POLITECNICO
MILANO 1863

Optimizing the viewing graph. IACV project

Safuan Iusupov: safuan.iusupov@mail.polimi.it
Seyederfan Eshghollahi: seyederfan.eshghollahi@mail.polimi.it

January 24, 2025

1 Abstract

In Structure-from-Motion (SfM), viewgraphs of unordered image collections often contain redundant edges that can be sparsified for efficiency without compromising reconstruction quality. Additionally, false edges from incorrect image retrieval or structural symmetries can cause ghosting and reconstruction artifacts. We propose a unified approach to simultaneously sparsify the viewgraph and eliminate false edges.

The goal of this project is to provide a critical review of the approach presented in [1], analyze advantages / disadvantages with respect to other methods and propose possible improvements, such as how to extend the approach to manage general graphs not covered by triplets.

2 Introduction

Structure-from-Motion (SfM) reconstructs 3D scenes from images by identifying overlapping image pairs. For large datasets, descriptors are assigned to images and matched, forming a *viewgraph*, where nodes represent cameras (or a single camera in different positions) and edges indicate overlapping views. Keypoints are extracted and matched across images, with relative motion estimated for each edge using epipolar geometry. RANSAC [2] filters outliers, leaving **epipolar inliers** [1], which satisfy the epipolar constraint [2] and ensure reliable matches. The viewgraph, built from these inliers, is processed by incremental or global SfM methods to reconstruct the 3D scene and camera motions.

SfM can be broadly divided into two main steps: **Correspondence Search** and **Incremental Reconstruction**, as illustrated in Figure 1.

1. **Correspondence Search**: In this step, features are extracted and matched between image pairs to identify correspondences. Geometric verification using RANSAC ensures reliable matches, producing a *viewgraph* that represents overlapping views as edges and camera positions as nodes. This viewgraph serves as the input to the next step.
2. **Incremental Reconstruction**: Using the viewgraph, an incremental pipeline is employed to estimate the 3D structure of the scene and camera motions. The pipeline typically involves initialization, image registration, triangulation, outlier filtering, and bundle adjustment to refine both the structure and motion.

The proposed method operates between these two steps by optimizing the viewgraph. Specifically, it sparsifies the viewgraph and removes ambiguous nodes to improve the quality of the reconstruction process. By enhancing the input to the Incremental Reconstruction step, the method ensures better efficiency and accuracy in reconstructing the 3D scene.

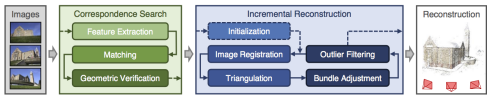


Figure 1: Incremental Structure-from-Motion pipeline.

In the article [1], the authors focus on two types of datasets: **generic** and **ambiguous**. **Generic** datasets arise from unordered image collections where popular views are frequently captured, leading to **redundant** images (nodes) and edges that represent the same scene areas. Conversely, **ambiguous** datasets involve scenes with repeated

structures. These often produce false edges between cameras capturing **similar but distinct** areas, as such images are mistakenly identified as belonging to the same part of the scene. Examples of generic and ambiguous datasets presented in 2 and in 3 accordingly. The number of epipolar inliers are shown for the respective edges. The part of the image where most of the inliers are matched is shown in rectangular boxes with colours corresponding to the edge colours. The red dots on the images show the detected keypoints.

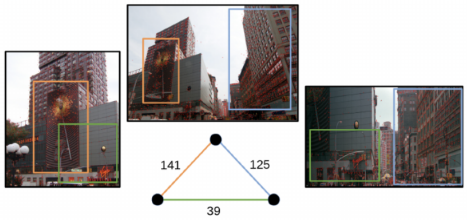


Figure 2: Generic case. *Green* edge correspond to low number of inliers, so this edge may be removed, to sparsify dataset.

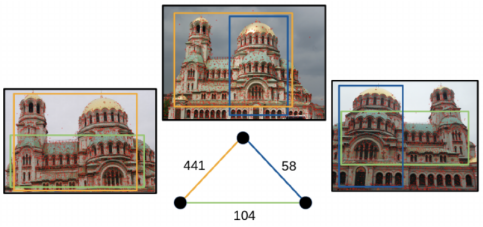


Figure 3: Ambiguous case. *Blue* and *green* edges correspond to the false ones, caused by the corresponding features of two different facades of the same building. The number of inliers for these edges is smaller compared to the true edge, and they can also be removed to deal with ambiguity.

