

大型有形文化財計測のための気球搭載型レンジセンサ

阪野 貴彦 長谷川 一英 池内 克史

† 東京大学 生産技術研究所

E-mail: †{vanno,k-hase,ki}@cvl.iis.u-tokyo.ac.jp

巨大物体の3次元形状計測では、クレーンやヘリコプタ等を利用した空中からのスキャンが効果的な方法の1つとして考えられる。ただし、対象物体が貴重な文化遺産である場合、安全性や効率性の面から、このような計測にはさまざまな問題が発生する。そのためわれわれは、レンジセンサを気球に搭載した Flying Laser Range Sensor (FLRS) を開発した。しかし、FLRS では計測中にレンジセンサが運動するため、獲得する形状データに歪みが生じてしまう。そこで、本稿では計測中に移動するレンジセンサから得られるデータを補正する手法を提案する。まず、気球に取り付けたビデオカメラによる画像列と、歪んだ形状データそのものから抽出できる情報を用いて、カメラ運動に関するパラメータの高精度な推定をおこなう。このとき、初期解として Full-Perspective Factorization を用いた。推定されたセンサの運動パラメータを用いて、歪んだ形状データを補正した。この手法をカンボジア、パイヨン寺院の計測に適用したところ、FLRS から得られた歪んだ形状データを精度良く復元することができた。

Flying laser range sensor for scanning large-scale cultural heritages

Atsuhiko BANNO, Kazuhide HASEGAWA, and Katsushi IKEUCHI

† Institute of Industrial Science, The University of Tokyo

Abstract For a large scale object, scanning from the air is one of the most efficient methods of obtaining 3D data. In the case of large cultural heritage objects, there are some difficulties in scanning them with respect to safety and efficiency. To remedy these problems, we have been developing a novel 3D measurement system, the Flying Laser Range Sensor (FLRS), in which a range sensor is suspended beneath a balloon. The obtained data, however, have some distortion due to the intra-scanning movement. In this paper, we propose a method to recover 3D range data obtained by a moving laser range sensor; this method is applicable not only to our FLRS, but also to a general moving range sensor. Using image sequences from a video camera mounted on the FLRS enables us to estimate the motion of the FLRS without any physical sensors such as gyros and GPS. At first, the initial values of camera motion parameters are estimated by perspective factorization. The next stage refines camera motion parameters using the relationships between camera images and the range data distortion. Finally, by using the refined parameter, the distorted range data are recovered. We applied this method to an actual scanning project and the results showed the effectiveness of our method.

1. はじめに

最近の3次元形状計測技術の発達により、実物体のモデリングに関する研究がさかんにおこなわれている。実物体モデリングの技術は、学術、産業、エンターテインメントなど多くの分野で必要とされ、また波及効果が期待できる。

その中でも、巨大文化遺産のモデリングはもっとも

重要で、包括的な内容を含んだアプリケーションのひとつである。これら文化遺産のモデル化は、多くの分野で重大な意義をもたらす。まず、モデル化をおこなうことで、文化遺産の形状をデジタル化されたデータとして、アーカイブ化することができる。次に、文化遺産の形状をデジタル化という劣化しないデータとして保存できるため、自然災害、火災、戦争等などによって、たとえ破壊されたとしても、修復・復元が可能となる。また、イ

インターネットやDVDなどの媒体を通して、自宅に居ながらにして、その文化遺産を訪れたような擬似体験を提供できるシステムを構築することも可能となる。

このように、実物体のモデリングの技術は多くのアプリケーションに利用することができる。われわれはこれまでに、大仏、歴史的建造物、街並などの巨大文化遺産を対象としたモデリングをおこなってきた [7]。このようなモデル化をおこなうとき、まず対象物体の3次元形状を計測することから始まる。基本的に、計測する際には、計測機器を三脚に取り付け、安定したところに置く。しかし、対象物体が巨大になると、地上においた計測機器から計測できない部分が生じることになる。そのため、その近くに一時的に櫓を組んで巨大物体を計測してきた。ところが、この方法はコストがかかるうえ、櫓を組むにも時間がかかってしまう。さらに、対象物体が巨大になれば、周りの櫓からは計測できない部分が大きくなり、この計測方法での限界が訪れる。

われわれは現在、カンボジア王国においてバイヨン寺院 [14] をデジタルコンテンツ化するプロジェクトをおこなっている。この寺院の大きさは、およそ100メートル四方もあり、これまでようにいくつか櫓を組んだだけでは、全体を計測することは不可能である。このような超巨大物体を計測する場合、これまでいくつかの方法が考えられてきた。たとえば、ヘリコプターや航空機にレンジセンサを搭載しての3次元計測である [12]。しかし、この場合、ヘリコプターなどのエンジンによって、高周波数の振動が発生してしまい得られる形状データに影響を与えてしまう、近くに接近できない、等の問題が起こる。また、対象物体が文化的に非常に価値が高いことを考慮すれば、ヘリコプターはおろかクレーンなどの重機の使用も、安全面を考慮して避けるべきであろう。

