ACCV’16 Tutorial: Large-scale 3D Reconstruction from Images

LONG QUAN, TIANWEI SHEN, JINGLU WANG
Part I
Tianwei Shen

Large-scale Structure-from-Motion: A Modern Synthesis
Outline

- Introduction to Structure-from-Motion (SfM)
- Component I: Feature Detection and Matching
- Component II: From Feature matches to 3D
- Component III: Large-scale Bundle Adjustment
- Applications and Future Directions
SfM - The entry point to 3D computer vision

- From pictures to 3D scenes

---

Imagery Credit: Hanyu@altizure.com

Not covered in this talk
Notations

- Views/Frames/Images: \{I_i\}
- Features: 2D salient regions/blobs (edges, corners), e.g. SIFT
- Tracks: 3D point structures that correspond to 2D features in images
- Camera Intrinsic / Extrinsic: \{P_i\} => K [R T]
- Residual error: distance between 2D features and 3D projection
- Triangulation: the process of determining a point in 3D space given its projections onto two, or more images
A typical pipeline of SfM

- Feature extraction: images \{I_i\} \rightarrow local feature collections \{F_i\}
- Feature matching: \{F_i\} \rightarrow match pairs \{M_{ij}\}, epipolar geometry \{f, h, R_{ij}, t_{ij}\}
- Match graph construction: \{M_{ij}, R_{ij}, t_{ij}\} \rightarrow camera poses \{P_i\}, tracks \{p_k\}
  - Graph initialization (select a robust initial match pair to build a metric reconstruction)
  - How we add edges to the match graph (global / incremental)
- Bundle adjustment: \{P_i\}, \{p_k\} \rightarrow optimized \{P_i\}, \{p_k\}
- *Building Rome in a day* (2009) – the first practical large-scale SfM system
SfM is just a large-scale optimization problem

- 2-view/3-view optimization (epipolar geometry)
- Match graph optimization
- Pose averaging
- Bundle adjustment (non-linear least squares)
Topic I: Local Features and Matching

- Local feature - the basis for SfM
- Scale Invariant Feature Transform (SIFT)
  - Scale-space extrema detection
  - Keypoint localization
  - Orientation assignment
  - Keypoint description
- Invariant to translation, scaling, and rotation
Problems with feature matching

- Tradeoff: SIFT is not invariant under geometric transformations

- Problem 1: Pairwise feature matching is costly.

- Problem 2: Erroneous matches is evitable, thus robust estimation is used.

  An extreme case:

  ![Front-front match](image1.png)  ![Erroneous front-back match](image2.png)
To tackle problem 1: matching efficiency

- Use image retrieval to compute a candidate match set
- Vocabulary tree: train -> build -> match
- Reduce cost from $O(n^2)$ to $O(kn)$, $k$ decided by users
- Main problem with this approach:
  - $k$ is not known beforehand
  - Too small $k$ is not sufficient
  - Too large $k$ slows down the process

To tackle problem 1: matching efficiency

- **Other approaches:**
  - Relevance feedback and entropy minimization ([1] Lou et al.)
  - Match features in larger pyramid scale ([2] Wu)
  - Learning-based method to predict overlaps ([3] Schoënberger et al.)
  - A hashing-based cascading matching ([4] Cheng et al.)

To tackle problem 2: erroneous matches

- Identification and removal of erroneous epipolar geometry is a recent research focus for SfM.
- Can lead to catastrophic results for SfM.
To tackle problem 2: erroneous matches

- Loop consistency [1]:
  - Chained relative motion should be an identity map: \( R_{12}R_{23}R_{31} = I \)
  - Start from a full match graph
  - Sample cycles from the full graph
  - The problem is casted as a Bayesian inference task
  - Strong assumption on variable independence

---

To tackle problem 2: erroneous matches

- Other works:
  - Sampling match graph based on missing correspondences and time stamp cue. [1]
  - Analysis of visibility graph. [2]
  - Splits the camera graph and then leverages conflicting observations. [3]

Motivation: Solve two problems together

- All disambiguation methods start from a relatively full match graph
- Construct an error-free match graph in a bottom-up fashion
- Select a sufficient match set that can guarantee a reconstruction
- Prevent additions of erroneous pairs
Graph-based consistent matching for SfM

- **Multi-stage matching process:**
  - **Stage 1:** Starts from a minimal spanning tree based on vocabulary tree ranks
  - **Stage 2:** Expand the spanning tree with loop consistency guaranteed
  - **Stage 3:** Find loop closures by community detection

Graph-based consistent matching for SfM

Stage 1: start from a minimal spanning tree
- The purpose is to quickly chain the views

