Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

深度カメラによる立体形状追跡を用いた AR 化石発掘アプリケーション

玉木和鷹 [1]　高井昌彰 [2]　飯田勝吉 [2]　高井那美 [3]

概要：拡張現実（AR）において形状変化する 3 次元物体を認識対象とする場合，対象物体の形状を取得し，その変化を実時間で追跡する必要がある．本研究では立体形状の変化に焦点を当て，深度カメラを用いて 3 次元物体の形状変化を追跡するシステムを実現する．また，その応用として AR 化石発掘アプリケーションを開発する．本システムでは形状変化を追跡する対象物の形状データを点群形式で保存し，形状データを，毎フレーム取得する情報をもとに繰り返し更新することで立体形状変化の追跡を実現する．立体形状の取得には現在普及の進む一般的な深度カメラを用い，深度画像を点群データに変換した後，立体形状データを再構築する．立体形状データの再構築にはマーチングキューブ法を用いる．

キーワード：拡張現実感，深度カメラ，立体形状構築，AR アプリケーション，化石発掘

AR fossil excavation application using 3D shape tracking with a depth camera

KAZUTAKA TAMAKI[†1]　YOSHIAKI TAKAI[†2]
KATSUYOSI IIDA[†2]　NAMI TAKAI[†3]

1. はじめに

拡張現実（AR）において仮想物体の実在感を考える際，幾何学的整合性，光学的整合性，時間的整合性，そして物理的整合性が重要視される．一般のコンシューマ向け AR アプリケーションでは実在する物体の形状が変化しないことを前提とすることが多いが，実在する物体の形状変化の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れなければならない．

対象物体の形状変化を実時間で取得する一つの方法として，深度カメラから得られる深度画像の利用がある．深度カメラを AR に応用した研究事例では，移動する物体にプロジェクションマッピングを行うもの[1]，車いすにおけるバリア検証を行うもの[2]，様々な姿勢での仮想の衣服の試着を実現するもの[3]などが挙げられる．深度カメラ機能を有するスマートフォンなど，安価な深度カメラが近年急速に市場普及している．

本研究では，実在する物体の 3 次元形状変化を実時間で追跡する AR アプリケーションに焦点を当て，深度カメラを用いて対象物の局所的な形状変化を実時間追跡する手法を構築する．また，その応用となる AR システムとして，Unity をプラットフォームとした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，形状変化を追跡する対象物として，屋内で紙粘土や丸めた紙等を用いて形作った簡単な堆積物の山を取り上げる．これを化石等の埋まっている地層に見立て，ユーザが実際に発掘操作を行って山の立体形状を局所的に変化させていくことで仮想の化石あるいは埋蔵物を掘り出すアプリケーションである．

1 北海道大学大学院情報科学院
Graduate School of Information Science and Technology, Hokkaido University
2 北海道大学情報基盤センター
Information Initiative Center, Hokkaido University
3 北海道情報大学経営情報学部
Hokkaido Information University

2. 立体形状モデルと可視化の基礎

2.1 点群

点群（ポイントクラウド）は対象物を 3 次元空間に分布する多数の点の集まりとして表現するモデリング手法である．各点は一般に 3 次元の直交座標と色情報の属性値を有する．点群データは 3D レーザースキャナや深度カメラ等から取得されるが，複雑な形状の表現には点群の分布密度を適切に保つ必要がある．

2.2 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る．空間を立方体で格子状に分割し，それらの立方体ごと，あるいは立方体の頂点ごとに密度など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．また，ボクセルやその頂点に格納するデータをボクセル値と呼ぶ．

2.3 マーチングキューブ法

3 次元物体の形状を表現する場合，その表面となる三角形の集合であるポリゴンデータを用いることが一般的であるが，ボクセルデータはその性質上，そのままではポリゴ

Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

ンデータで表現することができない．ボクセルデータをポリゴンデータに変換する際に使用されるアルゴリズムの一つがマーチングキューブ法である[4]．マーチングキューブ法は 3 次元ボクセル空間内の等値面をポリゴンパッチに変換するアルゴリズムであり，医療ボリュームデータセットからの表面再構成などに広く利用されている[5]．

