

# 深度カメラを用いた AR 化石発掘アプリケーション

## AR fossil digging application by using a depth camera

玉木 和鷹\*  
Kazutaka Tamaki

高井 昌彰†  
Yoshiaki Takai

飯田 勝吉‡  
Katsuyoshi Iida

### 1. はじめに

拡張現実 (AR) において仮想物体の実在感向上を考える際、4つの整合性、すなわち幾何学的整合性、光学的整合性、時間的整合性及び物理的整合性が重視される。一般のコンシューマー向け AR コンテンツでは対象物そのものが変形しないことを前提とすることが多いが、対象物の変形過程を含めて取り扱う場合でも、これら 4つの整合性は常に保たなければならない。

本研究では、AR コンテンツにおいて対象物の 3次元形状変化を実時間で追跡することに焦点を当てる。本研究では、近年急速に市場普及している安価な深度カメラを用い、対象物の局所的な変化を実時間で追跡する手法を構築し、その応用となる AR システムとして、化石発掘アプリケーションを開発する。このアプリケーションは、対象物として水羊羹やゼリーなどの柔らかい実物体を取り上げ、これをユーザがスプーンなどでえぐり取ることで、仮想の化石を掘り出すアプリケーションである。

### 2. 関連技術

#### 2.1 ボクセルデータ

ボクセルデータは立体形状のモデリング手法の一つであり、空間全体を立方体で格子状に分割し、それらの立方体ごとに密度等の属性を格納することで任意形状を表現する。ボクセルデータを構成する最小の立方体をボクセルと呼ぶ。

#### 2.2 マーチングキューブ法

マーチングキューブ法はボクセルデータをポリゴンデータに変換するアルゴリズムの一つであり、医療ボリュームデータセットからの表面再構成などに利用されている [1]。

マーチングキューブ法でポリゴンデータを生成する際、各ボクセルが 2 値の状態をもつとすれば、最隣接 8 ボクセルは 256 通りの状態を有する。このそれぞれの場合に対して、表面を近似するポリゴンパッチを形成することで、ボクセル空間全体にわたるポリゴンデータを高速に生成することができる。

### 3. システムの概要と設計

#### 3.1 基本的アプローチ

立体形状の実時間変化を追跡し、これを形状データとして保持するため、本システムでは深度カメラを使用する。深度カメラによって、AR 表示に必要な色情報と空間の奥行き情報を同時に容易に取得できる。立体形状のデータ構造にはボクセルデータを利用する。

形状変化の追跡のアプローチは以下の通りである。最初に対象物の変化前の初期立体形状データを取得し、その後、ユーザの対象物への操作で形状変化が進行していく過程において、立体形状データを繰り返し更新していく。そのためシステム構成を初期化部と主要部の 2 つに分け、初期化部で初期形状データを作成し、主要部でデータ更新を行う。

システムの動作イメージを図 1 に示す。まず初期化部では、ユーザは深度カメラを用いて対象物を周囲から撮影し、システムが対象物の初期形状データを作成する。主要部では、ユーザは対象物を適宜えぐり取って形状を変化させ、同時にこの様子を深度カメラで撮影する。システムは対象物の立体形状変化を追跡し、形状データを更新するとともに、変化した対象形状との整合性が保たれた AR 表示を行う。

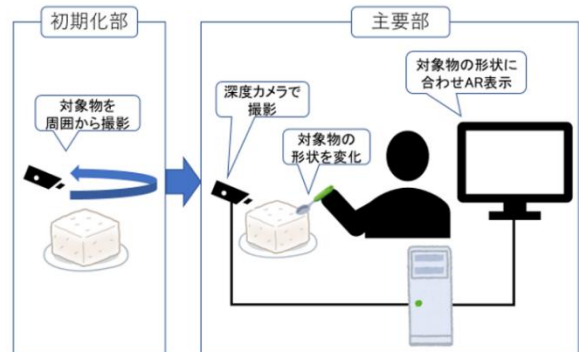


図 1. システムの動作イメージ

#### 3.2 システムの設計

システムの処理の流れを図 2 に示す。初期化部では対象物の形状データを保存するための空のボクセル空間を作成する。次に対象物を周囲から深度カメラで撮影し、それらの深さ情報をもとに対象物の表面をボクセル空間に表現する。最後に対象物の内部にあたるボクセルを充填する。

主要部では、深度カメラの撮影フレームごとに対象物の立体形状を更新し、その形状に合わせた AR 表示を行う。

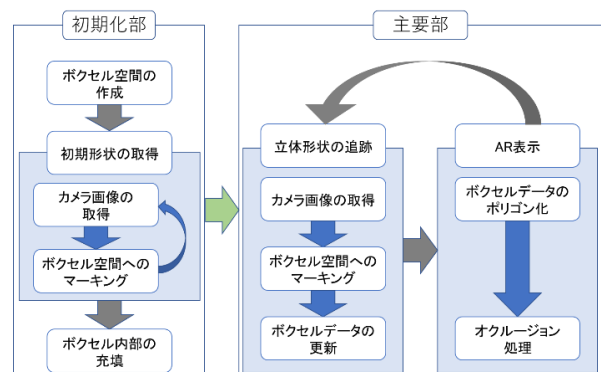


図 2. システムの処理の流れ

\*北海道大学大学院情報科学院 Graduate school of Info. Sci. Technology, Hokkaido University

†北海道大学情報基盤センター Information Initiative Center, Hokkaido University

‡北海道大学情報基盤センター Information Initiative Center, Hokkaido University

以上の基本設計のもとに、システムを実現するためには、次に示す 7 つの構成要素機能が必要になる。これらの各実装については次節で述べる。

- ・対象物の位置、姿勢の認識
- ・ボクセル空間の作成
- ・深度画像のボクセル空間への反映
- ・ボクセル空間の充填
- ・ボクセル空間の更新
- ・ボクセルデータのポリゴン化
- ・オクルージョン処理

