

植芝俊夫

Toshio UESHIBA

富田文明

Fumiaki TOMITA

## 1 はじめに

コンピュータビジョンにおける幾何学問題を解く際に、回転行列(行列式が1である3×3直交行列)を含む評価関数が与えられ、これを最小(大)化するような行列の値を求める必要がしばしば生じる。たとえば、複数画像間の特徴点対応から特徴点の3次元位置と各視点におけるカメラの位置・姿勢を求める structure from motion がその典型であり、bundle adjustment[1] と呼ばれる非線形最適化によってカメラの姿勢を表わす回転行列が他のパラメータと共に推定される。

回転行列は9つの成分を持つにもかかわらずその自由度は3であるので、3つの回転角(Euler角)や4元数によりパラメトライズして冗長性を除去し、このパラメータを反復的に更新して直交性を厳密に満たす回転行列を推定するのが一般的である。ところが、更新量の計算にはパラメータに関する評価関数の1次微分(gradient)が必要であるが、回転行列を陽にパラメトライズすると gradient は複雑な式となり、プログラムの実装が難しくなると共に、計算コストの点でも不利になる。さらに、Euler角によってパラメトライズした場合には、回転行列とパラメータが1対1に対応しない特異点が生じ、その近傍では更新量の計算が不安定になる。一方、4元数を用いた場合は、パラメータとなる4次元ベクトルの長さが1との拘束条件を課す必要があり、実装の複雑化を招く。

これらの問題を避けるために、本稿では回転行列を陽にパラメトライズすることなしに評価関数の gradient を計算する手法を提案する。また、直交性を厳密に満たしつつ回転行列を更新する方法についても述べる。これら2つのテクニックにより、回転行列を含む評価関数の最適化の簡便な実装が可能になる。提案手法のセルフキャリブレーションへの応用についても述べる。

## 2 回転行列を含む評価関数の非線形最適化

本稿では、回転行列を  $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$  のように3つの3次元列ベクトル  $\mathbf{r}_j$  で表わし、これを含む評価関数  $f(\mathbf{R})$  を最小(大)にする  $\mathbf{R}$  を求める最適化問題を考える。

以下の記法を用いる; 3次元ベクトル  $\mathbf{x} = [x_1, x_2, x_3]^T$  に対し、3×3反対称行列

$$[\mathbf{x}]_{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

を定義する。任意の3次元ベクトル  $\mathbf{x}, \mathbf{y}$  に対して  $[\mathbf{x}]_{\times} \mathbf{y} = \mathbf{x} \times \mathbf{y}$  が成り立つ。また、 $n$ 次元ベクトル  $\mathbf{x} = [x_1, \dots, x_n]^T$  の関数  $f(\mathbf{x})$  について、その gradient を  $\frac{\partial f}{\partial \mathbf{x}} = [\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}]$  なる行ベクトルで表わす。

産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology, AIST  
Tsukuba Central 2, Tsukuba, 305-8568 Japan

## 2.1 評価関数の gradient

回転行列の直交性  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$  より、 $\mathbf{R}$  の微小な変動  $\Delta\mathbf{R}$  に対して  $\Delta\mathbf{R}\mathbf{R}^T + \mathbf{R}\Delta\mathbf{R}^T = \mathbf{O}$  が1次近似の範囲で成り立つ。よって  $\mathbf{R}\Delta\mathbf{R}^T$  は3×3反対称行列であり、ある微小な3次元ベクトル  $\Delta\theta$  を用いて  $\mathbf{R}\Delta\mathbf{R}^T = [\Delta\theta]_{\times}$  と書ける。この  $\Delta\theta$  を用いれば、 $\Delta\mathbf{R}$  を

$$\begin{aligned} \Delta\mathbf{R} &= [\Delta\theta]_{\times}^T \mathbf{R} = -[\Delta\theta]_{\times} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{r}_1 \times \Delta\theta & \mathbf{r}_2 \times \Delta\theta & \mathbf{r}_3 \times \Delta\theta \end{bmatrix} \end{aligned}$$

と表せる。したがって、 $\mathbf{R}$  を  $\mathbf{R} + \Delta\mathbf{R}$  に変化させた時の  $f$  の増分は  $(\Delta\mathbf{R} = [\Delta\mathbf{r}_1 \ \Delta\mathbf{r}_2 \ \Delta\mathbf{r}_3])$  と書いて

$$\begin{aligned} f(\mathbf{R} + \Delta\mathbf{R}) - f(\mathbf{R}) &\approx \frac{\partial f}{\partial \mathbf{r}_1} \Delta\mathbf{r}_1 + \frac{\partial f}{\partial \mathbf{r}_2} \Delta\mathbf{r}_2 + \frac{\partial f}{\partial \mathbf{r}_3} \Delta\mathbf{r}_3 \\ &= \frac{\partial f}{\partial \mathbf{r}_1} (\mathbf{r}_1 \times \Delta\theta) + \frac{\partial f}{\partial \mathbf{r}_2} (\mathbf{r}_2 \times \Delta\theta) + \frac{\partial f}{\partial \mathbf{r}_3} (\mathbf{r}_3 \times \Delta\theta) \\ &= \underbrace{\left( \frac{\partial f}{\partial \mathbf{r}_1} [\mathbf{r}_1]_{\times} + \frac{\partial f}{\partial \mathbf{r}_2} [\mathbf{r}_2]_{\times} + \frac{\partial f}{\partial \mathbf{r}_3} [\mathbf{r}_3]_{\times} \right)}_{\frac{\partial f}{\partial \theta}} \Delta\theta \quad (1) \end{aligned}$$

