



Inter-frame Delay を考慮したローリングシャッターカメラの トラッキング手法の検討

瀬戸 将志*¹ 武富 貴史*¹ 山本 豪志朗*¹
Christian Sandor*¹ 加藤 博一*¹

Abstract – 本報告ではローリングシャッターカメラによって取得された画像に対するカメラ位置・姿勢推定手法を提案する。ローリングシャッターカメラの位置・姿勢推定問題では、現フレームの画像取得終了から次フレームの画像取得開始の間に生じる **Inter-frame Delay** を考慮する必要がある。この **Inter-frame Delay** の取得には特殊な機器が必要であり、また、**Inter-frame Delay** の大きさはカメラの露光時間に依存して動的に変化するという問題がある。そこで、本研究では、**Inter-frame Delay** の大きさに依存しないカメラ位置・姿勢推定を実現する。具体的には、複数の **2D-3D** の対応関係を用い、単一フレームからフレームの取得開始時のカメラ位置・姿勢とフレームの取得終了時のカメラ位置・姿勢の **12** パラメータを推定することで、フレーム取得中のカメラ運動を推定する。シミュレーション環境および実画像を用いた実験により、提案手法はローリングシャッターによる画像歪が生じた場合においても高精度にカメラ位置・姿勢を推定できることを確認した。

Keywords : カメラ位置・姿勢推定, ローリングシャッター, **Inter-frame Delay**, 拡張現実感

1 はじめに

現実環境中に仮想物体を位置合わせして重畳表示する拡張現実感、ナビゲーションや作業支援などの様々な分野への応用が期待される。これらの分野では、タブレット型端末の普及により、多くのハンドヘルド型拡張現実感アプリケーションが開発されている。このようなアプリケーションでは、タブレット端末に搭載されているカメラによって撮影された画像を仮想物体の合成対象として用いるビデオスルー型拡張現実感が採用されている。現在、ほとんどのタブレット端末に搭載されているカメラは、シャッター方式としてローリングシャッターを採用している。ローリングシャッターカメラでは、スキャンラインごとに画像データの取得タイミングが異なる。このような、ローリングシャッターカメラを用いたビデオスルー型拡張現実感では、シャッター方式に起因した歪により以下の問題が生じる。

- カメラ位置・姿勢推定精度の低下
- 合成結果画像の品質の低下

多くの場合、ビデオスルー型拡張現実感では、カメラで取得された画像を合成対象として用いるだけでなく、現実環境と仮想環境の位置合わせを実現するために必要なカメラの位置・姿勢推定にも用いている。このようなカメラ位置・姿勢推定技術はカメラトラッキングとも呼ばれる。現在提案されている多くのトラッ

キング手法は、グローバルシャッターカメラを想定している。グローバルシャッターカメラでは全てのスキャンラインの情報が同時に取得されるため、各フレームごとに単一のカメラ位置・姿勢を推定すればよい。しかし、ローリングシャッターカメラでは、各スキャンラインごとに画像情報の取得タイミングが異なるため、各スキャンラインごとのカメラ位置・姿勢を推定する必要がある。そのため、グローバルシャッターカメラを想定した手法を用いた場合には、位置・姿勢推定精度が低下するという問題が生じる。一方で、拡張現実感画像を生成するための背景画像として、ローリングシャッターカメラで撮影した画像をそのまま背景画像として用いた場合には、ローリングシャッターに起因した画像歪により仮想物体と背景画像間で幾何学的な不整合が生じる。

これらのローリングシャッターカメラを用いたビデオスルー型拡張現実感において生じる問題を解決するためには、フレームの取得中に生じるカメラ運動を推定する必要がある。そこで、本研究では、複数の二次元座標と三次元座標の対応関係を用いて、単一の画像からフレーム内運動を推定する手法を提案する。

