

Skin-off: テクスチャ指向型 2 次元平面展開による 3 次元ビデオの圧縮

波部 斉[†] 曾良 洋介[‡] 松山 隆司[‡]

[†] 京都大学大学院工学研究科電子工学専攻 〒606-8501 京都市左京区吉田本町

[‡] 京都大学大学院情報学研究科知能情報学専攻 〒606-8501 京都市左京区吉田本町

E-mail: [†] habe@media.kyoto-u.ac.jp, [‡] {katsura, tm}@vision.kuee.kyoto-u.ac.jp

あらまし 3次元ビデオは実世界の対象の形状・動き・色・模様をそのまま保存する3次元映像メディアであり、我々の日常生活の様々な面への応用が期待されている。3次元ビデオを保存・伝送・放送するためには、標準化された圧縮技術が必須である。本稿では、3次元ビデオを効率的に表現・圧縮するための圧縮手法“Skin-off”を提案する。本手法では、3次元メッシュを切り開き、2次元平面へ展開することで、3次元ビデオがもつ幾何・テクスチャ情報を平面上で効率的に表現し、圧縮する。Guら[5]によって提案されたGeometry Imagesは我々の提案手法と類似の考え方をを用いているが、Geometry Imagesは幾何情報のみを用いて展開操作を最適化しているのに対し、Skin-offでは幾何情報とテクスチャ情報を用いて最適化する。テクスチャ情報は映像品質を決定する重要な要素であり、それを考慮することで、データ効率の向上が期待できる。本稿では、基本的アイデアを実現するアルゴリズムを示し、実験結果によってその有効性を示す。

キーワード 3次元ビデオ, 圧縮, 符号化, 幾何情報, テクスチャ, Skin-off

Skin-off: 3D Video Compression Based on Texture-Oriented 2D Plane Mapping

Hitoshi HABA[†] Yosuke KATSURA[‡] and Takashi MATSUYAMA[‡]

[†] Graduate School of Engineering, Kyoto University Yoshida-Hommachi, Kyoto, 606-8501 Japan

[‡] Graduate School of Informatics, Kyoto University Yoshida-Hommachi, Kyoto, 606-8501 Japan

E-mail: [†] habe@media.kyoto-u.ac.jp, [‡] {katsura, tm}@vision.kuee.kyoto-u.ac.jp

Abstract 3D video records dynamic 3D visual events as is. The application areas of 3D video include wide varieties of human activities. To promote these applications in our everyday life, a standardized compression scheme for 3D video is required. In this paper, we propose a practical and effective scheme for representing and compressing 3D video named skin-off, in which both the geometric and visual information are efficiently represented by cutting open a 3D mesh and mapping it onto a 2D array. Our skin-off scheme shares much with geometry images, proposed by Gu et al [5]. However, while geometry images employ the 3D surface shape information alone to generate 2D images, the skin-off scheme we are proposing employs both 3D shape and texture information to generate them. This enables us to achieve higher image quality with limited bandwidth. Experimental results demonstrate the effectiveness of the skin-off scheme.

Keyword 3D Video, Compression, Geometry, Texture, Skin-off

1. まえがき

複数のビデオカメラを用いると、実世界の物体の3次元形状・動き・色・模様をそのままデジタル化することができる[1]。舞踊などの無形文化財、スポーツの動作、動物の生態などをデジタル化すると、娯楽、教育、研究、文化財保護などの多岐にわたる分野に応用が可能である。我々はこのような新しい映像メディアを「3次元ビデオ」と呼び、高精細な3次元ビデオを獲得・伝送・提示するための研究を行っている[2]。本稿では、その中の圧縮技術に焦点をあて、我々が提案するSkin-offによる圧縮について述べる。

3次元ビデオを用いると、鑑賞者が自分の見たい視点からの映像をみることができ、2次元映像にはなかったインタラクティブな効果を得ることができる。そ

のため、次世代の映像メディアとしての期待が集まっており、効率のよい伝送を実現するための統一された圧縮法が望まれている。MPEGなどでの標準化検討も行われている[3, 4]。

