maps textures onto the surface of 3D object depending on the specific viewpoint. Experimental results demonstrate the effectiveness of the algorithm in generating high fidelity images of objects from arbitrary viewpoints.

キーワード：3次元ビデオ映像，テクスチャマッピング，3次元映像編集，任意視点映像生成，多視点ビデオ，3次元動作復元

1. ま え が き

我々は，多視点ビデオ映像から運動対象の3次元形状とその表面情報を時系列データとして生成し，それらを任意の視点から鑑賞できるメディアとして「3次元ビデオ映像」を提案している¹⁾⁻³⁾．先の論文1)では，多視点ビデオ映像から対象の動的な3次元形状を実時間で復元する手法を提案した．

最近では，多視点映像から任意視点映像を生成する手法が国内外で盛んに研究されている⁴⁾⁻⁸⁾が，計算コストやデータ量の観点から，3次元形状を陽に復元しない例も多い．しかし，見え方のみの復元ではなく，現実世界の対象の動作をそのまま記録し photometric な属性の推定や，動作の認識，高精細デジタルアーカイブとしての利用を考慮した場合，対象の3次元形状復元は不可欠であると考えられる．

本稿では，まず論文1)の方法で得られた3次元ボクセル表現(図1)に対して離散マーチングキューブ法¹⁴⁾を適用して対象の多面体表現を求める．そして

(1) 多視点映像より得られた表面テクスチャ情報を



図1 対象動作のボクセル表現
Voxel representations of a moving object

付加し対象を任意視点から高精細に表示するアルゴリズム

(2) 3次元ビデオ映像を全方位背景画像とともにインタラクティブに編集するシステム

を提案する．

本論文では，特に，必ずしも正確であるとは言えない対象の3次元ボクセルデータを基にして，いかに自然な任意視点映像を生成するかという問題に焦点を当てて検討を行う．以下では，多視点映像からの任意視点映像の生成法を概観した後，上記(1)，(2)のためのアルゴリズムを示し，実験によってその有効性を明らかにする．

2. 多視点映像からの任意視点映像の生成

多視点映像から任意視点映像を生成する手法は，そのアプローチから以下のように分類できる．

まず，対象の形状については，
・3次元形状を復元しない手法

2001年8月31日，映像情報メディア学会 映像表現研究会で発表
2001年8月31日受付，2001年12月10日再受付，2002年1月10日採録
†京都大学 大学院 情報学研究科
(〒606-8501 京都市左京区吉田本町，TEL 075-753-5058)
†Graduate School of Informatics, Kyoto University
(Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan)

表 1 任意視点映像生成法の分類
Approaches to the arbitrary-viewpoint image generation

形状	app	app	gen	gen
テクスチャ	app	gen	app	gen
Type	I	II	III	IV

- appearance の復元
 - ・ 3次元形状を復元する手法
 - generic property の復元
- また、表面テクスチャ情報については、
- ・ 撮影した照明環境における見え方を復元する手法
 - appearance の復元
 - ・ 照明環境に不変な表面情報を復元する手法
 - generic property の復元

となる。まとめると、表 1 のように書ける。

Type I (形状:app & テクスチャ:app)

陽に 3次元形状を復元することなく、任意視点から見た映像を生成する手法である。3次元形状を復元することは、見えない部分の形状データの生成や保持を伴うため、計算コストを増大させデータも冗長になると言える。そこで、非常に多くの多視点映像から、指定された視点・視線に近い映像を合成する手法⁹⁾や、中間表現としては対象の 3次元形状を復元しつつも、最終的にはこの形状データを含まない手法⁷⁾、また、中間表現として 2次元平面の組合せを用いる手法⁸⁾などが研究されている。しかし、動作認識やアーカイブとしての利用、また表面の反射特性の推定には 3次元形状が不可欠なため、このタイプの手法の応用は限られている。

Type II (形状:app & テクスチャ:gen)

一般に、照明に不変な対象表面の反射特性を推定するには、対象の 3次元形状を復元し反射モデルに基づいた解析¹¹⁾を行う必要があり、このタイプの手法の実現は難しい。一方、文献 10) や 12) では、反射モデルを利用することなく、多様な照明下で撮られた多数枚の画像に対して固有画像解析を施すことによって、任意照明下でのテクスチャの復元が行えることを示している。

Type III (形状:gen & テクスチャ:app)

対象の 3次元形状を復元し、表面情報は見え方をベースに復元する手法で、本研究のアプローチはこのタイプである。この手法では、対象の 3次元動作に同期したカメラワークなど、見え方のみの手法とは異なる映像表現が可能となる。ただし、表面テクスチャは撮影時の照明環境下でのものが使われるため、照明条件を変更した映像は生成できない。

Type IV (形状:gen & テクスチャ:gen)

対象の 3次元形状の復元と共に、撮影されたビデオ映像から撮影時の照明環境に依存する要素を取り除き、対象表面の反射特性を推定する手法である。このタイプでは、任意の光源下での見え方の復元といった、他のタイ

プでは表現不可能な映像を生成することが可能となるが、対象の反射モデルの推定には、正確な 3次元形状情報の取得や撮影時の照明環境の推定¹³⁾といった難しい処理が必要となる。

以上述べたように、多視点映像から任意視点映像を生成する手法は、幾つかの立場の異なったコンセプトを元にして研究されている。本研究では、3次元空間における 3次元ビデオ映像の編集、対象の動作に同期したカメラワーク設定といった 3D ビデオ映像とユーザとのインタラクションの実現を目指しているため、対象の 3次元形状を復元し、それに基づいた任意視点映像の生成を考えた。

3. 3次元ビデオ映像の表示

3次元ビデオ映像は 3次元対象、全方位背景、仮想カメラからなり、これらの要素は図 2 に示すような仮想シーン内に配置されている。この仮想シーン内において、仮想カメラを任意の数・位置に配置し、任意の 3次元的なカメラワークを施すことで、多様な映像を編集・表示することができる。

