

AR ジオラマにおける特徴点の可視性に基づく 四面体カービングによる三次元形状復元手法

安原 広幸^{†1} 間下 以大^{†1,†2}
清川 清^{†1,†2} 竹村 治雄^{†1,†2}

近年、拡張現実感 (Augmented Reality, AR) と呼ばれる、現実空間に対して計算機情報を重畳表示しユーザを支援する技術が身近なものとなり始めている。多くの AR システムが予め設定された作業環境で予め用意されたコンテンツを再生するのみであるのに対して、我々が従来提案している AR ジオラマシステムは、実物体を三次元モデルとして復元し、AR 空間内で配置や移動ができるインタラクティブな AR システムである。しかしこのシステムには、復元されるモデルの精度が不十分であるという問題があった。本研究では、実物体の三次元モデルを AR 環境にしながら実時間で高精度に構築する特徴点ベースの形状復元手法を提案する。復元対象の物体に含まれる自然特徴点群からドロネー四面体分割によってポリゴンメッシュを作成し、物体表面の三次元形状を復元する。また、実際には存在しない四面体を削除するために特徴点の可視性に基づいてポリゴンメッシュのカービングを行う。実装の結果、従来の AR ジオラマシステムに比べて精度の高い三次元形状復元を実現できることを確認した。

A 3D Reconstruction Method Using Tetrahedron Carving Based on Feature Point Visibility for AR Diorama System

HIROYUKI YASUHARA,^{†1} TOMOHIRO MASHITA,^{†1,†2}
KIYOSHI KIYOKAWA^{†1,†2} and HARUO TAKEMURA^{†1,†2}

Recently, augmented reality (AR) technologies, which overlay computer generated information onto the real world, have been widespread. Most AR systems, however, can play only AR contents that have been prepared in advance of the usage. We have previously proposed AR Diorama System that constructs AR contents by using a handheld camera and stroke input. However, there were problems in virtual object creation and object behavior authoring due to low precision of the 3D reconstruction. In this study, we consider to employ a better realtime 3D reconstruction method to improve AR Diorama System. Our approach is based on feature point tracking. A feature point cloud is extracted

from the real object, and then converted into a polygon mesh through a Delaunay tetrahedralisation. Tetrahedrons are carved away based on a feature-point visibility test to obtain a surface mesh approximating the target object, which is then texture mapped to create a virtual object. Experimental results show that our method can reconstruct better 3D models in terms of precision compared to the previous system, though the carving function needs to be further improved to acquire a good result for a concave object.

1. はじめに

近年、拡張現実感 (AR, Augmented Reality) と呼ばれる、現実空間に対して計算機情報を重畳表示しユーザを支援する技術が身近なものとなり始めている。これは、加藤らによる ARToolKit¹⁾ などの扱いやすいマーカベースのトラッキングライブラリの登場によることも大きい。また、近年では自然特徴点追跡によりマーカを用いない AR システムも数多く開発されており、PTAM²⁾ などの特徴点ベースのトラッキングライブラリが公開されている。

しかしながら、多くの AR システムは予め設定された作業環境で予め用意されたコンテンツを再生するのみである。即ち、AR 環境を体験しながら、ユーザが随時実環境の一部を直接コンテンツとして取り込んだり、コンテンツの振る舞いを AR 環境にしながら付加・編集する (オーサリング)、というようなことは、従来の AR システムでは困難であった。

一般的に利用され得る AR オーサリングシステムの開発によって、AR システムの利便性やエンタテインメント性の向上が期待できる。例えば、実世界の物体表面に沿ってバーチャルな注釈を書き込む、生成した CG キャラクタが実世界の物体を避けながら歩き回る、実物体の色や形を変化させたりコピーや削除を行うといった様々な利用が期待できる。

このような AR システムの実現が難しい原因としては、まず実環境から実物体を復元した三次元形状モデル (以下、単にモデル) を生成することが容易ではないという点が挙げられる。特に実物体の観測に基づくモデル生成では、一般に複雑な操作や特殊なシステムが必要である³⁾。この問題を解決するために、インタラクティブな三次元形状復元の試みがなされている⁴⁾が、トラッキングにおける ARToolKit のように扱いやすく、復元精度の高い手法は確立されていない。

^{†1} 大阪大学 大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

^{†2} 大阪大学 サイバーメディアセンター

Cybermedia Center, Osaka University

AR オーサリングシステムに関する先行研究として、立石の AR ジオラマシステム⁵⁾がある。このシステムでは、単眼カメラとストローク入力を用いて、実物体の三次元形状を復元し、生成したモデルをジオラマのパーツとして実環境中に配置できる。モデルには物理エンジンを利用して実物体に近い挙動を実現し、ユーザはストローク入力によりモデルの移動、回転等の操作を行うことができる。

