Improving Initial Estimations for Structure from Motion Methods

Christopher Schwartz

Institute of Computer Science II
University of Bonn
Outline

- Motivation
- Computer-Vision Basics
  - Stereo Vision
  - Bundle Adjustment
  - Feature Matching
- Global Initial Estimation
- Component Merging
- Results
- Conclusion
Motivation

SfM used in:

- Computer Vision
- Photogrammetry
- Computer Graphics
- Autonomous Systems
Motivation (cont'd)

Snavely et al. 2006
“Phototourism”

Microsoft Photosynth
**Structure from Motion**

From (unordered) photo collection:
- sparse point cloud of scene geometry
- positions and orientations of cameras
Structure from Motion (cont'd)

“Top-Down”-Approach:

- widely used
  - Brown & Lowe 2005, Snavely et al. 2008 (freely available “bundler”), Microsoft Photosynth

Images

- camera registration & point triangulation
- initial estimate
- optimization
- points & cameras
Example: Hallway
Example: Hallway
Example: Hallway
Example: Hallway (cont'd)
Example: Hallway (cont'd)
Motivation (cont'd)

Find initial model:

- current methods sometimes fail to find one single model
- use more input data
  - expensive ?
  - impossible ?
- use our method
  - works in many cases
  - speeds up the process
Motivation (cont'd)

Find initial model:

- current methods sometimes fail to find one single model
- use more input data
  - expensive ?
  - impossible ?
- use our method
  - works in many cases
  - speeds up the process
Basics

Image Plane

Camera Lens

Object
Basics (cont'd)

\[ x' = C \cdot x \]

Object (world coordinates)

Image (image coordinates)

Projection Matrix
Basics (cont'd)

Measure

\[ x'_1, x'_2 \]

Calculate

\[ x' = C \cdot x \]
Relative Orientation

- 2 cameras
- find homologous points
- rays through points and projection centers
- intersection of rays = original point in 3D
- relative to arbitrary coordinate system
- David Nistér (2004): directly solvable with 5 homologous points
Bundle Adjustment

- Triggs et al., 1999
- Maximum Likelihood Estimator
- gold standard in image registration
- global solution for arbitrary number of correspondences (or images and 3D points)
- easily stuck in bad local minimum
- requires good initial (global) model
Structure from Motion

1. find global initial estimation
2. run bundle adjustment

- How to find homologous points?
- How to get initial global model?
**Feature Detection & Matching**

How to find homologous points?

- **manually (user clicks on pixels):**
  - error prone
  - lengthy and dull activity

- **automatic feature detection and matching:**
  - W. Förstner, 1986: detection and matching of points
  - D.G. Lowe, 2004: SIFT - scale invariant features
  - recent research: affine transformation invariant features
Feature Detection & Matching (cont'd)
Feature Detection & Matching (cont'd)
Feature Detection & Matching (cont'd)

- problem with outliers
- use RANSAC for outlier detection
Feature Detection & Matching

- problem with outliers
- use RANSAC for outlier detection
- works well with *enough* detected correspondences
Initial Global Estimation

- how to find initial global model?
- example: 3 images
- easy:
  - create stereo model with 2 images
  - add 3rd image
- requires at least one common point!

Feature detected and matched in all 3 images, not removed by RANSAC!
Initial Global Estimation (cont'd)

Example: bundler's approach to an initial estimation

- choose initial pair (stereo model)
- add next image with points observed by ≥2 registered cameras to global model
- (run small bundle-adjustment on new camera)
Example: Store (checkout area)
Example: Store (checkout area)
Problems

- too few features:
  - no feature seen by \( \geq 2 \) registered cameras
  - bundler fails to add next image

- restart with unregistered images

→ segmentation in several components
Problems (cont'd)

Speed and memory consumption:
- feature matching is $O(n^2)$
- building global model slow and memory consuming
Component Merging
practical experience:

- different components have common images
Component Merging
practical experience:
- different components have common images
Component Merging (cont'd)

- ≥ 2 common images:
  - orientation ✓
  - position ✓
  - scale ✓

- can combine components!
Component Merging (cont'd)

We propose:

- register images like before
- restart process:
  - pick still unregistered images as initial pair
- merge resulting components afterwards
Component Merging (cont'd)

2 components, ≥ 2 cameras overlap

- use RANSAC to eliminate outliers
  = incorrectly registered cameras

- find optimal transformation (least squared sense)

- transform cameras and points in 2\textsuperscript{nd} component

- remove doubled points and cameras
Component Merging (cont'd)

≥ 3 components?
Component Merging (cont'd)

connection graph of components
Tree Optimization

With connection graph:

- find minimal spanning tree
  → minimizes accumulation of errors
- graph might contain loops
  → possibly useful
- see Bendels et al. 2004
Component Merging (cont'd)

- only small overlap between components
  → fast calculation of transformation

- sparse pointcloud
  → only few ops when merging (< 1M)

- merging components is a matter of seconds

- bundle adjustment step as post processing
  → accumulation of small errors negligible
Results - Hallway example

- 480 images
Results - Hallway example

- 480 images
- sparse feature environment
Results - Hallway example

- 480 images
- sparse feature environment
- unwanted segmentation into two components
- 14 images overlap
Results - Hallway example
Results - Hallway example

- with bundler or Photosynth: always 2 components
- one single model possible using our method
Results - Store example

- 1749 images
Results - Store example

- 1749 images
- feature rich environment
Results - Store example

- 1749 images
- Feature rich environment
- Manually divided input set of images:
  - 7 components
  - About 10 images overlap
Results - Hallway example
Results - Hallway example

- using bundler with all images at once:
  - ran for several days
  - Then exceeded 8GB of memory and crashed

- using manually divided set and our method:
  - parallel computation possible:
    - ~22 h on four Q6600
  - does not exceed memory
Conclusion

- improved initial estimation of scene model
- applicable in many situations
- reduces time and memory consumption

- no guarantee if segmented unintentionally

future work:
  - Automatic partition of input set (e.g. histogram based)
  - Comparison to “Bottom-Up” approaches
Thank you!

Questions?
\[
\begin{pmatrix}
    x' \\
    y' \\
    1
\end{pmatrix} = K \cdot \begin{pmatrix} R & -t \end{pmatrix} \cdot \begin{pmatrix}
    x \\
    y \\
    z \\
    1
\end{pmatrix}
\]
Basics (cont'd)

\( \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = K \cdot \begin{pmatrix} R & -t \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \)

5 intrinsic parameters

6 extrinsic parameters:
- rotation \( R \)
- translation \( t \)
Common simplification: **Pinhole-camera**

Intrinsic calibration:
- $c$: camera constant
- $h$: principal point
- $s$: axis shear factor
- $m$: axis scale factor

$$K = \begin{pmatrix} c & s & h_x \\ 0 & c \cdot (1 + m) & h_y \\ 0 & 0 & 1 \end{pmatrix}$$
Relative Orientation (cont'd)

\[ x_1 = C_1 \cdot x \]
\[ x_2 = C_2 \cdot x \]
\[ C_1^{-1} \cdot x_1 = C_2^{-1} \cdot x_2 = x \]
Arbitrary Scale in Stereo Models
Combining Stereo Models (cont'd)

For triplets of images:

- 2 stereo models & common camera:
  → combining not possible

- at least one common point $x$
  - $x'$ in first relative model
  - $x''$ in second relative model

Feature detected and matched in all 3 images, not removed by RANSAC!

Scale

$$s = \frac{|x' - t|}{|x'' - t|}$$