Vol.2015-CVIM-195 No.65
2015/1/23

異なる光源環境における画像特徴の頑健性の調査

工藤 彰[1,a)]　Alexander Plopski[1]　Tobias Höllerer[3]　間下 以大[1,2]　清川 清[1,2]　竹村 治雄[1,2]

概要：拡張現実感 (AR) でバーチャル空間と実空間の幾何学的整合性を得る方法の一つに，事前に作成した現実環境の特徴点データベースとシーン中の特徴点のマッチングを行うことでカメラの自己位置推定を行う方法がある．しかし，光源環境が変化した場合，抽出される特徴点とデータベースの対応がとれず，カメラの自己位置推定の精度が低下する場合がある．この問題に対し本研究では，バーチャル空間におけるシミュレーション実験を通して，異なる光源環境における画像特徴の頑健性を調査した．その結果，光源環境が同じ状況で構築された特徴点データベースを参照する場合に自己位置推定の誤差が小さくなることが確認できた．

1. はじめに

拡張現実感（AR）技術を利用することで，カメラ画面上に実空間と幾何学的整合性のとれた仮想的なグラフィックを描画する研究が幅広く進められている．描画物体の位置合わせの精度を高めるためには，カメラの位置・姿勢を正確に推定する必要がある．そこで，画像特徴点の３次元位置座標を記録したデータベースをあらかじめ作成し，実環境から抽出される特徴点とマッチングを行うことで自己位置推定を行う方法が注目を集めている [5]．しかし，実環境における光源環境は時々刻々変化しているため，データベースとは異なる画像特徴点が抽出される可能性が考えられる．屋外を例に挙げると，時間帯による太陽の位置の変化や天気により景観が変化した場合，データベースに記録されている画像特徴点とのマッチングが難しくなることが予想される．一方，異なる光源環境に対する複数の特徴点データベースをあらかじめ用意し，適切なデータベースを動的に切り替えて参照することができれば，自己位置推定の精度向上が期待できる．このとき，撮影した画像から光源環境の推定する技術が必要となる [4]．本研究では，バーチャル空間におけるシミュレーションにより光源の変化に対する画像特徴点の頑健性を調査し，光源環境の異なる環境下で作成した異なる特徴点データベースを用いたカメラの自己位置推定精度の比較とその評価について述べる．

[1] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[2] 大阪大学，サイバーメディアセンター
Cybermedia Center, Osaka University
[3] University of California, Santa Barbara, Department of Computer Science
[a)] kudo.akira@lab.ime.cmc.osaka-u.ac.jp

2. 特徴記述子の評価

一般に，入力となる撮影画像のカメラ位置・姿勢が全く同じ状態で抽出される画像特徴点がデータベースに登録されているわけではない．故に，画像の拡大縮小および回転に対して耐性のある画像特徴点を使用することが望ましい．本研究においては二種類の画像特徴量，SIFT（Scale-Invariant Feature Transform）および SIFT を高速化した手法である SURF（Speeded-Up Robust Features）に関して評価を行う．

SIFT は DoG（Difference Of Gaussian）の使用により画像の拡大縮小に不変な特徴量を抽出し，特徴点周りの輝度変化から向きを定義することで回転に対しても不変な特徴量を抽出することができる．さらに，特徴点周りの 16 (4×4) 分割ブロックごとに 8 方向の輝度変化を求めた 128 次元のベクトルを定義することで，光源環境の変化に対しても比較的頑健となる．

カメラの正確な自己位置推定が求められる場面では，データベースとの誤対応は少ないほうが好ましい．光源環境が変化した場合，データベース記録時には影の影響で検出されていなかった特徴点が数多く検出されるようになり，特徴点の対応付けに失敗する回数が増加する．このような問題の発生と誤差の程度は光源環境と対象物体の組み合わせによって変化すると考えられる．そこで，本研究ではバーチャル空間にて異なる光源環境下で作成した複数の特徴点データベースを元に，光源環境の異なる状況で撮影した多数の画像に関してカメラの自己位置推定のシミュレーションを行い，その精度を比較する．シミュレーション実験は大きく次の２段階のアプローチで進める．