2 関連研究

コンピュータビジョンの研究分野では、これまでにフレーム内でのカメラ運動を推定するための様々な手法が提案されている。これらの手法は、フレーム間で得られる二次元点の対応関係 (2D-2D 対応) からカメラの運動を推定する手法と、環境中の三次元点とその

*¹奈良先端科学技術大学院大学

画像上での観測座標の対応関係 (2D-3D 対応) からカメラ運動を推定する手法に分類できる。以下では、それぞれの手法について概観する。

2.1 2D-2D 対応に基づく方法

2D-2D 対応に基づく手法では、連続フレーム間で画像上の二次元座標の対応関係を取得し、これらの対応関係より各スキャンラインのカメラ位置・姿勢を推定する。Forssen ら [1] はカメラの回転を等角速度運動、並進を等速度運動であると仮定してカメラ位置・姿勢を補間することでフレーム内運動を推定した。Lovegrove ら [2] はカメラ位置・姿勢と環境の三次元構造を同時に推定する Simultaneous Localization and Mapping において、B スプライン曲線を用いてローリングシャッターカメラの運動の軌跡を連続的に表すことでフレーム内運動を推定している。また Hedborg ら [3] や Klein ら [4] はバンドル調整の中で、フレーム内の姿勢を隣接する 2 フレーム間との線形補間を用いて推定している。

文献 [2][4] では、最終スキャンラインの画像が取得された直後に次フレームの第 1 スキャンラインの画像の取得が開始されることを仮定している。しかし、実際のローリングシャッターカメラでは、図 1 に示すように、画像情報読み込み時間 (Read out time) に加え、連続するフレーム間にフレームの取得タイミングを一定とするための Inter-frame Delay と呼ばれる時間が存在する。そのため、これらの手法では、Inter-frame Delay が大きい場合に誤差が生じるという問題がある。この問題に対して、文献 [1][3] では、Inter-frame Delay を考慮したローリングシャッターカメラモデルを用いたフレーム内運動の推定を実現している。この手法では、事前に計測した Inter-frame Delay の値を推定処理に用いているが、この Inter-frame Delay の計測は煩雑であるという問題がある。また、Inter-frame Delay はカメラの露光時間に依存して動的に変化するため、推定中は露光時間を Inter-frame Delay 計測時と同じ値に固定する必要がある。

2.2 2D-3D 対応に基づく方法

2D-3D の対応関係を用いる手法は、多くの場合、単一フレームからフレーム内運動を推定することができる。入力となる 2D-3D の対応関係は三次元モデルやマーカなどを用いて拡張現実感を実現するアプリケーションでは、容易に取得することができる。

Magerand ら [5] や Albl ら [6] はフレーム内のカメラ運動が微小であることを仮定し、カメラの回転運動を線形近似した。このモデルを線形に解くことでカメラの位置・姿勢を推定している。特に文献 [6] は線形解法に必要な最小点数を明らかにし、高速に推定することが可能な手法を提案している。しかし、一般的に

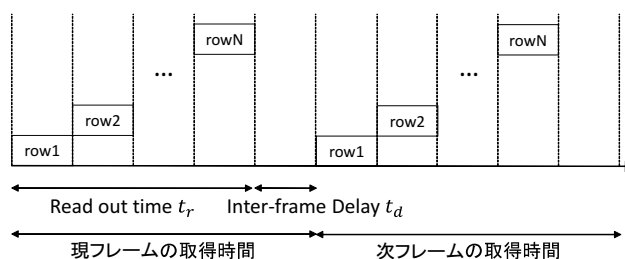


図 1: ローリングシャッターカメラが画像を取得する流れ

は線形解法はノイズがある場合には推定結果が不安定になるという問題がある。一方で、Ait-Aider ら [7] は非線形最小化の枠組みでフレーム内運動を推定する手法を提案している。しかし、Ait-Aider らの手法では、露光時間の変化による各スキャンラインの取得タイミングの変化および Inter-frame Delay の存在が考慮されていないという問題がある。

