

バンドルアジャストメント

岡 谷 貴 之^{†1}

バンドルアジャストメントは、画像から幾何学的なモデルのパラメータを推定する問題において、高い推定精度の達成を目指して、非線形最適化を数値的に—それが大規模なものになっても避けることなく—実行する方法である。本稿では、このバンドルアジャストメントを解説する。具体的には、数値計算の基本的アルゴリズムと実装方法、統計的推測としての理論的背景、そして座標選択に関わるゲージの自由度についてそれぞれ述べる。また、時系列の観測に対するオンライン推定を実現するための再帰的計算方法についても述べる。

Bundle Adjustment

TAKAYUKI OKATANI^{†1}

Bundle adjustment is a general method for the problem of estimating parameters of a geometric model from images. It is characterized by that nonlinear optimization is numerically performed to attain high estimation accuracy, even if the size of the optimization is very large. In this article I present a tutorial on this method. Specifically, I describe basic numerical algorithms with their implementation techniques, theoretical grounds in terms of statistical inference, and gauge freedom associated with the choice of a coordinate system. Moreover, I present a method for recursive computation that realizes online estimation for time series observations.

1. はじめに

バンドルアジャストメント (bundle adjustment, 以降バンドル調整と書く) とは、画像から、その背後にある幾何学的なモデルのパラメータを推定するための方法である。主要な対

象は、多視点画像を用いた 3 次元形状復元やカメラの位置・姿勢の推定問題であるが、バンドル調整自体は方法というよりむしろ方法論であって、多様な問題を対象とし得る。特に推定の精度を重視し (多くの場合大規模な) 非線形最適化を数値的に実行することが特徴である。

そのルーツは、写真測量 (photogrammetry) の分野で 1960 年代頃から開始された一連の研究にある。バンドル調整という名称は、地形に位置する複数の基準点を撮影した画像から、カメラの位置姿勢等を計算するのに、基準点とカメラを結ぶ光線の束 (bundle) を画像に対して微調整することから来ている (歴史については文献¹⁰⁾ の付録 A を参照)。90 年代に入って、コンピュータビジョンの分野でこの方法が改めて見直され、写真測量では想定されなかった問題に応用されるとともに、いくつかの改良も加えられてきた。現在では極めて標準的な手法として広く使われている。

本稿は、このバンドル調整を解説したものである。特に

- (1) 初学者向けのチュートリアルとなり、
- (2) 本稿の内容だけでバンドル調整が一通り利用できるようになり、
- (3) より発展的な話題への導入をも与える。

ことを目標に書いた。

コンピュータビジョンの研究者が、バンドル調整についてまとめた文献に、Triggs, McLauchlan, Hartley, Fitzgibbons による極めて優れた解説¹⁰⁾ がある。様々な実装上の技法や考え方について詳細な記述があり、出版後 10 年が経過してなおその内容は古くなってはいない。しかしながら、多様な話題をカバーすべく分量が総じて多く、一方である程度の知識を前提に複雑な内容を簡潔な記述で済ませている部分もあり、初学者には敷居が高いと思われる。特に、再帰的な計算方法を扱った章やゲージの自由度を扱った章で、それは顕著であるように思われる。その一方、バンドル調整の統計的推測の側面がほとんど述べられていない問題がある。

これを踏まえて本稿では、実用性の観点を重視しつつ内容を絞り、記述をコンパクトにすることで、初学者にも理解しやすくなるように心がけた。単に Triggs らの解説¹⁰⁾ のダイジェストとするのではなく、バンドル調整の統計的推測の側面の記述 (4 節) と、再帰的計算法の異なるアプローチでの説明 (6 節) に差異もある。本稿を読んだ後に改めて Triggs らの解説を読めば、一層理解が深まるだろう。

以降、2 節で基本的な例題を示し、バンドル調整を概説した後、3 節で基本的な数値計算のアルゴリズムおよびその実装方法について述べ、4 節で統計的推測の観点からバンドル調

^{†1} 東北大学 大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

整を説明し、推定精度の評価方法を述べる。5節では座標選択に関わるゲージの自由度を解説し、6節では時系列観測を対象としたオンライン推定のための再帰的計算方法を説明する。

なお以降、数式の書体は Hartley-Zisserman の教科書³⁾ にならい、ベクトルをボールドローマン体 (\mathbf{x} や θ など)、行列はタイプライタ体 (\mathbf{A} や \mathbf{H} など) で表記する。ただし行列のうち、フィッシャー情報行列および観測情報行列のみ、筆記体で \mathcal{I} , \mathcal{J} と書く。

2. バンドル調整の概要

最初に具体例として、多視点画像からシーンとカメラ運動を同時に推定する次のような典型的問題を取り上げ、バンドル調整の概要を述べる。

例 2.1. 3次元空間に適当に配置された m 台のカメラで、空間に散らばった n 個の点を m 枚の画像に撮影したとする。これらの画像から、点の3次元座標およびカメラの位置・姿勢等パラメータを求めたい。

点やカメラの位置の関係は次のように記述する。カメラを $i = 1, \dots, m$ で表し、カメラ i に対するカメラ行列 \mathbf{P}_i (3×4 行列) を

$$\mathbf{P}_i \equiv \mathbf{K}_i [\mathbf{R}_i \mid \mathbf{t}_i] \quad (1)$$

と書く (\mathbf{K}_i はカメラの内部行列, \mathbf{R}_i , \mathbf{t}_i はカメラの姿勢を表す回転行列および並進ベクトル)。点は $j = 1, \dots, n$ で表し, j 番目の点の3次元座標を収めた3次元ベクトルを $\mathbf{q}_j = [X_j, Y_j, Z_j]^T$ と表記する。この点の、カメラ i の画像上での座標 $\mathbf{z}_{ij} = [u_{ij}, v_{ij}]^T$ は

$$\begin{bmatrix} \mathbf{z}_{ij} \\ 1 \end{bmatrix} \propto \mathbf{P}_i \begin{bmatrix} \mathbf{q}_j \\ 1 \end{bmatrix} \quad (2)$$

のように与えられる³⁾。ただし \propto は両辺のベクトルが長さを除いて等しいことを表す。

(1) 式のようにカメラ行列 \mathbf{P}_i を構成するパラメータのうち, \mathbf{K}_i はすべて事前に分かっているか、あるいはその一部のみ未知 (焦点距離のみを未知とする自己校正 (self-calibration))

が実用上便利⁹⁾) であるとし, \mathbf{R}_i および \mathbf{t}_i はすべて未知である^{*1}。これら未知のもの表現する適切なパラメータを選び^{*2}, それらを並べてベクトルを \mathbf{p}_i を作る。目標は、カメラの未知パラメータ $\mathbf{p}_1, \dots, \mathbf{p}_m$ および特徴点の3次元座標 $\mathbf{q}_1, \dots, \mathbf{q}_n$ を, (2) 式の関係に基づいて推定することである。観測 $\mathbf{z}_{ij} = [u_{ij}, v_{ij}]^T$ には、画像を取得し各点を抽出 (画像座標を位置決め) する過程で誤差が混入する。そのため \mathbf{z}_{ij} は, (2) 式を満たさなくなる。そのような誤差の存在を前提として、可能な限り精度良く推定を行いたい。

仮にすべてのパラメータ ($\mathbf{p}_1, \dots, \mathbf{p}_m, \mathbf{q}_1, \dots, \mathbf{q}_n$) の真の値が分かっているとしたり、これを (2) 式に代入すれば、各特徴点の画像上の位置を (誤差を除いて) 「再現」できる。逆に、その再現した位置が、実際に観測した位置と近くなるようにパラメータを決めれば、パラメータの良い推定値が得られそうである。バンドル調整は、このような考え方の下でコスト関数

$$E(\mathbf{p}_1, \dots, \mathbf{p}_m, \mathbf{q}_1, \dots, \mathbf{q}_n) = \frac{1}{2} \sum_{i,j} (u_{ij} - u(\mathbf{p}_i, \mathbf{q}_j))^2 + (v_{ij} - v(\mathbf{p}_i, \mathbf{q}_j))^2 \quad (3)$$

を最小化する。ただし, $u(\mathbf{p}_i, \mathbf{q}_j)$, $v(\mathbf{p}_i, \mathbf{q}_j)$ は、次のように、パラメータを指定したときの特徴点の画像座標である:

$$u(\mathbf{p}_i, \mathbf{q}_j) = \frac{(\mathbf{P}_i)_1 [\mathbf{q}_j^T \ 1]^T}{(\mathbf{P}_i)_3 [\mathbf{q}_j^T \ 1]^T}, \quad v(\mathbf{p}_i, \mathbf{q}_j) = \frac{(\mathbf{P}_i)_2 [\mathbf{q}_j^T \ 1]^T}{(\mathbf{P}_i)_3 [\mathbf{q}_j^T \ 1]^T}. \quad (4)$$

ここで $(\mathbf{P}_i)_k$ は \mathbf{P}_i の k 行ベクトルを表す。(3) 式で定義されるコスト関数 E のことを再投影誤差 (reprojection error) と呼ぶ。

すべての未知パラメータを並べてできるベクトルを \mathbf{x} と書けば, (3) 式の再投影誤差は \mathbf{x} の関数

$$E(\mathbf{x}) = E(\mathbf{p}_1, \dots, \mathbf{p}_m, \mathbf{q}_1, \dots, \mathbf{q}_n) \quad (5)$$

である。 E は \mathbf{x} の非線形関数なので、その最小値を与える \mathbf{x} を計算するには、反復に基づく数値解法を用いることになる。つまり、適当な初期値 \mathbf{x} から出発し、微小な更新 $\mathbf{x} \rightarrow \mathbf{x} + \delta \mathbf{x}$

*1 もちろん、どんな未知数でも定められるわけではない。どれを定めることができるかは、通常は解析的に明らかにする。例えば、カメラの内部パラメータがすべて分からない状況では、これらを、点の位置とカメラの位置・姿勢とともに定めるのは不可能である (射影復元しかできない) が、いくつかのパラメータが分かれば、それ (メトリック復元) が可能であることが示されている³⁾。ただし問題によっては解析が困難で、実験的に確かめることも少なくない。

*2 中でも注意を要する回転について、その表現の方法を付録 A.1 に記した。

を反復して、最小解を数値的に探索し*1, 得た解がパラメータの推定値となる。

以上、一つの典型的な問題に対し、バンドル調整がどのように使用されるかを示した。以下の節では、具体的な実装方法や背景にある理論等を述べてゆく。以下にその内容を、本稿の構成とともに示す。

再投影誤差 $E(\mathbf{x})$ の最小化は、非線形関数最小化のための汎用のアルゴリズムを用いて行う。収束性能の高さから、ニュートン法—特に E が二乗和であることから、これを最小二乗問題用に修正したもの—を用いる。ニュートン法および派生アルゴリズムは、良い収束性能 (= 反復回数が少ない) を持ち、反復 1 回の計算コストは大きいものの、一般にトータルの計算量は小さい。ただし、問題の規模が極端に大きい場合には、反復 1 回の計算コストのより小さい非ニュートン法 (共役勾配法など) の使用も検討に値する。3 節では、ニュートン法およびその実装について述べる。既存のソフトウェアパッケージやライブラリを使う場合も多いと考えられるが、それらについてもまとめた。

$E(\mathbf{x})$ を最小化することは、観測誤差についてあるモデルを過程したときの最尤推定に相当する。統計学的な背景、および推定結果の精度評価 (漸近分散推定) などの話題を 4 節に述べる。

カメラや空間の点が与える幾何は、座標系を定めることによって初めて記述される。この座標系の選定には通常任意性があり、計算を実装するためにはこの任意性を取り除く (つまり具体的な座標系を 1 つ選ぶ) 必要がある。任意であるということは、基本的にはどれを選んでも良いということでもあるが、選択いかんで、推定結果の精度が見かけ上大きく変わる。このことおよび周辺の議論を 5 節に示す。