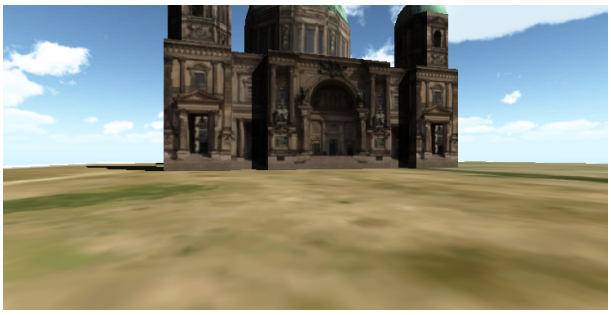

図 1 City of Sights モデルをインポートした Unity のシーンの一例

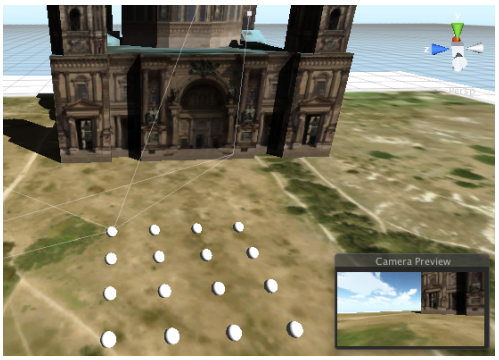

図 2 16 ヶ所のカメラ位置

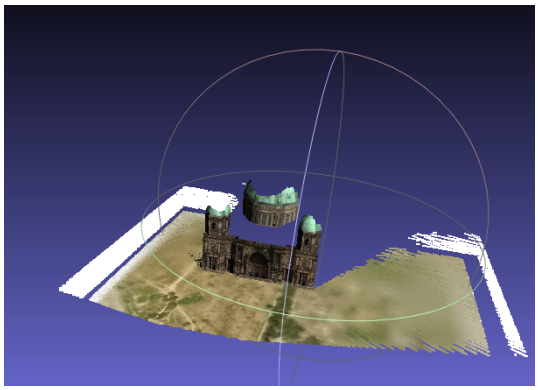

図 3 抽出したモデルファイル

(1) 各光源環境に関して代表的な特徴点に関するデータベースの作成
(2) 作成した複数のデータベースを用いて，光源環境の異なる環境でのカメラ自己位置推定の精度評価

3. シミュレーション実験の方針

シミュレーション実験の方針についての説明を行う．バーチャル空間における実験には，光源や影のパラメータ設定が豊富で容易に設定することができるため，ゲームエンジン Unity[*1]を利用する．3次元モデルファイルには City of Sights[2] のモデルを利用した（図 1）．City of Sights のウェブサイト[*2]では，バーチャル空間における 3D モデルファイル（.skp .fbx）に加えて，同一のモデルを実環境でも簡単に作成することができるペーパークラフト用素材（.pdo）も配布されている．

本実験では，同一の 3 次元モデルに関して光源環境の異なる状況下でそれぞれの特徴点データベースを作成する．光源としては，空間内にある全ての物体に対して同一方向に光があたる平行光源を使用し，光の進行方向のみを変化させる．各光源環境に関して，カメラの位置 16 種（図 2 において白球が描画されている位置），各位置に関して姿勢 5 種の合計 80 種類のシーンを取得し，それぞれのシーンに関して各ピクセルの Unity 空間における 3 次元位置座標をファイル出力する．図 2 のモデル内の建物の幅が 40m である環境において，カメラの高さは常に 1.7m とする．カメラの姿勢は，建物に対して垂直となる方向およびその角度から水平方向に ±15°，±30° となる 5 種類を記録する．続いて，全シーン画像の中から代表となる画像特徴点を選出し，特徴点データベースを作成する．本実験では，SIFT 記述子および SURF 記述子の 2 種類の特徴記述子に関してそれぞれシミュレーション評価を行う．平行光源の光の進行方向を変更し，同様の手順で複数の特徴点データベースを作成する．生成したデータベースを使用して，各光源環境において複数のシーンを入力としたカメラの自己位置推定テストを行う．

