

## AR ジオラマにおける前景抽出インタフェースを用いた 三次元形状復元手法

馬 勝 淳 史<sup>†1</sup> 間 下 以 大<sup>†1,†2</sup>  
清 川 清<sup>†1,†2</sup> 竹 村 治 雄<sup>†1,†2</sup>

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する技術が身近なものとなり始めている。本研究では、対象の実物体の三次元形状復元、ストローク入力による動作記述が可能である AR ジオラマと呼ぶシステムを提案していた。しかし、このシステムが復元する三次元モデルは復元精度が良くないという問題があった。本研究では、AR ジオラマの三次元形状復元に関して新たな手法を検討する。提案手法では、色情報を用いて領域分割を行うことによって、三次元物体形状復元を行う手法を用いる。特徴点情報ではなく、色情報を用いることで、三次元物体形状復元のテクスチャ依存性を低くする。ユーザからのマウスによるストロークを入力とし、ストロークによって復元したい物体を指定してシルエットを抽出し、得られたシルエットとカメラ自己位置姿勢から三次元形状の復元を行う。実装の結果、従来手法では復元不可能であったテクスチャが少ない物体の三次元形状復元ができること、従来手法と比べて精度のよい三次元形状モデルの生成ができることを確認した。

### A 3D Reconstruction Method Using Foreground Extraction Interface for an AR Diorama System

ATSUSHI UMAKATSU,<sup>†1</sup> TOMOHIRO MASHITA,<sup>†1,†2</sup>  
KIYOSHI KIYOKAWA<sup>†1,†2</sup> and HARUO TAKEMURA<sup>†1,†2</sup>

Recently, augmented reality (AR) technologies, which overlay computer generated information onto the real world, have been widespread. We have previously proposed AR Diorama System that can construct AR contents by using a handheld camera and stroke input, which enable the user author behaviors. However, there were problems in virtual object creation and object behavior authoring due to the low precision of the 3D reconstruction. In this study, we

consider to employ a better 3D reconstruction method to improve AR Diorama System. Our approach is based on real-time region segmentation and silhouette carving. Region segmentation uses color information of the real object of concern so that the new system does not require texture information unlike a feature point tracking based approach used in the previous system. A user first draw a stroke on screen to specify a real object to reconstruct, then its silhouette is automatically extracted and tracked in real-time by using a graph cut algorithm. A 3D model of the real object is gradually reconstructed on-the-fly by using silhouette and camera pose data associated with multiple keyframes. Experimental results show that our method can reconstruct better 3D models in terms of precision compared to the previous system regardless of texture information, though a camera must be moved around the object carefully to acquire a good result.

### 1. 序 論

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する技術が身近なものとなり始めている。AR 技術が普及しつつあるのは、加藤らによる ARToolKit<sup>1)</sup> などの扱い易い、マーカベーストラッキングを行うライブラリの登場によるところも大きい。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 も数多く開発されており、PTAM<sup>2)</sup> などのマーカレストラッキングを行う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、それに対する振る舞いを AR 環境にしながら付加・編集していく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。このような AR システムの実現が難しい原因としては、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。

AR 環境においてモデルやその振る舞いを記述するシステム (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 Computer Graphics (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

<sup>†1</sup> 大阪大学 大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

<sup>†2</sup> 大阪大学 サイバーメディアセンター

Cybermedia Center, Osaka University

コピーや削除を行う、といった様々な利用が期待できる。

このようなシステムの先行研究として、立石らの AR ジオラマシステム<sup>3)</sup>がある。立石らのシステムでは、実環境から復元したモデルをインタラクティブに生成、動作を記述できる特徴を持つ。しかし、モデル精度が良くないという深刻な問題がある。重畳表示させるモデルの精度が悪ければ、リアリティに欠ける、衝突判定動作などが正確にできないなどの問題がある。つまり、AR システムを構築する上で重畳表示させるモデルの精度は、非常に重要である。

本稿では、上述の問題を解決する三次元形状復元手法を提案する。提案手法では、カメラ映像からの色情報を用いて、復元したい物体の領域（以下、前景）と復元したい物体以外の領域（以下、背景）の領域分割を行う。そして、前景のシルエットとカメラの自己位置姿勢推定の結果から視体積交差法によりボリュームデータを生成する。領域分割では、マウスからのストローク入力によって前景を指定する。生成されたボリュームデータに対して、入力画像をテクスチャとしてマッピングすることでモデルを生成する。

以下、2 節では従来手法の概要、3、4 節では本研究の提案手法の設計方針、5 節では実装方法について述べる。6 節では動作実験の結果、7 節では結論と今後の課題について述べる。

