

Harmonized Texture Mapping

高井勇志, 松山隆司

概要

This paper presents a novel texture-mapping method, called harmonized texture mapping (HTM), for reducing the blurring and ghosting artifacts that are created by most existing visualization methods which do not consider the inaccuracy of a generated 3D shape and camera parameters. HTM can significantly reduce the occurrence of artifacts by dynamically adjusting textures generated from multi-viewpoint images in accordance with both textural features of the images and geometrical features of the generated 3D shape. The performance of HTM is evaluated by quantitative and qualitative experiments using several 3D video sequences.

1 まえがき

撮影対象の容姿・動作を完全に記録し自由な視点から鑑賞する事を可能とする3次元ビデオが提案されてほぼ10年が経とうとしている [5, 3]. この間, 多くの機関において本技術に関する研究開発が行われており, 3次元ビデオは, 映像制作, エンタテインメント, 教育, デジタルアーカイブ, 医療など様々な分野での応用が期待されている新たな映像メディアであるといえる.

ここで, 最初に3次元ビデオ生成の流れと各要素技術について俯瞰する. また, 本論文で用いる3次元ビデオやコンピュータグラフィックス (CG) に関する語句について表1にまとめておく. まず, スタジオの壁や天井に配置されたキャリブレーション済のカメラを用いて, 対象の多視点同期撮影を行い, この撮影画像から対象領域を抽出し3次元形状を復元する. この3次元形状は, 頂点群とそれらの連結関係によって表現されたメッシュ構造となっている (以後, 形状メッシュと呼ぶ). 次に, この3次元形状に対して, 撮影画像よりテクスチャを生成することによって, 任意の視点より鑑賞可能な3次元ビデオデータが生成される. このようにして生成した3次元ビデオデータは, フレーム毎に独立したデータ構造となっているため, そのデータ量が膨大となることから, 幾つかの圧縮法が提案されている [13, 16]. 上記の処理を経た後, 通常の2次元画面のみならず, 立体視可能なステレオディスプレイや全周型3次元ディスプレイ [7] といった, 様々な表示装置に応じた映像生成を行うことにより, 我々は自由な視点から3次元ビデオを鑑賞することができる. また, 3次元ビデオの応用として, 生成した3次元ビデオからその構造や運動記述を得る研究 [6, 17] や, 映像制作に用いるプロジェクトも行われてきている [12]. このように, 3次元ビデオの要素技術は, コンピュータビジョンやコンピュータグラフィックス等の多岐の分野に渡っており, 生成方法のみならずこれを応用するための多くの研究がなされてきている.

他の任意視点映像生成法 (CGモデル+モーションキャプチャや単純なイメージベースドレンダリング (IBR)^{*1}) と比較して3次元ビデオの優れている点として下記の例が挙げられる.

- 完全な3次元形状の取得: モーションキャプチャで得られるデータは, マーカの任意時刻における位置

^{*1} 以下本論文では, 3次元形状を復元或は仮定せずに画像のみから任意視点画像を生成する手法を“単純なIBR”と呼ぶ. また, 特に断らない限り“IBR”とは“単純なIBR”を指している.

表1 3次元ビデオやCGに関する語句の定義

- テクスチャ: 撮影画像から生成した形状メッシュ上に貼付ける画像
- テクスチャ座標: テクスチャ上の点と形状メッシュの頂点を対応付ける座標
- 仮想視点・視線: 任意視点画像を生成する際の視点・視線
- 出力画像: 形状メッシュ・テクスチャ・仮想視点を元に描画される画像

であるため、人手を用いて何らかの3次元形状を生成する必要がある。また、単純なIBRでは3次元形状は基本的に用いないため、任意視点画像生成の際にオクルージョンを適切に扱うことは容易ではない。

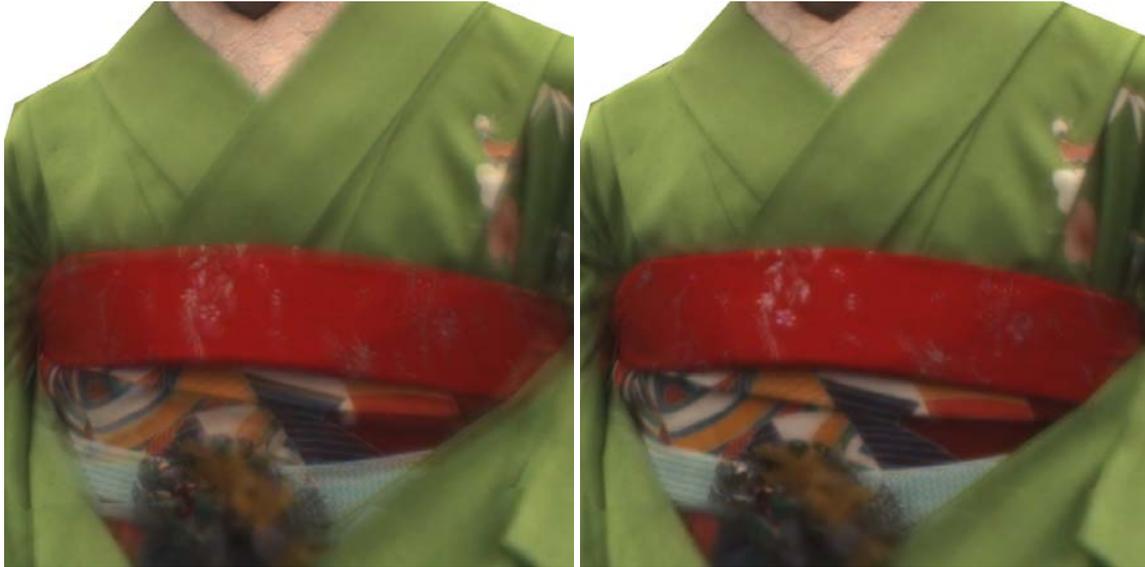
- 衣装を含めた自然な動きの取得: モーションキャプチャでは、基本的にタイトなスーツを着た上に幾つかのマーカをつけて動作を行う必要がある。したがって、実際にその動きを行う環境とは異なる身体拘束が生じるため、理想とする動作が収録できない可能性がある。例えば、着物を着ての立ち回りや伸縮性のある衣装や小道具を利用する場合などが挙げられる。また、IBRにとっても、複雑な動きを伴う衣装などを画像のみの対応から任意視点画像生成することは容易ではない。

3次元ビデオ技術には、以上のような利点があるにもかかわらず、他の映像生成技術と比べるとまだまだ一般的に利用されているとは言い難い。その理由のうち最も大きな点の一つとして、映像化された3次元ビデオの“クオリティ”の低さが挙げられる。現在実現されている多くの方法では、レオタードやTシャツなどタイトな服装の対象については、ある程度高いクオリティの映像を生成することができるが、長い袖やひらひらしたスカートといった複雑な形状を含む対象については、ボケや二重写りが多く生じてしまい、芳しい結果を得ることが困難である。このことが、大きく揺れ動く服装のような、CGなどでは自然な表現が困難な対象を容易に表現可能な3次元ビデオの優れた特徴を、実際の映像制作に活かすことを阻害している一因である。

では、このクオリティ低下の原因—ボケや二重写り—はなぜ生じるのだろうか。3次元ビデオの生成においては、まず何らかの参照物体を用いて、多視点カメラの校正を行っている。実用上においては、撮影可能な全空間に対して完全に正確なカメラパラメータを得ることは非常に困難であり、実際には多少の誤差が含まれてしまう。次に、撮影された多視点画像から対象の形状復元を行うが、形状復元法に起因する誤差をなくすことは困難であり、また、カメラパラメータに誤差があることから、完全に正確な形状を得ることはこれもまた困難である。さらに、テクスチャの生成においては、上記で得られた完全に正確とは言えないカメラパラメータと形状を利用するため、テクスチャについても誤差が含まれていることは明らかである。このように、復元された形状とテクスチャは基本的に真の対象と一致することはない。したがって、この不一致を考慮しない映像化を行うと必然的にボケや二重写りが生じることになる。

以上を踏まえ、本論文ではこの復元された形状（以下形状メッシュと呼ぶ）とメッシュを基に生成したテクスチャ（以下単にテクスチャと呼ぶ）の不一致を考慮した調和テクスチャマッピング法（Harmonized Texture Mapping; HTM法）を提案する。HTM法の基本的な考え方は以下の通りである。

- 完全とは言えないがほぼ正確なカメラパラメータと形状メッシュを得ているものとする。これらを用いて、ボケや二重写りのない画像を生成するために、カメラパラメータ推定や形状復元法の精度の向上を試みるのではなく、多視点画像より生成したテクスチャを適切に変形することによって目的を達成する。
- テクスチャの変形は、多視点画像より生成したテクスチャの画像としての一致度のみではなく、下記の



(a) 単純なテクスチャレンディング法

(b) 提案手法 (HTM)

図1 提案手法の効果

ような形状メッシュの特徴を活かして行う。

- 面単位の一致度評価
- テクスチャの一致度に応じた面の細分割による適応的な変形
- 画像の一致度の探索に用いる初期値や探索範囲などの制約
- 任意視点画像生成時の視線に応じて動的にテクスチャを生成する (視点依存レンダリング)。

以上により、誤差の含まれたカメラパラメタや形状メッシュを用いても、ボケや二重写りのない、撮影画像とほぼ同じ画質の任意視点画像を得ることが可能となる (図1)。さらに本手法は、形状メッシュを単純化することによってそのデータ量を削減し、その後必要な箇所に頂点を追加することによって、テクスチャ整合性の観点から、適切な形状メッシュを再構成を行っている。つまり言い換えると、カメラパラメタや形状復元の誤差を出力画像生成の観点から減少させるようなテクスチャ座標—つまりこれが調和テクスチャ座標である—が得られるように、得られた形状メッシュの最適化を行っているわけである。これは、莫大な量となる3次元ビデオのデータ削減の観点からも効果的であると言える。また、本論文では、インタラクティブな表示を実現するため本手法を DirectX を用いて実装し、リアルタイム (≥ 30 fps) での映像化が可能であることを確認した。

2 関連研究

多視点画像から任意視点画像を生成する方法について、提案手法と関連の深いイメージベースドレンダリング方式および3次元ビデオ方式を概観しその特徴について述べる。また、提案手法と同様にカメラパラメタや形状メッシュの不正確さを考慮し、単純な IBR 方式と3次元ビデオ方式の特徴を活かした方法である Floating textures [2] に触れ、本手法との違いを明らかにする。

単純な IBR 方式は、2枚の画像を用意し、それらの画像に現れる適当な特徴を利用して画像間の対応関係を求め、中間視点の画像を作り出す方法である。単純な IBR 方式はカメラパラメタや3次元形状を考慮してい