以上のことを踏まえて、われわれは巨大文化遺産を計測するための新しい方法を考案した。これを Flying Laser range Sensor(以下、FLRS, 図1 参照)[4] と名付けた。このシステムは、形状計測機器を気球に吊るして、空中から計測をおこなうものである。この方法により、高周波数の振動を発生させずに、広範囲にわたって形状計測が可能となった。ただし、問題点も発生する。計測時間中に計測機器そのものが運動するため、結果として得られた形状データが歪んでしまうことである。

そこで本稿では、このようなFLRSから得られた歪んだデータを復元する手法を提案する。レンジセンサが動くことによって歪んだレンジデータが得られるが、同時にFLRSに搭載したビデオカメラによって画像列も獲得する。ここでは、GPSやジャイロ等のセンサを用いずに、この画像列からFLRSの動きを推定する。FLRSの動きを推定するにあたっては、歪んだレンジデータそのもの



図1 FLRSとバイヨン寺院

からも条件を抽出して、いくつかの制約条件下でのコスト関数の最小化をおこなったさらに、ここで提案する手法は、われわれのFLRSに特化した手法ではなく、一般的に滑らかに運動するレンジセンサに対して適用可能である。

このような非線型関数の最適化問題を解く際、ローカルミニマムに陥るのを避けるために、われわれは初期解としてFactorization[13][2][8][9]による解をもちいた。特に今回、実際のカメラモデルである透視投影下でのPerspective-Factorization[3]の解を採用した。このようにして推定されたFLRSの運動をもちいて、最終的に歪んだレンジデータの修正をおこなう。

本稿の構成は以下の通りである。2.では、われわれが開発したFLRSシステムについて簡単に紹介する。3.では、われわれが提案するセンサ運動推定のための手法について述べる。ここでは、3つの制約条件とコスト関数の設定について説明する。4.では、本手法をバイヨン寺院に適用した実験とその結果について述べる。また、本手法を評価するため、修正したレンジデータと、地上に固定したレンジセンサから得られたレンジデータとの比較をおこなった。そして最後にまとめとする。

2. FLRSシステム

われわれのFLRSシステムは、スキャナ、コントローラ、PCの3つによって構成され、この3つを気球によって吊るす。また、気球そのものは、取り付けられた4本のロープを地上から人手によって支える。

スキャナ部分は、レーザーレンジファインダを含み、特に気球に吊るすことを考えて製作した。図2にその内部の様子を示す。この部分は、スポット照射型のレーザー測距機と2つのミラーとからなり、計測原理はtime-of-flightである。レーザー光源には、Z+F社のLARA25200を使用した。2つのミラーは、レーザーが向かう方向を制御しており、高速に回転するポリゴンミラーが距離画像における横方向のスキャンをおこない、平板のスイングミ

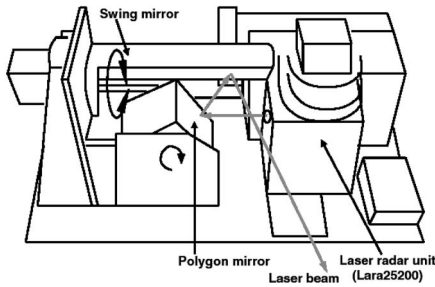
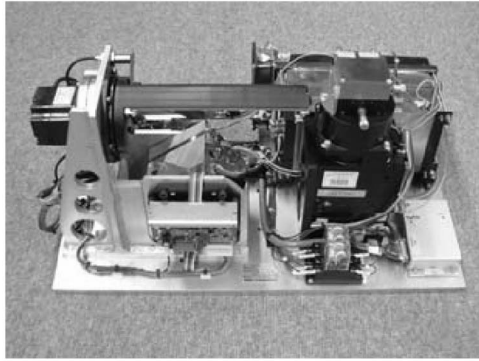


図 2 FLRS スキャナ部分

ラーが縦方向のスキャンをおこなう。詳細を表 1 に示す。

表 1 FLRS のスペック

| | |
|-----------------------------------|-------------------|
| Light source | 23mW, 780nm |
| Scanning length | 2 - 25.2 m |
| Range resolution | 0.38mm |
| Vertical resolution | 160 pixel |
| Horizontal resolution | 1800 pixel |
| Rotational speed (polygon mirror) | 2400 rpm |
| Swing range (swing mirror) | -15 - +15 deg |
| Scanning time | 1 sec/frame |
| Weight | 21.2kg |
| Size (depth × width × height) | 27 × 53 × 30 (cm) |

コントローラ部分は、信号処理装置とインターフェースとからなる。信号処理装置では PC からの信号を受け取り、2 つのミラーの制御をおこなう。レーザー測距機からのレンジデータと、エンコーダからのミラー角度のデータとを用いて、1 枚のレンジデータに変換し、PC に蓄積される。

また、スキャナ部分を覆うボックス外側に、ビデオカメラを固定して搭載している。このカメラは、レンジセンサと同期が取れており、スキャン開始時での撮影フレームが分かるようになっている。