3 Inter-frame Delay を考慮したカメラ位置・姿勢推定

本研究では、2D-3D の対応関係よりフレーム内運動を推定する手法を提案する。提案手法は、単一のフレームからフレーム内運動を推定することが可能であり、また、露光時間の変化による Inter-frame Delay の変動に依存しないという特長を持つ。以下では、まず、提案手法で採用するローリングシャッターカメラモデルについて述べる。その後、2D-3D の対応関係よりフレーム内運動を推定する手法を述べる。

3.1 ローリングシャッターカメラモデル

グローバルシャッターカメラを用いた場合における三次元点の画像上への投影は、内部パラメータ行列 \mathbf{K} とカメラ位置・姿勢行列 \mathbf{T} を用いて以下のように表される。

$$s_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \mathbf{KT}\mathbf{X}_i \quad (1)$$

ここで、 (x_i, y_i) は三次元点 \mathbf{X}_i の画像上への投影位置を表し、 s_i はカメラ座標系における三次元点 \mathbf{X}_i の奥行きを表す。

一方、ローリングシャッターカメラでは画像中の各スキャンラインは異なる時刻に取得される。図 2 にスキャンラインとカメラ位置・姿勢の関係を示す。図に示すように、ローリングシャッターカメラにおける三次元点 \mathbf{X}_i の画像上への投影は、スキャンライン y_i に対応するカメラ位置・姿勢行列 $\mathbf{T}(y_i)$ を用いて表される。

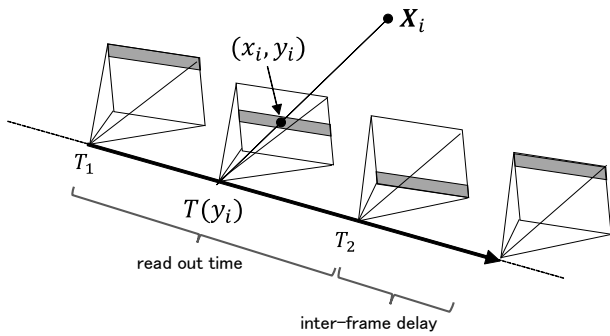


図2: スキャンラインとカメラ位置・姿勢の関係

$$s_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \mathbf{K} \mathbf{T}(y_i) \mathbf{X}_i \quad (2)$$

このようにカメラ位置・姿勢 \mathbf{T} は画像の y 座標に依存しているため、フレーム内運動を推定するためには、画像の全スキャンラインに対してカメラ位置・姿勢を推定する必要がある。一般的にカメラ位置・姿勢は6パラメータによって表されるため、画像のスキャンライン数 $\times 6$ 個のパラメータを推定する必要がある。これらのパラメータを推定するためには、多くの対応点が必要となり、また、推定処理には多くの計算コストが必要となるという問題がある。このような問題に対し、本研究では、次項で詳述するフレーム内運動を補間によって算出するアプローチにより推定すべきパラメータ数の削減を行う。

3.2 2D-3D 対応からのフレーム内運動の推定

推定すべきパラメータが膨大になるという問題に対して、これまでに各スキャンライン間の位置・姿勢の変化は微小であるという仮定 [5][6] や補間 [1] によって各スキャンラインにおけるカメラ位置・姿勢を内挿する手法が提案されている。しかし、これらの手法では、カメラ運動が大きい場合は対応できない、位置と姿勢を独立に補間しているという問題がある。このような問題に対して、本研究では、第1スキャンラインのカメラ位置・姿勢 \mathbf{T}_1 と最終スキャンラインのカメラ位置・姿勢 \mathbf{T}_2 を推定し、他のスキャンラインのカメラ位置・姿勢については $\text{SE}(3)$ を補間することで算出する。これにより、カメラ運動が大きい場合においても対応可能となり、また、位置・姿勢のパラメータを同時に補間することが可能となる。ここで、 $\text{SE}(3)$ の補間は Exponential map を用いて以下のように表される。

$$\mathbf{T}(y_i) = \exp\left(\frac{y_i}{N_r} \ln(\mathbf{T}_2 \mathbf{T}_1^{-1})\right) \quad (3)$$