しかしこのシステムにおける問題として、違和感なく操作を行うためにはモデルの精度が不十分であるということが挙げられる。物体の三次元形状復元の精度を向上させることで、ユーザはより臨場感のある AR 環境を楽しむことができ、システムのエンターテインメント性の向上が期待できる。

物体の三次元形状復元手法としては、立石らの手法⁵⁾や Pan らの手法⁶⁾のような特徴点ベースと、不殿らの手法⁷⁾や Bastian らの手法⁸⁾のような領域ベースの手法が考えられる。特徴点ベースの手法は領域ベースの手法に比べて計算コストが小さく、周囲の色情報などの環境に影響されことなく三次元復元が可能である。また、個々の特徴点の三次元座標が直接得られるため、凹面を含んだ物体についても形状の復元が可能である。この手法は、色やテクスチャの多い、形状がはっきりとした物体に対して特に有効である。それに対し領域ベースの手法は背景と物体の色分布が異なる仮定を置くことで物体領域の抽出を行う手法であるが、これは物体の復元環境を制限することになる。領域ベースの手法は視体積交差法により形状復元を行うため、凹面を含む物体については復元が困難である。しかし、特徴点ベースの手法と異なり、色やテクスチャの少ない、形状があいまいな物体に対しても有効な手法である。

本研究では先行研究である立石の AR ジオラマシステムをベースとして、単眼カメラとマウスストロークを用いた、自然特徴点に基づく三次元形状復元システムを提案する。特に、先行研究の問題を解決する高精度な三次元形状復元手法を実現する。

2. システムの設計

本研究で提案する手法の目的は、AR ジオラマシステムにおいて、物体に含まれる特徴点と背景に含まれる特徴点の分類をインタラクティブに行い、高精度な三次元モデルをリアルタイムで復元することである。本節では AR ジオラマシステムにおける提案手法の三次元形状復元手法の構成と、それを踏まえた AR ジオラマシステム全体の設計について述べる。

2.1 設計方針

本節では提案システムの設計方針について述べる。提案システムは、従来システムにおけ

る形状復元手法のかわりに Pan らの ProFORMA⁶⁾を参考にした形状復元を行う。そこで、まず ProFORMA の形状復元手法について述べ、その後、提案システムにおける形状復元手法の設計方針について述べる。

2.1.1 ProFORMA の形状復元手法

ProFORMA は、従来システムと同様に特徴点ベースの復元手法を用いており、高精度な復元モデルの生成が可能なシステムである。従来システムへの実装が容易であり、また、従来システムにおける復元モデルの精度向上という本研究の目的に合致するものとして、ProFORMA の形状復元手法を参考にした。

ProFORMA は、トラッキングスレッドと形状復元スレッドという2つのスレッドからなるシステムである。トラッキングスレッドは、固定カメラの画像から復元対象の物体に含まれる特徴点を抽出し、ユーザがカメラの前で物体を回転させるに従って移動する特徴点を追跡する。復元スレッドでは、トラッキングスレッドから取得した特徴点の三次元位置を元に、特徴点ベースの形状復元を行う。

本研究において参考にしたのは、ProFORMA の復元スレッドにおける処理内容である。ProFORMA の復元スレッドのシステム構成を下記に示す。

- **四面体カービングアルゴリズム** ProFORMA は、復元対象の物体に含まれる特徴点群から四面体集合を構成し、特徴点群の凸包を求めて復元のための初期モデルとする。凸包とは、与えられた点群を含む最小の凸図形のことである。ただし、得られた凸包が物体の正しい三次元形状を表現しているとは限らない。そこで、特徴点群の可視性に基づく四面体のカービングを行い、物体が凹部を含む場合でも正しく三次元形状を復元する。
- **ドロネー四面体分割** 特徴点群から四面体集合を構成する際には、ドロネー四面体分割を用いる。ドロネー四面体分割は、三次元空間においてボロノイ図の隣接する領域同士を全て結ぶことで生成される。そしてその結果として四面体が形成され、これをドロネー四面体と呼ぶ。ボロノイ図は最も近い点がどれかにより空間を分割したものである。ドロネー四面体分割により、全ての特徴点を利用し、かつ重ならない四面体集合が生成される。
- **特徴点抽出** ドロネー四面体を構成する特徴点群は、トラッキングスレッドにおいてカメラ画像中から自動的に抽出される。トラッキングスレッドは、物体を固定カメラの前でユーザが回転させるに従って物体に含まれる特徴点が新しく見え始めるたびに、その三次元位置を形状復元スレッドへと渡す。
- **テクスチャマッピング** 四面体カービングにより物体の表面形状を表すポリゴンメッシュ