*1 http://japan.unity3d.com/
*2 http://cityofsights.icg.tugraz.at/

3.1 各ピクセルの 3 次元座標の取得

Unity 内のモデルの表面に衝突判定を行う Collider を設定し，シーンの各ピクセルに関して Physics.Raycast を適用することで，全ピクセルの 3 次元位置座標を取得した．このようにして図 2 の 16 ヶ所 5 方向から取得した各ピクセルの 3 次元位置情報と RGB 情報を合成し，ply 形式の 3 D モデルファイルとして出力したものが図 3 である．

3.2 代表的な特徴点の抽出

取得した 80 枚のシーンと各ピクセルの 3 次元位置情報データを元に，代表的な特徴点の選択方法について説明する．代表的な特徴点集合 $D' \subset D$ は，異なる画像の間で正確なマッチングが行われた回数が多い特徴点の集合である．アルゴリズムは論文 [3] を参考にし，2000 個の代表的な特徴点を選択してデータベースとしてバイナリファイルに記録した．代表的な特徴点の選択アルゴリズムは次の流れに従う．

(1) 全シーンについて特徴点の抽出
(2) 総当りで全シーンの特徴点のマッチング
(3) 誤対応の除去
(4) 各マッチング情報を一つの行列 M に保存
(5) マッチングされた回数が最大の特徴点 $d_{i_{max}}$ をデータベースに追加
(6) $d_{i_{max}}$ とのマッチング情報を M から除去
(7) (5) (6) を 2000 回繰り返す

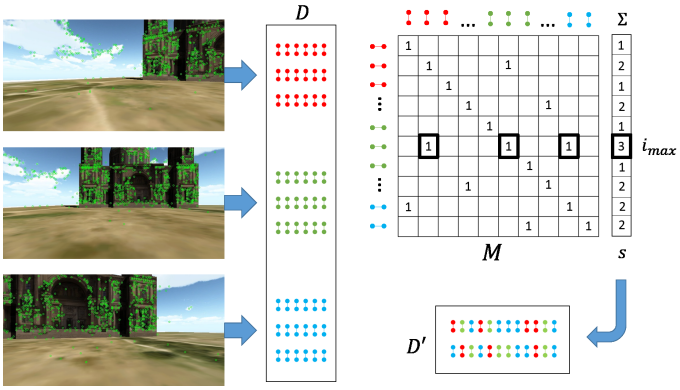

図 4 代表的な特徴点の選出の流れ

はじめに，全てのシーン画像に関して画像特徴点を抽出し，それぞれの特徴量を保存する（1）．続いて，一枚のシーン画像の特徴点 $d_i \in D$ と各シーンの特徴点の集合 D_j でマッチングを行う（2）．同様に各シーンに関してマッチングを行っていき，すべての特徴点のマッチングの終了後，正しくマッチングできなかった組み合わせを除外する．具体的には，マッチングが行われた 2 つの点の 3 次元座標を比較し，距離が 0.5 以上のものをすべて除外する（3）．続いて，残された正確なマッチングに関して図 4 のような $n \times n$ 行列 M を作成する（$n = |D|$ は特徴点の総数を表す）．行列の要素 $M_{i,j}$ は i 番目の特徴点が j 番目の特徴点とマッチングされていた場合 1，そうでなければ 0 とする．s_i は i 行の要素の合計値，つまり d_i とマッチングが行われた点の数を表すとする（4）．最もマッチング回数が多かった特徴点の指数 $i_{max} = argmax_i(s_i)$ を取得し，代表的な特徴点の集合に追加する（$D' = D' \cup d_{i_{max}}$）(5)．以後，同じ特徴点が選択されないように i_{max} 行の要素に 0 を代入し，この時 0 以外の要素を持っていた各列 j に関してもすべての要素に 0 を代入する．この操作により，すでに D' に含まれている特徴点 $d_{i_{max}}$，および $d_{i_{max}}$ とマッチングされていた特徴点が以後選択されないようにする（6）．操作完了後 s を更新して必要な特徴点の数 $|D'| = 2000$ に到達するまで同様の操作を繰り返す（7）．