ここで N_r は全スキャンラインの数である。このような補間方法を用いることにより、Inter-frame Delay に依存せずに各スキャンラインのカメラ位置・姿勢を表現することが可能となる。提案手法におけるローリングシャッターカメラによる三次元点の画像上への投影は次のように表される。

$$s_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \mathbf{K} \exp\left(\frac{y_i}{N_r} \ln(\mathbf{T}_2 \mathbf{T}_1^{-1})\right) \mathbf{T}_1 \mathbf{X}_i \quad (4)$$

このモデルでは第1スキャンラインのカメラ位置・姿勢と最終スキャンラインのカメラ位置・姿勢の2姿勢を同時に推定するため、12個のパラメータを推定する必要がある。本研究では、式(4)の投影モデルを用いて、2D-3Dの対応関係より以下の再投影誤差に基づくコスト関数が最小となるパラメータを推定する。

$$E = \sum_{i=0}^N \rho(\|\bar{\mathbf{x}}_i - \mathbf{x}_i\|^2) \quad (5)$$

$$\rho(t) = \frac{t^2/2}{1+t^2} \quad (6)$$

ここで、 $\bar{\mathbf{x}}_i$ は画像中で観測される対応点の二次元座標を表し、 \mathbf{x}_i は式(4)で与えられる三次元点 \mathbf{X}_i を画像上へ再投影した際の座標を表す。ただし、対応点の検出誤差や誤対応を考慮し、M推定によるロバスト推定を行う。本研究では、影響関数 $\rho(t)$ としてM推定における Geman-McClure 関数を用いた。また、コスト関数 E の最小化には Levenberg-Marquart 法を用いた。

4 実験

提案手法の有効性を示すために、シミュレーション環境によるカメラ位置・姿勢推定精度の定量評価を行った。また、実画像を用いてグローバルシャッターを仮定してカメラ位置・姿勢を推定した場合と提案手法を用いてカメラ位置・姿勢を推定した場合における再投影誤差の比較を行った。

4.1 シミュレーション環境実験

シミュレーションでは推定に用いる点の数を変化させた場合の推定精度と、画像上の観測座標に含まれる誤差の大きさを変化させた場合の推定精度について検証を行った。

点の数を12点~60点まで変化させたシミュレーション結果を図3に示す。この実験では対応点の画像座標に対して平均0画素、標準偏差1画素のガウシアンノイズを付加した。三次元点はカメラ前方3000mm \times 3000mm \times 3000mmの空間内でランダムに生成した。また、カメラ位置・姿勢の初期値は、姿勢に対して真値から5

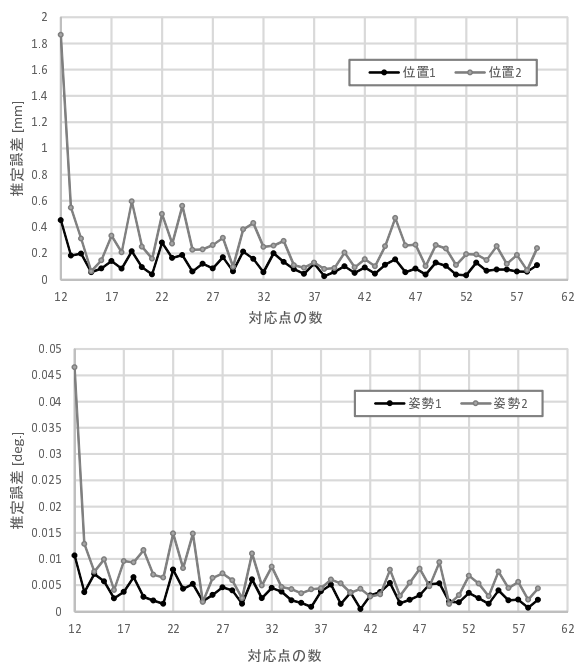


図3: 対応点数とカメラ位置・姿勢推定精度の関係

度, 位置の各要素に対して真値から 5mm のオフセットを加えた値を与えた. 推定されたカメラ位置・姿勢の位置誤差および姿勢誤差は以下のように算出した.