3. 運動パラメータ推定手法

本研究では、FLRS から得られる画像と歪んだ形状データを用いて、センサの動きを推定することによって、

歪んだ形状データを修復することである。そこで、われわれは 3 つの制約条件を課し、最適化問題のフレームワークによって、カメラ位置・姿勢の推定をおこなう。この条件のもと、コスト関数の最小化の枠組みで最適化問題を解くことで、カメラパラメータの高精度な推定をおこなった。われわれの用いたコスト関数は非線型であり、ローカルミニマムに陥るのを避けるために、あらかじめ精度の良い初期解を必要とする。

そこで、ピンホールカメラモデルを忠実に再現した Full-Perspective Factorization による解を初期解とすることで、最適化計算の安定化をはかった。

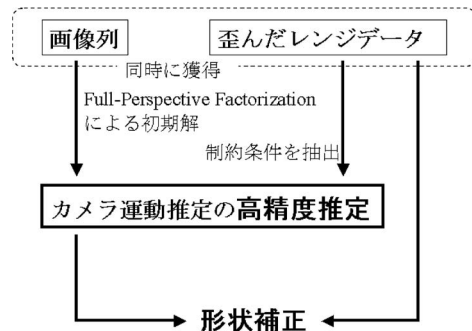


図 3 処理の流れ

3.1 推定するパラメータ

ビデオカメラでは、 F 枚のフレームにわたって、 P 個の特徴点をトラッキングする。フレーム番号 f において、トラックされた特徴点 p は画像座標系で、 $\{(u_{fp}, v_{fp}) | f = 1, \dots, F, p = 1, \dots, P, 2F \geq P\}$ に表示されているとする。

画像フレーム f を取得するときのカメラ位置を、世界座標系で \vec{t}_f とする。また、そのときのカメラの姿勢を 3 つの単位ベクトル $\vec{i}_f, \vec{j}_f, \vec{k}_f$ で表す。この 3 つのベクトルは、カメラ座標系での座標軸を表していて、 \vec{i}_f, \vec{j}_f は画像面に平行な 2 つの軸であり、 \vec{k}_f は光軸に平行な軸とする。

したがって、推定すべきパラメータは、 P 個の 3 次元座標と F 個のカメラパラメータ (位置, 回転) であり、すべてで $(3 \times P + 6 \times F)$ 個ある。

ピンホールカメラモデルでは、3 次元世界座標系において \vec{s}_p にある点は、次の式によってフレーム f の画像面で (u_{fp}, v_{fp}) に投影される。

$$u_{fp} = f \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{\vec{k}_f(\vec{s}_p - \vec{t}_f)} \quad (1)$$

$$v_{fp} = f \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{\vec{k}_f(\vec{s}_p - \vec{t}_f)} \quad (2)$$

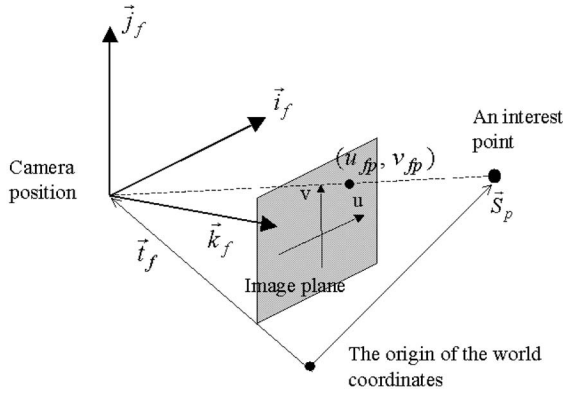


図4 カメラ運動のパラメータ

3.2 トラッキングにおける制約

最も基本的な制約条件として、3次元空間中の特徴点 \vec{s}_p は、各フレームで画像中の (u_{fp}, v_{fp}) に投影されていなければならない。つまり、実際にえられた画像中の特徴点と、再構成された画像面に投影された特徴点との誤差が最小となるように制約を課すこの制約条件によって、次のコスト関数を最小化しなければならないことがわかる。

$$F_A = \sum_{f=1}^F \sum_{p=1}^P \left(\left(u_{fp} - f \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{k_f(\vec{s}_p - \vec{t}_f)} \right)^2 + \left(v_{fp} - f \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{k_f(\vec{s}_p - \vec{t}_f)} \right)^2 \right) \quad (3)$$

F_A を最小化することによって、3次元空間で固定している特徴点を、移動するカメラによって正しくトラックする様子が再現できる。ただし、実際には微分計算を簡単にするため、次のように変形したものを実装する。

$$F_A = \sum_{f=1}^F \sum_{p=1}^P \left(\left(\vec{k}_f(\vec{s}_p - \vec{t}_f) u_{fp} - f \vec{i}_f(\vec{s}_p - \vec{t}_f) \right)^2 + \left(\vec{k}_f(\vec{s}_p - \vec{t}_f) v_{fp} - f \vec{j}_f(\vec{s}_p - \vec{t}_f) \right)^2 \right) \quad (4)$$

