

光源環境の変化にロバストな自己位置推定のための画像特徴データベースの構築

工藤 彰^{1,a)} Alexander Plopski¹ Tobias Höllerer³ 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位置合わせをする方法の一つに、事前に作成した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、実際の環境、特に屋外環境では頻りに光源環境が変化するため、抽出される特徴点とデータベースの特徴点との対応付けに失敗し、カメラの自己位置推定の精度が低下する可能性がある。そこで、現実環境から構築した CG モデルを用いて光源環境のシミュレーションを行い、各環境下で特徴点を抽出する。これらの特徴点集合の中から代表点を選択し、マハラノビス距離を用いて特徴点のマッチングを行うことで、全てのデータを参照することなく特徴点の対応付けを行う手法を提案する。さらに、特徴点同士のマッチングを直接行う手法と提案手法について、シミュレーション実験により自己位置推定精度の比較を行った。結果、提案手法では処理時間が大きくなるが、少ない特徴点数で精度の高い自己位置推定を行うことができることを確認した。

1. はじめに

拡張現実感 (Augmented Reality, 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、図1に示すように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、光源の変化がデータベース内の特徴点に与える影響を考慮した対応付けを行うことができれば、自己位置推定の精度向上が期待できる。本報告では、異なる光源環境における



図1 現実環境の様々な光源環境 [2]

各特徴点の分散を考慮したマハラノビス距離の考えを導入したマッチング手法を提案する。異なる光源環境での特徴量の変化は、バーチャル空間内のシミュレーションによって再現する。本研究では、光源の変化に対する画像特徴点の頑健性を調査した上で、従来手法と比較を行うことで提案手法の有効性を確認する。

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

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

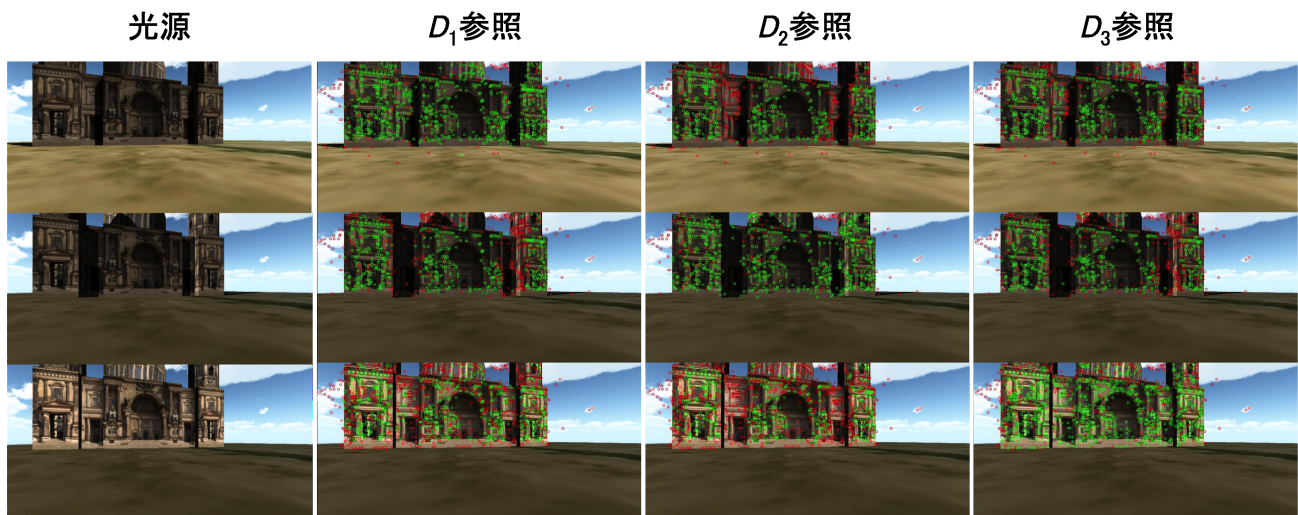


図 2 3種類の光源環境内の特徴点を用いて構築した3つのデータベース DB_1 , DB_2 , DB_3 を参照した際の、各3環境で撮影した画像を対象に特徴点のマッチングの正誤を調べた例である。緑色の点は正しくマッチングされた特徴点、赤色の点は誤ってマッチングされた特徴点を表す。

