計測自動制御学会東北支部 第205回研究集会 (2002.11.09)
資料番号 205-8

マルチカメラシステムでのセルフキャリブレーションの応用について

On Application of Self-calibration at Multicamera System

○久保田俊作*，岡谷貴之*，出口光一郎*

○Syunsaku Kubota*, Takayuki Okatani*, Koichiro Deguchi*

*東北大学

*Tohoku University

キーワード： マルチカメラシステム (Multicamera system)，セルフキャリブレーション(Self-calibration)，射影復元 (Projective reconstruction)，絶対円錐曲線 (Absolute conic)

連絡先： 〒982 仙台市青葉区青葉荒巻青葉01 東北大学 工学部 情報科学研究科 出口研究室
久保田俊作，Tel.: (022)217-7017，Fax.: (022)217-7015，E-mail: kubota@fractal.is.tohoku.ac.jp

1. 緒言

多視点カメラを用いて一つの三次元シーンを多視点から観察したり,またビデオカメラなどで移動しながら三次元シーンを撮影して，動的な状況を理解する研究が行われている．前者はマルチカメラシステム，後者はStructure from Motionと呼ばれている．このような多視点カメラシステムを有効に活用するためには，各視点でのカメラのパラメータを知ること，すなわちカメラキャリブレーションが重要である．

カメラキャリブレーションには，寸法や角度の正確に分かった参照物体を用意し，その物体の画像からカメラのパラメータを求める古典的な方法と，参照物体の詳しい情報を必要とせず，画像上の対応点のみからカメラのパラメータを推定するセルフキャリブレーションがある．近年，後者の研究がよく行われてきた[2]．

マルチカメラシステムを競技場などの大きな空間を観察する際に用いる研究も盛んに行われている[3) 4)]．こういった大きな空間では精度の問題から，必要とされる参照物体も大きくなくてはならず，参照物体の用意が難しくなる．その点で，セルフキャリブレーションはこういったマルチカメラシステムには向いていると言える．

セルフキャリブレーションは一般にはPollefeyの方法が良く知られている[1]．しかし，カメラの配置が球面上に乗っていて，カメラの光軸がある一点を向いているような配置ではPollefeyの方法は使えなくなることが研究により明らかとなった．このような配置はカーネギーメロン大学のVirtualized Reality[5]のようなドーム型のマルチカメラシステムでは問題となることが考えられます．本研究発表ではこの問題について発表していきます．

2. 原理

2.1 表記

三次元空間の点$\boldsymbol{M}_j = (X, Y, Z, 1)^T$が$i$番目のカメラの画像上の点$\boldsymbol{m}_{ij} = (x, y, 1)^T$に投影されたとき，両者は次のような関係で結ばれる．

$$\boldsymbol{m}_{ij} \propto \boldsymbol{P}_i \boldsymbol{M}_j \tag{1}$$

ここで$\boldsymbol{P}_i$は3×4行列で，カメラ行列と呼ぶ．この$\boldsymbol{P}_i$は，物理的には内部パラメータ$\boldsymbol{K}_i$と外部パラメータ$\boldsymbol{R}_i$,$\boldsymbol{t}_i$の積である：

$$\boldsymbol{P}_i = \boldsymbol{K}_i \left[\begin{array}{ccc} \boldsymbol{R}_i & | & -\boldsymbol{R}_i \, \boldsymbol{t}_i \end{array} \right] \tag{2}$$

ここで$\boldsymbol{K}_i$は次のようなカメラの内部パラメータからなる行列である．

$$\boldsymbol{K}_i = \begin{pmatrix} f_1 & s & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

2.2 バンドルアジャストメント

セルフキャリブレーションでは，カメラの内部パラメータや対象の三次元形状を決定するのに，最終的に非線形最適化を行う．例えば，(1)式のモデルでカメラ行列$\boldsymbol{P}$と三次元座標$\boldsymbol{M}$を求める時は，

$$J(\boldsymbol{P}_i, \boldsymbol{M}_j) = \sum_{i,j} |x_{ij} - \hat{x}(\boldsymbol{P}_i, \boldsymbol{M}_j)| + |y_{ij} - \hat{y}(\boldsymbol{P}_i, \boldsymbol{M}_j)| \tag{4}$$

の$J(\boldsymbol{P}_i, \boldsymbol{M}_j)$を最小にするような$\boldsymbol{P}_i, \boldsymbol{M}_j$の組を求める．また$(\hat{x}, \hat{y})$は，点$\boldsymbol{M}_j$を，$\boldsymbol{K}_i, \boldsymbol{R}_i, \boldsymbol{t}_i$を元に投影した画像の点の座標である．つまり，復元される画像$(\hat{x}, \hat{y})^T$が与えられた画像$(x, y)^T$に出来るだけ一致するような$\boldsymbol{P}_i, \boldsymbol{M}_j$の組を求めている．一般にノイズが含まれているとき，この$J(\boldsymbol{P}_i, \boldsymbol{M}_j)$は0には収束しない．収束後の$J(\boldsymbol{P}_i, \boldsymbol{M}_j)$の値を再投影誤差という．