ないため、実際には起こりえない変形が生じた中間視点画像を生成することが多々起こる。この問題に対処するため、Seitz と Dyer は、対象形状として単純な平面を当てはめ、カメラパラメタを利用することによって、矛盾のない中間視点画像を生成する方法を提案した [8]。この方法では、画像中で対象の見えが回転するような場合においても、対象形状の不自然な変化を起こさずに中間視点画像を生成することが可能であるが、視点の移動に応じて可視領域が変化する箇所、つまりセルフオクルージョンが生じる箇所については対処不可能である。Einarsson らは、対象を取り囲む多数のカメラを用いてオプティカルフローベースの画像変形による任意視点画像生成を行った [1] が、この方法についてもセルフオクルージョンは考慮されていない。

一方、3次元ビデオ方式は、推定した形状メッシュ上に撮影画像より生成したテクスチャをマッピングすることによって、任意視点画像を生成する方式である [4]。3次元ビデオ方式において画像生成に用いるデータは、通常の 3D CG にて良く用いられる、いわゆるポリゴンメッシュとテクスチャ画像であり、任意視点画像を生成するための視点（仮想視点）のカメラパラメタや座標変換行列などをコントロールして画像を生成する。この3次元ビデオ方式の映像化で問題となるのは、複数撮影画像からどのように表示用のテクスチャを生成するのかという点である。一般に、形状メッシュのある面を観測可能な撮影カメラは複数存在するため、これらから適切な1枚を選び出す、あるいは、ブレンディングといった処理が必要となる。ここで、もしカメラパラメタや形状メッシュが完全に正確であれば、その面に貼られる複数の撮影画像より生成されたテクスチャは完全に一致するため、これらをブレンディングしたとしてもボケや二重写りが生じることはない^{*2}。このような理想的な状況であれば、複数撮影画像から表示用のテクスチャを生成することは比較的容易であり、ある程度正確なカメラパラメタと復元しやすい撮影対象について、球上に展開した形状メッシュに対して、多重解像度のテクスチャブレンディングを行うことによって、継ぎ目のない単一テクスチャを生成する方法が提案されている [9]。しかし、カメラパラメタや形状メッシュが必ずしも正確とはいえない場合や、袖や裾がひらひらした服装のようにセルフオクルージョンが多発する複雑な形状については、複数撮影画像より生成されたテクスチャは一致しておらず、単純に重ね合わせただけではボケや二重写りによる画質の低下が生じてしまう。このような画質の低下に対処するため、仮想視点と撮影カメラの関係に応じた重みを持たせたテクスチャを重ね合わせてボケや二重写りを低減させる方法（視点依存法）が提案されている（視点依存頂点ベース方式 [15]、視点依存面ベース方式 [14]）が、いずれもカメラパラメタや形状メッシュの不正確に起因する画質の低下の問題を積極的に解決するアルゴリズムとはなっていない。

最近になって、IBR 方式と3次元ビデオ方式を適切に組合せ、カメラパラメタや形状メッシュが不正確であることに起因する生成画像のボケや二重写りを軽減するアルゴリズム—Floating textures 法—が提案された [2]。Floating textures 法の基本的な考え方は、“各撮影画像から生成したテクスチャはほぼ一致しているが若干のズレがあるため、このまま出力画像を生成するとボケや二重写りが生じる。そこで、テクスチャを2次元画像として互いに一致するように変形させることによって、このズレを減少させ高品位の出力画像を得ること”である。この基本部分のアルゴリズムを Alg. 1 に示す (図 2)。ここで、対象表面上のセルフオクルージョンによる不可視点は、予めオプティカルフローの計算から除外し、また、その境界付近のテクスチャをぼかすことによって、テクスチャの不連続性を抑制するアルゴリズムとなっている。このように Floating textures 法では、テクスチャの変形によってボケや二重写りのない出力画像を生成することができる。文献 [2] において Eisemann らは、1コマのみの3次元ビデオについて、仮想視点に最も近い3枚の画像のみを適宜用いることによって、5から24 fps で任意視点画像を生成できると述べている。

^{*2} これが成立するのは対象表面が完全拡散反射面と見なせる場合のみであるが、本論文では、“テクスチャの一致”については対象表面が完全拡散反射面であると考えることとする。

Algorithm 1: Basic procedure of Floating Textures.

Specify virtual viewing direction, \vec{V}_{eye} .

foreach camera C_i *in all cameras* **do**

 └ Generate image, I_i^V , at viewpoint V by mapping captured image I_i to mesh M .

Output image, I_{output} is given by

$$I_{output} = \sum_{i=0}^N (W_{I_i^V} \circ I_i^V) w_i, \quad (1)$$

where

$$W_{I_i^V} = \sum_{j=0}^N w_j W_{I_i^V \rightarrow I_j^V}, \quad (2)$$

$W_{I_i^V} \circ I_i^V$ denotes image warping with computed optical flow $W_{I_i^V}$, and $W_{I_i^V \rightarrow I_j^V}$ denotes the optical flow from I_i^V to I_j^V . w_i is a weighting factor given by the viewing directions of the viewpoint and the camera.

Floating textures 法と比較した上での提案手法の主張点は、

1. 完全に正確であるとは言えないカメラパラメータや形状メッシュに対しても、撮影画像と同等の画質の出力画像が得られるテクスチャマッピング法
2. テクスチャ整合性に基づく形状メッシュの最適化法
3. 3次元ビデオシーケンスのリアルタイム映像生成法

の3点である。1については基本的な考え方は提案手法と Floating textures 法は同じであるが、2については提案手法のみの主張点となっており、3については Floating textures 法は1コマの映像生成のみ実験を行っている。

1においてキーとなるオプティカルフローの取得については、Floating texture 法は既存手法を用い、セルフオクルージョンを考慮する以外は形状メッシュを積極的に利用するものとはなっていない。このため、求めたオプティカルフローが妥当なものかどうかの検証が、そのアルゴリズム中では不可能である。一方、提案手法では、形状メッシュの制約（対応点探索の初期位置や範囲）を元にオプティカルフローを求め、その結果を用いてテクスチャ整合性を評価し、必要な箇所については面の細分割を行って再探索を行うといった、より安定にオプティカルフローを求めるアルゴリズムとなっている。

2については、前書きにて述べた通り、提案手法は、カメラパラメータや形状復元の誤差を出力画像生成の観点から減少させるようなテクスチャ座標が得られるように、得られた形状メッシュを最適化している。膨大なデータ量となる3次元ビデオについて、このような観点からのデータ削減法は他にはないと考えられる。

3については、提案手法では計算コストのかかるオプティカルフロー計算は予め行っているため、効率的な映像化が可能である。Floating textures 法では、画素ベースのオプティカルフロー計算を行っているため、任意視点におけるオプティカルフローを予め計算しておくことは実用上不可能である。そこで、彼らはGPUを用いた高速な計算法を用いてオプティカルフローを求めているが、仮想視線に最も近い3台のカメラの画像を用いるという制限の下でも5から24 fpsの映像生成となっている。一方、提案手法では15台のカメラの撮影画像すべてを用いても30 fps以上の表示が可能であるため、この点についても提案手法の優位性が示されて

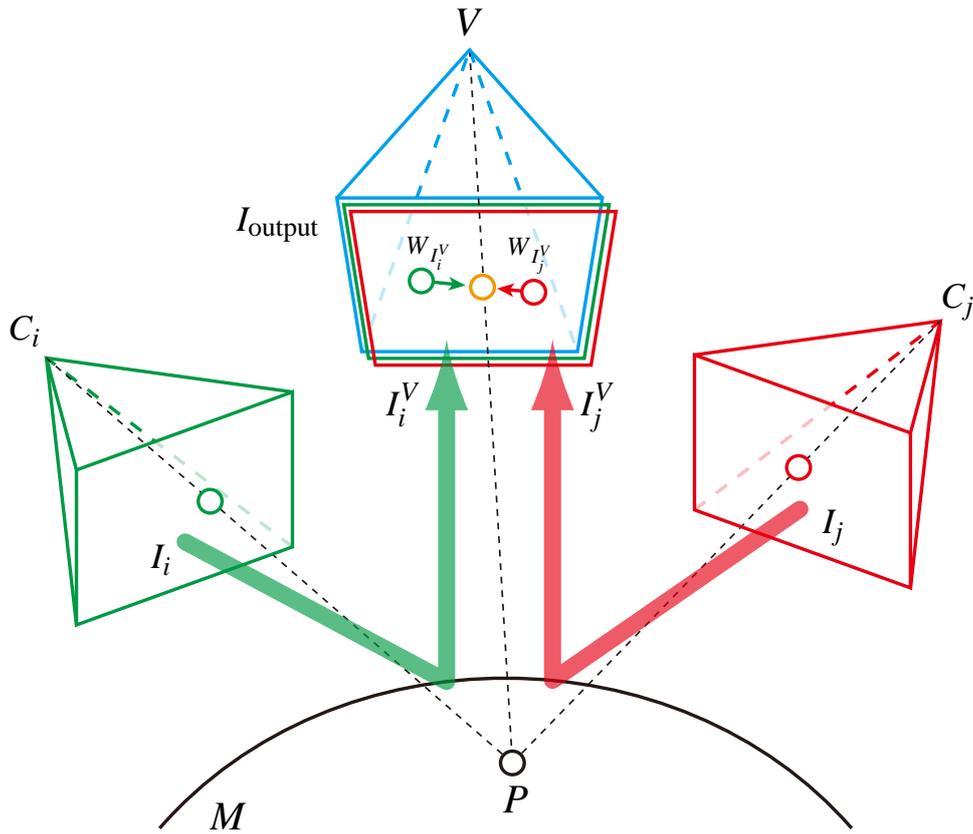


図2 Floating textures 概念図

いる。

3 提案手法

本手法の目的は、完全に正確であるとは言えないカメラパラメータや対象の形状メッシュを基に、撮影画像と同等の画質を持つ任意視点画像を作り出すことである。まえがきにて述べたように、本手法では、撮影画像や形状メッシュを直接変形させるのではなく、テクスチャ座標を仮想視点に応じて動的にコントロールしてテクスチャの画像特徴が互いに一致するように変形し、ポケや二重写りのない出力画像の生成を行う。本節で用いる語句の説明および処理の流れは以下の通りである（表2、図3）。

0. **形状メッシュの単純化**：以下の処理（1）における複数撮影画像間における投影頂点間対応点の取得では、投影メッシュの各頂点周囲の画像のマッチングを coarse to fine 戦略に基づいて行う。つまり、形状メッシュを予め単純化しておき必要に応じてメッシュの細分割を行うことによって、形状メッシュのトポロジーの変化を抑制しつつ大局的な対応点を得られることが期待される。またこの処理によって、形状メッシュのデータサイズを大幅に減少させることができる。