3.3 カメラの運動における制約

われわれが気球を選択した理由のひとつが、気球によって高周波数の振動の影響を受けない、というものであった。裏を返せば、気球は低周波数のみの滑らかな振動の影響を受ける、といえる。われわれが気球を用いて計測をおこなう場合、地上から人手でロープによって支えておこなっているため、気球の運動に影響を与えるものは、風や、ロープを伝う人の力のみである。したがって、気球の運動は非常に滑らかであり、急加速、急減速、折れ曲がった軌道を描くような運動は実際に発生することがない。このことから、つぎのコスト関数を設定する。

$$F_B = \int \left(w_1 \left(\frac{\partial^2 \vec{t}_f}{\partial t^2} \right)^2 + w_2 \left(\frac{\partial^2 \mathbf{q}_f}{\partial t^2} \right)^2 \right) dt \quad (5)$$

ここで、 \vec{t}_f はカメラの位置を表し、 t は時間、 w_1, w_2 は重み係数を表す。カメラの回転については、単位クォータニオン \mathbf{q}_f で評価している。このクォータニオンは $\vec{i}_f, \vec{j}_f, \vec{k}_f$ から直ちに求めることができる。被積分関数の第1項は、カメラの位置に関する移動が滑らかであることの制約となっており、第2項はカメラの回転に関して滑らかであることの制約になっている。したがって、カメラの運動が滑らかであるとき、関数 F_B は小さな値をとる。ただし、実装において積分は離散的におこなわれる。

$$F_B = \sum_{f=1}^F \left(w_1 \left(\frac{\partial^2 \vec{t}_f}{\partial t^2} \right)^2 + w_2 \left(\frac{\partial^2 \mathbf{q}_f}{\partial t^2} \right)^2 \right) \quad (6)$$

3.4 歪んだレンジデータに関する制約

FLRS から得られる距離画像は、センサそのものが運動しているため、結果として正しくない形状が得られる。ただし、“正しくない” というのは距離画像全体を見た場合であって、局所的には正確な情報を含んでいる。以下の制約では、この局所的に正確な距離情報を用いた制約条件を示す。

われわれの FLRS では、計測原理に time-of-flight を用いており、距離画像面でみて、ラスタスキャン順にレーザーを発している。1回の計測に要する時間は既知であるので、距離画像の各ピクセルでの距離値を計測したときの、計測開始時からの時刻が求められる。つぎに、気球に搭載したレンジセンサとビデオカメラの同期がとれていれば、その距離画像の各ピクセルを計測したときのビデオ画像のフレーム番号 f を見つけることができる。さらに、レンジセンサとビデオカメラとのキャリブレーションができていれば、その計測された距離画像中のピクセルが、ビデオ画像 f のどの部分を計測しているのかがわかる。このとき、ビデオ画像に対応する部分が、トラックしている特徴点に一致していると、ビデオ画像 f を撮影している時刻でのセンサと3次元空間中の特徴点との相対位置関係が特定できる。したがって、第3の制約条件として、以下の関数を最小化をおこなう。

$$F_C = \sum_{p=1}^P \left\| \mathbf{x}_{fp} - (\vec{s}_p - \vec{t}_{fp}) \right\|^2 \quad (7)$$

ここで、添え字 fp は、特徴点 p がレンジセンサによってスキャンされたときのフレーム番号を示す。また、 \mathbf{x}_{fp} はこのときのレンジセンサの出力である。

この制約は非常に重要である。画像のみからカメラの運動を推定する“Structure from Motion” や “Shape from Motion” では、スケールに関する曖昧性が残されてしまう。一方、この制約では絶対的な距離データを用い

ているため、カメラパラメータ推定の精度向上のみならず、スケールの曖昧性の除去にも有用である。また、 $\mathbf{x}_{fp} = (x_{fp}, y_{fp}, z_{fp})$ とすれば、上の制約をさらに厳しくしたものとて、

$$F_C = \sum_{p=1}^P \left((x_{fp} - \vec{i}_{fp}(\vec{s}_p - \vec{t}_{fp}))^2 + (y_{fp} - \vec{j}_{fp}(\vec{s}_p - \vec{t}_{fp}))^2 + (z_{fp} - \vec{k}_{fp}(\vec{s}_p - \vec{t}_{fp}))^2 \right) \quad (8)$$

が得られる。

3.5 全コスト関数

以上の3つの関数の重みつき加算をおこなった全コスト関数

$$F = w_A F_A + w_B F_B + w_C F_C \quad (9)$$

を最小化することによって、特徴点位置とカメラの運動パラメータに関する推定をおこなう。

この全コスト関数の最小化には、共役勾配法 [10][11] を用いる。各共役方向に関する1次元の最小化探索には、黄金分割法を採用することで、計算の効率化を図った [5]。

また、前述したように最小化計算の初期解として、Full-Perspective Factorization による解をもちいた (Appendix 参照)。これは、ローカルミニマムに陥るを防ぐのに、真の解に近い初期解が必要であるためである。実際、Weak-Perspective Factorization による解を初期解として採用した場合、最小化計算が収束しないケースが多く発生し、Perspective Factorization による非常に近い近似解を採用することができたため、最小化問題によるカメラパラメータ推定が成功したと考えられる。

