

ビルボードを用いた多視点映像からの効率的な仮想視点映像生成法

金野龍人[†] 藤本忠博^{††}

本研究では、複数のビデオカメラで撮影した対象物体の多視点映像をもとに、複数のビルボードで物体を疑似表現した3次元モデルにより、仮想視点映像を効率的に生成する手法を提案する。本手法では、フレーム時刻ごとに多視点映像にSURFを適用して推定した物体上の複数の3次元点上に仮想視点に向けてビルボードを配置する。全てのビルボードで仮想視点スクリーン全体を覆うように各ビルボードの大きさを決定し、各ビルボードに最適なビデオ映像を投影することで仮想視点映像を生成する。

An Efficient Billboard-based Method for Generating Virtual-viewpoint Images from Multi-view Images

Ryuto Konno[†] and Tadahiro Fujimoto^{††}

In this paper, we propose an efficient billboard-based method for generating virtual-viewpoint images from multi-view images on which a target object is captured by multiple cameras. First, at each frame time, 3D points on the object are obtained using SURF points extracted on multi-view images. Then, a billboard is placed on each 3D point such that the billboard is directed toward the current virtual-viewpoint. The size of each billboard is determined such that the projection of all billboards covers the virtual-viewpoint screen completely. Finally, a virtual-viewpoint image is generated by projecting an appropriate frame image onto the billboards.

1. はじめに

複数の異なる視点位置から対象物体をカメラ撮影した多視点カメラ映像を利用して任意の仮想視点位置から見た仮想視点映像を生成するイメージベースCG技術の研究が盛んに行われている。これまでに様々な手法が提案されているが、その代表的なものとして、以下の手法が挙げられる。

- 複数のカメラ映像上の対象物体の輪郭や色の情報を利用して、ボクセルモデルにより3次元形状を復元する手法[1]
- ステレオマッチングによる異なるカメラ映像間のピクセルごとの対応付けにより、対象物体までのデプス値を求めて3次元形状を復元する手法[2]

これらの手法では、カメラ映像の情報から物体の3次元形状を復元し、それを仮想視点からレンダリングすることで仮想視点映像を得る。しかし、ボクセルモデルによる手法は、ボクセル空間の解像度により形状の復元精度が限定される問題がある。また、ステレオマッチングによる手法は、ピクセルごとの対応付けに時間がかかり、リアルタイム処理が困難であるという問題がある。そこで、本研究では、リアルタイムでの仮想視点映像の生成を目標とし、ビルボードを用いて効率的に仮想視点映像を生成する手法を提案する。ビルボードを用いて物体の仮想視点映像を表示する代表的な方法として、文献[3]のマイクロファセットビルボードによる方法がある。この方法では、静的な物体のボクセルモデルを用い、各ボクセルにビルボードを配置することで物体の3次元モデルを表現する。一方、本研究の提案手法は、動的な物体をリアルタイムで表示することを目的とし、カメラ映像のフレーム時刻ごとに物体表面上の点群をリアルタイムで推定し、その各点上にビルボードを配置することで仮想視点映像を生成する。本手法の主な流れを図1に示す。全てのカメラをキャリブレーションした後、フレーム時刻ごとの処理として、まず、各カメラ映像上でSURF特徴点を抽出し、異なるカメラ映像間で共通の特徴点を対応付ける。次に、カメラキャリブレーションで得られた情報を用いて各特徴点の3次元位置を推定する。そして、その3次元位置に適切な大きさのビルボードを仮想視点に向けて配置する。最後に、ビルボード上に適切なカメラ映像の投影を行うことで、仮想視点映像をリアルタイムで生成する。

[†] 岩手大学大学院工学研究科
Graduate School of Engineering, Iwate University
^{††} 岩手大学工学部
Faculty of Engineering, Iwate University