一方、単純化によって形状メッシュの誤差が新たに発生することになるが、新たな頂点を加えつつ面ベースでのテクスチャの整合性を評価しているため、結果的に、その誤差の修正も含めた調和テクスチャ座標の計算がなされることになる。

表2 語句の定義

- カメラ, $C_i, \{i = 1, 2, \dots, n_c\}, n_c$ = カメラ台数: スタジオに設置したカメラを表す. また, カメラ C_i によって撮影された画像を I_i とする.
- 形状メッシュ, M : 撮影対象の3次元形状を, 頂点 ($v_v \in \mathbb{R}^3, \{v = 1, 2, \dots, n_v\}, n_v$ = 頂点数) とその連結関係によるメッシュ表現としたもの. 本論文で用いる形状メッシュの面 f はすべて三角形であるとする.
- 投影メッシュ, $\hat{M}_i, \{i = 1, 2, \dots, n_c\}$: M を C_i に投影したメッシュ. M の持つ頂点連結情報と, 投影された各頂点座標 (投影頂点; $\hat{v}_v^{\hat{M}_i} \in \mathbb{R}^2$) および C_i からの各頂点の可視情報を持つ.
- 小領域画像 (面), $I_{\hat{M}_i \leftarrow \hat{M}_j}^f$: I_j 上における形状メッシュの面 f の画像を, M と C_i および C_j のカメラパラメータを用いて I_i 上に投影したもの.
- 小領域画像 (頂点), $I_{\hat{M}_i \leftarrow \hat{M}_j}^v$: I_j より切り出され I_i 上に投影された, 頂点 v を共有するすべての面の小領域画像 (面) の集合.
- 画像のマッチング: 2枚の画像が与えられたとき, 式 (3) によって画像の一致度を評価する. ある1枚の画像の位置を少しずつずらしながら評価を繰り返し, 最も低い評価値が出る位置を探し出すことを画像のマッチングと呼び, その位置をマッチングの取れた点と呼ぶ.

$$\|I_i - I_j\| = \frac{1}{3n_p} \sum_y \sum_x \sum_{c \in \{R, G, B\}} \|I_i^{x,y,c} - I_j^{x,y,c}\| \quad (3)$$

ここで $I^{x,y,c}$ は画像 I 上の点 (x, y) の輝度値 c を表し, n_p は比較する画素の総数を表す.

1. **投影頂点間対応 (projected vertices correspondence; PVC) の取得 (3.1 節)**: 形状メッシュの各小領域画像 (頂点) を各撮影画像上へ投影し, その周囲の画像間でのマッチングを行うことによって投影頂点間対応 (PVC) を得る. ここまでの処理はオフラインで行い, この PVC を用いて実時間での出力画像生成を以下の処理によって実現する.
2. **調和テクスチャ座標 (harmonized texture coordinates; HTC) の計算 (3.2 節)**: 前処理によって得られた PVC を基に, 仮想視線と撮影カメラ視線の関係により得られた重みを考慮して, 動的に各画像におけるテクスチャ座標 (HTC) を求める. HTC を用いて生成した各テクスチャの画像特徴は互いに一致しているため, これらを用いるとボケや二重写りのない出力画像が得られることになる.
3. **テクスチャ生成 (3.3 節)**: HTC に基づいて各撮影画像よりテクスチャを生成する.
4. **出力画像生成 (3.4 節)**: 生成したテクスチャを, 仮想視線と撮影カメラ視線の関係により得られた重みに基づいてブレンディングし出力画像を得る.

以下本節では, 本アルゴリズムの各ステップについて詳細を述べる. 3次元ビデオデータは各時刻において独立して生成されるため, 以下の説明では時刻を表す係数は省略する.

3.1 投影頂点間対応 (PVC) の取得

画像 I_j より切り出された小領域画像 (頂点) $I_{\hat{M}_i \leftarrow \hat{M}_j}^v$ と, 投影頂点 $\hat{v}_v^{\hat{M}_i}$ の周囲で最も良くマッチングの取れた画像 I_i 上の点を, 投影頂点間対応 $\hat{v}_v^{\hat{M}_i \leftarrow \hat{M}_j}$ と定義する. もし, カメラパラメータや形状メッシュが完全に正確

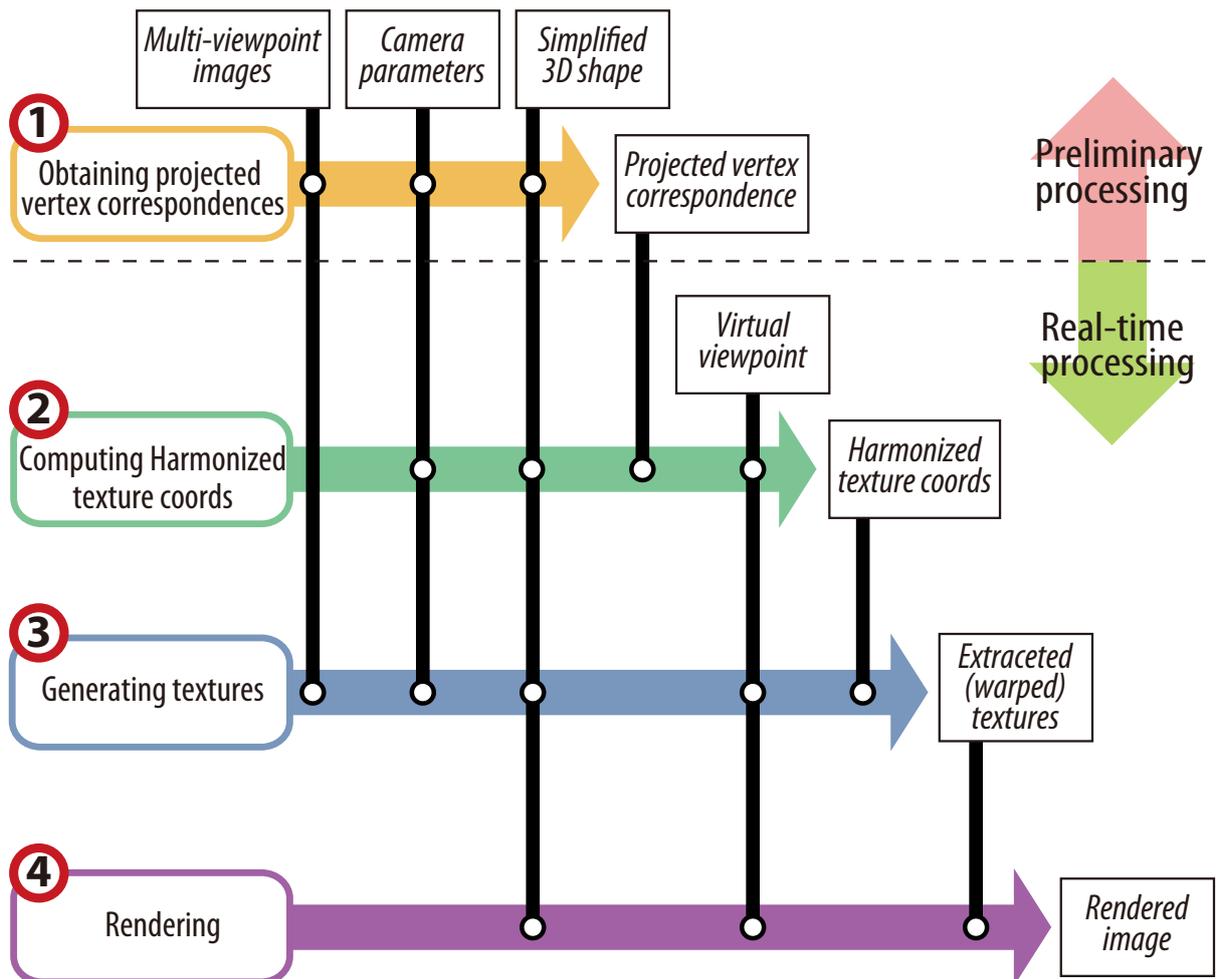


図3 Harmonized texture mapping の概要

であれば、 $\hat{v}_v^{\hat{M}_i \leftarrow \hat{M}_j}$ は $\hat{v}_v^{\hat{M}_i}$ と完全に一致することになるが、完全に正確であるとは言えないカメラパラメータや形状メッシュを用いるとこれらは一致せず、 $\hat{v}_v^{\hat{M}_i \leftarrow \hat{M}_j}$ は $\hat{v}_v^{\hat{M}_i}$ の周囲に分布することになる (図4)。このように、純粋な幾何学計算によって得られるテクスチャ座標周辺の画像特徴が互いに一致しないことが、出力画像にボケや二重写りを生じさせる原因である。したがって、このような不正確なカメラパラメータや形状メッシュから一致したテクスチャを生成し、さらに撮影画像と同等の画質を持つ出力画像を得るには、視点の移動に応じて、投影頂点間対応を用いてテクスチャ座標を適切にコントロールすれば良いことがわかる。

投影頂点間対応は Alg. 2 に示す3段階のアプローチによって、coarse to fine 戦略に基づいた反復計算によって求められる。以下では各ステップの詳細について述べていく。

3.1.1 投影頂点間対応の初期値の取得

すべての投影メッシュのすべての投影頂点について、小領域画像 (頂点) と撮影画像のマッチングを基に投影頂点間対応点を求める (Alg. 3)。この探索の際、投影メッシュにねじれが生じないように探索範囲はその頂点を含むエッジの最小値の半分とする。投影頂点間対応はその定義から、形状メッシュの頂点数 × 撮影カメラの台数 × 撮影カメラの台数分のデータとなる。

Algorithm 2: Flow of obtaining projected vertices correspondence.

repeat

 Obtaining PVC (see Alg. 3)

 Computing differences of extracted textures with PVC for evaluation (see Alg. 5)

 continue \leftarrow Evaluation and remeshing (see Alg. 7)

until continue is true

Algorithm 3: Obtaining PVC.

