

多眼カメラシステムによるカメラ運動の高精度実時間推定

武田悟郎^{†1} 岡谷貴之^{†1} 出口光一郎^{†1}

本稿では、カメラを動かして得た画像系列を元に、カメラの運動およびシーンの3次元形状を復元する運動からの形状復元 (structure from motion, SFM) において、実時間でこれを行うこと (Realtime SFM) を考える。その成否は、シーンの同一点を画像系列上で追跡した特徴点軌跡を以下に安定して高精度に得られるかにかかっている。本稿では、複数のカメラを剛体結合した多眼カメラシステムを用いて、特徴点軌跡のロバストかつ高精度な計算を行うことを提案する。カメラが運動する中、常に新しい特徴点が画像に発生しまた消えてゆくという最も難しい状況において、実時間性の前提から、現在までの画像系列しか得られない因果性の制約を、効果的に解決できることを示す。

Highly Accurate Realtime Camera Motion Estimation Using a Multicamera System

GORO TAKEDA,^{†1} TAKAYUKI OKATANI^{†1}
and KOICHIRO DEGUCHI^{†1}

This paper considers the problem of realtime structure from motion (SFM), in which the camera motion and the scene structure are recovered in realtime from the image sequence captured by moving the camera. Its success entirely depends on whether or not the loci of feature points are obtained stably and accurately that are the results of tracking identical scene points across the image sequence. In this paper, we propose to use a multi-camera system such that multiple cameras are rigidly combined in order to robustly and accurately obtain the loci of feature points. We show that this method can solve the difficulty with the temporal causality that only the images up to the latest frame are available; as a result, the most difficult cases can be dealt with where new feature points always appear and some of the current points always disappear while the camera moves.

^{†1} 東北大学大学院情報科学研究科
仙台市青葉区荒巻字青葉 6-6-01

1. はじめに

本稿では、カメラが空間を自由に移動するとき、撮影された画像を元にカメラの運動およびシーンの3次元構造を推定する問題、すなわち、運動からの形状復元 (structure from motion, 以下 SFM) を考える。特に、実時間性を前提とする実時間 SFM (Realtime SFM) が議論の対象である。実時間性とは、カメラを動かしている最中に実時間 (オンライン) で、カメラの運動およびシーンの3次元形状を推定することを指す。なお本稿では、SFM の最も基本的な状況、すなわち静止シーンの中をカメラが運動する場合を想定する。

現在、SFM には多くの応用や異なる呼称があるが、元々の SFM では、実時間性は中心的課題ではなかったと言える。つまり、まずビデオカメラを自由に動かして画像系列を撮影しておき、その後、その画像系列から3次元形状をオフラインで復元することが目的であった。これに対し、近年では、実時間性が求められる応用が数多くある。例えば、推定したカメラの運動を用いて、実画像に CG で合成した物体を重ねこむ複合現実感 (Mixed reality) や、ロボティクスの分野での SLAM (Simultaneous Localization And Mapping) である。SLAM を使えば、移動ロボットにカメラを搭載し、ロボットの周囲環境の地図を構築しつつ、またロボットの運動を知ることが出来る。

実時間性を実現する上で考えるべきは、計算量および因果性である。計算量が小さいことは、画像1フレームを短い時間で処理できる必要があるから、当然求められる。もう一つの因果性とは、得られる時系列データ (= 画像系列) が、時間軸上で因果的であることを指す。つまり、パラメータの推定に利用できるのは、撮影した最新の画像とそれ以前の時刻の画像のみである。もし、実時間性を考えなくてもよければ、カメラが運動して撮影した画像系列すべてを使ってパラメータを推定することができる。この2つの場合の違いは決して小さくない。本稿では、この因果性から生じる諸問題に焦点をあて、それを解決する方法を考える。

SFM がうまくいくかどうかは、実時間であってもそうでなくても、シーンの同一点を画像系列上で追跡した軌跡を多数、安定して高精度に得られるかどうかにかかっている。もしこれらを得ることができれば、その後の処理、すなわちカメラの運動と3次元形状 (それらシーンの点の3次元位置) の計算は、基本的には既に解決された問題であって、技術的には困難さはない。

このような点の軌跡を得る一般的な方法は、Harris コーナー検出器、SIFT⁶⁾、SURF¹⁾ 等の方法を、各時刻の画像に適用して特徴点を抽出し、その特徴点をそれぞれの方法ごとに異なる特徴量を用いることで、画像系列上で追跡する。しかしながら、このようなボトムアップ