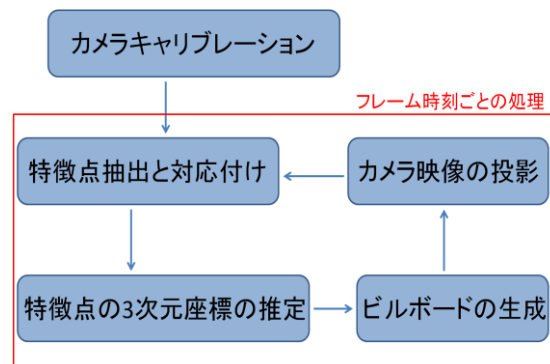


図 1 本手法の主な流れ

2. カメラキャリブレーション

カメラキャリブレーションとは、複数のカメラに対して、各カメラの内部パラメータと外部パラメータを求める処理である。内部パラメータは、焦点距離などの個々のカメラに固有のパラメータであり、通常は、カメラ座標系から画像座標系への変換行列の形式で表現できる。外部パラメータは、カメラ間の相対的な位置と姿勢をあらわすパラメータであり、通常は、ワールド座標系からカメラ座標系への変換を行うための平行移動ベクトルと回転行列の形式で表現できる。本研究における実験では、カメラキャリブレーションの実施にあたり、コンピュータビジョン用ライブラリ OpenCV で提供されている Zhang による平面を用いた方法を利用した。この方法では、図 2 のようなチェッカーパターンを持つ平面状のボードを用い、はじめに、このボードをカメラごとに複数回撮影することにより内部パラメータを求め、続いて、全てのカメラで一度に撮影することで外部パラメータを求めた。

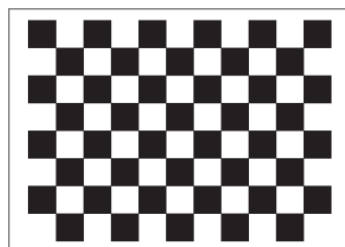


図 2 カメラキャリブレーション用チェッカーパターン

3. 特徴点の抽出と対応付け

本手法では、フレーム時刻ごとのカメラ映像内の特徴点の抽出と対応付けには、高速な特徴点抽出の方法である SURF (Speeded-Up Robust Features) [4, 5] という手法を利用する。この手法では、抽出した特徴点の特徴量が 64 次元、あるいは、128 次元の特徴ベクトルとして表現される。本研究における実験では、SURF 特徴点の抽出には OpenCV で提供されている関数を用いた。

複数のカメラ映像に対して、SURF 特徴点の抽出後、2 つのカメラ映像ごとに共通の SURF 特徴点の対応付けを行った。対応付けのための特徴量の類似度の評価基準として、2 つのカメラ映像内の特徴点の間で、特徴ベクトルの間のユークリッド距離が最短であり、かつ、その距離が事前に決めた閾値より小さいという基準で対応付けを行った。このとき、誤対応を減らすために、2 つのカメラ映像間の双方向で特徴ベクトルが最短距離となる特徴点の組だけを共通点として対応付けた。さらに、対応付けの正確さを増すため、最短距離のものだけでなく、2 番目に距離が近いものを求め、その二つの距離の比率が事前に決めた閾値よりも小さな場合のみ、対応付けの曖昧さが無いものとして、最短距離の特徴点の組を共通点として対応付けた [6]。

上記の方法で 2 つのカメラ映像間で特徴点を対応付ける場合、各カメラ映像内の全ての特徴点どうしを総当たりで組み合わせると特徴ベクトル間の距離を計算すると、特徴ベクトルは 64 次元あるいは 128 次元という高次元のため、大きな計算時間を要する。そこで、計算量を軽減し、同時に、対応付けの精度を上げるために、エピポーラ幾何を用いて組み合わせる特徴点を限定する。