$\hat{M} = \{\hat{M}_i | i = 1, 2, \dots, n\} \leftarrow$ Compute projected meshes with M and a set of C

foreach projected mesh \hat{M}_i in \hat{M} **do**

foreach vertex v in M **do**

foreach projected mesh \hat{M}_j in \hat{M} **do**

$\hat{v}_{v \leftarrow \hat{M}_j}^{\hat{M}_i} \leftarrow$ Search corresponding point with template matching by projecting image tip $\mathcal{I}_{\hat{M}_i \leftarrow \hat{M}_j}^v$ to

I_j

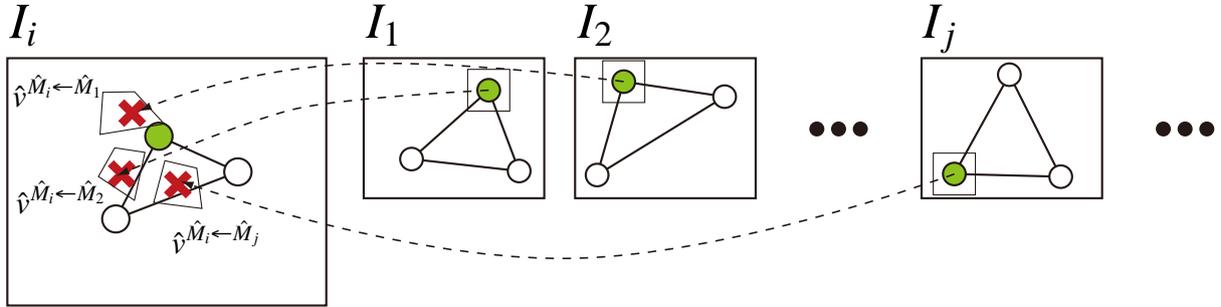


図4 投影頂点間対応点

Algorithm 4: Obtaining warped image tip of face.

Suppose the vertices of f as $\{v_1, v_2, v_3\}$, adjusted texture coordinates of f is given by $\hat{v}_{v_k}^{\hat{M}_i} = \hat{v}_{v_k}^{\hat{M}_i \leftarrow \hat{M}_j}$, $k = 1, 2, 3$.

With these texture coordinates, we obtain warped image tip of face, $\mathcal{I}_{\hat{M}_i \leftarrow \hat{M}_j}^{f'}$.

3.1.2 投影頂点間対応の初期値の評価値計算

前節によって求められた投影頂点間対応の初期値を、それらを用いて生成した小領域画像（面）—以下これを補正済小領域画像（面） $\mathcal{I}_{\hat{M}_i \leftarrow \hat{M}_j}^{f'}$ と呼ぶ—と撮影画像との差を用いて評価する（3.1.3節）。補正済小領域画像（面）は Alg. 4 によって生成され、撮影画像との差、つまり投影頂点間対応の評価値は Alg. 5 によって求められる（図5）^{*3}。

^{*3} 重み係数 α および γ はそれぞれ経験的に決めた値である 0.3, 10 とした。

Algorithm 5: Computing differences of textures of each face f .

foreach face f in M **do**

 # We define a best camera as a camera that captures face f from its front and larger.

 Find best camera C_\star that has the highest value of e defined by

$$e_k = \alpha \bar{d}_{C_k} + (1 - \alpha) \bar{s}_f, \quad (4)$$

 where

$$\bar{d}_{C_k} = \frac{d_{C_k}}{\sum_i^C d_{C_i}}, \quad d_{C_k} = \left(\frac{\vec{V}_{C_k} \cdot \vec{N}_f + 1}{2} \right)^\gamma, \quad \bar{s}_f = \frac{s_f}{\sum_i^C s_{f_i}}, \quad (5)$$

- \vec{V}_{C_k} : Viewing vector of camera C_k ,
- \vec{N}_f : Surface normal of face f ,
- s_f : Size of face f , and
- α, γ : Weighting coefficient.

 Compute differences by

$$E_f = \frac{1}{n_C - 1} \sum_{k, k \neq \star} e_k \|\mathcal{I}_{\hat{M}_k \leftarrow \hat{M}_\star}^{f'} - I_k\|, \quad (6)$$

 where

- n_C : Number of cameras in which the face is projected, and,
 - $\|\mathcal{I}_{\hat{M}_k \leftarrow \hat{M}_\star}^{f'} - I_k\|$: Difference computed by Eq. (3).
-

3.1.3 投影頂点間対応点の評価とメッシュの細分化

前節で得られた評価値は、その値が小さいほど良い投影頂点間対応点を得られていることを表している。本手法で扱うカメラパラメータと形状メッシュは、完全とはいえないがおおよそ正確であるため、対象の大部分については良い評価値を得ることができる。つまり、上記によって得られた投影頂点間対応点を用いれば、対象上の大部分においてはボケや二重写りのない出力画像を得ることができる。問題となるのは評価値の悪い箇所であるが、この多くは真の対象形状が凹の部分である。なぜならば、このような箇所はもしカメラパラメータが正確あったとしても、3次元ビデオ生成にて多く用いられるシルエットベース法では、原理的に復元不可能であるからである。また、ステレオ法的な拘束を加えることによって、形状復元の精度を向上させる方法もあるが、正確とはいえないカメラパラメータの下で高精度に行うことは容易ではない。また、形状メッシュの単純化によっても多少の不正確さが生じることになる。さらに、真の対象形状が凹の箇所では、投影頂点間対応点を求めるために用いた小領域画像（頂点）が一致していても、補正済小領域画像（面）が一致しない場合がある（図6）。

このような評価値の悪い面については、その面を細分割して繰り返し Alg. 3 を行うことによって、テクスチャの一致度を向上させる。つまり、単純な線形変換で合わせることのできない面については、これを細分割して処理を繰り返すことによって、結果的に元の撮影画像に対して非線形な変換をかけることになり、テクス

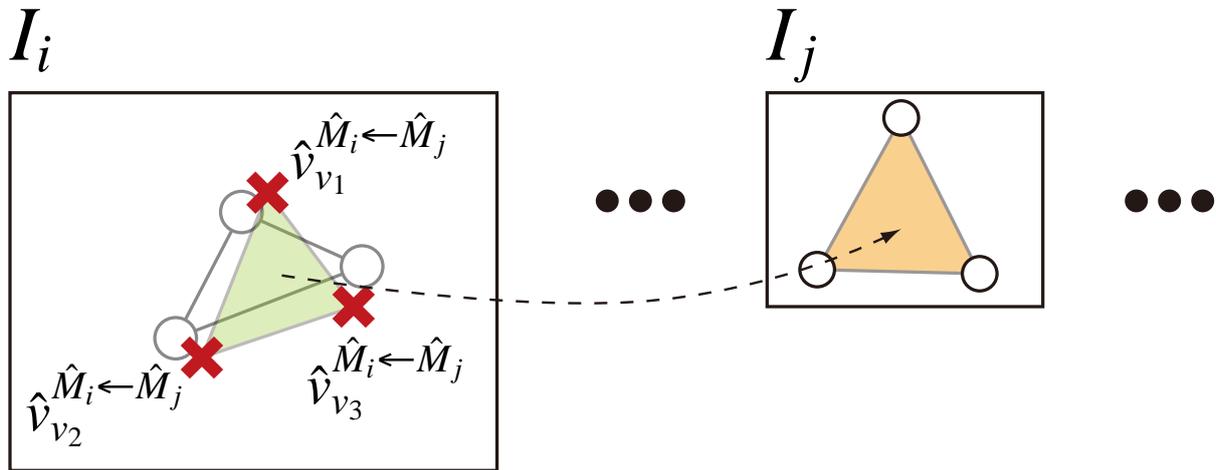


図5 評価値計算のための補正済小領域画像(面)の投影

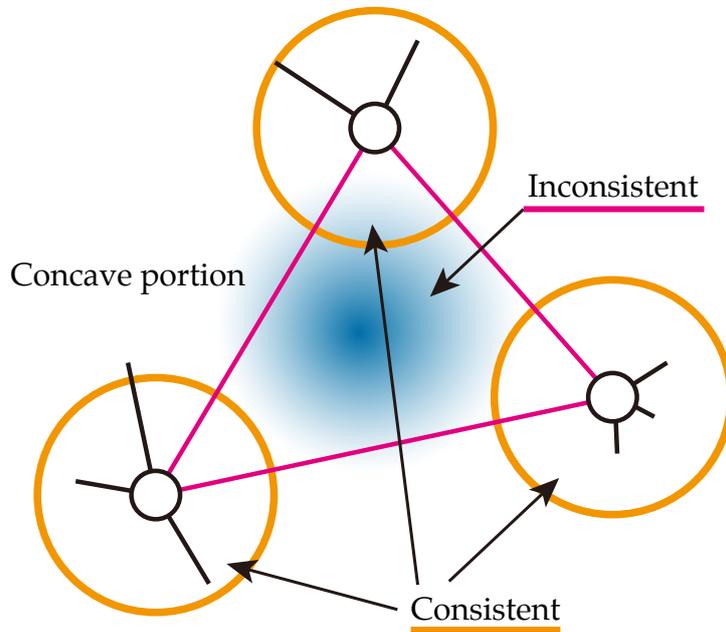


図6 凹面で起こるテクスチャの不一致

チャの一致度を向上させることができる、ということとなる。

面の細分割は、各辺の中点に新たな頂点を加えて四分分割することによって行う。このとき分割する面に隣接する面が三角形となるように、上記の評価値に基づいて分割する面を決定したあと、形状メッシュのすべての面を図7のように五つのタイプに分けて処理を行う (Alg. 6)。また、テクスチャ座標の計算は各カメラにおいて可視である頂点にのみ行われるため、図8に示すようなセルフオクルージョンがある場合*4はテクスチャが貼られない面が現れる。このような面についても上記の細分割を行うことによって、最低限のテクスチャが

*4 ある面において、少なくとも一つの頂点が少なくとも1台のカメラから可視であって、かつ、すべての頂点が可視であるカメラがない場合。

Algorithm 6: Subdivision of Faces in a Mesh.

```
foreach face  $f$  in  $M$  do
  if  $E_f > th$  then
    typeOfFace $_f \leftarrow 4$ 

foreach face  $f$  in  $M$  do
  if typeOfFace $_f \neq 4$  then
    typeOfFace $_f \leftarrow 0$ 
    foreach adjacent face  $af$  of  $f$  do
      if typeOfFace $_{af} = 4$  then
        typeOfFace $_f \leftarrow \text{typeOfFace}_f + 1$ 

foreach face  $f$  in  $M$  do
  if typeOfFace $_f > 0$  then
    Subdivide face  $f$  as shown in 図 7 by typeOfFace $_f$ .
```

Algorithm 7: Evaluation and Remeshing.

```
# Evaluate differences  $E = \{E_f | f = 1, 2, \dots, m\}$ 
if average( $E$ ) > threshold or Requires subdivision for self-occluded patches then
  Subdivide faces of  $M$  (See Alg. 6)
  continue  $\leftarrow$  true
else
  continue  $\leftarrow$  false
```

生成されるようにする。

細分割の反復処理は、誤差の大きい面の面積が充分小さくなったとき、あるいは、セルフオクルージョンによる分割の必要がなくなったときに終了する (Alg. 7)。セルフオクルージョンによる分割は、誤差に基づく細分割の反復計算を行う前に予め行っておくことも考えられるが、本手法では coarse to fine 戦略に基づいてテクスチャの一致度を高めていくため、誤差に基づく反復計算と同時に行うこととする。

