Vol.2012-HCI-146 No.3
2012/1/19

IBVHの形状誤差における補正箇所の推定

宮田慎也[†1]　坂本竜基[†1]

自由視点映像の生成手法としてIBVHが注目されている．IBVHは，Visual Hullのうち仮想視点からみた表面までの奥行きのみを推定する手法である．しかし，Visual Hullは，被写体を凸包体で近似するため，本来の形に対して冗長な体積が発生してしまう．これは，特に丸みのある形状を復元する際によく発生するため，被写体が人間であることが多い自由視点映像では問題となる．そこで，本稿では，人間のような滑らかな形状においては，このような形状誤差はシルエットが形成する視体積の輪郭が交差する箇所に存在することが多いという仮定の下，このような箇所をIBVHにおける奥行情報から推定する手法について述べる．

Estimation for Redundancies on Image-Based Visual Hulls

Shinya Miyata[†1] and Ryuuki Sakamoto[†1]

Image-based visual hulls(IBVH) is a method for generating free-viewpoint videos, and only the depths from virtual camera to the surface of the visual hull are estimated instead of reconstructing whole visual hull. Since the visual hull is approximation of target object as convex hull, it includes redundancies around crossing points of silhouette volumes. This commonly occurs for round shape objects, and thus human is not good target for estimating correct shape. In this paper, we propose a method for estimating that redundancies from depth in IBVH outcome.

1. はじめに

次世代の映像コンテンツとして自由視点映像が注目されている．これは，従来の固定的な視点の映像ではなく，ユーザが任意の位置に視点をインタラクティブに移動させて閲覧可能な映像である．自由視点映像は，いくつかの生成手法が提案されているがマルチカメラで撮影された映像から被写体の3次元形状を推定して3DCGとして復元するアプローチが一般的になりつつある．この形状は，各カメラの被写体部分のシルエットが形成する視体積が重なる部分であるVisual Hull[1)]と呼ばれる領域として推定され，Voxelとして推定したうえでポリゴンメッシュやマイクロファセット[2)]としてレンダリングされるのが一般的である．

一方，近年は形状全体を復元するのではなく，視点からVisual Hullの表層部分までの深度のみをピクセル単位で推定し，レンダリングするIBVH（Image-Based Visual Hulls）[3)]と呼ばれる手法も注目されている[4),5)]．IBVHは，既存のVisual Hull全体を推定する手法に比べてレイトレーシング法のように各ピクセルに対して処理をおこなうため，GPUによる並列化に適しておりリアルタイムなレンダリングが可能な点と，画素毎の推定をおこなうため比較的高精度な点で有利である．

しかし，Visual HullもIBVHも推定結果は被写体を視体積，つまり凸包体で近似した形状となるため，曲面や凹みといった凸包体では表現ができない複雑な形状を精度よく推定することはできない．特に，ほとんどの部分が曲面である人間を被写体とした場合は，このような誤差が様々な箇所で発生してしまう．そこで，本研究では，このような形状誤差が凸包体内の突き出た部分，すなわちシルエットから得られた視体積の輪郭が交差する箇所に特に多く存在していることに注目し，IBVHの深度情報を基に誤差が発生しやすい箇所を推定する手法を提案する．また，この箇所は角ばった形状をしているためその部分を滑らかにし，補正をおこなった結果についても報告する．

2. Visual Hullの問題点

IBVHは入力となるカメラである参照カメラと出力となる任意の視点を表す仮想カメラで構成され，仮想カメラの光学中心から画像平面の各画素に対してそれぞれ飛ばされたレイが，被写体とどこで交差しているかを参照カメラ画像上でのレイとシルエットの交点から求めることで任意視点から見たシーンを生成する (**図1**)．

IBVHでは，実際にVisual Hullの復元までは行わないが，結果として復元したVisual Hullに対して仮想視点から見たときの表面の情報のみを利用していることに相当する．そのため，Visual Hullに形状誤差が含まれる環境では，IBVHを用いた場合でも同様に生成結果に影響を及ぼすことがある．