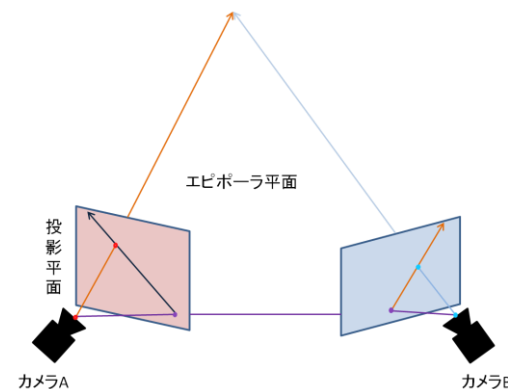


図 3 エピポーラ幾何

図 3 に示すように、3 次元空間上の点が 2 つのカメラ映像上に投影された場合、その 2

つの投影点は、エビポーラ平面によって決まる各カメラ映像上のエビポーラ線上に位置することになる。その2つの投影点の画像座標を同次座標で $[x_1, y_1, 1]$, $[x_2, y_2, 1]$ とした場合、それらは基礎行列 $\mathbf{F} = [f_{ij}]$ ($i, j = 1, 2, 3$)を用いて式(1)で関係付けられる。

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} = 0 \quad (1)$$

基礎行列 $\mathbf{F}$ は各カメラの内部パラメータと外部パラメータが固定されていれば不変である。そこで、カメラキャリブレーションの際に各カメラ映像間の基礎行列 $\mathbf{F}$ を求めておき、2つのカメラ映像間で特徴点を対応付ける際には、一方のカメラ映像内の一つの特徴点に対して、他方のカメラ映像内でその特徴点に対応するエビポーラ線上に位置する特徴点だけを対応付けの対象とする。これにより、総当たりの場合に比べ、計算時間は大幅に短縮される。

4. 特徴点の3次元位置の推定

カメラキャリブレーションで得られた内部パラメータ行列 $\mathbf{A} = [a_{ij}]$ ($i, j = 1, 2, 3$)と外部パラメータをあらわす回転行列 $\mathbf{R} = [r_{ij}]$ ($i, j = 1, 2, 3$)と平行移動ベクトル $\mathbf{T} = [t_x, t_y, t_z]^T$ を利用すれば、特徴点の世界座標 $\tilde{\mathbf{X}} = [X, Y, Z, 1]^T$ をカメラ

ラ映像上に投影した画像座標 $\tilde{\mathbf{x}} = [x, y, 1]^T$ (ともに同次座標) は式(2)で表される。

$$k\tilde{\mathbf{x}} = \mathbf{A}[\mathbf{R} | \mathbf{T}]\tilde{\mathbf{X}} \quad (2)$$

ここで、左辺の変数 k はカメラ視点から特徴点の3次元位置までのデプス値を表す。式(2)を変形することで式(3)が得られる。

$$\mathbf{X} = k\mathbf{R}^{-1}\mathbf{A}^{-1}\tilde{\mathbf{x}} - \mathbf{R}^{-1}\mathbf{T} \quad (3)$$

ここで、世界座標 $\mathbf{X} = [X, Y, Z]^T$ は同次座標ではなく、通常座標である。

3節の方法で2つのカメラ映像間で対応付けた特徴点について、各カメラの内部パラメータ行列を $\mathbf{A}_1, \mathbf{A}_2$, 回転行列を $\mathbf{R}_1, \mathbf{R}_2$, 平行移動ベクトルを $\mathbf{T}_1, \mathbf{T}_2$ とすると、式(3)より、それぞれの画像座標 $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2$ に対する世界座標 $\mathbf{X}_1, \mathbf{X}_2$ は式(4)で表さ

れる。

$$\begin{cases} \mathbf{X}_1 = k_1\mathbf{R}_1^{-1}\mathbf{A}_1^{-1}\tilde{\mathbf{x}}_1 - \mathbf{R}_1^{-1}\mathbf{T}_1 \\ \mathbf{X}_2 = k_2\mathbf{R}_2^{-1}\mathbf{A}_2^{-1}\tilde{\mathbf{x}}_2 - \mathbf{R}_2^{-1}\mathbf{T}_2 \end{cases} \quad (4)$$