$$\mathbf{R}_{error} = \mathbf{R}_{est} \mathbf{R}_{gt}^T \quad (7)$$

$$Error_t = \|\mathbf{t}_{est} - \mathbf{t}_{gt}\| \quad (8)$$

ここで \mathbf{R}_{est} , \mathbf{t}_{est} はそれぞれ推定されたカメラの姿勢と位置であり, \mathbf{R}_{gt} , \mathbf{t}_{gt} はそれぞれカメラの姿勢と位置の真値を表す. ただし, 姿勢誤差については, 得られた姿勢行列をロドリゲスの公式により回転軸, 回転角度に変換した際の回転角度を誤差とした. 図3からわかるように点の数が25点以上であれば, 対応点が誤差を含んでいる場合においても安定した推定結果を得られることが確認できる. この結果より, 以下の実験では安定した推定結果を得るために十分な点数として60点の対応点を入力として用いた.

対応点座標に対して1~10画素の間で標準偏差を変化させたガウシアンノイズを加えた. 三次元点は, カメラ前方 3000mm×3000mm×3000mm の空間内でランダムに生成した. 図4は各ノイズレベルで50回ずつカメラ位置・姿勢を計算した結果の統計値を示している. この結果より, 対応点に付加した誤差が10画素以下の場合, 姿勢の推定誤差は0~1.5度程度となっている. また, 位置の推定誤差は3000mm四方の空間に対して0~40mmとなっている. これらの結果より, 提案手法は対応点の観測座標に誤差が含まれる

場合においても高精度に推定が可能であることが確認できる.

4.2 実環境実験

実画像を用いた実験では, パンおよびチルトしたカメラによって, チェッカーパターンを撮影した. 三次元点としてチェッカーパターンの格子点を利用し, 対応する画像上での格子点座標は手作業によって27点抽出した. これらの対応点を用いて推定されたカメラ位置・姿勢を用いて三次元点を画像上へ再投影した際の誤差を算出した. 比較対象として, グローバルシャッターカメラのためのカメラ位置・姿勢推定結果をOpenCVのsolvePnP()関数を用いて取得した. この際, solvePnP()のオプションとしてcv::SOLVEPNP_ITERATIVEを指定した.

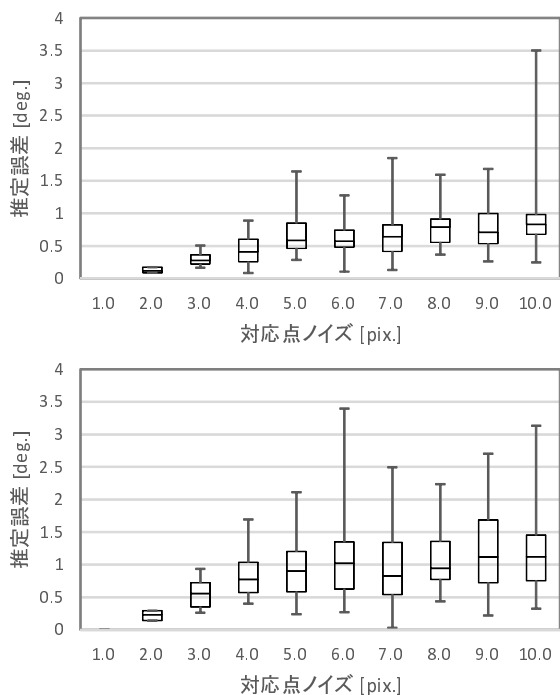
表1に各画像における各対応点の再投影誤差の総和を示す. 推定した12枚の画像全てにおいて, 提案手法の再投影誤差が小さくなっていることが確認できる.

