

# Visual hullとMicrofacet billboardingを用いた自由視点サッカー視聴システムの検討

末永 諒<sup>1,a)</sup> 鈴木 一克<sup>2</sup> パナヒプル テヘラニ メヒルダド<sup>1</sup> 高橋 桂太<sup>1</sup> 藤井 俊彰<sup>1</sup>

**概要:** 自由視点視聴システムは、ユーザーが、3D オブジェクトを任意の視点から見ることできるシステムである。これは、スポーツの分野、特にサッカーと相性がいいとされており、サッカーにおける実用的な自由視点視聴システムの実装が要求されている。しかし、合成画像の品質の低さや、複雑なマニュアル操作が要求されることが実用化の障害となっている。そこで本稿では、マニュアル処理を可能な限り減らした自由視点サッカー視聴システムの開発に向けて、背景差分法により自動で分離された前景を用いて Visual hull を自動で生成し、これを入力として Microfacet billboarding を用いて自由視点映像を表示し、その有効性を確認した。

## 1. はじめに

ユーザーが自由な位置の視点の映像を見ることできる自由視点映像 [1] は、次世代の放送システムとして関心を集めている。一般的な放送では、サッカーのように、広大なフィールドに多数の選手が存在するスポーツでは、視聴者は放送者の選択した視点からの映像を見るだけである。そのようなスポーツに自由視点映像の技術を利用することで、視聴者は任意の視点からシーンを視聴することができる。そのため、自由視点映像技術を用いた放送システムは、まさに理想のスポーツ視聴システムであるといえる。特に、サッカーは、日常的に放送、または配信されるコンテンツであるため、サッカーを対象とした自由視点視聴システムの開発が求められている。しかし、文献 [2], [3] で述べられているように、合成画像の品質が低いことや、マニュアル処理が多数要求されることが主な要因となり、実用化はされていない。そこで、本研究では、マニュアル処理の自動化に焦点を当て、サッカーの試合における、多視点映像の撮影から自由視点映像の表示まで、可能な限りマニュアル処理を排除したシステムを開発することを目的として、シーンを前景と背景に分離し、前景画像から Visual hull [4] を生成し、Microfacet billboarding [5] を用いて自由視点映像の表示を行う手法を提案する。

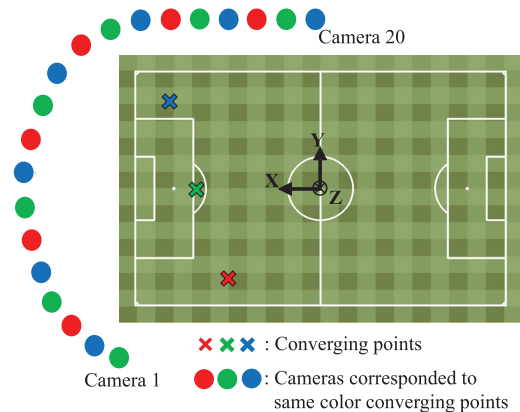


図 1 カメラ配置と世界座標系

## 2. 自由視点サッカー視聴システム

自由視点映像生成には多視点映像が必要であるため、筆者らは、TOYOTA スタジアムで行われた大学サッカーの試合を撮影した。撮影条件として、業務用フル HD カメラ (CANON XF305) 20 台を図 1 のように配置した。このとき、20 台のカメラは、図 1 に示すように 3 台ずつ光軸を変え、すべてのカメラを GENLOCK 信号で同期させ撮影した。また、本研究で使用する世界座標系も図 1 に示す。このようにして撮影した映像を用いて以下の処理手順を実装する。

まず、各カメラでチェッカーボードを撮影し内部パラメータを求める。次に、サッカーフィールド上のラインの交点を特徴点として外部パラメータを求める。ここで、内部パラメータと外部パラメータは二つ合わせてカメラパラ

<sup>1</sup> 名古屋大学大学院 工学研究科  
Furo-Cho, Chikusa-Ku, Nagoya, Aichi 464-8603, Japan  
<sup>2</sup> 名古屋大学大学院 情報科学研究科  
Furo-Cho, Chikusa-Ku, Nagoya, Aichi 464-8603, Japan  
<sup>a)</sup> suenaga@fujii.nuee.nagoya-u.ac.jp