マーチングキューブ法でポリゴンデータを生成する際，各ボクセルのボクセル値が設定した閾値を超えるかどうかで各ボクセルの状態，すなわち対象物の内部か外部かを判断する．この場合，各ボクセルは 2 通りの状態をもつため，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，表面を近似するポリゴンパッチを形成することで，ボクセル空間全体のポリゴンデータを高速に生成できる．また，ポリゴンパッチ生成の際，ボクセル値に応じてポリゴンの頂点座標値を変化させることで，より実物の形状に忠実なポリゴンパッチの生成が可能となる．

マーチングキューブ法の応用として，事前にボリュームデータから特異点グラフを作成することでポリゴン生成を高速化する手法[6]や，サンプリング間隔内部のボクセル情報を考慮することで分解能を向上させる手法[7]がある．

3. システムの概要と設計

3.1 基本的アプローチ

対象物の立体形状の実時間変化を追跡するため，本システムでは深度カメラを用いる．一般的な RGB カメラと深度カメラを併用することにより，AR 表示に必要な色情報と空間の奥行き情報を同時に取得できる．

立体形状の変化を追跡する対象物の形状データはボクセル形式と点群を組み合わせて保存する．その際，各ボクセルにはそのボクセルに属する各点の座標値を保存する．

形状変化追跡のアプローチは以下の通りである．最初にデータを保存するための空のボクセル空間を作成する．その後，ユーザが対象物の立体形状を変化させる様子を深度カメラで撮影し，システム内で深度カメラからの深度画像を点群に変換する．これをもとに，形状データ保存用の点群データに対して点の追加や座標値の更新を行い，ボクセルデータを更新する．この操作を繰り返し行うことで形状変化の追跡を実現する[8]．

本システムの AR 化石発掘アプリケーションとしての動作イメージを図 1 に示す．ユーザが操作する実際の対象物としては，屋内において紙粘土や丸めた紙等を用いて形作った簡単な堆積物の山を想定し，その内部に仮想の化石あるいは埋蔵物が予め埋められているものとする．ユーザが山を徐々に掘り進んでいく過程を深度カメラで撮影することで，対象物である山の 3 次元形状変化を追跡すると同時に，発掘による形状変化と整合性のとれた仮想の化石・埋蔵物の AR 重畳表示を行う．

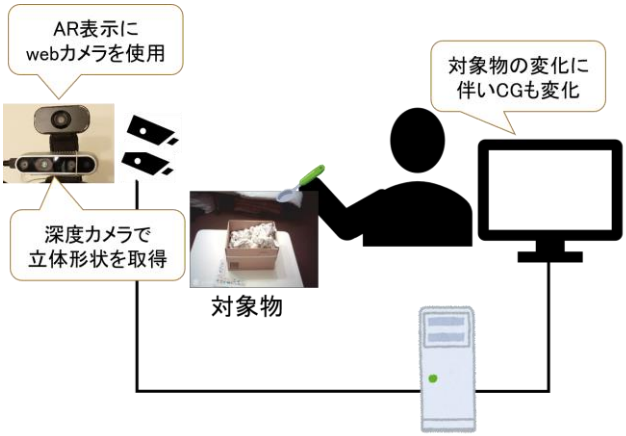

図 1　システムの利用イメージ

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．最初に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し，その後深度カメラの撮影フレームごとに次の操作を繰り返す．

まず深度カメラから深度画像を取得し，そのデータを点群データに変換する．次にその点群データをもとに形状データを更新する．その後，形状データをマーチングキューブ法でポリゴン化し，それに合わせた AR 表示を行う[9]．

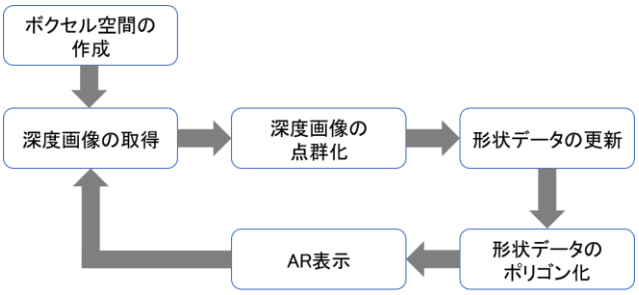

図 2　システムの処理の流れ

以上の基本設計のもと，システムの実現には，次の 6 つの構成要素機能が必要になる．すなわち，対象物の位置と姿勢の認識，深度画像の点群化，形状データの保存，形状データの更新，形状データのポリゴン化，そしてオクルージョン処理（隠面消去）である．これら各要素の実装については次章で述べる．

