

モバイルプロジェクタのための プロジェクタカメラシステムのオンライン自動校正

岡谷 貴之, 阿部 大輔, 琢磨 直弘, 出口 光一郎^{†1}

本稿では、モバイルプロジェクタから平面スクリーンに像を投影し、その像を固定カメラが常時捉えるような構成のプロジェクタカメラシステムに対する校正法を述べる。応用の1つは、手で持った小型プロジェクタでスクリーンに像を投影するとき、プロジェクタの姿勢によらず常にスクリーンの決まった場所に、幾何学的に補正された画像が表示されるようなシステムである。提案方法は、投影面上の(人工)マーカーを前提とせず、校正用の特別な画像を投影するなどといった、オフラインでの校正手続きなしに、自動的にシステムを校正する。このため、ユーザは見たい映像を投影しながらただプロジェクタを動かすだけでよい。この間、プロジェクタからスクリーン上に投影される画像と、これをカメラで撮影した画像のペアから、カメラ画像とスクリーン間の射影変換がオンラインで推定される。このアプローチは、投影面上のマーカーならびに校正専用の画像を使わないことから、高精度を確保するのが難しい。これを解決するため、逐次バンドル調整に基づく校正方法を示す。この方法は、オンライン校正の性質、すなわち観測データを時々刻々蓄積でき、故に大量のデータが利用可能となることを利用して、計算量と記憶域を最小化しつつ高精度の校正を実現する。

Calibration of a Projector-Camera System for Mobile Projectors

TAKAYUKI OKATANI, DAISUKE ABE, NAOHIRO TAKUMA,
KOICHIRO DEGUCHI^{†1}

This paper proposes a calibration method for a projector-camera system consisting of a mobile projector and a stationary camera. As a target application, we consider a system in which rectified images are always displayed in a fixed position of a planar screen, no matter how the projector is moved. The proposed method automatically calibrates the system without using fiducials on the screen or a special calibration procedure in which dedicated images for calibration are used. Owing to this automatic nature, the user needs only to arbitrarily move the projector by his/her hand. While the user is moving the projector, from

the pairs of an image to be projected from the projector and the image of the projected image on the screen that is captured by the stationary camera, the projective transformation between the camera and the screen is estimated in an online manner.

1. はじめに

近年、プロジェクタとカメラを組み合わせたプロジェクタカメラシステムが盛んに研究されている^{1),8)-10)}。プロジェクタが投影した画像をカメラで撮影し、そこから取り出した情報をプロジェクタ側にフィードバックすることで多様な目的が実現される。例えば、単体のプロジェクタの投影像の幾何学的・光学的補正や、複数プロジェクタによる高解像度画像投影の実現を始めとして、様々な用途がある。

最近、プロジェクタの小型化が著しく、手のひらに収まる程度の小さなサイズのプロジェクタ、いわゆるモバイルプロジェクタ(携帯プロジェクタ)が次々と市販されている(図1)。単体のプロジェクタとしての利用のみならず、携帯電話等のモバイル情報端末等への搭載も想定されている。これらの小型プロジェクタは、プロジェクタがもつ他の映像表示機器にはない特筆すべき性質—すなわち、プロジェクタ自体の物理的制約とほぼ無関係に、投影画像の位置やサイズを柔軟に設定でき、さらに、物体表面が拡散反射成分を一定程度有すれば(投影専用のスクリーンでなくても)そこに映像を投影できること—をいかに発揮できる。例えば、壁面等に画像を投影し、メールや動画等を大画面で見たり、物体の見え



図1 モバイルプロジェクタの例。

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

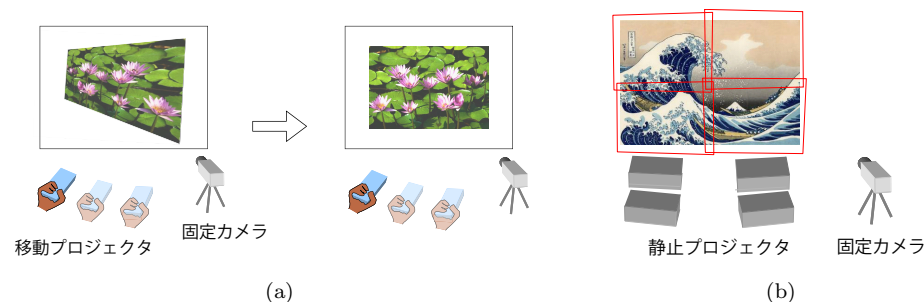


図 2 (a) モバイルプロジェクタの表示画像の正規化．(b) 複数プロジェクタを使った高解像度画像投影．

に重ねこむ形で像を投影し、仮想的な見えを実現するアプリケーションにおいて、プロジェクタカメラシステムのさらなる発展が期待される．

このような今後の発展も含め、プロジェクタカメラシステムの多くの応用において中心的課題の一つとなるのが、システムのキャリブレーションである．本稿では、モバイルプロジェクタを使ったシステムおよびその校正の具体例として、図 2(a) のように、一台の運動するプロジェクタと固定カメラおよび平面スクリーンからなるシステムにおいて、小型プロジェクタを手で持って映像を投影する際、スクリーン上の決まった位置に、常に正しく正規化された画像が表示されるようにすることを考える．「正しく正規化」とは、画像に幾何学的な歪（台形歪他）がなく、正しいアスペクト比が再現できていることを指す^{*1}．

