

Bundler

Structure from Motion for Unordered Image Collections



my recommendations on research & development tools.

満上育久†

キーワード：ストラクチャフロムモーション，因子分解，特徴点，バンドル調整，マルチビューステレオ

1. ま え が き

Structure from Motion (以下SFM)とは、あるシーンをカメラの視点を変えながら撮影した複数枚の画像から、そのシーンの3次元形状とカメラの位置を同時に復元する手法である(図1)。この技術は、得られるシーンの3次元形状に着目すれば、コンピュータビジョンにおける形状復元問題の1解法であり、一方、カメラの位置推定に着目すれば、ロボットビジョンにおける自己位置推定手法と捉えることもできる。このように、SFMは、応用範囲の広い基本的かつ重要な技術である。

2. SFMの要素技術と経緯

SFMの実現方法として、Tomasi-Kanadeの因子分解法¹⁾が知られている。この手法では、まずシーンを撮影したF枚の時系列画像に対してP組の対応点を取得して、その対応点群から $2F \times P$ の行列を作成する。そして、この行列の階数が3以下であることを利用して、特徴点の3次元位置を表す行列とカメラ位置を表す行列に分解する。この因子

分解法は、コンピュータビジョンにおける基礎的なテクニックとして大変意義深く、現在のSFMの解法のベースとなっているが、この手法が提案された当時は、前段の処理である対応点の取得方法がまだ非力であったために、その適用範囲は非常に限定的であった。具体的には、異なる視点の画像間での点の対応関係は、連続的に視点移動した際の画像列中の特徴点追跡によって得なければならなかった。そのため、画像の順序の情報が必要となり、また、シーンの遮蔽を扱うのが困難であった。

その後、SIFT²⁾やSURF³⁾など、撮影する向きや大きさが変化しても、不変な特徴ベクトルが得られるオペレータが提案され、この問題が克服される。これらのオペレータを用いると、特徴点の追跡処理をせず対応点が得られるため、順序情報のない、とびとびの画像群からでも、SFMが可能となってきた。本稿で紹介するBundlerは、このような手法の一つである。

3. Bundlerの概要

Bundler⁴⁾は、図2のような複数の画像からSFMを行うオープンソースプログラムである。開発者はCornell UniversityのN. Snavely氏で、本稿執筆時の最新バージョンは0.4である。前述のとおり、用意すべき画像は撮影の順序関係を問わないだけでなく、異なる焦点距離あるいは異なるカメラで撮影された画像も統一的に取り扱うことができる。そのため、例えば、ある建築物についてさまざまな人が撮影した画像をWebから収集し、それらからSFMを行うことも可能である。

なお、Bundlerが担当する処理は、特徴点群の最適な対応付けを求めるためのバンドル調整と、前述の因子分解である。その前処理となる特徴点抽出については、適当なオペレータを別途用意することになっている。READMEでは、D.G. Lowe氏のSIFT実装²⁾の利用方法を紹介しているが、スクリプトの軽微な修正によってR. Hess氏の実装⁵⁾など他のオペレータも使用可能である。

4. Bundlerの実行方法

Bundlerの実行は至って簡単である。JPEG形式で保存された画像列のあるディレクトリ(image_dirとする)に格

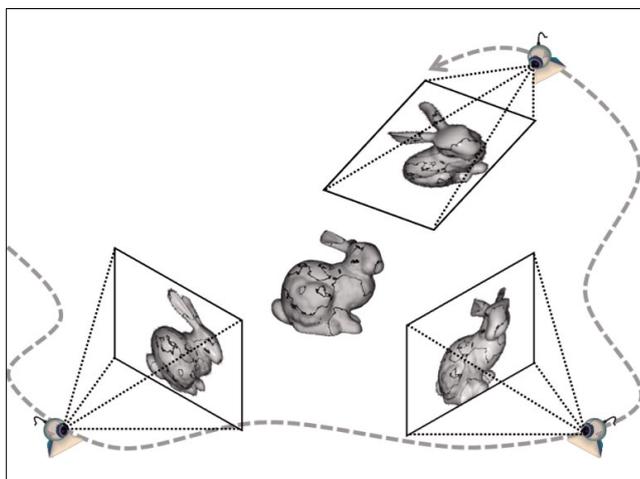


図1 Structure from Motion

† 大阪大学産業科学研究所

"Bundler: Structure from motion for unordered image collections" by Ikuhisa Mitsugami (The Institute of Scientific and Industrial Research, Osaka University, Osaka)