バンドル調整は元々、すべての観測を用いて関連する全パラメータを同時に推定する方法である。しかしながら、カメラの数や点の数が増加するにつれ、上述の最小化問題の規模が大きくなり、手に余る場合が出てくる。現実には、一部のパラメータを推定できれば良い (例えば、ナビゲーションが目的であればカメラの運動のみ、形状復元が目的であれば点の位置のみ) 場合も多い。関連して、時々刻々もたらされる観測に対し、オンラインでパラメータを推定したい場合もある。このような場合に記憶域や計算量を小さくする方法および、若干の精度を犠牲にしてオンライン推定を可能にする再帰的計算法を 6 節に述べる。

*1 初期値は極めて重要である。多くの場合、この初期値をどれだけ最小解に近いものとするかは、探索の過程で局所解に陥らず、無事大域解を探し当てられるかどうかを左右する。多くの場合、DLT (Direct Linear Transformation) 法³⁾ など、誤差を考慮しない線形計算を用いてそれなりの精度の推定値を得て初期値とするが、すべての問題に適用可能な系統的な方法は存在しない。本稿では、初期値の計算については触れない。

3. 最小化のための数値計算

バンドル調整の中核をなすのは、再投影誤差 $E(\mathbf{x})$ の数値最小化である。基本的には、非線形最小化を行うためのものならば、どのような汎用アルゴリズムもこの目的のために使うことができるが、以下の (6) 式のように $E(\mathbf{x})$ は通常、二乗和の形をとるので、最小二乗アルゴリズムを使うのが一般的である。

$$E(\mathbf{x}) = \frac{1}{2} \|\mathbf{e}(\mathbf{x})\|^2 = \frac{1}{2} \sum_k e_k^2 \quad (6)$$

本節最後にまとめたように、以上の計算を行うための既成の数値計算ライブラリはいくつか存在する。ごく標準的なバンドル調整しか行わないのであれば、可能ならばそれらの利用が推奨される。バンドル調整の対象となる多くの問題で、疎 (sparse) な行列を扱うことになる。問題の規模が一定以上の大きさを持つときは、疎行列を適切に取り扱えることが必要である。その場合は、疎行列に対応したライブラリを使う必要がある。

ただし、単に行列が疎であることを利用するだけでなく、行列の構造自体を利用して、計算量と記憶域について最適化しようとするれば、数値計算ライブラリに頼ることはできず、自分で実装する必要が生じる。6 節に述べる再帰的計算を実現する場合も同じである。

このような観点から、本節では最も基礎的な事柄を述べるとともに、発展的な場合への導入を簡潔にまとめる。

3.1 最小二乗のためのニュートン法

まず最初に、ニュートン (Newton) 法に基づく非線形最小二乗の方法を述べる。 $E(\mathbf{x})$ の 2 次微分 (あるいはその近似) を直接求め、利用するのがこの方法の特徴である。詳しい説明は数値計算の教科書^{11),12)} などに譲り、ここでは方法の概要のみを示す。

3.1.1 ガウス・ニュートン法

非線形方程式の解を数値的に見つける一般的な方法がニュートン法である。今、目標は $E(\mathbf{x})$ を最小化する \mathbf{x} を見つけることであるが、そのような \mathbf{x} では $dE/d\mathbf{x} = \mathbf{0}$ となることから、方程式 $dE/d\mathbf{x} = \mathbf{0}$ を \mathbf{x} についてニュートン法で解くことを考える。

ニュートン法は、現在の推定値 \mathbf{x} を、適切に定めた更新量 $\delta\mathbf{x}$ を使って $\mathbf{x} \rightarrow \mathbf{x} + \delta\mathbf{x}$ のように更新し、これを収束するまで何度か反復することで、解を見出そうとする方法である。

現在の推定値 \mathbf{x} まわりでの $E(\mathbf{x})$ をテイラー展開を

$$E(\mathbf{x} + \delta\mathbf{x}) \approx E(\mathbf{x}) + \mathbf{g}^T \delta\mathbf{x} + \frac{1}{2} \delta\mathbf{x}^T \mathbf{H} \delta\mathbf{x} \quad (7)$$

と書く。ただし \mathbf{g} は E の \mathbf{x} での勾配, \mathbf{H} は \mathbf{x} でのヘッセ行列である。

$$\mathbf{g} = \left. \frac{dE}{d\mathbf{x}} \right|_{\mathbf{x}}, \quad \mathbf{H} = \left. \frac{d^2E}{d\mathbf{x}^2} \right|_{\mathbf{x}} \quad (8)$$

\mathbf{g} は \mathbf{x} とは同じ長さのベクトルで, \mathbf{H} は行数, 列数が \mathbf{x} の長さに等しい正方行列である。

\mathbf{x} を定数, $\delta\mathbf{x}$ のみ変数と考えると, (7) 式右辺の最小値を与える $\delta\mathbf{x}$ は, 次を満たすようなものである。

$$\mathbf{H} \delta\mathbf{x} = -\mathbf{g} \quad (9)$$

つまり (\mathbf{H} に逆行列があるとすれば) $\delta\mathbf{x} = -\mathbf{H}^{-1} \mathbf{g}$ である。もし \mathbf{x} が解に十分近ければ, \mathbf{H} は正定値 (positive-definite) すなわち固有値がすべて正となる。そうであれば, 上の更新を反復することで急速に解に近づくことが保証される。

ただし \mathbf{x} が解から離れているときは, \mathbf{H} が正定値である保証はない。さらに, \mathbf{H} を計算する (特にそのためのコードを書く) のはそれなりに大変である。ガウス・ニュートン (Gauss-Newton) 法はこれらの問題を回避するため, (6) 式のように $E(\mathbf{x})$ が書けることを利用し, \mathbf{H} を次のように近似する:

$$\mathbf{H} \approx \mathbf{A} \equiv \mathbf{J}^T \mathbf{J} \quad (10)$$

ただし, \mathbf{J} は $\mathbf{e}(\mathbf{x})$ のヤコビ行列

$$\mathbf{J} = \frac{d\mathbf{e}}{d\mathbf{x}} \quad (11)$$

である。ヤコビ行列を使うと, (9) 式の右边を $\mathbf{a} \equiv -\mathbf{g} = -\mathbf{J}^T \mathbf{e}$ と書ける。これら \mathbf{A} , \mathbf{a} を用いて, $\delta\mathbf{x}$ を定める (9) 式を

$$\mathbf{A} \delta\mathbf{x} = \mathbf{a} \quad (12)$$

と書き換える。この式による解の更新を繰り返すのがガウス・ニュートン法である。なお, 鍵となる近似 $\mathbf{H} \approx \mathbf{A}$ の精度が高くなるのは, 最小解付近で $\|\mathbf{e}\|$ が小さい場合や, あるいは \mathbf{e} の 2 階微分の大きさが小さい場合などである。

3.1.2 レベンバーグ・マーカート法

レベンバーグ・マーカート (Levenberg-Marquardt) 法は, 上のガウス・ニュートン法の更新式 (12) を, さらに次のように修正したものである。

$$(\mathbf{A} + \lambda \mathbf{I}) \delta\mathbf{x} = \mathbf{a} \quad (13)$$

λ はダンピングファクタ (damping factor) などと呼ばれ, 最小化の進み具合によって解の更新のたびに変更 (ただし常に $\lambda \geq 0$) される。狙いは, ガウス・ニュートン法の大域収束性の改善である。新しい更新式は, $\lambda = 0$ のときにガウス・ニュートン法のそれと一致し, λ が大なるときに最急降下法 (steepest descent 法) に一致するから, 両者の長所を引き出すとする方法と言える。

λ の選択方法は色々考えられるが, 例えば次のようにする。適当な値 (例えば $\lambda = 0.001$) から始め, (13) 式を解いて $\delta\mathbf{x}$ を求める。 $E(\mathbf{x} + \delta\mathbf{x}) \geq E(\mathbf{x})$ なら λ を 10 倍し, 再度 (13) 式を解き直す。 $E(\mathbf{x} + \delta\mathbf{x}) < E(\mathbf{x})$ なら $\mathbf{x} \leftarrow \mathbf{x} + \delta\mathbf{x}$ と実際に更新し, λ を 0.1 倍する。以上を反復する。つまり, $E(\mathbf{x})$ が確実に減少するまで λ を大きくし続け, 減少するときに解を更新し, 同時に λ を小さくする。ただし上はほとんど説明のためだけのもので, 実際にはもっと洗練された方法が生み出されている。

このように λ を修正し, $\delta\mathbf{x}$ のステップ量を制御するのは, 最小二乗に留まらないより一般の関数の最小化のための方法である Trust-region 法を, ガウス・ニュートン法にあてはめたものと見なせる (なお, Trust-region 法とでは, $\delta\mathbf{x}$ の方向の決定と, ステップ幅 $\|\delta\mathbf{x}\|$ の決定の順番が異なる。)

解に十分近いところでは, 総じてガウス・ニュートン法の方が効率が良いが, レベンバーグ・マーカート法は大域収束性能において優れるため, 非線形最小二乗アルゴリズムのスタンダードである。

3.2 ニュートン法以外の方法

レベンバーグ・マーカート法などニュートン法の派生アルゴリズムは, 特に解の周辺領域での収束性能が優れているので, 標準的に使用すべきである。問題が大規模あるいは疎でないために, ヘッセ行列の分解計算のコストが負担できない場合のみ, それ以外の手法, 例えば共役勾配法の使用を検討すればよい。ただし次節に述べるように, ニュートン法の計算コストは, ヘッセ行列が疎となることやその行列自体の構造を利用すれば引き下げることができる。それでもまだ計算コストが大きければ, 6 節に述べるような更新や再帰的計算といった方法を使うことで, わずかな精度の犠牲で計算量を大きく引き下げることが可能である。

したがって、ニュートン法系のアルゴリズムを使わない方が良い場合は限られると考えるよ
 かる。

3.3 実装方法

3.3.1 更新量の計算

上述の2つの方法はいずれも、 $A\delta x = a$ の形の線形方程式を解いて更新量 δx を求める。
 問題が大規模になるとき A のサイズが大きくなるが、その場合に方程式をいかに効率良く、
 また精度良く解くかが、主要な課題となる。

線形方程式を解くには、 A が一般の行列ならばガウスの消去法か LU 分解を使うが、今の
 場合 A は対称かつ正定値なので、コレスキー分解が使える。LU 分解の倍速く、ピボット選
 択を行わなくても安定である長所を持つことから、標準的方法となっている。具体的には A
 のコレスキー分解を使って、次のように $A\delta x = a$ を解く。

1. A をコレスキー分解する。 L は、対角成分が正の下三角行列である。

$$A \rightarrow LL^T \quad (14)$$

2. $Ly = a$ を y について解く（前進代入によって計算）。
3. $L^T\delta x = y$ を δx について解く（後退代入によって計算）。

なお、代数的には、逆行列 A^{-1} を使えば解は $\delta x = A^{-1}a$ のように書けるから、 A^{-1} を数
 値計算しても計算できるが、計算量および精度の両面で上記手法に劣るので、通常行われな
 い（もちろん、逆行列自体が欲しい場合（後述の推定量の分散推定）は計算することにな
 るが。）また、コレスキー分解の代わりにガウスの消去法を使うのは悪くないが、若干遅く
 なることと、観測の追加に伴う A の逐次更新（6節で述べる）のための有効な方法がない。
 また、 A が半正定値のとき（つまり0固有値を持つとき）は、ピボット選択付きコレスキー
 分解を行う。これは冗長なパラメータ表現を行う場合に必要となる（5節参照）。

なお、 $A\delta x = a$ を解く代わりに、等価な線形最小二乗問題（ $\|J\delta x + e\|^2 \rightarrow \min.$ ）を解い
 ても良い。この場合には $A = J^T J$ を計算する必要がない。レベンバーグ・マーカート法
 の場合（ $(A + \lambda I)\delta x = a$ ）は、代わりに

$$\left\| \begin{bmatrix} J \\ \lambda^{1/2} I \end{bmatrix} \delta x + \begin{bmatrix} e \\ 0 \end{bmatrix} \right\|^2 \quad (15)$$