3次元ビデオは、時間経過とともに動的に変化する3次元メッシュと、その表面のテクスチャから構成されている。本稿で提案する手法は、(1)3次元メッシュを適当な切り目(カットパス)に沿って切り開き、(2)3次元メッシュの幾何情報とテクスチャ情報を2次元平面へ展開し、平面上でそれらの情報を記述する(図1)、(3)平面上の3次元ビデオデータに対し、既存の平面ビデオ用のコーデックを用いて圧縮する、という処理から構成される。3次元物体を平面に切り開くという操作から、このアルゴリズムをSkin-offアルゴリズム

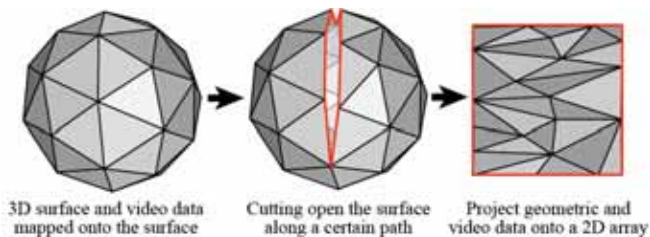


図 1 3次元メッシュの平面への展開

ムと呼ぶ。Skin-off アルゴリズムは、情報圧縮に通常の2次元映像用のコーデックを利用でき、既存技術との親和性が高いという特徴を持っている。

3次元メッシュの幾何情報を圧縮する手法は、CG分野を中心に数多く提案されている[5, 7]。その中で、Guらが提案している Geometry Images[5]は、我々が提案する Skin-off アルゴリズムと類似する考え方を採用し、与えられた3次元メッシュに対してある切り目を入れて平面上へ展開し、3次元メッシュの幾何情報を平面上で表現している。また、動的に形状が変化するメッシュへ Geometry Images を拡張した Geometry Videos[9]も提案されている。しかし、いずれもメッシュの幾何歪みのみに着目し、切り目の選択や展開操作によって生じる幾何歪みが最小になるよう最適化を行っている。

これに対し、我々は3次元メッシュの幾何情報と3次元メッシュ表面に対応づけられたテクスチャ情報の双方に基づく2次元平面への展開を行う。復号された3次元ビデオを鑑賞する際に、たとえ3次元形状に歪みが生じていても、その表面に有意なテクスチャがなければ鑑賞者は歪みの存在を感じないと考えられる。逆に細かなテクスチャがあれば、わずかな幾何的歪みも目立つと考えられる。このように、最終的に鑑賞者に提供する映像の品質を向上させるには、幾何情報だけでなく、テクスチャ情報の利用が必須であることは明らかである。また、撮影対象の動きによって幾何形状が動的に変化するのに対し、その模様は変化しないことが多い。すなわち、テクスチャは時間軸方向の冗長性の高い情報であるため、時間的に変化する3次元ビデオの圧縮でも重要な役割を果たすと期待される。

本稿では、まず2節において、3次元ビデオデータの定義を与え、本研究の中心的アイデアであるテクスチャ情報の利用の有効性を論じ、関連研究との比較を行う。次いで、3節ではそのアイデアに基づくアルゴリズムを述べ、4節でシミュレーションデータを用いてアルゴリズムの基本的有効性を示す。