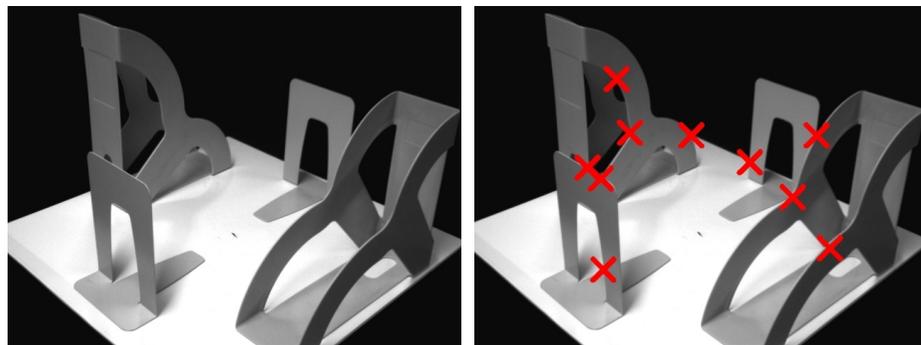


図1 遮蔽輪郭上に現れる特徴点の例

プの画像解析手法のみによって得られる特徴点軌跡には、誤りが発生することが避けられない。これは、特徴点として通常、画像上で目立つ点が選択されるが、そのような点（に対応するシーンの点）が、3次元的に視点を動かしたとき、安定して画像上で追跡できるような点であるとは限らないことによる。われわれの経験上最も問題となるのは、図1に例を示すような、物体の遮蔽輪郭上で、遮蔽された側の物体の輪郭と遮蔽輪郭との間で作るT字型のジャンクションが偽の特徴点となる場合である。

もちろん、この種の軌跡の誤りは、画像間の幾何学的拘束によって除去できることはよく知られている⁹⁾。この幾何学的拘束はカメラ運動によって変化し、それを知るには正しい特徴点軌跡が必要であるから、両者を同時に推定する必要がある。通常、カメラ運動に伴う多視点幾何拘束を、RANSACに代表されるロバスト推定の方法と組み合わせて、この同時推定を実現する方法が採られる。上述のような、物体の遮蔽輪郭上に現れる偽の特徴点は条件が良ければこの方法によって効果的に除去される。

しかしながら、この方法も万能ではない。ロバスト推定の方法は、基本的には統計的な多数決の原理に従うものであるから、あまりに多数の嘘の特徴点軌跡が存在したり、あるいは得られる特徴点そのものの数が少ない場合には、容易に破綻を来す。

一方、近年盛んに研究されているSLAMの研究では、追跡すべき特徴点を、その3次元位置等が確定されたものと、そうでないものとに分け、それぞれで扱い方を変える方法が主流である。これらの方法は、ある意味では、3次元位置が既知のランドマークがシーン中にあるときに、これを基準にカメラ運動を推定する方法に近い。したがって、カメラが同じシーンにある程度長い時間停留するような場合に大変うまく働く（あるいはそのように設

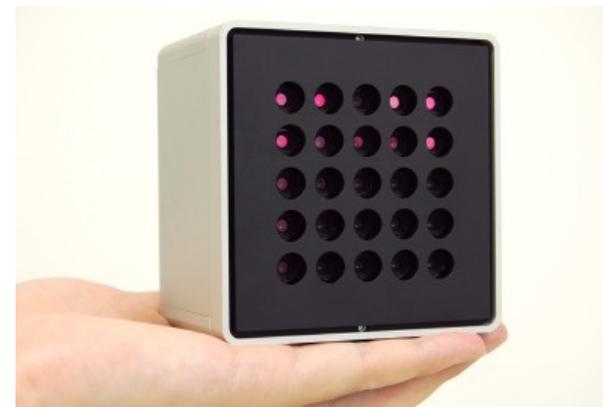


図2 本研究で使用した多眼カメラシステム (ProFUSION25)。

計されている³⁾。しかしながら、これらの方法は、SFM本来の姿である、カメラ運動と3次元形状の同時推定を、なるべく行わないような戦略をとっていると言える。それゆえに、カメラがシーン内を大きく動き回り、常に新しい特徴点が画像内に飛び込んでくるような状況はどうしても不得意である。

SFM本来のカメラ運動と3次元形状の同時推定を行うことで、カメラがシーン内を大きく動き回り、常に新しい特徴点が画像内に飛び込んでくるような状況にも対処できるようにしたい。実時間性を前提とし、上述の因果性を考慮に入れるとき、そのためには、画像に現れた新しい特徴点を、カメラを動かすことなく「一瞥」するだけで追跡すべき特徴点とそうでないものに振り分けることができれば理想的である。本稿では、これを、複数カメラが剛体結合された多眼カメラシステムを用いて実現することを考える（図2に本研究で使用したシステムを示す）。具体的には、多視点幾何拘束を使って、最新の画像1枚の上で、各特徴点が追跡すべきものかどうかを計算する。多眼カメラシステムは事前に高精度に校正しておき、各カメラの内部パラメータに加えて外部パラメータ（カメラ間の相対的なもの）を求めておいて、ここから多視点線形幾何拘束（multilinear constraint）を計算し、利用する。