を最小化すればよい。これらは優決定線型方程式の解の計算であり、標準的には行列の QR
 分解（ハウスホルダー（Householder）変換およびギブンス（Givens）回転）と、後退代入

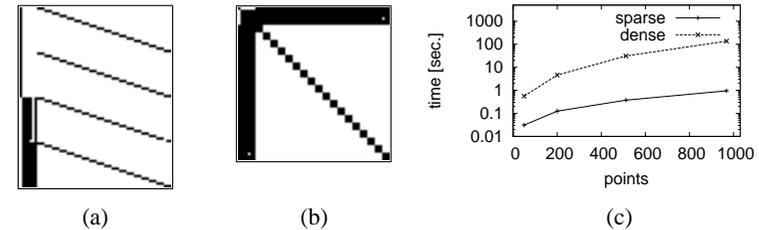


図1 例2.1でカメラ2台、特徴点数200点とした場合の、(a) ヤコビ行列 J と、(b) ヘッセ行列 H の非零成分（黒）
 と零成分（白）。(c) H を疎行列とした場合と故意に密な行列とした場合との間の、計算時間の比較（ガウス
 ニュートン法を使用し、収束までの反復回数はすべて4回）。

の組み合わせによって実行する。

3.3.2 疎行列の取り扱い

バンドル調整が対象とする多くの問題では、ヤコビ行列 J やヘッセ行列 H およびその近
 似 A は普通、疎行列（sparse matrix）となる。つまり、これら行列の多くの成分は0であり、
 限られた数の成分のみ0でない値が入る。 J の (i, j) 成分は、 e の成分を x の成分で微分し
 た $\partial e_i / \partial x_j$ であるが、 e_i が x_j に依存していなければ、この成分は永久に0である。例え
 ば SFM では、一つの成分 e_i は、ある空間の点の、ある画像上の像の（例えば） x 座標の誤
 差にあたる。 e_i は明らかにこの点以外の点の空間座標には依存しないし、この画像に関連
 しないカメラの姿勢にも依存しない。つまり、それらパラメータ x_k に対する $\partial e_i / \partial x_k \equiv 0$
 である。例2.1でカメラの台数を2としたときの、 J と H の非零成分の分布の様子を、図1
 に示す（両カメラの焦点距離を除き内部パラメータの既知な基礎行列を推定しているのと
 同じ。）行列の大部分が0で、疎行列であることが分かる。

方程式を解いて更新量を計算する際、このように行列が疎であることを利用すれば、効率
 は大幅に向上する。行列が疎であることを利用する方法は、2段階に分けられる。

第1段階は、当然だが、疎行列の非零成分のみを記憶し、また計算で利用することであ
 る。その効果はあまりにも大きく、ごく小規模の問題でない限り、そうすべきである。疎行
 列を扱える数値計算ライブラリを使える環境にあれば、すぐに実現できる。上の例で、 H を
 疎行列として適切に扱った場合とわざと密な行列として扱った場合とで、計算時間がどのく
 らい変わるかを図1(c)に示す。点の数が数百のとき、2桁の速度差があり、点の数が増える
 にしたがってこの差が広がることが見て取れる。

行列が疎であることを利用する2番目の段階は、計算途中および最終的に得られる行列

がなるべく疎行列となるように、行列の置換 (permutation, 行と列の入れ替え) を行うことである。例えばコレスキー分解の場合、 A を分解するよりも、適当な置換行列 P で置換した PAP^T を分解する方が、計算途中および結果 LL^T の行列がより疎になり得る。実際、この種の行列分解や逆行列の計算では、元の行列が疎であっても、計算途中で非零成分が新たに発生する (fill-in) ことが急速に起こるので、対策の効果は大きい。

ただし、行列の最適な置換を求めるのは NP 問題であることが分かっているため、準最適なものをヒューリスティックな方法により求めることになる。バンドル調整の場合、対象とする問題の多くで、 J や H が (図 1 に示したように) 特別な構造 (ブロック対角性や “arrow-head” 構造など) を持ち、これを活用する方法が考えられている。これらについては、Triggs らの解説¹⁰⁾ に説明およびサーベイがある。なお、 A の行・列を置換するとは、パラメータをベクトル x に並べる順番を変更するのと同じであり、つまりその並べ方が効率を変化させるということである。なお、汎用の疎行列計算ライブラリの中には、fill-in を抑えるように事前に行列の置換を行うものもあり、これらを使えばユーザ側で何もしなくともまずまずの結果を得られる。

3.3.3 シューア補行列と線形性の利用

SFM では、パラメータはカメラのパラメータと点の空間座標の 2 種類からなる。このようにパラメータ x を x_1 と x_2 の 2 つに分けたとき、 $\partial E / \partial x = 0$ の解を $x + \delta x \rightarrow x$ の更新を反復して探索する代わりに、まず x_1 を固定し $\partial E / \partial x_2 = 0$ の解 x_2 を計算し、次に x_2 を固定し $\partial E / \partial x_1 = 0$ の解 x_1 を計算し、これを交互に繰り返すことが考えられる。1 回の反復における計算の規模を小さく抑えられる利点があり、時折利用される (resection-intersection と呼ばれる)。ただし、交互に反復すること自体に収束性の保証はなく、反復回数が増加してしまうことも多く、必ずしも有効な方法とは言えない。

これに対して、上述の 2 つの方程式のうち、どちらか一方について解の計算が線形に行える場合は有効な計算法がある。例えば、 x_1 を固定したときの $\partial E / \partial x_2 = 0$ が、 $F(x_1)x_2 = 0$ のように x_2 の線型方程式となる場合である。この場合、その解 x_2 は (反復計算なしに) 直接与えられる。そこで、その解を x_1 の関数と見て $x_2 = \hat{x}_2(x_1)$ と書けば、これを E に代入することで、 x_1 だけの関数 $E'(x_1) \equiv E(x_1, \hat{x}_2(x_1))$ を得る。そして $E'(x_1)$ の最小解をニュートン法によって求めれば、 x をニュートン法で求める場合よりも問題の規模が小さくなる。非観測成分を含む場合の行列分解*1 に対する Wiberg のアルゴリズムはこの方法に

*1 観測からなる行列を、ランクの小さな 2 つの行列の積に $Z \rightarrow MS$ と分解する問題において、 Z の成分の一部が

基づく⁷⁾。

同じ考え方は、上の 2 つの方程式のいずれもその解を線形に計算できなくても、有効である。 $x + \delta x \rightarrow x$ の更新量を求める方程式 $A\delta x = a$ を、次のようにブロック分割する。

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (16)$$

上は、2 つの方程式 $A_{11}\delta x_1 + A_{12}\delta x_2 = a_1$ および $A_{21}\delta x_1 + A_{22}\delta x_2 = a_2$ からなる。 δx_1 を固定し後者を解けば、 $\delta x_2 = A_{22}^{-1}(-A_{21}\delta x_1 + a_2)$ となる。これを前者に代入すれば、 δx_1 のみの方程式を得る。つまり

$$(A_{11} - A_{12}A_{22}^{-1}A_{21})\delta x_1 = (I - A_{12}A_{22}^{-1}A_{21})a_2 \quad (17a)$$

$$A_{22}\delta x_2 = a_2 - A_{21}\delta x_1 \quad (17b)$$

である。この 2 式を順に計算すれば、更新量 $\delta x = [\delta x_1^T, \delta x_2^T]^T$ を得る。なお、 $A_{11} - A_{12}A_{22}^{-1}A_{21}$ は A_{11} のシューア補行列 (Schur's complement) と呼び、 A の逆行列 A^{-1} の、 A_{11} の位置のブロック行列はこの逆行列に一致する性質がある。

x_1 が短く x_2 が長いとき、上の方法を採用すると一般に計算量を低減できる。SFM では、カメラの台数が少なく、点の数が多いたがこれにあたり、図 2 の場合が良い例である。図に見られるように、上の行列分割で言う A_{22} ブロックは、点のパラメータ部分にあたり、それ自身がブロック対角行列となる。したがって A_{22} の逆行列は、ブロック小行列の逆行列を求めるだけの少ない計算量で求まる。カメラの台数が多く、点の数が少ないときは、入れ替えれば同じである。

3.4 数値計算ライブラリを利用

ここまで述べてきた数値計算の各アルゴリズムは、そのほとんどがソフトウェアとして実装されている。それらは多様なライセンス条件の下、C/C++ や Fortran などの言語から利用可能なライブラリとして提供される。これらを以下にごく簡潔にまとめる (なおこの内容はすぐに古いものとなり得るので、利用にあたっては最新の情報を調べられたい。)

まず、MATLAB および Optimization Toolbox を利用する場合、話は簡単である。非線形最小二乗のための関数 lsqnonlin を利用する。Trust-region 法、レベンバーグ・マーカート法、およびガウス・ニュートン法を選択できる (デフォルトは Trust-region 法で、オプションに

与えられない場合、 Z が完全ならば、特異値分解等により常に解を計算できるが、この場合は非線形最小二乗問題となる。

よって残りの2つを選択)。疎行列も簡単に扱える。ユーザ側では、 e およびヤコビ行列 J を計算する関数を用意する。ヤコビ行列 J は、ライブラリ側に有限差分近似にて自動的に計算させることも可能だが、疎行列の効果を得るにはユーザ側で指定することが必須となる。

MATLAB を使わない場合はいくつかの選択肢がある。

まず、既存の数値計算のソフトウェアはいくつかの階層に分かれて構成されている。基本線形演算のライブラリである BLAS が最下層に位置し、後述のライブラリで利用される。BLAS は、正確にはライブラリ API で、CPU のアーキテクチャごとに最適化されたライブラリが存在する。一般的な PC でも、CPU に搭載されるベクトル演算機能 (SIMD) を使うため、最適化されたものの利用が推奨される。Intel のものや GotoBLAS²⁾ 等が良く知られているようである。

コレスキー分解や QR 分解等の行列分解を行うライブラリがあれば、ガウス・ニュートン法はもちろん、レベンバーグ・マーカート法も手法それ自体の実装は容易であるので、そうするのも方法である。行列分解を行うソフトウェアは以下のように数多い。

LAPACK は、BLAS の上に構成された線形代数演算ライブラリのスタンダードである。コレスキー分解等の基本行列分解や線形方程式ソルバーを含む。ただし疎行列には対応しないので、用途は限定されよう。疎な対称正定行列のコレスキー分解は、CHOLMOD^{*1} が計算速度および精度の面で実績があり、MATLAB 7.2 以降で採用されている^{*2}。疎な非対称行列の分解ならば UMFPACK^{*3} に定評があり、やはり MATLAB に採用されている。また、疎行列のための線形方程式ソルバーに、PARDISO^{*4} があり、これは Intel の Math Kernel Library に採用されている。これらソフトウェアでは、疎行列の fill-in の増加を抑えるための行列の置換が実装されている^{*5}。なお、Math Kernel Library を導入すれば、最適化された BLAS も導入され、便利である。

よりハイレベルな実装としては、最適化アルゴリズムのライブラリである MINPACK に、レベンバーグ・マーカート法の実装 (lmdif, lmdcr) がある。ただし疎行列には対応しないので、用途は限定される。また、コンピュータビジョンの分野の著者によって作られたソフトウェアに sba/lemvar がある^{*6}。sba は、SFM のための疎行列を前提としたバンドル調整

である。目的が平凡な SFM であるならば、これを使うのが簡単かもしれない(ただし自由度はあまり高くない)。同じ著者がレベンバーグ・マーカート法を実装した lemvar を公開している。これは疎行列に対応しており、一般のバンドル調整に利用可能かもしれない。

4. バンドル調整と統計的推測

画像上で抽出した特徴点などの観測には、必ず誤差が伴う。この誤差の存在の下でパラメータをいかに精度良く推定するかに関心がある。観測の誤差を確率的にモデル化することにより、この問題を統計的推測の枠組みで捉えることができる。これにより、誤差モデルの正しさの範囲内で、推定精度の理論上の限界と、それを達成するための方法論がもたらされる。本節では、バンドル調整の統計的推測の側面を述べる。