この方法は繰り返し計算により行われるが，初期値は他の方法によって求める必要がある．このとき，正しい収束値を得るには当然正しい解に近い初期値が必要になる．

2.3 セルフキャリブレーションの原理

Pollefeyらが提案した一般的に最もよく知られているセルフキャリブレーションの方法について説明する．画像間での対応点が与えられた時の，一般的なセルフキャリブレーションの手順は，

- 対象とカメラのパラメータの射影復元を行う
- ユークリッド系へアップグレードする

というものである．これを順に見ていく．

2.3.1 射影復元

画像間での対応点を求め射影復元を行うことから始める．キャリブレーションされていないカメラからの画像からでも，射影復元を行うことは出来る．ただし，この復元は直行すべきものが直行しないような復元である．射影復元とは画像からカメラ行列$\boldsymbol{P}$と三次元点$\boldsymbol{M}$を取り出すことである．すなわち

$$\boldsymbol{m}_{ij} \propto \boldsymbol{P}_i \boldsymbol{M}_j = (\boldsymbol{P}_i \boldsymbol{H})(\boldsymbol{H}^{-1} \boldsymbol{M}_j) \tag{5}$$

という式になる．ここで，画像から取り出される$\boldsymbol{P}$と三次元点$\boldsymbol{M}$は何通りも存在する．この不定性を任意の4×4行列$\boldsymbol{H}$を使って表している．つまり，この式からは$\boldsymbol{P}_i$と$\boldsymbol{P}_i\boldsymbol{H}$，$\boldsymbol{M}_j$と$\boldsymbol{H}^{-1}\boldsymbol{M}_j$の区別がつかない．この$\boldsymbol{P}_i\boldsymbol{H}$，$\boldsymbol{H}^{-1}\boldsymbol{M}_j$を以後$\boldsymbol{P}_i^*, \boldsymbol{M}_j^*$と表記する．この不定性を除去することがセルフキャリブレーションの主要な問題であり，それには次の絶対円錐曲線の方程式が用いられる．

2.3.2 ユークリッド系へアップグレード

射影復元で得られた一番目の視点のカメラ行列を$\boldsymbol{P}_1^* = [\boldsymbol{I}|\boldsymbol{0}]$としたとき，その他の視点のカメラ

行列は次のような絶対円錐曲線の方程式を満たす.

$$\boldsymbol{K}_i\boldsymbol{K}_i^T \propto \boldsymbol{P}_i^* \begin{bmatrix} \boldsymbol{K}_1\boldsymbol{K}_1^T & -\boldsymbol{K}_1\boldsymbol{K}_1^T\boldsymbol{a} \\ -\boldsymbol{a}^T\boldsymbol{K}_1\boldsymbol{K}_1^T & \boldsymbol{a}^T\boldsymbol{K}_1\boldsymbol{K}_1^T\boldsymbol{a} \end{bmatrix} \boldsymbol{P}_i^{*T} \tag{6}$$

ここで求められる$\boldsymbol{K}_1, \boldsymbol{a}$より, 先程の行列$\boldsymbol{H}$を求めることが出来る[1]. これによってカメラ行列の射影的不定性を取り除き, 射影復元からユークリッド復元へ変換することが出来る. 最後に, $\boldsymbol{P}$を(2)式のように$\boldsymbol{K}, \boldsymbol{R}, \boldsymbol{t}$へ分解して終了する.

このとき内部パラメータのうち, ごく一部が既知であれば, 上の絶対円錐曲線の式は解ける. しかし, 実際には安定性を考えて, 内部パラメータのほとんどが既知で, 残りの一部(焦点距離等)のみを推定するのが現実的であり, それで十分実用的である. ユークリッド復元において, ほとんどのパラメータが既知としたカメラモデルは, 射影復元におけるカメラモデルよりも, 自由度が大幅に低くなる.

2.4 絶対円錐曲線の線形方程式

$\boldsymbol{P}_i^*$を用いて(6)式を解き, 変換行列Hを求める方法を記す. ここでは, (3)式のパラメータを$s=0$, $f_1 = f_2$とし, u_0, v_0を既知とする場合を考える. この場合, (6)式はある線形の方程式

$$\Omega(b_1, ..., b_5)^T = \boldsymbol{y} \quad (\boldsymbol{y} \neq \boldsymbol{0}) \tag{7}$$

に変換できる[1]. この線形の方程式を絶対円錐曲線の線形方程式と呼ぶこととする. この$\boldsymbol{b} = (b_1, ..., b_5)$は(6)式の右辺の真中の行列の要素を変数変換したもので, $\boldsymbol{K}_1, \boldsymbol{a}$からなる. この$\boldsymbol{b}$を求められれば, $\boldsymbol{K}_1, \boldsymbol{a}$も求まる. $\boldsymbol{K}, \boldsymbol{a}$に戻す過程で, $\sqrt{b_1}$の計算を行うが, $b_1 < 0.0$となることがある.