## 2. 単眼カメラとマウスストロークを用いた三次元形状復元による AR ジオラマシステム

本節では、従来の立石らの AR ジオラマの目的と内容を述べる。AR ジオラマの目的は、簡単なインタフェースでユーザの身近な環境を AR 環境として構築・操作を可能にすることである。そのため、システムに用いるインタフェースや入力デバイスなどは容易な入力動作を必要とする。例えば、キーボードを用いずにタッチパネルを用いて画面に直接ストロークを行うことで、直感的な操作をする。また、AR ジオラマは事前の準備の必要なしに様々な環境で利用できることを想定している。従って、環境中にマーカを要するマーカベースのトラッキング手法は適さず、マーカレスで動作する必要がある。

### 2.1 従来の AR ジオラマの概要

従来の AR ジオラマは、SLAM を用いて求まる物体の特徴点から、特徴点群のドロネー三角形分割を行うことでモデルを生成する。生成された物体をユーザからの入力とデバイスからの入力を組み合わせることで、実環境に生成モデルを配置し、移動や回転の動作を記述する。図 1 は移動の動作記述の例である。

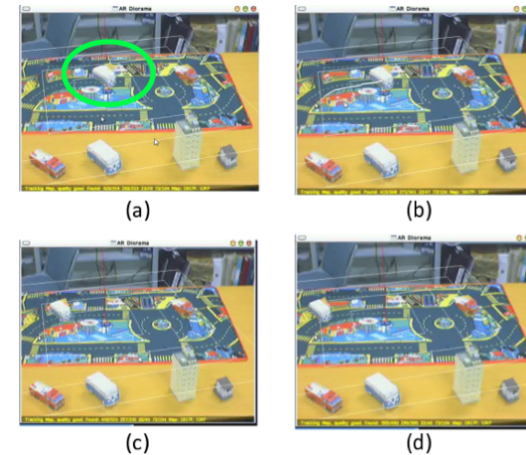


図 1 従来の AR ジオラマにおけるストロークを用いたシーン編集<sup>3)</sup>  
(a) モデル指定 (b) 動作記述 (c)(d) モデルの平行移動  
Fig. 1 Authoring the scene using stroke for conventional AR Diorama<sup>3)</sup>  
(a)Designation (b)Authoring behaviors (c)(d)Translation

### 2.2 従来の AR ジオラマの問題

従来の AR ジオラマではモデル生成とモデルの動作記述を可能とする AR システムを提案している。しかし、モデルの復元手法に 2 つの問題がある。1 つ目は復元対象の物体に特徴点が存在しない場合、復元が困難もしくは不可能であるという問題である。2 つ目はカメラ自己位置姿勢などの誤差により、モデル統合がうまくいかない問題である。上記の問題から、モデルの復元精度が悪い場合がある。モデルの精度がよくない場合、動作記述が正確に行うことができない、物理シミュレーションによる接触判定を正確にできないなどの問題がある。上記の問題を解決するために、本研究では新たな三次元物体形状復元手法を検討する。

## 3. 領域分割に基づく三次元形状復元手法

本節では、本研究で参考にしたシステムの内容、本システムの三次元形状復元手法の構成について述べる。

### 3.1 参考システム

本研究では、従来の AR ジオラマを基盤とした上で、Bastian らのシステム<sup>4)</sup>の三次元

形状復元手法を参考にした、Bastian らのシステムの特徴を以下に示す。

- (1) 色情報を用いた領域分割結果を利用した視体積交差法による物体形状復元
- (2) 単眼カメラとマウス入力による形状復元
- (3) PTAM によるカメラ自己位置姿勢推定
- (4) 実環境に配置されたバーチャル物体の拡大縮小，移動，回転，オクルージョン処理。

提案システムは，従来の AR ジオラマをベースに開発する。従来の AR ジオラマでも，カメラトラッキングには PTAM を用い，インターフェースは単眼カメラとマウスストロークのみである。これらの類似点から，他のシステムと比べ，Bastian らのシステムは AR ジオラマへの移植性が高いと言える。

また，従来の AR ジオラマでは，物体のテクスチャから得られる特徴点情報のみを用いて，モデルを復元する。そのため，特徴点が少ない物体 (e.g. 風船のような特徴点を得づらい物体) の復元は困難である。そこで，色情報を用いて三次元形状復元を行うことで，特徴点情報に依存することなくモデルの復元が可能である。

以上の内容を鑑みて，本研究では Bastian らの手法を参考に AR ジオラマのモデル復元手法の構築を行う。

### 3.2 システム構成

AR ジオラマ全体の構成図を図 2 に示す。システムは，以下の 3 つのモジュールから構成されている。カメラパラメータを推定するモジュール，マウスを用いたユーザインタラクションモジュール，カメラ映像とマウスストロークから三次元物体を抽出する形状復元モジュールである。本研究では，物体のモデルの復元が目的である形状復元モジュールの実装を中心に行う。

