3D Reconstruction Pipelines

GVG 2022 - Lecture 13
GVG - Brief Summary

- Previously in the GVG lecture:
  - Absolute camera pose estimation
  - Homograph estimation
  - Fundamental matrix estimation
  - Reconstruction from two views
  - Essential matrix estimation

- This lecture: Putting things together for full 3D reconstruction
Structure-from-Motion (SfM)

Input: images

Output: (sparse) 3D point cloud, camera poses

model computed using Colmap
Sequential / Incremental SfM
Sequential / Incremental SfM

Detect interest points and extract descriptors for them, e.g., SIFT features (see lecture 06)
Sequential / Incremental SfM

- Nearest neighbor search in descriptor space to establish feature matches
- Robust model fitting via RANSAC
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model > $\eta$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$
Estimate model from $n$ random data points

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model > $\eta$
Estimate model from $n$ random data points
Estimate support inliers of model

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model >\( \eta \)

- Estimate model from \( n \) random data points
- Estimate support \textbf{inliers} of model
- If new best model
  - update best model, \( \eta \)

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$

   Estimate model from $n$ random data points

   Estimate support inliers of model

   If new best model

      update best model, $\eta$

Return: Model with most inliers

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Torsten Sattler
RANSAC

2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Repeat:

- Draw Minimal Sample
- Estimate Model
- Count Inliers
RANSAC

2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

best number of inliers: 3

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:
- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 3

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Torsten Sattler
RANSAC

2D line fitting example

Repeat:

- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 3

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:
- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:

- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

Torsten Sattler
RANSAC

2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:

1. Draw Minimal Sample
2. Estimate Model
3. Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:

- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 4

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC

2D line fitting example

Repeat:
- Draw Minimal Sample
- Estimate Model
- Count Inliers

best number of inliers: 5

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

• Let’s assume we know the **inlier ratio** $\varepsilon$ (fraction of inliers)
RANSAC - Termination Criterion

- Let’s assume we know the **inlier ratio** $\varepsilon$ (fraction of inliers)
- Probability of picking an inlier randomly: $\varepsilon$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Let’s assume we know the **inlier ratio** $\varepsilon$ (fraction of inliers)
- Probability of picking an inlier randomly: $\varepsilon$
- Probability of picking $n$ inlier randomly: $\varepsilon^n$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Let’s assume we know the inlier ratio $\varepsilon$ (fraction of inliers)
- Probability of picking an inlier randomly: $\varepsilon$
- Probability of picking n inlier randomly: $\varepsilon^n$
- Probability of non-all inlier sample ($\geq 1$ outlier): $(1-\varepsilon^n)$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Let’s assume we know the inlier ratio $\varepsilon$ (fraction of inliers)
- Probability of picking an inlier randomly: $\varepsilon$
- Probability of picking $n$ inlier randomly: $\varepsilon^n$
- Probability of non-all inlier sample ($\geq 1$ outlier): $(1-\varepsilon^n)$
- Probability of not picking all-inlier sample in $k$ iterations: $(1-\varepsilon^n)^k$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if $(1 - \varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations $k_{\text{max}}$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if $(1-\varepsilon^n)^k < \eta$
- In practice: Compute maximum number iterations $k_{\text{max}}$
- Find $k_{\text{max}}$ such that $(1-\varepsilon^n)^{k_{\text{max}}} = \eta$

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if \((1-\varepsilon^n)^k<\eta\)
- In practice: Compute maximum number iterations \(k_{\text{max}}\)
  - Find \(k_{\text{max}}\) such that \((1-\varepsilon^n)^{k_{\text{max}}} = \eta\)
  \[ k_{\text{max}} \ln(1-\varepsilon^n) = \ln(\eta) \]

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

• Terminate if \((1-\varepsilon^n)^k < \eta\)

• In practice: Compute maximum number iterations \(k_{\text{max}}\)
  
  • Find \(k_{\text{max}}\) such that \((1-\varepsilon^n)^{k_{\text{max}}} = \eta\)
  
  \[ \Leftrightarrow k_{\text{max}} \ln(1-\varepsilon^n) = \ln(\eta) \]
  
  \[ \Leftrightarrow k_{\text{max}} = \ln(\eta) / \ln(1-\varepsilon^n) \]