本稿ではこのようなシステムに対し、ユーザの負担を最小限とできるような広い応用範囲を持つ校正法を提案する．その方法は、投影面上に人工的なマーカなどの目印を必要としない．スクリーンの境界などの「自然な」目印も不要である．上述のようなモバイルプロジェクタの応用を考える場合、このように投影面上の目印が不要なことは重要である．また提案方法は、ユーザがプロジェクタで見たい映像を投影している間に、オンラインでシステムを校正する．そこでは、校正用の特別な画像を投影するなどといった、オフラインでの校正手続きは不要である．ユーザは具体的には以下を行えばよい．

1. 平面スクリーンに対し適当にカメラを置く（配置の慎重な調整は不要）．
2. 手で持ったモバイルプロジェクタから画像を投影し、プロジェクタを自由に動かす．

*1 なお、この校正の問題は、同図 (b) に示すような、複数のプロジェクタを使って 1 枚の高解像度の画像を投影するシステムの校正^{5),7)} と、多くの部分で共有している．本稿の研究も、そのような問題に対するわれわれの以前の研究³⁾⁻⁵⁾ に基礎を置いている．

3. ステップ 2 でプロジェクタを動かしているうちに、校正が完了する．いったん校正されれば、図 2(a) のように、正規化された画像を、プロジェクタの姿勢に関係なく常に表示できるようになる．

このように校正が簡単にできれば、システムを利用しやすくなり、その応用範囲を拡大できるだろう一方で、その簡単さを下支えする上述の 2 つの要素— マーカーを要しないことと、校正専用の画像でなくて自然画像を校正に用いること— は問題をより難しいものとする．とりわけ、この 2 つを両立するのは簡単ではない．

スクリーン上にマーカを必要としない校正方法は、問題の解法そのものは論文³⁾ にて明らかにされているものの、十分な精度を実現するのは難しい．これは人工マーカを用いない校正では、カメラの自己校正⁶⁾ の場合と同様、多数の未知数を決定する必要があることによる．また、校正専用ではない自然画像を使用することによって、データの質が悪くなり、精度を低下させる．現に図 2(b) に示したような複数の静止プロジェクタの校正では、校正専用の画像を使い、校正をオフラインで行うことで精度を確保している^{4),5),7)} ．

このような校正精度に関わる困難さを解決するため、われわれはオンラインで校正を行うが故の利点を利用する．つまり、オンラインで時々刻々得られる観測を蓄積してゆけば、校正のためのデータを大量に得られる．これを利用すれば、データの質の低いに由来する困難さを克服できる可能性がある．

ただし、大量にデータを得て、これからパラメータの計算を行おうとすれば、計算量と記憶域が大きくなる別の問題が生じる．過去の研究^{3),5),7)} に見られるように、推定精度の観点で最良の方法は、全データを用いて全パラメータを同時に推定するバンドル調整である．この方法は時にバッチバンドル調整 (batch bundle adjustment) と呼ばれるが、データ量が大きく、したがってパラメータ数も大きいときは実用的でない．オンラインでの推定を想定すれば、データの到着につれて、少ない計算量でパラメータの精度を少しずつ改善してゆけることが望ましい．逐次バンドル調整 (sequential bundle adjustment) と呼ばれるそのような方法が、SFM (structure from motion) のために過去に開発されている^{2),11),12)} ．そこで本稿では、同じ考え方を、プロジェクタカメラシステムの校正に応用し、ここでの問題の構造を最大限に利用したアルゴリズムを導出する．なお本稿では紙面の都合から、校正方法およびその数値計算の方法の原理の記述に重点を置く．プロジェクタの画像とカメラの画像間での高速な点対応の算出や高速な画像の変形など、それ以外の課題については省略する．

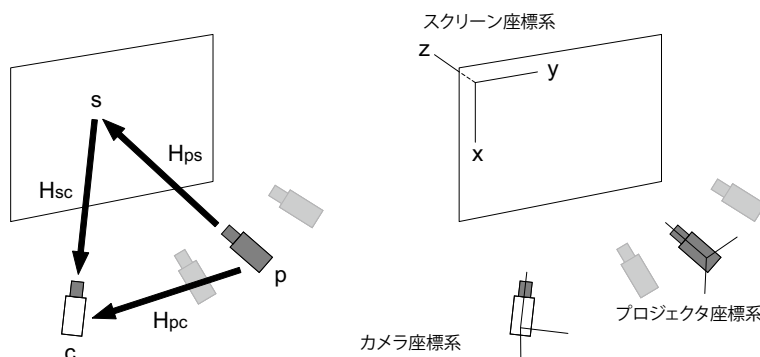


図 3 左：平面スクリーン，カメラの画像面およびプロジェクタの画像面の間で定義される 3 つの 2 次元射影変換．
右：議論で使用する 3 種類の座標系．詳しくは本文参照．