図2 Bundlerに入力される画像群の例

納し、そのディレクトリ上でRunBundler.shスクリプトを実行すればよい(BUNDLER_PATHはBundlerのディレクトリパス)。

```
% cd image_dir/
% BUNDLER_PATH/bin/RunBundler.sh
```

このスクリプトは、image_dir内の画像リストの作成、SIFTオペレータの実行、バンドル調整など、必要となる一連の処理を順に呼び出している。他のオペレータを使いたい場合など、一連処理の一部を他の手法に置き換えたい場合にはこのスクリプトを編集する。しかし、通常の使用であれば、このスクリプトを書換えることなく実行するだけで充分である。

実行後、image_dir内にlist.txtを含むいくつかのテキストファイルとbundleというディレクトリが生成される。list.txtは読み込まれた画像のリストである。その他のテキストファイルは一連の処理の中間データであり、ここでは説明を省略する。SFMの実行結果は、bundleディレクトリ内に格納されている。このディレクトリ内にもいくつかファイルが存在するが、ここでは最も重要な二つのファイルはbundle.outとpoints***.plyを取り上げる。以下に、これらのファイルについて説明する。

(1) bundle.out

Bundlerによって得られるSFM処理結果というべきファ

イルであり、各カメラの内部パラメータ(焦点距離・歪みパラメータ等)、外部パラメータ(3次元位置・姿勢)、各特徴点の3次元位置・色、その点の復元に使用された各画像の特徴点の情報が網羅されている。このファイルの構造を以下に示す。

冒頭の2行がヘッダであり、まず1行目にバージョン情報、2行目にカメラ数と3次元座標が得られた特徴点の数が記されている。

3行目以降は、記されたカメラ数分のカメラパラメータが列挙される。各カメラのパラメータは5行からなり、1行目は焦点距離 f と歪みパラメータ k_1, k_2 、2~4行目はカメラの姿勢を表す回転行列 R 、5行目はカメラの位置を表す並進ベクトル t を示している。

すべてのカメラのパラメータが列挙されると、次に特徴点の情報が列挙される。各特徴点の情報は3行で表現され、1行目はその3次元位置、2行目は色情報である。3行目は少し複雑で、冒頭の実数 n はその特徴点に対応する点が映っている画像の枚数(視点の個数)である。この各画像に対して、(カメラ番号)・(そのカメラ画像内での2次元特徴点番号)・(X座標)・(Y座標)が列挙される。すなわち、この行は $1+4n$ 個という可変長の要素を持つ。

```
# Bundle file v0.3
16 1500 } カメラ数, 特徴点数
1.4647020813e+03 5.7837047967e-02 5.0398823818e-03 } f, k1, k2
9.7921951660e-01 1.6739662044e-01 -1.1448803333e-01 } R
-1.8898781645e-01 9.5799602965e-01 -2.1570167457e-01 }
-7.3571350029e-02 -2.3285613293e-01 -9.6972433083e-01 }
2.7498985595e-01 3.5379968267e-01 1.9223362389e+00 } t
1.4836367620e+03 5.5990662876e-02 6.2676944262e-03 }
9.7708578715e-01 1.7602754097e-01 -1.1965646397e-01 }
-2.1101380065e-01 8.7471457767e-01 -4.3628841785e-01 }
-2.7866475998e-02 -4.5154037742e-01 -8.9181542209e-01 }
2.7698372342e-01 1.0085493795e+00 1.7818091558e+00 }
.
.
.
-8.6698319909e-02 3.2256824222e-01 2.9020394139e+00 } 位置
84 85 87 } 特徴点の色 (R, G, B)
2 0 31 -134.6200 81.2100 1 32 -153.4400 67.0400 } 対応する2次元座標
1.9650482229e-01 2.1354755333e-01 2.8620601094e+00 }
36 30 30 }
3 0 74 281.1300 -154.0400 1 77 279.7200 -161.6300 11 95 265.1100 75.7400 }
.
.
.
```