- A modified Kruskal’s algorithm (online version): reject outliers

- Edge weight parameterized by rank information given by vocabulary tree:

\[ w(e_{ij}) = \sqrt{\frac{\text{Rank}^2_i(j) + \text{Rank}^2_j(i)}{2}} \]

Graph-based consistent matching for SfM

Stage 2: Graph Expansion by Strong Triplets

- Verifying all loops is hard to achieve, even verifying all triplets is $O(n^3)$

- Generate a consistent match graph in a bottom-up way

- A empirical choice: traversing two steps starting from each node

Graph-based consistent matching for SfM

Stage 3: Community-Based Graph Reinforcement
- Too sparse connection after triplet expansion
- Longer loops are not verified

Community detection: divide a graph into groups with denser connections inside and sparser connections outside.

Graph-based consistent matching for SfM

Results – Internet data

Graph-based consistent matching for SfM

Results – ambiguity data

Future direction: learning local features

- Feature is the most important factor in SfM accuracy
- Deep learning approaches: learning local feature descriptors
- Speed up matching and improve matching accuracy
Topic II: From Feature matches to 3D

- Incremental

- Hierarchical

- Global
Some Recent Representative Architectures

Sequential/Incremental Approaches

Building Rome in a day
Colmap: SfM Revisited

Global Approaches

Hierarchical

Optimizing the Viewing Graph for Structure-from-Motion

Randomized structure from motion based on atomic 3d models from camera triplets

Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion.
## Three SfM Paradigms

<table>
<thead>
<tr>
<th></th>
<th>Incremental</th>
<th>Hierarchical</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature extraction and matching</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Match graph initialization</td>
<td>Initialized by carefully selected two-view</td>
<td>Atomic models</td>
<td>All views are treated equally</td>
</tr>
<tr>
<td>Image Registration</td>
<td>Perspective-n-Point (PnP), 2D-3D correspondences</td>
<td>3D-3D fusion</td>
<td>Rotation and translation averaging</td>
</tr>
<tr>
<td>Bundle adjustment</td>
<td>Iterative, many times</td>
<td>BA when merging</td>
<td>one time</td>
</tr>
<tr>
<td>Advantages</td>
<td>Robust</td>
<td>Fewer BA steps</td>
<td>Evenly-distributed errors</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Prune to drifting errors</td>
<td>Model merging, graph partition</td>
<td>Prune to noisy pairwise matches</td>
</tr>
<tr>
<td>Softwares</td>
<td>Bundler, openMVG, VisualSfM, MVE</td>
<td>Research papers</td>
<td>openMVG, Theia</td>
</tr>
</tbody>
</table>
Key technique: motion averaging

- Correct accumulating errors in chained pose estimation
- First rotation averaging, then translation averaging

Images \{I_i\} → relative motions \{R_{ij}, t_{ij}\} → rotation averaging \{R_i\} → translation averaging \{T_i\} → 3D structures \{X_k\}
Convex optimization in SfM

- Convex optimization becomes popular because of its elegant mathematical forms and the existence of global minimum.

- First investigated by Hartley et al. on triangulation.


- Not practical due to its sensitivity to noises, but theoretically interesting.
Rotation averaging on a graph

Viewing Graph: \( G = (\mathcal{V}, \mathcal{E}) \)

Globally consistent rotation: \( R_{ij} = R_j R_i^{-1}, \; \forall (i, j) \in \mathcal{E} \)

Minimize Riemannian distance: \( d(X, Y) = \| \log(YX^{-1}) \| \)

Rotation average is non-convex

Rotation averaging: other approaches

- Quaternions parameterization (Martinec et al. [1])

- L1 norm based on Weiszfeld algorithm (Hartley et al. [2])

Translation averaging

- Long been characterized as a convex optimization problem

- Min-max formulation, SOCP

- Same L-infinity drawbacks: prune to outliers
Translation averaging

- Considering observed points together (triplet bundle)

- Re-projection error: \( \rho(t_i, X_j) = \left\| \frac{R_i^{(1)T}X_j + t_i^{(1)}}{R_i^{(3)T}X_j + t_i^{(3)}} - \frac{R_i^{(2)T}X_j + t_i^{(2)}}{R_i^{(3)T}X_j + t_i^{(3)}} \right\|_\infty \)

- Linear program minimal case with RANSAC

\[
\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad \rho(t_i, X_j) \leq \gamma, \\
& \quad R_i^{(3)}X_j + t_i^{(3)} \geq 1, \\
& \quad t_i = (0, 0, 0) \forall i, j.
\end{align*}
\]