2. 平面を投影面とするプロジェクタカメラシステムの校正

2.1 プロジェクタカメラシステムの幾何

まず最初に，われわれの過去の研究³⁾に基づいて，上述の校正方法を実現するのに解くべき問題を定式化する．

先述の通り本稿では，平面スクリーンと 1 台の固定カメラおよび単一の移動プロジェクタからなるシステムを考える．移動プロジェクタの各姿勢を $p = 1, \dots$ によって表す．図 3 のように，プロジェクタ，カメラおよびスクリーンのそれぞれに 3 次元座標系をとる．プロジェクタ座標系およびカメラ座標系は，プロジェクタおよびカメラの光学系を基準とし，通常通りの方法で定義する．スクリーン座標系は，その xy 平面がスクリーン平面に一致するようにとる．つまりその z 軸は，自動的にスクリーン平面と垂直となる．以上に加えて，プロジェクタおよびカメラに対してそれぞれ 2 次元画像座標系をとり，プロジェクタ画像座標系，カメラ画像座標系と呼ぶことにする．

プロジェクタの画像，それが投影されるスクリーン平面，およびそれを撮影したカメラ画像の 3 つの平面の間には，2 次元射影変換が存在する．プロジェクタの画像上の点を u ，この点がスクリーン上に投影される点を w ，そのカメラ画像上の点を v と書き，それぞれ，3 つの対応する座標系（プロジェクタ画像座標，スクリーン座標の xy 座標，カメラ画像座標）における同次座標を保持したベクトルとする．これらは図 3 に示すように，次の 3 つ

の 2 次元射影変換で相互に変換される：

$$w \propto H_{ps} u, \quad (1a)$$

$$v \propto H_{sc} w, \quad (1b)$$

$$u \propto H_{pc} v, \quad (1c)$$

ここで \propto は定数倍を除いて両辺のベクトルが等しいことを表す．この 3 つの式から，これら 3 つの射影変換の間には次の関係が成り立つ：

$$H_{ps} \propto H_{sc}^{-1} H_{pc}. \quad (2)$$

H_{ps} は，スクリーンとプロジェクタが作るシステムをスクリーンを画像面とする擬似的なカメラと見なすことにより，次のようなスクリーンに対するプロジェクタの姿勢および内部パラメータを表す 3 つの 3×3 行列の積に分解できる³⁾：

$$H_{ps} \propto T_p R_p K_p^{-1}, \quad (3)$$

ここで T_p は，上述の擬似カメラの内部行列に相当し，プロジェクタ p の位置を表すスクリーン座標 $[x_p, y_p, z_p]^T$ を次のように並べて出来る：

$$T_p = \begin{bmatrix} z_p & 0 & x_p \\ 0 & z_p & y_p \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

また R_p は，プロジェクタ座標からスクリーン座標への座標変換の回転成分であり，このプロジェクタの姿勢の回転成分を表す． K_p はこのプロジェクタの内部行列を表す．

2.2 システムの校正と H_{sc} の決定

今考えているプロジェクタカメラシステムを校正するとは，以下に述べるように， H_{sc} を決定することに他ならない．

まず，最新の姿勢 p での H_{ps} が分かればスクリーン上の表示画像を正規化できる．なぜなら， H_{ps} はプロジェクタ画像とその投影像間の直接的な関係を与えるからである．スクリーン上に投影すべき望みの画像を決めれば，それを実現するようなプロジェクタ画像を生成できる．

校正に利用できる観測データは，カメラによって撮影したプロジェクタからの投影像およ

び、投影前の各プロジェクタへ入力した画像である。以降、前者をカメラ画像、後者をプロジェクタ画像と呼ぶ。上述の3つの射影変換のうち、 H_{pc} は、カメラ画像とプロジェクタ画像間で、4点以上の複数の点の対応関係 ($u \leftrightarrow v$) を見つけられれば計算できる。ここでは、プロジェクタ画像において特徴点を抽出し、その対応点をカメラ画像において探索することで、そのような対応は常に得られるものとする。このようにして、各姿勢 p の H_{pc} は得られる。

ここで (2) 式の関係、 $H_{ps} \propto H_{sc}^{-1} H_{pc}$ を使うと、ひとたび射影変換 H_{sc} を知ることが出来れば、上のように得られる H_{pc} から、望みの H_{ps} を計算できることが分かる。今カメラおよびスクリーンが固定であると仮定しているので、 H_{sc} はプロジェクタの姿勢数に関係なく (当然) 一つである。

なお、スクリーン上にその座標が既知である目印 (マーカー) がある場合には、 H_{sc} はそれを用いて直接的に推定できる。具体的には、そのような点が4点以上あって、それらをカメラの画像上で同定できるとき、これは可能である。しかしながら、部屋の壁など専用の投影スクリーンでない平面上への投影を考えると、そのような目印の存在を前提とすることは応用の範囲を狭める。

2.3 投影面上のマーカーを要しない校正

そのようなマーカー (目印) を必要としない校正方法が、論文³⁾にある。そこでは、プロジェクタが校正済み (内部パラメータがすべて既知) のとき、プロジェクタ画像とカメラ画像の組だけから、 H_{sc} を閉形式のアルゴリズムで計算できることが示されている。われわれは、プロジェクタ画像とカメラ画像の組だけを用いる点で同じ戦略を採用し、 H_{sc} を決定する。プロジェクタは一部未校正でも構わない。

論文³⁾の結果は次のように要約できる。 K_p が既知であるとき、プロジェクタ画像とカメラ画像の組を2つ ($p = 1, 2$) 用意すれば、そこから H_{sc} を8通りの不定性を残して決定できる。さらに1つ以上の画像の組 ($p = 1, \dots$) を用いれば、それら8つの解を2つにまで絞り込むことができる。この2種類の解の不定性は、プロジェクタがスクリーンを挟んでカメラと同じ側に位置するかそうでないかに相当し、これら不定性は解決できない。