3.2 調和テクスチャ座標の計算

前節によって求められた投影頂点間対応点 (PVC) を用いて、仮想視点の移動に応じた動的な調和テクスチャ座標の計算を行う (Alg. 8, 図 9)。これによって、仮想視点に応じて各撮影画像から生成されるテクスチャは動的に変形し、結果として互いに一致することになる。仮想視線とカメラ視線が近い (あるいは一致する) 場合において、そのカメラにて撮影された画像を大きく変形させることなくテクスチャを生成するため、重みの調整係数として γ^{*5} を用いている (式 (7))。ある程度大きな値の γ を用いると、仮想視線とカメラ視線が一致する場合は、式 (8) 中の正規化された重みは、そのカメラについてほぼ 1 となり他は 0 となる。この結果、そのカメラの撮影画像から生成されるテクスチャは元の撮影画像そのものとなる。

*5 γ には経験的に決定した値である 50 を用いた。

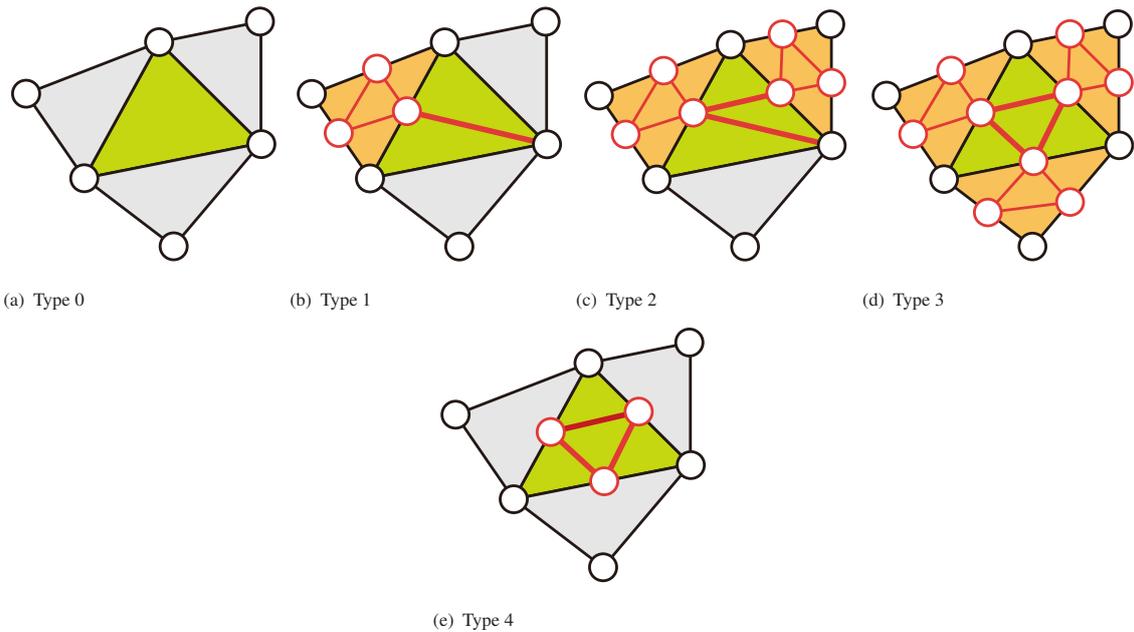


図7 面の細分化

中央の面（緑）に注目する．この面に隣接する三つの面の内，分割対象になっている面数によって注目面を分割し，形状メッシュのすべての面が三角形となるようにする．Type 0 は分割なし，Type 4 は注目面自身が分割対象面であることを表している．黒丸が分割によって生じた頂点を表している．

3.3 テクスチャ生成

前節で得られた調和テクスチャ座標を基にして各撮影画像からテクスチャを生成する．図 10 に生成されたテクスチャの例を示す．この図は，各撮影画像毎に調和テクスチャ座標を基に切り出したテクスチャを，仮想視点から見た形状メッシュに貼付けたものである．上段左からカメラ 1 から 5，中段・下段も同様にそれぞれカメラ 6 から 10，カメラ 11 から 15 の撮影画像より生成したテクスチャである．

3.4 出力画像生成

前節により生成されたテクスチャを仮想視線と撮影カメラ視線の関係により得られた重みに基づいて，出力画像の各ピクセルごとにブレンディングして出力画像を生成する (Alg. 9)．

4 評価実験

提案手法の有効性を示すために，定量的・定性的評価と 3次元ビデオシーケンス（複数フレーム）の再生について検証を行った．4.1, 4.2, 4.3 については“舞妓データ”を，4.4 については“舞妓データ”および“カンフーデータ”を用いて実験を行った．スタジオの設定や各データの詳細は表 3 に示す通りである．

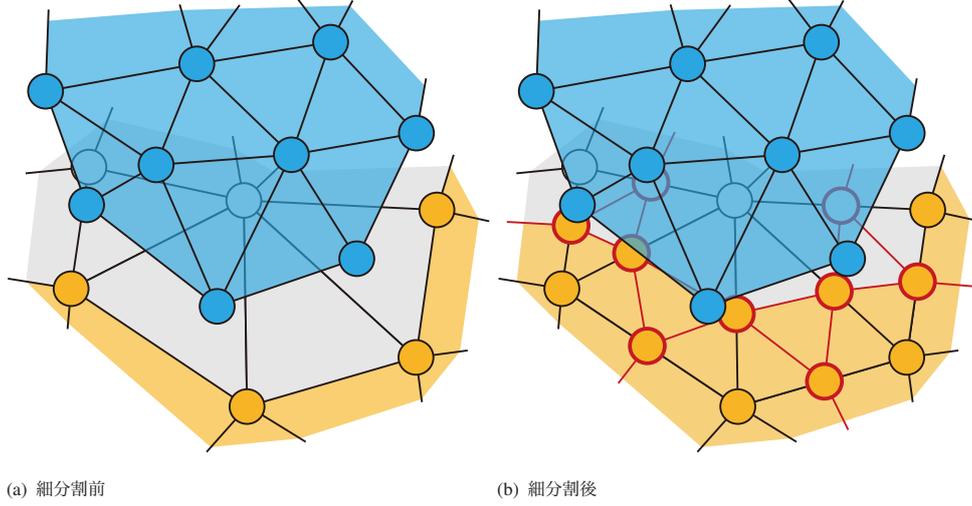


図8 細分割を用いたセルフオクルージョンによる未テクスチャ面に対するテクスチャ座標の生成

Algorithm 8: Computing harmonized texture coordinates with PVC.

Set viewing direction, \vec{V}_{eye} .

foreach projected mesh \hat{M}_i in \hat{M} **do**

 Compute weighting factor by

$$w_{\hat{M}_i} = \left(\frac{\vec{V}_{eye} \cdot \vec{V}_{C_i} + 1}{2} \right)^\gamma, \quad (7)$$

 where γ denotes a weighting coefficient.

foreach vertex v in M **do**

 Compute point for extracting texture, i.e., harmonized texture coordinates, $\hat{v}_v^{\hat{M}_i}$, with the weighting factors and the PVC by

$$\hat{v}_v^{\hat{M}_i} = \sum_j \frac{w_{\hat{M}_j}}{\sum_k w_{\hat{M}_k}} \hat{v}_v^{\hat{M}_i \leftarrow \hat{M}_j}. \quad (8)$$

4.1 提案手法の有効範囲

本論文にて扱う“不正確さ”は、カメラキャリブレーションの精度と形状復元の精度の両方によって生じたものである。カメラパラメタについては、キャリブレーション時の誤差の評価値を参考とすることが可能であるが、形状復元については対象形状や姿勢に応じて復元精度が変化するため、事前にどの程度の誤差が生じるかを予測することは容易ではない。

したがって、この不正確さ（推定誤差 d と呼ぶ）については、上記の両誤差を区別せずにまとめて扱うこととし、Alg. 3 によって求めた投影頂点間対応を用いて以下のように定義する。

$$d_v^{\hat{M}_i \leftarrow \hat{M}_j} = \left\| S_i(\hat{v}_v^{\hat{M}_i}) - S_i(\hat{v}_v^{\hat{M}_i \leftarrow \hat{M}_j}) \right\| \quad (10)$$

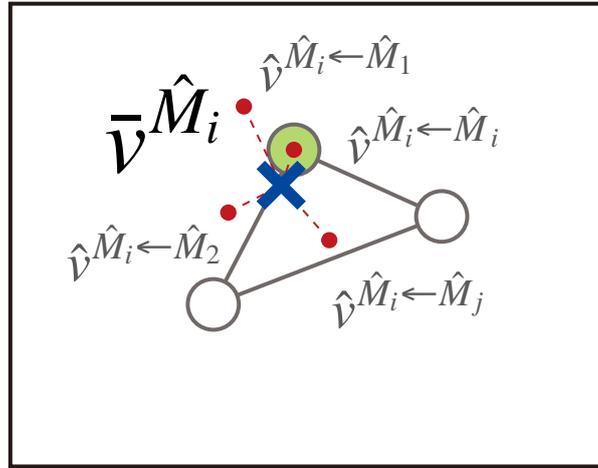
I_i 

図9 調和テクスチャ座標の計算



図10 生成されたテクスチャの例

ここで、 $S_i(v)$ はカメラ i の撮影画像上の点 v とそのカメラの光学中心を結ぶ直線が、3次元空間中の対象メッシュ表面と交わる点を返す関数である。つまり d は、画像上の特徴から推定した一致すべき2点間の、3次元空間中でのユークリッド距離を表している。したがって、Alg. 3 によって対応可能なこの d の最大値が、提案手法によって“改善可能な誤差”の最大値である。ある頂点における改善可能な誤差は、Alg. 3 の探索範囲の定義より、その頂点を共有するエッジの最小値の半分となる。

舞妓データについて、この改善可能な誤差の平均、標準偏差、最大値を求めた結果を表4に示す。このように形状メッシュの単純化レベルによって、改善可能な誤差の程度が変化することがわかる。ここで、面数1,500の形状メッシュに対して提案手法を適用した際に、式(10)より求めた各頂点の推定誤差 d の平均値は

Algorithm 9: Output image rendering.

foreach pixel p in a rendered image **do**

foreach projected mesh \hat{M}_i in \hat{M} **do**

if $T_{\hat{M}_i}^p.Alpha = 0$ **then**

$\bar{w}_{\hat{M}_i} \leftarrow 0$

else

$\bar{w}_{\hat{M}_i} \leftarrow w_{\hat{M}_i}$

 Compute color by

$$Col^p = \sum_i \frac{\bar{w}_{\hat{M}_i}}{\sum_j \bar{w}_{\hat{M}_j}} T_{\hat{M}_i}^p, \quad (9)$$

 where $T_{\hat{M}_i}^p$ denotes a color value for pixel p extracted from I_i with $\bar{\mathbf{r}}_v^{\hat{M}_i}$.