(2) points***.ply

このファイルにはBundlerによって得られた各特徴点の3次元座標とカメラ位置が含まれる。これらはすべて上述

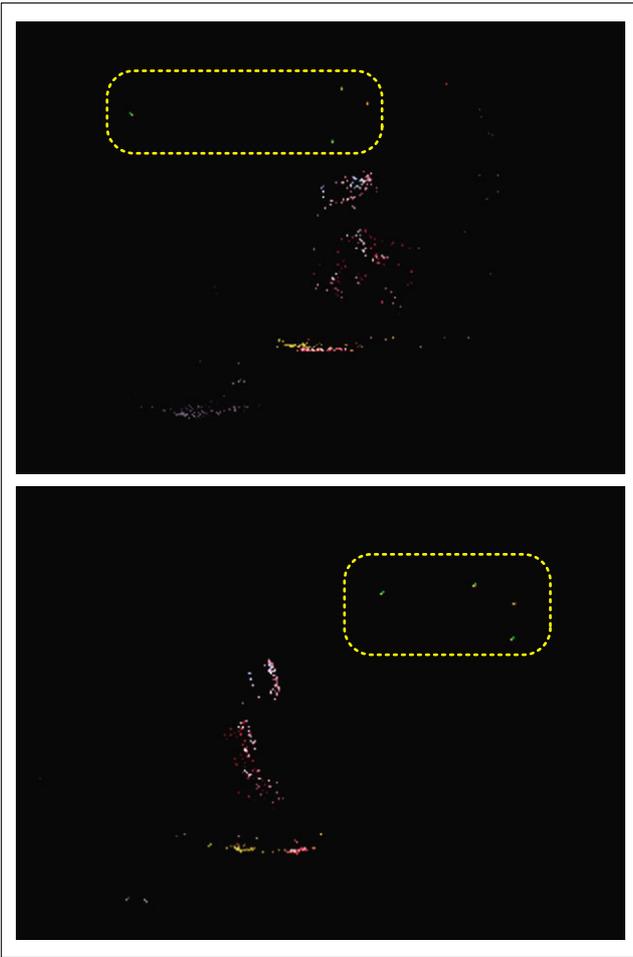


図3 Bundlerで得られる疎な3次元点群とカメラ位置

のbundle.outに内在する情報だが、PLY形式で書かれているため、後述するMeshLab⁶⁾等の3次元点群表示・編集ソフトで開いて結果を視覚的に確認することができる。また、PLYファイルの読み書き用のライブラリー⁷⁾⁸⁾も公開されており、これらを使用すれば特徴点に対する後処理のプログラムなどを容易に開発することができる。なお、ファイル名の***には復元に使用されたカメラ数が入る。基本的には最も大きい値のファイルを確認すればよい。

5. MeshLabによる推定結果表示

MeshLab⁶⁾は、3次元の点群データやメッシュデータの表示・編集が可能なオープンソースプログラムである。Windows/Linux/MacOSXで動作し、各OS用のインストーラも提供されているため、容易に導入できる。多くのファイルフォーマットをサポートしており、上述のPLY形式も開くことができる。また、点群に対するメッシュ生成、メッシュの位置合わせや統合、欠損修復などに関する最近の手法がいち早く実装されており、これらを容易に利用することができる点で非常に有益である。

このMeshLabで、上述のpoints***.plyを開いた様子を図3に示す。1枚目がフロントビュー、2枚目がサイドビューである。

図中の点線枠内の点が推定されたカメラ位置、それ以外の点がシーン中の3次元点群である。得られる3次元点群は、SIFT特徴量で対応がとれた点のみのため、通常、シーン全体に対して密に得られることはなく、図のように疎な点群として得られる。

6. 密な形状復元のためのPatch-based Multi-view Stereo (PMVS)

シーンの3次元形状復元の目的でSFMを用いる場合、上述のような疎な点群では十分な結果とは言えない。密なシーン形状を獲得するためには、このSFMの結果をもとにMulti-view Stereoなどの形状復元手法を適用しなければならない。このMulti-view Stereoの実装はいくつか存在するが、ここでは、Bundlerとの親和性の高さからY. Furukawa氏らのPatch-based Multi-view Stereo Version 2 (PMVS2)⁹⁾を取り上げる。

実行手順は以下の通りである。まず、Bundlerに用意されているBundle2PMVSコマンドを実行する。

```
% cd image_dir/
% BUNDLER_PATH/bin/Bundle2PMVS list.txt ./bundle/bundle.out
```

これにより、image_dir内にpmvsというディレクトリが作成され、その中にいくつかのファイルも生成される。次に、pmvs/prep_pmvs.shをエディタで開き、BUNDLER_BIN_PATHを以下のように書換える。

```
# Script for preparing images and calibration data
# for Yasutaka Furukawa's PMVS system

BUNDLER_BIN_PATH=BUNDLE_PATH/bin }ここを書き換える
if [ "$BUNDLER_BIN_PATH" == "" ]; then
    .
    .
    .
```

書換えたスクリプトを実行した後、最後にPMVS2を実行する。

```
% sh pmvs/prep_pmvs.sh
% cd pmvs
% PMVS2_PATH/program/main/pmvs2 ./pmvs_options.txt
```