例えば，**図2**のようにある被写体を撮影したカメラの中間に仮想視点を配置した時を考えよう．この時，誤差のないモデルに対してテクスチャマッピングを行った場合には図2左

†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

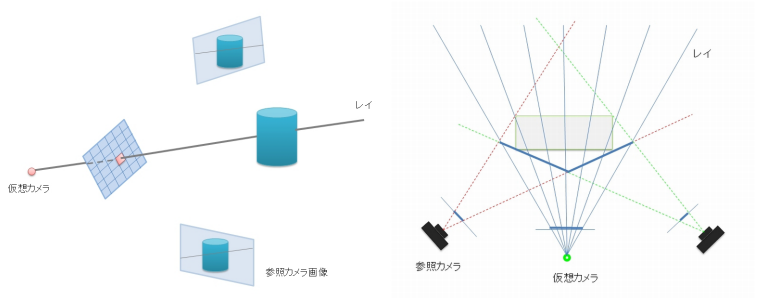

図 1 Image-Based Visual Hulls
Fig. 1 Image-Based Visual Hulls

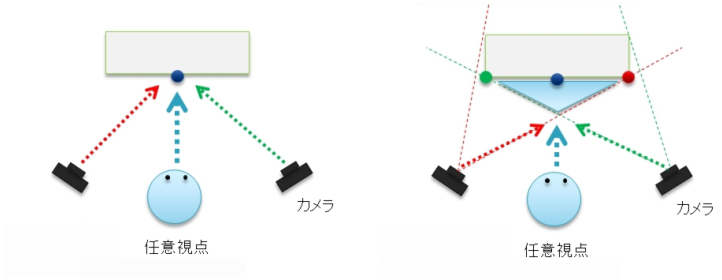

図 2 誤差のあるモデルが引き起こすテクスチャのずれ
Fig. 2 Invoking texture mapping errors by inexact estimated surface.

のように注目している点とテクスチャの元となるそれぞれのカメラから見た点は同じ箇所を映したものとなるため，どちらのカメラが選択された場合でも結果は等しくなる．しかし，誤差のあるモデルでは図 2 右のように注目している点は実際には誤差によって手前の点となり，これを左のカメラから見た場合には右端の点が，右のカメラから見た場合には左端の点の情報が得られることになり，これらの点は大きく離れた異なる点であるため，結果として映像に不整合をきたす結果となる．

従来の撮影環境は数百台という膨大な数のカメラを設置したスタジオ環境であることが多く，またそれに伴って自由視点映像の生成手法もそれを前提とした手法が一般的である．形状誤差は一般的にはカメラの台数が多くなるにしたがって小さくなるため，このような環境では無視できることがほとんどであった．しかし，このような環境を対象とした手法は本来利用される環境から考えると実用的ではなく，また，カメラを用意するコストの面からも多数のカメラを設置して撮影するということは好ましくない．そのため，このような環境を対象として自由視点映像の生成を行う場合には，カメラが少数台の環境であっても不整合が起こらないよう抑制するための補正手法が必要となる．

3. 関 連 研 究

Visual Hull での近似によって生じる形状誤差を小さくすることを目的とした研究はいくつかなされている．

延原らは弾性メッシュモデルを提案し，形状モデルの高精度化を行っている[6]．この研究では，ボクセルモデルとして復元した Visual Hull をメッシュモデルへと変換し，実際の形状が Visual Hull 内部に含まれることを利用して，メッシュモデルを初期モデルとしてそこから整合性を満たすように内部へと補正する手法が提案されている．しかし，整合性を取るためには複数のカメラにおいて情報が共通する領域が必要となるが，少ない台数で撮影を行った場合にはそもそも重なる領域は広くないうえ，非線形最適化問題であるため，初期モデルには比較的高い精度が必要となる．