い。本研究においては画像特徴量 SIFT (Scale-Invariant Feature Transform) 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 [4]。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、データベースの特徴点とシーン上の特徴点との誤対応は少ないことが好ましい。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 [6]。図 2 は 3 種類の光源環境 (上から環境 1, 環境 2, 環境 3 とする) の下で構築した 3 つのデータベース DB_1 , DB_2 , DB_3 をそれぞれ参照して、3 枚の対象画像に対して特徴点の対応付けを行った結果を示している。図から明らかなように、対象画像の光源環境がデータベース生成時の環境と一致している場合と比べて、異なる光源環境を対象にする場合は多くマッチング誤りが発生している。また、表 1 は図 2 と同様の 3 種類の環境下の様々な位置・角度で撮影した 80 枚の画像に対しての自己位置推定の成功率をまとめたものである。1 枚ごとの入力画像に対して自己位置推定を行い、実際の撮影時のカメラ位置 (真値) との誤差を調べた。誤差が小さかった場合に自己位置推定を成功とみなし、80 枚の画像に対する自己位置推定の成功率を示している。この結果からわかるように、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

表 1 自己位置推定の成功率

	環境 1	環境 2	環境 3
D_1	91.25	90.00	63.75
D_2	71.25	85.00	48.75
D_3	86.25	81.25	87.75

る。また、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。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、多くの特徴点がデータベースの誤った特徴点とのマッチングが行われると自己位置推定結果の誤差が大きくなると考えられる。

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、対象物体の CG モデルを構築し、様々な光源をバーチャル空間でシミュレーションすることで、より多様な特徴点を抽出したデータベースを構築することができる。ただ、精度を良くするためには大きなデータベースが必要になり、データベースを参照して特徴点マッチングを行うとデータベース内の特徴点数に比例して計算時間が大きくなってしまふ。以上のことを踏まえて、全てのデータを参照することなく効率的に特徴点の対応付けを行う方法として、マハラノビス距離を用いたマッチング手法を提案する。次節では、提案するマッチング手法と、利用する特徴点データベースの構築方法について説明する。

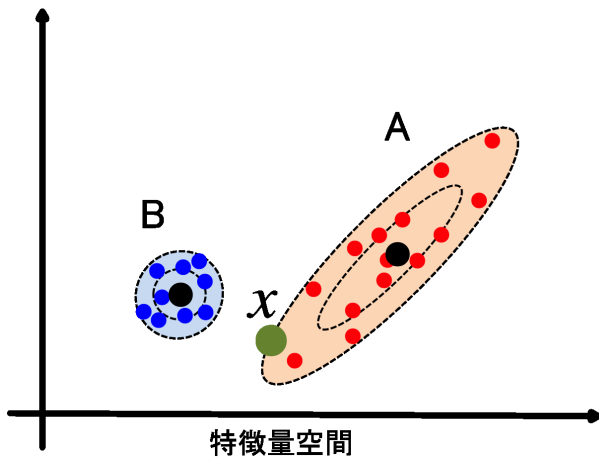


図3 データベース内の点Aの分布は赤色の点で表し、点Bの分布は青色の点で表している。このとき、一般的なマッチングではユークリッド距離の近さから、特徴点 x は点Bとマッチングされる。一方、マハラノビス距離の考えを使うことで、Aとマッチングを行う。

3. 光源環境のシミュレーションおよびマハラノビス距離を用いた特徴点マッチング

2節で示した様に、ある特定の光源環境下で構築した特徴点データベースを参照する一般的な手法の場合、他の条件下でロバストな自己位置推定をすることが難しい。前節で述べたように、バーチャル空間内の3次元CGモデルに対して様々な光源をシミュレーションすることで、特徴点のサンプル数を増やすことが有効だと考えられる。さらに本研究では、巨大なデータベースを構築して直接参照するのではなく、各特徴点の位置毎に光源環境の変化を表現するパラメトリックモデルを設定し、特徴点のマッチングをそのモデルに対して行うことで、データベースの特徴点数に依存しない特徴点マッチングを提案する。具体的には、位置毎の特徴点群が正規分布を成していると仮定し、入力された特徴点と、各特徴点群とのマッチングをマハラノビス距離によって評価を行う。図3の例で示すように、ユークリッド距離を比較に使用した一般的なマッチングと比べて特微量の分散に応じたパラメトリックな基準となる。