パラメータ推定モジュールでは，SLAM によるカメラの自己位置姿勢推定を行う。

ユーザインタラクションモジュールでは，ユーザがマウスストロークによって復元する物体を指定する。また，形状復元モジュールの復元結果が誤っている場合はユーザが対話的に修正できる。本システムでは，領域を追加するストロークを “Inclusion brush”，領域を削除するストロークを “Exclusion brush” と呼ぶ。Inclusion brush, Exclusion brush は形状復元モジュールの処理に用いられる。本システムでは，Inclusion brush と Exclusion brush の切り替えなどをボタンを使わずに，ストロークによる文字入力を介して行う。このモジュールでは，ストローク結果から得られた文字を認識し，認識結果を他のモジュールに渡すことで，システムの動作を制御する。

形状復元モジュールでは，上述したモジュールの処理結果を用いて，復元するモデルを特

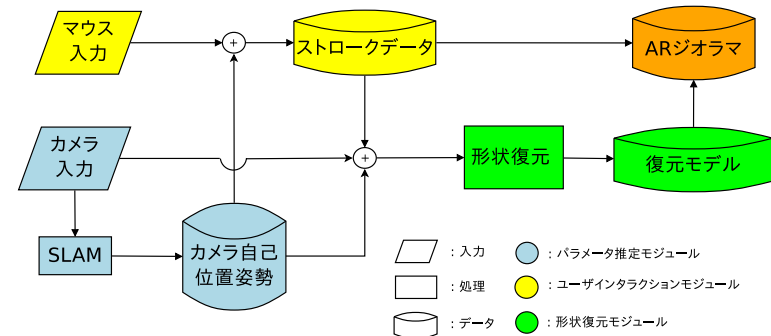


図 2 AR ジオラマの構成  
Fig.2 AR Diorama structure

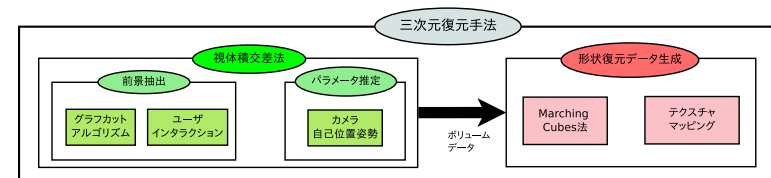


図 3 形状復元モジュールの構成  
Fig.3 Reconstruction module structure

定し，モデル形状を復元する。復元手法は，視体積交差法，サーフェスレンダリングとテクスチャマッピングから構成される。視体積交差法を用いるには，前景抽出が必要となる。そこで，領域分割を行うグラフカット<sup>5)</sup>とユーザによるインタラクションを行うことで前景抽出を実現する。視体積交差法によって生成されたボリュームデータに対して，サーフェスレンダリングを用いて表面形状データを復元する。復元された表面形状データに対して，テクスチャマッピングを行い，モデルの復元を実現する。

## 4. 物体復元モジュールの構成

本節では，三次元形状復元を行う形状復元モジュールの構成を述べる。形状復元モジュールの全体の構成図を図 3 に示す。

#### 4.1 視体積交差法

本節では、復元したい物体のボリュームデータを生成する視体積交差法の構成について述べる。視体積交差法とは、複数のカメラから得られる多視点の画像から対象物体の三次元形状復元を行なう手法である。まず対象物体を各視点から見たときのシルエットを撮影対象の世界座標系で表される三次元空間（対象空間）に再投影し、視点の位置を頂点、シルエットを断面とする錐体を生成する。これを視体積と呼び、対象物体が存在し得る部分空間を表す。多くの視点からの視体積の共通部分を求めることで、対象モデルの復元を行う。

##### 4.1.1 前景抽出

本研究における前景抽出とは、ユーザが復元したい物体のシルエットを抽出することを指す。グラフカットを用いた領域分割とユーザインタラクションによるシルエットの修正によって構成される。グラフカットによる領域分割は、最初の領域分割とそれ以降の領域分割で異なる式を用いる。ユーザが指定した物体を前景と背景に分割するための最初の領域分割処理を“Initial segmentation”と呼び、分割後に自動でセグメンテーションを行う領域分割処理を“Dynamic segmentation”と呼ぶ。Dynamic segmentationは、グラフカットに距離変換による空間距離画像を用いる。グラフカットによる自動的な領域分割とユーザ自身による手動の領域分割を組み合わせることで、精度の良い前景抽出を行うことを考える。