が得られるので、各メッシュに対してテクスチャマッピングを行う。ProFORMAは、物体が固定カメラの前で一定角度回転する度に、キーフレームを取得して保存している。各メッシュの法線方向に最も近い撮影位置のキーフレームをそのメッシュに対応するテクスチャとして利用する。

ProFORMAは、必要なデバイスが単眼カメラのみであり、固定したカメラの前で物体を回転させるという容易な操作によって物体の復元が可能である。また、特徴点を物体から自動的に抽出するため、物体に含まれる特徴点をユーザが指定する必要がない。さらに、凹面を含む物体の復元も可能であることなどが利点としてあげられる。しかし、特徴点抽出に基づいているため、果物やクッションのように色やテクスチャの少なく、形状があいまいな物体の復元は難しい。

2.1.2 バーチャル物体の復元手法の設計

本節では、上述のProFORMAの手法を参考にした上で、提案システムにおける三次元形状復元手法の設計方針について述べる。

復元手法は、四面体カービング、ドロネー四面体分割、特徴点抽出、そしてテクスチャマッピングの4つの処理からなる。

ProFORMAと同様に、四面体カービング、ドロネー四面体分割を用いて、物体の三次元形状を復元する。復元対象の物体に含まれる特徴点の抽出には、従来システムにおけるマウスストロークを用いたユーザインタラクションを導入する。最後に、ドロネー四面体分割により生成した物体のポリゴンメッシュに対しテクスチャマッピングを行い、バーチャル物体の復元を実現する。

ProFORMAはトラッキングスレッドにおいて特徴点を自動的に抽出するので、提案システムにおける三次元復元手法との主な差異は特徴点の抽出方法にあるといえる。

2.2 システム構成

提案システムへの入力として、実環境を観測するデバイスはカメラからの映像のみとする。カメラは入手しやすいデバイスであり持ち運びやすく、本システムに適している。入力をマーカレスなカメラ画像に限定するため、マーカや他のセンサを用いずにカメラの位置姿勢と実環境の幾何形状を推定する必要がある。本システムは一般にSimultaneously Localization And Mapping (SLAM) と呼ばれる手法を用いる。SLAMはカメラ位置と特徴点群等の位置の推定を同時に行う手法である。SLAMにより得られたカメラの位置推定結果は、実環境への復元モデルの重量表示に利用される。

ユーザからの入力はマウスやタッチペンによるストローク入力とする。ストローク入力は

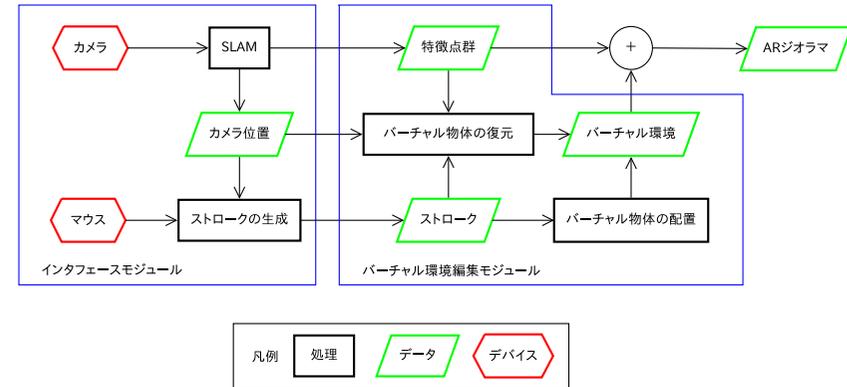


図1 提案システム全体の構成図
Fig.1 System flow

復元対象に含まれる特徴点群を指定するために利用する。これは、バーチャル物体をストローク入力で囲むという操作が直感的であるという考えに基づく。

形状復元システム内部でのバーチャル物体生成はSLAMで得られる特徴点群を元に行うものとする。SLAMだけではバーチャル物体に含まれる特徴点だけを抽出することが困難なため、ストローク入力を用いる。また、生成されたバーチャル物体の動作は物理モデルを利用して生成する。動作の指定もストローク入力によって行う。

上記のことから、提案システムはユーザからの入力とデバイスからの入力を受けるインタフェースモジュールと、デバイスとユーザからの入力を元にバーチャル物体を生成する、バーチャル環境編集モジュールに大別される。提案システム全体の構成を図1に示す。以下では、各モジュールの処理内容について述べる。