式(4)中の世界座標 $\mathbf{X}_1, \mathbf{X}_2$ は理想的には一致するものであるが、実際にはカメラキャリブレーションの誤差等により一致しない。そこで、図4に示すように、式(4)の2つの式を k_1 と k_2 をパラメータとした2直線とみなし、この2直線が最も接近するときの各直線上の点 $\mathbf{X}_1, \mathbf{X}_2$ を考え、その2点の中点 $\mathbf{X}_{12} = (\mathbf{X}_1 + \mathbf{X}_2)/2$ を特徴点の世界座標とする。図4より、この2直線はベクトル $\mathbf{X}_2 - \mathbf{X}_1$ とそれぞれ直交するため、各直線方向ベクトルとベクトル $\mathbf{X}_2 - \mathbf{X}_1$ との内積は0となる。

$$\begin{cases} (\mathbf{R}_1^{-1}\mathbf{A}_1^{-1}\tilde{\mathbf{x}}_1) \cdot (\mathbf{X}_2 - \mathbf{X}_1) = 0 \\ (\mathbf{R}_2^{-1}\mathbf{A}_2^{-1}\tilde{\mathbf{x}}_2) \cdot (\mathbf{X}_2 - \mathbf{X}_1) = 0 \end{cases} \quad (5)$$

式(4)と(5)から k_1 と k_2 の値を求め、特徴点の世界座標 $\mathbf{X}_{12}$ を得る。

また、上記の方法で世界座標 $\mathbf{X}_{12}$ を求める際、2点 $\mathbf{X}_1, \mathbf{X}_2$ の距離が大きすぎる場合は、誤対応点である可能性が高いため、それらを除去する。

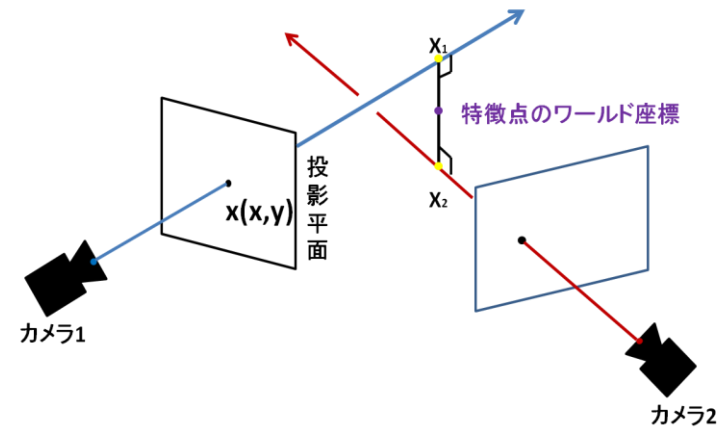


図4 特徴点の世界座標の計算

5. 特徴点位置に基づくビルボードの生成

本手法では、仮想視点に依存する方法でビルボードによる3次元モデルを構築する。図5に示すように、まず、4節の方法で推定したワールド座標位置にある各特徴点を仮想視点のスクリーンに向けて逆投影し、投影されるピクセル位置を算出する。このとき、スクリーンから外れるものは実際には見えていないものとして除外する。次に、図5のように、スクリーンができるだけ埋まるように、スクリーン上で各投影ピクセル位置を中心とした4角形領域を生成する。そして、各4角形領域を仮想視点から各特徴点のワールド座標位置まで投影して4角形ポリゴン（ビルボード）とする。

4角形領域の決め方は、各特徴点の投影ピクセル位置から周囲に向けて、左、上、右、下方向の順に1ピクセル幅ずつ領域を拡張させる方法を取る。図6のように、4角形領域ごとに、未処理のピクセルをたどっていき、拡張する4方向が他の領域の範囲に進入するか、または、スクリーンの4辺に到達するまで処理を続ける。そして、図7に示すように、上記の方法で得られた各4角形領域を仮想視点から各特徴点のワールド座標位置まで投影し、スクリーンに平行な4角形ポリゴンとしてビルボードを生成する。ただし、上記の方法だけを行った場合は、図8の赤い四角で囲んだ部分のように、全ての4角形領域の拡張が停止した後でスクリーン上に4角形領域で埋められない穴が生じる。これを解消するために、穴の領域に対して最近隣の4角形領域のビルボードと同じ奥行値を与えることで新たなビルボードを生成し、穴を埋める。