4.1 観測誤差のモデル

特徴点の座標の観測値 (u, v) は、画像から何らかの方法で抽出する。画像間のマッチングでは例えば SIFT などが、動画像における点のトラッキングでは KLT が良く使われる。いかなる手法を用いても、画像上の点の位置を決める際、一定の誤差の生じることは避けられない。これには量子化誤差も含む。

誤差は通常予測不可能であり、故にランダムな確率変数と考えられる。さらに誤差は加算的、すなわち観測は、真の値に誤差が加算されたものとするのが一般的である。つまり、ある点の座標について、観測値を $\mathbf{z}_i = [u_i, v_i]^T$ 、その真の値を $\bar{\mathbf{z}}_i = [\bar{u}_i, \bar{v}_i]^T$ と書くとき

$$u_i = \bar{u}_i + \varepsilon \quad (18a)$$

$$v_i = \bar{v}_i + \varepsilon' \quad (18b)$$

と表す。 $\varepsilon, \varepsilon'$ が誤差である。多くの場合、異なる観測間、つまり (u_i, v_i) と (u_j, v_j) の間では、誤差は互いに独立であると考えてよい。

そしてこれら誤差 $\varepsilon, \varepsilon'$ が、ある確率分布に従うと仮定する。観測間では独立だが、共通の分布に従うとすることが多い。最も良く仮定される分布は、平均 0 の正規分布 (ガウス分布) である。中でも一番簡単なのは、 ε と ε' それぞれが、独立に平均 0、標準偏差 σ の正規分布 $N(0, \sigma^2)$ に従うとする場合である。2 次元正規分布を用いれば、 $[\varepsilon, \varepsilon']^T$ が $N(\mathbf{0}, \sigma^2 \mathbf{I})$ に従うと表現される。 ε と ε' が独立でないなどより複雑な場合は、 2×2 行列 \mathbf{V} を共分散とする正規分布 $N(\mathbf{0}, \mathbf{V})$ に従うと考える。

なお、誤差の分布が本当に正規分布であるかどうかは (後述する外れ値の問題を別にしても) 一考の余地がある。理想的には、観測の誤差がどのような物理的なメカニズムにより

*1 <http://www.cise.ufl.edu/research/sparse/cholmod/>

*2 線形方程式の解を計算する演算子 “\” の中身がこれである。

*3 <http://www.cise.ufl.edu/research/sparse/umfpack/>

*4 <http://www.pardiso-project.org/>

*5 Minimum degree アルゴリズムや Nested dissection アルゴリズムによる行列置換

*6 <http://www.ics.forth.gr/~lourakis/sba/>

生じるかを調べて明らかにすべきである。しかしながら、これには、カメラの光学系や撮像系の仕組みや、SIFT や KLT 等のアルゴリズムなど多くの要素が関与し、厳密な解析は難しい。正規分布を仮定する一つの根拠は、最終的な誤差が、このような多くの要素において生じる誤差の蓄積と見なせることである。

また、SIFT や KLT などの方法により画像から点を抽出し画像間での対応付けを行うと、対応付けに失敗する場合が少なからずある。そのような場合を上の誤差モデルで表現するのは不適當であり、このときの観測は外れ値 (outlier) と呼ばれる。観測に外れ値が存在するにも関わらず上の誤差モデルに基づいて統計的推測を行うと、結果はほとんど役に立たなくな。これに対処する方法は 2 通りある。一つは外れ値も表現し得るように誤差モデルを拡張することであり、もう一つは事前に外れ値を観測の中で同定しそれらを取り除く方法である。前者の場合、誤差の分布を正規分布ではなくコーシー (Cauchy) 分布で表現したり、あるいは正規分布と一様分布の混合分布として表すなどする。この方法は、問題が一般に非線形最小二乗でなくなるから、数値計算がより難しくなる。後者は、問題が多視点幾何の場合、エピソード条件などの多重線形拘束 (multi-linear constraint) を用いて行う。前者よりも後者が一般的だろう。

4.2 幾何学推定問題の構造

コンピュータビジョンにおける幾何学的なパラメータの推定問題の多くは、共通の構造を持っている。具体的には、1 枚の画像上での直線や楕円の推定から、複数画像間の基礎行列や 3 重焦点テンソル (trifocal tensor) の推定、そして例 2.1 のような SFM などである。これらは共通して次のように定式化される。

問題 4.1 (陽的拘束). 複数の観測 \mathbf{z}_i ($i = 1, \dots, n$) が与えられたとする。1 つの観測 \mathbf{z}_i は、その (未知なる) 真の値 $\bar{\mathbf{z}}_i$ に誤差 $\boldsymbol{\varepsilon}_i$ が加算されたものとする:

$$\mathbf{z}_i = \bar{\mathbf{z}}_i + \boldsymbol{\varepsilon}_i \quad (19)$$

誤差 $\boldsymbol{\varepsilon}_i$ は、各観測間 ($i = 1, \dots, n$) で独立だが、すべて共通して平均 0 のガウス分布 $N(\mathbf{0}, \mathbf{V})$ に従うとする。一方、真の値 $\bar{\mathbf{z}}_i$ は、観測 i ごとに未知パラメータ $\boldsymbol{\xi}_i$ と $\boldsymbol{\theta}$ によって次のように与えられる。

$$\bar{\mathbf{z}}_i = \mathbf{f}(\boldsymbol{\xi}_i, \boldsymbol{\theta}), \quad i = 1, \dots, n \quad (20)$$

このとき、 $\boldsymbol{\theta}$ および $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n$ を推定したい。

問題によっては、真の値 $\bar{\mathbf{z}}_i$ が陰的に拘束されると考えたほうが自然な場合があり、この場合は次のように定式化される。

問題 4.2 (陰的拘束). 複数の観測 \mathbf{z}_i ($i = 1, \dots, n$) が与えられ、1 つの観測 \mathbf{z}_i は、(19) 式のように真の値 $\bar{\mathbf{z}}_i$ に誤差 $\boldsymbol{\varepsilon}$ が加算されたものとする。真の値 $\bar{\mathbf{z}}_i$ は、ある式

$$\mathbf{f}_c(\bar{\mathbf{z}}_i, \boldsymbol{\theta}) = \mathbf{0}, \quad i = 1, \dots, n. \quad (21)$$

を満たす $\boldsymbol{\theta}$ を推定したい。

問題 4.1 の (20) 式で $\boldsymbol{\theta}$ を固定したまま $\boldsymbol{\xi}_i$ を自由に動かせば、同式が与える $\bar{\mathbf{z}}_i$ は、 d 次元空間 (d はベクトル \mathbf{z}_i の長さ) の部分多様体を与える。問題 4.2 では、(21) 式が $\bar{\mathbf{z}}_i$ を拘束し、結果的に同じような d 次元空間の部分多様体を与える。つまりどちらの場合も問題の構造は基本的に同じである。実際、問題 4.2 のように表現される場合も、観測 \mathbf{z}_i ごとにパラメータ $\boldsymbol{\xi}_i$ を導入して問題 4.1 の形に書き換えることが可能である。

われわれの関心ある問題は上のように定式化されるが、一つの特徴は、1 つの観測 \mathbf{z}_i につき 1 つのパラメータ $\boldsymbol{\xi}_i$ が存在することである (問題 4.2 では $\bar{\mathbf{z}}_i$ が観測ごとの未知パラメータとなる)。この結果、観測の数に比例して未知パラメータの数が増えることになる。特に問題 4.2 のように定式化する場合に、 $\boldsymbol{\theta}$ の推定には関心があるが、 $\bar{\mathbf{z}}_i$ や $\boldsymbol{\xi}_i$ の推定には関心がないという場合が時々ある。定式化のために導入せざるを得ないパラメータであって、推定の邪魔になるという意味で、 $\bar{\mathbf{z}}_i$ や $\boldsymbol{\xi}_i$ のようなパラメータをかく乱母数 (nuisance parameter あるいは latent parameter) と呼ぶことがある。今の場合のように、このようなパラメータが観測の数とともに増加するとき、後述のようにその存在は推定精度の議論に大きく関与する。

さて、実際の問題が上の定式化にどのようにあてはまるかを以下に示す。

例 4.1 (直線推定). 1 枚の画像上、 n 個の点 (u_i, v_i) ($i = 1, \dots, n$) が与えられたとする。これらの点は、1 本の直線 $v = au + b$ 上の点 (\bar{u}_i, \bar{v}_i) を観測したもので、観測と真の値は

$$u_i = \bar{u}_i + \varepsilon_i \quad (22a)$$

$$v_i = \bar{v}_i + \varepsilon'_i \quad (22b)$$

のような関係があり、 $\varepsilon_i, \varepsilon'_i$ はそれぞれ独立に $N(0, \sigma^2)$ に従うとする。真の位置 (\bar{u}_i, \bar{v}_i) は直線上にあるので、 $\bar{v}_i = a\bar{u}_i + b$ である。これは (21) 式の陰的拘束にあたる。新たなパラ

メータ ξ_i を導入すると, $[\bar{u}_i, \bar{v}_i]^T = [\xi_i, a\xi_i + b]^T$ と書ける. 右辺を $\mathbf{f}(\xi_i, [a, b]^T)$ とおけば (20) 式の陽的拘束を得る.

例 4.2(楕円推定). 画像上の単純な楕円 $C: u^2/a^2 + v^2/b^2 = 1$ を考える. n 個の点 (u_i, v_i) ($i = 1, \dots, n$) が与えられ, これらは, C 上の点 (\bar{u}_i, \bar{v}_i) を観測したもので, 直線と同様に (22) 式で与えられ, $\varepsilon_i, \varepsilon'_i$ はそれぞれ独立に $N(0, \sigma^2)$ に従うとする. 真の位置 (\bar{u}_i, \bar{v}_i) は C 上にあるから, $\bar{u}_i^2/a^2 + \bar{v}_i^2/b^2 = 1$ を満たし, これが (21) 式の陰的拘束にあたる. また, 新たなパラメータ ξ_i を導入すると, $[\bar{u}_i, \bar{v}_i]^T = [a \cos \xi_i, b \sin \xi_i]^T$ のように書け, こちらは (20) 式の陽的拘束に相当する.

例 4.3(画像間の 2次元射影変換). シーンにある 1 枚の平面を異なるカメラで撮影した 2 枚の画像上で, n 個の対応点 $(u_i, v_i) \leftrightarrow (u'_i, v'_i)$ が与えられているとする. このとき対応点は 2 次元射影変換 H で結ばれるが, この H を推定したい. 対応点の真の座標値を $(\bar{u}_i, \bar{v}_i) \leftrightarrow (\bar{u}'_i, \bar{v}'_i)$ とするとき,

$$\begin{bmatrix} \bar{u}'_i \\ \bar{v}'_i \\ 1 \end{bmatrix} \propto H \begin{bmatrix} \bar{u}_i \\ \bar{v}_i \\ 1 \end{bmatrix} \quad (23)$$

を満たす. θ は H の独立な成分をパラメータ化したものである (スケール倍の自由度を除去するのに H の例えば (3, 3) 成分を 1 に固定できるとすれば, 残りの 8 成分を並べたベクトルが θ). 上の式が陰的拘束に当たる. $\xi_i \equiv [\bar{u}_i, \bar{v}_i]^T$ とすると, \bar{u}'_i, \bar{v}'_i は (上の式から) θ および ξ_i の関数の形に書き直せ, これが陽的拘束となる.

