Toroidal Constraints for Two-Point Localization under High Outlier Ratios

Federico Camposeco\textsuperscript{1}, Torsten Sattler\textsuperscript{1}, Andrea Cohen\textsuperscript{1}, Andreas Geiger\textsuperscript{1,2}, Marc Pollefeys\textsuperscript{1,3}

\textsuperscript{1}ETH Zürich \textsuperscript{2}MPI for Intelligent Systems, Tübingen \textsuperscript{3}Microsoft

Overview

- Image-based localization w.r.t. 3D point clouds is crucial in many applications, e.g., autonomous navigation in AVE\textsuperscript{2}, etc.
- Outlier filtering is critical for large scale localization due to the sheer amount of wrong 3D-2D matches.
- Goal: reduce the number of outliers without assuming any prior.

Insights:

- Image descriptors are not perfectly viewpoint invariant.
- Closest descriptors correspond to closest viewpoints.

- Two matches constraint the camera to lie on a torus. The closest viewpoints can further constrain this pose.
- Proposed solution: use an approximate position from two matches to filter outliers.

Contribution:

- Derivatives of novel constraints for localization.
- First outlier filter that does not require any priors.

Notation

- \( q_0 \): Normalized image keypoint from the database.
- \( \tilde{b}_i \): Normalized image keypoint from the query image.
- \( \theta \): Angular coordinates on \( T^2 \).
- \( C \): Camera position.
- \( \Pi_0 \): Average plane.

\( T^2 \): Surface of the torus.

Toroidal Constraints

- Two 3D-2D matches with angle \( \theta \) define a torus. The camera position \( C \) should be close to \( q_0 \) and \( \tilde{b}_i \). on \( T^2 \).

- We minimize the angular error between the camera position and \( q_0 \) and \( \tilde{b}_i \):

\[
E(q, \theta) = \frac{1}{2} \left( \left( P(q, \theta, \tilde{b}_i) - q_0 \right)^2 + \left( P(q, \theta, \tilde{b}_i) - \tilde{b}_i \right)^2 \right)
\]

- Approximate error by projecting to \( \Pi_0 \):

\[
\tilde{E}(\theta) = \sum_{n} \left( \frac{z_n - z(\theta)}{1 + z_n z(x)} \right)^2 \text{ where } z_n = \frac{z_n}{z_0}, z(x) = \frac{z_0}{P_x}
\]

- Only 4 solutions.

Outlier Filter

- Goal: Decide if a 3D-2D match \( p_i \), \( \theta_i \), \( b_i \) is an outlier, \( i = 1 \ldots n \).

- Proposed solution:

- Given \( p_i \), \( \theta_i \), \( b_i \), compute \( n - 1 \) camera positions:

\[
\text{depths computed from a cluster around zero}
\]

- Why does it work?

- Outlier matches

\[
\text{depressions randomly distributed and large}
\]

\[
\text{depths clustered around zero}
\]

- Score \( p_i \) \( \theta_i \) \( b_i \) according to the population of the clusters.

- Synthetic evaluation

- Approximate solver accuracy

- Precision/Recall against RANSAC

- Real-world evaluation: Dubrovnik 8K dataset

We acknowledge the support of Google’s Tango.