Generic datasets may contain some false edges due to incorrect image retrieval, causing ghost artifacts in reconstructions. Ambiguous datasets, with repeated structures, generate many false edges. While viewgraph sparsification and disambiguation of repeated structures are typically addressed independently, provided article [1] propose a unified approach to tackle both issues.

The analyzed method leverages the informativeness of camera **triplets**, which form the smallest loops in a graph, to identify redundant and false edges. By **scoring edges** based on the 3D point connectivity of triplets across the entire viewgraph, authors remove unreliable edges using a single threshold. This approach eliminates the need for intermediate reconstructions and works seamlessly with both incremental and global SfM pipelines. Experiments demonstrate improved reconstruction times, removal of ghost artifacts in generic datasets, and effective disambiguation in ambiguous datasets.

3 The analyzed method

The key advantage of the proposed method is its ability to handle both types of datasets—generic and ambiguous—using a unified approach with only one hyperparameter, leveraging edge quality analysis to achieve efficient and accurate results.

Edge quality scores: Authors score edges based on the ratio of inliers in each edge to the maximum inliers among the triplet's edges, focusing on relative 3D point connectivity rather than absolute inlier counts. This approach reduces the impact of unmatched keypoints caused by noise, occlusion, or viewpoint changes, as other matched keypoints on the edge compensate for non-repeatable ones. Scores are averaged across all triplets an edge participates in, capturing global graph information. Edges not part of any triplet are removed, as they cannot be scored, with minimal impact on reconstruction due to well-connected viewgraphs.

Triplets:

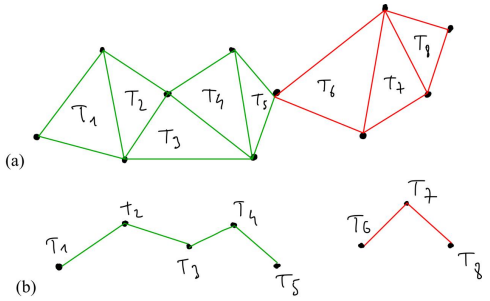


Figure 4: Graph with triplets. a - source graph, where green edges corresponds to first subset of triplets, and the red one to other. b - two set of triplets, where each node represent triplet, in algorithm we will use only G_{LCT}, so scores will be computed only for T1-T2-T3-T4-T5 triplets.

For graphs containing only triplets, two triplets connected by a node but not sharing a common edge form a **joint** (e.g., triplets $T5$ and $T6$ in Figure 4(a)). In such cases, edges in different colors are scored independently, making it challenging to determine if joints are valid, especially in ambiguous datasets.

To address this, authors construct a triplet

graph $G_T = (\mathcal{V}_T, \mathcal{E}_T)$, where nodes represent triplets, and edges connect nodes sharing a common edge in the original graph. Joints in the viewgraph G create multiple connected components in G_T (Figure 4(b)). To ensure global reasoning, we retain only the edges in G corresponding to the **largest connected component** of G_T (G_{LCT}). Empirically, separating joints has minimal impact on reconstructions.

Formalization. Given a quality score q_{ij} for each edge, authors aim to select edges that maximize the average quality score while penalizing edge removal to avoid degenerate solutions. Assume that $s_{ij} \in \{0, 1\}$ denote the binary selection ($s_{ij} = 1$) or removal ($s_{ij} = 0$) of edge. The optimization problem is formulated as:

$$\max_{\substack{s_{ij}\in\{0,1\} \\ (i,j)\in\mathcal{E}}} \frac{\sum_{(i,j)\in\mathcal{E}} s_{ij} q_{ij}}{\sum_{(i,j)\in\mathcal{E}} s_{ij}} - \lambda \frac{\sum_{(i,j)\in\mathcal{E}} (1 - s_{ij}) q_{ij}}{\sum_{(i,j)\in\mathcal{E}} (1 - s_{ij})}, \quad (1)$$

where $\lambda \geq 0$ is a regularization parameter controlling the aggressiveness of edge removal. The second term ensures that removed edges do not have high quality scores, regularizing the selection process. However, the algorithm computes edge scores from several triplets. Let T_{tp} be the set of all triplets in G_{LCT}, and $trp(i, j)$ be the set of triplets containing edge (i, j). Thus, the average score of an edge among all triplets will be $q_{ij} = \frac{\sum_{t\in trp(i,j)} q_{ij}^t}{|trp(i,j)|}$, putting it into Equation (1) we will receive a final optimization problem which can be solved by finding best combination. But the authors provide a simple threshold solution:

$$s_{ij} = \begin{cases} 1, & \text{if } q_{ij} \geq \tau - treshold, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Also they provide theorem, that says that this thresholding is equivalent to optimizing problem.