こうした任意のカメラワークに対応できる背景画像として、本研究では視点固定型パン・チルトカメラにより撮影された全周囲パノラマ画像¹⁵⁾(図 3)を用いている。この全周囲パノラマ画像は非常に高解像度であるため、映像化においてズーム倍率を大きくしても充分鑑賞に耐えることができる。

一方、対象の 3次元映像データは以下の処理によって生成される。



図 2 仮想シーンのセットアップ
Virtual scene setup



図 3 全周囲パノラマ画像
Omni-directional image

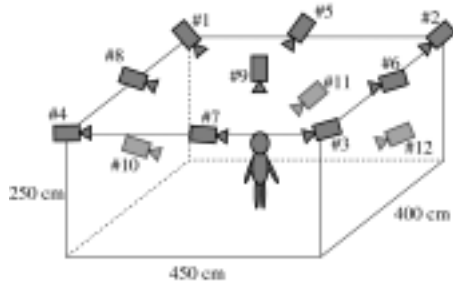


図 4 撮影環境
Camera setting

1. 多視点映像の撮影

まず、対象の動作を図 4 に示す撮影環境下で多視点映像として撮影し、以下の処理を多視点映像の各ビデオフレーム毎に行う。

2. 3次元形状の復元

多視点映像から対象のシルエットを抽出し、視体積交差法を用いた形状復元法¹⁾を適用することにより、対象形状のボクセル表現を得る。

3. ボクセル表現から多面体表現への変換

ボクセルデータに対して離散マーチングキューブ法¹⁴⁾を適用することにより、対象の多面体(三角形パッチ)表現を生成する。

4. テクスチャマッピング

カメラで撮影した元の多視点映像から、各三角形パッチに対するテクスチャを求めることにより、対象の実写3次元像が得られる。

しかし、上記の方法により得られる多面体表現には以下のような問題点がある。

- ・視体積交差法はその原理上、対象の形状を完全に正確には復元できないため、ボクセル表現に不正確な箇所が現れる。(直感的には対象の凹の部分は復元できない。)

- ・離散マーチングキューブ法により生成される面は、滑らかに連結されているとは限らない。

本研究では、このような必ずしも正確であるとは言えない対象の3次元形状に対して、写実性の高い映像を再現するためのテクスチャマッピング法を開発することを目指している。以下では、本論文で提案するテクスチャマッピング法を述べる。

4. 高精細テクスチャマッピング法

4.1 3次元パッチデータに対するテクスチャマッピング

まず、3次元パッチデータに対するテクスチャマッピング法を以下のように大別する。

(1) 視点独立/視点依存

映像化の際に、表示視点(図2の仮想カメラの位置)の情報を利用してテクスチャを生成する手法を「視点依存法」、そうでないものを「視点独立法」とする。

・視点独立法の特徴

各三角形パッチに固有のテクスチャをマッピングし、得られたテクスチャ付き3次元像を表示視点から見た映像として生成する。この方法では、一度テクスチャを生成すればそれを基に任意視点映像が生成できるため、高速な表示が可能である。しかし、各三角形パッチにテクスチャをマッピングする際に、どのカメラで撮られた映像からテクスチャを生成するかが大きな問題となり、不適当な選択を行うと対象表面のテクスチャにちらつきが生じてしまう。

・視点依存法の特徴

表示視点から見える三角形パッチのみを対象として、それらのパッチが表示視点からどのように見えるかに基づいて、テクスチャマッピングを行う。この方法では、表示視点に近いカメラで撮影された映像からテクスチャを得ることができるため、撮影時の照明環境下での対象を高い写実性で映像化できる。その反面、表示視点が移動する度に、テクスチャの再生成が必要となるため、表示が遅くなる。また、多視点映像を常に保持しておく必要があるため、メモリ使用量が大きくなるという欠点がある。

(2) 面ベース/頂点ベース

三角形パッチの面上のテクスチャを画像として生成しマッピングを行う一般的なテクスチャマッピングを「面ベース法」、面の各頂点に色情報を持たせ、それらの補間によって面上のテクスチャを生成する手法を「頂点ベース法」とする。

・面ベース法の特徴

三角形パッチ上にはカメラで取られた高解像度のテクスチャがマッピングされるので、写実性の高い映像が生成できる。反面、テクスチャを画像として保持する必要があるためメモリの使用量が大きくなる。また、同一のカメラ映像からテクスチャ生成を行ったとしても、隣接面パッチ間の繋がりを考慮しなければ、面パッチ間でテクスチャの不整合が生じることになり、対象表面にちらつきが生じてしまう。

・頂点ベース法の特徴

頂点の色データを保持するだけでよいので、メモリの使用量は非常に小さくなる。また、隣接する面パッチは頂点を共有しているため、面パッチ間でのテクスチャの不整合は起こらない。そのため、滑らかなテクスチャを生成することが可能となる。しかし、3次元形状の解像度が充分高くないと写実性の高い映像は得られない。

以下では、以前の研究²⁾³⁾で用いた視点独立面ベース法とその問題点、それらを解決するために本研究で考案した視点依存頂点ベース法について述べる。

4.2 視点独立面ベース法

テクスチャマッピングアルゴリズム(図5)²⁾³⁾

step.1 テクスチャマッピングを行う面 p_i とその面に隣

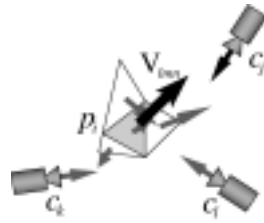


図 5 視点独立面ベースのテクスチャ生成法
Viewpoint-independent texture generation

接するすべての面の法線の平均ベクトル (局所平均法線ベクトル V_{lmn}) を求める。

step.2 V_{lmn} とカメラ c_j から面 p_i に向かうベクトル V_{c_j} のなす角が最大となるカメラ c_j を選ぶ。

step.3 カメラ c_j により撮影された映像を用いて、面パッチ p_i のテクスチャを生成する。

しかし、この手法では以下のような問題点が生じてしまう。

(1) カメラ間の撮影映像の差異によるちらつき

マーチングキューブ法により生成される面は滑らかではないため、隣接する面パッチ間で法線の向きが大きく変化する。そのため、テクスチャマッピングに使用するカメラ c_j が、隣接パッチで異なってしまふ。つまり、異なったカメラで撮られた画像が対象表面の隣接パッチに張られ、映像化された対象表面にちらつきが生じてしまう。(視点独立法の問題点)