例 4.4(SFM). 例 2.1 の SFM を考える. この場合, 上の定式化への当てはめ方は 2 通りある. 1 つの観測 \mathbf{z}_i を画像 1 枚に関連付けるか, あるいは点 1 つに関連付けるかの 2 通りである. まず, 画像 1 枚を 1 つの観測と見ると, 視点 i の画像上の点の座標値をすべて並べて $\mathbf{z}_i \equiv [u_{i1}, v_{i1}, u_{i2}, v_{i2}, \dots]^T$ を定義すれば, $\xi_i \equiv \mathbf{p}_i$ とすることで, (4) 式から陽的拘束が得られる. この場合 θ は, $\theta = [\mathbf{q}_1^T, \dots, \mathbf{q}_m^T]^T$ のように各点の空間座標をすべて並べたものである. 次に, 点 1 つの全画像上の像を 1 つの観測と見ると, 各点 j の各画像 ($i = 1, \dots$) 上の座標値を並べて $\mathbf{z}_j \equiv [u_{1j}, v_{1j}, u_{2j}, v_{2j}, \dots]^T$ を定義する. この点の空間座標を $\xi_j \equiv \mathbf{q}_j = [X_j, Y_j, Z_j]^T$ とすると, 同じように (4) 式から異なる陽的拘束を得る. この場合 θ は, $\theta = [\mathbf{p}_1^T, \dots, \mathbf{p}_m^T]^T$ のようにカメラのパラメータをすべて並べたものである.

上のすべての例で, 1 つの観測 \mathbf{z}_i に対し 1 つのパラメータ ξ_i が存在し, パラメータの数は観測数に比例して増加した. しかし, すべての問題がそうなるわけではない. 観測が増えなくてもパラメータが増えない例を以下に示す.

例 4.5(既知の平面・画像間の射影変換). 1 枚の平面とそれを撮影した画像間の 2 次元射影変換を求める場合に, 平面と画像の n 個の対応点 $(u_i, v_i) \leftrightarrow (u'_i, v'_i)$ が与えられるとする. 平面上の点 (u_i, v_i) は観測したものではなく, 真の値であるとする. 一方, 画像上の対応点 (u'_i, v'_i) は観測値で, その真の値 (\bar{u}'_i, \bar{v}'_i) は未知である. $[\bar{u}'_i, \bar{v}'_i, 1]^T \propto H[u_i, v_i, 1]^T$ なる H の推定が目的となる.

この例は, 先述の例 4.3 に一見似ているが, (u_i, v_i) は観測されるのではなく, その値が直接分かっているため, 観測 (u'_i, v'_i) ごとのパラメータが存在しない点で大きく異なる. この問題では H のみが未知パラメータで, 観測数が増えなくてもパラメータはこれだけである.

4.3 最尤推定

4.3.1 最尤推定としてのバンドル調整

観測 \mathbf{Z} が, 確率密度関数 $p(\mathbf{Z}; \Theta)$ に従うとき, \mathbf{Z} から Θ を推定したい. 最尤推定とは, 観測の実現値 \mathbf{Z} に対して, それが得られる確率の値を最大とするような Θ の値を, Θ の推定値として選ぶやり方を言う. $L(\Theta; \mathbf{Z}) \equiv p(\mathbf{Z}; \Theta)$ を \mathbf{Z} に対する Θ の尤度 (likelihood) と言う.

問題 4.1 および 4.2 において, 観測 $\mathbf{z}_i = \bar{\mathbf{z}}_i + \varepsilon_i$ の誤差 ε_i はガウス分布 $N(\mathbf{0}, \mathbf{V})$ に従うとした. これ以降, 簡単のため, 誤差の分布が等方的, すなわち $\mathbf{V} = \sigma^2 \mathbf{I}$ とする (ただし一般の \mathbf{V} に拡張するのは容易である). このとき, \mathbf{z}_i は $N(\bar{\mathbf{z}}_i, \sigma^2 \mathbf{I})$ に従う. 問題 4.1 のように観測の真の値が陽に $\bar{\mathbf{z}}_i = \mathbf{f}(\xi_i, \theta)$ と表現されたとすれば, \mathbf{z}_i の密度関数は (ξ_i, θ をパラメータとし)

$$p(\mathbf{z}_i; \xi_i, \theta) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{z}_i - \bar{\mathbf{z}}_i)^2\right) \quad (24)$$

となる. さらに, 全観測をまとめて $\mathbf{Z} = [\mathbf{z}_1^T, \dots, \mathbf{z}_n^T]^T$ と書けば, \mathbf{Z} の密度関数は, 各観測の誤差が互いに独立であるという仮定より

$$p(\mathbf{Z}; \mathbf{x}) = \prod_{i=1}^n p(\mathbf{z}_i; \xi_i, \theta) \quad (25)$$

となる. ただし $\mathbf{x} \equiv [\theta^T, \xi_1^T, \dots, \xi_n^T]^T$ である.

Z の実現値に対する尤度 $L(\mathbf{x}; \mathbf{Z}) \equiv p(\mathbf{Z}; \mathbf{x})$ を最大化する \mathbf{x} の値が、 \mathbf{x} の最尤推定量である。尤度 L の最大化を考える代わりに対数尤度 $l(\mathbf{x}; \mathbf{Z}) \equiv \log L$ の最大化を考えると式が簡単になる。

$$l(\mathbf{x}; \mathbf{Z}) = -\frac{1}{2\sigma^2} \sum_{i=1}^n (z_i - \mathbf{f}(\xi_i, \boldsymbol{\theta}))^2 - n \log 2\pi\sigma^2 \quad (26)$$

再投影誤差 $E(\mathbf{x})$ を

$$E(\mathbf{x}) \equiv \frac{1}{2} \sum_{i=1}^n (z_i - \mathbf{f}(\xi_i, \boldsymbol{\theta}))^2 \quad (27)$$

と定義すれば、 $l = -E/\sigma^2 - n \log 2\pi\sigma^2$ であり、誤差分散 σ^2 の値が未知であっても、 E を最小化する \mathbf{x} が最尤推定量に一致すると分かる。この式は、2 節で述べた再投影誤差 (3) 式に一致し、今想定する観測誤差モデルの下では、 E を最小化することが最尤推定に相当する。

また、問題 4.2 の $\mathbf{f}_c(\bar{z}_i, \boldsymbol{\theta}) = \mathbf{0}$ のように観測の真値 \bar{z}_i が陰的に拘束される場合 (問題 4.1 に書き換えないうき) は、観測の真値を表す変数 $\eta_i (\equiv \bar{z}_i)$ を導入し、次のような制約条件付きの最小化を行えばよい。

$$\text{Minimize } E(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (z_i - \eta_i)^2 \quad (28a)$$

$$\text{subject to } \mathbf{f}(\eta_i, \boldsymbol{\theta}) = \mathbf{0}, \quad i = 1, \dots, n \quad (28b)$$

この解が \mathbf{x} の最尤推定量となる。拘束条件が付帯する分、この数値計算は拘束なしの場合に比べて (一般にはかなり) 厄介である。

4.3.2 推定精度の限界 (クラメル・ラオの下界)

観測を元にモデルのパラメータを推定する方法は理論上無数にある。その中で、最も精度良くパラメータを推定する方法がどのようなものかに関心がある。推定量の「精度」を一義的に定義することはできないものの、一般的には、第 1 に推定量の期待値がその真の値に一致すること (不偏性 (unbiasedness) とする) をまず要請した上で、推定量の分散の小ささを精度と考える (このとき、推定誤差 (すなわち、推定量の真値との差) の 2 乗の期待値は、推定量の分散に等しくなる。) 一般的にも、分散が小さいことは推定量の望ましい性質となるが、与えられた観測から推定量の分散をどこまで小さくできるかは、理論的に導出できる。これはクラメル・ラオ (Cramèr-Rao) の下界として知られ、具体的には、 \mathbf{x} の任意の

推定量 $\hat{\mathbf{x}} = \mathbf{t}(\mathbf{Z})$ に対し、それが不偏 ($E_{\mathbf{x}}[\hat{\mathbf{x}}] = \mathbf{x}$) のとき、その共分散行列が

$$\text{Cov}_{\mathbf{x}}(\hat{\mathbf{x}}) = \text{Cov}_{\mathbf{x}}(\mathbf{t}(\mathbf{Z})) \geq \mathcal{I}^{-1}(\mathbf{x}) \quad (29)$$

となるというように与えられる ($\text{Cov}_{\mathbf{x}}$ は \mathbf{x} での期待値に基づくことを表す)。ここで、行列 A, B に対する $A \geq B$ は、 $A - B$ が半正定 (semi-positive definite) であることを表し、また $\mathcal{I}(\mathbf{x})$ は次のように定義される Fisher 情報行列である：

$$\mathcal{I}(\mathbf{x}) = E_{\mathbf{x}} \left[\left(\frac{\partial l}{\partial \mathbf{x}} \right) \left(\frac{\partial l}{\partial \mathbf{x}} \right)^{\top} \right] \quad (30)$$

l は (26) 式で定義した対数尤度 $l(\mathbf{x}; \mathbf{Z}) = \log p(\mathbf{Z}; \mathbf{x})$ で、 $E_{\mathbf{x}}[\cdot]$ は \mathbf{x} を指定した $p(\mathbf{Z}; \mathbf{x})$ に基づく期待値である (以下では、紛れのない場合は \mathbf{x} を省略し $E_{\mathbf{x}}[\cdot] = E[\cdot]$ と書く)。

上のクラメル・ラオの下界は、不偏推定量の分散の下限を与える。つまりこの下限に分散が一致する不偏推定量は、不偏推定量の中で最良だということである。

4.3.3 最尤推定の漸近最適性

4.3.1 節のようにして得られる最尤推定量は、観測誤差の分散 σ^2 が 0 に漸近する ($\sigma^2 \rightarrow 0$) 極限において、推定量の真の値を平均とし、フィッシャー情報行列の逆行列を共分散行列とする正規分布に従うことが示される。つまり、その極限で最尤推定量はクラメル・ラオの下界を達成する不偏推定量であり、その意味で最良である (つまり、どの推定量よりも良い)。実際には観測誤差分散は 0 ではないが、それが十分小さければ、最尤推定量はほぼ最良であると言ってよい。これが、最尤推定の漸近最適性である。

なお、最尤推定量は、 $\sigma^2 \rightarrow 0$ ではなく、観測数が無限大に漸近する極限において上の最適性を持つと述べられるのが普通である¹³⁾。これは最尤推定の性質のより一般的な表現だが、この「観測数」をわれわれの問題での観測 z_1, \dots, z_n の数 (つまり n) と考えると、上述の漸近最適性は導かれないことに注意する。これは、われわれの問題 (問題 4.1 および 4.2) では、観測 1 つにたいし 1 つのパラメータが存在するため、観測数 $n \rightarrow \infty$ とするとパラメータも比例して増加することによる。このような構造を持つ問題は、統計学ではネイマン・スコット (Neyman-Scott) 問題⁹⁾ として知られ、最尤推定の通常漸近最適性が成立しない例外として長い間研究の対象となっている。

われわれの問題における上述の漸近最適性は (やや強引なものの) 「観測数」を次のように読み替えることで導かれている⁴⁾。まず、一つの z_i (の真の位置 \bar{z}_i) を、仮想的に繰り返し観測すると考える。正確には、 \bar{z}_i を m 回繰り返し観測し、 m 個の観測値 z_{i1}, \dots, z_{im}

を得たとき、それぞれが独立に $N(\bar{z}_i, \sigma^2 \mathbf{I})$ に従うとする。この繰り返し観測回数 m を「観測数」とするとき、 m が無限大に漸近する極限において、漸近最適性が成立する。 $n \rightarrow \infty$ の場合と違って、 $m \rightarrow \infty$ ではパラメータの数が増えないことに注意する。そして、繰り返し観測の回数 m を無限大 $m \rightarrow \infty$ とするのは、1回の観測における分散を無限小 $\sigma^2 \rightarrow 0$ とすることと等価である。これにより、最初の漸近最適性の表現が導かれている。

結論は、観測誤差の分散が十分小さいとき、最尤推定量はほぼ最高の精度を達成するということである。逆に言うと、観測誤差の分散が小さくなければ、最尤推定量はそれほど良くない可能性がある*1。具体的には、推定量に偏差が生じたり、分散が大きくなることもある。ただし、バンドル調整で対象となる問題における観測の誤差分散は通常十分小さく、最尤推定量はほとんど最良であると経験的には考えられている*2。