2. 3次元ビデオとその圧縮

2.1. 3次元ビデオ

前節で述べたとおり、3次元ビデオデータは、時間経過とともに動的に変化する、

- ・ 3次元メッシュ $M(t)$
- ・ $M(t)$ の表面に対応づけられるテクスチャ $T(t)$

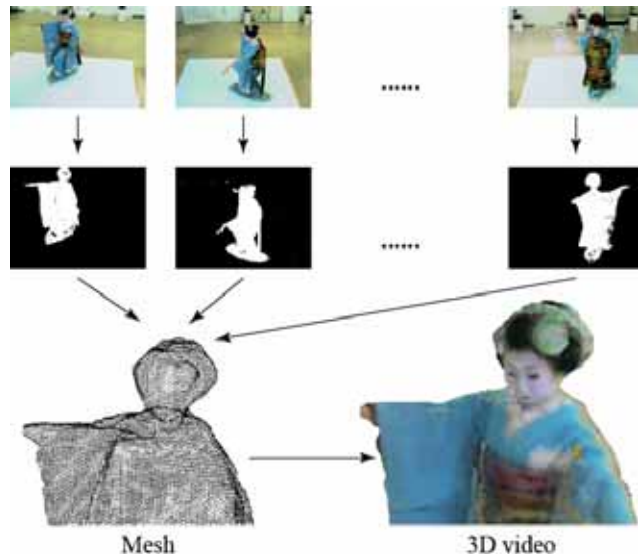


図 2 視体積交差法による3次元ビデオの生成

から構成されるとする。このような3次元ビデオは、複数のビデオカメラで撮影した対象の映像に、視体積交差法を適用して得ることができる[2] (図2)。

一般に、視体積交差法によって求めた3次元メッシュは時刻間の対応付けが取れていないが、弾性メッシュモデル[2]を適用すると、時刻間での対応が取れたメッシュデータが得られる。これにより、幾何情報の時間軸上の冗長性が高まるので圧縮効率が向上することが期待される。

本稿では、この3次元メッシュは単一の2次元多面体を表しているとする。複数の対象を圧縮する場合は個別に処理を行えばよいので、このように仮定しても一般性を失わない。また、本稿では、単一フレームの3次元ビデオデータを用いることとし、その中で Skin-off アルゴリズムの基本的な有効性を示す。

2.2. 幾何情報とテクスチャ情報の映像品質への影響

1節で述べたように、我々が提案する Skin-off アルゴリズムでは、3次元ビデオを構成する幾何情報とテクスチャ情報を利用して最適な展開操作を決定し、限られたデータ量での映像品質の向上を図る。ここでは、3次元ビデオの幾何情報とテクスチャ情報が映像品質に与える影響を議論する。

3次元メッシュとテクスチャのもつ情報量の大小は、メッシュの場合は形状の複雑さ、テクスチャの場合はその模様の複雑さを尺度として評価できる。圧縮によってある領域のメッシュあるいはテクスチャの情報量が失われたとき、他方の情報量の大小と映像品質との関係を考える。ここでは、議論を簡単にするために、それぞれの情報量が多い場合と少ない場合を考えることにする。

まず、メッシュの情報量が失われた、すなわち形状に歪みが生じたとする。もし、その領域に十分細かなテクスチャが存在していたとすると、映像上でも形状歪みが目立ち、品質が劣化したと知覚される。逆に、

有意なテクスチャが存在していなければ、若干の形状歪みは目立たず、品質の劣化はさほど感じられないと考えられる。次に、テクスチャの情報が失われ、ボケなどが生じた場合は、メッシュ形状にかかわらずそれが映像上に現れるため、品質の劣化につながる。

このように、映像品質の観点で3次元ビデオのもつメッシュ情報、テクスチャ情報の関係をまとめると、(1)幾何形状に許容される歪みは、その表面のテクスチャの情報量に依存し、テクスチャが細かい領域ほど幾何歪みを小さくしなければならない、(2)テクスチャ情報は、メッシュ形状に関係なく、生じた歪みが映像品質に影響する、といえる。本稿では、この考え方に基づく基本的なアルゴリズムを示し、それをを用いた実験結果から、その有効性を示す。

2.3. 関連研究

CG分野の中で3次元メッシュの効率的な表現・圧縮を目指す手法が数多く提案されている[7]。それらの多くは、メッシュの接続関係を特別なデータ構造で表現するため、特別なコーデックが必要となる。

一方、Guらは3次元メッシュの幾何情報をGeometry Imagesという2次元矩形画像で表現し、既存の映像圧縮法を用いて圧縮する手法を提案している[5]。この手法は、2次元矩形画像への変換が非可逆であり形状情報の損失が生じる。その損失を最小限にするようにGeometric Stretchと呼ばれる、変換前後の幾何形状の歪みを数値化した尺度を最小化している。また、Briceñoらによって時間的に変化するデータへの拡張も試みられている[9]。しかし、いずれの手法も幾何形状のみに着目したものであり、2.2節で述べた観点からは不十分であるといえる。

テクスチャ情報に扱いについても、CG分野で活発な研究が行われている。例えば、Sanderらは限られた2次元平面上で、テクスチャ情報をその情報量に応じて効率的に表現する手法を提案した[10]。Lévyらはいくつかのまとまりに分けて平面展開することで歪みを少なくするマッピング手法を提案している[6]。