このようにして選択された 2000 個の代表的な特徴点に関して，特徴量，3 次元位置座標，およびオリエンテーションをバイナリ形式で出力した．

3.3 カメラ自己位置推定

画像上で観測した特徴点の座標と，マッチングが行われたデータベース内の特徴点の 3 次元座標の対応からカメラの位置の推定を行う．この問題は一般に PnP 問題として知られ，OpenCV ではこれら 3 次元- 2 次元の点の対応とカメラ内部パラメータ行列および歪係数を入力に PnP 問題を解く関数が用意されている．特に，今回は特徴点の誤対応を排除する必要があるため，関数 cv::SolvePnPRansac を用いてカメラの外部パラメータ（回転ベクトル，並進ベクトル）をロバスト推定手法である RANSAC[1] を用いて導出した．

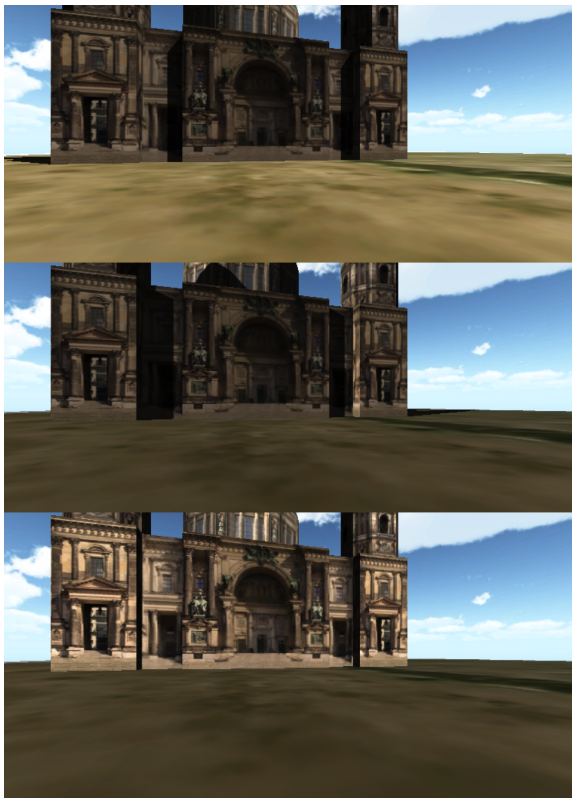

図 5 3 種類の光源環境（上から環境 1，環境 2，環境 3）

4. 特徴記述子の評価・考察

Unity 上で同一の 3 次元モデルに対して，3 種類の異なる光源環境を構築した（図 5）．光源の種類は平行光源を選択し，色・強度は一定で光の進行方向のみを変化させる．3 つの光源環境をそれぞれ環境 1，環境 2，環境 3 とし，各々の環境下で SIFT 記述子を利用して構築されたデータベースをそれぞれ $D_{SIFT}1$，$D_{SIFT}2$，$D_{SIFT}3$ とし，SURF 記述子を利用して構築されたデータベースをそれぞれ $D_{SURF}1$，$D_{SURF}2$，$D_{SURF}3$ とする．環境 1 と環境 2 では建物に向かって左右反対方向から光を照射し，環境 3 は建物の正面方向から光を当て，可能な限り影を少なくした．

はじめに，各光源環境において撮影された 3 種類のシーン画像（図 5）に対して，$D_{SIFT}1$，$D_{SIFT}2$，$D_{SIFT}3$ を用いて特徴点のマッチングを行い，結果を画像に出力した．画像上の点 (X, Y) とマッチングしたデータベース上の点 (x, y, z) が同一の点であるか調べるために，(x, y, z) を画像平面に投影した (X', Y') を導出した．(X, Y) と (X', Y') の距離が 20 ピクセル未満である場合は緑色の点，20 ピクセル以上離れている場合は対応誤りと見なして赤色の点を描画した．図 6，図 7，図 8 はそれぞれ $D_{SIFT}1$，$D_{SIFT}2$，$D_{SIFT}3$ を使用した場合の出力結果である．これらの結果