4. 実験結果

4.1 特徴点のトラッキング

われわれの FLRS システムでは、1回の計測で72フレーム撮影している (30fps)。各画像において特徴点を抽出するのに、スケール変化に対してロバストな SIFT key [6] を使用した。これは、FLRS が光軸方向に大きく運動する可能性もあり、画像中に撮影される物体のスケールが変化することを考慮したためである。画像間で特徴点を追跡するため、隣り合うフレーム間でマッチングをおこなっていき、シーケンシャルに全フレームわたって対応点をつなげた。以上の手続きを経て、72フレームの画像列から100点程度の特徴点を追跡することができた。

4.2 復元形状の評価

本論文で提案している形状復元手法の精度について評価するため、FLRS から得られたデータを復元した形状と、他のレンジセンサから得られた形状とを比較した。この”他のレンジセンサ”とは、地上に固定して置かれた

Cyrax2500 であり、このセンサから得られたデータを正しい解とする。

ICP アルゴリズム [1][15] によって両者のデータを位置合わせし、重なり合った部分について考察をおこなう。アラインメントの結果を図5に示す。

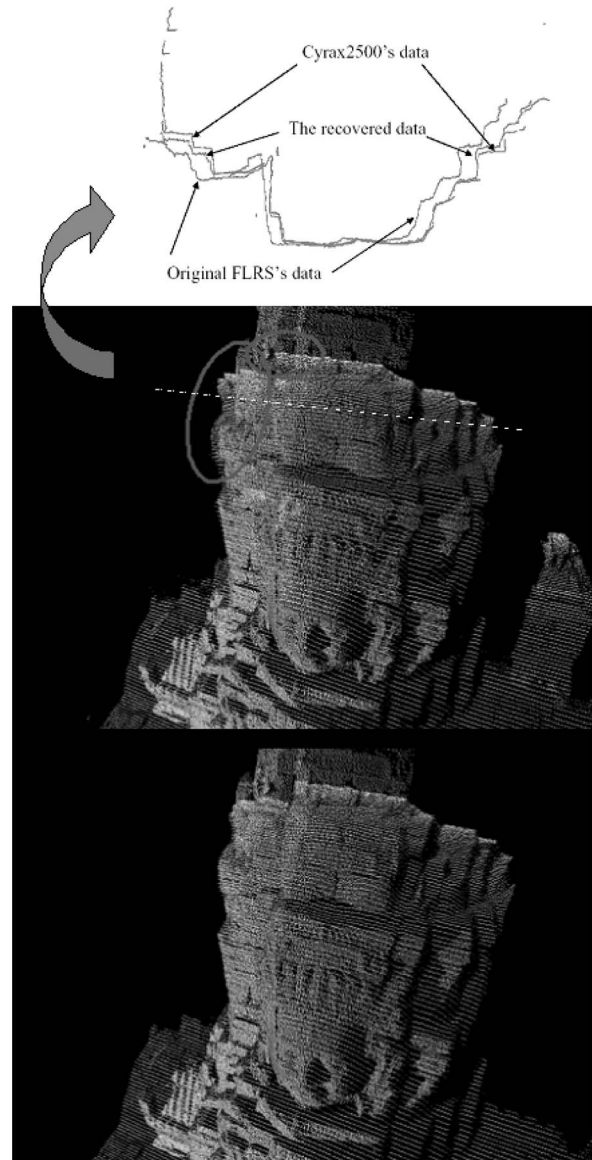


図5 地上固定レンジセンサとの比較:中央 FLRS から得られたオリジナルデータ,下 提案手法によって修復したデータ,上-中央の図で点線で示した部分をカットした断面図

中央の図は、Cyrax2500 によるデータと、FLRS から得られたそのまま (オリジナル) のデータとを合わせた結果である。尊顔およびそれより下での形状は合っているが、それより上の部分、特に楕円で示した部分で、正しい形状モデルとのずれが顕著である。一方、下の図では、Cyrax2500 によるデータと、本手法によって修正したデータとの合わせである。データ全体に渡って両者が

正しく重なっており、上の図において楕円で示しておいた部分もきちんと合っていることがわかる。また、図5上は、中央の図で点線で示した部分(尊顔の上)をカットしたときの断面である。

図6は、本手法による効果を視覚化したものである。ここでは、FLRSから得られた形状がどれだけCyrax2500による形状に近づいたかを示してある。ICPアルゴリズムによって両データを合わせたときの対応点に関して、点間距離が閾値(6.0cm)以下だったときには明るい色で表示し、それ以上であったときには暗い色で表示した。つまり、ICPによって位置合わせした2つのデータで、形状がマッチしている部分が明るい色で示されている。左側の図が、Cyrax2500によるデータと、FLRSから得られたオリジナルのデータとを合わせたものであり、右側の図が、Cyrax2500によるデータと本手法によって修正したデータとを合わせたものである。一瞥しただけでも、右の図で、明るい領域が増えているのがわかり、正しい形状モデルと、修復した形状データがマッチしている様子がわかる。このように、本手法によってFLRSから得られたデータが良好に修復されていることがわかる。