2. 関連研究

2.1 SLAM との比較

先述の通り、SLAMの研究は近年その発展が著しい³⁾。SLAMとSFMにはほとんど違いは

ないが、実時間性を前提とする点が唯一の違いだろう（すなわち **Realtime SFM = SLAM**）。コストや扱い易さの面から、単眼カメラを用いる **SLAM** の方法がこれまでよく研究されてきている。そこでは、追跡すべき特徴点を、その3次元位置等が確定されたものと、そうでないものに分け、それぞれで扱い方を変える方法が採られる。いわば3次元位置が既知の点（ランドマーク）を基準として、カメラ運動を推定するのに近い。そのため、3次元位置の確定した点を得るまでは、本来の **SFM** 同様、カメラ運動と3次元形状の同時推定を行う必要がある、それには（推定した3次元位置がある程度信頼できるようになるまでに）通常、一定の時間（あるいはカメラが運動すること）を要する（形状既知の参照用パターンが見えていることを前提とするものもある）。ある特徴点の集合について、その3次元位置が一度確定されれば、（それらが画像内に捉えられている限り）カメラ運動を（形状復元とは切り離して）単独で推定できる。ひとたびそのような特徴点集合を得られれば、それを足場に、その周囲のシーンの特徴点に対して、その3次元位置を次々に求めてゆくことで、ランドマークを増やして行ける。この辺の処理をエレガントに行うのが **SLAM** の手法の本質である³⁾。

このような **SLAM** の戦略は、同一シーンにカメラが長い時間停留するような場合に、カメラの運動をロバストかつ高精度に推定することを可能にする。その一方、カメラがよりダイナミックに空間内を移動するような場合には本質的に向かない。その場合には、時々刻々フレッシュな特徴点が画像内に現れては消え、それが繰り返されるからである。

2.2 多視点カメラの研究

先述の通り本稿では、多眼（多視点）カメラを利用する。これに関する研究を以下に述べる。Solàらは、多視点カメラによる **SLAM**⁹⁾ を提案している。これは、上述の **SLAM** 初期化時の「時間遅れ」を、多視点カメラでシーンの点の3次元位置を直接計算して、解決しようとするのが基本的な考え方である。本研究で用いる多眼カメラシステムでは、カメラ間のベースラインは微小なものであって、これをステレオカメラとして奥行きを計算してもその精度には限界があり、それ単独では最終的なシーンの形状推定結果とはしない。(Solàら同様、初期値に利用することは行いが、本稿の議論での本質ではない。)

Okutomiらは⁸⁾は、マルチベースラインステレオの概念を提案している。多視点カメラあるいは多視点画像を使ったステレオ復元のサーベイが、Gallupらの論文⁴⁾に詳しい。なお、Gallupらは同論文で、多視点カメラを使った可変ベースライン・可変解像度ステレオの方法を提案している。通常、ステレオ復元の精度は、カメラからの奥行きに応じて通常変化

するが、例えばビデオカメラを使って撮影するなど多視点からの画像がふんだんに得られる状況下で、適切な画像を選択する方法を示した。

Yangらは¹⁰⁾、特徴点を追跡することなくカメラの運動を推定するための方法として、微分カメラという概念を提案している。これは、複数のカメラを剛体結合させたものを使って、カメラが微小な並進運動および回転運動する際の画像変化を、現実の画像としてカメラから取得し、シーンのアピアランスに基づいてカメラ運動を推定するものである。物体認識で有名な固有空間法に基づき、カメラの姿勢変化とその際の画像変化の関係を線形近似し、これを実現している。特徴点を追跡しないことで、対象とするシーンの幅を広げることができるという。本稿の方法は、この研究に影響を受けている。特徴点こそ使うものの、カメラが姿勢を変化させたときに得られる将来の画像を、位置姿勢を変えて接合された別のカメラから直接取得するという考え方をそのまま利用する。

3. 多眼カメラを使った SFM

3.1 基本的な考え方

Yangらの微分カメラ¹⁰⁾の中心的なアイデアは、カメラが微小量だけ移動したときの画像変化を、複数カメラシステムを使って予想しようとするものである。本稿でもこの考え方を特徴点の取捨選択において利用する。具体的には、次のような方法で特徴点を追跡する。

まず、各カメラの最新の画像に対し、新しい特徴点を抽出し、既に追跡中の特徴点については直前の画像との間で追跡し、軌跡を更新する。カメラシステムを動かすとき、各カメラは剛体結合されているので常に同一の運動をする。なお、各カメラの時間的な同期はとれているとする。特徴点の抽出・追跡アルゴリズムはここでは特に限定せず、いわばブラックボックスとして扱う。それぞれのアルゴリズムに特徴があるので、状況に応じて使い分けることも考えられる。

次に、同時刻に撮影した各カメラの画像間で、全特徴点の追跡を行う。具体的には、基準カメラ（ここではカメラアレイの中心カメラを選んだ）の画像と、それ以外のカメラの画像間で、各特徴点を追跡する（マッチングを求める）。このときの追跡には、同一カメラの画像系列内で特徴点の追跡を行うアルゴリズムと同一のアルゴリズムを用いる。