さらに、 H_{sc} にはさらなる内部的な不定性がある。それは、 H_{sc} は、2次元射影変換として元々8つの自由度を持つが、そのうち4つだけを決定できるということである。決まらない4つの ($= 8 - 4$) 自由度は、スクリーン上にスクリーン座標系を定義する自由度に対応する。この自由度はスクリーン上に投影された画像に、相似変換 (並進, 回転, スケール倍) を加えることと対応する。この不定性は、バンドル調整を実装するにあたってパラメータ

表現を選ぶ際、適切に拘束する必要がある。

プロジェクタの内部パラメータがすべて未知であれば、 H_{sc} は決定できない。プロジェクタが部分的に校正済みの場合、具体的には内部パラメータ (K_p) のうちで1要素のみが未知で、それ以外はすべて既知の場合には、 H_{sc} は決定できる可能性がある。論文⁵⁾では、複数の静止プロジェクタからなるシステムを対象に、この問題を取り扱った。実用上特に重要なのは、プロジェクタの焦点距離が未知であり、かつ姿勢ごとにこれが変化する場合である。

このような部分的に校正済みの場合、プロジェクタ画像とカメラ画像間の対応関係のみから H_{sc} を閉形式にて計算する方法は見つかっていない。そこで、バンドル調整すなわち非線形最小化を用いる。プロジェクタが完全に校正済みの場合でも、最大限の精度を得るにはバンドル調整を使用すべきである。

3. 逐次バンドル調整

前節で述べたように、パラメータの値はバンドル調整によって計算するが、ここではオンラインでの校正を可能にする逐次バンドル調整の方法を述べる。

3.1 バッチバンドル調整

まず通常のバンドル調整、すなわち全観測を一度に用いて最適解を計算する方法を要約しておく。プロジェクタ p の画像上の i 番目の特徴点を u_{pi} と書く。これと対応するカメラの画像上の点を v_{pi} とする。以下では \sim は同次ベクトルを非同次化する演算子として用いる。すなわち、 $v = [v_1, v_2, v_3]^T$ に対して、 $\tilde{v} \equiv [v_1/v_3, v_2/v_3]^T$ である。

特徴点の観測 v_{pi} に対し、その推定値を \hat{v}_{pi} と書く。これは、推定すべき未知パラメータ H_{sc}, T_p, R_p および K_p を指定したとき、次のように算出される：

$$\hat{v}_{pi} \propto H_{sc} H_{ps} u_{pi} = H_{sc} T_p R_p K_p^{-1} u_{pi}. \quad (5)$$

次に、カメラの画像上での観測された点の位置とその推定との間で二乗距離を求め、これの全プロジェクタ姿勢および特徴点にわたる総和を定義する。これを

$$E_{1:m}(\mathbf{x}) = \sum_{p=1}^m E_p(\mathbf{x}), \quad (6)$$

と書く．ただし E_p はプロジェクタ p に関わる全特徴点に関する総和で，

$$E_p(\mathbf{x}) = \sum_{i=1}^n \left(\tilde{\mathbf{v}}_{pi} - \hat{\mathbf{v}}_{pi} \right)^2, \quad (7)$$

のように与えられる．ここで \mathbf{x} は，求めるべき未知パラメータを並べてできるベクトルである．具体的には， H_{sc} および T_p と R_p （さらに場合によっては K_p の一部）をそれぞれ適切にパラメータ表現し，それらパラメータを並べてできるベクトルである．なお H_{sc} のパラメータ表現の方法は後述する． $E_{1:m}(\mathbf{x})$ を最小化する \mathbf{x} が求めるべき解となる．

コスト関数 $E_{1:m}$ はニュートン法に基づく方法によって最小化する．最適解は，適当な \mathbf{x} の初期値 から開始し，これを $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x}$ のように更新することを繰り返して探索するが，その更新量 $\delta\mathbf{x}$ は

$$A\delta\mathbf{x} = \mathbf{a}, \quad (8)$$

の解として定める．ただし A および \mathbf{a} は，ガウス・ニュートン法では

$$A = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}} \right)^\top \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}} \right), \quad \mathbf{a} = - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}} \right)^\top \mathbf{f}, \quad (9)$$

のように与えられる．この \mathbf{f} は，再投影誤差を次のように書き換えたときのベクトルである：

$$E_{1:m} = \frac{1}{2} \mathbf{f}^\top \mathbf{f}. \quad (10)$$

またレベンバーグ・マーカート法を採用する場合には， A に対角行列 λI を加え，スカラー値 λ を適切に調整する¹²⁾．

3.1.1 H_{sc} のパラメータ表現について

先述した H_{sc} が内包する不定性を拘束するために，次のようなパラメータ表現を採用する．この不定性は先述の通り，スクリーン座標系を定義する際に生じる自由度を反映したものである．そこでその自由度を除去するのに，スクリーン座標系をカメラの画像を経由して間接的に定義する．そのため，カメラの画像上で適当な 2 つの点を選び，そのカメラ画像座標を $[\alpha_1, \beta_1]$ および $[\alpha_2, \beta_2]$ とする．そして，この 2 点に対応するスクリーン上の点のスクリーン座標がそれぞれ $[0, 0]$ と $[1, 0]$ になるとする（あるいはそのようにスクリーン座標系を定義すると考えても良い）．これら 2 点は H_{sc} を， $H_{sc}[0, 0, 1]^\top \propto [\alpha_1, \beta_1, 1]^\top$ および $H_{sc}[1, 0, 1]^\top \propto [\alpha_2, \beta_2, 1]^\top$ のように拘束する．これら 2 つの式を満たす H_{sc} は，4 つの