##### 4.1.1.1 グラフカット

画像  $P$  に対する各画素を  $p \in P$  とし、画素  $p$  に割り当てるラベルを  $L_p \in L$  とする。グラフカットでは、エネルギー関数式を式 (1) のように定義する。

$$E(L) = \lambda \cdot \sum_{p \in P} R(L_p) + \sum_p \sum_{q \in N(p)} B(p, q) \cdot \delta(L_p, L_q) \quad (1)$$

$$\delta(L_p, L_q) = \begin{cases} 1 & \text{if } L_p \neq L_q \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

ここで、 $L$  は画素に割り当てるラベル、 $\lambda$  はデータ項と平滑化項のパラメータ係数である。 $R(L_p)$  は画素  $p$  がラベル  $L_p$  である確率が高ければ値が小さくなるような関数として定義する。 $B(p, q)$  は隣接画素である  $p, q$  の輝度値が近ければ値が大きくなるような関数として定義する。グラフカットではエネルギー関数  $E(L)$  を最小にするようなラベル割り当てを計算することで、画像の前景と背景の領域を分割する。エネルギー最小化問題に対して、Boykovらは min-cut/max-flow algorithms<sup>6)</sup> を利用してエネルギーを最小化する方法を提案している。彼らの手法では、問題設定に合うようにグラフを作成し、そのグラフの min-cut を

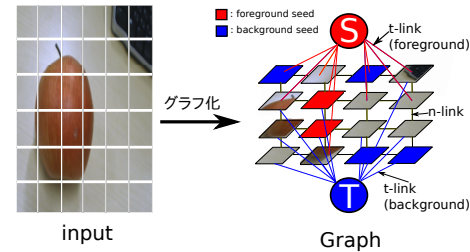


図4 グラフカットにおける画像のグラフ化  
Fig. 4 Conversion from image to graph for Graph Cuts

表1 グラフのエッジに与えるコスト

Table 1 Edge cost for graph

エッジ	コスト	場合分け
$\{p, q\}$	$B\{p, q\}$	$\{p, q\} \in N$
$\{p, S\}$	$\lambda \cdot R_b$ $K$ $0$	$p \in P, p \notin O \cup B$ $p \in O$ $p \in B$
$\{p, T\}$	$\lambda \cdot R_f$ $0$ $K$	$p \in P, p \notin O \cup B$ $p \in O$ $p \in B$

求めることで、エネルギー関数の最小化を行う。この min-cut は max-flow algorithm を用いることにより、エネルギー関数の最小値を効率的に計算することが可能である<sup>7)</sup>。入力画像から図4のようにグラフを作成し、min-cut/max-flow algorithms を用いてグラフを分割する。グラフは、画像の各画素に対応したノードと source と sink と呼ばれるターミナルからなる。source (S) または sink (T) からノードにつながるエッジを t-link と呼び、各ノードをつなぐエッジを n-link と呼ぶ。グラフのエッジのコストは、表1に従い設定する。

表1のエッジに与えるコストは式 (3) - 式 (6) で与える。

$$R_f = -\ln \Pr(I_p|O) \quad (3)$$

$$R_b = -\ln \Pr(I_p|B) \quad (4)$$

$$B(p, q) = \exp(-\beta(I_p - I_q)^2) \cdot \frac{1}{\text{dist}(p, q)} \quad (5)$$

$$K = 1 + \max_{p \in P} \sum_{q \in N(p)} B(p, q) \quad (6)$$

$O$  は物体、 $B$  は背景を意味する。 $I_p$  は画素  $p$  の輝度値である。通常のグラフカットの場合、ユーザは一部の画素に対して  $O, B$  を入力する。このユーザが指定するピクセル情報を seed と呼ぶ。しかし本システムでは、ユーザに背景 seed と前景 seed を与えるのはインタラクションの手間がかかると考え、前景 seed のみを与える。 $\Pr(I_p|O), \Pr(I_p|B)$  は seed 以外の画素の t-link (sink, source に繋がるエッジ) に設定する物体と背景の尤度である。 $\text{dist}(p, q)$  は画素  $p, q$  のユークリッド距離を用いる。



#### 4.1.1.2 ユーザインタラクション

Inclusion brush では領域を追加することができる。Inclusion brush は、マウスストロークとシルエットを結合し、その内部にある領域を追加する。Exclusion brush では領域を除くことができる。Exclusion brush は、シルエットをマウスストロークで分断し、面積が最大となるシルエットの領域のみを保持する。いずれもマウスを用いたユーザインタラクションによって、領域分割結果を修正できる。