基準カメラに対するその他のカメラは、上述のように、（大雑把にはあるが）基準カメラが将来運動したときの画像を、現時刻において得ているとみなせる。上のように、同一時刻において基準カメラとその他のカメラの画像間で特徴点を追跡することは、基準カメラを微小に移動させたときの特徴点の追跡結果を、再現したものと考えることができる。同一時

刻の異なるカメラ間の追跡と、異時刻の同一カメラ間での追跡に、同一アルゴリズムを用いるのはここに理由がある。

こうして得た、基準カメラから他カメラへの特徴点追跡結果に対し、多視点幾何拘束を用いて、その特徴点軌跡の正しさをテストする。例えば基準カメラと他のカメラ1台の間でのエピポラ拘束を、両画像間の追跡結果に適用し、条件を満足しない特徴点を除去する。カメラの台数が多いほど多数の幾何拘束が得られ、その分ロバスト性が向上すると考えられる。カメラの組み合わせによってベースラインが複数方向存在することも重要だろう。

カメラ間の多視点幾何拘束を利用するために、カメラシステムは事前に校正しておく。各カメラは内部パラメータが不変であるとし、各カメラの内部パラメータおよびカメラ間の相対的位置・姿勢を得ておく。これには、Camera Calibration Toolkit を用いた²⁾。多視点幾何拘束は事前に計算しておく。

上述のようにして、正しい特徴点軌跡をロバストに得ることを目指すが、多視点を持つことは、ロバスト性のみならず、純粋な特徴点検出精度の向上にも寄与すると考えられる。各特徴点の検出誤差が独立な確率分布に従うとすれば、多視点画像にてシーンの同一点を検出すれば、カメラ台数分の精度向上が望める（単一の特徴点の検出誤差の分散を、カメラ台数で除したものとなる）。

3.2 カメラ位置推定アルゴリズム

多眼カメラシステムを用いた Realtime SFM の計算の流れを図3に示す。単一カメラの場合の標準的な方法を基本とし、そこに後述の複数カメラによる特徴点除去のステップが追加された形となる。

(1) 初期化.

まず基準カメラの画像から特徴点を抽出し、複数カメラによる特徴点の除去を行う。除去後の特徴点について、複数のカメラ（上述の5つのカメラを用いたステレオにより、初期の特徴点の3次元位置を算出する。（これは次節の実験で単一のカメラの場合でも、このステップだけは複数カメラを用いて同様に、初期の3次元位置を算出する。）

(2) 特徴点の追跡.

基準カメラの画像において、最新の画像と直前の画像間で、特徴点を追跡する。この追跡には誤追跡が含まれている可能性があるため、RANSACを用いたロバスト推定を行う。具体的には、前回のキーフレームにおける特徴点の画像上の座標 \mathbf{m}_1 と追跡

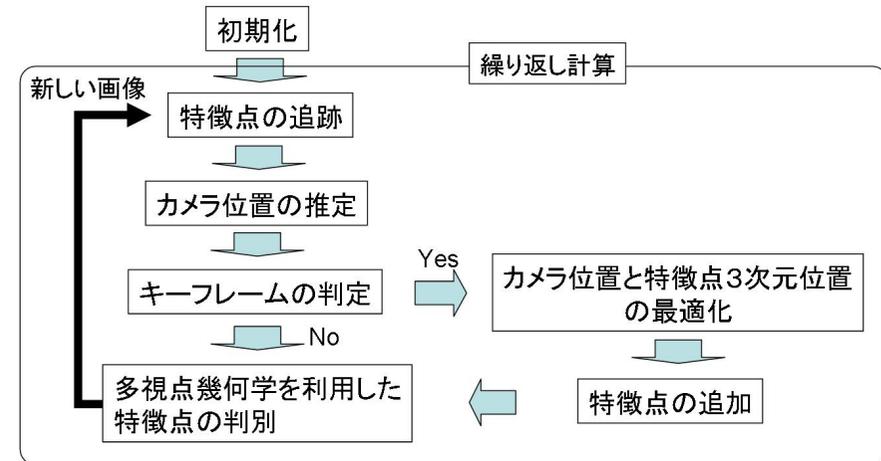


図3 多眼カメラシステムを用いた Realtime SFM の処理フロー

された特徴点の座標 \mathbf{m}_2 が、空間中で同一の点であるならば(1)式を満たすことを利用する。ここで、 \mathbf{F} は基礎行列と呼ばれる 3×3 の行列である。

$$\mathbf{m}_1^T \mathbf{F} \mathbf{m}_2 = 0 \quad (1)$$

(3) カメラ位置の推定.

特徴点の画像上の座標と、その特徴点の3次元位置を画像上に投影した座標との2乗距離を再投影誤差と定義し、ステップ2で追跡されたすべての特徴点の再投影誤差の和を最小化しカメラパラメータを推定する。

(4) キーフレームの判定.

