

物体表面の見えの変化に対応した生成型学習による 3次元物体トラッキング

篠塚 祐紀子^{*1}

ディサポン タチャソントム^{*2}

斎藤 英雄^{*3}

Abstract – 本稿では、生成型学習による3次元物体トラッキングの手法を提案する。SIFTやSURFなどの局所特徴量はスケール変化・回転に対しては頑健であるが、アフィン変換に対しては弱い。そのため、複合現実のためのトラッキングをするためには工夫が必要である。提案手法では、あらかじめ与えられた物体のテクスチャ付き3次元モデルを元に視点を変化させて撮影したようなパターン画像を事前に生成し、それら画像から特徴点を抽出して学習する。それらをデータベースに蓄積して、トラッキング時に参照することにより、視点変化に頑健な3次元物体のトラッキングが可能となった。

Keywords : 3次元物体, トラッキング, 生成型学習, 視点変化, 局所特徴量

1 まえがき

近年、実世界に仮想情報を重畳してみせる拡張現実感の研究が盛んである。この研究では、カメラの位置や姿勢の変化によらず、画像中の対象物体を認識する技術が重要である。

従来手法では、あらかじめ登録された画像パターンと入力画像の間で正確な点対応を得ることで、物体認識、拡張現実感を実現する。このとき、画像から局所特徴を抽出することで位置や姿勢変化への頑健性を獲得している。

本研究では、生成型学習を行うことで、大きな視点変化への頑健性のある3次元物体トラッキング手法を提案する。様々な視点から撮影されたかのようなパターン画像群を生成するこの手法は、より多くのパターンから繰り返し検出される特徴点を抽出して点対応が取れる可能性を広げ、様々なパターンで記述される特徴量を収集してデータベース化することにより正しいマッチングが可能になる。その結果、3次元物体のトラッキングをより安定、正確に行うことが可能となる。

2 関連研究

本章では、関連研究である画像の局所特徴量とトラッキングについて述べる。

局所特徴量であるSIFT特徴量[1]は、スケール変化、回転に頑健である。しかし、アフィン変換への頑健性は乏しくトラッキングのためのマッチングをするには適さない。局所特徴検出器であるMaximally-Stable Extremal Region Detector(MSER)[2], Harris-

Affine[3], Hessian-Affine[3]は、アフィン変換に強い検出器として知られているが、検出器のみが提案され、それぞれに対するアフィン変換に対する普遍性を備えた記述子は存在しない。頑健なマッチングを行うためには、アフィン変換に特徴記述子について別途検討する必要がある。

事前学習を用いてアフィン変換に頑健にして手法として、Randomized Tree[4], Random Ferns[5], Phony Ferns[6]が挙げられる。これら手法では、提案手法と同じく事前に視点を変えて撮影したかのようなパターン画像を生成し、それら画像から特徴点を抽出している。その後決定木アルゴリズムであるRandomized Treesを用いてクラスタリングすることにより、高速なマッチングを実現している。これらの手法は、提案手法と異なる点が2つある。

1) 特徴点同士のマッチングのためには、2枚の画像上で同じ特徴点が抽出されなければならない。Randomized Trees等では、正面画像で検出された特徴点を元に、生成した画像上でも検出されるかを検証している。しかし、この手法を用いると正面画像上からは検出されないが、他視点からは検出される特徴点についてデータベースに蓄積することはできない。提案手法では、そのような特徴点も含め蓄積することにより、視点の変化に頑健なトラッキングを実現している。

2) Randomized Treesによる学習では、決定木の数、木の高さ、ランダム性のパラメータなど様々なパラメータを定める必要がある。そのため、パラメータ設定が困難である。一方、提案手法では学習において点の抽出回数のしきい値の設定のみを行えばよい。

トラッキングでは、視点変化に対して頑健なカメラの位置や姿勢を推定する必要がある。Wagnerら[6]は、検出した特徴点周りに生成した画像パッチを追跡する

^{*1}慶應義塾大学

^{*2}慶應義塾大学

^{*3}慶應義塾大学

手法を提案した。連続するフレーム間では、画像パッチの見え方が大きく変わらないことを前提にし、画像パッチの変形を線形に予測して、次のフレームでの対応を頑健に行っている。Ito ら [7] は、平面をトラッキングの手法において、対象平面がカメラの視線方向に対して大きく傾いたり遠くに離れる際に生じる平面の解像度低下が、追跡性能に悪影響を与えることを指摘した。そこで画像の標本化過程をモデル化し、追跡アルゴリズムにこれを組み込むことで頑健なトラッキングを実現した。しかし、連続するフレーム間の情報を用いるこれらの手法では、対象を正確に追跡している場合には有効であるが、一度追跡に失敗してしまった時は、カメラ位置を初期状態に戻してトラッキングしなおさなければならない。一方、提案手法では、これら2つの手法と異なり、毎フレームごとにデータベースと照合を行なっているため、動画が入力であっても再初期化の手間がかからないという利点がある。