#### 4.1.2 パラメータ推定

パラメータ推定とは、カメラの内部および外部パラメータを推定する処理である。抽出した前景シルエットをカメラ映像に再投影するためにカメラパラメータが必要となる。外部パラメータには自己位置姿勢が含まれ、内部パラメータにはカメラの投影行列が含まれる。外部パラメータは SLAM による特徴点群を用いて推定する。内部パラメータはあらかじめキャリブレーションボードを用いてキャリブレーションを行い、推定する。これらはパラメータ推定モジュールで実行される。

#### 4.1.3 ボリュームデータ生成

前景のシルエットとカメラのパラメータを用いて、ボリュームデータを生成する。この処理では、視体積交差法である Space Carving Method (SCM) を用いて三次元形状復元を行う。SCM では、対象空間の各ボクセルを各カメラの画像平面に投影したとき、全てのシルエットの内部に投影されるボクセルを対象物体の内部と判定し、一つ以上のシルエットの外部に投影されるボクセルを対象物体の外部の点と判定する。対象空間はすべてのカメラの視野に共通に含まれる直方体領域として設定する。上述のように、この手法では得られたシルエット画像のカメラの自己位置姿勢が既知である必要がある。シルエット画像とカメラの投影行列を用いることで、ボクセル空間を削っていく。ボクセル空間を削り出す処理を“カービング”と呼ぶ。そして、最終的に残されたボクセル空間が復元すべき三次元形状モデルとなる。通常の SCM の処理内容は式 (7) に従う。

$$v_t^i = v_{t-1}^i f(I_t^\alpha(\mathbf{P}_t W(v^i))) \quad (7)$$

$v_t^i$  は  $t$  フレーム目の  $i$  番目のボクセルを指す。  $v_0^i = 1.0$  で初期化を行う。  $\mathbf{P}_t$  は射影行列、  $W(\cdot)$  は世界座標系からカメラ座標系への変換を意味する。  $f(\cdot)$  は変数にラベルを取る。  $f(0)$  ならば 1、  $f(B)$  ならば 0 を返す。また SCM では一度ボクセルを削ると、そのボクセルは常に空になる。よって自己位置姿勢が安定していない場合、残すべきボクセルが削られてしまう可能性がある。そこで、複数視点からの処理結果の vote を取ることで、自己位置姿勢推定のエラーによる削りすぎを防ぐ。このときの処理内容は式 (8) に従う。

$$v_t^i = \text{sgn}(v_{t-1}^i) \left( v_{t-1}^i + \frac{2f(I_t^\alpha(\mathbf{P}_t W(v^i))) - 1}{n} \right) \quad (8)$$

$\text{sgn}(\cdot)$  は符号関数であり、ボクセル値が正 (負) なら 1 (0) を返し、ボクセル値が 0 (空のボクセル) ならば 0 を返す。またパラメータ  $n$  は SCM によってボクセルが削られるまでの vote 数を表す。

#### 4.2 形状復元データ生成

物体の三次元形状はサーフェスレンダリングによって復元する。さらに、テクスチャをはりつけることで写実的な形状復元データを生成する。ただし、一般には復元された物体に対して、単一視点からのマッピングでは側面などに正しいテクスチャがはられない。そこで、複数視点から得られるテクスチャをモデルに復元することで、生成された三次元形状に対して、正しいテクスチャをはりつけることを考える。メッシュに対して、どの視点からのテクスチャをはりつけるかの判定は、カメラ位置やメッシュの情報などから決定する。

### 5. システム実装

提案システムでは、高速な処理が必要となる。そこで、複数のモジュールをスレッド化し、マルチスレッドプログラミングを行う。パラメータ推定モジュールをトラッキングスレッドとマッピングスレッドの 2 つのスレッド、ユーザインタラクションモジュールを 1 つのスレッド、形状復元モジュールを前景抽出スレッドとボリュームデータ生成スレッドの 2 つのスレッド、合計 5 つのスレッドで構成する。

#### 5.1 パラメータ推定モジュール

本研究では、自己位置姿勢推定を行うにあたり、PTAM を利用した。実装には、公開されているライブラリ<sup>8)</sup> を利用した。

#### 5.2 ユーザインタラクションモジュール

画面中のストロークを文字として扱い、文字認識を行うために、Tesseract-OCR<sup>9)</sup> を用いた。ストローク入力を解析し、Tesseract-OCR にデータを渡し、Tesseract-OCR から得られた結果をシステムに入力することで、Inclusion brush と Exclusion brush の切り替え、SCM のスタートなどを実現した。

#### 5.3 形状復元モジュール

##### 5.3.1 グラフカット

グラフカットを適用するためには、ユーザによる前景指定とグラフモデルが必要となる。前景指定のデータは、ユーザインタラクションモジュールから渡される。グラフモデルは、

画像の画素の輝度値と色モデルを用いて生成される。色モデルは複数考えることができるが、本研究では Gaussian Mixture Model(GMM)<sup>10)</sup> に当てはめる。GMM の推定には EM アルゴリズム<sup>11)</sup> を用いる。