以上のように、3次元メッシュとテクスチャ情報それぞれを効率よく表現・圧縮するための手法は数多く提案されているが、提案手法のようにそれらの相互の関係を考慮した手法は、筆者らが知る限り提案されていない。

3. Skin-off: 平面展開による3次元ビデオ圧縮

前節までの議論を踏まえ、幾何情報とテクスチャ情報を利用したSkin-offアルゴリズムについて述べる。本節で述べるアルゴリズムはSkin-offの基本的有効性を示すための簡単な実装の例である。

3.1. Skin-off アルゴリズムの概略

Skin-offアルゴリズムの概略およびそれぞれの処理での問題点は以下ようになる。

Step1. 3次元メッシュの切り開き

任意の3次元メッシュは、適当なエッジで切り開けば平面へ変換(展開)可能であるため、初期値

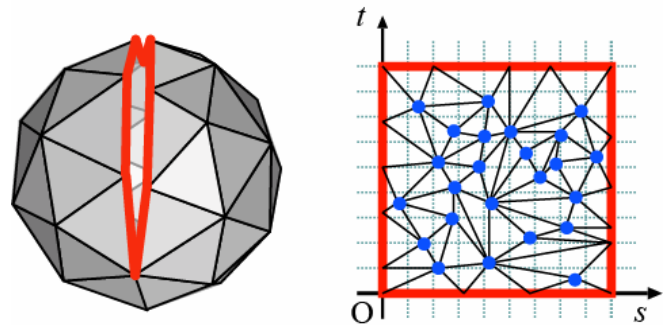


図3 カットパスと平面展開

として、切り開きに必要となるエッジ集合(カットパスと呼ぶ。図3の太線部分)を選択する。

Step2. 平面上の頂点位置の最適化

カットパスを選択した後で、矩形平面内でのメッシュ頂点(図3右で示す点)の配置を最適化する。この操作が幾何情報とテクスチャ情報の表現効率を決定するので、それぞれを過不足なく表現するようにする必要がある。

Step3. カットパスの更新

Step2で頂点配置を最適化したあと、Step1で選ばれたカットパスを更新する。カットパスの選び方によっては2次元平面上での歪みが生じるため、歪みを最小化するために、幾何情報とテクスチャ情報を利用して更新する。

Step4. 矩形領域内でのサンプリング

矩形平面上に適当な解像度の格子点を設定し、格子点上で3次元ビデオデータをサンプリングする。これによって3次元ビデオデータを表現する2次元配列が得られる。具体的には、メッシュ情報は頂点の座標値を、テクスチャ情報はRGBの輝度値などを配列に格納する。図3右では平面上に展開された3次元頂点座標データが、破線で示す格子点でサンプリングされ、2次元配列に記録される。

Step5. 映像圧縮手法の適用

矩形画像に対して、既存の映像圧縮手法を適用する。

以上の処理の中で、平面への展開を行うStep1, 2, 3が表現・圧縮の効率を決定する処理である。本手法ではテクスチャ情報と幾何情報を評価する目的関数を定め、繰り返し演算でそれを最小化する。次節以降ではその詳細について述べる。

なお、3次元ビデオを構成する幾何情報とテクスチャ情報はそれぞれ異なる頂点配置による矩形配列で表現可能であるが、その場合Step4のサンプリングの影響で幾何情報とテクスチャ情報の整合が取れなくなる。よって、本手法では幾何情報とテクスチャ情報は同じ頂点配置で表現されるものとする。

3.2. 幾何情報・テクスチャ情報の評価関数

アルゴリズムの詳細を述べる準備として、3次元ビデオデータの情報量の大小を評価する尺度を定義する。幾何情報に関しては、形状の複雑さが情報量の大小を決定するので、メッシュ表面で計算される曲率が幾何