パラメータ x_1, x_2, x_3 および x_4 によって次のように表現できる：

$$H_{sc} = \begin{bmatrix} \alpha_2(x_1 + 1) - \alpha_1 & x_2 & \alpha_1 \\ \beta_2(x_1 + 1) - \beta_1 & x_3 & \beta_1 \\ x_1 & x_4 & 1 \end{bmatrix}. \quad (11)$$

このように定義した x_1, \dots, x_4 の他にも，個別のプロジェクタの姿勢他一部の内部パラメータが未知数となる．具体的には姿勢の位置および向きと（必要な場合には）焦点距離である．これらは通常のやり方でパラメータ表現する．こうして， x_1, \dots, x_4 を \mathbf{x} の先頭に並べ，その後プロジェクタの個別のパラメータを並べて \mathbf{x} を構成する．

3.2 パラメータの削除

逐次バンドル調整は，入ってくる観測データをすべて推定に用いながらパラメータの数を一定に保つ点で，バッチバンドル調整と異なる．新しい観測に伴ってほとんどの場合に新しいパラメータが生じるので，過去の観測に関連するパラメータを削除できることが求められる．

$E_{1:m}(\mathbf{x})$ を \mathbf{x} に関して最小化するバッチバンドル調整を再度考える．上述のように，パラメータ \mathbf{x} は 2 つの部分からなり，一つはプロジェクタの全姿勢共通の H_{sc} （と焦点距離が全姿勢共通で未知の場合，焦点距離も）であり，もう一つは各姿勢固有のパラメータ（プロジェクタの位置と向き，さらに各姿勢で焦点距離が変動する場合には焦点距離）である．前者の共通パラメータを \mathbf{x}_1 ，後者の各姿勢ごとのパラメータ（を m 姿勢分まとめたもの）を \mathbf{x}_2 と書き， $\mathbf{x}^\top = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$ とする．先述の通り，われわれは， H_{sc} を中心とする共通パラメータすなわち \mathbf{x}_1 を推定することに関心がある^{*1}（この意味で，各プロジェクタの姿勢すなわち \mathbf{x}_2 は，かく乱母数 (nuisance parameter) である）．したがって，プロジェクタの新しい姿勢に伴う新しい観測が到着するごとに， \mathbf{x}_1 を継続的にリファインすることができれば十分である．以下，そのような方法を設計する．

今 $E_{1:m}$ の最小化をちょうど完了したところだとし， \mathbf{x}^* を見つけた最適解だとする．このとき，微小な変動 $\delta\mathbf{x}$ に対し，コスト $E_{1:m}(\mathbf{x})$ は， \mathbf{x}^* のまわりで次のように近似される：

$$E_{1:m}(\mathbf{x}^* + \delta\mathbf{x}) \approx \frac{1}{2} \delta\mathbf{x}^\top A \delta\mathbf{x} + \text{const}. \quad (12)$$

ここで， A は \mathbf{x}^* で評価した (9) 式の行列である．この近似コストを $\hat{E}_{1:m}$ と書く．行列 A

*1 校正後に画像を補正する段階では，最新のプロジェクタ姿勢の推定も必要となる．

を上述のパラメータの分割 $\mathbf{x}^\top = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$ にあわせて分割すれば、 $\hat{E}_{1:m}$ は、次のように書きなおせる．

$$\hat{E}_{1:m} = \frac{1}{2} [\delta \mathbf{x}_1^\top \ \delta \mathbf{x}_2^\top] \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_1 \\ \delta \mathbf{x}_2 \end{bmatrix} + \text{const.} \quad (13)$$

上述のように、 \mathbf{x} のうち \mathbf{x}_1 の部分のみ、推定することに關心がある．そこで、上の式から $\delta \mathbf{x}_2$ を消去し、 $\delta \mathbf{x}_1$ のみの関数に書き換えたい． $\delta \mathbf{x}_1$ を与えたとき、近似コスト $\hat{E}_{1:m}$ を $\delta \mathbf{x}_2$ について最小化することを考えれば、 $\delta \mathbf{x}_1$ が固定されているときに $\hat{E}_{1:m}$ を最小化する $\delta \mathbf{x}_2$ は $\delta \mathbf{x}_2 = -\mathbf{A}_{22}^{-1} \mathbf{A}_{21} \delta \mathbf{x}_1$ のように与えられる．この式は、 $\partial((13) \text{ 式の左辺}) / \partial \delta \mathbf{x}_2 = 0$ を $\delta \mathbf{x}_2$ について解くことで導出される．これを (13) 式に代入すると

$$\hat{E}_{1:m} = \frac{1}{2} \delta \mathbf{x}_1^\top (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21}) \delta \mathbf{x}_1 + \text{const.} \quad (14)$$