表3 実験環境

- スタジオ：直径4 m, 高さ2.4 mの十二角柱型 (図11).
- カメラ：上記スタジオに15台のカメラ (Sony XCD-X710CR; XGA, 25 fps) を設置した (図11). カメラ13のみ顔を詳細撮影するため焦点距離の長いレンズを装着している. 図12は撮影画像の例を示しており, 上段左からカメラ1から5, 中段・下段も同様にそれぞれカメラ6から10, カメラ11から15によって撮影された画像である.
- 舞妓データ：舞妓の衣装を着た女性. 両脇や腹部については凹形状となっており形状メッシュは完全には正しく復元されていない. また, だらりの帯についても, 実際より厚みのある形状に復元されているため, 各テクスチャ画像において文様は完全には一致していない. 形状メッシュは, 空間解像度5 mmのボクセル群より生成した形状メッシュ (頂点数: 107,971, 面数: 215,946) を752頂点, 1,500面まで単純化したものを用いた. 単純化のレベル (面数) については, 単純化後のメッシュを目視で確認し経験的に決めた値である. 以下, カンフーデータも同様である. 単純化法は, edge collapse をベースとしたソフトウェア [10] を用いた.
- カンフーデータ：カンフーの演技を行う男性. 舞妓に比べて衣装は単純である. 形状メッシュは空間解像度10mmのデータを舞妓データと同様の方法で単純化したものである. 評価に用いたデータは, 全部で116フレームであり, 平均の頂点数, 面数はそれぞれ, 単純化前: 約24,000, 約50,000, 単純化後: 約1,200, 約2,500である.

16.3 mm, 標準偏差は9.77 mmであった. つまり, 提案手法においてこれだけの誤差が改善されたことになる.

ここで単純化レベルをより大きくした場合 (つまり面数をより削減した場合) は, 単純化による形状メッシュの誤差が大きくなりすぎて, 調和テクスチャマッピングにて改善しきれない場合がありうる. そこで, 表4と式(10)の値を元に適切な単純化レベルを決定する方法が考えられるが, これについては今後の検討課題としたい.

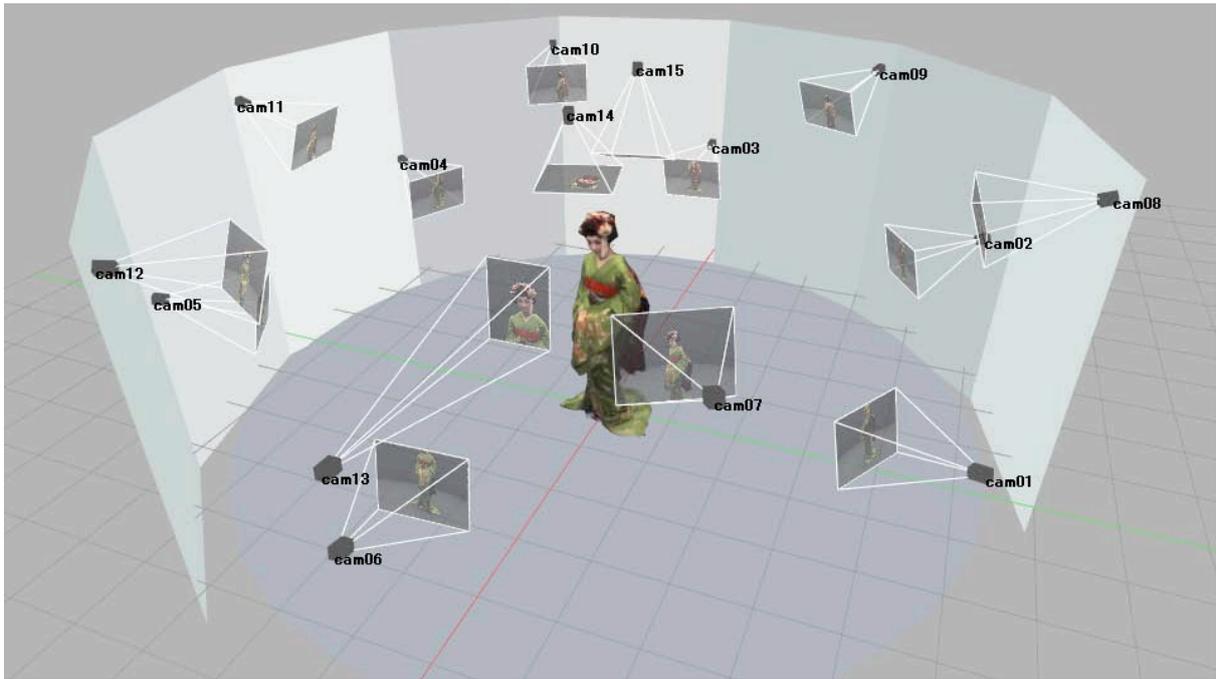


図 11 3次元ビデオ撮影スタジオ模式図



図 12 撮影画像例

4.2 定量評価

本評価においては、面の細分割によってテクスチャー一致度が向上していることを検証し、その結果、出力画像の画質が向上していることを示す。

表4 各単純化レベルにおける改善可能な誤差の例 (mm)

面数	平均値	標準偏差	最大値
500	35.9	9.88	62.1
1,500	21.3	5.82	51.1
5,000	11.8	3.31	23.8
10,000	8.27	2.38	19.7

4.2.1 面の細分割によるテクスチャー一致度の向上

面の細分割の反復処理によって、形状メッシュの面数が増加し、テクスチャー一致度が向上している様子を図13、図14に示す。一致度の向上度の計算は以下のようにして行った。

1. 投影頂点間対応を求める前に、すべての面について式(6)より不一致度を求めその平均を計算する。これを $E_{\text{all}}^{\text{pre}}$ とする。
2. 投影頂点間対応を求めた後に、上記と同様にして不一致度の平均を求める。これを $E_{\text{all}}^{\text{post}}$ とする。
3. 一致度の向上度を $E_{\text{all}}^{\text{pre}} - E_{\text{all}}^{\text{post}}$ として計算する。

また、図14(g)-(m)は、各面の不一致度を可視化したものであり、明度が明るくなるにつれて誤差値が大きくなっていることを示している。

図14(a)-(f)を見ると、脇の部分や袖、裾といった箇所において細分割が行われていることがわかる。これらの箇所は、凹の部分であったり、上方から撮影しているカメラ(14, 15)のoccluding contourであるため、カメラパラメータや形状メッシュの不正確さによるテクスチャの不一致が現れやすい箇所である。そのため、他の箇所に比べると一致度が向上しにくい箇所であり盛んに細分割が行われている。これらの箇所では、テクスチャを完全に一致させることはできていないが、細分割処理が進むにつれて、一致している領域が増えていっていることがわかる。一致度が極端に低い箇所が左裾に現れているが、これは形状メッシュに大きな不正確さが生じているためである。この箇所は、着物が床の上に広がっており、また、上方カメラから不可視であることから最も形状復元の困難な箇所であるといえる。

図13を見ると反復計算の早い段階において一致度の向上度が飽和していることがわかる。これは、テクスチャ変形で一致させられる箇所については提案手法が上手く働いているが、その他の箇所、つまり、前述したoccluding contourのように全く異なるテクスチャが乗ってしまう面については、提案手法のみで一致させることは困難であるため向上度が頭打ちになっているためだと考えられる。しかし図14(g)-(m)の全体を見ると、初期誤差と比べて一致度が向上していることがわかる。

4.2.2 出力画像の画質評価

出力画像の画質評価は以下の方法にて行った。仮想視点・視線をカメラ C_i の視点・視線と一致させ、カメラ C_i によって撮影した画像 I_i を使用せずに、出力画像 I'_i を生成する。この I_i と I'_i とのPSNRを求めることによって、提案手法によるテクスチャ変形の有効性を検証した。比較のために、視点依存頂点ベース法(以下、頂点ベース法と呼ぶ)と視点依存面ベーステクスチャブレンディング法(提案手法のアルゴリズムのうち、調和テクスチャ座標を用いず、幾何計算のみより得られたテクスチャ座標を用いてテクスチャ生成を行ったもの。以下、面ベースブレンディング法と呼ぶ)によって生成した出力画像とのPSNRを共に示す(図15)。こ

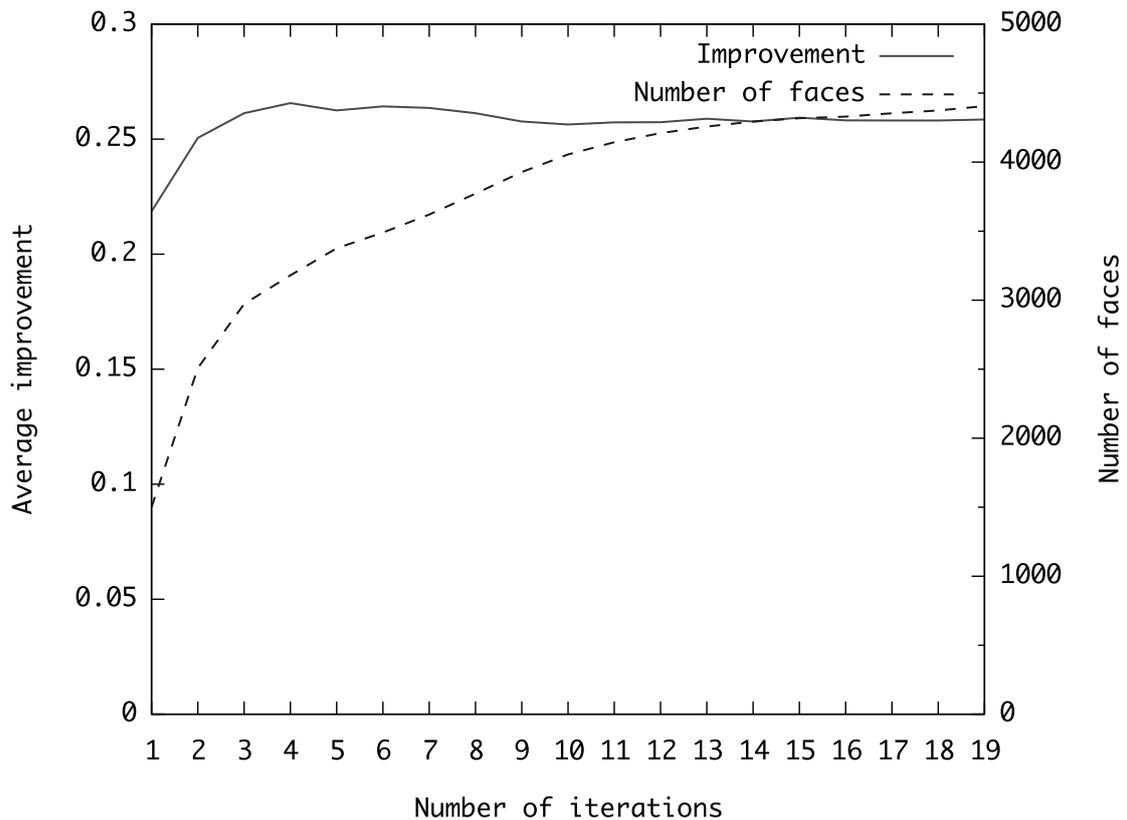


図 13 反復計算によるテクスチャー致度の向上度と面数

の評価は、焦点距離の異なるレンズを付けたカメラ 13 と頭上に設置したカメラ 14, 15 以外のカメラ、つまり、図 12 の上段と中段および下段の左から 2 枚を撮影したカメラを用いて行った。グラフから明らかのように、提案手法は他の二つと比べて高い評価値となっており、その有効性を確かめることができた。