情報の情報量を表すと考えることができる。テクスチャ情報に関しては、3次元メッシュを構成する3角形パッチ表面でのテクスチャの複雑さが情報量に相当すると考えられる。それを評価するにはいくつかの手段が考えられるが、ここでは空間微分値の強度の和

$$D(f) = \frac{1}{N_f} \sum_{p \in f} \sqrt{d_x(p)^2 + d_y(p)^2},$$

が該当する面 f のテクスチャ情報を示すものと定義する。ここで p はパッチ表面の画素を示す。

3.3. 切り開きのためのカットパス選択と更新

Step1 および Step3 では、与えられた3次元メッシュを2次元平面へ展開するためのカットパス選択と更新を行う。本稿で用いたアルゴリズムでは、初期カットパスの選択 (Step1) では、Guらが提案した Geometry Images[5]と同様の手法 (詳細は省略) でトポロジー変換に必要な最低限のカットパスを選択し、Step3 のカットパス更新の際に、幾何情報とテクスチャ情報を利用する。

次節で述べるように、Step2 の頂点配置の最適化処理において、カットパスは平面矩形画像の輪郭部に固定されるため、カットパス周辺での幾何情報、テクスチャ情報の歪みは小さくならない可能性がある。2.2節の議論を踏まえると、この影響を避けるためにはカットパス周辺には有意なテクスチャが存在しないようにすればよい。

また、幾何情報を考えると、曲率が非常に大きい部分がある場合、その領域を平面に投影すると幾何歪みが非常に大きくなる。曲率が大きい部分にカットパスを設けることで、全体的な幾何歪みを小さくできると考えられる。

Geometry Images では、後者の考え方のみに基づき最適なカットパスを探索しているが、ここではそれにテクスチャ情報を加味してカットパスの更新を行う。その際には、各面のテクスチャ密度をもとに、各エッジに対してテクスチャ密度

$$D(e) = D(f_1) + D(f_2)$$

定義する。ここで、 f_1, f_2 は、注目するエッジ e に接する面である。具体的なアルゴリズムは以下のようになる。

Step3-1. 曲率が最大になる頂点を求める。

Step3-2. Step3-1 で得られた頂点と現在のカットパスの間でテクスチャ密度 $D(e)$ の和が最も小さくなる経路を探索し、それを新たなカットパスとする。

3.4. 頂点配置の最適化

次に、得られたカットパスを用いて3次元メッシュを矩形平面へ展開する際のメッシュ頂点の配置を最適化する。

まず、カットパス上のエッジを矩形の輪郭上に配置する。切り開く際にはカットパス上のエッジが複製される。Step4 では各々のエッジに対してサンプリング

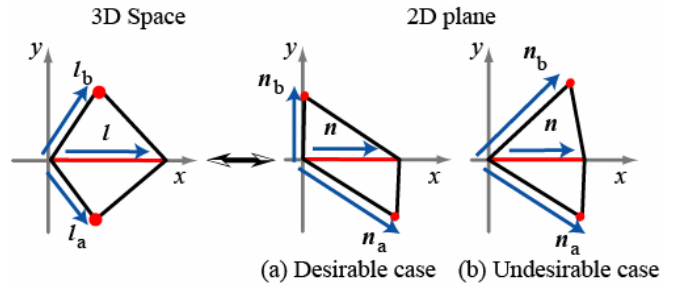


図 4 2次元平面への変換の方向性

処理が行われるため、平面展開後に再構成したメッシュに隙間が生じないようにカットパス上の頂点を配置する。

3.4.1. 目的関数の定義

矩形の内部の頂点配置は、幾何情報とテクスチャ情報のサンプリング間隔を決定する。2.2節で述べたように、限られたデータ量の中では、テクスチャ密度が高い領域で幾何歪みを少なくすると映像品質が良くなると考えられる。ここでは、(1)テクスチャ密度 $D(f)$ が大きい面が平面上でも大きくなる、(2)隣接する面では平面上の大きさが極端に変化しない、(3)平面へ展開される際に極端な形状変化が起きず、引き伸ばされる方向が隣接面で極端に変化しない、という条件を与える。この中で(1)は、面 f のサンプリング間隔を $D(f)$ に応じて変化させるための条件であり、(2)、(3)は隣接面でのサンプリング間隔や引き伸ばされる方向が極端に変化することによって、幾何歪みやテクスチャ歪みが生じないようにするための条件である。最適化の目的関数 TG は、

$$TG = \sum_f (\alpha \cdot TS(f) + \beta \cdot GS(f) + \gamma \cdot SD(f))$$

と定義する。ここで、 α, β, γ はスケールファクタである。 $TS(f)$ は上記の条件(1)、(2)を実現するための項であり、

$$TS(f) = \left(\frac{A_{3D}(f)}{A_{2D}(f)} \right) \sum_{f_n \in F} w(f, f_n) D(f)$$