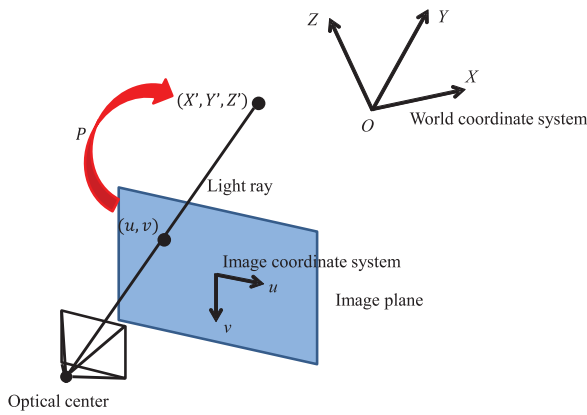


図 2 カメラパラメータによる画像座標と世界座標の対応付け

メータと呼ばれ、式 (1) で表されるように、画像上のある画素とその三次元座標を結びつける行列となっている。

$$\begin{bmatrix} x \\ y \\ s \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

$x, y$  はある画素の画像座標、 $X', Y', Z'$  はある画素の三次元座標、 $P$  はカメラパラメータ、 $s$  は定数倍の不定性を表している。そのモデルを図 2 に示す。そして、背景差分法 [6] を用いて前景と背景に分離する。これは、サッカーフィールドを撮影した際、前景にあたる、動きのある領域（選手、審判、ボール）がフィールド全体に占める割合はとも低く、背景であるフィールド平面は変化がほとんどないため、前景領域にのみ手の込んだ処理を施すことで計算コストを削減するためである。そして、前景と背景それぞれに処理を行い、得られたデータをビューワーに入力し、自由視点映像を表示する。

### 3. 提案手法

本章では、自動の前景処理手法および三次元レンダリング手法について提案する。提案手法では入力として、元画像、推定したカメラパラメータ、および前景マスク画像を使用する。前景マスク画像は、文献 [6] で紹介されている時間領域での背景差分法によってシーンを前景と背景に自動で分離し、得られた前景画像に対してノイズ除去と 2 値化処理を行い生成した。

はじめに、自動の前景処理手法である、Space carving method [7] を用いた Visual hull の生成手法を提案する。この手法では、フィールド上の図 3 の領域に、一辺 25mm の大きさの立方体（ボクセル）が、 $X$  軸方向に 1920 列、 $Y$  軸方向に 2560 列、 $Z$  軸方向に 128 列、隙間なく敷き詰められていると仮定する。このボクセルは、三次元座標、色情報、ボクセルの大きさの情報を持っている。まず、任意のボクセルの三次元座標から、ボクセルを構成する正方形そ

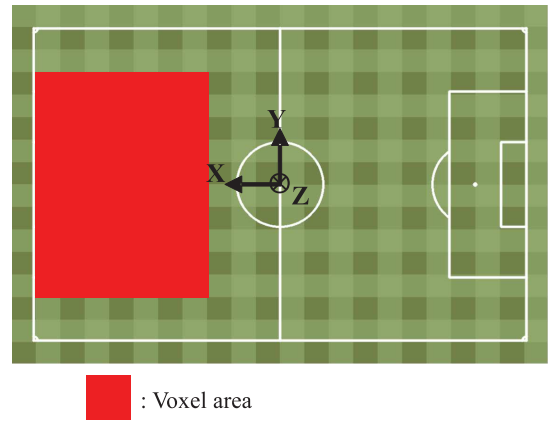


図 3 ボクセル空間

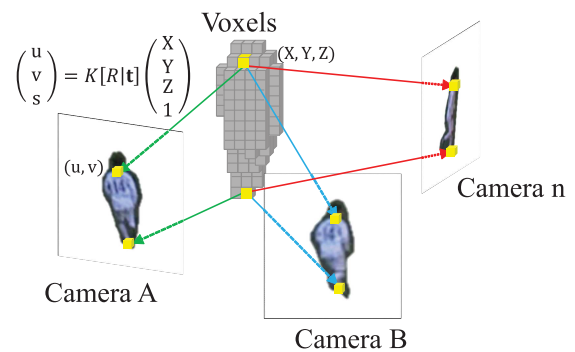


図 4 Space carving method

れぞれに対して、三次元座標から二次元座標へとカメラパラメータを用いて座標変換を行う。そうすることで、そのボクセルが撮影カメラ画像中でどの領域に映っているのかわかる（図 4）。これをボクセルの投影と呼ぶ。そして、すべてのカメラにボクセルを投影し、投影領域が物体領域と 1 ピクセルでも重なったカメラの数をカウントする。そして、そのカウントが一定数以上であった場合、そのボクセルは物体のものであるとみなしボクセルを残し、さもなければボクセルを削除する。この手順をすべてのボクセルに対して行い、最終的に残ったボクセルの集合が Visual hull となる。