また，富山らはステレオマッチングを利用してモデルを補正する手法を提案している[7]．この研究では，ステレオ視によって得られた各カメラから被写体までの実際の距離を用いて，MBR で復元した形状との差分，すなわち形状誤差である部分を推定し，その分を外力によって押し込むことによって補正を行う．しかし，この手法でステレオマッチングを行うためには，1 つの視点に 1 台のステレオカメラ，あるいは 2 台のカメラが必要となり，これは余分なカメラを用意しなければならないことを意味する．しかし，これらの手法は計算コストがかかるため，インタラクティブコンテンツとして自由視点映像を捉えた場合は問題となる．

4. 提 案 手 法

ここで，**図 3** のような IBVH を用いて推定した Visual Hull の表面について考えよう．点 A，B はどちらも IBVH によって求まった点なのだが，点 A では形状誤差が大きく，実際の位置とは離れた点であるのに対し，点 B では形状誤差が小さく，実際の形状に近い位置が求まっていることがわかる．点 A は凸包体の中で最も突き出た (以下，ピークと呼ぶ) 点であり，この点が 2 台のカメラのシルエットから形成される視体積において輪郭が交差する点で形成されていることがわかる．形状誤差は，ピークに近いところほど大きく，反対に遠

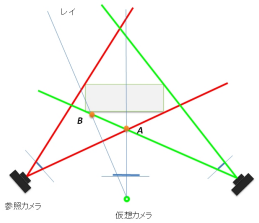

図 3 IBVH における誤差のある箇所の推定
Fig. 3 Estimation of redundancies in IBVH.

ざかるほど小さくなる．

そこで，本研究では，形状誤差が 2 台以上のカメラの視体積の輪郭上の交点に多く存在し，その点をピークとして周囲に分布していることが多いことに注目して，この箇所を推定したうえで滑らかな表面になるように補正する．ここで，視体積の中で輪郭に該当する部分は参照カメラ画像上では，被写体を抽出したシルエット映像のエッジであるため，IBVH で求まった点を各参照カメラに投影し，2 台以上でエッジに投影されたものをピークとして推定する．

具体的にピークを推定し，補正する手順は以下のようになる．

(1) IBVH によって仮想視点から Visual Hull の表面までのデプスを求める
(2) デプスから求めた 3 次元点を各参照カメラ上に投影し，その画素がエッジとなるか調べる
(3) 2 台以上の参照カメラでエッジとなる点であれば，その点をピークとする
(4) ピークとして推定された点から，凸形状を形成しているベクトルを求める
(5) ピークの点，およびピークからそれぞれ (4) で求めたベクトル方向に一定距離にある点を含めた 3 点からバーンスタイン基底関数で補間する
(6) 補間を行ったピクセルに対し，平均化フィルタをかける

このうち，処理 (4) では，そのピークの凸形状を形成している 2 つのベクトルを求めるが，これは (3) でエッジに投影されたカメラの光学中心とピークとなる点から求めることができる (**図 4**).

また，処理 (5) では，**図 5** のように (4) で求めたベクトル方向にそれぞれ 3 次元的に一定距離の 2 点，およびピークの点を制御点として，3 点をバーンスタイン基底関数により補間することでデプスの補正を行う．このとき，この 3 点を仮想カメラの画像座標へと投影しておくことで，IBVH で求めたデプスを用いて二次元的に補正を行うことができる．

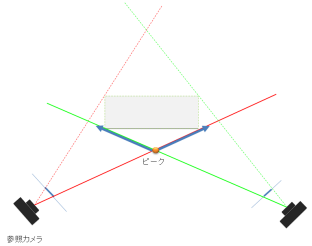

図 4 ピークと光学中心の関係
Fig. 4 Relationship between the peak and camera centers.

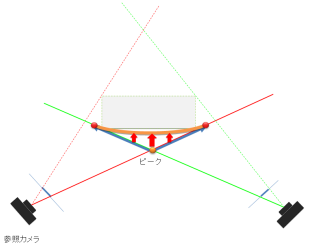

図 5 バーンスタイン基底関数による補正
Fig. 5 Interpolation with the Bernstein basis function.