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if \((1-\varepsilon^n)^k < \eta\)
- In practice: Compute maximum number iterations \(k_{\text{max}}\)
  
  - Find \(k_{\text{max}}\) such that \((1-\varepsilon^n)^{k_{\text{max}}} = \eta\)
  
  \[ \Leftrightarrow k_{\text{max}} \ln(1-\varepsilon^n) = \ln(\eta) \]
  \[ \Leftrightarrow k_{\text{max}} = \frac{\ln(\eta)}{\ln(1-\varepsilon^n)} \]
  
  - Note: \(k_{\text{max}}(\varepsilon) > k_{\text{max}}(\varepsilon')\) if \(\varepsilon < \varepsilon'\)

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if \((1-\varepsilon^n)^k<\eta\)
- In practice: Compute maximum number iterations \(k_{\text{max}}\)
  - Find \(k_{\text{max}}\) such that \((1-\varepsilon^n)^{k_{\text{max}}} = \eta\)
    \[ k_{\text{max}} \ln(1-\varepsilon^n) = \ln(\eta) \]
    \[ k_{\text{max}} = \frac{\ln(\eta)}{\ln(1-\varepsilon^n)} \]
  - Note: \(k_{\text{max}}(\varepsilon) > k_{\text{max}}(\varepsilon')\) if \(\varepsilon < \varepsilon'\)
- How do we know inlier ratio \(\varepsilon\)?

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
RANSAC - Termination Criterion

- Terminate if \((1-\varepsilon^n)^k<\eta\)
- In practice: Compute maximum number iterations \(k_{max}\)
  - Find \(k_{max}\) such that \((1-\varepsilon^n)^{k_{max}} = \eta\)
    \[ k_{max} \ln(1-\varepsilon^n) = \ln(\eta) \]
    \[ k_{max} = \frac{\ln(\eta)}{\ln(1-\varepsilon^n)} \]
  - Note: \(k_{max}(\varepsilon) > k_{max}(\varepsilon')\) if \(\varepsilon < \varepsilon'\)
  - How do we know inlier ratio \(\varepsilon\)?
  - In practice more than \(k_{max}(\varepsilon)\) steps necessary as not every all-inlier sample leads to best model (due to, e.g., noise, degeneracies, etc.)

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$

- See also USAC [Raguram et al., PAMI’13] [code] (good overview, nice implementation)
- Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$

Estimate model from $n$ random data points

- See also USAC [Raguram et al., PAMI’13] [code] (good overview, nice implementation)
- Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$

Estimate model from $n$ random data points

Estimate support ($\#\text{inliers} / \text{robust cost func.}$) of model

• See also USAC [Raguram et al., PAMI’13] [code] (good overview, nice implementation)

• Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model > \eta

Estimate model from \( n \) random data points

Estimate support (\#inliers / robust cost func.) of model

- See also USAC [Raguram et al., PAMI’13] [code] (good overview, nice implementation)
- Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model > \( \eta \)
   Estimate model from \( n \) random data points
   Estimate support (\#inliers / robust cost func.) of model
   If new best model
      Perform Local Optimization (LO)

- See also USAC [Raguram et al., PAMI'13] [code] (good overview, nice implementation)
- Never use standard RANSAC!

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]
[Lebeda, Matas, Chum, Fixing the Locally Optimized RANSAC. BMVC 2012] [code]

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANdom SAmple Consensus (RANSAC)

While probability of missing correct model $> \eta$
  Estimate model from $n$ random data points
  Estimate support ($\#inliers / \text{robust cost func.}$) of model
  If new best model
    Perform Local Optimization (LO)
    update best model, $\eta$

• See also USAC [Raguram et al., PAMI'13] [code] (good overview, nice implementation)
• Never use standard RANSAC!