2.5 セルフキャリブレーションの破綻

Pollefeyらの提案した手法では, 先ほども述べたとおりカメラが球上に配置され全ての光軸が一点で交わるような場合には使えなくなる. これは具体的には前節の絶対円錐曲線の線形方程式のランクが落ちることが, 線型方程式の解が定まらなくなることが研究により明らかになった. これがそもそも引き起こされる原因は解析的にはまだ明らかになっていない. そこで, ここではシミュレーションによってこの現象を詳しく見ていくこととする.

3. シミュレーション実験

ドーム型マルチカメラシステムを想定したシミュレーション実験を行った. カメラは8台使用し, それぞれが同じ球面上に配置されている. また全てのカメラの光軸は球の中心の一点で交わっている. カメラは全て焦点距離400としている. これは市販されているビデオカメラの焦点距離が大体この値の近くだからである.

次に, 対象となる三次元点を球の中心にランダムな位置に15点作成する. 本実験では球の半径は1000とし, 三次元点の配置は300, 焦点距離は400としている. 全体的な距離感が想像しにくいが, これはカメラから見える三次元点の画像が300×300の画像領域に収まるように決定した値である. これは市販されているビデオカメラでは平均的な大きさである320×240と似た大きさになるように決定された結果である. つまり, このカメラシステムが想定しているのはドームの中心部に置かれたドームの半径と比較して3割程度の大きさの物体がなんとかカメラ全体で捉えられるようなシステムである.

このような条件で作られた画像でセルフキャリブレーションを行う. 全ての実験はこの過程を数百回行っている. もちろん, 三次元点の位置はランダムに決定されるため、毎回条件は異なる. 実験は主に、画像にノイズを加えていない時と、画像にノイズを加えた時に分けられる. ノイズを加える必要性については後述する. 最初の実験では

画像にノイズが含まれていない．これは数値計算にとって理想的な状態であり，普通であればセルフキャリブレーションは問題なく行えるはずである．しかし，カメラの配置はセルフキャリブレーションが破綻する配置となっているため，これは解けないことが予想される．

この次に，カメラにずれを与える．このずれによって，全てのカメラの光軸が一点で交わらなくなり，セルフキャリブレーションは破綻しなくなることが予想される．

次に画像にノイズを加える．実際のカメラ，特にデジタルカメラを使用する場合は画像にノイズが含まれることは防げない．つまり実験の一般性を失わないためにシミュレーションでも得られた画像にノイズを加える必要がある．ほとんどの数値計算では，ノイズが与えられる不安定になる．セルフキャリブレーションでもそれは例外ではないし、避けられない問題である．ノイズがあったなりにセルフキャリブレーションが解けていた場合、ノイズが無かった場合の焦点距離の値を中心にある一定の分布を示すことが予想される．ここではその分布が示されるかどうかを見る．また，ある大きさのノイズが加えられた時に，ずれの大きさに対して計算結果がどのようになるかも示す．

本実験で確かめることをまとめると次の点についてである

- ずれがないときは解けないことを明らかにする
- ノイズが無いとき，どの程度のずれで解けるようになるのかを確かめる
- ノイズがあるとき，計算精度とずれの大きさの関係を調べる
- ずれがどの程度，計算精度に影響を与えられるかを調べる

4. 結言

半球状のドームにカメラが配置されているようなマルチカメラシステムでは，画像にノイズが存在しない状況でも，一般に良く知られたセルフキャリブレーションは行えないことを示した．これはカメラの位置をわずかにずらし、カメラが中心を向かないようにすることで解決できる．しかし，画像にノイズが無い場合は一般的には存在せず，ずらす量が十分でないとノイズに対して非常に計算が不安定になる．セルフキャリブレーションを使用することを前提としたマルチカメラシステムでは，設計する際に想定されるノイズの大きさを考えた上でカメラ配置を決めなくてはならない．

今後の研究で，絶対円錐曲線の線型方程式が解けなくなる理由を解析的に求めたい．

参考文献

1) M. Pollefeys. : "Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters", International Journal of Computer Vision 32(1). pp. 7-25 (1999)

2) R. I. Hartley and A. Zisserman. : "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

3) 矢口悟志，木村誠，斎藤英雄，金出武雄．："未構成多視点カメラシステムを用いた任意視点画像生成"，情報処理学会 CVIM研究会論文誌，Vol. 42, pp.9-21，2001．

4) 斎藤英雄，木村誠，矢口悟志，稲本菜穂．："多視点映像による現実シーンの仮想化"，情報処理学会 CVIM研究会報告 131-8，pp.53-60，2002．

5) http://www2.cs.cmu.edu/~virtualized-reality/