また, 図5に推定されたカメラ位置・姿勢を用いて, 三次元点を画像上に再投影した結果を示す. 赤は三次元点を画像上に投影した点, 緑は画像上で観測される格子点である. 撮影中カメラは動いているため, 画像はローリングシャッターによって歪められている. そのため, 従来のグローバルシャッターを仮定して推定したカメラ位置・姿勢から得られる三次元点の投影点は, 画像中央から離れるほどに画像上の観測点から大きくずれていることがわかる. これは実際にはスキャンラインごとに異なるカメラ位置・姿勢を一つのカメラ位置・姿勢に対応させたためである. 一方, 提案手法では, 画像全体で三次元点の画像上への投影点と画像上の観測点が一致しており, 高精度にフレーム内運動を推定できていることが確認できる.

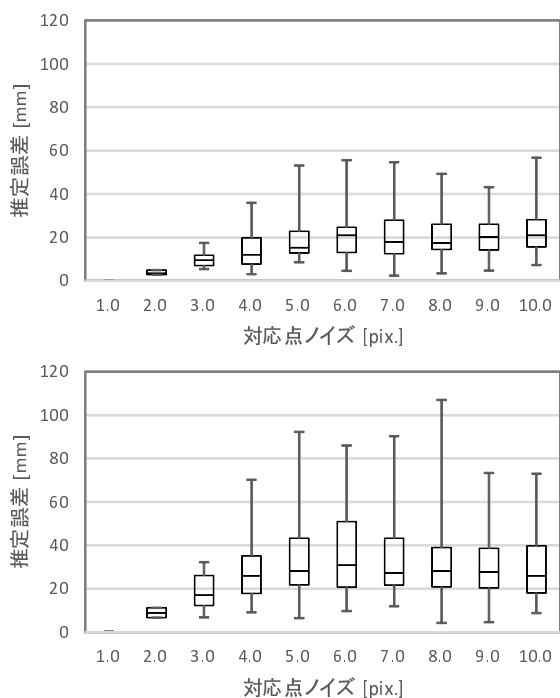
5 まとめ

本報告では, ローリングシャッターカメラに対するカメラ位置・姿勢推定手法を提案した. 提案手法では第1スキャンラインにおけるカメラ位置・姿勢と最終スキャンラインにおけるカメラ位置・姿勢を推定し, その間のスキャンラインにおけるカメラ姿勢をSE(3)補間によって算出した. これにより, Inter-frame Delayの大きさに依存しないカメラ位置・姿勢の推定を実現した. シミュレーション環境における定量評価と実画像を用いた比較実験により, 提案手法を用いることで, ローリングシャッターによる画像歪が生じた場合においても, 高精度にカメラ位置・姿勢が推定できることを確認した. 今後の課題として, フレーム内運動を考慮した対応点の検出などがあげられる.

瀬戸・武富・山本・Sandor・加藤：Inter-frame Delay を考慮したローリングシャッターカメラのトラッキング手法の検討



(a) 姿勢の推定誤差 (上：第1 スキャンラインのカメラ姿勢，下：最終スキャンラインのカメラ姿勢)



(b) 位置の推定誤差 (上：第1 スキャンラインのカメラ位置，下：最終スキャンラインのカメラ位置)