2.2.1 インタフェースモジュール

2.2.1.1 システムへの入力

インタフェースモジュールではストローク入力を認識し、物体が置かれる平面や、実環境から形状復元される物体の指定を行う。以降、ストローク入力に対応するステージの設定、バーチャル物体の配置、バーチャル物体の生成について述べる。

2.2.1.2 ステージの設定

ステージとは、ARジオラマを編集する際の基本となる平面である。バーチャル物体の移動や配置はこのステージを基準に行われる。本システムは最初に、ステージの設定を行うよ

うユーザに要求する。ステージの設定はストローク入力によって閉曲線を描くことで行う。ユーザが基準としたい平面上の特徴点群を閉曲線で囲むと、システムが閉曲線とカメラ画像を用いて平面を抽出する。そして平面上にステージの位置を示す正方形と法線を表示する。

2.2.1.3 バーチャル物体の復元

復元対象物体の指定は、ユーザがストローク入力により対象を閉曲線で囲むことで行われる。システムは囲まれた内部の画像から特徴点群を抽出し、それらを物体に含まれる特徴点としてバーチャル環境編集モジュールへ渡す。そしてバーチャル環境編集モジュールでは、渡された特徴点群をもとにバーチャル物体を復元する。ユーザのストロークによるインタラクティブな復元により、背景の色やテクスチャ等の条件に影響されることなく物体のセグメンテーションが可能である。

2.2.1.4 バーチャル物体の配置

バーチャル物体の配置は回転と平行移動の2段階で行う。ファイルを選択するとステージにバーチャル物体が配置される。それに対して回転と平行移動を行う。回転はバーチャル物体を中心とするような円弧を描くことで指定する。回転角は円弧の始点から終点への角度とする。回転軸は物体の中心を通り、ステージの法線と平行な軸とする。平行移動はバーチャル物体から配置したい地点までの経路を描くことで実現する。その経路を書き終えると、バーチャル物体はその経路を辿り平行移動する。

本システムでは実環境から三次元復元したバーチャル物体をARジオラマ内に自由に配置することが可能である。ユーザのストローク入力に基づいてバーチャル物体は回転したり、バーチャル環境内を移動することができる。また、それぞれが物理モデルによる接触判定を持ち、バーチャル物体同士は重なることはない。

2.2.2 バーチャル環境編集モジュール

ARジオラマではカメラが撮影している実環境にバーチャル環境が重畳して表示されている。バーチャル環境で描画されるものは次の3種類である。

- **特徴点** システムがカメラ画像から抽出する、追跡に適した点である。この特徴点を利用して環境と実環境の位置合わせが行われる。特徴点はシステムが自動で抽出し表示するものであり、ユーザが操作することはできない。しかしバーチャル物体の三次元形状復元を行う際に、ユーザの補助となる。
- **バーチャル物体** バーチャル物体はユーザのストローク入力により抽出された特徴点群に基づいて復元される。一度のストローク入力で復元可能なのはユーザが指定した時点でのカメラから見える部分のみである。そのため本システムではカメラを動かして新た

に抽出した特徴点をモデルに追加し、多視点からの復元を行った結果を統合することでより欠落の少ないバーチャル物体を復元する。これにより、どの視点からでも表示可能なバーチャル物体を復元することができる。また、提案システムと従来のARジオラマシステムにおける最も大きな差異は、ストロークによる特徴点抽出後の復元手法にある。2.1.2節で述べた手法を用いてバーチャル物体を復元する。バーチャル物体の復元処理の流れを図2に示す。

提案システムでは実環境から三次元復元したバーチャル物体をARジオラマ内に自由に配置することが可能である。上述のようにユーザのストローク入力に基づいてバーチャル物体は回転したり、バーチャル環境内を移動することができる。また、それぞれが物理モデルによる接触判定を持ち、バーチャル物体同士は重なることはない。

- **ステージ** ステージは無限の広さを持つ平面であり、上述のようにユーザが実環境中の物体をステージと指定したもとのから生成される。指定されるものとしては、机や床面などを想定している。システムは指定された対象から平面を検出し、それを基準面とする。

3. システムの実装

本節では提案するシステムの実装について述べる。インタフェースモジュールにおいて、ユーザのストローク入力により復元対象の物体に含まれる特徴点が抽出されたという前提のもとで、物体の三次元形状復元手法の実装を行う。以降、提案システムの形状復元部をバーチャル物体復元モジュールと呼び、これに関する実装の詳細を説明する。

本システムでは、カメラの位置姿勢推定と特徴点マップの生成を、KleinらのParallel Tracking And Mapping (PTAM)²⁾によって行う。特徴点マップとは、PTAMがカメラ画像中から特徴点の抽出を行い、作成する特徴点群の三次元地図のことである。PTAMはシステムの他の部分とは独立に動作する。