入力フレームがキーフレームかどうかを後述の条件で判定する。キーフレームと判定された場合は、ステップ5, 6を行う。

(5) カメラ位置と特徴点の3次元位置の最適化.

後述する逐次バンドルアジャストメントを行い、カメラ位置と特徴点の3次元位置の最適化計算を行う。

(6) 特徴点の追加.

画像から特徴点を抽出し、既存の特徴点から一定距離はなれている特徴点を新しい特徴点として追加する。このような処理を行うのは、既存の点と同一の点を選ばないよ

うにするためである。

(7) 多視点幾何拘束を利用した特徴点の除去.

追跡すべき特徴点を選択し、そうでないものを削除する。

ステップ 1 で初期化を行った後、新しい画像が入ってくる度にステップ 2 へ戻って、繰り返し計算を行う。

3.3 特徴点の除去

次節に示す実験では、図 2 に示した 25 眼カメラシステムのうち、中央の 1 台のカメラを基準カメラとし、できるだけ視差が大きくなる 5×5 正方形の対角線上の 4 台のカメラ（右上、左上、右下、左下）を選び、合計 5 つのカメラを用いて上述のアルゴリズムを適用し、特徴点の除去を行った。もちろん 25 台すべてを利用すればそれだけ精度の向上が見込められると思われるが、複数カメラを用いる効果を検証する目的でこのようにした。

そして、基準カメラとその他のカメラのカメラ間のエピポーラ条件を用いて正しい特徴点を選択し、それ以外を除去した。この判定は、基準カメラの特徴点位置を元に対応するカメラ画像上に引いたエピポーラ線と、特徴点間の距離を閾値処理することで行った。

この 5 台のカメラ間の多視点幾何学拘束を利用して特徴点の除去を行った結果を図 4 に示す。物体の遮蔽輪郭上で、遮蔽された側の物体の輪郭と遮蔽輪郭との間で作る偽の特徴点がきちんと除去できていることが確認できる。

3.4 逐次バンドルアジャストメント

オフラインでの SFM では、計算時間を考慮する必要がないため、観測されたすべてのデータを用いて、バンドルアジャストメントを行い、全パラメータを推定することができる。(この方法を以後バッチ計算と呼ぶことにする。) 一方、Realtime SFM では、最新の時刻までに得たすべての観測データを利用してバッチ計算を行うと、次第に問題の規模が大きくなり、短時間での計算が難しくなってしまう、リアルタイム性が失われる。これを解決するため、若干の精度を犠牲に、問題の規模を一定に保つ計算方法が、逐次計算法⁷⁾である。

系の規模を一定以下に保つには、明らかにパラメータのうちのどれかを、系から削除する必要がある。将来の観測と関連付けられるパラメータは削除できないので、そうでないパラメータの削除を考えることになる。中でも、古い観測だけにしか関連付けられていないパラメータは削除しても影響が少なそうである。本稿では、最新のカメラ位置と、そのひとつ前のカメラ位置、現在、追跡できている特徴点の 3 次元位置を推定すべきパラメータとし、それ以外のパラメータを逐次削除していく。