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]
[Lebeda, Matas, Chum, Fixing the Locally Optimized RANSAC. BMVC 2012] [code]

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]
Model Fitting With RANDom SAmple Consensus (RANSAC)

While probability of missing correct model $\eta$
  Estimate model from $n$ random data points
  Estimate support (#inliers / robust cost func.) of model
  If new best model
    Perform Local Optimization (LO)
      update best model, $\eta$
  Return: Model with most inliers / lowest cost

• See also USAC [Raguram et al., PAMI'13] [code] (good overview, nice implementation)
• Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]

[Lebeda, Matas, Chum, Fixing the Locally Optimized RANSAC. BMVC 2012] [code]
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure

R, t with $||t|| = 1$
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure

Features of Sequential / Incremental SfM:

1. **Feature Detection**
2. **Feature Matching & H/E/F Matrix Fitting**
3. **Two-View Initialization**

\[ R, t \text{ with } \|t\| = 1 \]
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure

\[ R, t \text{ with } ||t|| = 1 \]
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure
Sequential / Incremental SfM

- Feature Detection
- Feature Matching & H/E/F Matrix Fitting
- Two-View Initialization

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure

$R, t$ with $||t|| = 1$
Sequential / Incremental SfM

- Extract relative rotation and translation from H/E/F matrix
- Use 2D-2D matches to **triangulate** 3D structure
Sequential / Incremental SfM

• How to select a good initial pair?
Sequential / Incremental SfM

- How to select a good initial pair?
- Criteria:

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization
Sequential / Incremental SfM

- How to select a good initial pair?
- Criteria:
  - Accurate relative pose \( \approx \) many inlier matches
Sequential / Incremental SfM

• How to select a good initial pair?
• Criteria:
  • Accurate relative pose ≈ many inlier matches
  • Non-planar scene (planar scenes are degeneracy for F-matrix fitting)
Sequential / Incremental SfM

- How to select a good initial pair?
- Criteria:
  - Accurate relative pose ≈ many inlier matches
  - Non-planar scene (planar scenes are degeneracy for F-matrix fitting)
  - Compute both H and E/F matrix
Sequential / Incremental SfM

- **How to select a good initial pair?**

- **Criteria:**
  - Accurate relative pose $\approx$ many inlier matches
  - Non-planar scene (planar scenes are degeneracy for F-matrix fitting)
  - Compute both H and E/F matrix
  - Select pair with large ratio $\#\text{inliers(E/F)} / \#\text{inliers(H)}$
Sequential / Incremental SfM

• How to select a good initial pair?

• Criteria:
  • Accurate relative pose \( \approx \) many inlier matches
  • Non-planar scene (planar scenes are degeneracy for F-matrix fitting)
    • Compute both H and E/F matrix
    • Select pair with large ratio \#inliers(E/F) / \#inliers(H)
  • No pure forward motion (triangulation inaccurate / impossible)
Sequential / Incremental SfM

- How to select a good initial pair?

- Criteria:
  - Accurate relative pose ≈ many inlier matches
  - Non-planar scene (planar scenes are degeneracy for F-matrix fitting)
    - Compute both H and E/F matrix
    - Select pair with large ratio #inliers(E/F) / #inliers(H)
  - No pure forward motion (triangulation inaccurate / impossible)

- In practice, try out multiple initial pairs
Sequential / Incremental SfM

- Pick image(s) with large number of matches to existing cameras
- Obtain 2D-3D matches from 2D-2D matches
- Estimate absolute pose of new image

Diagram:
- Feature Detection
- Feature Matching & H/E/F Matrix Fitting
- Two-View Initialization
- Extend Motion

Torsten Sattler
Sequential / Incremental SfM

- Pick image(s) with large number of matches to existing cameras
- Obtain 2D-3D matches from 2D-2D matches
- Estimate absolute pose of new image
Sequential / Incremental SfM

- Pick image(s) with large number of matches to existing cameras
- Obtain 2D-3D matches from 2D-2D matches
- Estimate absolute pose of new image

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion
Sequential / Incremental SfM

- Pick image(s) with large number of matches to existing cameras
- Obtain 2D-3D matches from 2D-2D matches
- Estimate absolute pose of new image

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Torsten Sattler
Sequential / Incremental SfM

