

運動する人体の全周に対する リアルタイムプロジェクションマッピング

森 正樹^{1,a)} 床井 浩平^{2,b)}

概要: プロジェクションマッピングでは一般に映像の投影を行う対象の形状が既知である必要がある。したがって、不規則な運動をする人体等にプロジェクションマッピングを行うためには、何らかの方法で対象の形状を動的に取得する必要がある。本研究では複数の距離画像センサで人体の全周形状を取得し、それに対してテクスチャをマッピングした映像を投影することによって、不規則な運動をする人体に動的なプロジェクションマッピングを行う手法を開発している。本稿ではセンサ1台、プロジェクタ1台による人体へのプロジェクションマッピングを行った結果について報告する。

Real-time Projection Mapping on the Entire Moving Human Body

MORI MASAKI^{1,a)} TOKOI KOHE^{2,b)}

Abstract: In general, the shape of the projection target should be known when performing the projection mapping. Therefore, in order to perform the projection mapping on the irregularly moving human body, needs to get the shape of the target dynamically in some way. In this research, we are developing the dynamic projection mapping technique to the irregularly moving human body by projecting the texture mapped image of the entire shape of the human body that is obtained from the multiple range image sensor. In this paper, we report the results of the projection mapping on the human body using one projector and one sensor.

1. 研究動機

現在、世界各地のテーマパーク、商業施設、有名建築物、アート作品など、至る所でプロジェクションマッピングが実演されている。しかし、そのほとんどは建物などの静止した物体の、ある一面に対してのみ映像を投影しているものである。

Rabarama360°[1] は一面だけでなく、物体の全周に対してプロジェクションマッピングを行っている数少ない例である。こちらは静止した人物モデルの全周に対して、繋ぎ

目を感じさせることなく映像を投影することができている。

近年では、運動する物体に映像を投影している例も見られるようになった。Box[2] では運動するロボットアームに取り付けたパネルに対して、動的にプロジェクションマッピングを行っている。このように運動する物体に対して、遅延を生じさせることなく映像を投影するためには、一般的にロボットアームや回転台の動きに応じて投影像を変形させるという手法が取られる。

これに対して人体など、不規則な運動をする物体に対してプロジェクションマッピングを行おうとする場合、投影像の遅延は回避しがたい問題であり、これを実現している例は数少ない。OMOTE[3] はマーカーを用いてフェイストラッキングをすることで、運動する人物の顔面に対する動的なプロジェクションマッピングを実現している。

本研究ではこのような既存のプロジェクションマッピン

¹ 和歌山大学大学院システム工学研究科
Graduate School of Systems Engineering, Wakayama University

² 和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University

a) s145047@sys.wakayama-u.ac.jp

b) tokoi@sys.wakayama-u.ac.jp

グを拡張し、身体に接触するマーカー類を使用せずに、運動する人体の全身、全周に対して動的にプロジェクションマッピングを行うことを目指す。

人体の運動に対して柔軟に変形する投影像を生成するためには、人体の形状計測を行うと同時に、その形状を可視化することが必要であると考えられる。そこで本研究では、距離画像センサを用いて、運動する人体の3次元形状を動的に復元することを考えた。

2. 3次元形状復元の関連研究

Kanadeら[4]、富山ら[5]は視体積交差法を用いて、多視点画像から動物体の3次元モデルを生成することに成功している。しかし、視体積交差法では複数のカメラから得られる物体のシルエットから形状を求めるため、多数のカメラが必要であり、モデルの生成をリアルタイムに行うことが容易ではない。

Furukawaら[6]は近赤外プロジェクタとカメラを円周上に配置し、動物体の全周形状を復元している。こちらはプロジェクタによって、パターン光を物体に投影して形状をアクティブに計測する手法である。このような手法では、プロジェクタはカメラの数と同じだけ必要となり、こちらの研究では6セット用いられている。

また、Chenら[7]はMicrosoft社が提供している3DスキャンツールであるKinect Fusionを拡張し、大規模なシーンの復元を行っている。しかし、こちらは1台のKinectを動かすことで3次元形状を復元するため、運動する物体の動的な復元は不可能である。