しかし、一辺 25mm のボクセルが先述の空間に敷き詰められていると仮定すると、その数は 629145600 個となり、これらすべてに対してそれぞれ一つずつ処理を行うのでは計算コストが膨大になってしまう。そこで、八分木を用いて処理を効率化する手法を提案する。この手法は、まずはボクセル空間を立方体で粗く区切り、その区切られた領域をボクセルとして八分木の親ノードとする。そして、そのボクセルを全カメラに投影し、削除判定を行う。こうすることで、選手が確実に存在しない領域のボクセルを削除することができる。つまり、あらかじめ選手が存在するであろう領域を大まかに限定し、少ない計算コストで選手が存在しない領域を判別できる。次に、残ったボクセルは選手

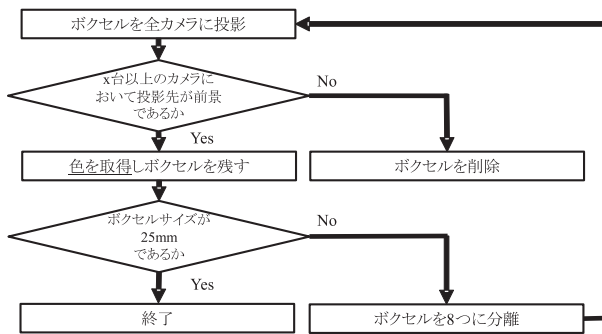


図 5 八分木を用いた Visual hull 生成フローチャート

が存在すると思われる領域であるので、その領域をさらに八等分する。そして、分割された領域をボクセルとして以下同様に処理していく。こうして、段階的にボクセルを小さくしていくことで、効率的に Visual hull 生成が可能になる。以下に提案手法の詳細を示す。まず、ボクセル空間を一辺 3200mm のボクセルの集合として、 $X$  軸方向に 15 列、 $Y$  軸方向に 20 列、 $Z$  軸方向に 1 列、ボクセルが敷き詰められているとする。この 3200mm のボクセルを八分木の親ノードとして、一つの親ノードに対する処理を図 5 に示す。ボクセル一つに対する処理は Space carving method を用いた Visual hull 生成手法と同様であり、まず、ボクセルを全使用カメラに投影し、ボクセルの削除判定を行う。削除されたとき、そのボクセルの処理を終了し次のボクセルの処理に移る。削除されなかったとき、全カメラ画像のボクセル投影領域中の前景マスク領域の色の平均をそのボクセルの色とする。そして、残ったボクセルがあらかじめ決めておいた最小サイズでなければ、そのボクセルを 8 等分し、分割されたボクセルそれぞれに対して同様に処理を繰り返す。もしボクセルが最小サイズ（一辺 25mm とする）であった場合、そのボクセルはそれ以上の分割を行わない。こうして、すべての枝の探索を行う。最終的に、一辺 25mm のボクセルによる Visual hull が生成される。

続いて、生成した Visual hull から自由視点映像を表示する手法を提案する。生成した Visual hull は前景の三次元再構成が行われたボクセル群であるため、これを三次元空間にレンダリングすることで前景の三次元モデルが復元される。ボクセル群の最も簡単な可視化手法として、ボクセルを一つの点として、点群を三次元空間に描画する手法が考えられる。実際に、点群の描画に優れる PCL - Point Cloud Library [8] を用いて、フィールドの CG モデルと Visual hull を点群で描画したものを図 6 (a) に示す。図 6 (a) からわかるように、仮想視点が点群に近づくと、オブジェクトが疎な点群表示となってしまう。これを解決するために、点群にメッシュを張る手法があるが、計算コストが非常に大きい。そこで、点ではなく、正方形を組み合わせてボクセルを描画する手法が考えられる。図 6 (b) は、生成した Visual hull に対して、グラフィックライブラリ