を得る．この式は、上のように \mathbf{x}_2 を自動的に決定されるメカニズムの下で、元のコスト $E_{1:m}$ をその最小点 \mathbf{x}_1^* まわりの微小変動 $\delta \mathbf{x}_1$ について近似したものとなっている．この近似コスト $\hat{E}_{1:m}$ の最小点まわりの形状を表現した行列を

$$\mathbf{A}_{11}^* \equiv \mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21} \quad (15)$$

と定義する．以上の考察に基づいて、新しい観測が到着したとき \mathbf{x}_2 は陽には更新しないことにし、 \mathbf{x}_1 のみを修正する．その後、 \mathbf{A}_{11}^* および \mathbf{x}_1^* を将来に伝達する．

3.3 新しい観測の取り扱い

今、新たな姿勢 $m+1$ が追加されたとする（1枚の画像が取得される）．この新しい姿勢を記述する新たなパラメータを \mathbf{x}'_2 と書く．上述のように、プロジェクト姿勢 m までの個々のプロジェクトのパラメータは既に取り除いてしまっているので、 $[\mathbf{x}_1^\top, \mathbf{x}'_2{}^\top]^\top$ が推定すべきパラメータとなる．

新たな姿勢 $m+1$ が追加されたとき、バッチバンドル調整のコスト $E_{1:m+1}$ は

$$E_{1:m+1} = E_{1:m} + E_{m+1}. \quad (16)$$

のように更新される． $E_{1:m}$ を最小化する \mathbf{x}_1^* の精度が十分よい（すなわち、その真値とのずれが小さい）と仮定すると、(14) 式の $\hat{E}_{1:m}$ で $E_{1:m}$ を置き換えることで、 $E_{1:m+1}$ は

$$E_{1:m+1} \approx \frac{1}{2} (\mathbf{x}_1 - \mathbf{x}_1^*)^\top \mathbf{A}_{11}^* (\mathbf{x}_1 - \mathbf{x}_1^*) + E_{m+1}. \quad (17)$$

と近似できる．次に、式の右辺を $[\mathbf{x}_1^\top, \mathbf{x}'_2{}^\top]^\top$ について最小化することを考え、この最小化を果たすための $[\mathbf{x}_1^\top, \mathbf{x}'_2{}^\top]^\top$ から $[(\mathbf{x}_1 + \delta \mathbf{x}_1)^\top, (\mathbf{x}'_2 + \delta \mathbf{x}'_2)^\top]^\top$ への更新則を導く． E_{m+1} のみを $[\mathbf{x}_1^\top, \mathbf{x}'_2{}^\top]^\top$ について最小化する場合の更新則を

$$\begin{bmatrix} \mathbf{A}'_{11} & \mathbf{A}'_{12} \\ \mathbf{A}'_{21} & \mathbf{A}'_{22} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_1 \\ \delta \mathbf{x}'_2 \end{bmatrix} = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \end{bmatrix}. \quad (18)$$

をのように書く．これを用いると、(17) 式の右辺（2つの項の和）を最小化するための更新則は

$$\begin{bmatrix} \mathbf{A}_{11}^* + \mathbf{A}'_{11} & \mathbf{A}'_{12} \\ \mathbf{A}'_{21} & \mathbf{A}'_{22} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_1 \\ \delta \mathbf{x}'_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_{11}^* (\mathbf{x}_1 - \mathbf{x}_1^*) + \mathbf{a}'_1 \\ \mathbf{a}'_2 \end{bmatrix}. \quad (19)$$

のように書ける．

この更新を繰り返し、最適解を探索する．解が見つかったら、3.2節の方法を適用し \mathbf{x}'_2 を削除する．そして \mathbf{A}^* が更新され、新たな解 \mathbf{x}^* とともに将来に伝達される．この一連の手続きを、新たな観測が到着する都度毎回実行する．これがわれわれの問題に対する逐次バンドル調整の方法である．

4. 実験結果

前節のアルゴリズムの性能を評価するため、合成データを使うシミュレーション実験を行った．スクリーンに対して、カメラを適当な位置に固定し、プロジェクトは各姿勢をランダムに生成した．生成したプロジェクト姿勢20個を、スクリーンとカメラの配置とともに表示したものが図4(a)である．これらプロジェクト姿勢に対し、その投影像をカメラから見たときの画像例（10個分）を図4(b)に示す．複数の四角形は、スクリーン上のプロジェクト投影像の枠の、カメラ画像上の像を示す．プロジェクトおよびカメラの画像は 640×480 画素とした．プロジェクトは校正済みとした（カメラは未校正である）．

このような設定の下で、プロジェクト画像とカメラ画像のペアに対し、その間に n 個の点対応を与え、これを用いて校正を行った．点対応は、まずプロジェクト画像上で n 個の点を画像上、一様ランダムに発生させ、それらのカメラ画像上の対応点を計算した後、それらの座標に、カメラ画像上での特徴点検出をシミュレートするガウス性誤差（分布の形は $N(0, \sigma^2 \mathbf{I}_2)$ ）を加算して得た．