3. 提案手法

物体の全周形状を復元するために、多くの関連研究では多数のカメラやプロジェクタが使われているが、本研究では安価な距離画像センサ3台のみを用いて、それらから得られるデプスマップを元に、できる限り低コストな方法で3次元形状の復元を行う。デプスマップとはそれぞれの画素にカメラからの距離（深度）の値を格納したデータである。これを用いれば物体の形状計測が容易にできるため、3次元形状復元の研究で盛んに利用されている。池畑ら[8]、Liら[9]はカラー画像からデプスマップを推定し、融合を行っている。しかし、近年では安価な距離画像センサが実用化されているので、本研究ではそれを利用してデプスマップを得るため、デプスマップの推定の必要はない。撮影は図1のように物体の周囲にセンサを配置して行い、それぞれの方向からデプスマップを取得し、それらを融合することによって動物体の全周形状を同時に復元する。形状処理はGPUを用いることで高速化を実現し、動的な形状復元を可能にしている。

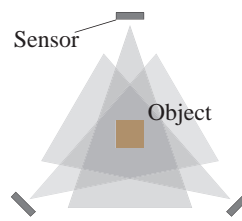


図1 3台の距離画像センサを用いた撮影方法
Fig. 1 3D scanning method using 3 sensors

4. デプスマップを用いた形状復元

本研究で用いている距離画像センサはSoftKinetic社のDepthSense311である。画像の取得はDepthSenseSDKを用いて行う。形状処理と可視化にはOpenGLを用い、DepthSense311から得られるデプスマップを元に形状を復元し、同時にカラー画像をマッピングして描画する。また、画像取得以外の形状処理全般を、GPU上のシェーダで行うことによって高速な処理を実現している。

4.1 画像の取得

DepthSenseSDKによってDepthSense311から取得しているデータは解像度QQVGA(160×120)のデプスマップ(値はfloat型)、VGA(640×480)のカラーマップである。

尚、DepthSenseからは、実座標であるVertices Floating Pointを取得することも可能であるが、提案手法では後述するデプスマップ融合の実装上、デプス値のみを用いている。

4.2 実座標の計算

各センサから得られるデプスマップはデプス値を格納した2次元のデータであり、その x, y 座標は透視投影変換が施された後の位置である。よって、透視投影変換の逆変換を行うことにより、現実空間における座標を求める必要がある。

透視投影変換されたデプスマップの画像面と、現実空間に存在する物体との垂直方向における位置関係は図2のように表すことができ、実座標 y' は次式で求めることができる。また、水平方向についても同様である。

$$y' = depth \cdot y \cdot \tan\left(\frac{fovy}{2}\right) \quad (1)$$

以上の実座標変換は、GPUで実行されるバーテックスシェーダにおいて頂点単位で行っている。画角についてはDepthSense311の公表値である水平方向57.3°、垂直方向42.0°を用いて(1)の計算を行うと、概ね正確な形状が求められる。

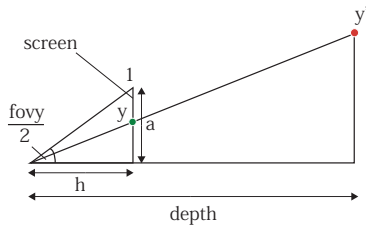


図 2 実座標への変換 (垂直方向)

Fig. 2 Transformation to real point (Vertical direction)

4.3 形状の位置合わせ

3台のセンサで復元した形状の位置合わせをするためには、各センサの位置姿勢を求め、それぞれの形状を OpenGL のワールド座標系に正しく配置する必要がある。

そこで、本研究では OpenCV のカメラキャリブレーション関数を用いて、センサの位置姿勢を取得し、位置合わせを行った。OpenCV の solvePnP 関数によって、チェスボードの最初のコーナーを原点としたときの、カメラの回転ベクトルと並進ベクトルを求めることができる。しかし、OpenCV のカメラ座標系と DepthSense311 のカメラ座標系は y 軸の向きが異なっているため、それぞれの y 成分の符号を反転する。その後、回転ベクトルを回転行列に変換し並進ベクトルと組み合わせることで、下記の 4×4 の位置姿勢変換行列 M ができる。ここでさらに、solvePnP 関数で取得する位置姿勢の基準である、チェスボードの座標系と OpenGL のワールド座標系が図 3 のように異なっているため、変換行列を変形して M' とする。

$$M = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} M' = \begin{pmatrix} -r_1 & -r_2 & -r_3 & -t_x \\ r_7 & r_8 & r_9 & t_z \\ -r_4 & -r_5 & -r_6 & -t_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

この変換行列 M' を先述の処理でデプス値から求めた実座標に乗じることによって、それぞれの形状が同一の座標系に配置され、これによって正確な位置合わせができています。