3.1 ユークリッド距離とマハラノビス距離

N 次元である特微量 x と特微量 y のユークリッド距離 D^{euc} は次のように定義される。

$$D^{euc} = \sqrt{\sum_{k=1}^N (x_k - y_k)^2} \quad (1)$$

ユークリッド距離に対して、マハラノビス距離は特徴ごとの分散の相関を考慮した考え方である。異なる光源環境で検出された同一とみなせる m 個の特徴点の内、 i 番目のサンプルの特微量は次のように表す。

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iN})^T$$

このとき、 m 個のサンプルの平均値ベクトル μ_i 、分散共分散行列 Σ_i はそれぞれ次のように表される。

$$\mu_i = \frac{1}{m} \sum_{k=1}^m x_i \quad (2)$$

$$\Sigma_i = \frac{1}{m} \sum_{k=1}^m (x_i - \mu_i)(x_i - \mu_i)^T \quad (3)$$

以上を用いて、入力 x に対する i 番目の特徴点とのマハラノビス距離 D_i^{mah} が定義される。

$$D_i^{mah} = \sqrt{\frac{1}{m} (x_i - \mu_i)^T \Sigma_i^{-1} (x_i - \mu_i)} \quad (4)$$

3.2 主成分分析とデータベースへの記録

各特徴点群の分布は、特微量空間の特定の方向について広がりを持たない場合がある。このような分布に対して、マハラノビス距離に用いられる共分散行列を求めつつ、不要な次元と計算量を削減するため、主成分分析を行う。マハラノビス距離の考えを使用する上で主成分分析を行うメリットとして以下の2つが考えられる。

- データセットの種類が少ない（特微量の次数 N 以下）場合でも実行可能
- 特微量の次元を削減することにより処理時間の短縮が可能

選出した点の情報（平均値、共分散行列）をデータベースに記録する際に、各特徴点についてその特微量を主成分分析により次元圧縮を行う。具体的には、 N 次元の特微量空間のデータ分布のうち、最も分散の大きくなる方向から順番に $p (< N)$ 個の基底を選択していき、各データを N 次元空間から p 次元空間の座標へと投影する。このとき、各特徴点群ごとに射影行列も同時にデータベースに記録する。

3.3 マッチングの評価

自己位置推定を行う際は、対象画像から特徴点を抽出し、各点についてデータベース内の点とのマハラノビス距離を調べていく。マハラノビス距離が最も小さい点 y_1 との距離 $D_i^{mah}(x, y_1)$ と2番目に小さい点 y_2 との距離 $D_i^{mah}(x, y_2)$ との間に閾値 θ を用いて表される次の関係が成り立つ時に、特徴点 x と距離が最小の特徴点 y_1 とマッチングを行う。

$$\frac{D_i^{mah}(x, y_1)}{D_i^{mah}(x, y_2)} < \theta \quad (5)$$

この関係式は、誤ったマッチングを減らすために、ユニークな点の組み合わせのみを採用することを意味している。閾値 θ はユークリッド距離を用いた一般的な特徴点マッチングを行う際にもよく用いられる。また、マハラノビス距離の値が大きくなると外れ値の可能性が高くなるため、距