- Associate existing 3D points with new features
- Triangulate new 3D points for features without associated 3D points
Sequential / Incremental SfM

- Associate existing 3D points with new features
- Triangulate new 3D points for features without associated 3D points
Sequential / Incremental SfM

- Associate existing 3D points with new features
- Triangulate new 3D points for features without associated 3D points

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure
Sequential / Incremental SfM

- Associate existing 3D points with new features
- Triangulate new 3D points for features without associated 3D points

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure
Sequential / Incremental SfM

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure

True trajectory
Sequential / Incremental SfM

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure

True trajectory
Estimated trajectory
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure

Torsten Sattler
Sequential / Incremental SfM

- Feature Detection
- Feature Matching & H/E/F Matrix Fitting
- Two-View Initialization
- Extend Motion
- Extend Structure

True trajectory
Estimated trajectory
Sequential / Incremental SfM

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

True trajectory

Estimated trajectory
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure
Sequential / Incremental SfM

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

True trajectory
Estimated trajectory

Torsten Sattler
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure
Sequential / Incremental SfM

True trajectory
Estimated trajectory

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure

Torsten Sattler
Sequential / Incremental SfM

- Errors accumulate, leading to drift over time
- Adjust motion and structure frequently

True trajectory
Estimated trajectory

Drift

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Torsten Sattler
Sequential / Incremental SfM

- Errors accumulate, leading to drift over time
- Adjust motion and structure frequently

Feature Detection
Feature Matching & H/E/F Matrix Fitting
Two-View Initialization
Extend Motion
Extend Structure
Bundle Adjustment
Bundle Adjustment

\[ \rho_j(R_j X_i + t_j) \]

\[ [R_j \mid t_j] \]

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment
Bundle Adjustment

\( \rho_j (\mathbf{R}_j \mathbf{X}_i + \mathbf{t}_j) \)

reprojection error

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment
Bundle Adjustment

$$\rho_j(\mathbf{R}_j \mathbf{X}_i + \mathbf{t}_j)$$

reprojection error

$$\sum_i \sum_j \Delta_{ij} \| \mathbf{x}_{ij} - \rho_j(\mathbf{R}_j \mathbf{X}_i + \mathbf{t}_j) \|^2$$

argmin camera poses, 3D points

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

Torsten Sattler
Bundle Adjustment

\[ \rho_j (R_j X_i + t_j) \]

Point \( i \) visible in image \( j \)?

\[ \sum_i \sum_j \Delta_{ij} \| x_{ij} - \rho_j (R_j X_i + t_j) \|^2 \]

argmin

camera poses, 3D points

reprojection error

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

Torsten Sattler
Bundle Adjustment

\[ \rho_j(R_j X_i + t_j) \]

\[ \sum_i \sum_j \Delta_{ij} \left\| x_{ij} - \rho_j(R_j X_i + t_j) \right\|^2 \]

Can also refine camera intrinsics

Can also refine camera intrinsics

\[ \text{argmin} \]

camera poses, 3D points
Bundle Adjustment

\[ \rho_j(R_j X_i + t_j) \]

reprojection error

\[ \sum_i \sum_j \Delta_{ij} \| x_{ij} - \rho_j(R_j X_i + t_j) \|^2 \]

argmin camera poses, 3D points

Can also refine camera intrinsics

Cost function is highly non-linear → refine from initialization

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

Can also refine camera intrinsics

Torsten Sattler
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \left(\begin{array}{c}
P_i^1 X \\
P_i^3 X \\
P_i^2 X \\
P_i^3 X \end{array}\right),
\]

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \begin{pmatrix} P_i^1 X & P_i^2 X & P_i^3 X \end{pmatrix}, \quad \Delta = \begin{pmatrix} \Delta_1 \\ \vdots \\ \Delta_n \end{pmatrix}
\]

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

$$\min_{X} f(X) = \min_{X} \sum_{i} \Delta_i^T \Delta_i , \quad \Delta_i = x_i - \left( \begin{array}{c} P^i_1 x \\ P^i_2 x \\ P^i_3 x \end{array} \right), \quad \Delta = \left( \begin{array}{c} \Delta_1 \\ \vdots \\ \Delta_n \end{array} \right)$$