4. システムの実装

4.1 対象物の位置・姿勢の認識

本システムでは対象物の形状が変化することを想定するため，対象物の位置や姿勢を推定する際にその基準となるものが必要となる．本システムでは図 3 に示す L 字型の画像マーカを使用する．このマーカを対象物の台座付近に設置し，これから得られた位置・姿勢の情報を対象物の基準となる位置・姿勢として扱う．このマーカの認識には Vuforia を使用する．また，使用したライブラリの仕様上，

Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

RGB-D 画像が出力される深度カメラ 1 台のみでマーカ認識と深度画像取得を同時に行うことが不可能であったため，一般的な RGB カメラ（小型 Web カメラ）と深度カメラをそれぞれ 1 台ずつ使用する．マーカ認識には Web カメラを使用し，立体形状の取得には深度カメラを使用する．

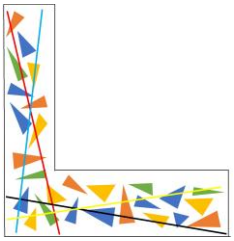

図 3　位置・姿勢の認識に使用するマーカ

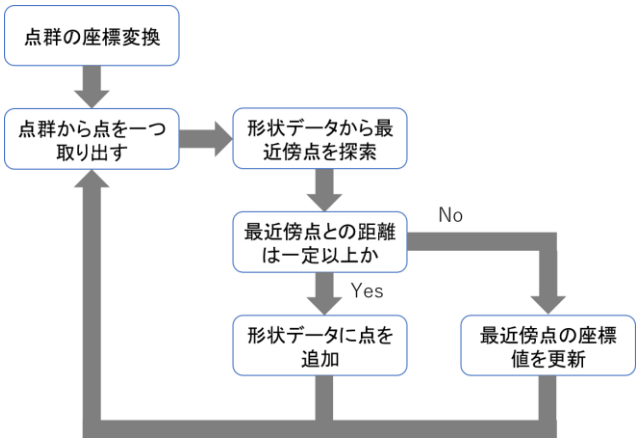

図 4　形状データ更新の流れ

4.2 深度画像の点群化

本システムでは，対象物の奥行情報を取得する深度カメラとして Intel RealSense D415 を使用する．奥行情報は深度画像（各画素に奥行方向の距離が格納された画像）で与えられる．本システムでは奥行方向の情報を点群の形式で扱うため，深度画像を点群に変換する必要がある．また，Intel RealSense D415 で取得する深度情報には，1cm 程の誤差が含まれるため，これを低減する前処理が必要である．

以上を踏まえ，Intel RealSense D415 からの深度画像にフィルタリング処理を行い，その後深度画像を点群に変換する．この前処理は RealSense の SDK で提供されているものを使用する．具体的には，誤差を修正するためのフィルタリング処理として，エッジを保ちつつ平滑化を行う Rs Spatial Filter を適用し，その後，前フレームのデータをもとに補正をかける Rs Temporal Filter を適用する．

4.3 形状データの保存

形状変化を追跡する対象物の形状データをボクセル形式で保存するため，マーカ周辺の 3 次元空間を，マーカ中心を原点とするボクセル空間と見なし，各ボクセルにそのボクセル内に存在する点群の点の座標を全て保存する．

また，ボクセル空間をマーチングキューブ法でポリゴン化するため，各ボクセルの頂点位置のボクセル値も保存する必要がある．この際ボクセル値は 0 から 1 の小数値で保存する．ボクセル値の計算については 4.5 節で述べる．

以上の実現のため，形状データ保存用のボクセル空間はリスト構造の 3 次元配列で実装し，各ボクセルのボクセル値の保存には浮動小数点型の 1 次元配列を用いる．

4.4 形状データの更新

形状データの更新は，深度画像から変換した点群データをもとにして行うことで実現する．形状データの更新の流れを図 4 に示す．

深度画像から得られた点群は深度カメラ位置を原点とした座標系で表現されるため，座標変換を行った後，マーカ中心を原点とした座標系で表現する．その後，取得した点群の各点について，形状データ内の最近傍点を探索する．

最近傍点の探索処理の際，形状データを単純に全探索すると計算量が膨大となり，実時間処理が困難になるため，探索範囲を限定する必要がある．そこで，最初に探索を行う点がボクセル空間のどのボクセルに属するかを求め，求めたボクセルと，それに隣接するボクセル内の点群から最近傍点を求める．