なお、以上の議論は、問題 4.1 および 4.2 のようにネイマン・スコット構造（観測 n に比例してパラメータ数が増加する）を持つ場合についてのものである。幾何学的なパラメータ推定の問題でも、例 4.5 のようにネイマン・スコット構造を持たない場合は、 $m \rightarrow \infty$ すなわち $\sigma^2 \rightarrow 0$ の漸近論に加えて、 $n \rightarrow \infty$ の漸近論においても、最尤推定の漸近最適性が成り立つことになる。

4.3.4 最尤推定量の精度の計算

先述のように、最尤推定量は漸近的に正規分布に従い、その共分散はフィッシャー情報行列の逆行列 $\mathcal{I}^{-1}(\mathbf{x})$ である。したがって、その漸近最適性が成り立つ範囲で、最尤推定量の精度、すなわち共分散を推定することができる。これには2通りの方法があり、一つは、 $\mathcal{I}(\mathbf{x})$ を期待値計算によって解析的に求めておき、計算した最尤推定量 $\hat{\mathbf{x}}$ で \mathbf{x} を置き換えたものを用いる方法である。その逆行列 $\mathcal{I}(\hat{\mathbf{x}})^{-1}$ が、推定量の共分散の推定値となる。もう一つは、次のように定義される観測情報行列（observed Fisher's information）

$$\mathcal{J}(\hat{\mathbf{x}}) = \left. \frac{\partial^2 l}{\partial \mathbf{x}^2} \right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (31)$$

を計算し、その逆行列 $\mathcal{J}^{-1}(\hat{\mathbf{x}})$ を求めて推定量の共分散の推定値とする方法である。

われわれの問題では、 E のヘッセ行列 \mathbf{H} およびそのガウス・ニュートン法における近似 \mathbf{A} ($\mathbf{A} = \mathbf{J}^T \mathbf{J}$) を使うと、 $\mathcal{J} = (1/\sigma^2)\mathbf{H}$ および $\mathcal{I} = (1/\sigma^2)\mathbf{A}$ と書ける。したがって $\mathcal{J}(\hat{\mathbf{x}})$

や $\mathcal{I}(\hat{\mathbf{x}})$ は、改めて計算しなくても E の最小化計算収束時（つまり $\mathbf{x} = \hat{\mathbf{x}}$ ）の \mathbf{H} および \mathbf{A} をそのまま用いればよい。最尤推定量の共分散の推定値は、これらの逆行列を求めることで $\sigma^2 \mathbf{H}^{-1}$ あるいは $\sigma^2 \mathbf{A}^{-1}$ と計算される。なお、共分散の推定値としては、 $\mathcal{I}^{-1}(\hat{\mathbf{x}})$ よりも $\mathcal{J}^{-1}(\hat{\mathbf{x}})$ の方が一般に良いとされる¹⁾ が、ガウス・ニュートン法やレベンバーグ・マーカート法を使う場合は $\mathcal{I}^{-1}(\hat{\mathbf{x}})$ の方が簡単に求まる。

また、 σ は未知数であるから、精度評価のためにはこれも推定する必要がある。これは、 E の最小値 $E(\hat{\mathbf{x}})$ （つまり E の最小化後の残差）を用いて推定できる。問題 4.1 のように定式化し、誤差分散を $\mathbf{V} = \sigma^2 \mathbf{I}$ とするとき、観測 \mathbf{z}_i の成分数（ベクトル長さ）を n_z 、 ξ_i の成分数を n_ξ としたとき、誤差 $\mathbf{e}_i (= \mathbf{z}_i - \bar{\mathbf{z}}_i)$ の2乗和の期待値が $E[\|\mathbf{e}_i\|^2] = \sigma^2(n_z - n_\xi)$ となることから（ $E(\mathbf{x}) = (1/2) \sum_i \|\mathbf{z}_i - \bar{\mathbf{z}}_i\|^2$ に注）、 σ^2 の推定値を $\sigma^2 = 2E(\hat{\mathbf{x}})/(n(n_z - n_\xi))$ と計算できる。

\mathbf{x} の一部（部分ベクトル）の精度を評価するには、上のように計算した \mathbf{x} 全体の共分散行列のうち、その部分ベクトルに対応するブロック対角行列を取り出せば良い。つまり、収束後の $\sigma^2 \mathbf{H}^{-1}$ あるいは $\sigma^2 \mathbf{A}^{-1}$ のブロック対角行列を取り出せば、それはその部分ベクトルの共分散を与える。

5. ゲージの自由度

5.1 ゲージの自由度とバンドル調整

カメラや空間の点が与える幾何は、空間に座標系を定めることで初めて記述できる。物理的な実体である幾何とその表現のための座標系は基本的に独立しているので、通常、座標系の選択には任意性がある。例えば、最初の例 2.1 の SFM ような、画像のみからカメラのパラメータおよび点の位置を計算する問題では、空間に定める座標系には、7自由度分の任意性（回転・並進の自由度がそれぞれ3、およびスケール倍の自由度1）がある*3。選んだ座標系ごとに、対象の幾何は（物理的には同一だが）異なる座標値によって表現されることになる。このような座標系の決め方のことをゲージ（gauge）と呼ぶ。

数値計算の際、この任意性は解決する必要がある。任意性があると、ヤコビ行列 \mathcal{J} やヘッセ行列 \mathbf{H} がフルランクでなくなる。このとき、ランクの低下数は、座標系の任意性もつ自由度に一致する*4。また座標系の任意性とは別に、冗長なパラメータ表現を選んだとき

*1 このとき、観測数を増やしても（ m ではなく n を増やす）、漸近最適性を成立させる方向には基本的に寄与しない。

*2 偏差を除いて推定量の精度を向上させる試み^{4),8)}（必ずしもバンドル調整を想定していないが）もある。

*3 スケール倍も定めるユークリッド復元の場合は、座標系の自由度は回転3と並進3の合計6（あまりバンドル調整では扱わないが）射影復元の場合は15である。

*4 座標系の任意性は、座標変換 $\mathbf{x}' = \mathbf{t}(\mathbf{x})$ によって表現される。変換 \mathbf{t} の前後で対象の幾何は不変なので、 $E = \|\mathbf{e}\|^2/2$

も、同様に J や H のランクは低下する*1。例えば、点の空間座標を同次座標 4 成分で冗長に表現する場合や、回転を四元数で表すときに、長さ 4 のベクトル q で ($\|q\|^2 = 1$ の拘束なしに) 冗長に表現する場合などである。 J や H は、やはりその冗長性の分だけランクが低下する。

ニュートン法での更新式 $H\delta x = -g$ の計算において、 H がフルランクでなければ δx は一意に定まらない。一般的な最小化アルゴリズムは、 H がフルランクであることを前提としており、対処が必要である。

もっとも簡単で一般的な方法は、 x の成分のなかから座標系の任意性の自由度と同数の成分を選んで、それを固定して (パラメータではなく) 定数としてしまうことである (これをトリビアルゲージ (trivial gauge) と呼ぶ¹⁰)。SFM では例えば、第 1 カメラのカメラ座標系を固定し (例えば $P_1 = K_1[I | 0]$ とする)、さらにスケール倍の不定性を除去するため、第 2 カメラの並進成分の座標の 1 つ (例えば x 座標) を固定し、上述の 7 自由度を拘束する。

この方法では、対応する数値計算の修正は次のように単純である。まず、 x の成分のうち固定する成分を取り除いて、新たにパラメータのみからなる x を定義し直す。これに対応して、 J は対応する列を取り除いたものを新たな J とし、 H は対応する行および列を取り除いたものを新たな H とする。この結果、 H はフルランクとなる。

H がフルランクでない場合に対処するもう一つの方法は、 $H\delta x = -g$ の計算の際、例えば $\|\delta x\|^2$ が最小になるようなものを選ぶ (μ -ア・ペンローズの逆行列 H^\dagger を用いれば $\delta x = -H^\dagger g$) など、何らかの基準により δx を一つ選ぶことである。冗長なパラメータ表現を選んだ場合には、このような方法で δx を計算し、 $x + \delta x \rightarrow x$ と更新した後、冗長性を除く拘束を満たすように x を修正し (例えば空間の点をその同次座標 M で表現しているならば $\|M\|^2 = 1$ となるように修正)、その後次の更新に移る、という手順は良く使われる。

5.2 ゲージと見かけの推定精度

座標系の選び方は無数にあり、 x の成分のいくつかを固定するトリビアルゲージに限っても、その成分の選び方は多数ある。そのうちどれを選んで、その結果得られる解は、単なる座標変換によって相互に変換可能であり、その意味でどのゲージを選んで、得られる結果は本質的に等価である。しかしながら、「見かけの精度」は、ゲージによって大きく変化

する。

例として、上述の 7 つの自由度を持つ SFM の場合で考える。第 1 カメラの座標系および第 2 カメラの x 成分を固定するものを方法 I とすれば、この他に例えば、復元する点のうち (同一直線状にない) 3 点を選んで、そのうち 2 点の空間座標を固定 (例えば $[0, 0, 0]$, $[1, 0, 0]$) し、残りの 1 点の z 座標を固定 (例えばその点が xy 面上にある) する方法も考えられる。これを方法 II とする。方法 I と II によって得られる 2 つの数値解は、いずれかの方法が数値不安定性を誘発するような特殊な状況にない限り (わずかな数値誤差を除いて) 互いに座標変換のみで変換可能である。つまり、2 つの解は等価である。

しかしながら、見かけの推定精度は両者で大きく異なる。方法 I では、カメラ (特に第 1 カメラ) は固定されているので、その位置・姿勢の推定誤差は 0 である。一方で、点の推定位置は大きくばらつく傾向を持つ。方法 II では、固定された点の位置の推定誤差は 0 であり、一方でカメラの位置・姿勢の推定が大きくばらつく。図 2 に例 2.1 を使った例を示す。これは、焦点距離と位置・姿勢の未知なカメラ 2 台で、直交する平面上に分布する 32 点の画像を撮影し、そこからカメラのパラメータと点の 3 次元位置を推定したものである。図では、点の 3 次元位置の推定結果のみ示しており、推定のばらつきを、4.3.4 節のように求めた共分散行列に基づく楕円体で表現した。方法 I では点のばらつきが大きく*2、方法 II ではそれは全体に小さく、特に固定した点まわりに近い点ほどその傾向がある。また、ばらつきの方向も、両方法でかなり異なることが見て取れる。カメラの姿勢については省略するが、点とちょうど反対のことが起こっている。トリビアルゲージでは、大雑把に言えば、固定したパラメータとよりつながりの深いものほど、推定値のばらつきが小さくなり、反対に薄いものほどばらつきは大きくなる。

このような違いが生じることは、ゲージの自由度の下で特定の座標系を選択せざるを得ないことが原因であり、避けることはできない。そこで一般に、応用ごとに、推定したい対象の側に基準座標をとる。例えば SFM では、カメラ側に座標系をとれば点のばらつきが、点側に座標系をとればカメラ側のばらつきが大きくなるから、カメラの位置・姿勢の推定が目的であればカメラ側に、点の位置の推定が目的であれば点側に、それぞれ座標系を選択する。

ゲージには上述のトリビアルゲージ以外にも、推定された結果そのものを基準とするものもある。例えば、復元された点の重心位置とその分散の方向で決まる座標系を選ぶなどであ

も不変で、さらに $e(x) = e(t(x))$ である。この式の微分から J, H のランクが導ける。

*1 さらにこれら以外にも、臨界条件 (critical configuration, 例えば点が 3 次元的に散らばらず平面上に乗るなど幾何学的に解が定まらない) の場合も、 J や H はフルランクでなくなる。ただしこれは、 x が特別な値をとることによって生起するもので、その点で座標系選定の自由度や冗長なパラメータ表現の場合とは異なる。