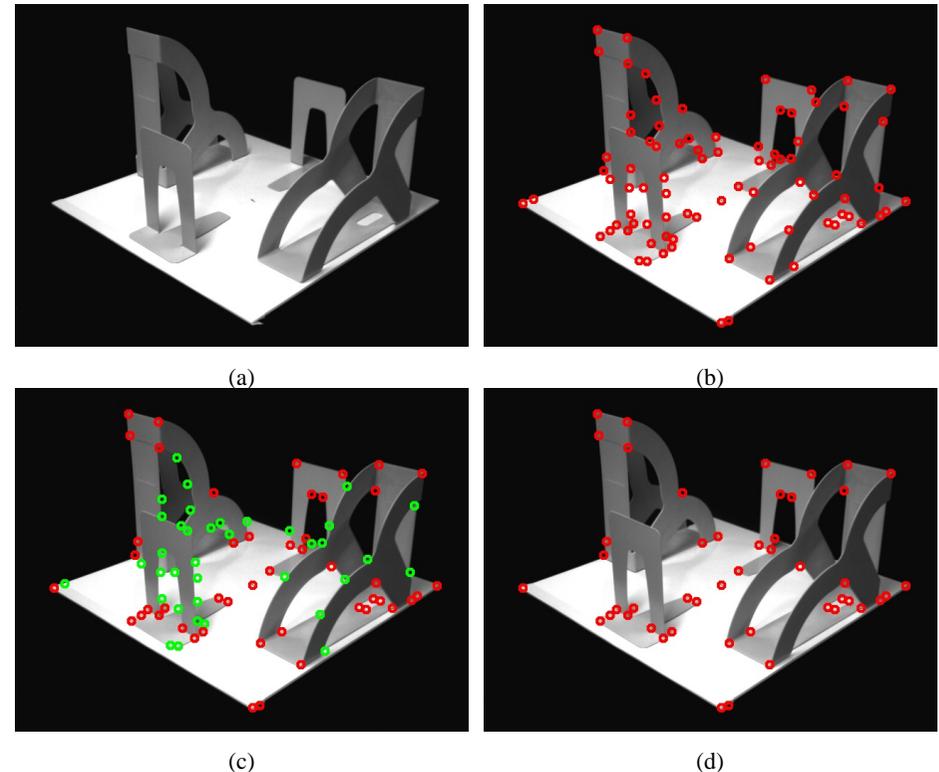


図 4 多視点幾何拘束を利用した特徴点の除去. (a) 元画像 (b) 赤の丸で示したところが抽出された特徴点. (c) 多視点幾何学拘束を利用して特徴点の除去を行う. 除去された特徴点を緑の丸で示す. (d) 除去後の特徴点

これまでに m 枚の画像を取得し、今、新たに $m + 1$ 枚目の画像 1 枚を取得したとする。このとき、これまでの評価関数を E_{prev} 、新しい観測による評価関数を E_{new} 、とすると、ここで新たに最小化すべき評価関数 E_{all} は以下のように表せる。

$$E_{all}(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{y}_1, \mathbf{y}_3) = E_{prev}(\mathbf{x}_{m-1}, \mathbf{x}_m, \mathbf{y}_1, \mathbf{y}_2) + E_{new}(\mathbf{x}_m, \mathbf{x}_{m+1}, \mathbf{y}_1, \mathbf{y}_3) \quad (2)$$

ここで、 \mathbf{x}_i は i 枚目におけるカメラパラメータであり、 \mathbf{y}_1 は m 枚目以前から観測されており $m + 1$ 枚目でも観測された特徴点の 3 次元位置、 \mathbf{y}_2 は m 枚目以前から観測されており $m + 1$ 枚目では追跡の失敗あるいは上述の特徴点の除去判定を受けたことにより観測で

きなかった特徴点の3次元位置, \mathbf{y}_3 は m 枚目から新たに観測された特徴点の3次元位置である. $\mathbf{x}_m, \mathbf{y}_1$ は E_{prev} から引き継がれるパラメータであり, $\mathbf{x}_{m-1}, \mathbf{y}_2$ は系から削除されるパラメータで, $\mathbf{x}_{m+1}, \mathbf{y}_3$ が新たに追加されるパラメータである. 本稿では, E_{prev} を2次関数で近似し, E_{new} のみを非線形として扱い, E_{all} を最小化する. 最小化の詳しい方法については, 参考文献⁷⁾を参照していただきたい.

3.5 キーフレームの判定

前節では, 逐次バンドルアジャストメントについて述べたが, 実際のカメラ位置推定アルゴリズムでは, 毎フレームこの処理を行うわけではない. なぜなら, m 枚目と $m+1$ 枚目でカメラ位置がほとんど変わらない場合では, 新しく追加された特徴点の3次元位置の奥行きに対する推定誤差が大きくなってしまふ. そこで次節で示す実験では, 入力フレームの特徴点の追跡結果からカメラ位置を推定し, その位置が前回のキーフレーム時のカメラ位置から一定距離以上であれば, その入力フレームをキーフレームと判定し, 逐次バンドルアジャストメントを行う.

4. 実験

上述のような多眼カメラシステムを用いて特徴点追跡を行った場合と, 単一カメラのみで従来通りの方法で特徴点追跡を行った場合とで, 位置推定の精度がどのように変化するか検証した. 上述のように, 本研究で用いた25眼のカメラシステムのうち5つのカメラを選択し, これらの特徴点追跡のために利用した. 単一カメラは, 多視点幾何学を利用した特徴点の除去以外は多眼カメラシステムと同一のアルゴリズムを使用した.

実験では, 推定精度を明快な形で評価するために, カメラシステムを動かす代わりにこれを固定し, 対象の方を回転させ, その位置姿勢を推定させた(つまり, カメラが対象の回りを回っているのと同じ状況). 特に偽の特徴点が発生しやすいシーンを用意し実験を行った.

実験結果を図5, 図6, 図7, 図8に示す. 図7は図6の一部(緑の枠で囲われている箇所)を拡大したものである. 図8は推定されたカメラ位置を点列としてプロットしたものである. 赤が多眼カメラシステム, 緑が単一カメラである. 単一カメラでは, 追加した新たな特徴点の中に嘘の特徴点がかくまれているため, その位置推定結果に影響が出ていることが見て取れる. 多眼カメラシステムでは, 安定して良い特徴点だけを瞬時に選び追跡できており, そのため, カメラの位置がほぼ真円に近く高精度に推定できていることがわかる.