本提案手法では、対象を3次元物体とし、視点生成型学習をもちいてトラッキングを行なった。

3 システム構成

本章では、3次元物体トラッキングのための流れを説明する。図1にトラッキングの流れを示す。本手法は、学習とトラッキングの2段階に分かれる。

学習では、対象モデル画像群を入力とし、三次元モデルを作成後、そのモデルをもとにOpenGL[9]を用いて、物体を回転させ疑似的に視点を変更して撮影したかのようなパターン画像を作成する。その画像から局所特徴量を抽出し、Stable Keypoint と呼ばれる複数パターン画像から抽出された同一点を探し出し、データベース化する。

トラッキングでは、任意の入力画像から局所特徴量を抽出し、データベースを参照して特徴点マッチングを行うことでカメラの位置姿勢を推定する。

第4章では、学習の手法である視点生成型学習について、第5章では、マッチングと位置姿勢推定について具体的に述べる。

4 視点生成型学習

局所特徴量の計算では拡大縮小や回転に対して不変性をもつようなアルゴリズムが組み込まれているが、視点が大きく変化すると記述される特徴量が変化するため、特徴点同士の対応が取れなくなる。そこで、様々な視点における特徴量を収集することで、拡大縮小や回転に加え、視点変化への頑健性を達成する。ここで提案手法の学習には村瀬が提唱した生成型学習 [8] を利用する。生成型学習による画像認識は、入力パターンの多様な変動や変形に対処するための手法として、学習パ

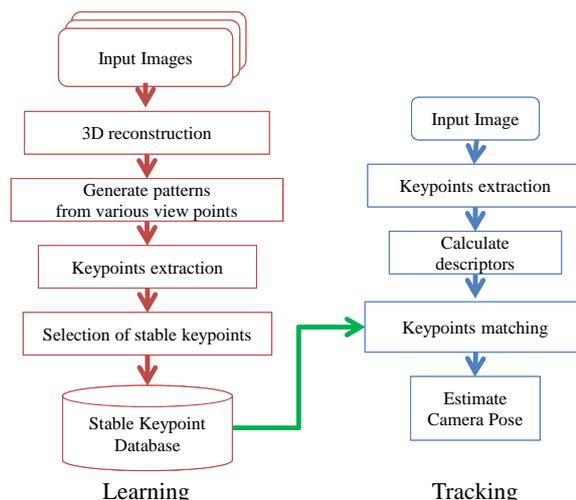


図1 トラッキングの流れ
Fig. 1 Overview of our Method

ターンを変動や変形により様々に生成し、これと入力パターンとを照合する。実際のシーンをサンプルとして収集し用いるのではなく、実際に起こりうるようなシーンを仮想的に生成し用いる学習法であり、収集できるパターンが少ない場合に特に有効といえる。よって視点生成型学習では、対象となる3次元物体のモデルを入力に用いて、様々に角度を変えて撮影したかのようなパターン画像群を生成して局所特徴を抽出、収集した後にデータベース化する。学習を終えると、正面画像と実際に撮影されたカメラからの入力画像はそれぞれ、データベース中の特徴量と抽出された特徴量を比較することで点対応を得る。

4.1 三次元モデル生成

トラッキングしたい対象物体の全周を撮影した画像を入力とし、Autodesk 123D Catch[10]を用いて三次元復元する。このモデルは、テクスチャ付の三次元モデルである。

4.2 視点変化によるパターン画像の生成

三次元モデルをもとに、仮想的に様々な視点から見たかのようなパターン画像群を生成する。このパターン画像は、カメラの外部パラメータを変化させることにより作成する。主にパラメータの回転行列 R は、パラメータ θ, ϕ, ψ によって定められる。(式(1))のうち、角度 ϕ と θ は、それぞれ半球を移動するカメラ位置を決定する緯度と経度に相当し、角度 ψ は、カメラの後軸周りの回転角である。本実験で用いる局所特徴は回転不変性を備えているため、光軸周りのパラメータ ψ は固定する。並進ベクトルはカメラと対象物体との距離を d として $t = (0, 0, d)^T$ として表されるが、同様に本実験で用いる局所特徴は拡大縮小への不変性