3.1 バーチャル物体復元モジュールの実装

本節では、PTAMの生成する特徴点マップをもとにした、実物体の三次元形状復元について述べる。2.1.2節の設計をふまえ、三次元形状復元手法は特徴点の抽出、ドロネー四面体分割、四面体カービングからなり、得られた形状モデルにテクスチャマッピングを施すことでバーチャル物体を復元する。以降で各処理の実装の詳細を説明する。

3.1.1 特徴点の抽出

復元対象の物体に所属している特徴点の抽出はストローク入力によって行う。カメラが撮影した画像中でストロークの内部に描画される特徴点は復元対象に所属していると考え、

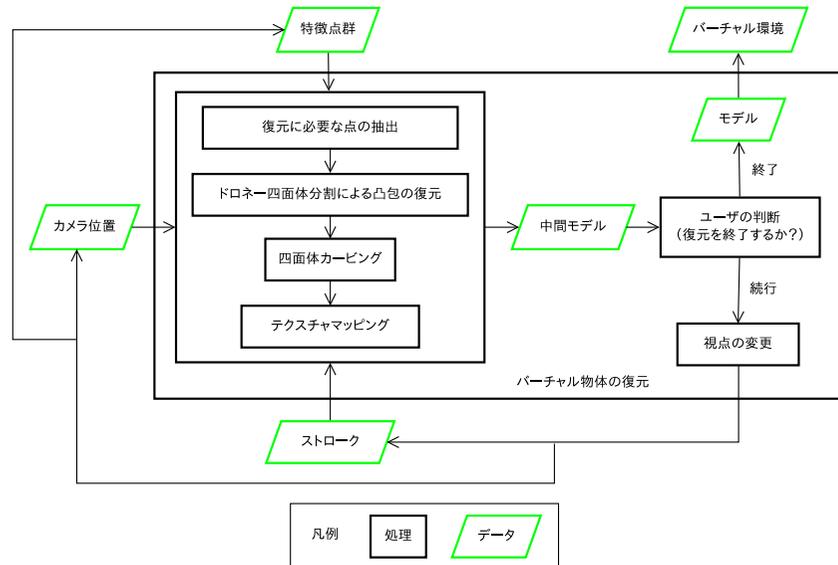


図2 バーチャル物体の復元処理

Fig. 2 Virtual object reconstruction process

ただし、本研究ではストローク入力による特徴点の抽出は完了しているものとして、取得した特徴点から三次元形状復元を行う部分を新たに実装する。

特徴点ベースの復元手法では、特徴点の位置情報の精度が復元モデルの精度に大きく影響する。また、本システムにおいて抽出した特徴点の位置情報の精度は、PTAMの特徴点のトラッキング精度に依存する。そのためPTAMが取得する特徴点群の中から信頼性の高い特徴点のみを選択して、形状復元に利用する。信頼性の判定基準としてPTAMの内部変数である Outlier Count を利用する。Outlier Count は、バンドル調整法により特徴点の再投影誤差を求める際に、その二乗誤差が閾値を超えた回数を表す。本システムにおいて利用する特徴点は、Outlier Count が10以下であるものに限定した。

システムは、ユーザが過去に撮影した方向から発見できる特徴点のみ抽出できるので、位置推定精度の良い特徴点を多数得るためには、多方向から対象を撮影する必要がある。そのためにユーザは1つのバーチャル物体に対して特徴点の抽出と形状復元を複数回実行できる。カメラ位置を変更して新たに取得した特徴点は全て復元対象の物体に含まれるものとみ

なし、システムはそれらを用いて復元を行う。最終的にユーザの望む精度が得られた時点でバーチャル物体の復元が終了する。

3.1.2 ドロネー四面体分割

抽出された特徴点群の凸包を求め、メッシュの生成を行う。メッシュの生成はドロネー四面体分割によって行う。ドロネー四面体分割の実装には CGAL⁹⁾ を利用した。

3.1.3 四面体カービング

ドロネー四面体分割により求めた特徴点群の凸包から、物体の正しい三次元形状を得るために、物体内に存在しない四面体を除去する必要がある。

効率的な四面体除去の手法として、特徴点の可視性に基づく判定を利用する。四面体を構成する四つの三角形面それぞれに対して、カメラ画像中に映っている特徴点からカメラへの全光線に対してテストを行う。その際、光線と交差する面、すなわちカメラ画像中において三角形面の内部に特徴点が映る面を1つでも含む四面体が除去の対象となる。特徴点の可視性に基づくモデル内の各三角形面の存在判定は、次式によって表現する。

$$P_{exist}(T_i|v) = \prod_v (T_i|R_{j,k}) = \prod_v (1 - Intersect(T_i, R_{j,k})) \quad (1)$$

$$Intersect(T_i, R_{j,k}) = \begin{cases} 1 & (R_{j,k} \text{が } T_i \text{ と交差するとき}) \\ 0 & (\text{それ以外のとき}) \end{cases} \quad (2)$$