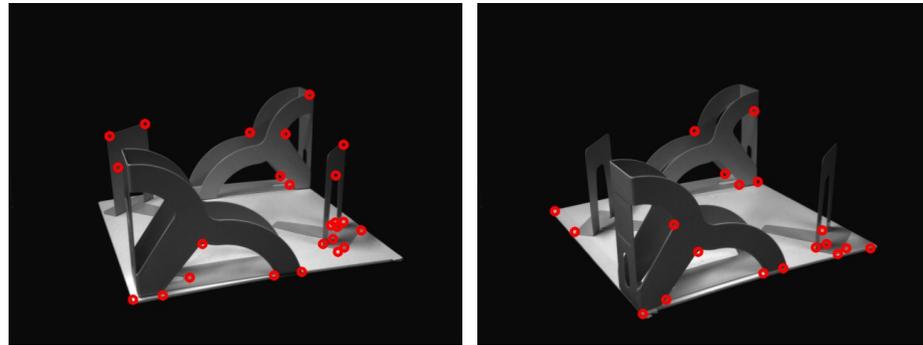
5. おわりに

SFMを実時間で行おうとする際, 現在までの画像系列しか使えないという因果性の制約が発生する. SFMでは特徴点の追跡を高精度に行うことが大切である. しかし, 単一カメラではある程度カメラを動かしたあとにならないと, 追跡すべき特徴点かどうかを判定できない. これでは, 次々と新しい特徴点が入ってくるようなシーンではそのカメラ位置推定は容易に破綻してしまう. 我々は, 多眼カメラシステムを用いることで, カメラを動かすことなく現れた新しい特徴点を追跡すべきか否かを判定する手法を提案した. これにより, 単一カメラに比べバーストにカメラの位置姿勢を推定できることを実験により確認した.

今後の課題としては, カメラ位置の推定結果を定量的に評価することや, キーフレームの判定条件の改善, 多視点を利用した特徴点検出精度の向上などに取り組みたいと考えている.

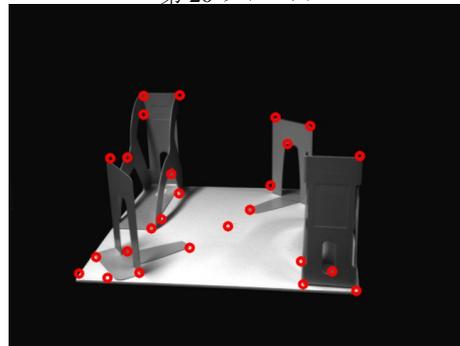
参考文献

- 1) H.Bay, T.Tuytelaars, and L.J.V. Gool. Surf: Speeded up robust features. In *Proc ECCV*, pages 404–417, 2006.
- 2) J.Bouguet. Camera calibration toolbox for matlab.
- 3) A.Davison, I.Reid, N.Molton, and O.Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.
- 4) D.Gallup, J.-M. Frahm, P.Mordohai, and M.Pollefeys. Variable baseline/resolution stereo. In *Proc. CVPR*, pages 1–8, 2008.
- 5) R.I. Hartley and A.Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2000.
- 6) D.G. Lowe. Object recognition from local scale-invariant features. In *Proc ICCV*, pages 1150–1157, 1999.
- 7) P.F. Mclauchlan. A batch/recursive algorithm for 3d scene reconstruction. In *Conf. Computer Vision and Pattern Recognition*, pages 738–743, 2000.
- 8) M.Okutomi and T.Kanade. A multiple-baseline stereo. *IEEE Trans. PAMI*, 15(4):353–363, 1993.
- 9) J.Solà, A.Monin, M.Devy, and T.Vidal-Calleja. Fusing monocular information in multicamera slam. *IEEE Trans. Robotics*, 24(5):958–968, 2008.
- 10) H. Yang, M. Pollefeys, G. Welch, J.-M. Frahm, and A. Ilie. Differential camera tracking through linearizing the local appearance manifold. In *Proc. CVPR*, pages 1–8, 2007.