*2 方法 I の結果で点のばらつきが非常に大きく見えるが、各点の推定位置の間には強い相関があり、各点が一緒に変動することに注。この相関の強さは、図示できない。

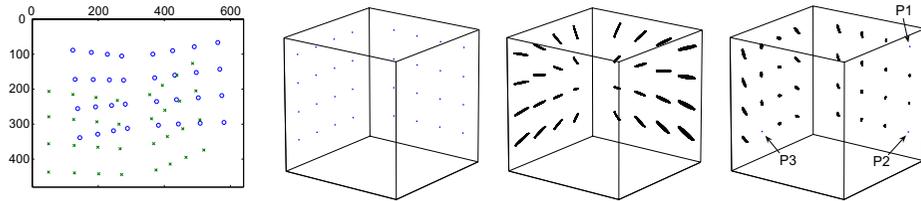


図 2 座標系の任意性の除去と見かけの精度（個々の点の推定位置の分散）．左から、使用した 2 枚の画像（重ねてある）、空間の点の真の位置、方法 I（カメラを固定）での点の誤差、方法 II（点の一部（P1,P2,P3）を固定）での点の誤差．

る．カメラ側でも同様なことが可能である．このようにした方が、推定精度をより自然に評価できるだろう（このようなゲージは内部ゲージ（inner gauge）と呼ばれる．）

そうするには次のような手順を踏めばよい．まず、適当なトリビアルゲージを選択し、最小化の数値計算を行ってパラメータを推定する．そして、得られた結果を、目的のゲージでの表現に変換する．これは単なる座標変換（例として考えている SFM では 3 次元の相似変換）である．もちろんこの変換の際、トリビアルゲージによって固定してあったパラメータの値も固定を解き、変換の対象とする．座標変換自体の算出は、例えば、上述のように復元した点側に座標系をとるならば、点の重心を算出し、原点が重心になるように座標系を決め（軸の方向は他の基準で選ばよいだろう）、そしてその新しい座標系と元の座標系間で求める．

6. 逐次計算と再帰的計算

6.1 バッチバンドル調整と逐次バンドル調整

ここまで、手元の観測をすべて使って、全パラメータを同時に推定することを考えてきた．この方法をバッチ計算と呼ぶことにする．

一方、観測が時間の経過とともに少しずつ獲得されるような場合に、時間軸に沿ってオンラインでパラメータを推定したい場合がある．例えば、運動カメラでシーンの 3 元形状情報を取得するとき、画像はカメラの運動（時間の経過）とともに一枚ずつ得られる．オンラインでパラメータを推定するとは、このように観測が到着するたびに、短い時間でパラメータを推定（あるいは既存のパラメータならば推定値を更新）し、これを繰り返すことである．

オンライン推定を実現するのに、最新の時刻までに得たすべての観測を使っ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、本節で述べる逐次計算法である．オリジナルは、McLauchlan の VSDF (Variable State Dimension Filter) である⁵⁾．実装によってはリアルタイムでの計算も十分可能となるから、ロボット視覚（例えば SLAM）やレスポンスを重視する UI など、様々な応用が考えられる．

なお、Triggs らの解説¹⁰⁾ では、ヘッセ行列 H の操作を主軸にこの方法が説明されているが、難解であるように思われるので、以下では、コスト関数 $E(x)$ を軸にした記述を試みる．

6.2 新しい観測の到着とパラメータの増加

運動カメラでシーンを撮影し、時々刻々画像が得られる状況を考える．新たな画像 1 枚を得るごとに、その画像撮影時のカメラのパラメータ（位置・姿勢および内部パラメータ）は、系にとって新しいパラメータとなる．画像上で抽出したシーンの点が、以前から観測し続けているものならば、その 3 次元位置は系の既存パラメータであり、この画像で初めて登場したものならば、その 3 次元位置は新しいパラメータとなる．このような形で、観測の到着に伴って新しいパラメータが系に追加される．

さて、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*1．1 枚目から m 枚目までの画像に対して定義される再投影誤差を $E_{1:m} \equiv \sum_{k=1}^m E_k$ と書き、新規に到着した $m + 1$ 枚目の画像 1 枚に関する再投影誤差を E_{m+1} と書く．定義から $E_{1:m+1} = E_{1:m} + E_{m+1}$ である． m 枚目までの画像に関連付けられたパラメータを x_1 、 $m + 1$ 枚目の画像の到着に伴って新規に追加されたパラメータを x_2 とすると、

$$E_{1:m+1}(x_1, x_2) = E_{1:m}(x_1) + E_{m+1}(x_1, x_2) \quad (32)$$

の関係がある．

6.3 観測の到着に伴う推定値の更新

上のように新たな観測が到着したとき、系の既存パラメータ x_1 は、それまでの観測を元にその時点での推定値を計算済みとする． x_1 のその推定値は、新たに到着した観測のおかげで修正される（よって精度が向上する）ことになるが、その修正を少ない計算量で求めることを考える．バッチ計算のボトルネックは、ニュートン法における更新量の計算ステップ ($A\delta x = a$ の求解) であるから、この計算量を小さくできれば良い．

まず、観測の追加に伴いパラメータが追加されない場合、つまり $E_{1:m+1}(x) = E_{1:m}(x) + E_{m+1}(x)$ となるときを考える．今、 m 枚目までの画像のコスト $E_{1:m}(x)$ に対し、これを最

*1 ここでは、画像を観測の単位として考えるが、問題によってその単位は変わるだろう．

小化する解 $\mathbf{x} = \mathbf{x}^*$ を得ているとする。このとき \mathbf{x}^* での \mathbf{A} を \mathbf{A}^* と書くと、次の 2 次関数 $\hat{E}_{1:m}(\mathbf{x})$

$$\hat{E}_{1:m}(\mathbf{x}) \equiv \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top \mathbf{A}^*(\mathbf{x} - \mathbf{x}^*) \quad (33)$$

は、極小点 \mathbf{x}^* まわりでの $E_{1:m}$ のよい近似を与えるだろう。

ここで $m+1$ 目の画像が追加されたとき、更新されたコスト $E_{1:m+1} = E_{1:m} + E_{m+1}$ を最小化する代わりに、その第 1 項 $E_{1:m}$ を上の近似 $\hat{E}_{1:m}(\mathbf{x})$ で置き換え、

$$E_{1:m+1} \approx \hat{E}_{1:m} + E_{m+1} \quad (34)$$

なる近似コストを最小化することとする。もし \mathbf{x}^* が十分真の値に近く、 \mathbf{x} の変動範囲内で $\hat{E}_{1:m}(\mathbf{x})$ が $E_{1:m}(\mathbf{x})$ に近いならば、上の近似コストの最小解も真のコスト $E_{1:m+1}$ のそれに近いと期待できる。

このように考えたとき、 E_{m+1} に対するニュートン法の更新ステップの方程式 (E_{m+1} を単独で最小化するときのもの) を $\mathbf{A}'\delta\mathbf{x} = \mathbf{a}'$ と書けば、(34) 式右辺の近似コスト全体を最小化するときの更新量の計算式は

$$(\mathbf{A}^* + \mathbf{A}')\delta\mathbf{x} = -\mathbf{A}^*(\mathbf{x} - \mathbf{x}^*) + \mathbf{a}' \quad (35)$$

と書ける。この式にしたがって、 $\delta\mathbf{x}$ を計算し、 $\mathbf{x} + \delta\mathbf{x} \rightarrow \mathbf{x}$ と更新する。更新後、 \mathbf{A}' および右辺を再計算し、再度上の式を解いて $\delta\mathbf{x}$ を計算しなおす。この手続きを収束するまで繰り返す。

このとき、 \mathbf{A}' のランクが小さければ、(35) 式は少ない計算量で解くことができる。 \mathbf{A}^* は、 $E_{1:m}$ の最小化時点で計算しており、しかもそれを係数行列とする方程式を既に一度以上解いてある。つまり、 \mathbf{A}^* のコレスキー分解 \mathbf{LL}^\top は計算済みである。したがって、 \mathbf{A}' のランクが小さければ (通常は、限られた量の観測に関わるものなので小さい)、 $\mathbf{LL}^\top + \mathbf{A}'$ のコレスキー分解は、 $\mathbf{A}^* + \mathbf{A}'$ 自体を改めてコレスキー分解するよりもずっと小さい計算量で計算できる (その一手法は、Triggs らの解説¹⁰⁾ の 59 頁付録 B.5 節に記載されている。) なおニュートン法の \mathbf{x} の修正 1 ステップごとに、 \mathbf{A}' は新たな \mathbf{x} を使って求め直すことに注意する。

反復が収束し近似コストの最小解 $\hat{\mathbf{x}}$ が見つかったら、その点での \mathbf{A}' を使って $\mathbf{A}^* + \mathbf{A}' \rightarrow \mathbf{A}^*$ および $\hat{\mathbf{x}} \rightarrow \mathbf{x}^*$ と更新する。次の観測が到着したとき、上の手続きを行い、これを継続する。

上では観測の到着に伴ってパラメータは新規に追加されないものとしたが、(32) 式のようにパラメータが追加される場合は次のようになる。 $E_{1:m}$ の最小値が $\mathbf{x}_1 = \mathbf{x}_1^*$ で

得られていて、そこでの更新量の方程式の係数行列が \mathbf{A}'_{11} であったとする。上と同様に $E_{1:m}(\mathbf{x}_1) = (1/2)(\mathbf{x}_1 - \mathbf{x}_1^*)^\top \mathbf{A}'_{11}(\mathbf{x}_1 - \mathbf{x}_1^*)$ と近似する。さらに E_{m+1} のみを最小化するときの更新量の計算式を

$$\begin{bmatrix} \mathbf{A}'_{11} & \mathbf{A}'_{12} \\ \mathbf{A}'_{21} & \mathbf{A}'_{22} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \end{bmatrix} \quad (36)$$

と書く。すると、トータルのコストに対する方程式は

$$\begin{bmatrix} \mathbf{A}^*_{11} + \mathbf{A}'_{11} & \mathbf{A}'_{12} \\ \mathbf{A}'_{21} & \mathbf{A}'_{22} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{A}^*_{11}(\mathbf{x}_1 - \mathbf{x}_1^*) + \mathbf{a}'_1 \\ \mathbf{a}'_2 \end{bmatrix} \quad (37)$$

となる。この場合も、上述のように \mathbf{A}^* のコレスキー分解が既知なので計算量の節約が可能である。

なお上では、 $E_{1:m+1} = E_{1:m} + E_{m+1}$ のように、最新の画像とそれ以外すべてに分けたが、 $E_{1:m+1} = E_{1:m-w+1} + E_{m-w+2:m+1}$ のように、最新の画像を含み w だけ過去にさかのぼった画像系列と、それ以前の古いものすべてに分けて、古い方を上のように 2 次関数で近似することもできる。これは、時系列の観測を w の窓を通して取り出し、その部分の観測はいわば生きた状態 (= 本来の非線形関数のまま) で取り扱い、それ以前の観測は死んだ状態 (= 2 次関数で近似し固定) とするイメージである。 w が長ければ長いほど、計算量は増加するが近似精度は良い (これらの上限は全観測と w を一致させるバッチバンドル調整であることに注意する。)

6.4 パラメータの削除と再帰的計算

観測の到着に伴ってパラメータは増加する一方であるので、系の規模は単調に増加する。これを一定以下に抑えるには*1、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可能性のあ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。例えば、カメラが大きく移動する場合に、時間軸上かなり以前の画像に対応するカメラの姿勢や、そのときのシーンの 3 次元座標などである。

*1 簡単な方法は、時系列の観測データを、一定のサイズ w の窓を通して見てやり、その窓部分の観測についてのみ、コストを最小化することである。すなわち、 $E_{1:m}$ を最小化するのではなく、 w を定数として $E_{m-w+1:m}$ を最小化する。ただしこれは毎回、小規模のバッチ計算をしていることに等しく、過去の観測は将来にまったく引き継がれない。