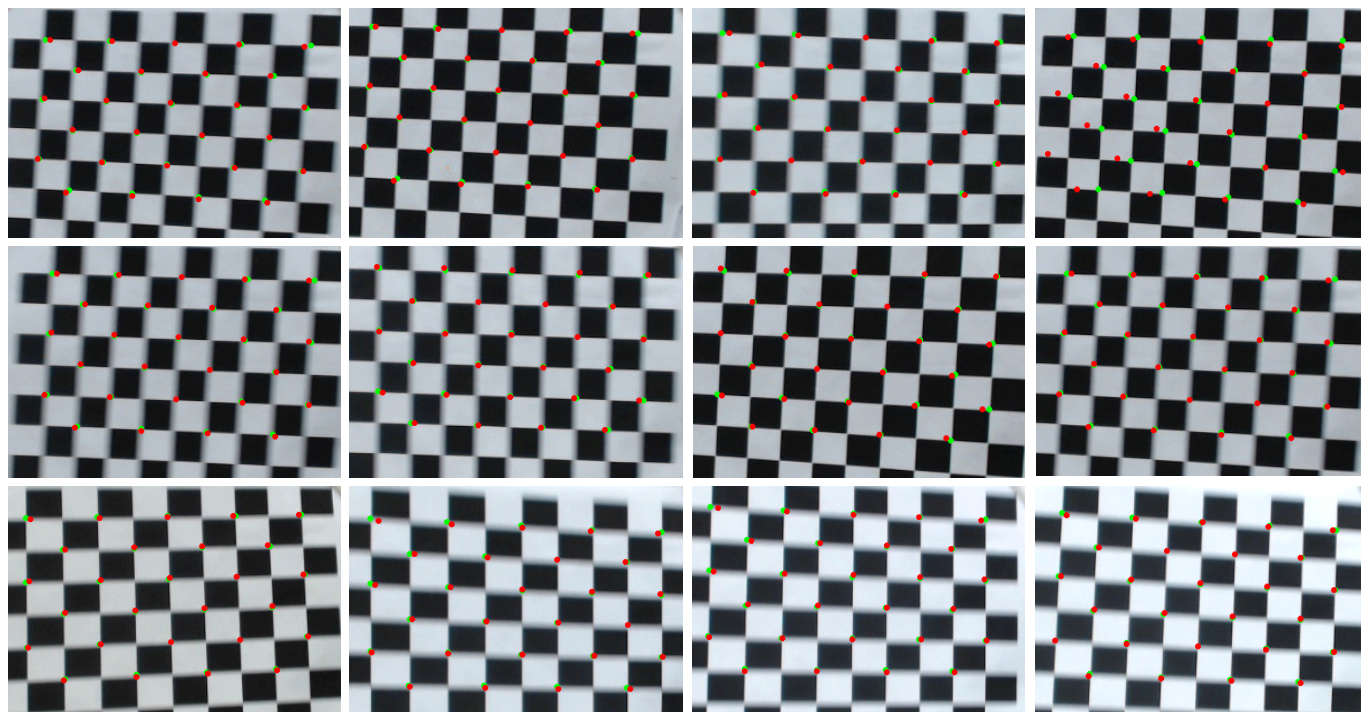
図4: 対応点に誤差を与えたときの推定誤差

表1: 実画像を用いた再投影誤差

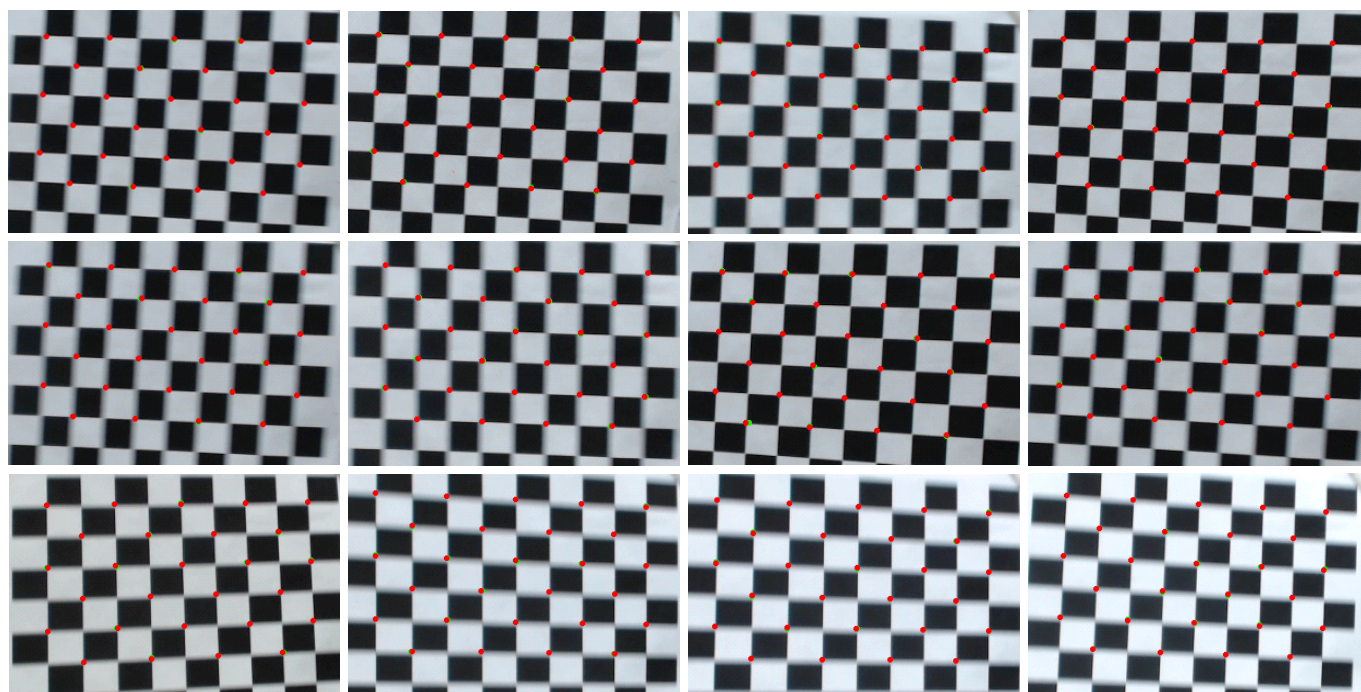
画像 No.	solvePnP[pix.]	提案手法 [pix.]
1	8.80	1.85
2	8.22	2.27
3	8.21	2.65
4	11.9	2.20
5	8.16	2.06
6	7.93	2.41
7	8.43	2.28
8	8.26	3.01
9	7.71	2.64
10	8.56	3.02
11	9.25	2.69
12	7.77	2.75
平均値	8.60	2.49
標準偏差	1.14	0.36

参考文献

- [1] Forssen, P.-E. and Ringaby, E.: Rectifying rolling shutter video from hand-held devices, Proc. Computer Vision and Pattern Recognition, pp. 507 - 514 (2010).
- [2] Lovegrove, S., Patron-Perez, A. and Sibley, G.: Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. Proc. British Machine Vision Conference, pp. 93.1 - 93.12 (2013)
- [3] Hedborg, J., Forssen, P.-E., Felsberg, M., and Ringaby, E.: Rolling shutter bundle adjustment, Proc. Computer Vision and Pattern Recognition, pp. 1434-144 (2012).
- [4] Klein, G., and Murray, D.: Parallel tracking and mapping on a camera phone, Proc. International Symposium on Mixed and Augmented Reality, pp. 83-86 (2009)
- [5] Magerand, L., Bartoli, A., Ait-Aider, O. and Pizarro, D.: Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching, European Conference on Computer Vision, Part 1, pp. 456 - 469 (2012)
- [6] Albl, C., Kukulova, Z. and Pajdla, T.: R6P - Rolling Shutter Absolute Camera Pose, Proc. Computer Vision and Pattern Recognition, pp. 2292 - 2300 (2015)
- [7] Ait-Aider, O., Andreff, N., Lavest, J.-M., and Martinet, P.: Simultaneous Object Pose and Velocity Computation Using a Single View from a Rolling Shutter Camera. European Conference on Computer Vision, Part 2, pp. 56-68 (2006)



(a) グローバルシャッターを仮定した推定



(b) 提案手法

図 5: 推定したカメラ位置姿勢による三次元点の画像上への投影結果