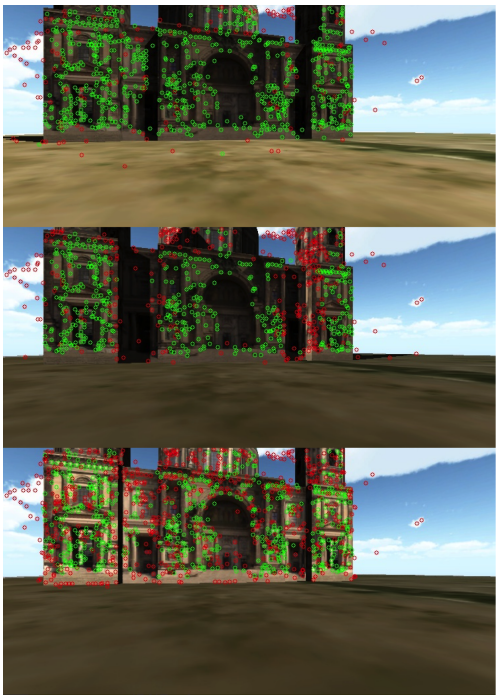

図 6 $D_{SIFT}1$ の使用

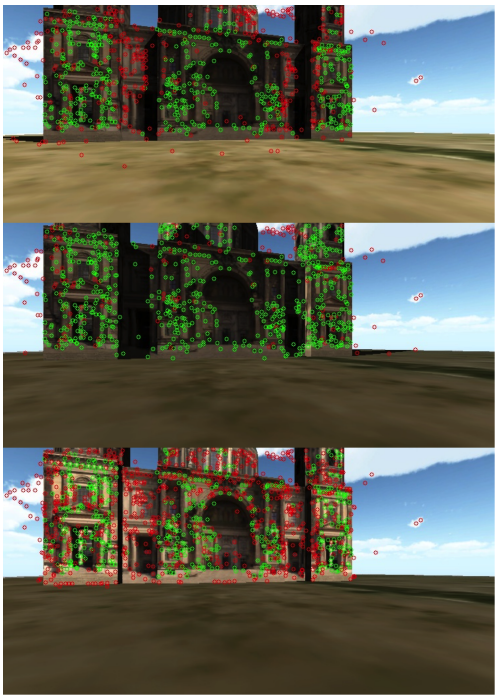

図 7 $D_{SIFT}2$ の使用

から，環境内の物体位置的には全く同じ環境であっても，光源環境が変化することで抽出される画像特徴点に変化が現れることが確認できた．$D_{SIFT}1$ を利用した図 6 の環境 1 と環境 2 の結果を比較すると，環境 1 において影となっている部分が環境 2 において多くのマッチング誤りを得ていることがわかる．これらの出力結果ではいずれも，光源環境の一致しているデータベースを使用した場合に，より多くの正確なマッチング（緑色の点）結果が得られている．

次に，各環境において撮影された 80 枚の画像のそれぞれにおいてカメラの 3 次元位置座標を推定し，真値との誤差を計算した．建物の横幅が 40m であるモデル環境において，真値との座標の誤差が 0.5m 未満であった場合に自己位置推定を成功とみなす．SIFT 記述子のデータベースを利用した場合に 80 枚全てのシーンに関しての実行結果の平均値をまとめたものが表 1 であり，同様に SURF 記述子のデータベースを利用した場合の結果を表 4 に示す．表 1 ではどのデータベースを利用した場合も，光源環境が同じ状況において自己位置推定の成功率が高くなっているのがわかる．SIFT，SURF いずれの記述子を用いた場合も，環境 3 に対して環境の異なるデータベースを使用した場合に自己位置推定の成功率が低くなっている．この原因としては，環境 3 は最も鮮明で影が少ないため，他の環境と比べて多くの特徴点が検出されていることが挙げられる．環境 3 において，光源環境の異なるデータベースを使用した場合，データベースに存在しない点が多く検出されるため自己位置推定結果の誤差が大きくなると考えられる．