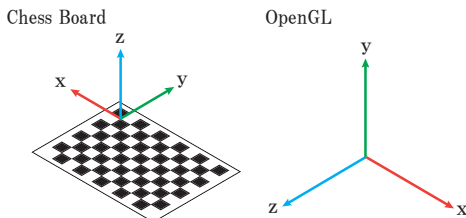


図 3 ワールド座標系の違い

Fig. 3 Difference of world coordinate system

4.4 デプスマップ融合

前節の処理によって復元された形状の位置合わせができ

たが、そのまま描画すると物体の影にあたる部分に不必要なポリゴンが生成されたり、形状同士の重なりが発生する。よって描画の際にフラグメントシェーダにおいて3つのデプスマップ全てを参照、比較し、描画する点の可視判定を行った。この処理をデプスマップ融合と呼ぶ。

図 4 はデプスマップ融合処理の手順を y 軸上方から見た形で示したものである。シェーダには OpenCV の関数によって得た、各センサの位置姿勢変換行列を転送しておく。この変換行列をセンサごとに M_a , M_b , M_c とする。

デプスマップ融合には、シャドウマッピングの手法を応用する。画面表示は対象形状のチェスボードを原点とする座標値を格子点に設定したメッシュをセンサごとに作成し、それらをレンダリングすることにより行う。その際、同じ座標値を格子点のテクスチャ座標にも設定しておく。

これにより、フラグメントシェーダにおいて処理対象のフラグメントのチェスボードを原点とする座標系における座標値が得られる。これに先程の変換行列の逆行列 M_a^{-1} , M_b^{-1} , M_c^{-1} を乗じて、そのフラグメントのそれぞれのセンサを原点とする座標系における座標値を求める。この座標値の xy 成分を用いて対応するセンサのデプスマップをサンプリングし、その値が z 成分より大きければ、そのフラグメントを破棄する。以上の処理により、すべてのセンサのデプスマップの交差部分にあるフラグメントが表示される。

図 5 は以上の処理を行い、復元した形状を描画したものである。

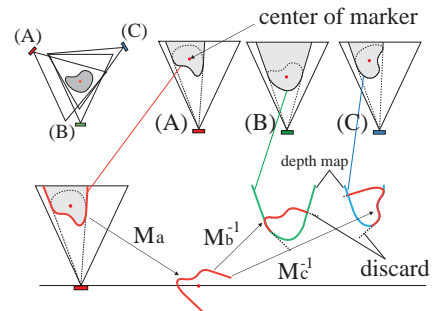


図 4 デプスマップの融合

Fig. 4 Depth map fusion



図 5 復元した形状の描画。(イスに乗せた箱)

Fig. 5 Drawing of the reconstructed model (box on the chair)

5. ノイズの除去

描画した形状を見ると、depth 方向にノイズが発生しており、形状を精密に復元できているとは言えない。この原因として、まずセンサ固有の計測誤差が考えられる。図 6 は、静止物体のある点のデプス値を、DepthSense311 の遠距離モード (1.50m -4.5m) で 1000 フレーム分記録したものである。このデータより、センサ単体で標準偏差約 0.0043m の計測誤差が発生していることがわかった。また、近距離モード (0.15m -1.0m) ではほとんど起こらないが、遠距離モードの場合センサ同士の干渉が起きて、ノイズが発生する。図 7 は、遠距離モードで計測した静止物体のある点のデプス値を、同じ方向を向くセンサの有無によって比較したものである。センサが複数ある場合、標準偏差が約 0.062m に増大した。

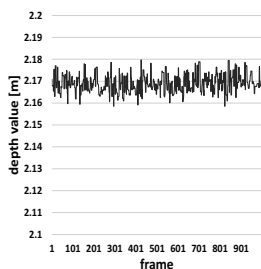


図 6 計測誤差

Fig. 6 Measurement error

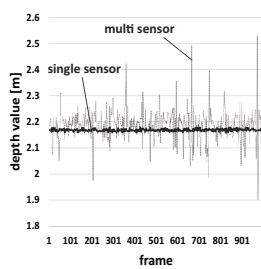


図 7 センサ同士の干渉

Fig. 7 Interference

5.1 特異値を除いた平滑化

ノイズへの対処として、 3×3 の Mean filter を用いた空間方向の平滑化を行った。シェーダ上に送られてくるデプスマップを参照し、処理点の近傍の値の平均値を取ったが、微小な値は含めないようにした。また、平均値が 0 になった点については、大きな値を与えてポリゴンを無限遠に伸ばし、デプスマップ融合で破棄することで対処した。図 8 は Mean filter の適用前後を比較したものである。



図 8 平滑化フィルター適用結果 (人物)

Fig. 8 Application of the smoothing filter (human)