5. 実　　験

5.1 GPGPU を用いた実装

提案手法は IBVH を用いた生成を行っているが，1 画素毎にレイを想定して交差判定を行うという独立した処理が多くを占めている．また，補正のステップも本質的には独立しているため，これらは並列計算が可能である．そこで，プログラマブルな GPU を利用する並列計算用ライブラリである CUDA を用いて，生成から補正までの手順を並列的に処理を行った．実装したプロトタイプでは，CUDA は，参照カメラ毎のシルエット画像，射影変換行列，カメラ位置，および仮想カメラの射影変換行列，カメラ位置を入力として，仮想カメラでの参照カメラに対するテクスチャ座標を出力した．

5.2 撮影環境と事前準備

図 6 のような環境に XGA の解像度をもつ IEEE1394 カラーカメラ (Pointgray 社製 Dragonfly2) を 8 台設置した．これらのカメラの射影変換行列は，予めカメラキャリブレーションによって求めておいた．また，被写体のシルエット画像は通常，背景差分をはじめとした手法でレンダリング時に求めるのが望ましいが，これらの手法では得られるシルエット画像の精度は一定ではない．今回は補正による効果のみを確かめるため，極力セグメンテーションによって精度が低下した場合の影響を受けないよう，撮影した映像に対してあらかじめ GrabCut[8] によってセグメンテーションを行い，比較的精度の高いシルエット画像を作

図 6 撮影環境とカメラ位置
Fig. 6 The studio and camera positions.

成しておいた (**図 7**).

5.3 結　　果

5.3.1 ピークの推定

5.2 節の環境で撮影した映像より IBVH を用いて，レンダリングを行った．この時のレンダリング結果と推定したピークを**図 8** に示す．ピークとして推定された白線の周辺のデプスマップを注目して見てみると，ピークを境界として表面の傾きが変化していることわかる．

これは，ピークを挟んで表面を切り取っているカメラが変わっており，その間の最も形状誤差が大きくなっている領域がピークとなっていることを表している．これより，提案手法によって形状誤差が最も大きくなるピークが検出できていると考えられる．

5.3.2 補 正 結 果

図 9 に同条件下で IBVH を用いて計算したデプスマップ，および提案手法で補正後のデプスマップを示す．左の IBVH で求めた補正前のデプスマップでは，表面の傾きがある点で大きく変化しており，角ばった形状となっているのがわかる．一方，右の補正後のデプスマップでは，補正をおこなったことによりピークの角が取れ，滑らかになっていることがわかる．

次に，**図 10** に推定されたピーク，および IBVH で求めたデプスマップに対して補正を行った過程を示す．(a) が推定されたピーク，(b) が IBVH のみでレンダリングした際の出力結果である．これを，(5) で補間した結果が (c)，そこから (6) で平滑化した結果が (d) である．(b) を見ると，亀裂が入っており，服の模様が連続していない．これは，(a) でみられる左側のピークによって実際の表面よりも手前の位置を取得しており，本来の結果よりも右側に縮んで引っ張られた状態となっていることに起因する．これにバーンスタイン基底関