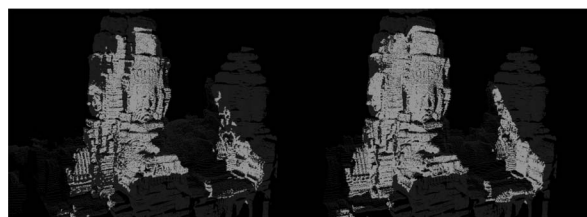


図6 地上に固定したセンサから得られた形状とのマッチング(明るい部分でマッチングがとれている):右 FLRSから得られたオリジナルデータ,左 提案手法によって修復したデータ

ここまでを示したものは、比較的穏やかな気球運動でのデータセットであった。図7は気球運動が激しいときに得られたデータセットである。左上の図は、FLRSから得られたオリジナルの形状であり、右上の図が提案手法で修復したものである。また、地上に固定したCyrax2500によって得られた形状と、修復したデータとを位置合わせしたものが下の図である。気球運動が激しい場合、オリジナルのデータが大きく歪んでおり、提案手法によって、正しい形状に修復されている様子がわかる。このように、気球運動の様子に関係なく、本手法によって歪んだデータを正しく修復することができた。

また、最後にパイヨン寺院全体を地上固定のレンジセンサとFLRSを使ってモデル化した様子を図8に示す。

5. まとめと今後の課題

本稿では、FLRSによる計測で問題となる計測形状



図7 気球の運動が激しい場合のデータ:左上 FLRSから得られたオリジナルデータ,右上 提案手法によって修復したデータ,下 Cyrax2500によって得られたデータと修復したデータとを位置合わせしたしたもの

データの修復手法について述べた。気球に同時に搭載したビデオカメラと歪んだ形状データから、センサの動きを精度良く推定することで、3次元形状の修復をおこなった。処理の流れとしてまず、Perspective-Factorizationをおこなうことで、カメラ運動の初期推定をおこなった。つぎに、歪んだ形状データそのものからも運動推定に有効な情報を抽出し、3つの条件を課した非線型最適化問題を解くことで、カメラ運動推定の精度を向上させた。その結果、歪んだデータを精度良く修復できることが示された。ここで設定したセンサは、同期のとれたレンジセンサとビデオカメラのみから構成されており、ジャイロやGPS等のセンサを用いずに運動パラメータの推定をおこなった。したがって、本手法は気球に特化したものではなく、滑らかに運動するレンジセンサに対して、一般的に適用することができる。

また、技術的な課題もまだ多く残されている。例えば、現システムでは、特徴点は全フレームに渡ってトラックされていなければならない、オクルージョンに対応できていない。したがって、気球の運動が激しい場合にはカメラの運動を推定することが難しい。今後は、このようなオクルージョンに対応できるよう改良する予定である。

謝 辞

本研究は科学技術振興機構戦略的創造研究推進事業(CREST)の支援を受けておこなわれた。パイヨン計測にあたっては、日本国政府アンコール遺跡救済チーム(JSA)の方々に感謝いたします。