以上のように作成したデータと用いて、逐次バッチバンドル調整の提案アルゴリズムを、

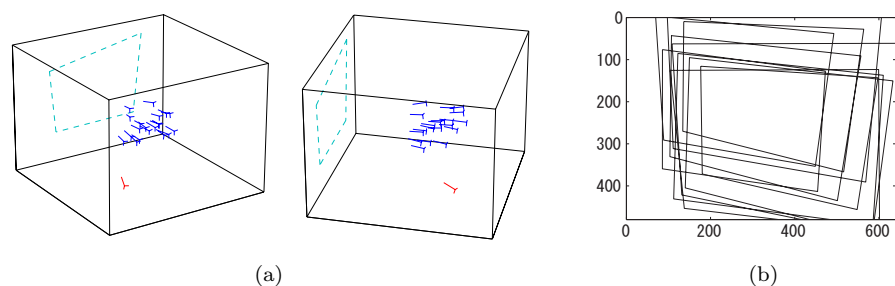


図 4 (a) 実験で使ったスクリーン, カメラ (赤線), プロジェクタ姿勢の一部 (青線). 破線で描かれた四角形はスクリーン上のカメラの視野を表す. (b) カメラ画像に現れるプロジェクタ画像境界の例.

計算量と校正精度の 2 点から, バッチバンドル調整と比較した. いずれのアルゴリズムの実装においても, (8) 式の更新則はガウス・ニュートン法とした (われわれの実験では, レベンバーグ・マーカート法よりも収束が速かった). なおバッチバンドル調整の実装では, やはりガウス・ニュートン法を採用し, その更新ステップでの線形方程式の求解においては, 行列 A を疎行列として適切に扱い, 余分な計算をしない方法をとった.

まず, 計算時間を調べた. 同一データ系列に対する, 両アルゴリズムの計算時間を図 5 に示す. 実験では Intel Core2 2.66GHz を CPU とする PC 上の MATLAB 7.1 を使って行った. 画像 1 枚あたりの特徴点数を 100 とし, 観測誤差は $\sigma = 0.5$ 画素に設定した. 図は, 画像 1 枚 (プロジェクタの 1 姿勢に対応) を変化させたときの計算時間を示し, (a) はトータル時間を, (b) は画像 (姿勢) 1 枚あたりの時間を, それぞれ示す. これらから, バッチ計算では 1 枚あたりの処理時間が, 枚数増に伴って大きくなるのが分かる. これは, バッチ計算の場合, 疎行列の性質を利用してもなお, 解の更新のために A を再計算する必要性からオーバーヘッドが生じていることによると思われる. 一方, 逐次計算では, 画像 1 枚あたりの処理時間が一定で, したがって枚数増に対し線形に計算時間が増加することが分かる. 必要な記憶域がより小さくて済む追加の利点も含めると, 逐次計算がバッチ計算に勝ることが分かる.

次に, 両アルゴリズムの精度を比較した. 結果を図 6 に示す. 図では, 推定した H_{sc} のうち, これをパラメータ化したときの 1 成分 (具体的には (3,1) 成分) のみについて, その相対的な平均二乗誤差を示している. なお, H_{sc} のどの成分をとっても結果の傾向は同じであった. 図 (a) は, 観測誤差の大きさ (σ) を変えたとき (画像数は 300 に固定), (b) は

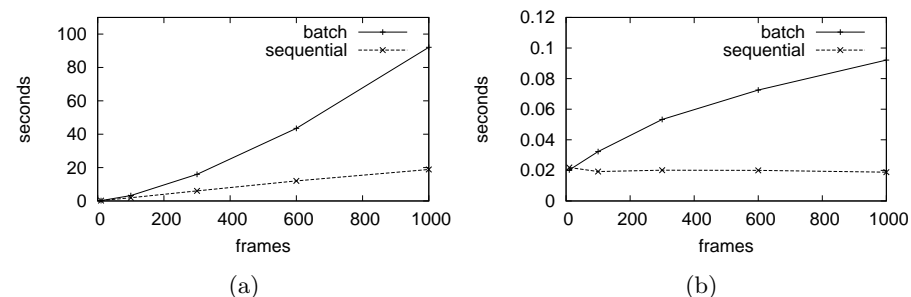


図 5 計算時間と画像の枚数 (つまりプロジェクタの総姿勢数). (a) 全計算時間. (b) 画像 1 枚あたりの計算時間.

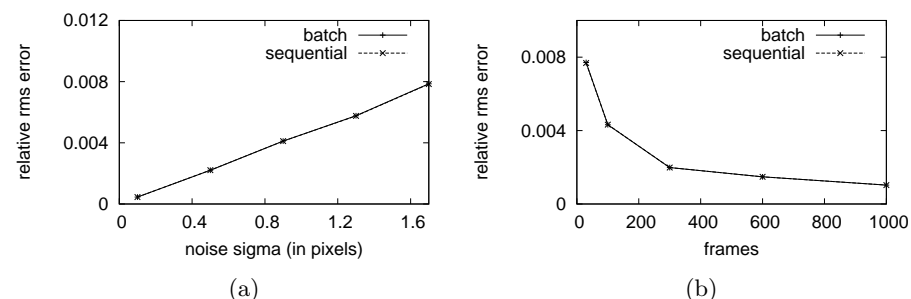


図 6 校正の RMS エラー (H_{sc} の (3,1) 要素についてのもの). (a) 観測誤差の標準偏差との関係. (b) 画像枚数との関係.

画像数を変えたとき ($\sigma = 0.5$ に固定) の, それぞれの結果である. バッチ計算と逐次計算の結果は完全に同一ではなかったが, 図ではそれらの差は目視できないくらい小さく, 両者の精度には差が見られない. 逐次計算はバッチ計算の近似に過ぎないから, 一般には, バッチ計算の方が精度が高くても不思議はない. 以上の逐次計算にとって好ましい結果は, ここでの問題の構造 (具体的には A のブロック構造) によるものと考えられる.