となり、微小回転を表わすパラメータ  $\Delta\theta$  に関する  $f$  の gradient:  $\frac{\partial f}{\partial \theta}$  が求められる。

(1)式の  $\frac{\partial f}{\partial \theta}$  は、 $\mathbf{R}$  を構成する3つの列ベクトル  $\mathbf{r}_j$  とそれらに関する  $f$  の gradient のみから成るので、 $\mathbf{R}$  の現在値に対して  $\frac{\partial f}{\partial \theta}$  の値を計算するコードを容易に実装できる。

一方、 $\mathbf{R}$  をパラメータベクトル  $\mathbf{p}$  により  $\mathbf{R} = \mathbf{R}(\mathbf{p})$  とパラメトライズした場合には、gradient は

$$\frac{\partial f}{\partial \mathbf{p}} = \frac{\partial f}{\partial \mathbf{r}_1} \frac{\partial \mathbf{r}_1}{\partial \mathbf{p}} + \frac{\partial f}{\partial \mathbf{r}_2} \frac{\partial \mathbf{r}_2}{\partial \mathbf{p}} + \frac{\partial f}{\partial \mathbf{r}_3} \frac{\partial \mathbf{r}_3}{\partial \mathbf{p}}$$

となり、 $\frac{\partial f}{\partial \mathbf{r}_j}$  の他に3つの Jacobian:  $\frac{\partial \mathbf{r}_j}{\partial \mathbf{p}}$  を計算しなければならないので、実装が複雑になる。さらに、パラメトライズの方法によっては、特異点が生じて計算が不安定になったり(Euler角)、付加的な拘束が必要で実装が複雑化する(4元数)問題がある。

## 2.2 直交性を保存した回転行列の更新

評価関数の gradient が計算できれば、gradient からパラメータの更新量を求める一般的な手法(最急降下法、共役勾配法など)によって評価関数  $f$  を最適化できる。すなわち、 $\frac{\partial f}{\partial \theta}$  から  $\Delta\theta$  を計算して  $\mathbf{R} \leftarrow \mathbf{R} + \Delta\mathbf{R} = (\mathbf{I} - [\Delta\theta]_{\times})\mathbf{R}$  と更新することを  $f$  が収束するまで反復すればよい。

しかし、この更新ルールをそのまま適用すると、 $\mathbf{I} - [\Delta\theta]_{\times}$  が1次近似の範囲でしか直交性を満たさないため、更新後の  $\mathbf{R}$  は厳密には直交行列とならない。そ

ここで,  $\Delta\theta = \|\Delta\theta\|$ ,  $\mathbf{n} = \Delta\theta/\Delta\theta$  と書き, 1次近似の範囲で  $\cos\Delta\theta \approx 1$ ,  $\sin\Delta\theta \approx \Delta\theta$  であることに注意して,  $\mathbf{I} - [\Delta\theta]_{\times}$  を

$$\begin{aligned} \mathbf{I} - [\Delta\theta]_{\times} &= \mathbf{I} - [\mathbf{n}]_{\times} \Delta\theta \\ &\approx \underbrace{\mathbf{I} \cos\Delta\theta + \mathbf{nn}^T(1 - \cos\Delta\theta) - [\mathbf{n}]_{\times} \sin\Delta\theta}_{\mathbf{Q}^T} \end{aligned} \quad (2)$$

に置き換える. この  $\mathbf{Q}$  は, 単位ベクトル  $\mathbf{n}$  まわりの微小角  $\Delta\theta$  の回転を表わす厳密な直交行列である. これを用いて  $\mathbf{R} \leftarrow \mathbf{Q}^T \mathbf{R}$  とすれば, 直交性を保ちつつ  $\mathbf{R}$  を更新できる.

### 2.3 重み付き最小自乗問題への適用

コンピュータビジョンにおける幾何学問題では, 最適すべき評価関数が  $n$  次元ベクトル関数  $\mathbf{f}(\mathbf{R})$  の2次形式  $\mathbf{f}^T \mathbf{W} \mathbf{f}$  ( $\mathbf{W}$  は  $n \times n$  重み行列) となることが多い(重み付き最小自乗問題). この場合,

$$\left(\frac{\partial \mathbf{f}}{\partial \theta}\right)^T \mathbf{W} \left(\frac{\partial \mathbf{f}}{\partial \theta}\right) \Delta\theta = \left(\frac{\partial \mathbf{f}}{\partial \theta}\right)^T \mathbf{W} \mathbf{f} \quad (3)$$

を解いて更新量  $\Delta\theta$  を得る Newton 法あるいはその変形である Levenberg-Marquardt 法 [2] によって速い収束が得られる.

