3D Computer Vision

Structure from Motion II

Prof. Didier Stricker

Kaiserslautern University
http://ags.cs.uni-kl.de/

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz
http://av.dfki.de
Outline

• Previous lecture:
  ▪ An introduction of Structure-from-Motion (SfM)
  ▪ Robust estimation of Fundamental matrix $\mathbf{F}$ and Essential matrix $\mathbf{E}$
  ▪ Extraction of cameras from $\mathbf{E}$
  ▪ Structure from Motion loop

• Today:
  ▪ Real-time application of SfM (e.g. VO and SLAM)
  ▪ Triangulation
  ▪ Bundle Adjustment
  ▪ Drift reduction methods
  ▪ State-of-the-art approaches

▪ Next lecture: Dense reconstruction
SLAM - Simultaneous Localization and Mapping

Maria Thomas, Oliver Wasenmüller, Didier Stricker. „Evaluation of ORB-SLAM2“, 2017. [https://youtu.be/OfrPAUyp1_s](https://youtu.be/OfrPAUyp1_s)
Offline SfM vs. Real-time SfM

**Offline SfM:**
- E.g. as basis for dense 3D model reconstruction
- No real-time requirements, all images are available at once

**Real-time SfM:**
- E.g. for mobile Augmented Reality in unknown environments
- Real-time requirements, images become available one by one, output required at each time-step
Offline SfM vs. Real-time SfM

• **Offline SfM:**
  - Uses many features and computationally intense image processing methods
  - Forward and backward feature matching
  - Global Bundle Adjustment (BA)

• **Real-time SfM:**
  - Reduced number of features and lightweight image processing methods
  - Sequential/recursive processing scheme
  - Local Bundle Adjustment (keyframe/window based)

→ harder problem
Structure and Motion - Basic Pipeline

- Image processing
- Point correspondences
- Pose estimation
- Scene model
- Structure estimation
- Camera pose

Additional sensor information?
Real-time SfM (calibrated case)

• Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

How do we initialize the first camera poses?
Recall last lecture: extraction cameras from $E$

\[
\text{2D feature location (from image processing)} \quad t = 1
\]

\[
\text{Cameras extracted from } E
\]

\[
\begin{align*}
\mathbf{m}_{n,t}^{(j)} \\
\mathbf{s}_t = \{R_{cw}, c_w\}_t
\end{align*}
\]
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

Now we have the two first camera poses. How do we go on?

\[ t = 1 \]
\[ t = 2 \]

2D feature location (from image processing)

Cameras extracted from \( E \)
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

\[ t = 1 \]
\[ t = 2 \]

3D feature location

2D feature location (from image processing)

Cameras extracted from E

Triangulate 3D points
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

\[ t = 1 \]
\[ t = 2 \]
\[ t = 3 \]

3D feature location

Estimate next camera pose (now from 2D/3D correspondences)

Cameras extracted from \( E \)

\[ s_t = \{ R_{cw}, c_w \}_t \]
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

\[ t = 1 \]

\[ t = 2 \]

\[ t = 3 \]

\[ \mathbf{s}_t = \{R_{cw}, c_w\}_t \]

Triangulate additional 3D points
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

Refine known 3D points with new camera poses
Real-time SfM (calibrated case)

- Alternating estimation of camera poses and 3D feature locations (triangulation) from a (continuous) image sequence.

\[
t = 1 \quad t = 2 \quad t = 3
\]

**E.g. for some selected keyframes or more extensively in offline SfM**

Refine known cameras with new 3D points

\[
m_{n,t}^{(j)} \quad s_t = \{R_{cw}, c_w\}_t
\]
Triangulation
Triangulation

- Given at least 2 known cameras $C$ and $C'$, and 2 corresponding feature points $x$ and $x'$ (i.e. 2 camera views).

- Estimate the 3D point $X$

Then, $y = y'$ and depth $Z$ can be computed from disparity:

$$Z = \frac{ft_x}{d}$$

$\Delta = x' - x$.

Derivation via equal triangles:

$$\frac{x}{f} = \frac{X}{Z}$$

$$\frac{x'}{f} = \frac{X + t_x}{Z}$$

$$\frac{x'}{f} = \frac{x}{f} + \frac{t_x}{Z}$$

Note: feature displacement (disparity) is inversely proportional to depth as $d \to 0, Z \to \infty$. 