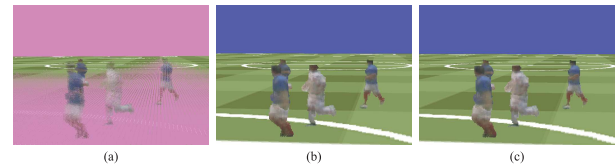


図 6 (a) 点群描画, (b) ボクセル描画, (c) Microfacet billboard を用いた描画

(OpenGL) を用いてボクセルを描画したものである。点群描画と比較すると、オブジェクトは十分に密な表示がなされたといえる。しかしながら、ボクセルのサイズは一辺 25mm とその形を確認できないほどに小さいため、描画を簡略化しても見た目に影響は出ないのではないかと考えた。そこで採用したのが Microfacet billboard である。この手法は、オブジェクトを小さなビルボードの集合で表現する。ここで、ビルボードとは、文献 [2], [3] に紹介されているように、ある平面に垂直に立つ板のことを指す。本手法では、一つのボクセル ( $r \times r \times r [mm]$ ) を一枚のビルボード ( $r \times r [mm]$ ) に置き換えることで Visual hull を描画する。図 6 (c) は、生成した Visual hull を、OpenGL を用いて微小なビルボードの集合で可視化したものである。ボクセルからビルボードに置き換えたために、見た目の品質の劣化が考えられるが、図 6 (b), (c) を見る限りでは劣化は全く見られない。

#### 4. 実験

まず、Visual hull 生成についての実験を行った。ボクセルの最大サイズを 1 辺 3200mm、最小サイズを 1 辺 25mm とした。図 7 の赤い領域に、最大サイズのボクセルを、 $x$  軸方向に 15 列、 $y$  軸方向に 20 列の計 300 個定義した。また、Visual hull 生成に使用するカメラは、計算コスト削減のため、選手がごくわずしか映っていない 8 台を除いた 12 台を使用する（図 7 参照）。図 5 中の削除判定に用いる閾値  $x$  は 9 とした。使用画像はゴールの瞬間の 5 秒間（150 フレーム）を用いた。また、このときの入力、元画像、文献 [6] の自動処理によって得られた前景画像に対して 2 値化とノイズ除去を行った前景マスク、カメラパラメータである。

八分木を用いた探索と、ボクセル空間を全探索する手法について、ボクセル探索回数と Visual hull 生成時間の全 150 フレームの平均を計測し、表 1 に示す。結果から、探索回数はおおよそ 4000 分の 1、探索時間はおよそ 3 分の 1 となった。探索回数に比べて探索時間の減少量が少ない原因として以下の理由が考えられる。全探索のボクセルと八分木の最小ボクセルである一辺 25mm のボクセルは、画像中ではせいぜい 1 ピクセルであるため、ボクセルを点として扱い、点の投影を行っている。しかし、八分木ではそれ以上の大きなボクセルを扱うため、点の投影ではなくボクセ

|                       | 全探索                 | 八分木                 |
|-----------------------|---------------------|---------------------|
| Visual hull 生成時間 [ms] | $4.018 \times 10^6$ | $1.341 \times 10^6$ |
| ボクセル探索回数 [回]          | $6.29 \times 10^8$  | $1.54 \times 10^5$  |

表 1 1 フレームあたりの平均 Visual hull 生成時間と平均ボクセル探索回数

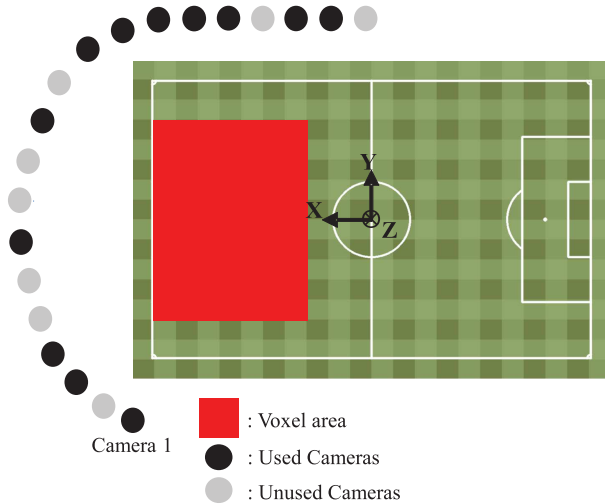


図 7 ボクセル空間と使用カメラ

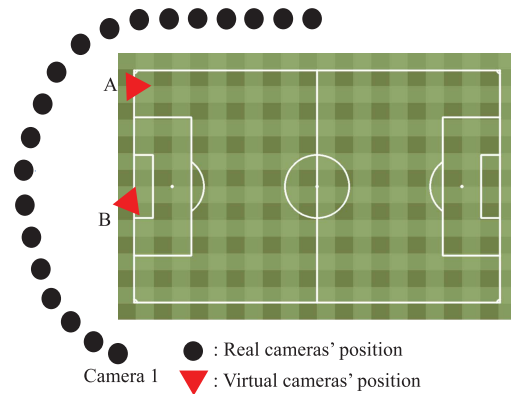
ルの面の投影が必要となってくる。そのため、ボクセルの底面を除いた 5 面を画像に投影しボクセルの投影領域としているため、処理時間が長くなってしまった。