T_i はモデル内の三角形、 j は復元に利用するキーフレームの番号、 k は特徴点番号、 $R_{j,k}$ はキーフレーム j におけるカメラ中心から特徴点 k への光線、 v は特徴点 k とそれが映っているキーフレーム j の組み合わせ (j, k) の全通りとする。式(1)において、 P_{exist} は光線が1つでも三角形面と交差している場合は0となり、全く交差しない場合は1となる。

この式による判定は、ノイズが原因でモデルの表面よりわずかに下に存在する特徴点によって、実際の表面が削り取られるために、滑らかなモデルの表面を生成する。そこで、表面に発生するノイズを考慮に入れ、より滑らかなモデル表面を生成することができる。確率的カービングアルゴリズムを使用する。

特徴点は、モデル表面の三角形 T_i に向けられた光線 $R_{j,k}$ に沿った、分散 σ^2 のガウシアンノイズによって除去された三角形に注目することで取得できる。この時面の存在する確率は次式のように表現できる。

$$P(L_k|R_{j,k}, T_i) = \int_{-\infty}^{l_k} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \quad (3)$$

$x = 0$ を $R_{j,k}$ と T_i の交点として定義し, x をカメラ方向を正とした時の $R_{j,k}$ に沿った符号付き距離, l_k を $x = 0$ から特徴点 L_k までの符号付き距離とする.

式 (3) より, 確率的なカービング式が導かれる.

$$P_{exist}(T_i|v) = \prod_v P_{exist}(T_i|R_{j,k}) \quad (4)$$

$$P_{exist}(T_i|R_{j,k}) = \begin{cases} P(L_k|R_{j,k}, T_i) & (R_{j,k} \text{ が } T_i \text{ と交差するとき}) \\ 1 & (\text{それ以外のとき}) \end{cases} \quad (5)$$

本システムの実装では $P_{exist} > 0.1$ の時, T_i は存在すると見なし, それ以外の場合には存在しないとして T_i を含む四面体を除去の対象とする.

各三角形に対しての確率式において, 全ての可視の光線を用いる必要はない. なぜなら三角形と交差しない光線は, 面の存在確率に影響を与えない一方で計算コストを増加させるからである. 三角形と交差している可能性のある光線集合を得るために, 物体の復元に使用される視点毎のカメラフレーム (キーフレーム) において可視である特徴点に対して, スクリーン座標系のレンジツリー⁹⁾ を生成する. そして各四面体の各三角形をキーフレームに投影する. キーフレームのレンジツリーを用いて, 投影された三角形のバウンディングボックス内に存在する特徴点を探し, その光線に対してのみ確率式を適用する. 各四面体は他の四面体と面を共有しており, それらの面は同じ確率スコアを持つので, 確率の計算を 1 回ですませることができる.

モデル表面の四面体を除去するという事は, それと隣接する四面体をモデル表面の四面体であるとみなすことと同義なので, 効率的な再帰カービングアルゴリズムを設計することができる. カービング関数はモデル表面の四面体を入力として, それらは除去すべき四面体であるかテストされる. もしその四面体が除去された場合, その四面体と隣接する四面体を入力としてカービング関数は自身を再帰的に呼び出す. そして, 四面体が除去されなかった場合は, その四面体はモデル表面に存在しているのでカービング関数は終了する. 復元対象の物体に含まれる特徴点群を四面体分割した結果, 得られる凸包の表面に存在する全ての四面体に対してこの関数を適用することで, 物体の正しい表面形状が得られる. 同じ四面体のテストを 1 回しか行わず, かつ隣接する四面体同士が共有する面の存在確率の計算を 1 回ですませることによって, 四面体除去のテストは効率的に実行される.

本システムは, ユーザが単眼カメラを用いて自由な視点から物体を撮影してバーチャル物体の復元を行う. 固定カメラの前でユーザが物体を一回転させるというような, 復元の際の

操作が定められている ProFORMA とは異なり, 四面体の存在確率がどの段階で決定されるか分からない. カービング機能が正しく動作しているかを復元途中でユーザに提示するために, 1 つの四面体に対して存在確率の計算を行う回数に上限を設ける. そしてその計算結果が閾値を超えているかどうかを判断基準としてカービングを行う. これらのパラメータは復元対象物体や利用環境によってチューニングする必要がある.