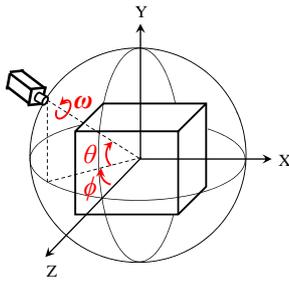


図2 三次元モデルに対するカメラの位置姿勢
 Fig.2 A model for pattern generation of various viewpoints

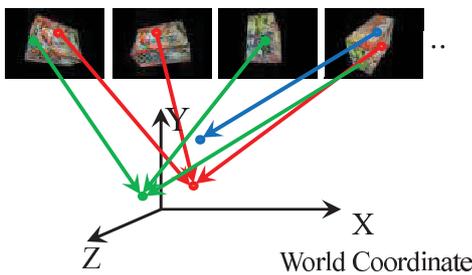


図3 三次元空間への再投影と Stable Keypoint
 Fig.3 Reprojection to 3D World Coordinate and Stable Keypoint

も備えているため固定値を用いる。

$$\mathbf{R} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (1)$$

4.3 Stable Keypoint データベースの作成

次に生成されたパターン画像群から局所特徴の検出器により特徴点を検出する。抽出された特徴点は射影行列変換に基づき三次元座標空間上へと再投影する。様々なパターン画像から同じ三次元位置に投影される特徴点は、見えの変化に頑健な、高い可検出性を持つ特徴点 (Stable Keypoint) であると考えられる。パターン画像が、射影行列変換 P_i で生成されるとすれば、パターン画像上で検出された特徴点 p_i は、三次元空間上では式 (2) によって投影点 p'_i に投影される。

$$p'_i = H^{-1}p_i \quad (2)$$

もし、投影点 p'_i と最近傍点とのユークリッド距離がしきい値以下であれば、2枚のパターン画像から同じ三次元位置にある特徴点が検出されたものとみなし、その特徴点に対して投票を行う。投票を行なった結果、投票数順に順位付けし、その中から上位 1200 のみを stable keypoint として抽出する。

5 マッチングと位置姿勢推定

本章では、任意の入力画像のカメラ位置姿勢推定の手法について述べる。まず、入力画像から特徴点を検出し、特徴量を記述する。次に入力画像から得た特徴量をデータベースと比較し、最もユークリッド距離が近い組を対応させる。そのとき、Mikolajczyk ら [11] の最近傍比を用いて誤対応を除去する手法を用い、2番目にユークリッド距離が近い特徴量と比べて、最近傍の特徴量がより接近している場合を正対応とする。抽出された特徴量 D_A に対し、最近傍の特徴量を D_B 、2番目に近い特徴量を D_C とする。以下の式 (3) を満たす時、正対応のマッチングとみなす。このとき、しきい値 t は経験的に 0.6 と定めた。

$$\frac{\|D_A - D_B\|}{\|D_A - D_C\|} < t \quad (3)$$

式 (3) にて誤対応除去をし、データベース中の Stable Keypoint とのマッチングを求めた後、RANSAC アルゴリズムを用いて射影行列を求め、カメラの位置姿勢を推定する。

6 評価実験

本章では提案手法の性能を評価するために行なった実験について述べる。Randomized Tree[4] を用いたトラッキングを比較手法とする。

6.1 カメラの外部パラメータによる評価

本節では、トラッキングにより求められたカメラの外部パラメータ (回転行列 R ・平行移動行列 T) について定量的な評価を行う。実験に用いる 3次元物体として、図4に示す **Head**, **Stadium**, **Ramen** を用いた。表1に、特徴点マッチングにより回転、平行移動行列が算出された成功率、回転行列の誤差平均、平行移動行列の誤差平均を示す。特徴点マッチングにより回転、平行移動行列が算出された成功率は、行列計算できたフレーム数を総フレーム数で割った値である。各行列の誤差平均は、誤差行列のユークリッドノルムの平均を求めることにより算出した。表1より、提案手法は Radomized Trees を用いた手法に比べ誤差が小さいことがわかった。