で定義される。 $A_{2D}(f), A_{3D}(f)$ はそれぞれ面 f の2次元平面、3次元空間中での面積、 F は面 f の近傍の面の集合、 $w(f, f_n)$ は f と f_n の距離が近いほど大きな値をとる重み関数を表す。テクスチャ密度 $D(f)$ が大きい面を平面上で大きく配置すると $TS(f)$ は小さな値をとる。 $GS(f)$ と $SD(f)$ は条件(3)を実現するための関数である。 $GS(f)$ は Geometry Images でも使用されていた Geometric Stretch [8]であり、

$$GS(f) = \sqrt{((\partial S / \partial s)^4 + (\partial S / \partial t)^4) / 2}$$

で定義される。ここで、 S は2次元平面から3次元空間への頂点の対応づけを表す関数であり、 s, t は2次元平面での座標を表す。 $GS(f)$ は2次元平面への展開でのメッシュ形状の変形の度合いを表している。

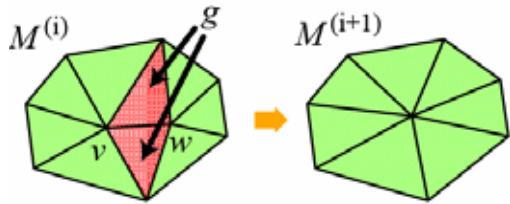


図 5 メッシュ簡略化のための頂点併合

$SD(f)$ は,

$$SD(f) = \sum_{l \in E} \left\| \frac{\mathbf{l}_a}{\|\mathbf{l}_a\|} + \frac{\mathbf{l}_b}{\|\mathbf{l}_b\|} - \frac{\mathbf{n}_a}{\|\mathbf{n}_a\|} - \frac{\mathbf{n}_b}{\|\mathbf{n}_b\|} \right\|$$

で定義される。図 4(b)に示すようにエッジ l に隣接する面の引き伸ばされ方が極端に異なる場合、エッジ l でテクスチャ情報のサンプリング間隔が大きく変化する。これは、図 9(a)にみられるような歪みにつながる。 $SD(f)$ はそれを避けるための評価関数であり、 E は注目している面 f を構成するエッジの集合を表す。他の文字は図 4 に示すように引き伸ばされたエッジを表す。 $SD(f)$ は図 4(a)のような場合に小さくなり、(b)の場合に大きくなる。

3.4.2. メッシュ簡略化による最適解探索

頂点の配置は非常に自由度が高いため、最適化処理が局所解に収束してしまう可能性が高い。これを避けるため、簡略化したメッシュを作成し、徐々に複雑なメッシュに戻しながらそれぞれの段階で最適化を行う。これは、初期段階で大局的に最適解を求めてから、徐々に詳細な最適解を探索していくものである。Geometry Images でも類似の発想の処理が行われているが、テクスチャ情報が考慮に入られていないため、そのまま適用することはできない。メッシュ簡略化を含んだ Step2 の処理全体は以下ようになる。

Step2-1. 3次元メッシュ $M^{(i)}$ の頂点 v とそれに隣接する頂点 w のすべての組み合わせに対し、 v と w のいずれかに接する面 f のテクスチャ密度の最大値と最小

値の差 $D_{\max} - D_{\min}$ を求める。この差が、注目する頂点付近のテクスチャ密度の一様性を表している。

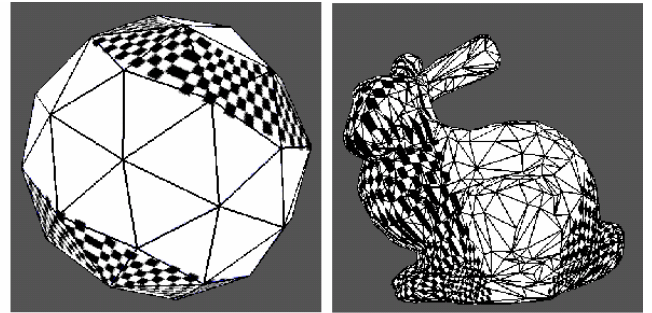
Step2-2. $D_{\max} - D_{\min}$ が最小となった頂点の組 v, w を併合し、一つの頂点とする(図 5)。これによって新たなメッシュ $M^{(i+1)}$ を得る。なお、カットパスに属するエッジは併合によって消去されないようにする。また、併合した v, w に隣接し、併合によって消去されない面 f のテクスチャ密度は、