$K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$R = I$

$t = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}$
Triangulation - Vector Solution

- Due to noise, two rays may not intersect
- Compute the mid-point of the shortest line between the two rays

How can this be solved algebraically?
Triangulation - Algebraic Solution

- Equation for one camera view (camera pose, and feature point):

\[ x \propto PX \iff \lambda x = PX \iff x \times \begin{bmatrix} P^{1T} \\ P^{2T} \\ P^{3T} \end{bmatrix} X = 0 \]

- Multiple camera views:

\[ x \times PX = 0 \quad \text{and} \quad x' \times P'X = 0 \]
Triangulation - Algebraic Solution

• One camera view can provide 3 equations, only 2 linearly independent
  $\Rightarrow$ drop third row

$P = K[R|t] = \begin{bmatrix} p^{1T} \\ p^{2T} \\ p^{3T} \end{bmatrix}$

\[
\begin{align*}
x(p^{3T}X) - (p^{1T}X) &= 0 \\
y(p^{3T}X) - (p^{2T}X) &= 0 \\
x(p^{2T}X) - y(p^{1T}X) &= 0
\end{align*}
\]

• Construction of a linear system

$\begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \end{bmatrix} X = 0$

2x4 matrix $A$
Overdetermined spatial intersection using projective geometry.
Triangulation - Algebraic Solution

• Stack equations for all camera views:

\[
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_n
\end{bmatrix}X = 0
\]

- Again, the linear system \(AX=0\) can be solved by SVD
- Recall:
  - Minimize \(\|AX\|_2^2\), subject to \(\|X\|_2^2 = 1\).
  - Then normalize the 3D point \(X\) by dividing the 4th entry of \(X\).

• However, it would be better to find the 3D point that minimize a meaningful geometric error, like the re-projection error:

\[
\epsilon_i = d(x, PX_i)^2 + d(x', P' X_i)^2
\]
• Estimate 3D point \( \hat{X} \), which exactly satisfies the supplied camera geometry \( P \) and \( P' \), so it projects as

\[
\hat{x} \propto PX \quad \quad \hat{x}' \propto P'X
\]

- If the measurement noise in image points is Gaussian with mean equal to zero, minimizing the reprojection error gives the Maximum Likelihood estimation of \( X \).
- Where \( \hat{x} \) and \( \hat{x}' \) are closest to the actual image measurements.

Assumes perfect camera poses! \( \rightarrow P \) is error free

Nonlinear problem: can be solved with e.g. Levenberg-Marquard \( \rightarrow \) parameter estimation lecture

\[
\min_{\hat{X}} d(x, \hat{x})^2 + d(x, \hat{x}')^2 \quad \text{subject to} \quad \hat{x} \propto P\hat{X} \quad \text{and} \quad \hat{x}' \propto P'\hat{X}
\]
Triangulation – Properties

- The smaller the angle (small baseline, big distance), the bigger the reconstruction uncertainty!
Bundle Adjustment
Bundle Adjustment (BA)

• Valid poses estimation \([R_1|t_1], [R_2|t_2], [R_3|t_3] \ldots\) and 3D points \(X^1, X^2, X^3 \ldots\) must let the re-projection close to the observation, i.e. to minimize the reprojection error

\[
\arg\min_{X^i, R_i, t_i} \sum_i \sum_j \| x^j_i - K[R_i|t_i]X^j_i \|^2_2
\]

• Global Bundle Adjustment (BA):
  - Refine the visual reconstruction to produce jointly optimal camera poses and 3D points

\[
\arg\min_{i, j} \sum_i \sum_j r^j_i
\]
Bundle Adjustment

- 6 parameters for each camera + 3 parameters for each 3D point
  (In SLAM: $K$ is assumed to be known and fixed, so not updated here)

  - parameters must be estimated
  - matrices are sparse!

  • Minimize the cost function:

  $$
  \arg\min_{X_j, R_i, t_i} \sum_i \sum_j \left\| x_i^j - K[R_i|t_i]X_j^j \right\|_2^2
  $$

  - non-linear optimization: e.g. Levenberg-Marquardt

  • State-of-the-art solvers:

    - g2o: [https://openslam-org.github.io/g2o.html](https://openslam-org.github.io/g2o.html)
Drift
Drift – A Big Problem

- **Drift: accumulating error**
  - Both offline SfM and SLAM will drift (diverge) very quickly
  - Measurement has errors
  - Uncertainties in the camera poses propagate to the triangulated 3D point points, and vice versa
Drift – Initializing with three images (instead of two)

- The length of the translation vector is set to 1 ($\|t\| = 1$) for two views.
- This defines the scaling (an arbitrary scaling) of the “world“.
- For a third view, the scaling must be computed and coherent with the defined scale.

- $R_{12}, R_{13}, R_{23}$ and $t_{12}, t_{13}, t_{23}$ are the rotation matrices and translation vectors among the three views 1, 2, and 3.

\[
P_1 = K_1(I, 0) \\
P_2 = K_2(R_{12}, t_{12}) \\
P_3 = K_3(R_{13}, \lambda_{13} t_{13})
\]

- $\lambda_{13}$ is the scaling factor between view 1 and 3. Then, we have:

\[
\lambda_{13} = \frac{(t_{23} \times t_{13})^T (t_{23} \times R_{23} t_{12})}{\|t_{23} \times t_{13}\|^2}
\]
Drift Reduction – Methods

• **Offline SfM:**
  - Normally the reprojection error is minimized through global Bundle Adjustment, where the estimated 3D points and camera poses will be refined globally as discussed before.