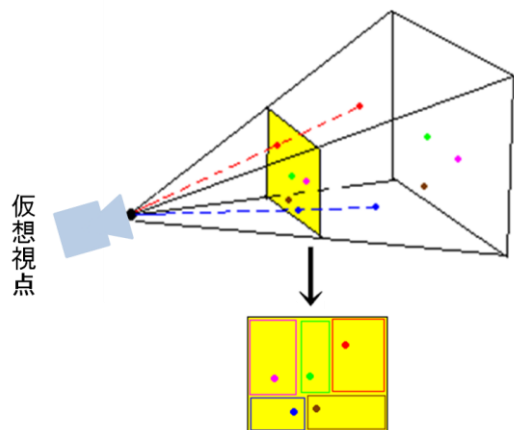


図5 特徴点によるビルボードの生成

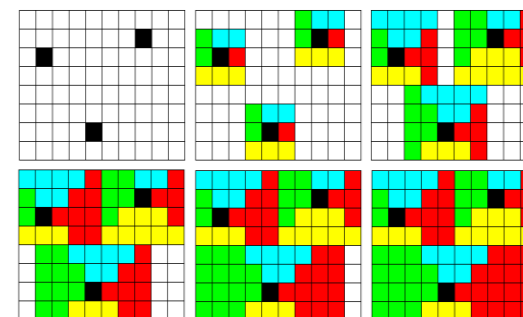


図6 4角形領域の生成

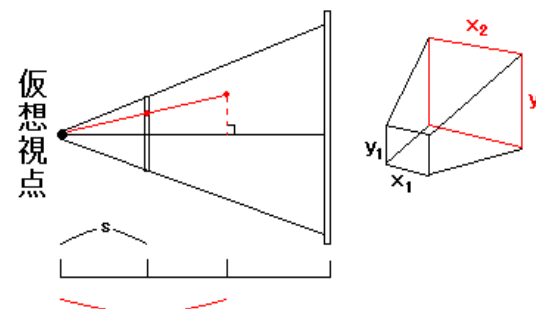


図7 ビルボードの生成

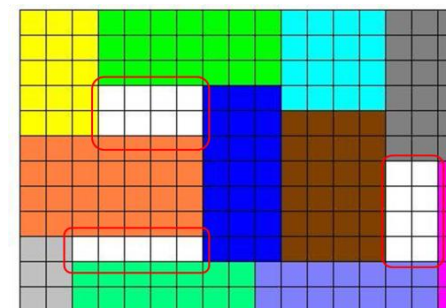


図8 穴領域の発生

6. カメラ映像投影による仮想視点映像の生成

本手法では、図 9 のように、5 節の方法で生成したビルボードに対してカメラ映像を投影することで仮想視点映像を生成する。図 10 に示すように、複数のカメラのうち、仮想視点と視線方向のなす角 θ が一番小さいものを、カメラ映像を投影する投影カメラとして選択する。