4.3 定性評価

定量的な画質の評価については PSNR を用いて行ったが、主観的画質評価と PSNR の値が必ずしも一致するとは限らないことは一般に知られており、そのことを取り上げた文献も存在する [11]。主観的な画質評価と一致するような定量的評価法は、われわれの知る限り確立されているものはなく、また、本論文の主眼は画質の評価方法ではなくあくまで 3 次元ビデオのテクスチャマッピング法にあるため、画質の評価については上記の PSNR と主観的判断によって下記の基準により行う。

- PSNR が高いこと
- 撮影画像と比較して解像度感が同程度であること
- テクスチャの二重写りが少ないこと

画質の定性評価を行うために撮影対象の一部を拡大した出力画像を示す (図 16)。比較のために前節と同様に、頂点ベース法と面ベースブレンディング法を用いて生成した画像を共に挙げている。図 16 より頂点ベー

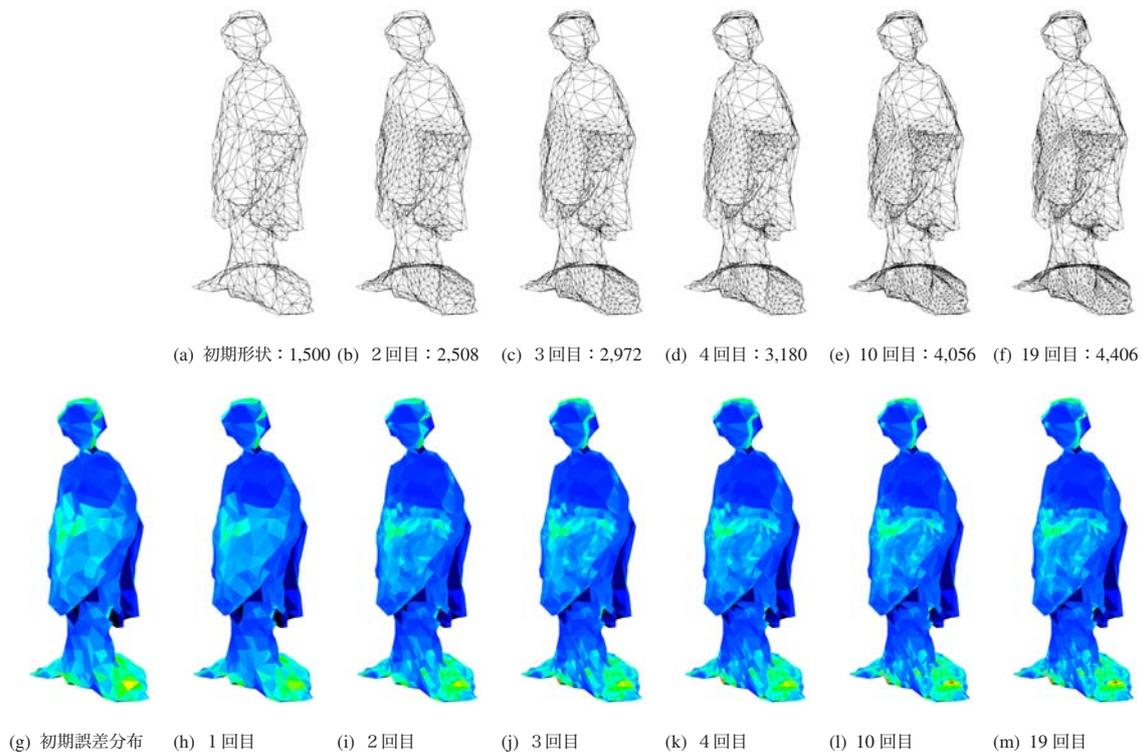


図 14 反復計算による面の細分割と誤差分布の変遷 (明るいほど誤差が大きいことを示す.)

ス法との画質の違いは一目瞭然である。頂点ベース法では、頂点色の補間によるボケや二重写りとカメラパラメータや形状メッシュの不正確さに起因するものの両方が混在しており、拡大すると鑑賞に堪えない画質となっている。より高密度な形状メッシュを用いることにより、高密度な出力画像を得ることは可能であるが、上記の不正確さに起因する画質の低下は解消することはできない。また、面ベースブレンディング法では、形状メッシュの解像度に関わらず撮影画像と同等の画質のテクスチャが生成できているが、カメラパラメータや形状メッシュの不正確さによるボケや二重写りが生じていることが確認できる。上記の手法と比べると、提案手法を用いることによって、ボケや二重写りのない画像が生成できることが明らかにわかる。特に、腹部の帯や袖の文様、だらりの帯の左上の鞠や下部の紋、腰の帯のズレに顕著な画質の差が見られる。また、各種法に用いたデータのサイズを表 5 に示す。ただし、形状メッシュおよび PVC は可逆圧縮 (gzip) を、テクスチャ画像については JPEG 圧縮をかけてあり、実際の運用の際に必要なデータサイズに近い値になっている。このように、データ量の側面からも提案手法が高い性能を示していることがわかる。

4.4 リアルタイム表示

本節では、本手法によるリアルタイム表示に関する実験結果を紹介する。この評価実験は上記にて用いた単一フレームの舞妓データと複数フレームのカンフーデータを用いて行った。カンフーデータについては、1 フレーム当たり約 0.3MB (形状メッシュ、テクスチャ) のデータ量となっており、これを 1 秒につき 25 フレームで再生するため単純に計算すると約 7.5MB/s のデータ量となる。ただし、ビューワ内部でデータを展開すると非圧縮の多視点テクスチャ画像となるため、データ量は膨大となる (約 200MB/s)。本実験では、データ展開の計算コストは考慮せず、レンダリングにおける実時間性を検証するため、より高速な主記憶装置に予め

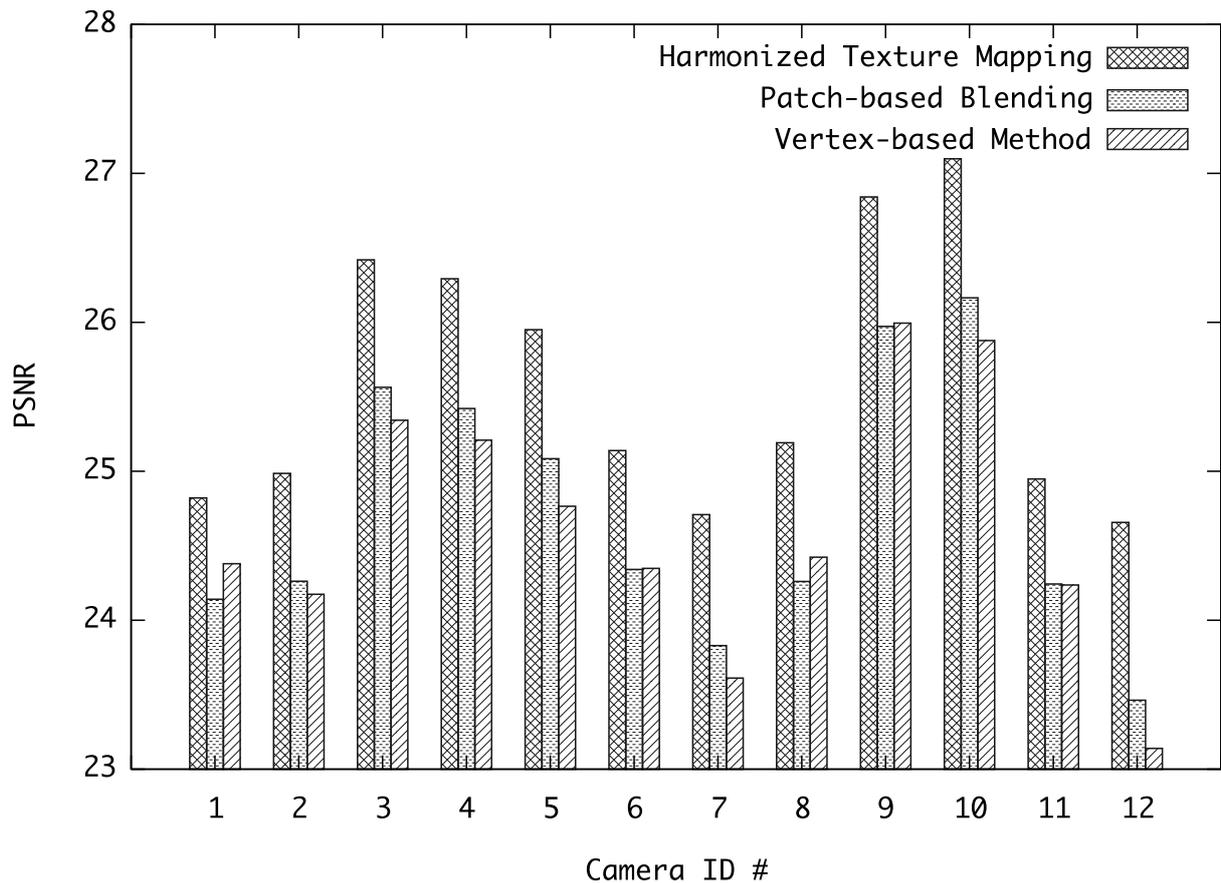


図 15 出力画像の画質評価 (PSNR)

表 5 1 フレームあたりのデータ量 (MB)

	Mesh	Texture(s)	PVC	Total
VDVBM	2.2	1.5	—	3.7
PBBM	0.1	0.4	—	0.5
HTM	0.1	0.4	0.3	0.8