## 4. システムの実装

### 4.1 対象物の位置、姿勢の認識

対象物の位置、姿勢の認識には単純な AR マーカを用いる。対象物の足下部分にマーカを配置し、これから対象物の基本的な位置、姿勢を定める。なお、このマーカ認識には OpenCV の contribution モジュール ArUco を使用する。

### 4.2 ボクセル空間の作成

形状データを表すボクセル空間は 2 値 3 次元配列で実装する。マーカ位置を原点とし、x, y, z 各方向に 30cm の空間を 1 辺 1cm のボクセルに切り分け、ボクセルと同数の要素をもつ配列を作成する。ボクセルに対象物が一定割合以上存在すればボクセル値 1、存在しなければ 0 である。

### 4.3 深度画像のボクセル空間への反映

深度カメラから取得した深度画像は、各画素に深度情報を格納した画像形式である。この情報をボクセル空間へ反映するには以下の 4 つのステップを踏む。

初めに、深度画像から 3 次元座標値を持つ点の集合（点群）に変換する。この時点では点群はカメラ位置を原点とした座標系で表現されている。次に点群の座標系を、マーカを原点とした座標系へ変換する。最後に、変換した点群とボクセル空間を照らし合わせ、各点が存在するボクセルにマーキングを行う。点群の処理には、Point Cloud Library[2]を使用する。

なお深度カメラとしては、Intel RealSense D435（解像度 640×480, 30fps）を用いた。

### 4.4 ボクセル空間の充填

初期化部において対象物の撮影終了後、ボクセル空間は対象物の表面が存在するボクセルにマーキングがなされた状態になっている。このボクセル空間のうち、表面より内側のボクセルを充填することで、対象物の立体形状を内部が密なボクセルデータとして表現する。充填は、マーキングされたボクセルのそれぞれについて、そのボクセルよりも下層にあり、かつマーキングされていない全てのボクセルの値を 1 に書き換えることで実現する。

### 4.5 ボクセル空間の更新

ボクセル空間の更新は各ボクセルの値を書き換えることで実現する。対象物の形状変化はえぐり取る操作に限定されていることから、深度画像の反映によってマーキングされたボクセルのボクセル値が 1 である場合、そのボクセル値を 0 に更新する。

### 4.6 ボクセルデータのポリゴン化

AR の描画処理は全て Unity で行う。しかし、Unity ではボクセルデータを直接扱うことができないため、ボクセルデータを Unity で扱えるポリゴンデータに変換する必要がある。この変換には前述のマーチングキューブ法を用いる。

## 4.7 オクルージョン処理

発掘される化石を AR 表示するためには、対象物の外部に露出している化石部分のみを描画する必要がある。そこで深度カメラのカラー画像をそのまま表示した後、ポリゴン化された対象物の形状データを描画する際、その画素値は使用せず、ポリゴンが描画されるべき画素位置だけを一時保存する。その後化石モデルを描画する際に、先に保存した画素位置よりも手前にくる化石モデルの画素のみを描画する。これにより化石モデルの対象物によるオクルージョンが実現される。

## 5. 動作結果

本システム（AR 化石発掘アプリ）を動作させた際の様子（対象物の形状変化前と変化後）を図 3 に示す。この実験では形状変化する対象物として水羊糞を用い、これをスプーンで少しずつえぐり取る操作を加えている。対象物を載せた皿の直径は 18cm である。

化石がまだ埋まっている部分は表示されず、対象物をえぐり取る操作の進行とともに、埋まっていた化石が徐々に露出してくることが確認できる。

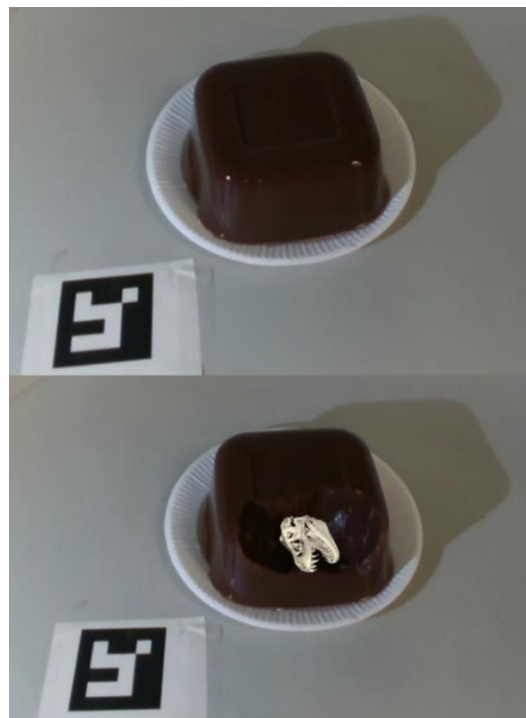


図 3. 立体形状の変化前（上）と変化後（下）

## 6. まとめ

深度カメラを用い、実物体から仮想の化石を発掘する AR アプリケーションを作成した。今後の課題としてはマーカを用いない位置・姿勢の推定、形状の膨張や付加など、えぐり取る以外の形状変化への対応、仮想の化石に触れた際の力覚へのフィードバックが挙げられる。

### 参考文献

- [1] Paul Bourke: “Polygonising a scalar field”  
<http://paulbourke.net/geometry/polygonise/>
- [2] “Point Cloud Library”  
<https://pointclouds.org/>