(2) 隣接面パッチ間でのテクスチャの不整合によるちらつき

同じカメラで撮影された映像を基にして、テクスチャマッピングを行った場合でも、テクスチャは面パッチ毎に独立して生成されるため、各面パッチの継ぎ目において不整合が生じることがある。このことによっても、映像化された対象表面にちらつきが生じてしまう。(面ベース法の問題点)

4.3 視点依存頂点ベース法

上記の問題を解決し、滑らかなテクスチャマッピングを実現するために、視点依存頂点ベース法を考えた。なお、頂点ベース法では、3次元形状の解像度が問題となるが、本研究で用いる3次元形状は1cmボクセルサイズで表現されているため、対象が人などである場合は、十分な解像度が得られていると考えてよい。

(1) 語句と記号の定義

まず、以下のように語句と記号を定義する(図6)。

- ・撮影カメラの集合： $C = \{c_1, c_2, \dots, c_n\}$
- ・表示視点： eye
- ・離散マーチングキューブ法で得られた多面体モデルを構成する面パッチの集合： $P = \{p_1, p_2, \dots, p_m\}$

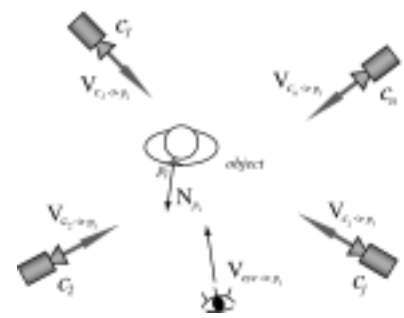


図 6 視点, カメラ視点
Viewpoint, camera position

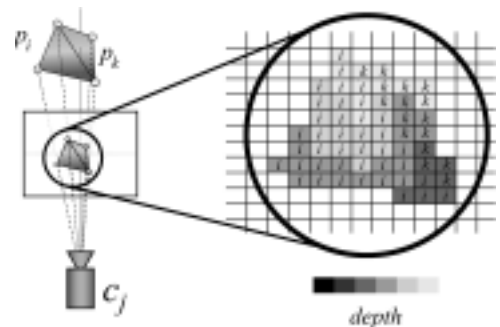


図 7 デプスバッファ
Depth buffer

- ・面パッチ p_i の法線： N_{p_i}
- ・表示視線： $V_{eye \to p_i}$ (表示視点から注目する面パッチ p_i の重心へのベクトル)
- ・カメラ c_j の視線： $V_{c_j \to p_i}$ (カメラ c_j から注目する面パッチ p_i の重心へのベクトル)
(それぞれのベクトルは正規化されているものとする。以降も同様。)

- ・面パッチ p_i の頂点： $\mathbf{v}_{p_i}^k$ ($k = 1, 2, 3$)
- ・カメラ c_j からの可視頂点： \mathbf{v}_{p_i, c_j}^k (後述)
- ・可視頂点 \mathbf{v}_{p_i, c_j}^k の RGB 値： $I(\mathbf{v}_{p_i, c_j}^k)$ (後述)
- ・カメラ c_j のデプスバッファ(図7)： B_{c_j} (カメラ c_j により撮影された映像と同じ座標系を持つ平面上に投影する。そして、各画素の位置に投影されたパッチの内、カメラの投影中心からの距離が最小であるパッチを求め、そのパッチ ID と距離値を画素の値とする。またパッチの頂点が投影された画素には、その頂点 ID も記録する。)

(2) カメラ c_j からの可視頂点

あるカメラ c_j から見えている面パッチ p_i の可視頂点 \mathbf{v}_{p_i, c_j}^k は以下のように定義できる。

(1) 法線 N_{p_i} とカメラ視線 $V_{c_j \to p_i}$ の間に式 (1) が成り立つ。

$$N_{p_i} \cdot V_{c_j \to p_i} < 0 \quad (1)$$

(2) 他の面パッチ $p_k \in P$ により隠蔽されていない。

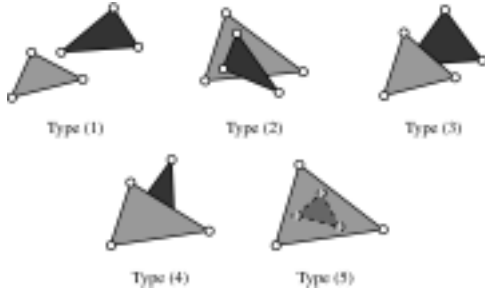


図 8 面パッチ頂点の関係
Relations between vertices and surface patches

\mathbf{v}_{p_i, c_j}^k を求める処理は以下のようなになる。まず与えられたカメラ視線 $V_{c_j \rightarrow p_i}$ に対して式 (1) が成り立つすべての面 $p_i \in P$ をデプスバッファ B_{c_j} に投影する。このとき、デプスバッファに記録されている面パッチ群とそれらの頂点の位置関係は図 8 のように分類でき、他の面によって隠蔽されていない頂点をカメラ c_j からの面パッチ p_i 上の可視頂点 \mathbf{v}_{p_i, c_j}^k とする。つまり、type(1) と type(2) はすべての頂点が可視頂点であり、type(3) と type(4) においては他の面に隠蔽されていない頂点が可視頂点である。また、type(5) のように、他の面パッチに完全に隠蔽された面パッチ（図中の破線の三角形）には可視頂点は存在しない。

可視頂点 \mathbf{v}_{p_i, c_j}^k の RGB ベクトル $I(\mathbf{v}_{p_i, c_j}^k)$ は以下のよう求める。

$$I(\mathbf{v}_{p_i, c_j}^k) = I_{c_j}(\hat{\mathbf{v}}_{p_i, c_j}^k) \quad (2)$$