Theorem 1: *For a given λ in Eq.* (1)*, there exists a threshold τ such that the values of s_{ij} obtained by solving the problems by Eq* (1) *and Eq* (2) *are the same.*

Proof. The left part of Eq. (1) corresponds to the score of the chosen edges, while the right part computes the score of the removed edges. Our goal is to find the best possible combination to maximize the equation. The authors use the fact that, given a set of numbers, removing numbers from the set that are less than their average increases the average of the final set. Maximization of the equation is achieved when the edges with the top k scores are retained while others are removed. This is because it leads to the minimum possible increase in the second term. Here, k depends on the value of λ. This is the same as thresholding the edges based on scores with $sq_{k+1} \leq \tau \leq sq_k$, where sq_z, $z = \{1, 2, \ldots, |\mathcal{E}|\}$, are the sorted scores q_{ij} in descending order. ■

In practice, τ is determined adaptively based on graph connectivity. Let $d_{\max}$ be the maximum node degree, and $|V|$ the number of nodes. The threshold is computed as:

$$\tau = m \left(1 - \frac{d_{\max}}{|V|}\right) + \frac{d_{\max}}{|V|}, \quad (3)$$

where m is the minimum acceptable score. This ensures $m \leq \tau < 1$, avoiding degenerate cases where $\tau = 1$. Choose that formula it can be interpreted as follows:

- Graphs with higher connectivity ($d_{\max}$ closer to $|V|$) have more redundancy, so a higher threshold is appropriate to prune less important edges aggressively.
- In sparse graphs ($d_{\max}$ significantly smaller than $|V|$), a lower threshold is needed to preserve essential edges and ensure that the graph remains connected.

The implementation of this algorithm is also presented on the author's site [3].

Algorithm 1: Edge Selection using Triplets

Input: Viewgraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with geometrically verified edges and a minimum edge score m.
Output: Sparsified viewgraph $\mathcal{G}_F = (\mathcal{V}_F, \mathcal{E}_F)$ with ambiguous edges removed.

Construct triplet graph $\mathcal{G}_T$ from $\mathcal{G}$.
Retain edges in $\mathcal{G}$ participating in largest connected component of $\mathcal{G}_T$, denoting it $\mathcal{G}_{LCT}$.
Find the list of all triplets Trp in $\mathcal{G}_{LCT}$.
for each triplet t in Trp **do**
    Get the number of inliers for all three edges as $n_{ij}, (i, j) \in t$.
    Assign a score to each edge based on triplet t: $q_{ij}^t = \frac{n_{ij}}{\max_{(k,l)\in t} n_{kl}}$.
end
Assign q_{ij} to each edge: $q_{ij} = \frac{\sum_{t\in trp(i,j)} q_{ij}^t}{|trp(i,j)|}$.
Compute τ from m using Eqn. 3.
Remove edges in $\mathcal{G}_{LCT}$ with $q_{ij} < \tau$ giving $\mathcal{G}_F$.
Extract the largest connected component of $\mathcal{G}_F$.

Figure 5: Sequence of analysed algorithm

4 Our trials

Although the authors had access to robust computational resources, including an Intel Xeon Silver 4210 processor, 128 GB of RAM, and two RTX 2080Ti

GPUs, our setup was significantly more constrained. We operated with an Intel Core i7-7700HQ CPU, 16 GB of RAM, and no GPU support. Consequently, generating a database with matched edges was infeasible, as processing even the smallest datasets from article [1], such as *Alamo* or *Madrid Metropolis* [4], required an entire day.