以上のようにして得られた最近傍点との距離が一定以内であれば点が移動したものと判断し，最近傍点の座標値を更新する．そうでなければ，新たな点が増えたものと判断し，形状データにその点を追加する．

4.5 形状データのポリゴン化

AR の描画処理は全てゲームエンジン Unity で行うが，Unity では点群データを直接扱うことができない．そのため，点群データからボクセルデータを作成し，そのボクセルデータを Unity で扱えるポリゴンデータに変換する．この変換にはマーチングキューブ法を用いる．

ボクセルデータにおいては，各ボクセルの頂点位置にボクセル値が設定されている．このボクセル値はボクセル内の点群の分布を考慮して次のようにして求める．

はじめに点群の各点に対して，頂点位置からのユークリッド距離に応じて等方的に減衰する関数（寄与値関数）を定義する．この関数に従い，各頂点に点群の各点からの寄与値を算出する．こうして得た寄与値の総和を各頂点のボクセル値とする．なお，このボクセル値の計算は GPU を使用して並列化し，それぞれのボクセル値を同時に求める．

点群の寄与値を求める寄与値関数については，GPU 処理の容易さの観点も踏まえて，ステップ関数，一次関数，反比例関数の 3 通りを実装し，マーチングキューブ法による可視化の結果を比較した．これらの寄与値関数を図 5 に示す．ここで，一次関数および反比例関数では距離 0cm の時に寄与値 0.5 をとるように設定し，ステップ関数では距離 0cm の時に寄与値 0.5 もしくは 0.25 をとるように設定している．また，どの関数も距離がある閾値 d（有効半径）以上に大きくなると寄与値が 0 となるよう設定している．この d 値の決定については次章で述べる．

Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

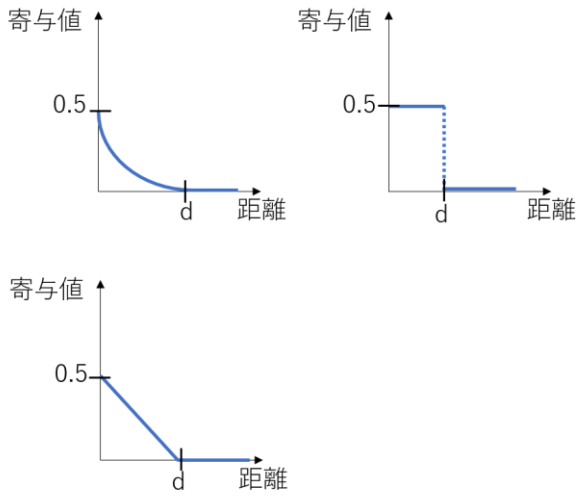

図 5　寄与値関数（左上：反比例関数，右上：ステップ関数，左下：一次関数）

4.6 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，仮想化石のうち，対象物の外部に露出している部分のみを描画する必要がある．このオクルージョン処理（隠面消去）にはステンシルバッファを使用する．AR 表示を行う際，まずは Web カメラからの RGB 画像をそのまま描画する．その後，ポリゴン化された対象物の形状データを描画する．この際，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をステンシルバッファに保存する．その後，化石モデルを描画する際にステンシルバッファを参照し，描画する画素位置が先に保存した画素位置に一致する際には，化石モデルがポリゴン化された対象物よりも手前にくる場合のみ描画を実行する．

5. 動作結果

5.1 動作環境

本システムの動作環境を表 1 に示す．また，本システムを室内で動作させた際のカメラと対象物の位置関係を図 6 に示す．対象物の立体形状変化を追跡する基本動作を確認するため，固定のカメラから 1m 離れた位置に台座を置き，その台座上にマーカと対象物を設置した．AR 化石発掘アプリケーションの最終形としては，深度カメラ機能を搭載した HMD や AR メガネ等の利用を想定する．

5.2 寄与値関数の効果

先に述べた各寄与値関数を用いてボクセル値を計算し，マーチングキューブ法を適用して可視化した結果について述べる．なお，対象物の元形状と構築されたポリゴンパッチ形状を比較しやすくするため，対象物として図 7 に示す直方体の段ボール箱を用いた．また，寄与値が 0 となる閾値 d は 1cm，3cm，5cm の 3 通りとした．

表 1　システムの動作環境

OS	Windows10 64bit
CPU	Intel Corei7-8086K　(memory 16GB)
GPU	NVIDIA GeForce GTX 1060
開発言語	C#
ゲームエンジン	Unity 2019.2.11f1
AR ライブラリ	Vuforia 8.5.9
Web カメラ	CMS-V41BK　(30FPS,640×480)
深度カメラ	Intel Realsense D415　(30FPS,640×480)