ただし、 $I_{c_j}(\mathbf{v})$ はカメラ c_j で撮影された画像の画素位置 \mathbf{v} での RGB 値を表し、 $\hat{\mathbf{v}}_{p_i, c_j}^k$ は可視頂点 \mathbf{v}_{p_i, c_j}^k をその画像上に投影したときの画素位置を表す。

(3) テクスチャマッピングアルゴリズム

- step.1 すべてのカメラ $C = \{c_1, \dots, c_n\}$ に対して、すべての頂点 $\mathbf{v}_{p_i}^k$ の可視・不可視判定を行い、可視頂点の RGB 値を予め決定しておく。
- step.2 表示視点 eye を決定する。
- step.3 すべての面パッチ $p_i \in P$ について step.4-9 の処理を行う。また、表示用として使う面パッチ p_i の各頂点の RGB 値 $I(\mathbf{v}_{p_i}^k) = \mathbf{0}$ ($k = 1, 2, 3$) と初期化しておく。
- step.4 $V_{\text{eye} \rightarrow p_i} \cdot N_{p_i} < 0$ が成り立つ場合のみ、step.5-9 の処理を行う。（その面パッチ p_i は視点 eye から見て表向きである。）
- step.5 すべてのカメラ $c_j \in C$ について、重み係数 $w_{c_j} = (V_{c_j \rightarrow p_i} \cdot V_{\text{eye} \rightarrow p_i})^m$ を計算する。ここで m は重み付け係数*である。また、 $w_{c_j} < 0$ となる場合は $w_{c_j} = 0$ とする。

*この m の値を大きくとれば、表示視線 ($V_{\text{eye} \rightarrow p_i}$) により近い視線

step.6 面パッチ p_i のすべての頂点 $\mathbf{v}_{p_i}^k$ について step.7-8 の処理を行う。

step.7 頂点 $\mathbf{v}_{p_i}^k$ がカメラ c_j より可視である場合は $w_{c_j}^k = w_{c_j}$ 、不可視である場合は $w_{c_j}^k = 0$ とし、式 (3) を用いて重みの正規化を行う。

$$\bar{w}_{c_j}^k = \frac{w_{c_j}^k}{\sum_l w_{c_l}^k} \quad (3)$$

step.8 頂点 $\mathbf{v}_{p_i}^k$ の RGB 値 $I(\mathbf{v}_{p_i}^k)$ を式 (4) より求める。

$$I(\mathbf{v}_{p_i}^k) = \sum_{j=1}^n \bar{w}_{c_j}^k I(\mathbf{v}_{p_i, c_j}^k) \quad (4)$$

step.9 頂点の RGB 値を用いた線形補間によって面パッチ p_i 上のテクスチャを生成する。すなわち、RGB 値を持つ頂点数が、

- ・ 3 箇の場合: 3 頂点の RGB 値の線形補間によって三角形内の各点の RGB 値を求める。
- ・ 2 箇の場合: 残りの 1 点の RGB 値を他の 2 点の RGB 値の平均値として与え、線形補間をする。
- ・ 1 箇の場合: 残りの 2 点の RGB 値はこの 1 点の値と同値とする。つまり生成されるテクスチャは 1 色となる。
- ・ 0 箇の場合: この面パッチにはテクスチャは生成されない。

RGB 値を持たない頂点の色付けは、大雑把な推定となるが、面パッチの大きさが対象に比べて充分小さい場合には、妥当な値が得られるものと考えられる。

以上の処理の結果、1 フレーム分の実写 3 次元像が生成される。

5. 3 次元ビデオ映像の編集システム

図 2 に示した仮想シーンに基づいて 3 次元ビデオ映像を表示するには、非常に多くのパラメータを設定する必要がある。具体的には、

- ・ 対象の 3 次元位置・姿勢・大きさ・数
- ・ 背景の 3 次元位置・姿勢・大きさ・数
- ・ 仮想カメラの 3 次元位置・パン・チルト・ズームを、時系列データとして設定する必要がある。このように、高い自由度を持つパラメータセットを有することは、多様な映像表現を可能にする反面、ユーザに対して大きな負担を強いることになる。

そこで本研究では、このパラメータ設定において以下の 2 通りの設定法を考え、これらのパラメータをインタラクティブに設定するための GUI を開発した。

- (1) キーフレームにおいて設定した各パラメータを時間的に補間する手法
- (2) 対象の 3 次元形状・運動を利用してパラメータ設定を自動的に行う手法

($V_{c_j \rightarrow p_i}$) を持つカメラ c_j の重み w_{c_j} が相対的に大きくなる。つまり、表示視線に近い視線を持つカメラ c_j の影響を大きくすることで、カメラ c_j により撮影された画像の見え方に近いテクスチャ生成が期待できる。

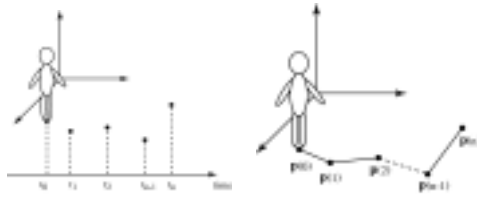


図 9 時間的補間によるパラメータの内挿
Temporal interpolation of parameters

以下にこれらの手法の詳細を述べる．なお、これらのパラメータは図 2 におけるシーン（ワールド）座標系において定義される．

5.1 キーフレーム補間法 (図 9)

従来の CG アニメーション制作において一般的に用いられる手法で、任意のいくつかのフレームにおいて、対象やカメラの 3 次元位置や姿勢などのパラメータを設定し、それらを時間的に補間するものである．本研究では、線形補間を行っているが、高次曲線の当てはめによる補間なども考えられる．

5.2 対象の 3 次元形状・運動を利用した自動パラメータ設定法

上記 (1) の手法では、生成される映像を考慮に入れながらパラメータ設定を行う必要がある．つまり、対象の正面を撮り続けるというようなカメラワークを設定する際には、対象の動きとカメラの位置関係をユーザが常に考慮する必要があるため、ユーザの負担は非常に大きくなる．この問題を解決するため、対象の 3 次元形状・運動を利用した自動パラメータ設定法を考案した．