6. 運動する人体に対するプロジェクションマッピング

プロジェクションマッピングの投影像は、以上の処理によって復元した 3 次元形状に対して動的にエフェクトを加えることで作成する。すなわち、DepthSense から取得したカラー画像を形状にマッピングする代わりに、その他のテクスチャをマッピングすることとなる。投影像はデプスマップ融合した形状をプロジェクタの数だけ用意し、これをそれぞれのプロジェクタから物体の形状に合わせて投影することで、全周に対するプロジェクションマッピングを行う。

6.1 テクスチャマッピング手法

復元した 3 次元形状へのテクスチャマッピングの手法として、現在環境マッピングの手法を用いている。環境マッピングとは、描画する形状の各頂点で映り込みを計算し、用意した周囲環境のテクスチャ (環境マップ) をサンプリングする手法である。これによって復元した 3 次元形状にくまなくテクスチャマッピングすることが可能となる。

環境マッピングの具体的な手法としては、以下の 2 つを考えている。2 つの手法はどちらも視点の位置を任意に設定できるという特徴を持っているため、プロジェクションマッピングの観測者の位置に応じてテクスチャのサンプリングが行えるという利点がある。

6.1.1 キューブマッピング

キューブマッピングとは、六面体に貼り付けたテクスチャを使って環境マッピングを行う手法である。図 9 のように描画する形状が六面体のキューブマップの内部にあるとして、形状への映り込みを表現する。

処理の流れは、まず視線ベクトル E と各頂点の法線ベクトル N を計算し、下記の式で反射ベクトル R を算出する。

$$R = -E + 2(N \cdot E)N \quad (3)$$

そして、求めた反射ベクトル R を用いて、フラグメントシェーダでキューブマップをサンプリングすることで、反射方向のピクセルをマッピングすることができる。

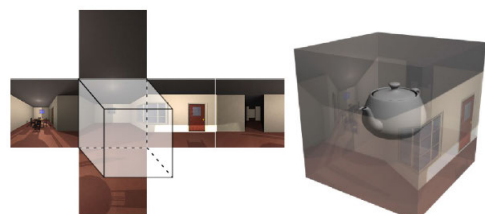


図 9 キューブマッピング

Fig. 9 Cube mapping

6.1.2 放物面マッピング

放物面マッピングとは、図 10 のような放物面への映り込みの特性を用いた環境マッピングである。テクスチャ座標の算出方法としてはまず、キューブマッピングと同様に復元した形状の各頂点の反射ベクトル $r = (r_x, r_y, r_z)$ を求める。そして、 r の始点を図 11 のように放物面の焦点に置いた時に、以下の式でテクスチャ座標 (s, t) を求める。

$$\begin{cases} s = \frac{r_x}{2(1+r_y)} - 0.5 & (4) \\ t = \frac{r_z}{2(1+r_y)} + 0.5 & (5) \end{cases}$$

テクスチャには魚眼レンズを搭載したカメラでビデオキャプチャしたものを用いる。これによってその場で撮影している周囲の環境を人体に映りこませることが可能となる。

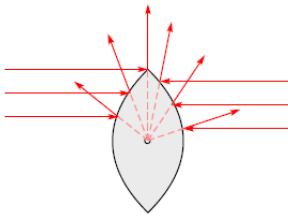


図 10 放物面への映り込み

Fig. 10 Reflection of the paraboloid

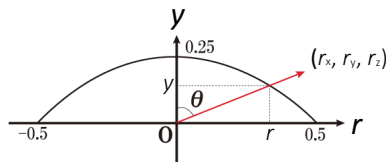


図 11 テクスチャ座標の算出

Fig. 11 Calculation of texture coordinates

6.1.3 プロジェクタ光の影響

プロジェクションマッピングを行う際、距離画像センサからの赤外線照射による計測と同時に、プロジェクタによる投影を行うことになる。よって、プロジェクタ光がセンサの計測に支障をきたす恐れがあった。

そこで、プロジェクタ光の有無の違いによるデプス値の比較実験を行った。図 12, 13 はそれぞれ、対象物体にプロジェクタ光を投影している場合、投影していない場合でのデプス値の計測結果（連続した 1000 フレーム間、ある 1 点のデプス値を計測したもの）である。その結果、プロジェクタ光無しの場合標準偏差が約 0.0053m、プロジェクタ光有りの場合標準偏差が約 0.0056m となり、プロジェクタ光有りの場合に標準偏差が少し増加するという結果が得

られた。

そこで、それぞれ次の 1000 フレーム分の標準偏差も求めると、表 1 のように、逆にプロジェクタ光無しの場合の標準偏差が大きくなったため、これはプロジェクタ光の有無に関係しない誤差の範囲であると考えた。よって、プロジェクタ光によるデプス値計測への影響は無いと考えられる。