- 色モデルに基づくエッジコスト

本研究では、表 1 に与えるエッジコストは GMM のパラメータを用いて、式 (9) – (10) と定義する。

$$\Pr(I_p|\cdot) = \sum_{i=1}^K \alpha_i p_i(I_p|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (9)$$

$$p(I_p|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}|^{1/2}} \cdot \exp\left(-\frac{1}{2}(I_p - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(I_p - \boldsymbol{\mu})\right) \quad (10)$$

t-link と n-link の間の重み係数  $\lambda=0.05$  とする。  $\lambda$  が大きすぎると t-link の影響が大きくなり、色情報の影響が大きすぎる。  $\lambda$  が小さすぎると n-link の影響が大きくなり、隣接画素の輝度値の影響が大きすぎる。それぞれの影響が出すぎないように、適切な値を試行錯誤して設定した。式 (9) における色モデルの重み  $\alpha_i$  は分類した色モデルに一律な重みを与える。

- Dynamic segmentation

Dynamic segmentation の処理内容では、画像全体へグラフカットを適用するのではなく、前景とその周辺のみをピクセルに対して行うことで高速化を図る。また式 (16) – (17) を用いることで、尤度に重みをつける (式 (11) – (14))。

$$R_f = -\ln \Pr(O|I_p) \quad (11)$$

$$R_b = -\ln \Pr(B|I_p) \quad (12)$$

$$\Pr(O|I_p) = \Pr(O) \Pr(I_p|O) \quad (13)$$

$$\Pr(B|I_p) = \Pr(B) \Pr(I_p|B) \quad (14)$$

- 距離変換

距離変換による確率分布生成では、境界からの距離  $d$  を 0.5 から 1 に正規化する。正規化を行う際に、正規分布における累積分布関数 (式 (15)) を距離変換画像に適用し、確率分布 (式 (16) – (17)) を形成する。

$$d = \frac{1}{2} \left( 1 + \operatorname{erf} \frac{x - \mu}{\sqrt{2\sigma^2}} \right) \quad (15)$$

$$\Pr(O) = \begin{cases} d_{fore} & \text{if } d_{fore} \geq d_{back} \\ 1 - d_{back} & \text{otherwise} \end{cases} \quad (16)$$

$$\Pr(B) = 1 - \Pr(O) \quad (17)$$

### 5.3.2 ボリュームデータ生成

SCM を実行するために、カービングを行う視点のカメラの自己位置姿勢と前景シルエットが必要になる。キーフレームは、複数回の Dynamic segmentation の実行が行われた後に取り出される。キーフレームのカメラの自己位置姿勢と前景シルエットを用いてカービングを行う。カービングには PTAM の推定誤差も含めた式 (8) を用いる。式 (8) のパラメータ  $n$  は、実際に動作実験を行った結果から  $n = 8$  に設定した。

### 5.3.3 形状復元データ生成

ボリュームデータをサーフェスレンダリングの 1 つである Marching Cubes 法 (MC 法) を用いて表面形状データにする。MC 法で生成したメッシュ群  $M$  に対して、テクスチャを貼り付けることで、写実性のある復元モデルを生成する。ここでメッシュ  $m (m \in M)$  に対して、どの視点から生成されたテクスチャ  $t (t \in T)$  を貼り付けるかが問題になる。メッシュ  $m$  に貼るテクスチャ  $t$  を式 (18) から決定する。  $\mathbf{n}_m$  をメッシュ  $m$  の法線ベクトル、  $\mathbf{v}_c$  をカメラの位置ベクトル、  $\mathbf{v}_m$  をメッシュ  $m$  の位置ベクトル、  $S(t, m)$  をメッシュ  $m$  をテクスチャ  $t$  上へ逆投影した面積とする。

$$m(t) = \begin{cases} \max S(t, m) & \text{if } \mathbf{n}_m \cdot (\mathbf{v}_c - \mathbf{v}_m) > 0 \\ \text{no texture} & \text{otherwise} \end{cases} \quad (18)$$

## 6. 動作実験

本節では、5 節で実装した物体復元モジュールの動作実験に関して述べる。動作実験に 3.40GHz Intel Core i7-2600CPU, nVidia GeForce GTX 460, 8GB メモリを使用した。画像キャプチャには PointGrey Flea3, 解像度 648x480 に画角 84.9x68.9° の 3.5mm レンズを装着した。

各処理における実行時間を表 2 に示す。三次元形状復元に必要な処理時間は約 0.8 秒である。ただし、5 節で説明したように、マルチスレッド処理を行うので、ユーザが感じる遅延