視体積交差法で得られた対象の 3 次元形状データは、以下のような情報を持っている．

・ 3 次元位置

対象全体のボクセルデータの重心を対象の位置とする．この位置は撮影環境 (図 4) である対象座標系上で定義される．映像化の際には、この座標値をシーン座標系に変換し対象の位置とする．対象座標系とシーン座標系は以下のような関係になっている．

$$\mathbf{x}_s = \mathbf{s}R \cdot \mathbf{x}_o + T \quad (5)$$

ここで、 \mathbf{x}_s 、 \mathbf{x}_o はそれぞれ、シーン座標系上の点、対象座標系上の点を表し、 \mathbf{s} は拡大・縮小、 R は回転、 T は並進を表す行列である．本研究では、

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

と設定した．

・ 高さ

対象の z 軸方向の長さを対象の高さとする．

・ 向き

対象の正面方向を表すベクトルで、映像化の際には、3 次元位置と同様にシーン座標上に投影される．この向きは、対象形状から推定することも可能であ

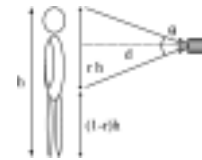


図 10 カメラと対象の位置関係 (I)
Relation between a virtual camera and an object (I)

るが、本研究では予め人間が判断し与えておいた．

これらの情報は、対象が直立した人間であるとして推定される．これらを利用し以下のような指定を行うことで、カメラパラメータを自動設定することが可能となる．

(1) 表示画面に対する対象の大きさの割合 (画面構成) の指定

フルショット、ミドルショット、クローズアップなど．

(2) 対象を撮影する方向の指定

正面、側面、背面など．

以上の指定を行うだけで、任意のカメラワークで映像化する事ができる．

画面構成設定のためのパラメータ計算

画面構成の指定に用いるカメラパラメータは以下のように表される．

◇ 対象までの距離 d

$$d = \frac{r \cdot h}{2 \tan \frac{\theta}{2}} \quad (6)$$

ここで、 h は対象の高さである．また、 r は表示画面に対する対象の大きさの割合であり、以下のように定義する¹⁶⁾．カメラの画角 θ は任意に設定する (図 10) ．

・ $r = 1.0$ フルショット

・ $r = 0.7$ ニーショット

・ $r = 0.5$ ミドルショット

・ $r = 0.3$ クローズショット

・ $r = 0.2$ クローズアップ

◇ カメラの位置 (x_c, y_c, z_c)

$$\begin{cases} x_c = d \cos \phi \cos(\psi + \delta) + x_p \\ y_c = d \cos \phi \sin(\psi + \delta) + y_p \\ z_c = d \sin \phi + z_p \end{cases} \quad (7)$$

ここで、 δ は対象の正面方向とシーン座標系の x 軸とがなす角である (図 12) ． $\psi \cdot \phi$ は任意に与えるものとし (図 11) ． d は式 (6) より得られる値である．また、点 (x_p, y_p, z_p) はカメラの注視点である．注視点の x_p 、 y_p は対象の位置の x 座標、 y 座標と一致させ、 z_p については画面構成に応じて変化させる (式 8) ．例えば、クローズアップ ($r = 0.2$) の場合は $z_p = 0.9h$ となり、注視点は対象の顔の辺りとなる．

$$z_p = h - \frac{h \cdot r}{2} \quad (0 \leq r \leq 1) \quad (8)$$



図 11 カメラと対象の位置関係 (II)
Relation between a virtual camera and an object (II)

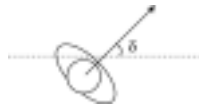


図 12 対象とシーン座標系の関係
Relation between object and scene coordinate systems

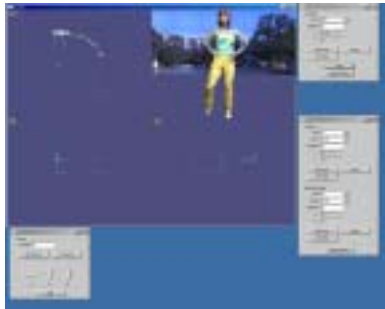


図 13 GUI
GUI

5.3 GUI

本研究で制作した GUI を図 13 に示す。メインウィンドウは、XY, YZ, ZX の各平面と生成された映像を表示する画面の 4 つの画面で構成されている。また、対象や背景の姿勢や大きさ、カメラパラメータを設定・変更するウィンドウや、時刻を制御するウィンドウといったインタフェースを備えている。

手法 (1) に基づくパラメータ設定の際は、任意の時刻において各平面上に対象や背景、カメラの各パラメータを設定していき、これらを時間的に補間することで必要なパラメータを得ることができる。

また、手法 (2) においては、任意の時刻において画面構成を指定し、最終的に時間的に補間することで必要なパラメータを得ることができる。

6. 実験

6.1 テクスチャマッピング

本論文で提案した視点依存頂点ベース法（以下手法 (I) と呼ぶ）の精度を評価するために、従来法である視点独立面ベース法（手法 (II)）との比較を行った。

まず、手法 (II) によるレンダリングの例を示す（図 14）。手法 (II) の欠点である、テクスチャのちらつきによる不均一、不整合部分が多く見られる。

同じデータを基にした手法 (I) によるレンダリングの結果を図 15 に示す。この際、アルゴリズムの step.5 に

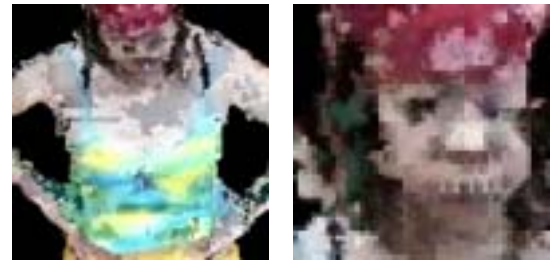


図 14 テクスチャのちらつきの例
Jittering of texture

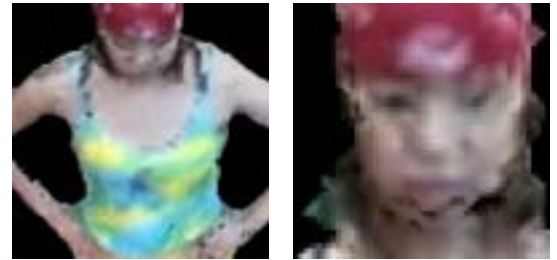


図 15 提案手法によるテクスチャマッピング
Result of our new method of texture mapping

おける重み付け係数は $m = 5$ とした。この値は実験により経験的に得られた値である。提案手法を用いることで、明らかにテクスチャのちらつきが軽減されていることがわかる。