$$D(f') = \frac{1}{A(f) + \sum A(g)} (A(f) \cdot D(f) + \sum A(g) \cdot D(g))$$

によって更新する。 g は消去される面、 A は面の面積を表す。

この処理により、同じようなテクスチャ密度をもつ面から統合されていき、簡略化されたメッシュでは大局的な最適化が可能になる。

Step2-3. 併合可能なすべての面が併合され、カットパ



(a) Eighty-faced Polyhedron (b) Stanford Bunny

図 6 実験に用いたデータ

スに属さない頂点 1 つのみを含むメッシュ $M^{(N)}$ が得られるまで、Step2-1, 2-2 を繰り返す。

Step2-4. 最も簡略されたメッシュ $M^{(N)}$ について、目的関数 TG を最小化するように頂点配置を最適化する。Step2-5. $M^{(N-1)}$ については、 $M^{(N)}$ で求めた最適な頂点配置を初期値とし、同様に最適化を行う。その後、 $M^{(0)}$ まで最適化処理を繰り返す。

Step2-6. $M^{(0)}$ での最適化の結果で得られた TG の値があらかじめ定めた閾値を下回れば、Step1~3 を終了し Step4 に移る。下回らなければ Step3 に戻りカットパスの更新を行う。

4. 評価実験

本節では Skin-off アルゴリズムの基本的有効性を評価実験によって示す。本手法の基本アイデアである、幾何情報とテクスチャ情報を用いた平面展開の有効性を確かめるために、3節で述べたアルゴリズムの Step1~3 の処理のみを行った。具体的には、図 6 に示す 80 面体と Stanford Bunny のデータを使用し、Step3 までの処理を行う。その後平面上でサンプリングされた幾何・テクスチャデータを用いて特定の視点からの画像を生成した。比較対照として、Step2, 3 の中のテクスチャ情報を用いた処理を行わず、幾何情報のみで展開操作を行う手法を用いて同じ大きさの矩形平面を生成した。これは、Geometry Images を作成する処理に相当する。

まず、Stanford Bunny に対して Step1 および Step3 の処理で最終的に得られたカットパスの結果を図 7 に示す。この図を見ると、テクスチャ情報を加味することでテクスチャが存在する領域を避けるようにカットパスが選ばれていることが分かる。

次に、Step3 の結果で得られた平面上の幾何およびテクスチャデータを図 8 に示す。図では、平面に展開されたテクスチャの上に、メッシュを重畳して表示している。(b)および(d)がテクスチャおよび幾何情報を用いた結果である。図をみると、Skin-off アルゴリズムによって、テクスチャが存在する領域が大きくマッピングされていることが分かる。

最後に、特定の視点から撮影対象を見たときの画像を生成した。図 9 にその結果を示す。図中では、円内の

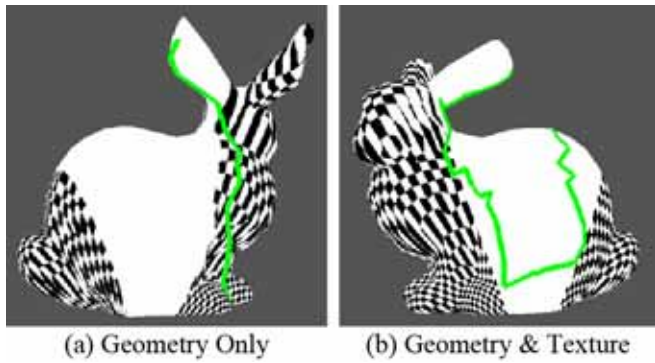


図 7 カットパスの選択結果 (Stanford Bunny)



図 8 平面展開結果

テクスチャを拡大して示している .80 面体を用いた(a)と(b)を比べると、幾何情報のみを用いた(a)では直線部分に段差が見られている。Stanford Bunny を用いた(c)と(d)では、テクスチャ情報を用いた(d)でテクスチャが鮮明になっている。これらの結果をみると、テクスチャ情報を考慮することで同じデータ量での画像品質が向上していることが分かる。