Initialization: $X_{k_i} = X_0$

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[ \min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \begin{pmatrix} P_i^1 X \\ P_i^2 X \\ P_i^3 X \end{pmatrix}, \quad \Delta = \begin{pmatrix} \Delta_1 \\ \vdots \\ \Delta_n \end{pmatrix} \]

Initialization: \( X_k = X_0 \)

Compute gradient: \( \nabla f(X_k) = J^T \Delta \)

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \begin{pmatrix}
P^i_1 X \\ P^i_2 X \\ P^i_3 X \\ \end{pmatrix}, \quad \Delta = \begin{pmatrix}
\Delta_1 \\
\vdots \\
\Delta_n 
\end{pmatrix}
\]

Initialization: \(X_k = X_0\)

Compute gradient: \(\nabla f(X_k) = J^T \Delta\)

\[
J = \frac{\partial f(X)}{\partial X} \quad \text{: Jacobian}
\]

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \left( \begin{array}{c} P_1^i X \\ P_2^i X \\ P_3^i X \end{array} \right), \quad \Delta = \left( \begin{array}{c} \Delta_1 \\ \vdots \\ \Delta_n \end{array} \right)
\]

Initialization: \( X_k = X_0 \)

Compute gradient: \( \nabla f(X_k) = J^T \Delta \)

Update: \( X_{k+1} = X_k - \eta \nabla f(X_k) \)

\( J = \frac{\partial f(X)}{\partial X} \) : Jacobian

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \left( \begin{array}{c} P_i^1 X \\ P_i^2 X \\ P_i^3 X \end{array} \right), \quad \Delta = \left( \begin{array}{c} \Delta_1 \\ \vdots \\ \Delta_n \end{array} \right)
\]

Initialization: \( X_k = X_0 \)

Compute gradient: \( \nabla f(X_k) = J^T \Delta \)

Update: \( X_{k+1} = X_k - \eta \nabla f(X_k) \)

\[
J = \frac{\partial f(X)}{\partial X} : \text{Jacobian} \quad \eta : \text{Step size}
\]

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \left(\begin{array}{c} \frac{p_i^1 X}{p_i^2 X} \\ \frac{p_i^3 X}{p_i^4 X} \end{array}\right), \quad \Delta = \left(\begin{array}{c} \Delta_1 \\ \vdots \\ \Delta_n \end{array}\right)
\]

Initialization: \(X_k = X_0\)

Iterate until convergence or for fixed number of iterations

Compute gradient: \(\nabla f(X_k) = J^T \Delta\)

Update: \(X_{k+1} = X_k - \eta \nabla f(X_k)\)

\[
J = \frac{\partial f(X)}{\partial X} : \text{Jacobian} \quad \eta : \text{Step size}
\]

slide credit: Gim Hee Lee
Bundle Adjustment

Gradient descent

\[
\min_X f(X) = \min_X \sum_i \Delta_i^T \Delta_i, \quad \Delta_i = x_i - \left(\frac{P_i^1 X}{P_i^3 X} \right), \quad \Delta = \begin{pmatrix} \Delta_1 \\ \vdots \\ \Delta_n \end{pmatrix}
\]

Initialization: \( X_k = X_0 \)

Iterate until convergence or for fixed number of iterations

Compute gradient: \( \nabla f(X_k) = J^T \Delta \)

Update: \( X_{k+1} = X_k - \eta \nabla f(X_k) \)

\[
J = \frac{\partial f(X)}{\partial X} : \text{Jacobian} \quad \eta : \text{Step size}
\]

Slow convergence near minimum point!

slide credit: Gim Hee Lee
Bundle Adjustment

Newton’s method

2\textsuperscript{nd} order approximation (quadratic Taylor expansion):
\[ f(X + \delta)|_{X=X_k} = f(X) + \nabla f(X)^T \delta + \frac{1}{2} \delta^T H \delta \]

slide credit: Gim Hee Lee
Bundle Adjustment

Newton’s method

2nd order approximation (quadratic Taylor expansion):

\[ f(X + \delta)|_{X=X_k} = f(X) + \nabla f(X)^T \delta + \frac{1}{2} \delta^T H \delta \]