3.1.4 テクスチャマッピング

可視性に基づく四面体カービングの終了後, 表面の全メッシュに対してテクスチャマッピングを行う. 復元に利用した視点におけるカメラ画像 (キーフレーム) は全て保存しているので, 各メッシュに対するテクスチャには, その三角形の投影面積が最大となるキーフレームを選択して使用する. そのためまず, 全メッシュをキーフレームに投影して, キーフレームにメッシュが映っているかどうかの判定を行う. メッシュが映っているキーフレームに投影した際の面積を判断基準として, 面積が最大となるキーフレームを用いるものとする.

4. 動作実験

本節では, バーチャル物体復元モジュールを実際に動作させた結果について述べる.

4.1 実験内容

物体を複数視点から単眼カメラで撮影し, その結果復元したバーチャル物体の精度を観察する. 四面体カービングの動作を確認するために, 凸な物体と凸でない物体に対してそれぞれ三次元形状復元を試みる. 凸な物体に関しては, 立石らの AR ジオラマにおける, 従来の復元手法によるバーチャル物体の精度と比較する.

なお, 提案手法によるバーチャル物体の復元を行うにあたり, 復元手法の妥当性を確認するために物体の特徴点が確実に抽出できる環境を用意した. 具体的には, 復元に利用するカメラ画像中に物体外の特徴点が含まれないように, 背景を無地に限定した.

また, 実験に用いる機材として PC (AMD Athlon(tm) 64 X2 Dual Core Processor 3800+, GeForce 7800 GTX, メモリ 4GByte) と, カメラ (Point Grey Flea3, 解像度 640 × 480, フレームレート 30fps) を利用した.

4.2 実験結果

4.2.1 凸な実物体に対するバーチャル物体復元

三次元形状復元の際に四面体カービングが必要ない, 凸な物体である,

- ルービックキューブ (図 3(a) 参照)



(a)



(b)

図3 凸形状の復元対象 (a) ルービックキューブ (b) アルミ缶

Fig.3 Convex reconstruction target (a) Rubik's Cube (b) Aluminum can

- アルミ缶 (図3(b) 参照)

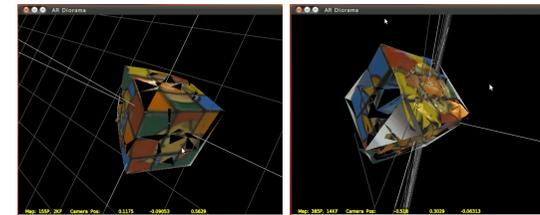
をそれぞれ従来手法と提案手法により復元した。

従来手法によるバーチャル物体を図4に、提案手法によるバーチャル物体を図5に示す。提案手法によるバーチャル物体における紫色のメッシュは、その時点までに復元に利用したカメラ画像中には映らない部分であるために、テクスチャが貼り付けられていないことを示す。

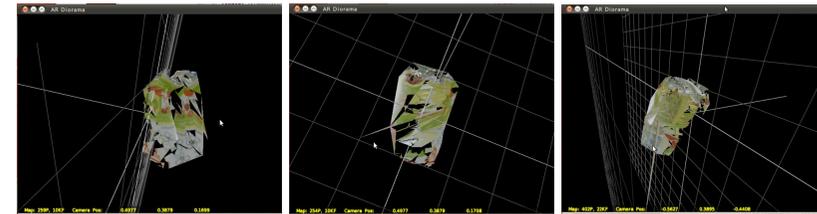
従来手法によるバーチャル物体はメッシュが欠けている部分が多く、物体の三次元形状の復元精度が低いため、複数の視点から観察すると違和感がある。それに対し、提案手法によるバーチャル物体は三次元形状の精度が向上していることが確認できた。

4.2.2 凸でない実物体に対するバーチャル物体復元

三次元形状復元の際に四面体カービングが必要な物体として、L字型の箱 (図6 参照) を提案手法により復元した。図7は、カービング後のバーチャル物体を2つの視点から観察した様子である。カービング後の形状は、本来のL字型に近づいていることが分かるが、削り残しの部分が多く見られる。実物体の形状に合わせてカービングが動作することは確認し



(a)



(b)

図4 従来手法によるバーチャル物体 (a) ルービックキューブ (b) アルミ缶

Fig.4 Reconstruction results of original method (a) Rubik's Cube (b) Aluminum can