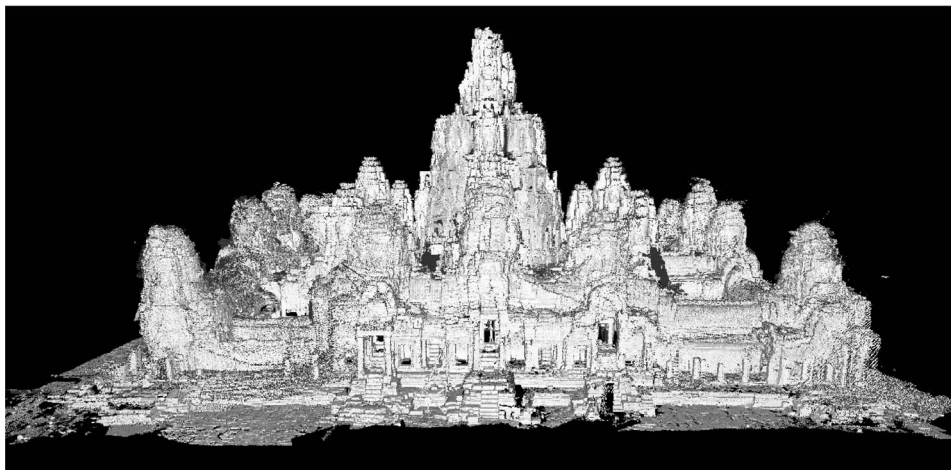


図 8 バイヨン寺院の全体モデル

参考文献

- [1] P.J.Besl and N.D.McKay, "A method for registration of 3-D shapes," IEEE Trans. on PAMI, vol.14, pp.239-256, 1992.
- [2] J.Costeira and T.Kanade, "A multi-body factorization method for motion analysis," Proc. of ICCV, pp.1071-1076, 1995.
- [3] M.Han and T.Kanade, "Perspective factorization methods for Euclidean reconstruction," CMU-RI-TR-99-22, 1999.
- [4] Y.Hirota, T.Masuda, R.Kurazume, K.Ogawara, K.Hasegawa and K.Ikeuchi, "Flying Laser Range Finder and its data registration algorithm," Proc. of ICRA, pp.3155-3160, 2004.
- [5] D. A. Jacobs, The State of the Art in Numerical Analysis, London: Academic Press, 1977.
- [6] D.G.Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, Vol.60, No.2, pp.91-110, 2004.
- [7] D.Miyazaki, T.Oishi, T.Nishikawa, R.Sagawa, K.Nishino, T.Tomomatsu, Y.Yakase and K.Ikeuchi "The great buddha project: Modelling cultural heritage through observation" Proc. of VSMM, pp.138-145, 2000.
- [8] T.Morita and T.Kanade, "A sequential factorization method for recovering shape and motion from image streams," IEEE Trans. on PAMI, vol.19, No.8, pp.858-867, 1997.
- [9] C.Poelmann and T.Kanade "A paraperspective factorization method for shape and motion recovery," IEEE Trans. on PAMI, vol.19, No.3, pp.206-218, 1997.
- [10] E. Polak, Computational Methods in Optimization, New York: Academic Press, 1971.
- [11] J. Stoer and R.Bulirsch, Introduction to Numerical Analysis, New York: Springer-Verlag, 1980.
- [12] S.Thrun, M.Diel and D.Haehnel, "Scan alignment and 3-D surface modeling with a helicopter platform," The 4th International Conference on Field and Service Robotics, 2003.
- [13] C.Tomasi and T.Kanade, "Shape and motion from image streams under orthography: a factorization method," International Journal of Computer Vision, Vol. 9, No.2, pp. 137-154, 1992.
- [14] J.Visnvcova, L.Zhang and A.Gruen, "Generating a 3D model of a bayon tower using non-metric imagery," Proc. of the International Workshop Recreating the Past -Visualization and Animation of Cultural Heritage, 2001.
- [15] Z.Zhang, "Iterative point matching for registration of free-form curves and surfaces," International Journal of Computer Vision, Vol.13, pp.119-152, 1994.

Appendix

A. Weak-Perspective Factorization

得られた画像列において、全フレーム (F 枚) にわたって、 P 個の特徴点がトラッキングされているとする。各特徴点 p は、3次元物体上の点 s_p に対応している。画像上での座標系では、フレーム番号 f において、特徴点 p は $\{(u_{fp}, v_{fp}) | f = 1, \dots, F, p = 1, \dots, P, 2F \geq P\}$ に表示されているとする。

画像特徴点の水平方向成分 u_{fp} をもちいて、 $F \times P$ の行列 U を定義する。同様に、画像特徴点の垂直方向成分 v_{fp} をもちいて、 $F \times P$ の行列 V を定義する。両者を合わせて、 $2F \times P$ の観測行列 W を次のように定義する。

$$W = \begin{pmatrix} U \\ V \end{pmatrix} \quad (\text{A.1})$$

弱透視投影カメラモデルでは、3次元世界座標系において全ての点がカメラから同じ奥行きにある、とされるので、 s_p と示される点はフレーム f の画像面で (u_{fp}, v_{fp}) に投影される。

$$u_{fp} = \frac{f}{z_f} i_f(\vec{s}_p - \vec{t}_f) \quad (\text{A.2})$$

$$v_{fp} = \frac{f}{z_f} j_f(\vec{s}_p - \vec{t}_f) \quad (\text{A.3})$$

$$z_f = k_f(\vec{c} - \vec{t}_f) \quad (\text{A.4})$$

ここで、 \vec{c} はトラックした特徴点の重心とする．世界座標系の原点はどこに設定しても一般性を失わないため、この重心を原点、つまり $\vec{c} = \sum \vec{s}_p = 0$ とすると、 $z_f = -\vec{k}_f \vec{t}_f$ となるため、上式は次のようになる．

$$u_{fp} = \vec{m}_f \vec{s}_p + \mathbf{x}_f \quad (\text{A.5})$$

$$v_{fp} = \vec{n}_f \vec{s}_p + \mathbf{y}_f \quad (\text{A.6})$$

$$\vec{m}_f = \frac{f}{z_f} \vec{i}_f \quad (\text{A.7})$$

$$\vec{n}_f = \frac{f}{z_f} \vec{j}_f \quad (\text{A.8})$$

$$\mathbf{x}_f = -\frac{f}{z_f} \vec{i}_f \vec{t}_f \quad (\text{A.9})$$

$$\mathbf{y}_f = -\frac{f}{z_f} \vec{j}_f \vec{t}_f \quad (\text{A.10})$$

これを行列形式で表現すれば、

$$\begin{pmatrix} u_{11} & \dots & u_{1P} \\ u_{21} & \dots & u_{2P} \\ \vdots & \vdots & \vdots \\ u_{F1} & \dots & u_{FP} \\ v_{11} & \dots & v_{1P} \\ \vdots & \vdots & \vdots \\ v_{F1} & \dots & v_{FP} \end{pmatrix} = \begin{pmatrix} \vec{m}_1^t \\ \vec{m}_2^t \\ \vdots \\ \vec{m}_F^t \\ \vec{n}_1^t \\ \vdots \\ \vec{n}_F^t \end{pmatrix} (\vec{s}_1 \dots \vec{s}_P) + \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_F \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_F \end{pmatrix} (1 \dots 1) \quad (\text{A.11})$$