まとめると, バッチバンドル調整と比較して, 逐次バンドル調整の提案アルゴリズムは, 少ない記憶域の使用でずっと高速に動作する上, ほぼ同じ精度を達成すると分かった.

またわれわれは, 実システム上にアルゴリズムを実装した. 図 7 はシステムの動作状況を示す一連のスナップショットを示す. プロジェクタの動きに無関係に, 画像が決まった場所に表示され, しかもそれが正規化されていることが見て取れる.

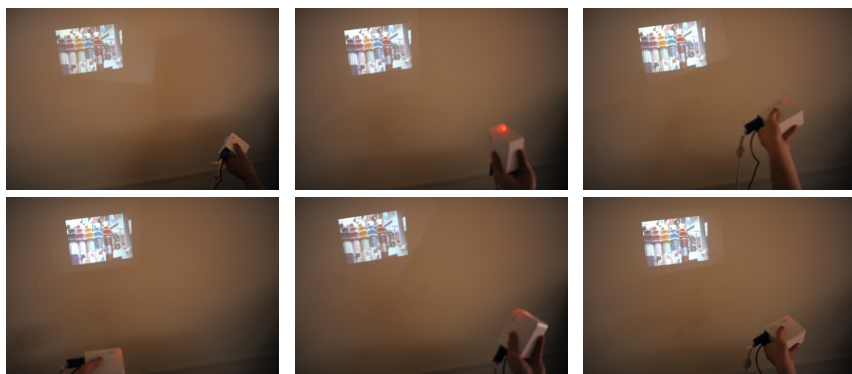


図 7 実システムの動作を撮影したスナップショット。手持ちのプロジェクタの姿勢に無関係に、正しく正規化された画像が決まった位置に表示されているのが分かる。カメラはこれらの画像の左下枠外に位置している。

5. ま と め

本稿では、モバイルプロジェクタとカメラからなるプロジェクタカメラシステムをオンラインで校正する方法を示した。応用の 1 つは、手でもったプロジェクタがある程度動いても、その動きとは無関係に、正しく正規化された（画像の 4 隅が正しく 90 度でアスペクト比も正しい）画像が、スクリーン上の決まった場所に常に表示されるようなシステムである。提案手法のオンライン性のおかげで、ユーザが自由にプロジェクタを手で動かしている間に、システムは自動的に校正される。従来のパッチバンドル調整と比較したとき、提案手法は、校正の精度を維持したまま、計算時間と記憶域の両方を最小化することに成功している。本稿では、紙面の都合で、計算原理を中心とする数学的な原理と、シミュレーション実験による検証を中心に記述を行った。提案する校正方法を真に使いやすいものとするには、高速な画像間対応付けの算出や、射影変換に基づく画像の変形、さらにはプロジェクタとカメラの時間的な同期の確立などハードウェアを含むいくつかの問題がある。これらについては今後改めて報告する。

参 考 文 献

1) Johnson, T. and Fuchs, H.: Real-Time Projector Tracking on Complex Geometry Using Ordinary Imagery, *Proceedings of IEEE Computer Society Conference on*

- Computer Vision and Pattern Recognition*, pp.1–8 (2007).
- 2) McLauchlan, P.: A Batch/Recursive Algorithm for 3D Scene Reconstruction, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.2738–2745 (2000).
- 3) Okatani, T. and Deguchi, K.: Autocalibration of a Projector-Camera System, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.27, No.12, pp.1845–1855 (2005).
- 4) Okatani, T. and Deguchi, K.: Autocalibration of an Ad Hoc Construction of Multi-Projector Displays, *Proc. Projector Camera Systems*, pp.1–4 (2006).
- 5) Okatani, T. and Deguchi, K.: Easy Calibration of a Multi-projector Display System, *International Journal of Computer Vision*, Vol.85, No.1, pp.1–18 (2009).
- 6) Pollefeys, M., Koch, R. and Gool, L.V.: Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters, *International Journal of Computer Vision*, Vol.32, No.1, pp.7–25 (1999).
- 7) Rajj, A. and Pollefeys, M.: Auto-calibration of Multi-projector Display Walls, *Proc. International Conference on Pattern Recognition* (2004).
- 8) Raskar, R. vanBaar, J., Beardsley, P., Willwacher, T., Rao, S. and Forlines, C.: iLamps: Geometrically aware and self-configuring projectors, *Proc. ACM SIGGRAPH* (2003).
- 9) Raskar, R. and Beardsley, P.: A self correcting projector, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.626–631 (2001).
- 10) Rehg, J.M., Flagg, M., Cham, T.J., Sukthankar, R. and Sukthankar, G.: Projected light displays using visual feedback, *Proc. International Conference on Control, Automation, Robotics and Vision* (2002).
- 11) Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A.: Bundle Adjustment — A Modern Synthesis, *Vision Algorithms: Theory & Practice* (Triggs, B., Zisserman, A. and Szeliski, R., eds.), Springer-Verlag LNCS 1883 (2000).
- 12) 岡谷貴之：バンドルアジャストメント，情報処理学会研究報告，Vol.2009-CVIM-167, No.37, 2009.