たが、高精度なバーチャル物体を復元することはできなかった。

5. おわりに

本研究では、立石のARジオラマシステムをベースに、PanらのProFORMAの形状復元アルゴリズムを組み合わせることで、従来システムにおける復元モデルの精度向上を検討した。本手法では、PTAMが取得する復元対象の物体に含まれる特徴点をもとに四面体分割を行うことで物体の凸包を求め、さらに特徴点の可視性に基づく面の存在確率の判定を行うことで、物体の表面形状を復元する。最後に、多視点からのテクスチャマッピングを行うことで三次元復元モデルを生成する。実装の結果、従来システムに比べて高精度な三次元形状復元を行えることを確認した。しかし、凸でない物体を復元する場合にカービングが上手く動作しないことも明らかになった。

今後の課題としては、特徴点の可視性の判定基準を見直し、カービングを正しく動作させることが挙げられる。また、ストローク入力を用いて復元対象の物体に含まれる特徴点の抽出を行うインタフェースモジュールと統合すること、復元した三次元モデルをARジオラ

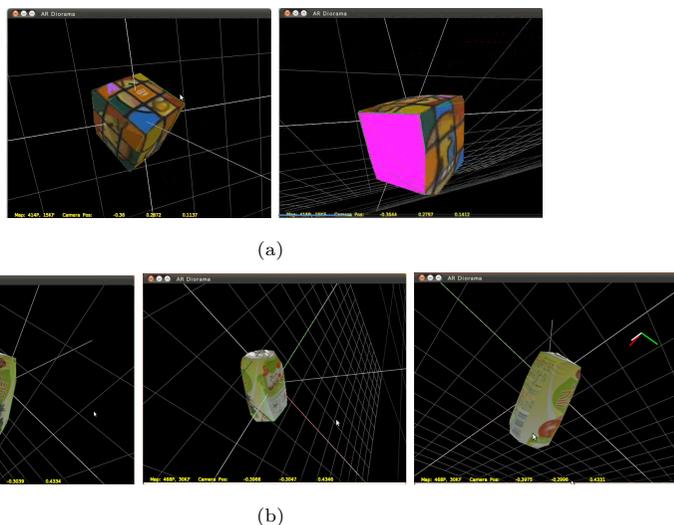


図 5 提案手法によるバーチャル物体 (a) ルービックキューブ (b) アルミ缶
Fig.5 Reconstruction results of our method (a) Rubik's Cube (b) Aluminum can



図 6 凸形状でない復元対象の物体
Fig.6 Non-convex reconstruction target

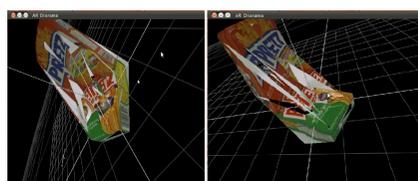


図 7 提案手法によるバーチャル物体 (カービング後)
Fig.7 Reconstruction result of our method after carving

マのコンテンツとして操作, 配置できるようにすること, AR ジオラマシステムの機能拡張を行うことなどが挙げられる.

参 考 文 献

1) H. Kato and M. Billinghurst: "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," Proc. of the 2nd International

Workshop on Augmented Reality, pp.85-94, 1999.
2) G. Klein and D. Murray: "Parallel Tracking and Mapping for Small Workspaces," Proc. of the 6th IEEE International Symposium on Mixed and Augmented Reality, pp.1-10, 2007.
3) Y. Yemez and C.J. Wetherilt: "A Volumetric Fusion Technique for Surface Reconstruction from Silhouettes and Range Data," Computer Vision and Image Understanding, Vol.105(1), pp.30-41, 2007.
4) M. Brown, T. Drummond, and R. Cipolla: "3D Model Acquisition by Tracking 2D Wireframes," Proc. of the 11th British Machine Vision Conference, 2000.
5) 立石 昂裕: "単眼カメラとストローク入力を用いた三次元復元による AR ジオラマシステム", 大阪大学大学院情報科学研究科修士学位論文, 2010.
6) Q. Pan, G. Reitmayr, and T. Drummond: "ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition," Proc. of the 20th British Machine Vision Conference, 2009.
7) 不殿 健二, 佐藤 智和, 横矢 直和: "ハンドヘルドカメラを用いた撮影位置指示機能を有するインタラクティブ 3次元モデリングシステム", 画像の認識・理解シンポジウム講演論文集, pp.243-250, 2005.
8) J. Bastian, B. Ward, R. Hill, A. Hengel and A. Dick: "Interactive Modelling for AR Applications," Proc. of the 9th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.199-205, 2010.
9) Computational Geometry Algorithms Library: "CGAL - Computational Geometry Algorithms Library," <http://www.cgal.org/>. (最終閲覧日 2011 年 2 月 6 日)