• **Real-time SfM:**
  - Filter based: estimated 3D points and camera poses could be refined recursively in Extended Kalman Filter (EKF).
  - Keyframe based: for improving real-time efficiency, local Bundle Adjustment is used only on keyframes (selected good frame), combined with global Bundle Adjustment when it is necessary.
  - Loop closure: recognize the place where the camera visited before, close the loop and eliminate drift.
Drift Reduction – Feature Level

- Reduce drift in feature tracking/matching
- Extend feature tracks
- Reacquire lost features

- See topics: image processing, recursive filtering, advanced visual detection and tracking algorithms
Drift Reduction – Geometric Level

• Careful (precise, robust) 3D point initialization
• How?
  ▪ Triangulate over a whole set of camera views (>> 2)
  ▪ Enforce a minimal angle $\theta$ between view rays
  ▪ Use RANSAC to eliminate outliers
  ▪ Use only well reconstructed points for further estimation

• Incorporate uncertainties, e.g. simple stochastic model and WLS (weighted least square) estimation $\Rightarrow$ lecture on parameter estimation: all entities modelled as Gaussian random variables
Initial triangulation using RANSAC

Algorithm:
- If $\theta \geq \theta_0$ : store the new camera view
- If size of history $> n$
  - Until no more options or valid result found
    - Triangulate pairs of camera views
    - Validate the results
- If a valid result is found:
  - Compute a least squares (LS) estimate from all inliers
  - Compute a weighted least squares (WLS) estimate from all inliers

See lecture on parameter estimation!
3D point refinement

- Incorporate new camera view, each time the feature is observed in an image
- Methods:
  - Repeated triangulation
  - Recursive filtering
    (e.g. Extended Kalman filter)
SLAM - Results (360° rotation)

With drift reduction

Without drift reduction
Loop closure

- Detection and recognition of already visited and mapped area after the camera exploring a long time

image adapted from: Wolfram Burgard
Loop closure

- Detection and recognition of already visited and mapped area after the camera exploring a long time

image adapted from: Wolfram Burgard
Scene recognition

- The content of an image can be inferred from the frequency of words
  - visual words = independent features
  - Represent the images based on a histogram of word occurrences (bag)
  - Each detected feature is assigned to the closest entry in the codebook

Slide adapted from: L. Fei-Fei and Wolfram Burgard
Bag-of-Words Representation

feature detection & representation

codewords dictionary

image representation

slide adapted from: L. Fei-Fei and Wolfram Burgard
Bag-of-Words Representation

feature detection

Word 753

compute descriptor vectors

quantize

Slides adapted from: Wolfram Burgard and Cyrill Stachniss
Example:

- **Appearance based loop closure:**
  - Construct or train a dictionary/vocabulary of representative words of the image in advance
  - Transform every keyframe into a Bag of Words (BoW) vector and saved in a dataset.
  - Query the dataset and compute similarity of BoW vector of different images, find the best match
  - Compute 3D similarity transformation to close the loop and propagate correction
  - Again, Bundle Adjustment used to refine the map globally

---

Loop detection

**ORB-SLAM**

State-of-the-art approaches

Published at International Symposium on Mixed and Augmented Reality (ISMAR)
Bleser et al., ISMAR 2006

- **Title:** *Online camera pose estimation in partially known and dynamic scenes*

- Real-time structure and motion for Augmented Reality applications

- **Topics:**
  - feature tracking with optical flow, weighted least squares estimation, recursive filtering for structure estimation, feature quality tracking, map management
2D feature tracking (Lost, Valid)

3D augmentation

- Line model
- Projected 3D covariances
- Feature quality (colour gradient)
- External view on 3D uncertainty ellipsoids

Bleser et al., ISMAR 2006
Klein and Murray, ISMAR 2007

• Title: *Parallel tracking and mapping for small AR workspaces*

- Known as PTAM system
- MANY features, (simple) correlation-based tracking
- Parallel pose tracking and 3D reconstruction threads
- Local bundle adjustment (based on keyframes)
- Code, videos, papers, slides available [here](#)
Ventura and Höllerer, ISMAR 2012

- Title: *Wide-Area Scene Mapping for Mobile Visual Tracking*

  - Offline structure and motion ⇒ tracking model (sparse point cloud)
  - Online tracking and model extension using mobile devices
  - Gyroscopes as additional sensors for capturing quick rotations
  - Server-side localization, jitter reduction

- Paper
- Video
- Code
Ventura and Höllerer, ISMAR 2012
Tan et al., ISMAR 2013

• **Title:** *Robust monocular SLAM in Dynamic Environments*

  - Based on PTAM
  - Uses invariant features (Scale Invariant Feature Transform)
  - Explicitly handles occlusions and moving scene parts ⇒ keyframe updating
  - Variation of RANSAC called PARSAC (prior-based sampling + enforcing even distribution of inliers, not just high inlier ratio)
Robust Monocular SLAM in Dynamic Environments

Submission # 293
ISMAR 2013
Thank you!

Next lecture: Dense reconstruction