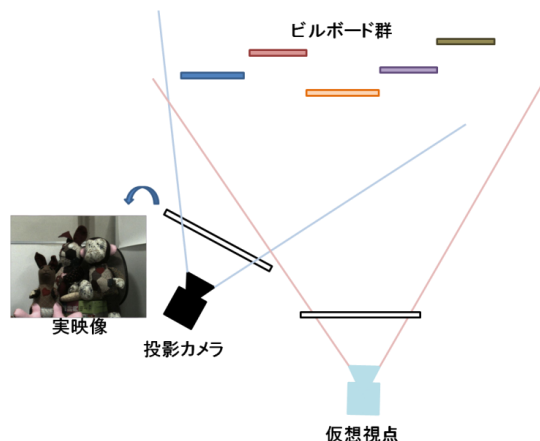


図 9 ビルボードへのカメラ映像の投影

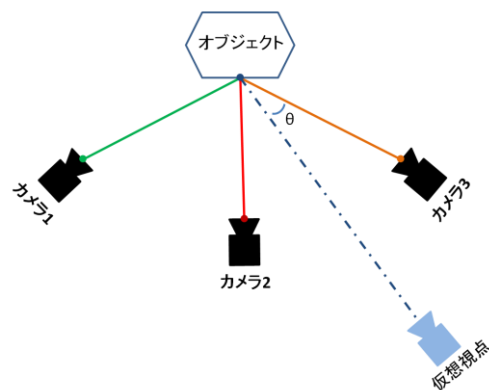


図 10 投影カメラの決定

7. 実験

7.1 実行環境

図 11 に示すように、今回の実験では、1 台の P C (CPU: AMDPhenom II X4 965 3.4GHz , メモリ:8GB) に接続した 3 台のビデオカメラ (PointGreyResearch 社製 Dragonfly2) を対象物体に向けて撮影した映像から仮想視点映像を生成した。



図 11 実行環境

7.2 実験結果

図 12 は対応付けた特徴点から推定した 3 次元位置を示す。(a), (b), (c) の右図は、それぞれ、図 11 中の 3 台のカメラのうちの右、中央、左のカメラで撮影した映像中の一つのフレーム画像であり、(a) と (b) の画像中の青点は右と中央のカメラ画像間で対応付けた特徴点であり、(b) と (c) の画像中の緑点は中央と左のカメラ画像間で対応付けた特徴点である。そして、それぞれの左図は、推定した各特徴点の 3 次元位置を描画したものである。ほぼ妥当な 3 次元位置が得られている。

図 13 と 14 は、推定した特徴点の 3 次元位置に 5 節の方法で求めたビルボードを配置し、仮想視点に視線方向が一番近いカメラの映像を投影した結果である。図 13 と 14 とともに、2 つのカメラのフレーム画像 (a) と (b) に対して、それらのカメラ位置のほぼ中間に仮想視点を置いて生成した仮想視点画像 (c) をあらわす。ビルボードに投影する画像としてそれぞれ (a) と (b) を用いている。

表 1 は、カメラ映像と仮想視点映像の解像度が 320*240 と 640*480 の場合に対して、本手法の各処理の実行時間を示す。現段階では、リアルタイム表示のためには実行速度が不十分であり、改善が必要である。表 1 から、実行時間の合計に対して SURF 特徴点の抽出と対応付けが大きな割合を占めており、この処理の効率化が今後の課題である。