次に、あるカメラ c_j の位置・視線方向を表示視点・視線とし、手法 (I) と手法 (II) によりそれぞれレンダリングを行い、生成された画像シーケンスの RGB 値とカメラ c_j によって撮影された生画像の RGB 値との 2 乗平均平方根誤差を求めた。なお、手法 (I)-1 はカメラ c_j によって撮影された画像を含めて映像化した場合、手法 (I)-2 はカメラ c_j による画像は含めずに映像化した場合である。実験環境は、以下の通りである。

- ・使用カメラ：12 台（図 4）
- ・カメラ画像：640×480[pixel] RGB 各 8bit
- ・復元視点：カメラ 5（図 4）
- ・重み付け係数： $m = 5$

図 16 が示すように、手法 (I)-1、手法 (I)-2 は手法 (II) に比べ、大幅に誤差を軽減している。また、手法 (I)-2 においても手法 (II) より良い結果を示したということは、提案手法の有効性を示していると言える。

また、手法 (II) によりレンダリングされた画像に見られる面パッチ間でのちらつきや不連続性が解消され、より写実性の高い 3D ビデオ映像がレンダリングされることが示されている（図 22）。

さらに、元の面パッチを図 17 のように 3 分割・6 分割と細分化し、提案手法によるさらなる高精細表示の有効性の検証を行った。

まず、細分化前、3 分割、6 分割によって得られた各面パッチを、撮影に用いた各カメラの画像平面上に投影

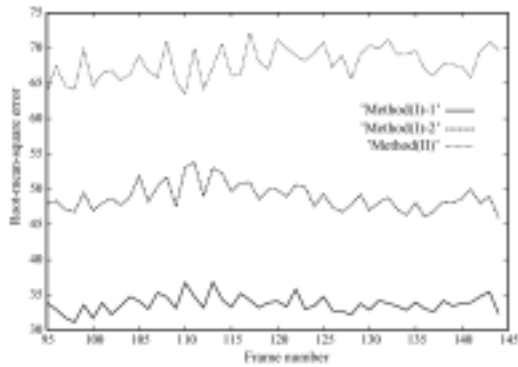


図 16 RGB 値の 2 乗平均平方根誤差 (I)
Root-mean-square error of RGB value (I)

縦軸は画素の RGB 値の 2 乗平均平方根誤差, 横軸は 3 次元ビデオ映像のフレーム番号を表しており, 表示しているデータは上から, 手法 (II), 手法 (I)-2, 手法 (I)-1 による結果となっている。

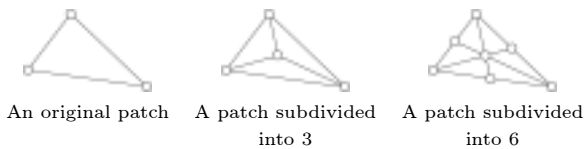


図 17 面パッチの細分化
Subdivision of a surface patch

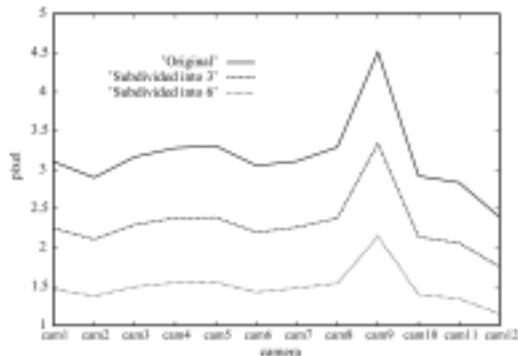


図 18 元画像上における面パッチ 1 辺の平均値 (pixel)
A mean of a side of a surface patch on original images (pixel)

縦軸は投影後の面パッチ 1 辺の平均値 (pixel), 横軸はカメラ番号を表しており, 表示しているデータは上から, 元面パッチ, 3 分割した面パッチ, 6 分割した面パッチを用いた結果を示している。カメラ 9 は対象の真上となる位置に取り付けたカメラで, 対象までの距離が他のカメラより近いいため撮影画像上に対象が大きく写っている。そのため, 他のカメラの値より大きな値が結果として表れている。

し, 画像上における面パッチの 1 辺の長さの平均を求めた (図 18)。

図 18 が示すように 6 分割した場合の面パッチの 1 辺は, ほぼ 1 ピクセルとなっており, これが本撮影システムでの解像度の限界であると言える。この 6 分割した面パッチの実空間での解像度は 1 辺約 5 mm である。

また, 分割した面パッチを用いた映像化の定量的な評価を行うため, 手法 (I)-1 によって映像化した映像 (元面パッチ, 3 分割, 6 分割) を用いて, 図 16 と同様にして 2 乗平均平方根誤差を求めた (図 19)。

この図によると, 面パッチの分割を行っても, 誤差値

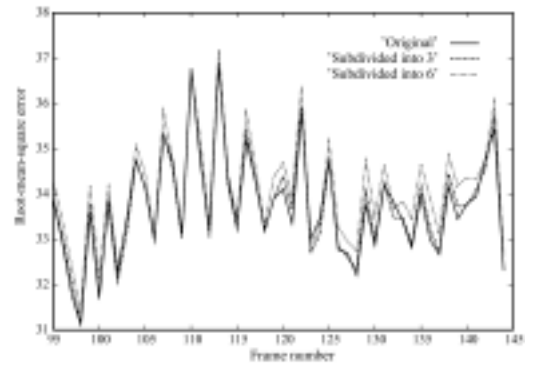


図 19 RGB 値の 2 乗平均平方根誤差 (II)
Root-mean-square error of RGB value (II)