6.2 結果画像

本節では、AR のモデルのトラッキング結果画像を示す。実験に用いる 3次元物体として、**Head** を用いた。図5に示す。この図のように、実環境においてもトラッキングが行われていることがわかる。

6.3 鏡面反射のある物体への応用とその課題

本節では、鏡面反射のある物体に対して提案手法を用いてトラッキングを行い、提案手法の限界と課題を述べる。鏡面反射のある物体として、車を用いた。動

表 1 従来手法と提案手法との比較

Table 1 Evaluation values of rotation and translation compared with randomized tree method[4]

object	method	success rate [%]	diff of rotation	diff of translation[cm]
Head	Randomized Tree[4]	79	2.40	27.53
Head	Proposed Method	100	2.39	2.17
Stadium	Randomized Tree[4]	88	2.41	37.24
Stadium	Proposed Method	60	2.30	8.43
Ramen	Randomized Tree[4]	89	2.40	35.56
Ramen	Proposed Method	62	2.39	10.60



図 4 トラッキング対象物体
Fig. 4 3D Model for the Experiments

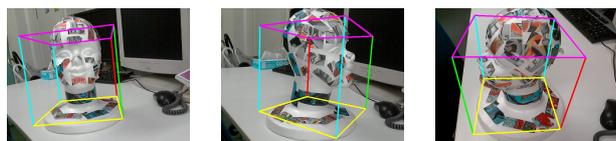


図 5 実環境におけるトラッキング結果画像
Fig. 5 Tracking the Object in the Real World

画全 1496 フレーム中、735 フレーム (49.1%) においてトラッキングのための特徴点マッチングが行えず、トラッキングができなかった。図 6 には、特徴点マッチングが成功しトラッキングとして出力されたものを示す。特徴点マッチングに誤対応が多かったため、カメラの外部パラメータが正しく算出されず、車のトラッキングができなかった。

これら結果より、本手法は鏡面反射のある物体に対して有効でないことがわかった。本提案手法では視点生成型学習においてトラッキング対象物体が完全拡散面であると仮定してパターン画像群を生成している。しかし、本節の実験で用いた車のように鏡面反射のある物体は、視点方向により鏡面反射箇所が変化し見えが異なるため、特徴点マッチングが正しく行われず、物体の位置姿勢推定精度が悪くなったと考えられる。

7 まとめ

本論文では、視点変化に対して頑健な三次元物体の位置姿勢推定法を提案した。様々な視点から撮影したかのようなパターン画像を生成し、繰り返し検出され

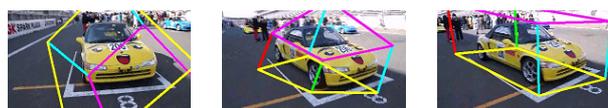


図 6 車のトラッキング結果画像
Fig. 6 Car Model Tracking

た特徴量をデータベースとすることで、アフィン変換に対して頑健な特徴点マッチングを実現することができた。今後の課題として、鏡面反射のある物体に対して視点生成型学習を用いてトラッキングに取り組んでいく。

参考文献

- [1] Lowe, D. G : Distinctive image features from scale invariant keypoints; International Journal of Computer Vision, 60, 91—110. (2004).
- [2] J.Matas, O.Chum, M.Urban and T.Pajdal : Robust wide baseline stereo from maximally stable extremal regions; BMVC, pp.384-393 (2002)
- [3] Mikolajczyk, K. and Schmid, C : Scale & affine invariant interest point detectors. IJCV, 60:63—86. (2004).
- [4] Lepetit, V. and Fua, P. : Keypoint recognition using randomized trees; IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9):1465—1479. (2006)
- [5] Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P. : Fast keypoint recognition using random ferns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(3):448—461.(2009).
- [6] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond and Dieter Schmalstieg : Real-time detection and tracking for augmented reality on mobile phones ; TVCG, Vol.16, 355-368, (2010)
- [7] E.Ito, T.Okatani, and K.Deguchi :Accurate and robust planar tracking based on a model of image sampling and reconstruction process ; ISMAR, (2011)
- [8] 村瀬洋, 画像認識のための生成型学習, 情報処理学会誌 (CVIM),2005.
- [9] OpenGL : <http://www.opengl.org/>
- [10] Autodesk, 123D Catch, <http://www.123dapp.com/catch>
- [11] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. IJCV, 65:43—72.