To adapt to these limitations, we utilized pre-matched datasets from COLMAP [5] and worked with small datasets (fewer than 400 images) curated by Carl Olson [6], where the matching process was more manageable. For similar reasons, we did not compute rectification on G_{LCT} and G_{Dopp} — graphs derived after applying Doppelgangers [7] to the original graph G. Furthermore, while the results in article [1] included small ambiguous datasets as one of the test cases, they did not focus on small generic datasets. In contrast, we concentrated our analysis on small, generic datasets to evaluate the algorithm under different conditions.

Result of our experiments presented in Table (1) and in figures (14),(17), (8), (20), (11), with corresponding notations:

- **RT** G: Reconstruction time of the original viewgraph G.
- **RT** G_f: Reconstruction time of the graph obtained after Algorithm 1 (5).
- **MRE** G: Mean Reconstruction Error (in pixels) of the original viewgraph G.
- **MRE** G_f: Mean Reconstruction Error (in pixels) of the graph obtained after Algorithm 1 (5).
- **3D points** G: Number of points reconstructed from the original viewgraph G.
- **3D points** G_f: Number of points reconstructed from the viewgraph G_f.
- N **images**: Number of images in the dataset.

For all experiments we used minimum edge score equal to **0.7**.

To ensure reproducibility, we provide the algorithm [1] used to obtain the results, along with archives containing the COLMAP dataset formats for both G and G_f. However, for optimal results, all experiments should be conducted in an isolated environment with no concurrent running programs. Due to the constraints of our setup, we were unable to adhere to these conditions, and as a result, the outcomes may differ slightly from those reported in this document. **Instructions for running the process:**

1. Install COLMAP [8].
2. Download datasets from [5].
3. Open the database file extracted from COLMAP's feature extractor.
4. Use SQLite Browser to export the `inlier_matches` table to a text file.
5. Apply the provided MATLAB code [3] to the text file to generate an `output.txt`.
6. Open the `output.txt` in SQLite Browser.
7. Use the following SQL query to remove rows from the `inlier_matches` table where the match count is zero:

```
DELETE FROM inlier_matches
WHERE pair_id IN (SELECT * FROM output);
```

8. Attempt reconstruction again with COLMAP.

5 Analysis of Results

The following is an analysis of the results presented in Table 1 and based on reconstruction time (RT), mean reconstruction error (MRE), and the number of 3D points reconstructed.

5.1 Reconstruction Time (RT)

- **Reduction in Reconstruction Time:**

  For most datasets, there is a clear reduction in reconstruction time when using G_f compared to the original G, which aligns with the objective of sparsification—improving reconstruction efficiency.

  For example, the reduction is particularly significant for datasets like *Alcatrz* (RT of 6.57 vs. 4.33) and *Person Hall* (RT of 16.27 vs. 13.45). The differences in time are consistent with the expected outcomes of using a sparsified graph with fewer edges.

- **Minor or No Impact in Some Cases:** In datasets like *Eglise* and *Gerrard Hall*, the difference in reconstruction time is minimal (e.g., 4.35 vs. 4.35 for *Eglise*), indicating that in some cases, sparsification does not significantly affect the runtime, possibly because the original graph G is already quite sparse.

5.2 Mean Reconstruction Error (MRE)

- **Small or Improved Errors with G_f:** In many cases, the **MRE** decreases or remains the same when using G_f compared to G, such as for *South Building* (MRE $G = 0.510$ vs. 0.504 for G_f), *Gerrard Hall* (0.639 vs. 0.639), and *UrbanII* (0.3910 vs. 0.3867). This suggests that sparsification does not negatively affect the accuracy of the reconstructions.

  For some datasets, the **MRE** for G_f is slightly lower than for G, which indicates that the removal of edges might have led to more consistent camera parameters and 3D points, reducing errors. For example, *Gustav_wolf* shows a noticeable improvement in MRE (0.3589 vs. 0.3528).

- **Stable Accuracy Across Datasets:** Many of the datasets show very stable **MRE** results for both G and G_f, indicating that the sparsification process does not cause significant degradation in reconstruction quality.

5.3 Number of 3D Points

- **Minimal Impact on the Number of 3D Points:** The number of 3D points reconstructed generally remains high for both G and G_f, though in some cases, there is a small decrease in 3D points after sparsification. For example, *South Building* has a decrease in the number of 3D points (61338 vs. 50222), and *Person Hall* also shows a slight drop (141865 vs. 131369).