縦軸は画素の RGB 値の 2 乗平均平方根誤差, 横軸は 3 次元ビデオ映像のフレーム番号を表している。



元面パッチ 6 分割

図 20 元面パッチと 6 分割面パッチを用いて映像化した例 (frame #106)

Example images visualized with original patches and subdivided patches (frame #106)



図 21 元画像と生成画像 (6 分割パッチ) の差分画像 (frame #106)

Subtraction between an original image and a generated image (frame #106)

元画像と生成画像 (6 分割パッチ) において, 対象が映っている箇所の RGB 値の差の絶対値の平均を輝度値として生成した。なお, 結果を見やすくするために画像の輝度値を反転させている。

はあまり変わらないという結果になった。しかし, 生成された映像には明らかな違いが現れている (図 20)。

この誤差の主なものは, 対象の輪郭部分やテクスチャの境界部分 (肌と服との境界など) に現れている (図 21)。これらはカメラ間での撮影の時間同期が完全には取れていないことやモーションブラーが原因となって生じているため, 面パッチの分割数を多くしても減少させること

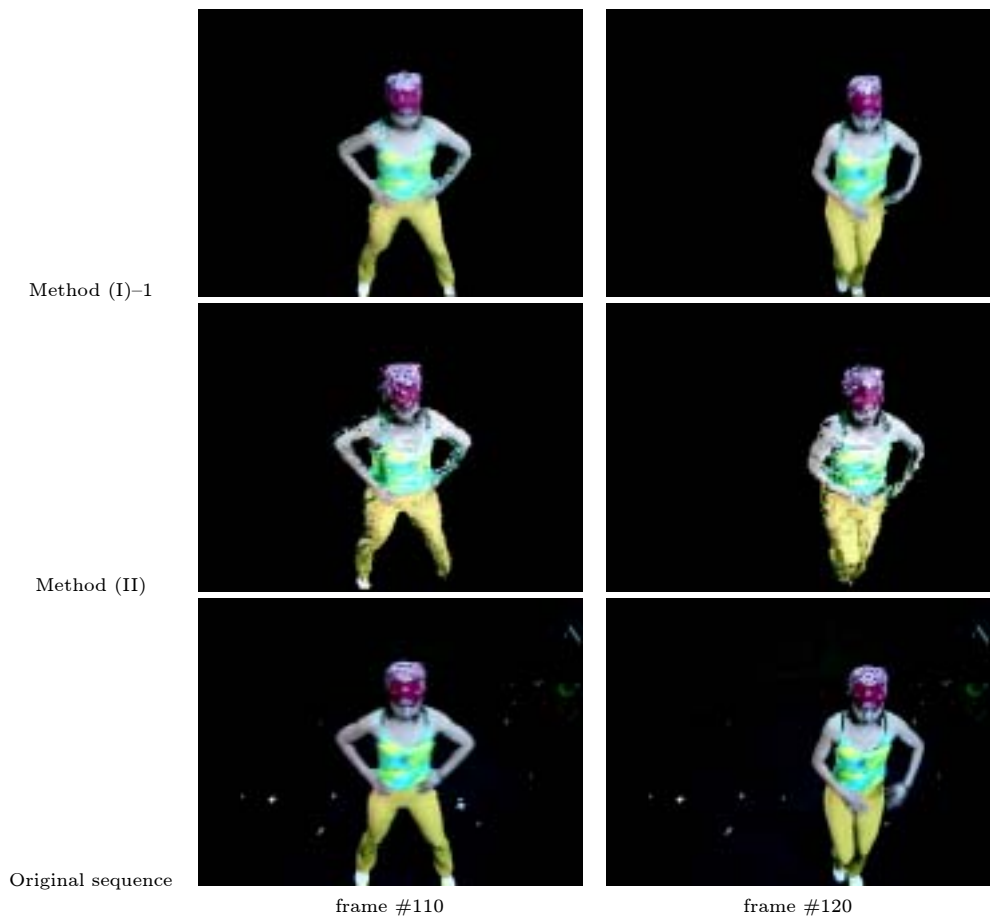


図 22 3次元ビデオ映像の例
Sample frames of 3D video

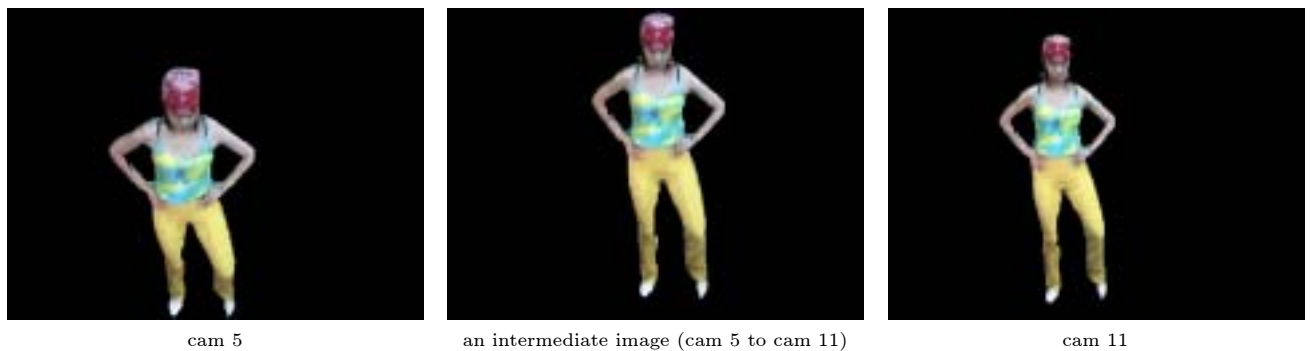


図 23 細分化した面パッチを用いた映像化 (frame #103)
Visualized 3D video with subdivided patches (frame #103)

は困難である。この結果、局部的に誤差が減少している個所があっても、全体的には誤差は減少しないことになり、図 19 の様な結果になったと考えられる。

最後に、6 分割した面パッチを用いてカメラ 5 と 11 およびその中間位置を表示視点として手法 (I)-1 によって映像化した例を図 23 に示す。これにより、任意の表示視点においても、ほぼ撮影された映像と同等の映像が生成できることがわかる。