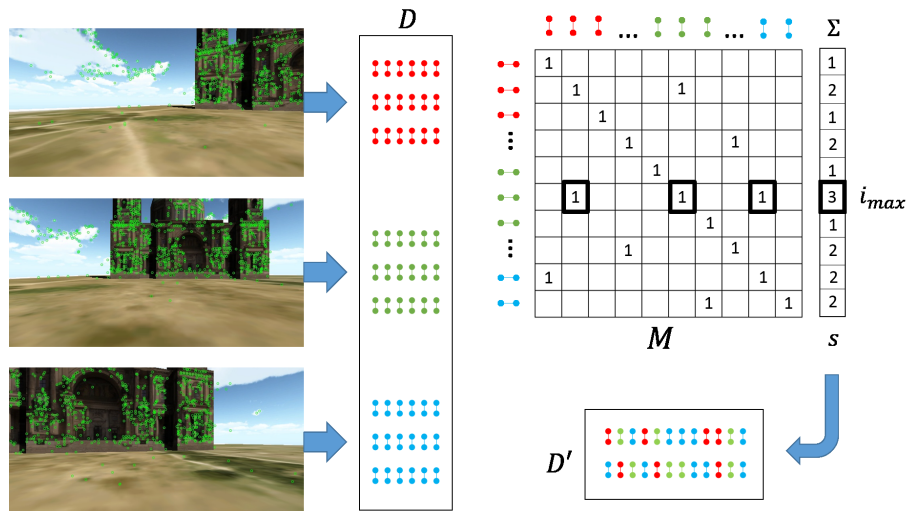


図4 マッチング回数が最大であった特徴点 $d_{i_{max}}$ をデータベース D' に追加する (処理 5).
追加した後, $d_{i_{max}}$ とのマッチング情報を削除し, 以降同様の操作を反復する.

距離が T 未満の場合に限りマッチングを成立させそれ以外の場合は対応点が存在しないと判断する.

$$D_i^{mah}(x, y_1) < T \quad (6)$$

3.4 データベースに記録する特徴点の選出

シミュレーションで生成した特徴点すべてをデータベースに記録してしまうと, 処理に膨大な時間が要してしまう. ここでは, 優先度の高い特徴点を選出する方法について議論する. 特徴点選出の大きな流れとしては, はじめに各光源環境内で出現頻度の高い K 個の特徴点を選択した後, それらの中から光源の異なる環境で出現回数が多い L 種類の点についてデータベースに情報を記録する.

同一の光源環境における代表的な特徴点集合 $D' \subset D$ は, 撮影位置の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. 特徴点の選出アルゴリズムは論文 [3] を参考にし, 以下の流れに従って処理を進める.

(1) 同一環境内の様々なシーンについて特徴点の抽出

全てのシーン画像に関して画像特徴点を抽出し, それぞれの特徴量と位置座標を保存する.

(2) 総当りで全シーンの特徴点のマッチング

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. 同様に各シーンに関して総当りで特徴点のマッチングを行う.

(3) 誤対応の除去

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, 距離が大きなものをすべて除外する.

(4) 各マッチング情報を一つの行列 M に保存

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.

(5) マッチング回数最大の点をデータベースに追加

最もマッチング回数が多かった特徴点の指数 $i_{max} = \text{argmax}_i(s_i)$ を取得し, 図4に示すように特徴点 $d_{i_{max}}$ を代表的な特徴点の集合に追加する ($D' = D' \cup d_{i_{max}}$).

(6) $d_{i_{max}}$ とのマッチング情報を M から除去

以後, 同じ特徴点を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.

(7) (5)~(6) を K 回繰り返す 操作完了後 s を更新して必要な特徴点の数 $|D'| = K$ に到達するまで同様の操作を繰り返す. このようにして選択された K 個の代表的な特徴点に関して特徴量および3次元位置座標をデータセットとして出力する. このデータセットに含まれる特徴点はカメラ位置・方向の変化に対して比較的ロバスト性が高いといえる.

(8) 全特徴点の中から特徴点を選択 異なる光源環境の各データセット間において同一と判断できる特徴点の中から出現頻度が高いものを L 種類選出する. データセット内の特徴点一つ一つについて, 他の環境のデータセット内の特徴点と位置座標の比較を行い, 距離が非常に小さく, かつ特徴量が閾値未満のものを同一の点であるとみなす. このようにして出現数が多いものから順に L 種類の点を選出する.