  This decrease in 3D points is expected, as sparsifying the graph removes some edges, which can result in fewer observations. However, as observed in our results, the remaining 3D points often exhibit lower MRE, indicating that the remaining points are more consistent.

6 Advantages and Disadvantages of analyzed method

6.1 Advantages

- The method can handle both generic and ambiguous datasets, unlike other approaches that focus solely on one of these tasks (sparsification [9, 10, 11] and disambiguation [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 7, 25, 26]).
- This method requires only a single hyperparameter, whereas other algorithms demand multiple parameters to be selected [27, 28, 20, 15, 24, 18, 22].
- Unlike other methods [11, 9, 10, 23, 14], this method does not rely on intermediate partial reconstruction.
- The method is adaptable to any pipeline, a feature not always present in other approaches [11, 9, 10, 24, 12].
- It does not require specific probability distributions for missing correspondences, unlike some methods [12, 13].

6.2 Disadvantages

- Despite requiring only one hyperparameter, optimization is still needed, as the optimal graph is often found within a specific range, that is why authors have selected different values of $m = \{0.3 - 0.9\}$.
- Relying solely on scores from G_{LCT} may neglect significant edges, which can lead to a graph with a similar number of triplets but a loss of essential information. This issue is especially prominent in small datasets [22], as noted by the authors of [3].

7 Conclusions

Our experiments confirmed that the algorithm from article [1] is effective for small datasets. While designed for larger datasets, it successfully reduced reconstruction times without sacrificing accuracy, even for datasets with fewer than 400 images, demonstrating its versatility, especially in resource-constrained environments.

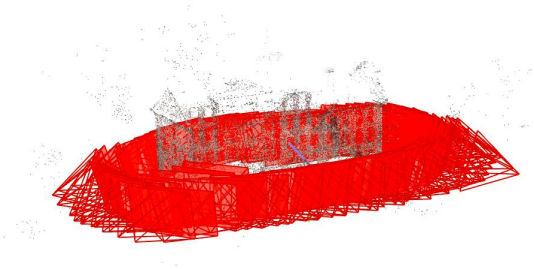
Overall, both our results and those in [1] show that the algorithm offers a strong balance between efficiency and accuracy across various dataset types (general/ambiguous) and sizes (small/medium/large). It reduces reconstruction time while maintaining 3D point count and improving mean reconstruction error (MRE), enhancing the Structure-from-Motion (SfM) process.

In conclusion, the method is a promising tool for improving SfM efficiency and accuracy, applicable to datasets of all sizes and complexities.

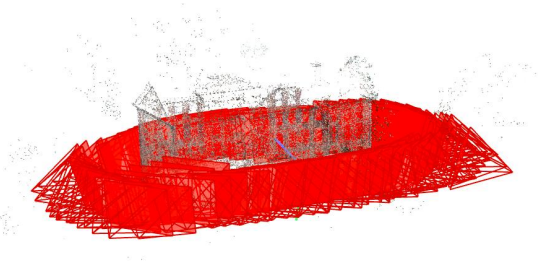
;

Table 1: Experiments

Data	**RT** G	**RT** G_f	**MRE** G	**MRE** G_f	**#3D points** G	**#3D points** G_f	$\#N$ **images**
South Building [5]	3.25	**2.53**	0.510	**0.504**	61338	50222	128
Person Hall [5]	16.27	**13.45**	0.677	**0.676**	141865	131369	330
Gerrard Hall [5]	3.25	**3.22**	**0.639**	**0.639**	42819	42515	100
Water Tower [6]	18.20	**17.37**	0.4283	**0.4279**	98831	98470	173
Alcatraz [6]	6.57	**4.33**	0.4085	**0.4077**	77713	77305	132
UWO [6]	1.40	**1.24**	0.4210	**0.4100**	26625	24607	57
Gustav_wolf [6]	2.08	**1.57**	0.3589	**0.3528**	56850	50595	57
Pumpkin [6]	11.14	**10.18**	**0.6124**	0.6129	101467	100208	209
UrbanII [6]	5.06	**3.29**	0.3910	**0.3867**	80408	78801	90
Eglise [6]	**4.35**	**4.35**	0.4264	**0.4263**	52702	50640	85
LUsphinx [6]	1.51	**1.42**	0.4619	**0.4584**	42999	41485	70
Round_church [6]	**3.18**	3.21	**0.5161**	0.5169	44650	43968	92
Porta san Donato [6]	**9.33**	9.34	0.4583	**0.4570**	81839	80839	141
King's College, Toronto [6]	4.24	**4.20**	**0.4254**	0.4254	53822	52202	77
Sri_Thendayuthapani [6]	5.29	**4.47**	0.5936	**0.5862**	46938	45051	98