また、原点と重心とを一致させたことにより、観測行列の要素を各行で加算すると、

$$\sum_{p=1}^P u_{fp} = \sum_{p=1}^P \vec{m}_f \vec{s}_p + \sum_{p=1}^P \mathbf{x}_f = P \mathbf{x}_f \quad (\text{A.12})$$

同様に、

$$\sum_{p=1}^P v_{fp} = P \mathbf{y}_f \quad (\text{A.13})$$

となる．

そこで、 $\tilde{W} = W - (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_F \ \mathbf{y}_1 \ \dots \ \mathbf{y}_F)^t (1 \ 1 \ \dots \ 1)$ と変換することで、Rank Theorem より行列 \tilde{W} は2つの行列 M と S の積となる．

$$\tilde{W} = M \cdot S \quad (\text{A.14})$$

ここで、 M は $2F \times 3$ 、 S は $3 \times P$ の行列である．また、この分解では次の拘束条件を課す．

$$|\vec{m}_f| = |\vec{n}_f| \quad (\text{A.15})$$

$$\vec{m}_f \cdot \vec{n}_f = 0 \quad (\text{A.16})$$

このようにして、動画像からトラッキングされる特徴点から、特徴点の3次元位置と、カメラの位置・姿勢が計算のみによって求められる．

B. Full-Perspective Factorization

透視投影において、3次元中の点 $s_p = (X, Y, Z)$ は次の変換によって、画像面上 (u_{fp}, v_{fp}) に射影される．

$$u_{fp} = f \frac{\vec{i}_f (\vec{s}_p - \vec{t}_f)}{k_f (\vec{s}_p - \vec{t}_f)} \quad (\text{B.1})$$

$$v_{fp} = f \frac{\vec{j}_f (\vec{s}_p - \vec{t}_f)}{k_f (\vec{s}_p - \vec{t}_f)} \quad (\text{B.2})$$

簡単な式変形により、右边を弱透視投影での式と同じ形で、次のようになる．

$$\frac{\vec{k}_f (\vec{s}_p - \vec{t}_f)}{z_f} u_{fp} = \frac{f}{z_f} \vec{i}_f (\vec{s}_p - \vec{t}_f) \quad (\text{B.3})$$

$$\frac{\vec{k}_f (\vec{s}_p - \vec{t}_f)}{z_f} v_{fp} = \frac{f}{z_f} \vec{j}_f (\vec{s}_p - \vec{t}_f) \quad (\text{B.4})$$

$z_f = -\vec{k}_f \vec{t}_f$ と置き換えることにより、

$$(\lambda_{fp} + 1) u_{fp} = \frac{f}{z_f} \vec{i}_f (\vec{s}_p - \vec{t}_f) \quad (\text{B.5})$$

$$(\lambda_{fp} + 1) v_{fp} = \frac{f}{z_f} \vec{j}_f (\vec{s}_p - \vec{t}_f) \quad (\text{B.6})$$

$$\lambda_{fp} = \frac{\vec{k}_f \cdot \vec{s}_p}{z_f} \quad (\text{B.7})$$

となる．つまり、2次元画像上における各特徴点の座標値 (u_{fp}, v_{fp}) にある数 λ_{fp} を掛けることにより、透視投影空間からアフィン空間である弱透視投影空間に変換することができる．したがって、この変換に必要な λ_{fp} を繰り返し計算によって求めることで、弱透視投影下でのFactorizationを用いて、透視投影空間での物体形状、カメラの運動パラメータを求めることができる．以下にそのアルゴリズムを示す．

Input 各フレームにおいて P 個の特徴点をトラッキングした F 枚の画像

Output P 個の特徴点の3次元座標 \vec{s}_p およびフレーム番号 f を撮影した時のカメラ位置 \vec{t}_f と姿勢ベクトル $\vec{i}_f, \vec{j}_f, \vec{k}_f$

(1) $\lambda_{fp} = 0$ と仮定する．

(2) $(\lambda_{fp} + 1) u_{fp} = f \vec{i}_f (\vec{s}_p - \vec{t}_f) / z_f, (\lambda_{fp} + 1) v_{fp} = f \vec{j}_f (\vec{s}_p - \vec{t}_f) / z_f$ と置き、弱透視投影でのFactorizationをおこない、 $\vec{s}_p, \vec{t}_f, \vec{i}_f, \vec{j}_f, \vec{k}_f, z_f$ を求める．

(3) $\lambda_{fp} = \vec{k}_f \cdot \vec{s}_p / z_f$ を計算する．

(4) step3で求めた λ_{fp} をstep2に代入し、再びFactorizationを解いて新たに λ_{fp} を求める．これを繰り返す．

Until λ_{fp} の更新する値が充分小さくなるまで．