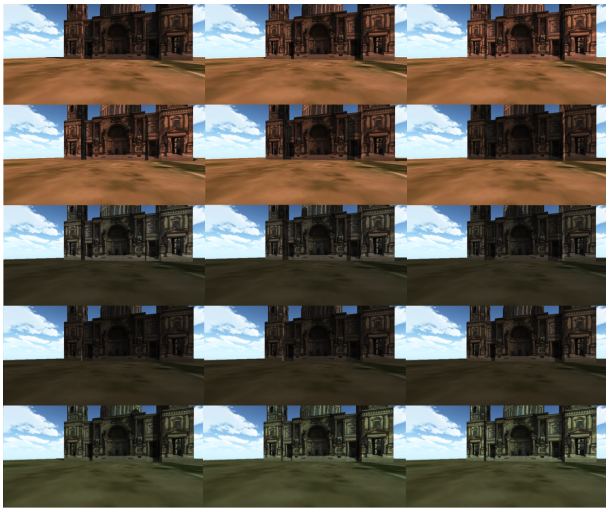


図 5 光源のシミュレーション

4. 自己位置推定シミュレーションの実装と評価

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[1] のモデルを利用した。unity 内にインポートしたモデルファイルを対象に、光源のシミュレーションを行う。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、光の進行方向と光源色を変化させたものを 200 種類用意した (図 5)。その内、100 種類の光源環境はデータベースの構築に使用し、残り 100 種類の光源環境は試験用に未知の光源環境として使用する。各光源環境に関して、50 種類の異なる視点のシーンを取得し、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。続いて、3.4 のアルゴリズムに従い、全シーン画像の中から代表となる画像特徴点を K 個選出し、各々の特徴量と 3次元位置座標をともにデータベースに記録する。

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::SolvePnP Ransac` を用いてカメラの外部パラメータを導出した。

本実験では、大きく 2 種類のデータベース・マッチング手法について比較を行う。1 種類目は、光源環境ごとに生成したデータベースを直接使用し、一般的なユークリッド距離によるマッチングを行う。 P 種類の各光源環境において出現頻度の高い特徴点 K 個を取り出し、データベー

*1 <http://japan.unity3d.com/>

表 2 ユークリッド距離を用いたマッチング手法による自己位置推定成功率

対象	P	K	時間 [ms]	成功率
訓練	1	2000	32	0.88
テスト	1	2000	32	0.52
訓練	10	2000	241	0.87
訓練	100	200	264	0.68
訓練	100	2000	3127	0.97
テスト	10	2000	279	0.79
テスト	100	200	294	0.62
テスト	100	2000	3423	0.90

スに記録したものを参照する。ここでは、ユークリッド距離によるマッチングを行い、具体的には OpenCV で用意されている L2 距離基準に基づいて全探索対応付けを行う `cv::BruteForceMatcher` を使用した。2 種類目は、3 節で説明したマハラノビス距離を用いたマッチング手法を利用する。 P 種類の光源で抽出した特徴点の中から L 種類の特徴点を選びデータベースを構築した。

以上のデータベース・マッチング手法を用いて、光源環境および撮影位置がそれぞれ無作為に選択された 100 枚の画像を対象に自己位置推定の実験を行った。自己位置推定を行い導出した 3次元座標と真値と比較を行い、誤差が小さい場合に自己位置推定を成功と判断する。データベース作成時に使用した光源環境の中から無作為に一つの環境を選択し、さらにその環境において撮影された画像の中から無作為に一枚選択する。このようにして光源、位置ともに無作為に 100 種類の画像を選択したものを「訓練光源画像」と呼ぶ。一方、データベース作成時には使用していない 100 種類の光源環境において撮影された画像を無作為に 100 種類用意したものを「テスト光源画像」と呼ぶ。本実験では、各手法においてこの 2 種類の画像データ集合に対して自己位置推定のテストを実施した。

表 2 では、ユークリッド距離を用いたマッチング手法の実験結果をまとめたものである。1~2 行目では、ある一つの光源環境において作成したデータベースを参照し、自己位置推定テストはそれぞれ同一環境と、異なる環境で実施している。異なる光源環境 (テスト光源画像) に対しては成功率が 0.52 となっており、失敗する確率が高いのがわかる。一般的なデータベースでは、異なる環境での自己位置推定の成功率が下がることがわかる。次に、100 種類の光源の中から代表的な特徴点を選出してマッチングを行ったものについてみると、訓練・テストどちらの対象画像でも、特徴点の数や光源の種類が増えるにつれ成功率が上がるが、処理速度が大きくなる傾向にある。

続いて、マハラノビス距離を使用した手法について評価を行う。表 3 は主成分の数、表 4 は記録する特徴点数 P 、表 5 はマッチングの閾値 (T, θ) をそれぞれ変化させてデータベースを構築した場合の実験結果である。これらを見て

表 3 マハラノビス距離によるマッチングを行った場合の自己位置推定成功率 (変数: 主成分数)