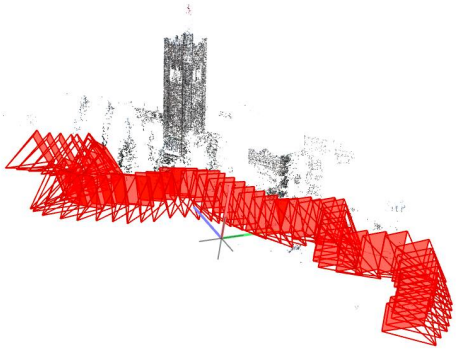


Original graph G: **RT** = 3.25, **MRE** = 0.510

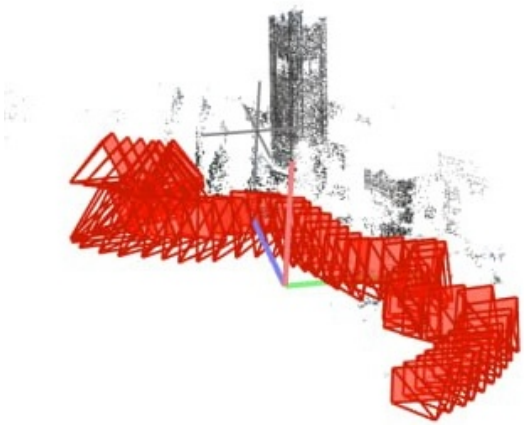


Sparsified Graph G_f: **RT** = 2.53, **MRE** = 0.504

Figure 8: South Building [5]