表 2 各処理の実行時間

Table 2 Timing results for each stage of our approach

処理内容	処理時間 [sec]
前景抽出の前処理	0.0124
前景の色モデル更新	0.0325
背景の色モデル更新	0.0083
グラフカット	0.0133
SCM	0.07405
MC 法	0.315

は少なく、前景抽出は毎秒 10 回程度で動作し、形状復元は毎秒 5 回程度で動作している。前処理のステージでは、前フレームの領域分割結果に対するラベル割り振り、距離変換などを行った。Dynamic segmentation の処理ではカメラ映像中では、前景の面積が背景の面積よりも大きい場合が多い。そのため、前景と背景の色モデルの更新において、前景の更新の方が処理に時間がかかっている。MC 法の処理は、同時に複数視点からのテクスチャマッピングも行うために、処理に時間がかかっている。

### 6.1 三次元形状復元

実環境の物体の形状を復元した過程を図 5 に示す。提案手法には復元困難な物体が存在する。これは、どの視点から見た場合でも、領域分割したシルエットにカービングされるべき部分が含まれていて、シルエット内部の領域はカービングされないような物体を復元する場合である。そのため、図 5 の下段の左図の例では宮殿の前方の不要な領域が残ってしまっている。視体積交差法の問題であるので、提案手法では凹物体の復元が困難である。この問題を解決するためには、photoconsistency (色の一貫性) やテクスチャマッチングなどを行い、シルエット情報だけでなくシルエット内部の色情報を用いる、特徴点ベースの手法と組み合わせて復元を行うなどの手法を今後検討する。

### 6.2 従来手法と提案手法の復元結果比較

最後に、従来の AR ジオラマで用いられた特徴点ベースの復元手法と、本研究の領域ベースの復元手法を比較する。比較物体はルービックキューブ、リンゴの 2 つの物体とする。ルービックキューブには、特徴点が多く存在する。そのため、図 6(b) のように、従来手法を用いた場合でもおおよそその形状を復元できる。しかし、ドロネー三角形分割の誤差から必要なメッシュも削除しているためモデルに欠損が生じている。提案手法を用いた場合、図 6(c) のようにメッシュに欠損を作ることなく物体形状が復元できていることが確認できる。しかし、モデルのエッジ部分が少し削り取られて完全なキューブの形状をなさない場合がある。

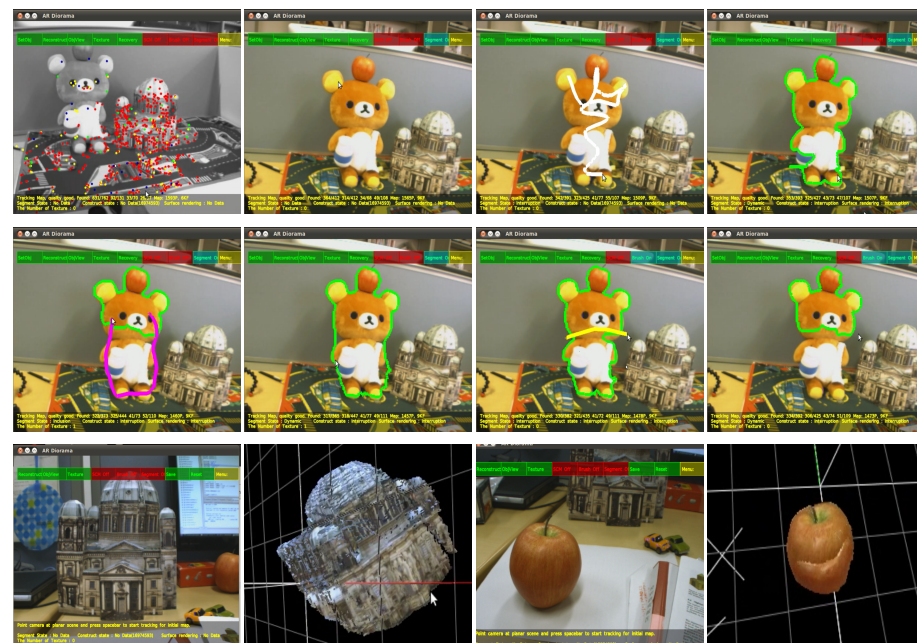


図 5 三次元物体形状復元 (上段) 左からトラッキング, カラー画像, 前景指定, 領域分割結果 (中段) 左側:Inclusion brush, 右側:Exclusion brush (下段) 左側:紙の宮殿, 右側:リンゴ

Fig. 5 3D reconstruction (top)From left, tracking, color image, foreground designation, segmentation result (middle)Left:Inclusion brush, Right:Exclusion brush (bottom)Left:paper craft palace, Right:apple