Hessian matrix:

\[ H = \frac{\partial^2 f(X + \delta)}{\partial^2 \delta} |_{X=X_k} \]

slide credit: Gim Hee Lee
Newton’s method

2\textsuperscript{nd} order approximation (quadratic Taylor expansion):

\[ f(X + \delta)|_{X=X_k} = f(X) + \nabla f(X)^T \delta + \frac{1}{2} \delta^T H \delta \]

Hessian matrix:

\[ H = \frac{\partial^2 f(X + \delta)}{\partial^2 \delta} \]

Find \( \delta \) that minimizes \( f(X + \delta)|_{X=X_k} \).
Newton’s method

Differentiate and set to 0 gives:

$$\delta = -H^{-1} \nabla f(X_k)$$

Update:

$$X_{k+1} = X_k + \delta$$
Bundle Adjustment

Newton’s method

Differentiate and set to 0 gives:

$$\delta = -H^{-1} \nabla f(X_k)$$

Update:

$$X_{k+1} = X_k + \delta$$

Computation of $H$ is not trivial (2nd order derivatives) and optimization might get stuck at saddle point!

slide credit: Gim Hee Lee
Bundle Adjustment

Gauss-Newton

Approximate Hessian matrix by dropping 2nd order terms:

\[ H \approx J^T J \]
Bundle Adjustment

Gauss-Newton

Approximate Hessian matrix by dropping 2nd order terms:

\[ H \approx J^T J \]

Solve normal equation:

\[ J^T J \delta = -J^t \Delta \]
Bundle Adjustment

Gauss-Newton

Approximate Hessian matrix by dropping 2nd order terms:

\[ H \approx J^T J \]

Solve normal equation:

\[ J^T J \delta = -J^T \Delta \]

Might get stuck and slow convergence at saddle point!

slide credit: Gim Hee Lee
Bundle Adjustment

Levenberg-Marquardt

Regularized Gauss-Newton with damping factor

\[
(J^T J + \lambda I) \delta = -J^t \Delta
\]
Bundle Adjustment

Levenberg-Marquardt

Regularized Gauss-Newton with damping factor

\[
\left( J^T J + \lambda I \right) \delta = -J^t \Delta
\]

\( \lambda \to 0 \): Gauss-Newton (when convergence is rapid)
\( \lambda \to \infty \): Gradient descent (when convergence is slow)

slide credit: Gim Hee Lee
Bundle Adjustment

Levenberg-Marquardt

Regularized Gauss-Newton with damping factor

\[ J^T J + \lambda I \delta = -J^t \Delta \]

\( \lambda \to 0 \): Gauss-Newton (when convergence is rapid)
\( \lambda \to \infty \): Gradient descent (when convergence is slow)

Adapt \( \lambda \) during optimization:

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

slide credit: Gim Hee Lee
Bundle Adjustment

Levenberg-Marquardt

Regularized Gauss-Newton with damping factor

\[
(J^T J + \lambda I) \delta = -J^t \Delta
\]

\(\lambda \to 0\) : Gauss-Newton (when convergence is rapid)
\(\lambda \to \infty\) : Gradient descent (when convergence is slow)

Adapt \(\lambda\) during optimization:
- Decrease \(\lambda\) when function value decreases

Feature Detection → Feature Matching & H/E/F Matrix Fitting → Two-View Initialization → Extend Motion → Extend Structure → Bundle Adjustment

slide credit: Gim Hee Lee
Bundle Adjustment

Levenberg-Marquardt

Regularized Gauss-Newton with damping factor

\[
(J^T J + \lambda I) \delta = -J^t \Delta
\]

\( \lambda \to 0 \): Gauss-Newton (when convergence is rapid)
\( \lambda \to \infty \): Gradient descent (when convergence is slow)

Adapt \( \lambda \) during optimization:

- Decrease \( \lambda \) when function value decreases
- Increase \( \lambda \) otherwise

slide credit: Gim Hee Lee
Bundle Adjustment

Reconstruction of the old inner city of Aachen, Germany, using the Bundler SfM software

Feature Detection

Feature Matching & H/E/F Matrix Fitting

Two-View Initialization

Extend Motion

Extend Structure

Bundle Adjustment

slide credit: Gim Hee Lee
Bundle Adjustment

- Not covered here:
  - Sparse structure of the bundle adjustment problem
  - Efficient strategies (e.g., Schur Complement Trick)
  - ...