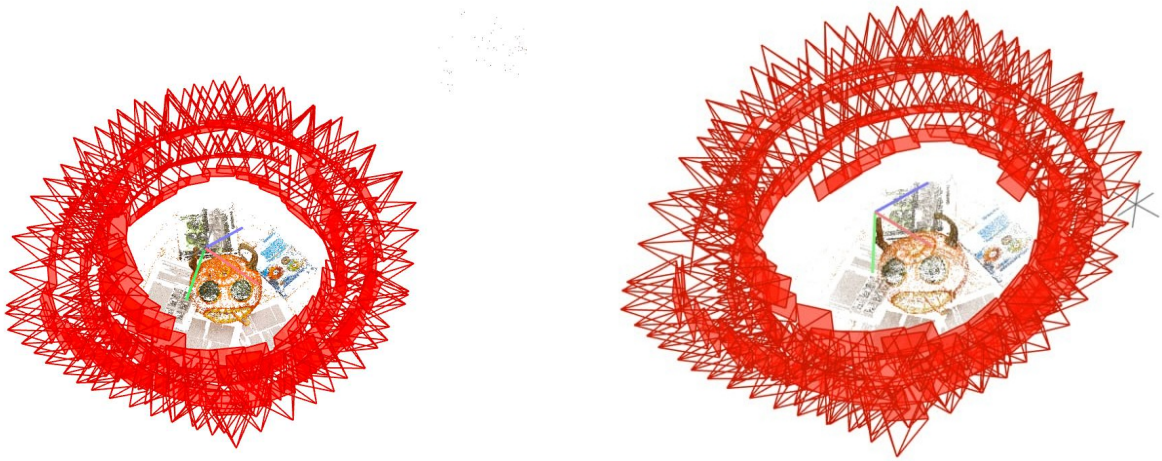


Original graph G: **RT** = 1.40, **MRE** = 0.4210



Sparsified Graph G_f: **RT** = 1.24, **MRE** = 0.41

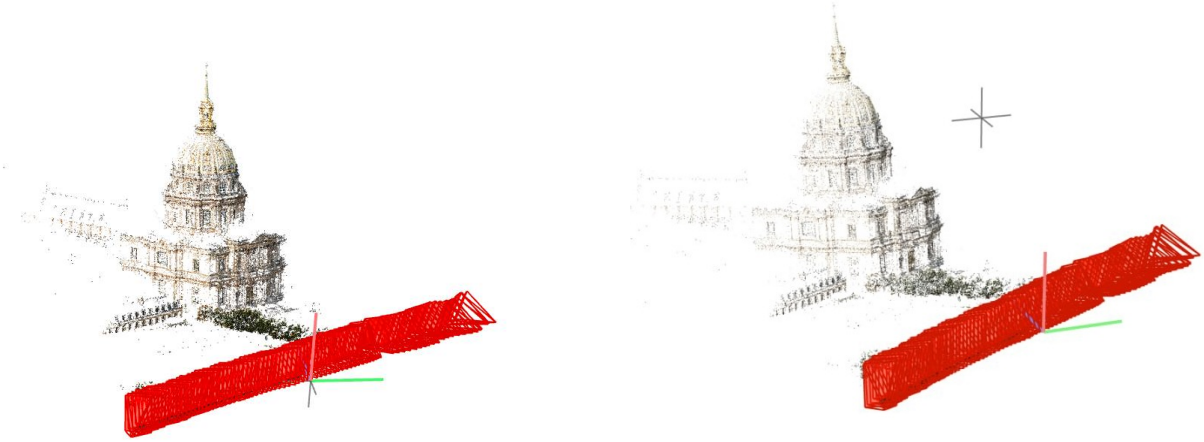
Figure 11: UWO [6]



Original graph G: **RT** = 11.44, **MRE** = 0.6124

Sparsified Graph G_f: **RT** = 10.18, **MRE** = 0.6129

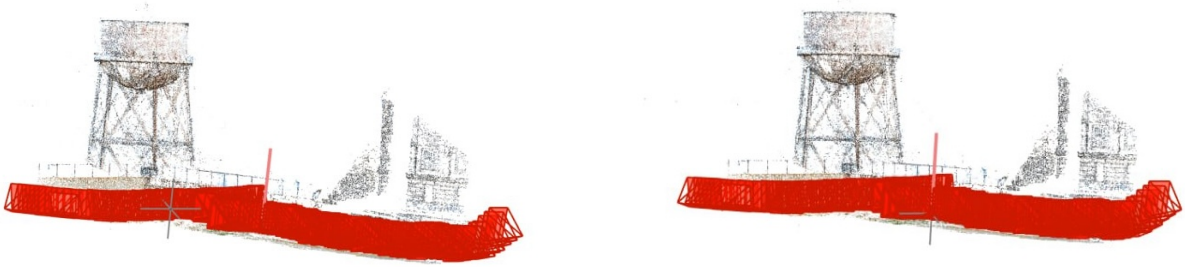
Figure 14: Pumpkin [6]



Original graph G: **RT** = 4.35, **MRE** = 0.4264

Sparsified Graph G_f: **RT** = 4.35, **MRE** = 0.4263

Figure 17: Eglise [6]



Original graph G: **RT** = 18.20, **MRE** = 0.4283

Sparsified Graph G_f: **RT** = 17.37, **MRE** = 0.4279

Figure 20: Water Tower [6]

References

[1] Lalit Manam and Venu Madhav Govindu. "Leveraging Camera Triplets for Efficient and Accurate Structure-from-Motion". In: (June 2024), pp. 4959–4968. URL: https://ee.iisc.ac.in/cvlab/research/camtripsfm/cam_triplets_sfm.pdf.

[2] Krystian Mikolajczyk and Cordelia Schmid. *Multiple View Geometry in Computer Vision Second Edition*. Cambridge University Press, 2003.

[3] Lalit Manam and Venu Madhav Govindu. *Leveraging Camera Triplets for Efficient and Accurate Structure-from-Motion*. 2024. URL: https://ee.iisc.ac.in/cvlab/research/camtripsfm/ (visited on 01/21/2025).

[4] Kyle Wilson and Noah Snavely. "Robust global translations with 1dsfm". In: *European Conference on Computer Vision, pages 61–75* (2014).

[5] Johannes Schönberger. *COLMAP Datasets*. 2024. URL: https://demuc.de/colmap/datasets/ (visited on 01/21/2025).

[6] Lund University Carl Olsson. Mathematical Imaging Group at the Centre for Mathematical Sciences Lund Institute of Technology. *Sdm datasets*. 2024. URL: https://www.maths.lth.se/matematiklth/personal/calle/dataset/dataset.html (visited on 01/23/2025).