こうして生成された自由視点映像を図 8 に示す。仮想視点位置は図 8 (a)、A からの仮想視点画像は図 8 (b)、B からの仮想視点画像は図 8 (c) にそれぞれ示す。このときボクセルサイズを一辺 25mm として Microfacet billboarding を適用したとき、平均表示フレームレートは 29.0fps を記録した。Microfacet billboarding を用いずにボクセルの描画を行った場合、平均表示フレームレートは 19.2fps であった。

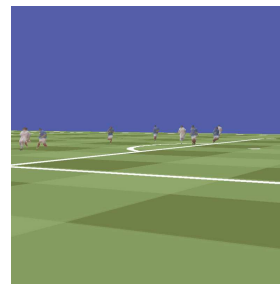
## 5. まとめ

マニュアル処理を可能な限り減らした自由視点サッカー視聴システムの開発のために、要素技術に提案手法を適用した。自動で抽出された多視点前景マスク画像をもとに Visual hull を生成し、それをを用いて自由視点映像を表示したため、マニュアル処理は多視点映像の撮影とカメラパラメータの推定のみとなった。そして、Microfacet billboarding を用いて描画することで、撮影時とほぼ同等の自由視点映像の表示フレームレートを記録した。以上から、提案手法の有効性を確認できた。

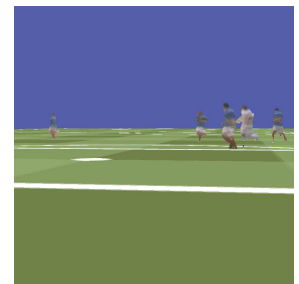
今後の課題として、現状では、Visual hull の生成の高速化、Visual hull の生成精度の向上が挙げられる。前者に関して、一つのボクセルの処理は単純であるため、GPU を使用した並列計算を行うことで処理の高速化が図れるのではないかと考えられる。後者に関して、カメラパラメータの推定精度の改善や、Visual hull 生成の際に、本研究では



(a) 仮想視点位置



(b) 仮想視点画像A



(c) 仮想視点画像B

図 8 仮想視点位置と仮想視点画像

ボクセルの再投影を行う画像には前景 2 値マスクを使用することで判定の簡略化を行っているが、再投影する画像に前景カラー画像を使用し、再投影先の色情報を参照することで Visual hull 生成の高精度化が図れると考えられる。

## 6. 謝辞

本研究の一部は、独立行政法人情報通信研究機構委託研究「革新的な三次元映像技術による超臨場感コミュニケーション技術の研究開発」課題カ 三次元映像 End-to-End 通信・放送システムの助成を受けたものである。

## 参考文献

- [1] T. Fujii, et al.: *Free-viewpoint TV system based on ray-space representation*, In Proceedings of the SPIE ITcom, Vol. 4864, pp. 175-189, 2002.
- [2] R. Suenaga, et al.: *A Practical implementation of free viewpoint video system*, IS&T/SPIE 2015 Electronic Imaging, 2015.
- [3] 三功 浩嗣, 他.: 選手領域の抽出と追跡によるサッカーの自由視点映像生成, 映像情報メディア学会誌, Vol. 68, No. 3, pp. J125 - J134 (2014).
- [4] T. Koyama, et al.: *Live Mixed-Reality 3D Video in Soccer Stadium*, Proc. of International Symposium on Mixed and Augmented Reality (ISMAR2003), pp. 178-187, Oct. 2003.
- [5] S. Yamazaki, et al.: *Microfacet billboarding*, Proceedings of the 13th Eurographics Workshop on Rendering (EGRW '02), pp. 169-180, 2002.
- [6] Y. Mase, et al.: *Precise Extraction of Players in a Soccer Game for Free-Navigation*, Proc. of the 1st Interna-

tional Conference on Advanced Imaging (ICAI2015), pp. 316–319, June 2015.

- [7] K. N. Kutulakos, et al.: *A Theory of Shape by Space Carving*, IJCV, Vol. 38, Issue 3, pp. 199–218, 2000.
- [8] Point Cloud Library, “<http://pointclouds.org/>”.