この問題は復元処理中のカメラをゆっくり移動させることで解決できる。

リンゴには特徴点が多く存在しなかった。そのため、従来手法では物体の復元不可能の結果に終わった。提案手法では、図 5 の下段の右図に示す結果が得られた。

従来手法と提案手法の比較を行った結果、従来手法では復元できない物体を提案手法で復元できることを確認した。また、特徴点が多い物体の場合についても、従来手法ではメッシュの欠損が発生する問題がある。これに対して、提案手法ではメッシュの欠損を発生させることなく形状を復元した。

以上の結果から、提案手法はテクスチャ依存性が低い利点を生かした復元が可能であることを確認した。

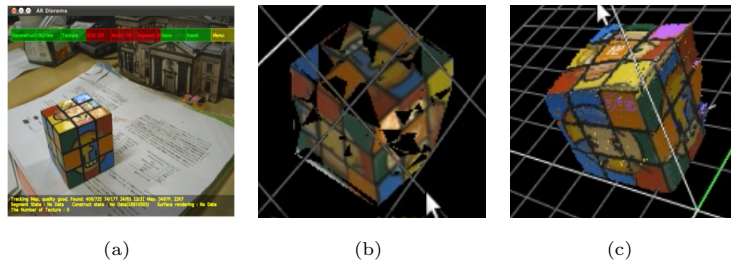


図6 ルービックキューブの復元結果 (a)入力画像 (b)従来手法 (c)提案手法  
Fig.6 The reconstruction result of a Rubic's cube (a)Input image (b)Original method (c)our approach

## 7. 結 論

本研究では、立石らのシステムをベースとして、Bastianらの形状復元手法を組み合わせることで、立石らのシステムにおける実物体の形状復元手法を向上させることを検討した。

本研究手法では、従来システムの特徴点を用いた復元手法ではなく、領域分割を用いた手法を用いた。色情報に基づく領域分割を行うことで、多視点からのシルエット画像を取得できる。そして、PTAMによるカメラ自己位置姿勢推定結果とシルエット画像から、視体積交差法によってボクセルデータとしての形状を復元する。次に、得られたボクセルデータに対してサーフェスレンダリングを行うことで、表面形状データを生成する。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、三次元形状モデルを生成した。

復元結果を見ると、領域分割の精度が高くない部分がある。また、処理速度の問題によりカメラをゆっくり動かさなければいけない。そして、領域分割の精度はカービング精度にも影響を与えるため、高精度なモデルを復元できない場合がある。しかし、ユーザがカメラをゆっくり動かし、領域分割がうまく動作するように、復元対象の物体の周囲をユーザが移動することで、高精度な、写実性の高い三次元形状モデルを生成できることを確認した。

今後の課題としては、領域分割の前処理の手法を再検討、処理フローの改善をすることで、処理速度の高速化と領域分割の精度向上を図る。現在はボクセルに対する処理をCPU側で行っている。GPU Shaderなどを用いて処理効率を上げることも検討する。ボリュームデータの精度向上後、リアルタイムかつ不整合を発生させないテクスチャマッピングを行う手法の調査、検討を進める。

本研究の最大の目的は、ARジオラマシステムの構築である。そのために、モデルを実環境に配置し、移動や回転などのインタラクション、光源の追加、ジェスチャ認識などの新しい機能を持たせたジオラマシステムを構築する。

## 参 考 文 献

- 1) H.Kato and M.Billinghurst: "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 AR Workspaces," Proc. of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.1-10, 2007.
- 3) 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.
- 4) 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.
- 5) 石川博: "グラフィカット (チュートリアル)", 情報処理学会研究報告. CVIM, pp.193-204, 2007.
- 6) Y.Boykov and V.Kolmogorov: "An Experimental Comparison of Min-cut/max-flow Algorithms for Energy Minimization in Vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.26(9), pp.1124-1137, 2004.
- 7) 永橋和行: "グラフィカットによる領域セグメンテーションに関する研究", 中部大学大学院工学研究科 修士学位論文, 2007.
- 8) Georg Klein Home Page: "Parallel Tracking and Mapping for Small AR Workspaces," <http://www.robots.ox.ac.uk/gk/PTAM/>. (最終閲覧日:2011年2月20日)
- 9) Google Project Hosting: "tesseract-ocr," <http://code.google.com/p/tesseract-ocr/>. (最終閲覧日:2011年4月13日)
- 10) A.P.Dempster, N.M.Laird, and D.B.Rubin: "Maximumlikelihood From Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society. Series B, Vol.39, No.1, pp.1-38, 1977.
- 11) C.Stauffer and W.E.L.Grimson: "Adaptive Background Mixture Models for Real-time Tracking," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp.246-252, 1999.