[7] Qianqian Wang Hadar Averbuch- Elor Bharath Hariharan Ruojin Cai Joseph Tung and Noah Snavely. "Doppelgangers: Learning to disambiguate images of similar structures". In: *In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 34–44* (2023).

[8] Johannes L. Schoenberger. *COLMAP*. 2024. URL: https://colmap.github.io/index.html (visited on 01/21/2025).

[9] Akihiko Torii Michal Havlena and Toma s̆ Pajdla. "Ef- ficient structure from motion by graph optimization". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11* (2010).

[10] Steven M Seitz Noah Snavely and Richard Szeliski. "Skele- tal graphs for efficient structure from motion." In: *IEEE Conference on Computer Vision and Pattern Recognition*. (2008). DOI: 10.1109/CVPR.2008.4587678.

[11] Toma s̆ Pajdla Michal Havlena Akihiko Torii and Jan Knopp. "Randomized Structure from Motion Based on Atomic 3D Models from Camera Triplets". In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[12] ArnoldIrschara ChristopherZach and HorstBischof. "What can missing correspondences tell us about 3d structure and motion". In: *IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008* (2008).

[13] Manfred Klopschitz Christopher Zach and Marc Pollefeys. "Disambiguating visual relations using loop constraints". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1426–1433* (2010).

[14] Richard Szeliski Richard Roberts Sudipta N Sinha and Drew Steedly. "Structure from motion for scenes with large duplicate structures". In: *CVPR 2011, pages 3137–3144* (2011).

[15] Ping Tan Nianjuan Jiang and Loong-Fah Cheong. "Seeing double without confusion: Structure-from-motion in highly ambiguous scenes". In: *IEEE Conference on Computer Vision and Pattern Recognition, pages 1458–1465* (2012).

[16] Geoffrey J. McLachlan and Thriyambakam Krishnan. "The EM algorithm and extensions". In: *Wiley, 2. ed edition* (2008).

[17] Sudipta N Sinha Andrea Cohen Christopher Zach and Marc Pollefeys. "Discovering and exploiting 3d symme- tries in structure from motion". In: *IEEE Conference on Computer Vision and Pattern Recognition, pages 1514–1521* (2012).

[18] Kyle Wilson and Noah Snavely. "Network principles for sfm: Disambiguating repeated structures with local con- text". In: *In Proceedings of the IEEE International Conference on Computer Vision, pages 513–520* (2013).

[19] Enrique Dunn Jared Heinly and Jan-Michael Frahm. "Correcting for duplicate scene structure in sparse 3d reconstruction". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, pages 780–795* (2014).

[20] Enrique Dunn Jared Heinly and Jan-Michael Frahm. "Recovering correct reconstructions from indistinguishable geometry". In: *2014 2nd International Conference on 3D Vision, pages 377–384. IEEE* (2014).

[21] Yogesh Agrawal Fan Wang Aditi Nayak and Roy Shilkrot. "Hierachical image link selection scheme for duplicate struc- ture disambiguation". In: *BMVC, page 221* (2018).

[22] Ling Zhang Qingan Yan Long Yang and Chunxia Xiao. "Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context". In: *Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3836–3844* (2017).

[23] Zhaopeng Cui and Ping Tan. "Global structure-from-motion by similarity averaging". In: *Proceedings of the IEEE International Conference on Computer Vision, pages 864– 872* (2015).

[24] Joseph DeGol Rajbir Kataria and Derek Hoiem. "Improving structure from motion with reliable resectioning". In: *international conference on 3D vision (3DV), pages 41–50* (2020).

[25] Josef Sivic Jan Knopp and Tomas Pajdla. "Avoiding confusing features in place recognition". In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11, pages 748–761* (2010).

[26] Tomas Pajdla Akihiko Torii Josef Sivic and Masatoshi Okutomi. "Visual place recognition with repetitive structures". In: *Proceedings of the IEEE conference on computer vision and pattern recognition, pages 883–890* (2013).

[27] Visesh Chari Rajvi Shah and PJ Narayanan. "View-graph selection framework for sfm". In: *Proceedings of the European Conference on Computer Vision (ECCV) pages 535–550* (2018).

[28] Tian Fang Runze Zhang Tianwei Shen Siyu Zhu and Long Quan. "Graph-based consistent matching for structure-from- motion". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, pages 139–155* (2016).