6.2 3D ビデオ映像の編集

運動対象 (人間) と背景を図 2 のように配置し 3D ビデオ映像の編集・映像化を行った結果を図 24 に示す。そ

の際に、対象の 3 次元形状を利用したカメラワークとして、以下の条件を用いた。

- (1) フルショット。
- (2) 常に対象の正面を撮り続ける。
($\psi = \phi = 0, \theta = 30 \text{ deg.}$)

7. む す び

我々は、多視点映像から運動対象の 3 次元形状とその表面情報を時系列データとして生成し記録する「3 次元ビデオ映像」を提案している。本研究では、必ずしも正確であるとは言えない対象の 3 次元ボクセルデータを基



図 24 全方位背景を持つ 3 次元ビデオ映像
Visualized 3D video with an omnidirectional background

にして、いかに自然な任意視点映像を生成するかという問題に焦点を当てて検討を行い、視点依存頂点ベース法を提案し、視点独立面ベース法と比べ大幅に画質が向上できることを示した。

また、対象の 3 次元形状・運動に基づいたインタラクティブなカメラワーク設定機能を持った 3 次元ビデオ映像編集システムを開発した。これにより、高い自由度を持つパラメータセットを簡単な操作で指定することができ、ユーザの望むカメラワークを低負荷で生成することが可能となった。

今後の展開としては、3 次元ビデオ映像の効率的な圧縮法や、照明環境に不変な対象の表面情報の復元、鑑賞に耐えうる効果的なカメラワークの設定法などが挙げられる。

本研究は、科学研究費補助金 基盤研究 A 13308017 の補助を受けて行った。また、実験に使用したデータは、技術組合 新情報処理開発機構の画像理解評価用画像データベース 2001 に収録されているものである。

〔 文 献 〕

- 1) 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”, 情報学論, 42, SIG 6(CVIM 2), pp.33-43, 2001.
- 2) 和田 俊和, ウ 小軍, 東海 彰吾, 松山 隆司: “3D ビデオ (1): PC クラスタによる身体動作の実時間 3 次元映像化”, The Proc. of IMPS, pp.9-10, 2000.
- 3) 圓藤 康平, 西出 義章, 和田 俊和, 松山 隆司: “3D ビデオ (2): 多視点映像からの 3D 映像の生成と表示”, The Proc. of IMPS, pp.11-12, 2000.
- 4) S. Moezzi, L. Tai, and P. Gerard: “Virtual View Generation for 3D Digital Video”, IEEE Multimedia, pp.18-26, 1997.
- 5) T. Kanade, Peter Rander, and P.J. Narayanan: “Virtualized Reality: Constructing Virtual Worlds from Real Scenes”, IEEE Multimedia, pp.34-47, 1997.
- 6) E. Borovikov and L. Davis: “A Distributed System for Real-time Volume Reconstruction”, Proc. of CAMP2000, pp.183-189, 2000.
- 7) 向川 康博, 中村 裕一, 大田 友一: “2 枚の顔写真を用いた任意方向の顔画像の生成”, 情報学論, 37, 4, pp.635-644, 1996.
- 8) 北原 格, 大田 友一: “多視点映像の融合によるスポーツシーンの自由視点映像生成 - 3 次元形状表現用平面の適応的配置 -”, PRMU2000-189, pp.23-30, 2001.
- 9) 映像情報メディア学会 編: “映像情報メディアハンドブック”, pp.515-519, オーム社 (2000).
- 10) 西野 恒, 佐藤 いまり, 佐藤 洋一, 池内 克史: “Eigen-Texture 法: 複合現実感のための 3 次元モデルに基づく見えの圧縮と合成”, 信学論, J82-D-II, 10, pp.1793-1803
- 11) 富永昌治: “コンピュータビジョンにおけるカラー情報の表現と解析”, コンピュータビジョン (松山, 久野, 井宮編集), pp.64-79, 1998.
- 12) 東海 彰吾, 弓場 竜, ウ 小軍, 松山 隆司: “アクティブカメラを用いた動的広域シーンの映像提示法”, 信学論, J84-D-II, 8, pp.1910-1920, 2001.
- 13) 竹内 英人, 池内 克史: “3 次元モデル再構築のための光源の影響を補正したテクスチャ合成”, 情報学研資 「コンピュータビジョンとイメージメディア」, No.123-6, pp.43-50, 2000.
- 14) 剣持 雪子, 小谷 一孔, 井宮 淳: “点の連結性を考慮したマーキング・キューブ法”, 信学技報, PRMU98-218, pp.197-204, 1999.
- 15) T. Wada and T. Matsuyama: “Appearance Sphere: Background

Model for Pan-Tilt-Zoom Camera”, Proc. of the 13th ICPR, pp.A-718-A-722, 1996.
16) ダニエル・アニボン/岩本 憲児, 出口 丈人 訳: “映画の文法 実作品に見る撮影と編集の技法”, 紀伊国屋書店 (1980)

	<p>たかい たけし 高井 勇志 1998 年, 同志社大学工学部電子工学科卒業。2000 年, 奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。現在, 京都大学大学院情報科学研究科博士後期課程在学中。コンピュータビジョン, コンピュータグラフィックスに関する研究に従事。学生会員。</p>
	<p>まつやま たかし 松山 隆司 1976 年, 京都大学大学院修士課程修了。京都大学助手, 東北大学助教授, 岡山大学教授を経て, 1995 年より, 京都大学大学院電子通信工学専攻教授。現在, 同大学院情報科学研究科知能情報学専攻教授。工博。画像理解, 人工知能, 分散協調視覚の研究に従事。1980 年, 情報処理学会創立 20 周年記念論文賞, 1990 年, 人工知能学会論文賞, 1993 年, 情報処理学会論文賞 1994 年, 電子情報通信学会論文賞, 1995 年, 第 5 回国際コンピュータビジョン会議 Marr Prize, 1996 年, 国際パターン認識連合 Fellow, 1999 年, 電子情報通信学会論文賞, 2000 年, 画像センシングシンポジウム優秀論文賞。</p>