ただしどんなに古い（観測と関連する）パラメータであっても、最新あるいは将来の観測と間接的には関わりがあり（なければバンドル調整ではない）、新しい観測の到着は、それら古いパラメータも必ず影響を受ける（一般には、より精度を向上させる）。故に、バッチ計算と比べて、以下に述べるのは、前節の更新計算同様、精度を犠牲に目的（パラメータの削除）を実現する方法である。

今、 $E_{1:m}$ をそのパラメータ \mathbf{x} について最小化済みとする。 $E_{1:m}$ のパラメータ \mathbf{x} を $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T]^T$ と 2 分し、 \mathbf{x}_2 をこれから削除するパラメータ、 \mathbf{x}_1 を今後も残るパラメータとする。このとき、将来の観測に対するコスト E_{new} は、 \mathbf{x}_2 に依存してはならない。そうでなければ \mathbf{x}_2 は削除できない。つまり、 E_{new} を含むトータルのコストは

$$E_{1:m}(\mathbf{x}_1, \mathbf{x}_2) + E_{new}(\mathbf{x}_1, \mathbf{x}_3) \quad (38)$$

なる構造を持つことになる。ここで \mathbf{x}_3 は、将来の観測に対して現れた新しいパラメータである。

(38) 式において、新たな観測によるコスト $E_{new}(\mathbf{x}_1, \mathbf{x}_3)$ の影響で、 \mathbf{x}_1 は変動する。 \mathbf{x}_2 は、この \mathbf{x}_1 の変動に対して、 $E_{1:m}(\mathbf{x}_1, \mathbf{x}_2)$ を経由して変動する。つまり \mathbf{x}_2 の変動は、 \mathbf{x}_1 のそれに対して従属的である。このとき、 $E_{1:m}(\mathbf{x}_1, \mathbf{x}_2)$ を前節同様に、その最小解まわりの 2 次関数で近似すれば、 \mathbf{x}_2 の変動は（ニュートン法の反復に頼るまでもなく）線形計算で \mathbf{x}_1 の変動に対して直接的に計算される。

以下に詳しく述べる。 $E_{1:m}$ に対して見出した極小点 $\mathbf{x}^* = [\mathbf{x}_1^{*T}, \mathbf{x}_2^{*T}]^T$ まわりで、 $E_{1:m}$ を 2 次近似する*1：

$$\hat{E}_{1:m}(\mathbf{x}_1^* + \delta\mathbf{x}_1, \mathbf{x}_2^* + \delta\mathbf{x}_2) = \frac{1}{2} \begin{bmatrix} \delta\mathbf{x}_1^T & \delta\mathbf{x}_2^T \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} \quad (39)$$

そして、(38) 式の代わりに、現在までのコストを近似コストで置き換えた $\hat{E}_{1:m}(\mathbf{x}_1, \mathbf{x}_2) + E_{new}(\mathbf{x}_1, \mathbf{x}_3)$ を、パラメータ $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ について最小化することを考える。 \mathbf{x}_1 を固定したとき、第 1 項 $\hat{E}_{1:m}(\mathbf{x}_1, \mathbf{x}_2)$ を最小にするような \mathbf{x}_2 は、 $\mathbf{x}_2 = \hat{\mathbf{x}}_2(\mathbf{x}_1)$ のように \mathbf{x}_1 の関数として決まる。そしてこれを $\hat{E}_{1:m}(\mathbf{x}_1, \mathbf{x}_2)$ の \mathbf{x}_2 に挿入すれば、 $\hat{E}_{1:m} = \hat{E}_{1:m}(\mathbf{x}_1, \hat{\mathbf{x}}_2(\mathbf{x}_1))$ となって \mathbf{x}_1 のみの関数となる。具体的には $\partial \hat{E}_{1:m} / \partial \delta\mathbf{x}_2 = \mathbf{0}$ を解くと、

$$\delta\mathbf{x}_2 = -A_{22}^{-1} A_{21} \delta\mathbf{x}_1 \quad (40)$$

1 前節の記法に習えば、 A_{11}^ のように * をつけるべきだが、簡素化のためそうしていないが、これら行列は定数。

のように決まる。これを $\hat{E}_{1:m}$ に代入すると、

$$\hat{E}_{1:m} = \frac{1}{2} \delta\mathbf{x}_1^T (A_{11} - A_{12} A_{22}^{-1} A_{21}) \delta\mathbf{x}_1 \quad (41)$$

と書き換えられる。この書き換えを行えば、(38) 式を $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ について最小化する問題を、(41) 式と E_{new} の和を $(\mathbf{x}_1, \mathbf{x}_3)$ について最小化する問題へ書き換えられる。ここに、 \mathbf{x}_2 が削除された。

ここで重要なのは、この書き換え後に \mathbf{x}_2 は形式上姿を消すが、書き換えられた最小化問題は、(38) 式を全パラメータについて最小化するのと等価であることである。現に、 \mathbf{x}_1 が定まった後に、(40) 式を用いて $\mathbf{x}_2 = \mathbf{x}_2^* - A_{22}^{-1} A_{21} (\mathbf{x}_1 - \mathbf{x}_1^*)$ と計算すれば、 \mathbf{x}_2 を求めることもできる。

さて、書き換えられた最小化問題に対する更新の式は、前節同様 E_{new} のみを最小化するときの方程式を

$$\begin{bmatrix} A'_{11} & A'_{13} \\ A'_{31} & A'_{33} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_3 \end{bmatrix} \quad (42)$$

とすると、トータルのコストに対する方程式は

$$\begin{bmatrix} A_{11}^* + A'_{11} & A'_{13} \\ A'_{31} & A'_{33} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} -A_{11}^* (\mathbf{x}_1 - \mathbf{x}_1^*) \mathbf{a}'_1 \\ \mathbf{a}'_3 \end{bmatrix} \quad (43)$$

となり、前節の逐次的更新の式と同じ形となる。ただし、

$$A_{11}^* = A_{11} - A_{12} A_{22}^{-1} A_{21} \quad (44)$$

である（シューア補行列に一致することに注意。） \mathbf{x}_2 を削除したことがここに現れる。

手順をまとめれば、 \mathbf{x}_2 を削除するには、 $(\mathbf{x}_1, \mathbf{x}_2)$ に関する最小化の計算が収束した後、その係数行列 A に対し、上の式にしたがって A_{11}^* を計算し、これを最小解 \mathbf{x}_1^* とともに、将来に引き継げば良い。

上では、 $E_{1:m}$ と E_{new} に分け、 $E_{1:m}$ を 2 次関数で近似したが、前節同様、 $E_{1:m}$ をさらに $E_{1:m-w+1}$ と $E_{m-w+2:m}$ の 2 つに分け、 $E_{1:m-w+1}$ のみを 2 次関数で近似し、残りは非線形のまま評価することもできる。

なお、以上の再帰的計算の仕組みは、カルマンフィルタのそれと基本的に同じであることに注意する。もちろんカルマンフィルタは、ダイナミクスを扱う点でそもそも問題が異なる

が、計算の仕組みは同じである。その場合、問題は観測 z_1, \dots, z_m を元に、状態ベクトル x_1, \dots, x_m を推定することである。全観測から全状態ベクトルを最尤法により（ダイナミクスと一緒に考慮して）計算すれば、これはバンドル調整のバッチ計算に相当する。1つの観測 z_k が到着したとき、状態ベクトル x_k を推定するように上の再帰的計算を構成すれば、カルマンフィルタが導かれる（情報行列を保持する形になるため、厳密には情報フィルタ（information filter）に近い）。

7. おわりに

記述量を小さくするため、拘束条件付きの最小化やロバストなコスト関数やその最小化の話題など、副次的な話題は割愛した。これらについても Triggs らの解説¹⁰⁾に記載があり、必要な場合は参照されたい。

付 録

A.1 回転の表現について

3次元空間の回転を表現する方法はいくつか存在する

- (a) 回転軸方向 \mathbf{a} ($\|\mathbf{a}\|^2 = 1$) と回転角 θ のペアあるいはそれをまとめた $\boldsymbol{\omega} = \theta \mathbf{a}$
- (b) 四元数 (quaternion) $\mathbf{q} = [\cos(\theta/2), \sin(\theta/2)\boldsymbol{\omega}]$
- (c) オイラー角 (ϕ, θ, ψ) (順に z, x, z 軸周りに回転: $R = R_z(\psi)R_x(\theta)R_z(\phi)$) である。

なお (a) の表現から回転行列を計算するには、Rodrigues の公式を用いる：

$$R = I + \sin \theta [\mathbf{a}]_{\times} + (1 - \cos \theta) [\mathbf{a}]_{\times}^2 \quad (45)$$

ただし $\mathbf{a} = [a_1, a_2, a_3]^T$ ($\|\mathbf{a}\|^2 = 1$) に対して

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (46)$$

である。

オイラー角表現は、いわゆるジンバルロック (gimbal lock) の特異性等の問題があり、これらを十分意識できている場合を除き使用しないのが良い。一般には、四元数を使うか、ある固定した回転 R からの差分 δR を用いて目的の回転を $R\delta R$ と表すのが良い。四元数は、 \mathbf{q} の4成分を $\|\mathbf{q}\|^2 = 1$ の拘束条件とともに使うことになるが、 E の最小化の数値計算を行うの

に、拘束付きの最小化よりも拘束無しの最小化の方が自由度が高い。そのため、 $\mathbf{x} \rightarrow \mathbf{x} + \delta \mathbf{x}$ の修正量 $\delta \mathbf{x}$ を計算する際、 \mathbf{q} の拘束を無視し、その後に $\|\mathbf{q}\|^2 = 1$ となるように修正するなどする（なおその場合には、ヤコビ行列 J やヘッセ行列 H が縮退するので、その対策は必要）。このような処理を嫌う場合には、回転の差分表示 $R\delta R$ が良いだろう。差分量 δR は、オイラー角や回転軸・角度表現を用いれば良い（これらは大域的な回転を表現するには問題があるが、差分の回転角が小さいことを前提とすれば問題ない）。

参 考 文 献

- 1) Efron, B. and Hinkley, D.V.: Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher Information, *Biometrika*, Vol.65, No.3, pp.457–487 (1978).
- 2) Goto, K. and vande Geijn, R.A.: High-performance implementation of the level-3 BLAS, *ACM Trans. Math. Softw.*, Vol.35, No.1 (2008).
- 3) Hartley, R. and Zisserman, A.: *Multi-View Geometry in Computer Vision*, Cambridge University Press, 2nd edition (2003).
- 4) Kanatani, K.: Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis, *International Journal of Computer Vision*, Vol.80, No.2, pp.167–188 (2007).
- 5) McLauchlan, P.F.: A Batch/Recursive Algorithm for 3D Scene Reconstruction, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2738–2743 (2000).
- 6) Neyman, J. and Scott, E.: Consistent estimates based on partially consistent observations, *Econometrica*, Vol.16, No.1, pp.1–32 (1948).
- 7) Okatani, T. and Deguchi, K.: On the Wiberg Algorithm for Matrix Factorization in the Presence of Missing Components, *International Journal of Computer Vision*, Vol.72, No.3, pp. 329–337 (2007).
- 8) Okatani, T. and Deguchi, K.: On bias correction for geometric parameter estimation in computer vision, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2009).
- 9) Pollefeys, M., Koch, R. and Gool, L.V.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters, *International Journal of Computer Vision*, Vol.32, No.1, pp.7–25 (1999).
- 10) Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A.: Bundle Adjustment — A Modern Synthesis, *Vision Algorithm: Theory & Practice* (Triggs, B., Zisserman, A. and Szeliski, R., eds.), Springer-Verlag LNCS 1883 (2000).
- 11) 藤田 宏, 今野 浩, 田邊 士: 最適化法 (岩波講座 応用数学), 岩波書店 (1994).
- 12) 森 正武: 数値解析法 (朝倉現代物理学講座), 朝倉書店 (1984).
- 13) 竹村 彰通: 現代数理統計学, 創文社 (1991).