VDVBM: 視点依存頂点ベース法,

PBBM: 面ベースブレンディング法,

HTM: 調和テクスチャマッピング法 (提案手法)

再生するデータをすべて読み込むことによって複数フレームの再生を行った。リアルタイム表示における各アルゴリズムについて、それぞれの計算時間を表 6 に示す。実験で用いた PC の仕様は CPU: Core2Duo 2.4 GHz, Memory: 4GB, GPU: GeForce 8800 GTX, VRAM: 750 MB であり、ソフトウェアは Managed DirectX および C# を用いて作成した。また、テクスチャ生成およびラスターライズは DirectX Pixel shader 2.0 を用いている。さらに、図 17 に本実験に用いた 3 次元ビデオデータの出力画像例を示す。このように従来法と比較して、遙かに高画質の 3 次元ビデオデータをリアルタイム (≥ 30 fps) で視点を移動しながら鑑賞できることが確認できた。

表6 画像1フレームの生成に要する計算時間：仮想視点を移動させながら1000フレーム描画した際の平均値 (ms)

	舞妓	カンフー
重み計算 (Alg. 8 (7))	< 1	< 1
調和テクスチャ座標計算 (Alg. 8 (8))	27.2	15.8
テクスチャ生成およびラスライズ (Alg. 9 (9))	< 1	< 1

5 むすび

本論文では、完全に正確であるとは言えないカメラパラメタと形状メッシュによって生じる出力画像のボケや二重写りを低減させるテクスチャマッピングアルゴリズムとして、調和テクスチャマッピング法 (Harmonized Texture Mapping) を提案した。提案手法の特徴は、映像化の際の視点に依存してテクスチャを適切に変形させることによって、これらの不正確さから生じるボケや二重写りを低減しつつ、撮影画像と同等の画質の任意視点画像を生成可能なことである。また、PSNRによる評価値においても、既存手法と比べて平均1 dBの向上が見られた。テクスチャの変形については、テクスチャ画像の一致度のみではなく、形状メッシュの制約に基づくため、単なる画素ベースの変形方法と比較すると安定した変形が可能である。さらに単純化した形状メッシュに対して、効果的に映像生成が可能なアルゴリズムであるため、生成される映像の画質を落とすことなく、3次元ビデオの形状データ量を大幅に削減できること、つまり、テクスチャ整合性の観点から形状メッシュを最適化できることを示した。また、提案手法をDirectXを用いて実装し一般的に用いられる性能のPCにおいてリアルタイムに映像生成できることを確認した。

今後、提案手法を向上させるために必要な議論としては、極端に形状が不正確な箇所に対する対処法や、カメラパラメタや形状復元の精度を考慮した最適なメッシュの単純化レベルの決定法が挙げられる。また、われわれは3次元ビデオの形状メッシュの圧縮法としてSkin-off [16]を提案しているが、この方法とうまく協調するテクスチャマッピングおよびテクスチャ圧縮方法についても取り組んでいく。

最後に本研究は、独立行政法人科学技術振興機構 (JST) チーム型研究CREST「デジタルメディア作品の制作を支援する基盤技術」および文部科学省プロジェクト「知的資産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」における研究開発課題「大型有形・無形文化財の高精度デジタル化ソフトウェアの開発」の支援を受けて行った。

参考文献

- [1] Per Einarsson, Charles-Felix Chabert, Andrew Jones, Wan-Chun Ma, Bruce Lamond, Tim Hawkins, Mark Bolas, Sebastian Sylwan, and Paul Debevec. Relighting human locomotion with flowed reflectance fields. In *Eurographics Symposium on Rendering* (2006)
- [2] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Eurographics EG '08*, **27**, 2 (2008)
- [3] Takeo Kanade, Peter Rander, and P.J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia, Immersive Telepresence*, **4**, 1, pp. 34–47 (1997)
- [4] Takashi Matsuyama, Xiao jun Wu, Takeshi Takai, and Shohei Nobuhara. Real-time 3D shape reconstruc-

- tion, dynamic 3D mesh deformation, and high fidelity visualization for 3D video. *International Journal of Computer Vision and Image Understanding*, **96**, 3, pp. 393–434 (2004)
- [5] Saied Moezzi, Li-Cheng Tai, and Philippe Gerard. Virtual view generation for 3d digital video. *IEEE Multimedia, Immersive Telepresence*, **4**, 1, pp. 18–26 (1997)
- [6] Tomoyuki Mukasa, Shohei Nobuhara, Atsuto Maki, and Takashi Matsuyama. Finding articulated body in time-series volume data. In *The 4th International Conference on Articulated Motion and Deformable Objects (F. J. Perales and R. B. Fisher: AMDO 2006, LNCS 4069)*, pp. 395–404 (2006)
- [7] Rieko Otsuka, Takeshi Hoshino, and Youichi Horry. Transpost: A novel approach to the display and transmission of 360 degrees-viewable 3d solid images. *IEEE Transactions on Visualization and Computer Graphics*, **12**, 2, pp. 178–185 (2006)
- [8] Steven M. Seitz and Charles R. Dyer. View morphing. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive technique*, pp. 21–30 (1996)
- [9] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, pp. 21–31 (2007)
- [10] Visual Computing Lab, Italian National Research Council. *VCGLib* (2005)
- [11] 河合紀彦, 佐藤智和, 横矢直和. 画像修復における定性的・定量的評価法に関する考察. 情報科学技術フォーラム (FIT) 情報科学技術レターズ, **6**, pp. 245–248 (2007)
- [12] 田村秀行, 柴田史久. 可視化技術で創造力を高める映画制作支援. 情報処理, **48**, 12, pp. 1365–1372 (2007)
- [13] 韓昇龍, 山崎俊彦, 相澤清晴. 拡張ブロックマッチングアルゴリズムを用いた3次元映像の圧縮. 映情学誌, **61**, 8, pp. 1177–1187 (2007)
- [14] 富山仁博, 片山美和, 折原豊, 岩館祐一. 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. 映情学誌, **61**, 4, pp. 147–481 (2007)
- [15] 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. 映情学誌, **58**, 4, pp. 593–602 (2002)
- [16] 曾良洋介, 波部斉, 松山隆司. Skin-off: 3次元ビデオ圧縮のためのテクスチャ指向型2次元平面展開法. 映情学誌, **60**, 4, pp. 553–560 (2006)
- [17] 唯野隆一, 山崎俊彦, 相澤清晴. 測地線距離による高速かつロバストな3次元ビデオからのスケルトン抽出. 信学論, **J90-D**, 7, pp. 1716–1717 (2007)



(a) 撮影画像

(b) 撮影画像



(c) 頂点ベース法



(d) 頂点ベース法



(e) 面ベースレンディング法



(f) 面ベースレンディング法



(g) 提案手法 (HTM)



(h) 提案手法 (HTM)

図 16 定性評価画像

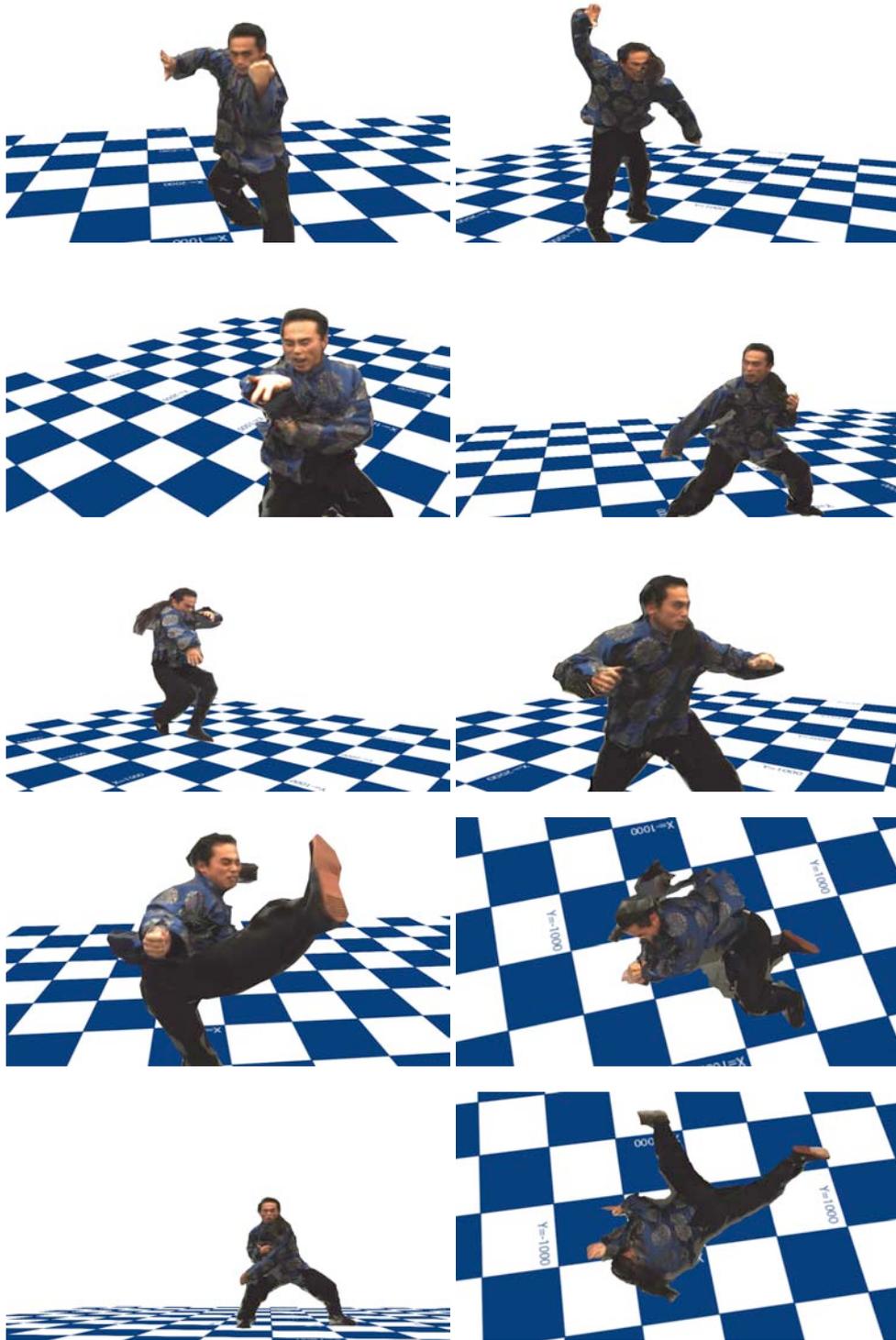


図 17 複数フレーム再生の例