自己位置推定が成功した場合の真値との誤差の平均値をそれぞれ算出した結果を表 3，表 4 に示す．この結果から，

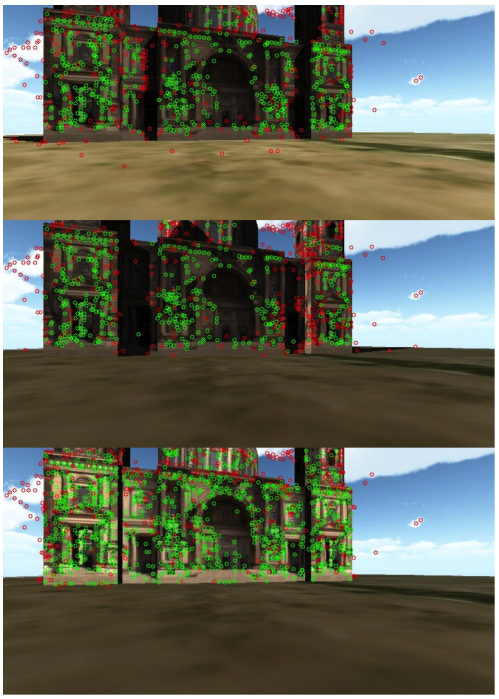

図 8 $D_{SIFT}3$ の使用

表 1 自己位置推定の成功率（SIFT）

	環境 1	環境 2	環境 3
$D_{SIFT}1$	91.25	90.00	63.75
$D_{SIFT}2$	71.25	85.00	48.75
$D_{SIFT}3$	86.25	81.25	87.75

SIFT 記述子を利用した場合の方が SURF 記述子を利用した場合と比べて全体的に誤差が小さくなっていることがわかる．

表 2 自己位置推定の成功率（SURF）

	環境 1	環境 2	環境 3
$D_{SURF}1$	90.0	78.75	51.25
$D_{SURF}2$	66.25	78.75	30.0
$D_{SURF}3$	85.00	77.5	77.5

表 3 自己位置推定成功時の誤差の平均値（SIFT）

	環境 1	環境 2	環境 3
$D_{SIFT}1$	0.132714	0.153362	0.12094
$D_{SIFT}2$	0.164058	0.183394	0.152777
$D_{SIFT}3$	0.114435	0.149724	0.140405

表 4 自己位置推定成功時の誤差の平均値（SURF）

	環境 1	環境 2	環境 3
$D_{SURF}1$	0.217	0.1929	0.229486
$D_{SURF}2$	0.192249	0.173924	0.19482
$D_{SURF}3$	0.195431	0.161075	0.175

5. まとめ

本研究では，バーチャル空間において光源環境を変化させるシミュレーションにより特徴点の頑健性を調査した．同じ光源環境においては，カメラの位置・姿勢の変化に対して比較的頑健な特徴点を選択して記録したデータベースを使用したが，光源環境の変化に対しては頑健性が小さくなることが確認できた．また，カメラの自己位置推定を行う際には，同一の光源環境において構築されたデータベースを参照した場合に，より精度の高い結果が得られることが確認できた．このことから，幾何学的整合性のとれた拡張現実感を実現させるためには，光源環境の変化に対する頑健性を持つデータベースを参照することが必要と言える．今後の最初の課題は，実環境において同様の実験を行い，画像特徴の頑健性を調査することである．さらには，光源の位置だけでなく環境光の考慮や，光の色，光度の変化が画像特徴点にもたらす影響について調べ，自己位置推定の精度を高める手法を調査することである．

参考文献

[1] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Vol. 24, No. 6, pp. 381–395, June 1981.

[2] Lukas Gruber, Steffen Gauglitz, Jonathan Ventura, Stefanie Zollmann, Manuel Huber, Michael Schlegel, Gudrun Klinker, Dieter Schmalstieg, and Tobias Hollerer. The City of Sights: Design, construction, and measurement of an Augmented Reality stage set. *2010 IEEE International Symposium on Mixed and Augmented Reality*, pp. 157–163, October 2010.

[3] Daniel Kurz, Thomas Olszamowski, and Selim Benhimane. Representative feature descriptor sets for robust handheld camera localization. *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 65–70, November 2012.

[4] Jean-François Lalonde, Alexei a. Efros, and Srinivasa G. Narasimhan. Estimating the Natural Illumination Conditions from a Single Outdoor Image. *International Journal of Computer Vision*, Vol. 98, No. 2, pp. 123–145, October 2011.

[5] Jonathan Ventura and Tobias Hollerer. Wide-area scene mapping for mobile visual tracking. *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 3–12, November 2012.