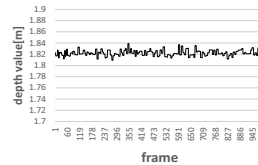


図 12 プロジェクタ光無し

Fig. 12 Projector light OFF

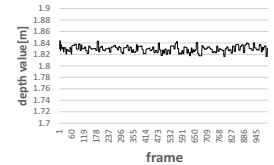


図 13 プロジェクタ光有り

Fig. 13 Projector light ON

表 1 デプス値計測結果

Table 1 Result of measuring the depth value

フレーム	標準偏差 (プロジェクタ光無)	標準偏差 (プロジェクタ光有)
1 ~ 1000	約 0.0053m	約 0.0056m
1001 ~ 2000	約 0.0059m	約 0.0054m

6.2 実験

運動する人体に対するプロジェクションマッピングの実験として、まずセンサ 1 台で復元した形状を、プロジェクタ 1 台から人物に投影する実験を行った。図 14 がその様子である。テクスチャマッピングの手法はキューブマッピングを用いた。実験した結果、人体の運動に対してほぼ遅延することなく投影像も変形していることが確認できた。現在、環境マッピングの際に視線ベクトルの始点を OpenGL のワールド座標系の原点に設定しているが、今後は実際の観測者の位置をトラッキングして視線ベクトルを求めることで、観測者の位置に応じたサンプリングの実装を行う。



図 14 運動する人体に対するプロジェクションマッピング

Fig. 14 Projection mapping on the moving human body

7. おわりに

本研究の提案手法により3台の距離画像センサを用いて、運動する物体の全周形状を動的に復元することができた。

そして、センサとプロジェクタを1台ずつ使って、運動する人体の1面に対して、動的にプロジェクションマッピングをすることができた。今後は、センサ3台でデプスマップ融合した形状を、プロジェクタ3台で人物の全周に対して投影する実験を行う。

また、現在は人物全体にテクスチャを映り込ませる環境マッピングを行っているが、衣服のテクスチャなど人物の部位単位のテクスチャを使用したプロジェクションマッピングの例も多く存在する。現段階ではそういったテクスチャを人物の動きに合わせて変形させることができない。これに対しては、センサから得られる骨格情報を用いるか、頂点群の時間的変化を見て変形させる方法を検討する。

参考文献

- [1] DrawLight.net : Rabarama 360° Full 3D Mapping Projection, 入手先 (<http://www.drawlight.net/en/project-details/items/rabarama-360-full-3d-mapping-projection.html>) (参照 2014.12.10).
- [2] the creators project, BOX, 入手先 (<http://thecreatorsproject.vice.com/show/projection-mapping-and-robots-combine-in-bot-dollys-new-film>) (参照 2014.12.10).
- [3] Nobumichi Asai, OMOTE, 入手先 (<http://www.nobumichiasai.com/jp/omote.html>) (参照 2014.12.10).
- [4] T.Kanade, P.J.Narayanan and P.W.Rander, Virtualized reality: concepts and early results, Proc. of IEEE Workshop on Representation of Visual Scenes, pp.69-76, 1995.
- [5] 富山仁博, 片山美和, 岩館祐一, 今泉浩幸: 視体積交差法とステレオマッチング法を用いた多視点画像からの3次元動オブジェクト生成手法, 映像情報メディア学会誌, Vol. 58, No. 6, pp. 797-806, 2004.
- [6] R Furukawa, R. Sagawa, H. Kawasaki, K. Sakashita, Y. Yagi, N. Asada : one-shot entire shape acquisition method using multiple projectors and cameras, Fourth Pacific-Rim Symposium on Image and Video Technology, IEEE Computer Society, pp. 107-114, 2010.
- [7] Jiawen Chen, Dennis Bautembach, Shahram Izadi : Scalable Real-time Volumetric Surface Reconstruction, ACM Transactions on Graphics (TOG) 32.4 (2013): 113.
- [8] 池畑諭, 山崎俊彦, 相澤清晴: 多視点ステレオのための信頼度に基づく高精度デプスマップの推定と融合, 画像の認識・理解シンポジウム (MIRU 2011), pp. 1405-1412, 2011.
- [9] Jianguo Li, Eric Li, Yurong Chen, Lin Xu, Yimin Zhang : Bundled depth-map merging for multi-view stereo, IEEE Computer Vision and Pattern Recognition, pp. 2769-2776, 2010.