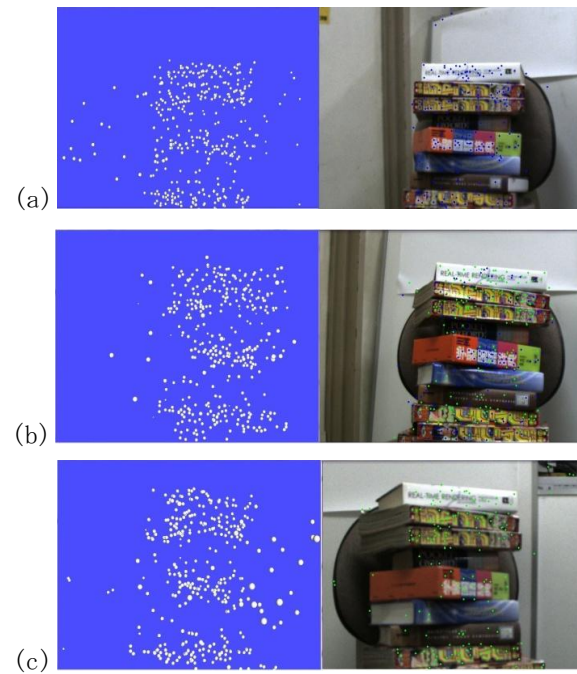


図 12 推定された特徴点の 3 次元位置



図 13 実験結果 1

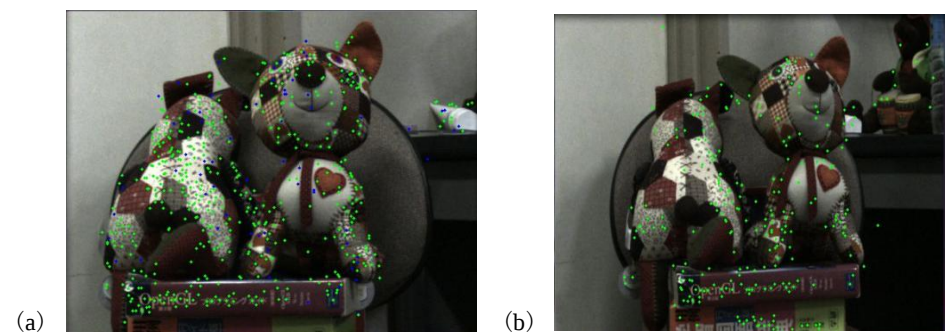


図 14 実験結果 2

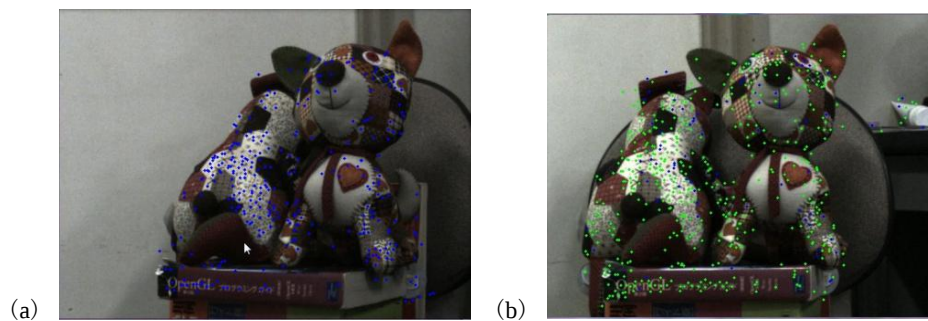


表 1 各処理の実行時間

	320×240 (s/f)	640×480 (s/f)
SURF 特徴点抽出	0.034	0.146
特徴点对応付け	0.065	0.270
3次元位置計算	0.004	0.004
ビルボード生成	0.014	0.057
描画時間	0.002	0.003
合計	0.119	0.484

8. おわりに

本研究では、仮想視点の位置に依存したビルボードを用いて対象物体の3次元形状を表現し、効率的に仮想視点映像を生成する方法を提案した。今後の課題としては、まず、リアルタイム表示の実現をめざし、実行時間のうちで大きな割合を占める SURF 特徴点の抽出と対応付けの高速化が挙げられる。また、背景差分等によりカメラ映像中の対象物体を抽出し、対象物体上の特徴点のみを得ることで、特徴点の対応付けや3次元位置の推定の精度を高め、処理速度を向上させることが挙げられる。さらに、マルチコアを用いた並列処理による高速化も検討している。

謝辞

本研究を実施するにあたり、様々な貴重な御意見を頂いた岩手大学工学部千葉則茂教授に感謝致します。本研究の一部は科学研究費補助金(基盤研究(C) 21500090)の援助を受けている。

参考文献

[1] G. Slabaugh, B. Culbertson, T. Malzbender, R. Shafer, A Survey of Methods for

Volumetric Scene Reconstruction from Photographs, Proceedings of the International Workshop on Volume Graphics 2001, pp.81-100, 2001.

[2] M. Z. Brown, D. Burschka, G. D. Hager, Advances in Computational Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.8, pp.993-1008, 2003.

[3] S. Yamazaki, R. Sagawa, H. Kawasaki, K. Ikeuchi, M. Sakauchi, Microfacet Billboarding, Proceedings of the 13th Eurographics Workshop on Rendering, pp.169-179, 2002.

[4] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding, 110(3), pp.346-359, 2008.

[5] 3日で作る高速特定物体認識
<http://d.hatena.ne.jp/aidiary/20091018/1255862734>

[6] S. Hu, J. Qiao, A. Zhang, Q. Huang, 3D Reconstruction from Image Sequence Taken with a Handheld Camera, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol.XXXVII, Part B4, pp.559