- Recommended reading:
  - Triggs et al., Bundle Adjustment - A Modern Synthesis, 1999

Slide credit: Gim Hee Lee
Multi-View Stereo (MVS)

Input: calibrated images, camera poses, SfM model

Output: dense 3D point cloud or (textured) 3D mesh

model computed using Colmap and Poisson Surface Reconstruction
Multi-View Stereo In a Nutshell

- Use known epipolar relation to find dense matches between images
- Create dense point cloud
Multi-View Stereo In a Nutshell

- Use known epipolar relation to find dense matches between images
- Create dense point cloud
Multi-View Stereo In a Nutshell

- Use known epipolar relation to find dense matches between images
- Create dense point cloud
Multi-View Stereo In a Nutshell

- Use known epipolar relation to find dense matches between images
- Create dense point cloud
3D Reconstruction Packages

- Bundler ([https://github.com/snavely/bundler_sfm](https://github.com/snavely/bundler_sfm))
- Linux (Windows also supported), open source
- SfM pipeline, MVS pipelines can read file format
- Showed nice results on internet photo collections
- Not state-of-the-art anymore
3D Reconstruction Packages

- VisualSFM (http://ccwu.me/vsfm/)
- Linux, Mac OS X, Windows, closed source
- SfM pipeline, interface to external MVS software
- Graphical User Interface
- Very efficient due to use of GPU

https://www.youtube.com/watch?v=5ceiOd8Yx3g
3D Reconstruction Packages

- OpenMVG (https://github.com/openMVG/openMVG)
- Linux, Mac OS X, Windows, open source
- SfM pipeline, MVS pipelines can read file format
- Very modular, functionality not in other packages (full multi-camera support)
- Not very efficient, no GUI
3D Reconstruction Packages

• COLMAP (https://colmap.github.io/index.html)
• Linux, Mac OS X, Windows, open source
• SfM and MVS (NVidia GPU required for MVS)
• Efficient pipeline, GUI
• High code quality, very great tool for research!
3D Reconstruction Packages

- COLMAP (https://colmap.github.io/index.html)
- Linux, Mac OS X, Windows, open source
- SfM and MVS (NVidia GPU required for MVS)
- Efficient pipeline, GUI
- High code quality, very great tool for research!

Demo!
3D Reconstruction Packages

- AliceVision Meshroom (https://alicevision.org/)
- Linux, Windows, open source
- SfM and MVS (NVidia GPU required for both)
- Includes work by Tomas Pajdla and his PhD students
- Have not tried it yet, on my Todo list
3D Reconstruction Packages

- RealityCapture (https://www.capturingreality.com/)
- Commercial software, Windows only
- Start-up (CapturingReality) out of Slovakia, former PhD students at CVUT, recently acquired by Epic Games
- Both SfM and MVS (MVS requires NVidia GPU)
- Highly efficient, SfM takes a few minutes even for large scenes
- Very high quality (probably best software out there)
3D Reconstruction Packages

- RealityCapture (https://www.capturingreality.com/)
- Commercial software, Windows only
- Start-up (CapturingReality) out of Slovakia, former PhD students at CVUT, recently acquired by Epic
- Both SfM and MVS (MVS requires NVidia GPU)
- Highly efficient, SfM takes a few minutes even for large scenes
- Very high quality (probably best software out there)
Results with RealityCapture
Results with RealityCapture
3D Reconstruction Packages

- Many more commercial packages available
- Pix4D ([https://www.pix4d.com/](https://www.pix4d.com/))
Bonus: Neural Radiance Fields (NeRFs)