Translation averaging

Then global translation averaging

Formulation under $L$-infinity:

\[
\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad \|T_j - R_{ij} T_i - \lambda_{ij} t_{ij}\|_\infty \leq \gamma, \\
& \quad \lambda_{ij} \geq 1, \forall i, j \\
& \quad T_1 = (0,0,0).
\end{align*}
\]

Minimizing two-side of $\lambda_{ij} t_{ij} = T_j - R_{ij} T_i$

Translation averaging: robust formulation

- A small robust L1 formulation improvement: consider L1 norm of the re-projection error vector:

\[ (\cdots, \rho(t_i, X_j), \cdots) \]

\[
\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad \rho(t_i, X_j) \leq \gamma, \\
& \quad R_i^{(3)} X_j + t_i^{(3)} \geq 1, \\
& \quad t_i = (0, 0, 0) \forall i, j.
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_{i} \gamma_i \\
\text{subject to} & \quad \rho(t_i, X_j) \leq \gamma_i, \\
& \quad R_i^{(3)} X_j + t_i^{(3)} \geq 1, \\
& \quad t_i = (0, 0, 0) \forall i, j.
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad \|T_j - R_{ij} T_i - \lambda_{ij} t_{ij}\|_\infty \leq \gamma, \\
& \quad \lambda_{ij} \geq 1, \forall i, j \\
& \quad T_1 = (0, 0, 0).
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i,j)} \gamma(i,j) \\
\text{subject to} & \quad \|T_j - R_{ij} T_i - \lambda_{ij} t_{ij}\|_\infty \leq \gamma(i,j), \\
& \quad \lambda_{ij} \geq 1, \forall i, j \\
& \quad T_1 = (0, 0, 0).
\end{align*}
\]

Translation averaging: comparison

- RobustL1 outperforms other methods
Translation averaging: comparison

- However, advantage is not evident after bundle adjustment (BA)

- Also, the problem scale is larger

- Future direction: no large-scale benchmark datasets for testing

- A potential useful settings is SLAM, where BA is costly
Topic 3: Bundle adjustment

- Joint optimization of camera poses and 3D tracks

\[
\min_{P_i \in \mathcal{P}} \sum_{i=1}^{m} \sum_{j=1}^{n} v_{ij} f(u_{ij} - \Pi(P_i, X_j))
\]

- Error model:

\[
f(\Delta z_{ij}) = \frac{1}{2} \Delta z_{ij}^T W_{ij} \Delta z_{ij}
\]

\[
\Delta z_{ij} = u_{ij} - \Pi(P_i, X_j)
\]
Bundle adjustment

- Levenberg-Marquardt algorithm
  - Taylor expansion: \[ f(x + \delta x) \approx f(x) + g^T \delta x + \frac{1}{2} \delta x^T H \delta x, g \equiv \frac{df}{dx}(x), H \equiv \frac{d^2 f}{dx^2}(x) \]
  - Newton step: \[ \frac{d}{dx} f(x + \delta x) \approx H \delta x + g = 0 \implies \delta x = -H^{-1}g \]
  - New value: \[ f(x + \delta x) \approx f(x) - \frac{1}{2} g^T H^{-1}g \]
  - Damped Newton’s methods: \( (H + \lambda W) \delta x = -g \)
Bundle adjustment

- Large-scale endeavors
  - Multi-core bundle adjustment [1]

- Distributed settings [2]

- Essentially a non-linear least square problem, thus generally useful for other vision problems.


What can we do with SfM?

- 3D reconstruction
- Simultaneous localization and mapping (SLAM)
- Test base for local features (distinctiveness, efficiency, matchability)
- Color correction for image collections
- Visual effects
- …
Application: Large-Scale Color Correction

- **Motivation:** Images captured for 3D reconstruction are color-inconsistent

- **Optimize color of image collections, based on geometric information**

Application: Large-Scale Color Correction

- Non-linear optimization on color histogram:

\[
\begin{align*}
& \text{minimize } \sum_{i,j,k} \rho \left( \frac{(s_i Q_{ij}^{(k)} + o_i) - (s_j Q_{ji}^{(k)} + o_j)}{s_i + s_j} \right)^2 \\
& \text{subject to } 1 - \delta_s \leq s_i \leq 1 + \delta_s, -\delta_o \leq o_i \leq \delta_o, \ \forall i.
\end{align*}
\]

\[
\rho(x) = \delta^2 \left( \sqrt{1 + (x/\delta)^2} - 1 \right)
\]

Application: Large-Scale Color Correction

Consistent texturing:

Before

After

Final Remarks

- Merge ground-level street-view images with aerial images
- Better local invariant features and efficient matching
- Distributed everything in SfM