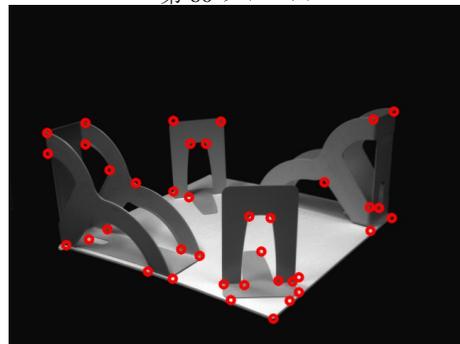
第 20 フレーム

第 140 フレーム



第 60 フレーム

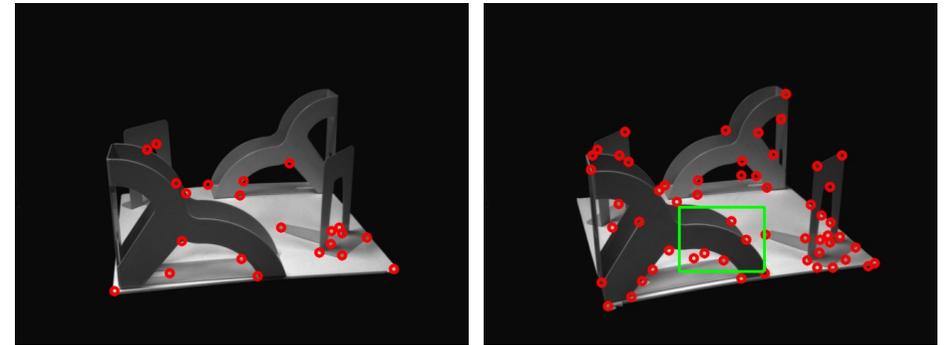
第 180 フレーム



第 100 フレーム

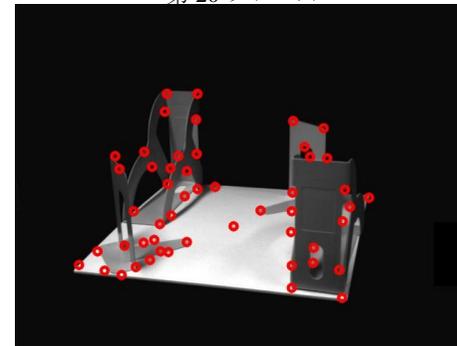
第 220 フレーム

図 5 多眼カメラシステムを用いたときの特徴点



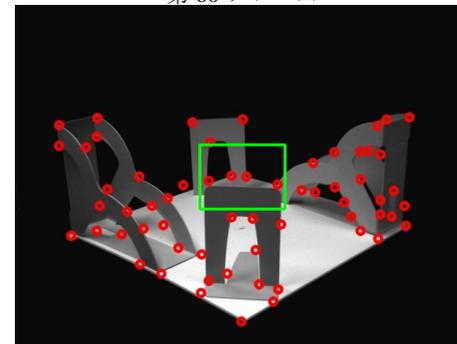
第 20 フレーム

第 140 フレーム



第 60 フレーム

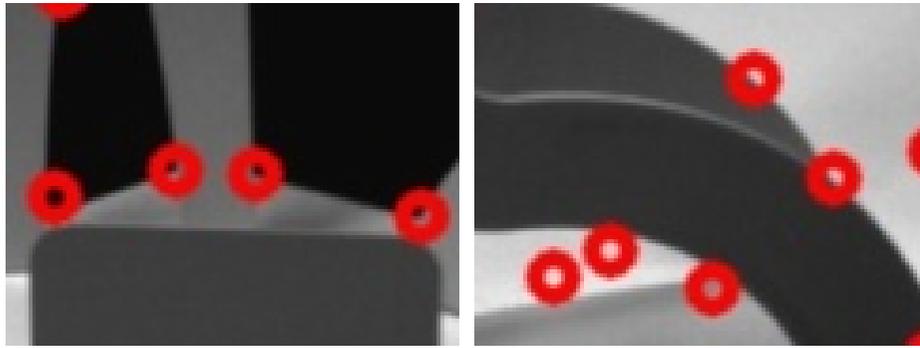
第 180 フレーム



第 100 フレーム

第 220 フレーム

図 6 単一カメラを用いたときの特徴点



第 100 フレーム

第 140 フレーム

図 7 単一カメラの特徴点には、RANSAC によるロバスト推定を行っても嘘の特徴点が追加されてしまっている。

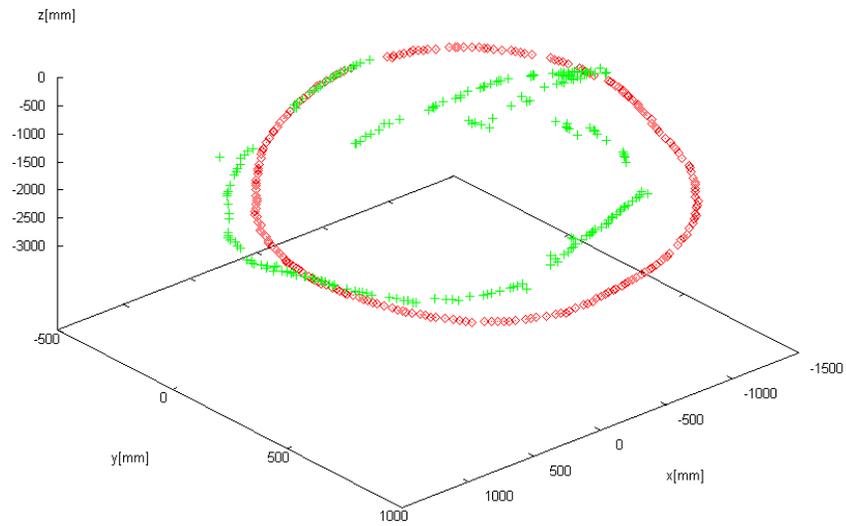


図 8 推定されたカメラ位置. 赤が多眼カメラ, 緑が単一カメラを示している。