3次元形状復元データはpmvs/models/に生成される。同ディレクトリ内のpmvs_options.txt.plyをMeshLabで開いた様子を図4に示す。SFMの結果と比較してシーンが密に復元されている様子が確認できる。

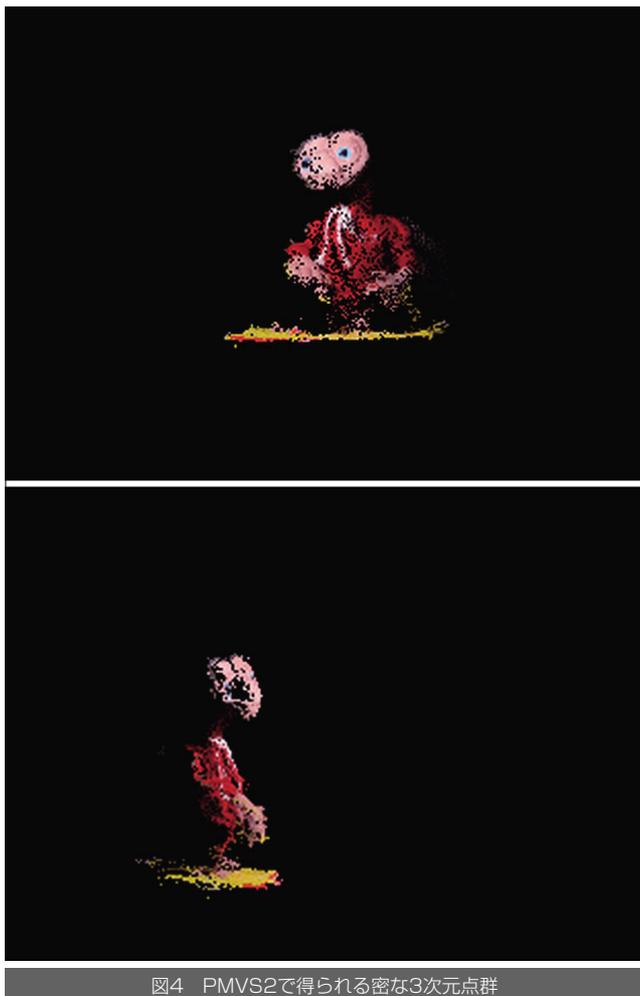


図4 PMVS2で得られる密な3次元点群

7. む す び

本稿では、Structure from Motion (SFM) についてその要素技術や経緯を簡単に述べるとともに、同技術を容易に実行するためのオープンソースプログラムBundlerを紹介

介した。また、Bundlerに関連するソフトウェアとしてMeshLabとPMVS2を紹介した。MeshLabはBundlerの実行結果のビューアとしても使用できる高機能な3次元データ表示・編集ソフトであり、PMVS2はBundlerの結果をもとにMulti-view Stereoを行い密な3次元形状復元を行うプログラムである。いずれのソフトウェアもインストールや実行方法が簡単なので、物体の3次元形状復元やカメラのモーショントラッキングのためのツールとして大いに活用されたい。

(2010年12月21日受付)

〔文 献〕

- 1) C. Tomasi, T. Kanade: "Shape and motion from image streams under orthography: A factorization method", International Journal of Computer Vision, 9, 2, pp.137-154 (1992)
- 2) D.G. Lowe: "Object recognition from local scale-invariant features", Proc. of IEEE International Conference on Computer Vision, pp.1150-1157 (1999)
- 3) H. Bay, A. Ess, T. Tuytelaars, L.V. Gool: "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding, 110, 3, pp.346-359 (2008)
- 4) N. Snavely, S.M. Seitz, R. Szeliski: "Modeling the World from Internet Photo Collections", International Journal of Computer Vision, 80, 2 (2008)
- 5) <http://blogs.oregonstate.edu/hess/code/sift/>
- 6) <http://meshlab.sourceforge.net/>
- 7) <http://people.cs.kuleuven.be/~ares.lagae/libply/>
- 8) http://www.cc.gatech.edu/projects/large_models/ply.html
- 9) Y. Furukawa J. Ponce: "Accurate, Dense and Robust Multi-View Stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)



みつがみ いくりき
満上 育久 2001年、京都大学工学部卒業。2007年、奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年、京都大学学術情報メディアセンター研究員。2010年、大阪大学産業科学研究所助教。ジオメトリを中心としたコンピュータビジョン、画像理解、対象抽出・追跡等に関する研究に従事。博士(工学)。