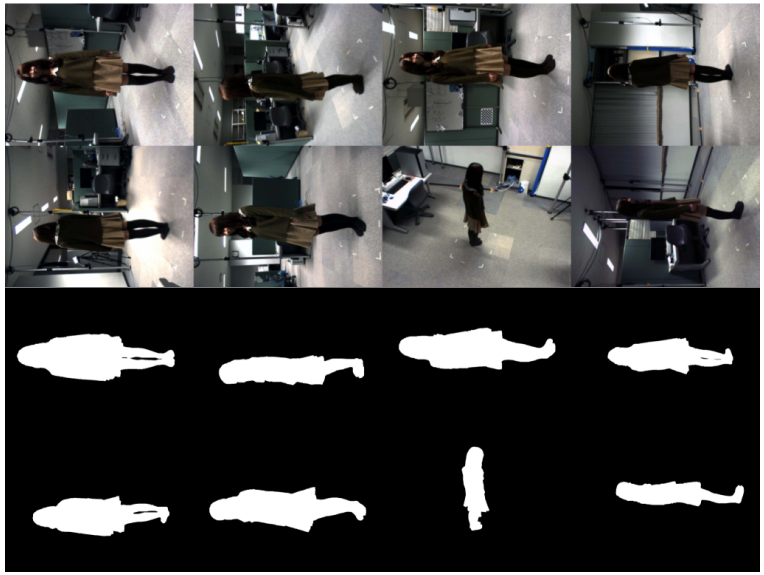

図 7 取得画像とシルエット画像
Fig. 7 Source images and silhouette images.

数を用いて補正をおこなうことで，ピーク周辺の形状誤差が小さくなり，(c) のように服の模様もほぼ連続したものへと変化している．一方で，(c) では量子化誤差や補正の範囲の影響で周囲との深度に差が生じ，ところどころで穴が開いたような状態となる．そこで，これに対して平滑化をおこなうことで周囲の画素との深度の差が低減され，(d) のようにより滑らかにすることができる．

その他の補正前後の比較は**図 11** に示す．いずれも左側が補正前，右側が補正後の出力画像である．

6. おわりに

本論文では，Visual Hull の形状誤差が凸包体内の突き出た部分，すなわちシルエットから得られた視体積の輪郭が交差する箇所に特に多く存在していることに注目し，IBVH によって得られた深度情報を基に誤差が発生しやすい箇所を推定する手法を提案した．同時に，ここで推定された箇所は角ばった形状をしていることから，これを滑らかにする補正手

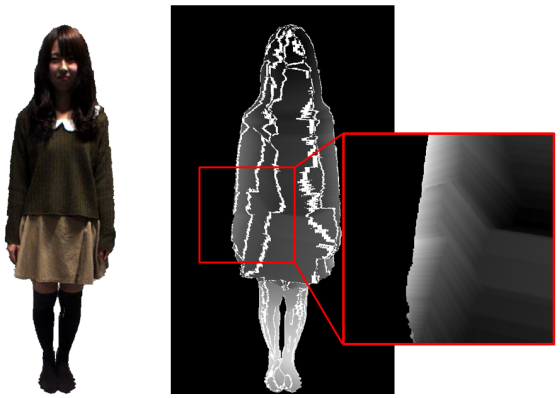

図 8 推定されたピークとその表面
Fig. 8 Estimated peaks.

法を示した．その結果，提案手法によって推定した箇所で表面の傾きが大きく変化しており，この箇所に形状誤差が多く存在していることが確認できた．また，補正は，角ばっていた表面を滑らかにし，実際の形状に近づける効果があることを確認した．

しかし，今回の手法では，IBVH によって求めた表面までの深度情報のみを利用しているため，ピークが端に存在し，制御点が裏側となってしまう場合にその点の深度情報が得られないため，補正がうまく機能しないパターンが存在した．今後は，このような場合にも対応した補正手法について研究していく予定である．

参 考 文 献

1) Magnor, M., Pollefeys, M., Cheung, G., Matusik, W. and Theobalt, C.: *Video-based rendering*, AK Peters (2005).
2) 山崎俊太郎，佐川立昌，　川崎洋，池内克史，坂内正夫：微小面ビルボーディングを用いた複雑なシーンの表示手法，画像の認識・理解シンポジウム (MIRU2002) 予稿集, Vol.1, pp.127–132 (2002).
3) MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S.J., and MCMILLAN, L.: Image-based visual hulls, *Proc. ACM SIGGRAPH 2000*, pp. 369–374 (2000).
4) Yue, Z., Zhao, L. and Chellappa, R.: View synthesis of articulating humans using visual hull, *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, Vol.1, IEEE Computer Society, pp.489–92 (2003).