5. むすび

本稿では、3次元ビデオの幾何・テクスチャ情報を効率よく表現・圧縮する Skin-off アルゴリズムについて述べた。3次元メッシュおよび表面のテクスチャ情報を表現・圧縮する手法は多数提案されているが、それらと比較した提案手法の特徴は、幾何情報とテクスチャ情報の相互関係を考慮しているところにある。4

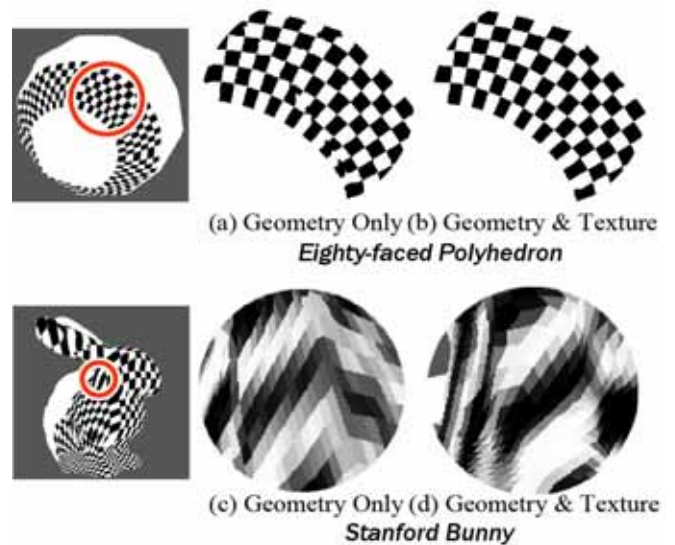


図 9 映像生成結果

節に述べた実験は、平面展開の処理のみを比較したものであるが、本手法の基本的有効性を示していると考えられる。

今後の課題としては、画質の定量評価、時間変化する3次元ビデオデータへの拡張などがあげられ、これらの検討を鋭意進めているところである。

謝辞

本研究は、文部科学省「知的財産の電子的な保存・活用を支援するソフトウェア技術基盤の構築」事業の「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」プロジェクト、実験で使用した Stanford Bunny は The Stanford 3D Scanning Repository のものである。

文 献

- [1] S.Moezzi, L.Tai, P.Gerard: Virtual view generation for 3d digital video. IEEE Multimedia, 4(1):18-26, Jan.-Mar. 1997.
- [2] T.Matsuyama, X.Wu, T.Takai, S.Nobuhara: Real-Time 3D Shape Reconstruction, Dynamic 3D Mesh Deformation, and High Fidelity Visualization for 3D Video. CVIU Vol.96, No.3, pp.393-434, 2004.
- [3] ISO IEC JTC1/SC29/WG11(MPEG): "Report on 3DAV Exploration". WG11 document N5878, Trondheim, Norway, 2003.
- [4] ISO IEC JTC1/SC29/WG11(MPEG): "Applications and Requirements for 3DAV". WG11 document N5326, Awaji, Japan, 2002.
- [5] X.Gu, S.J.Gortler, H.Hoppe: Geometry Images. SIGGRAPH 2002, pp.355-361, 2002.
- [6] B.Lévy, S.Petitjean, N.Ray, J.Maillot: Least Squares Conformal Maps for Automatic Texture Atlas Generation. SIGGRAPH 2002, pp.362-371, 2002.
- [7] G.Taubin, J.Rossignac: Geometric Compression Through Topological Surgery. ACM Transaction on Graphics, Vol.17, No.2, pp.84-115, 1998.
- [8] P.Sander, J.Snyder, S.Gortler, H.Hoppe: Texture Mapping Progressive Meshes. SIGGRAPH 2001, pp.409-416, 2002.
- [9] H.M.Briceño, P.V.Sander, L.McMlimman, H.Hoppe: Geometry Videos: A New Representation for 3D Animations. SIGGRAPH 2003, pp.136-146, 2003.
- [10] P.Sander, S.Gortler, J.Snyder, H.Hoppe: Signal Specialized Parametrization. Proc. 13th Eurographics Workshop on Rendering 2002, pp.87-100, 2003.