対象	P	L	主成分	(T, θ)	時間 [ms]	成功率
訓練	100	2000	8	10, 0.7	1294	0.62
訓練	100	2000	12	10, 0.7	1867	0.83
訓練	100	2000	16	10, 0.7	2557	0.86
テスト	100	2000	8	10, 0.7	1427	0.52
テスト	100	2000	12	10, 0.7	2061	0.82
テスト	100	2000	16	10, 0.7	2862	0.87

表 4 マハラノビス距離によるマッチングを行った場合の自己位置推定成功率 (変数: 特徴点の数)

対象	P	L	主成分	(T, θ)	時間 [ms]	成功率
訓練	100	200	12	10, 0.7	226	0.78
訓練	100	400	12	10, 0.7	469	0.81
訓練	100	1000	12	10, 0.7	936	0.86
訓練	100	2000	12	10, 0.7	1867	0.83
テスト	100	200	12	10, 0.7	237	0.82
テスト	100	400	12	10, 0.7	472	0.81
テスト	100	1000	12	10, 0.7	1032	0.87.5
テスト	100	2000	12	10, 0.7	2061	0.82

表 5 マハラノビス距離によるマッチングを行った場合の自己位置推定成功率 (変数: 閾値 T, θ)

対象	P	L	主成分	(T, θ)	時間 [ms]	成功率
訓練	100	1000	12	5, 0.5	952	0.86
訓練	100	1000	12	5, 0.7	984	0.87
訓練	100	1000	12	20, 0.5	951	0.90
訓練	100	1000	12	20, 0.7	987	0.82
テスト	100	1000	12	5, 0.5	1031	0.88
テスト	100	1000	12	5, 0.7	1044	0.85
テスト	100	1000	12	20, 0.5	1028	0.85
テスト	100	1000	12	20, 0.7	1029	0.85

みると、主成分数や特徴点数が増えるに連れて推定精度は上がっているが、処理時間も比例して増大していることがわかる。また、訓練光源画像とテスト光源画像どちらに対してもほぼ同じ精度を保っていて、未知の光源環境に対しても一定の精度を出すことができている。

以上の結果より、マハラノビス距離を利用した手法は、ユークリッド距離を使用した場合に比べて処理時間が大きくなってしまいが、少ない特徴点数で精度の高い自己位置推定を行うことができる。また、本実験でユークリッド距離によるマッチングは OpenCV の最適化された matcher を使用しているのに対し、マハラノビス距離の実装では最適化ができておらず、処理時間には改善の余地があると考えられる。

5. まとめ

本研究では、光源環境が変化する環境に対してもロバストな自己位置推定を行うことができる特徴点データベースの構築方法を提案した。バーチャル空間で異なる光源をシミュレーションにより多数のサンプルを再現することで、

同一の点の特徴量の分散を仮想的に評価した。特徴量の比較にはマハラノビス距離を使用して光源の影響を考慮したマッチングを行う手法を提案した。自己位置推定による評価実験の結果から、従来手法と比較して、位置推定の精度が向上することを確認した。ただし、従来手法と比較して処理時間は大きい。マハラノビス距離の計算に用いる主成分の数を少なくすることで処理時間はある程度改善されるが、これは推定の安定性とトレードオフの関係にある。今後の課題は、現実環境の3次元モデルを使用したデータベースの構築を行い、本研究と同様の手順で現実環境の撮影画像を対象とした自己位置推定の精度を確認することである。

参考文献

- [1] Lukas Gruber, Steffen Gargitz, 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.
- [2] Daniel Kurz, Peter Georg Meier, Alexander Plopski, and Gudrun Klinker. An outdoor ground truth evaluation dataset for sensor-aided visual handheld camera localization. *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 263–264, October 2013.
- [3] Daniel Kurz, Thomas Olszawski, and Selim Benhmane. 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] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, November 2004.
- [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.
- [6] 工藤彰, Alexander Plopski, Tobias Hollerer, 間下以大, 清川清, 竹村治雄. 異なる光源環境における画像特徴の頑健性の調査. 研究報告コンピュータビジョンとイメージメディア (CVIM), Vol. 2015, No. 65, pp. 1–5, jan 2015.