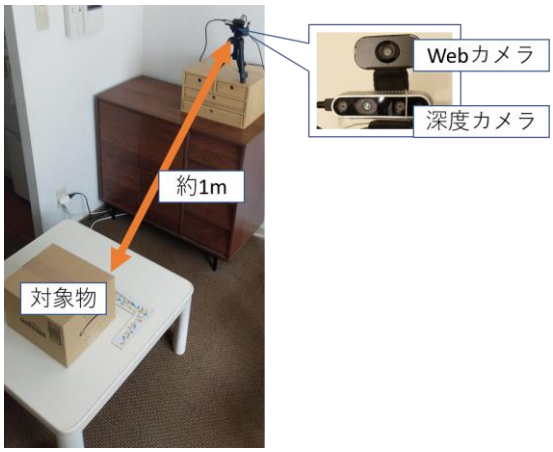

図 6　対象物とカメラの位置関係

ボクセル空間の大きさはマーカを中心に各軸 60cm（x：-20cm～+40cm　y：-15cm～+45cm　z：-20cm～+40cm）に設定し，分割数は 45×45×45（約 9.1 万ボクセル）である．

寄与値関数としてステップ関数を用いた結果を図 8 に示す．なお，ステップ関数での寄与値は 0.5 の場合と 0.25 の場合とで 2 通り実行した．一次関数を用いた結果を図 9 に，反比例関数を用いた結果を図 10 にそれぞれ示す．

有効半径 d が深度カメラの誤差と同程度である d=1cm の場合では形成される表面に粗さが目立つ．一方，d=5cm の場合にはポリゴン表面が不必要に膨張し，実際の対象物との隔たりが大きくなる．この実験では，有効半径として 3cm 程度が適当である．また，寄与値関数としては，反比例関数を用いることで，対象物の元形状に最もフィットした形状を得ている．

5.3 処理時間

システムの処理においては，3.2 節で示したように，深度カメラから得られる距離画像 1 フレームごとに，点群化，ポリゴン化及び AR 描画の処理を繰り返している．フレームあたりの描画処理に要する時間は，ボクセル値の計算に 7～9ms（寄与値関数に依存せず），マーチングキューブ法でのポリゴン生成に 3～5ms を要しており，処理全体で 45～55ms，18～22fps のフレームレートを実現している．

Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

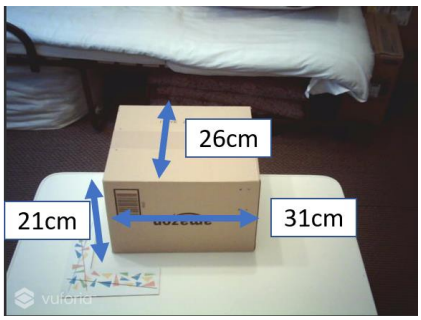

図 7　対象物として使用した段ボール箱

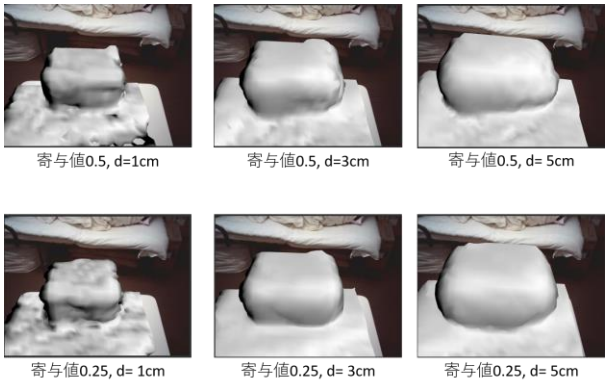

図 8　ステップ関数を用いた結果

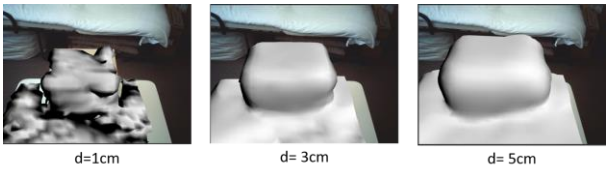

図 9　一次関数を用いた結果

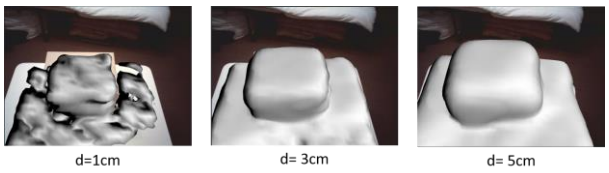

図 10　反比例関数を用いた結果

図 11　対象物（左）と仮想化石モデル（右）

図 12　可視化した対象物とオクルージョン処理の結果

5.4 オクルージョン処理の結果

仮想の化石発掘を実現するオクルージョン処理の効果を見るため，図 11 に示すように段ボール箱の中に丸めた紙を多数詰め込んだ仮想の山を対象物として用意した．AR 表示する化石としては図 11 右の 3D モデルを使用する．ポリゴン化の寄与値関数には反比例関数を用いた．対象物の可視化結果を図 12 上段に，最終的に仮想化石だけを描画した結果を同図下段に示す．紙の山の立体形状に概ねフィットした仮想化石の描画ができていることがわかる．また，丸めた紙を箱から徐々に取り除き化石発掘を進めた様子を図 13 に示す．同図上段は発掘開始前の様子で，中段から下段に向け発掘が進行している．

仮想化石と紙の山との隔たりをより少なくするためには，深度カメラの精度に見合った対象物のサイズ設定，ボクセル空間の解像度及び寄与値関数のパラメータチューニングが必要である．

6. まとめと今後の課題

Unity/Vuforia を開発プラットフォームに，深度カメラを用いて現実の物体の局所的な形状変化を追跡する手法を構築した．また，その手法を応用し，現実の物体から仮想の化石を発掘するアプリケーションを実現した．形状変化を追跡する対象物の形状データは点群形式で保存し，深度

Vol.2021-CG-184 No.6
Vol.2021-DCC-29 No.6
Vol.2021-CVIM-227 No.6
2021/11/5

カメラを通して取得された点群を，形状データに逐次反映することで立体形状の追跡を実現する．また，対象物の位置や姿勢は，対象物付近に設置した画像マーカによって認識する．点群で表現した形状データはマーチングキューブ法によってポリゴン化し,オクルージョン処理に利用する．

AR 化石発掘アプリケーションとしての完成度を高めるには，深度カメラの精度に見合った対象物サイズの設定に加え，点群が配置されるボクセル空間及び寄与値関数のさらなる精緻化が必要であり，これらを処理の実時間性の制約のもとに実装していくことは今後の課題である．

図 13　AR 化石発掘を進めた様子

参考文献

[1] 庭田直也，橋本直己：“高速な特徴点検出を用いた動的な空間型 AR の実現”，映像情報メディア学会技術報告，vol.40，no.5，pp.93-96（2016）
[2] 高橋里緒，檀 寛成，安室喜弘：“車椅子におけるバリア検証のためのデプスカメラによる AR 表示”，情報処理学会第 81 回全国大会，2ZE-03，vol.4，pp.439-440（2019）
[3] 川口侑希子，橋本直己：“バーチャル試着を手軽に実現する Dress Capture”，映像情報メディア学会技術報告，vol.37，no.7，pp.47-50（2013）
[4] William E. Lorensen and Harvey E. Cline：“Marching Cubes: A High Resolution 3D Surface Construction Algorithm”，*Computer Graphics*，Vol.21，No.4，pp.163-169（1987）
[5] Paul Bourke：“Polygonising a scalar field”，http://paulbourke.net/geometry/polygonise/”
[6] 竹島由里子，松本伸子，鈴木喜雄，中島憲宏，小山田耕二：“異点グラフを用いた等値面抽出法の評価” 可視化情報，A210，vol.25，no.1，pp.143-146（2005）
[7] 高波健太郎，藤野 勝，長坂 学，菊川孝明，緒方正人：“内部ボクセル情報を用いたマーチング・キューブ法における等値面分解能の向上”，情報処理学会研究報告，2009-CG-134，pp.7-12（2009）
[8] 玉木和鷹，高井昌彰，飯田勝吉：“深度カメラを用いた AR 化石発掘アプリケーション”，第 19 回情報科学技術フォーラム FIT2020 論文集，vol.3，pp.227-228（2020）
[9] 玉木和鷹，高井昌彰，飯田勝吉，高井那美：“深度カメラによる立体形状変化の追跡とその AR 応用”，情報処理学会第 83 回全国大会論文集，vol.4，pp.27-28（2021）