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

color $c_i$, volume density $\sigma_i$

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

color $c_i$, volume density $\sigma_i$

$$\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i$$

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

\[ \hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

color $c_i$, volume density $\sigma_i$

$\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i$

final color

color at i-th sample

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

\[ \hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i \]

final color

visibility

color at i-th sample

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

\[ \hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \]

\[ T_i = \exp \left( - \sum_{j=1}^{i-1} \sigma_j \delta_j \right) \]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]

Torsten Sattler
Volume Rendering

color $c_i$, volume density $\sigma_i$

\[
\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i
\]

final color

visibility

color at i-th sample

distance to previous sample

$T_i = \exp \left( - \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Rendering

\[
\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i
\]

- Final color
- Visibility
- Color at i-th sample

\[
T_i = \exp \left( - \sum_{j=1}^{i-1} \sigma_j \delta_j \right)
\]

Small if high density before this sample

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]

Torsten Sattler
Volume Rendering

\[ \hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \]

- Color \( \mathbf{c}_i \)
- Volume density \( \sigma_i \)
- Final color
- Visibility
- Color at i-th sample
- Fully differentiable!

\[ T_i = \exp \left( -\sum_{j=1}^{i-1} \sigma_j \delta_j \right) \]

- Distance to previous sample
- Small if high density before this sample

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

\[(x, y, z, \theta, \phi) \rightarrow F_\theta (RGB\sigma)\]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

input: 3D point and ray direction

\[(x, y, z, \theta, \phi) \rightarrow F_{\theta}(RGB\sigma)\]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

input: 3D point and ray direction

\[(x, y, z, \theta, \phi) \rightarrow F_\theta \rightarrow (RGB\sigma)\]

output: color and density

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

input: 3D point and ray direction

\[(x,y,z,\theta,\phi) \rightarrow F_\theta \rightarrow (RGB\sigma)\]  

output: color and density

Continuous scene representation (vs. discrete voxel volumes)

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Sampling

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Sampling

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Sampling

subdivide into equally sized intervals

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Volume Sampling

uniform sampling inside intervals $\rightarrow$ continuous sampling of the volume

subdivide into equally sized intervals

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Hierarchical Volume Sampling

Coarse sampling (coarse network): Uniform sampling in equally-spaced intervals

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Hierarchical Volume Sampling

Coarse sampling (coarse network): Uniform sampling in equally-spaced intervals

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Hierarchical Volume Sampling

Coarse sampling (coarse network): Uniform sampling in equally-spaced intervals

“Fine” sampling (“fine” network): Sample according to observed densities

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Hierarchical Volume Sampling

Coarse sampling (coarse network): Uniform sampling in equally-spaced intervals

“Fine” sampling (“fine” network): Sample according to observed densities

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Hierarchical Volume Sampling

Coarse sampling (coarse network): Uniform sampling in equally-spaced intervals

“Fine” sampling (“fine” network): Sample according to observed densities

All samples are used during volume rendering

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Training

\[ \mathcal{L} = \sum_{r \in \mathcal{R}} \left[ \left\| \hat{C}_c(r) - C(r) \right\|_2^2 + \left\| \hat{C}_f(r) - C(r) \right\|_2^2 \right] \]

[Müller, Evans, Schied, Keller, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, SIGGRAPH 2022]
[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Training

\[ \mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right] \]

[ Müller, Evans, Schied, Keller, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, SIGGRAPH 2022]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Training

\[ \mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right] \]

ground truth color

color predicted by “fine” network

[Müller, Evans, Schied, Keller, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, SIGGRAPH 2022]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Training

\[ \mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \|_2^2 + \| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \|_2^2 \right] \]

- color predicted by coarse network
- color predicted by "fine" network
- ground truth color

[Müller, Evans, Schied, Keller, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, SIGGRAPH 2022]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]

Torsten Sattler
Training

\[ \mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|^2_2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|^2_2 \]  

Trained individually per scene, can now be done in a matter of minutes

[Müller, Evans, Schied, Keller, Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, SIGGRAPH 2022]

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]
Neural Radiance Fields (NeRFs)

Synthetic Scenes

[Mildenhall, Srinivasan, Tancik, et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020]