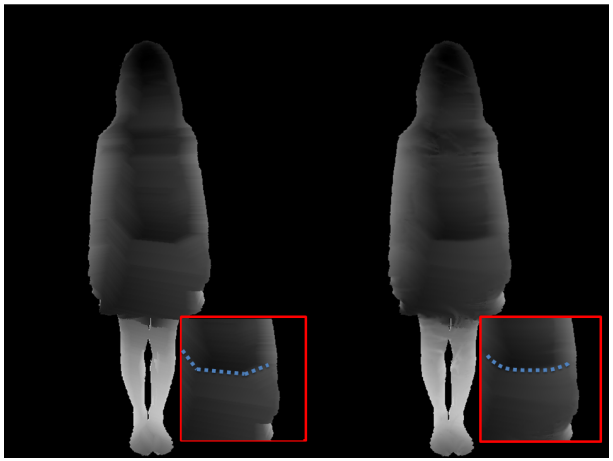

図 9 補正前後のデプスマップの変化
Fig. 9 Correction for the depth map.

5) Hauswiesner, S., Straka, M. and Reitmayr, G.: Coherent image-based rendering of real-world objects, *Symposium on Interactive 3D Graphics and Games*, I3D '11, ACM, pp.183–190 (2011).
6) 延原章平，和田俊和，松山隆司：弾性メッシュモデルを用いた多視点画像からの高精度 3 次元形状復元，情報処理学会論文誌. コンピュータビジョンとイメージメディア, Vol.43, No.11, pp.53–63 (2002).
7) 冨山仁博，片山美和，　折原豊，岩舘祐一：局所的形状特徴に拘束された 3 次元形状復元手法とそのリアルタイム動画表示，映像情報メディア学会誌：映像情報メディア, Vol.61, No.4, pp.471–481 (2007).
8) ROTHER, C.: GrabCut: interactive foreground extraction using iterated graph cuts, *ACM Transactions on Graphics*, Vol.23, No.3, pp.309–314 (2004).

Vol.2012-HCI-146 No.3
2012/1/19

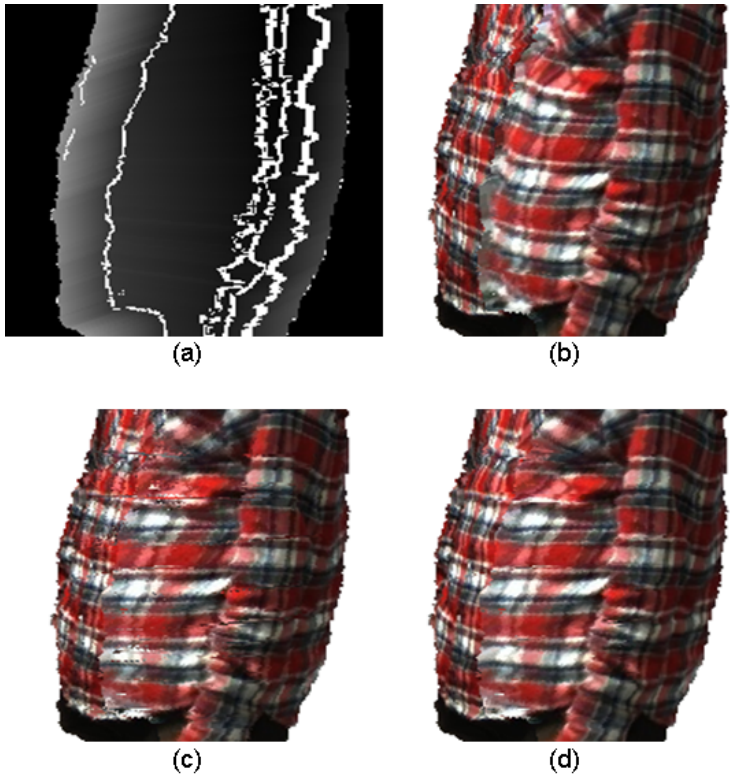

図 10 推定したピークと補正前後の比較
Fig. 10 Result of the corrected surface with texture.

図 11 (左) 補正前 (右) 補正後
Fig. 11 Left:naive rendering. Right:Corrected rendering.