(3) 式に現れる  $\mathbf{f}$  の Jacobian:  $\frac{\partial \mathbf{f}}{\partial \theta}$  は,  $\mathbf{f}$  の各成分について gradient を (1) 式により計算し, それらを縦に並べて構成できる. また,  $\mathbf{R}$  の更新では (2) 式が使える. よって, §2.1, §2.2 で述べた手法は, 重み付き最小自乗問題にもそのまま適用できる.

### 3 セルフキャリブレーション問題への応用

重み付き最小自乗問題の例として, 複数画像間の特徴点対応のみからカメラの内部パラメータ, 外部パラメータ (位置と姿勢) および特徴点の3次元座標を一括推定するセルフキャリブレーションを取り上げる.

評価関数として, 推定パラメータを用いて計算される特徴点の投影像と実際に観測された像との距離の自乗和を用いた. 透視投影画像のための因子分解法 [3] によって求めた初期値を, この評価関数が最小になるよう Levenberg-Marquardt 法で更新してパラメータを推定した. カメラの内部パラメータは, 焦点距離を未知として推定対象に含めた他は, アスペクト比を1, 歪み角を0, principal point を画像の中央にそれぞれ固定した. また, 焦点距離は画像間で不変との条件を課した.

実験に用いた画像列を図1に示す. 20個の特徴点を選んで手動により画像間の対応付けを行い, これを入力データとしてセルフキャリブレーションを実行した. 推定されたカメラパラメータを用いてステレオビジョンにより復元されたシーンを, 図2に示す. 上面図と側面図からわかるように, 直角であるべき箇所がそのように復元されており, ユークリッド空間における3次元構造が正しく再構成されている.

さらに, 最適化の過程で評価関数が収束する様子を図3に示す. 見やすさのため, 評価関数の値は計算される特徴点像と実際に観測される像との距離の平均値で表示した. 4回の更新によって倍精度浮動小数点数の精度で最終的な値 (0.84pixel) に到達しており, 十分な収束速度が得られている.

また, 3次元位置が既知の参照パターンを用いたオフラインキャリブレーションによって求めた焦点距離

834pixel に対し, セルフキャリブレーションで推定された焦点距離は 817 pixel であった.

### 4 おわりに

評価関数の非線形最適化によって回転行列を推定する際に汎用的に用いることのできるテクニックを提案した. 具体的には, 回転行列を陽にパラメトライズすることなく評価関数の gradient を計算すること ((1) 式) と, 直交性を厳密に保ちつつ回転行列を更新すること ((2) 式) の2点である. gradient からパラメータの更新量を求めるタイプの非線形最適化手法にこれらを適用すると, プログラムへの最適化過程の実装が簡便になる. さらに特異点や付加的な拘束条件などのパラメトライズに伴う種々の問題を避けることができる.

### 参考文献

- [1] B. Triggs et. al. Bundle Adjustment – A Modern Synthesis. In *Vision Algorithms: Theory & Practice*, Springer-Verlag LNCS 1883, 2000.
- [2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
- [3] 植芝俊夫, 富田文明. 奥行パラメータの逐次推定による多視点透視投影画像のための因子分解法. 電子情報通信学会論文誌, Vol. J81-D-II, No. 8, pp. 1718–1726, 1998.

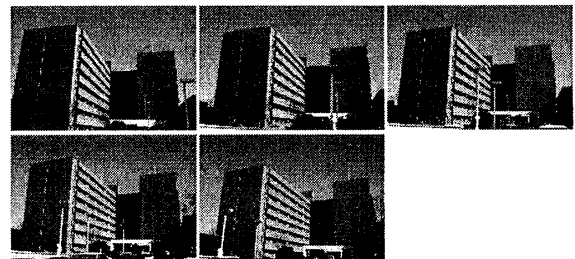


図1: 入力画像列

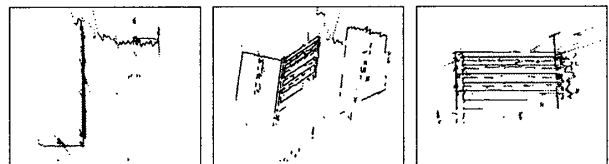


図2: シーンの3次元構造の復元

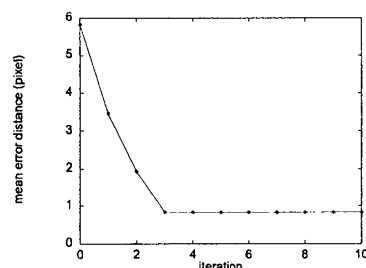


図3: 評価関数の収束の様子