1. Background

This report details the technical evolution of our solution strategy for the Image Matching Challenge 2025. Three distinct approaches were implemented and evaluated:

- Approach 1 (RoMa): A dense matching strategy that failed due to computational timeouts.
- Approach 2 (SIFT + DeDoDe): A hybrid strategy that failed due to feature dimension mismatches and fallback to inferior handcrafted features.
- Approach 3 (ALIKED + LightGlue): The successful "Sparse Deep Feature" strategy that balanced accuracy and speed.

Through iterative hyperparameter tuning, the score of Approach 3 improved significantly from 17 to 33.44, establishing it as the final submission.

2. Approach 1 - RoMa (Dense Matching)

Why this approach?

RoMa (Robust Dense Feature Matching) can handle extreme viewpoint changes and texture-less regions by operating at the pixel level, aiming for peak accuracy.

How it works:

1. Load DINOv2 (ViT-L/14) and RoMa (outdoor) with use_custom_corr=False.
2. Pairing: exhaustive for <300 images; sliding window (next 20) for larger sets.
3. Convert RoMa correspondences into quantized keypoints; write HDF5; reconstruct via pycolmap incremental mapping.

Why it failed (Notebook Timeout Error):

- Dense matching scales with resolution ($O(H \times W)$) making per-pair processing slow.
- Exhaustive fallback is still too heavy with dense features.
- Complex setup increased fragility and overhead.

Key hyperparameters:

- Sliding window size: 20 (For large datasets).
- Max sampled matches per pair: max_keypoints=5000.
- Matching threshold: match_threshold=0.2.
- Reconstruction: pycolmap incremental mapping; integer keypoint quantization.

3. Approach 2 - DeDoDe + DINOv2 clustering (Fallback to SIFT)

Why this approach?

Combine modern deep local features (DeDoDe) with scene-aware clustering (DINOv2 + DBSCAN) to split disjoint scenes before reconstruction.

How it works:

1. Patch hloc/extractors/dedode.py (padding to multiples of 14; API alignment).
2. Extract DINOv2 embeddings; cluster via DBSCAN (eps=0.18, min_samples=3, cosine metric).
3. Sanitize filenames (img_0000.jpg); clean-rewrite HDF5 keys; intended DeDoDe+LightGlue, but fell back to SIFT + nearest-neighbor mutual.

Why it failed:

- Feature dimension mismatch: DeDoDe outputs 256-d descriptors; available LightGlue weights expected 128-d. Offline limits prevented fetching correct weights.
- Fallback to SIFT reduced matching quality, especially under wide baselines and low texture.

Key hyperparameters:

- Clustering: DBSCAN eps=0.18, min_samples=3 (cosine on normalized embeddings).
- SIFT: resize_max=1600, first_octave=-1, peak_threshold=0.01, grayscale=true.
- Matching: nearest-neighbor mutual; no explicit distance threshold.
- Reconstruction: pycolmap.IncrementalPipelineOptions (min matches ≈ 15; init inliers ≈ 25).

4. Approach 3 - ALIKED + LightGlue + DINOv2 Shortlisting (Successful Approach)

Why this approach?

Balances speed and robustness: sparse deep features (fast like SIFT; robust like dense methods) + a learnable matcher (LightGlue). DINOv2 shortlisting constrains runtime.

How it works:

1. Shortlisting (DINOv2): get_image_pairs_shortlist(sim_th=0.4, min_pairs=25, exhaustive_if_less=20).
2. Feature Detection (ALIKED): max_num_keypoints=6000, detection_threshold=0.01, resize=1600.
3. Feature Matching (LightGlue, Kornia 0.7.2): KF.LightGlueMatcher('aliked', mp=True); require min_matches>=20.
4. Geometric Verification & Reconstruction: import into COLMAP; pycolmap.match_exhaustive (RANSAC); pycolmap.incremental_mapping with max_num_models=25.
5. Submission: 2025 spec dataset,scene,image,rotation_matrix,translation_vector; outliers to nan placeholders.

Why it succeeded:

- Runtime control via DINOv2
- Robust matches via LightGlue vs NN mutual.
- Sufficient keypoint density
- Operational safeguards

5. Hyperparameter Tuning - Score 17 to 33.44

Phase 1: Initial Optimization (17 to 29.5)

Key adjustments focused on increasing feature density and graph connectivity:

- ALIKED Features: num_features was increased to ensure better coverage in low-texture areas. The detection_threshold was also lowered, significantly increasing keypoint sensitivity.
- DINOv2 Shortlisting: The similarity threshold (sim_th) was also adjusted to capture more valid candidate pairs. Simultaneously, min_pairs was increased to improve graph connectivity and reduce fragmentation.
- LightGlue Matching: The min_matches threshold was raised to filter out weaker pairs and improve geometric consistency.

Phase 2: Refined Tuning (29.5 to 31.44 to 33.44)

Subsequent iterations involved fine-tuning these parameters further to maximize the trade-off between match quantity and quality. Through careful recalibration of matching thresholds and reconstruction constraints, the score was successfully improved first to 31.44 and finally to 33.44.

6. Conclusions

- Adapted Approach 3 under offline constraints.
- Maintain DINOv2 shortlisting and the finalized ALIKED+LightGlue settings that achieved the 33.44 benchmark.
- Keep strict CSV validation and cluster handling to avoid submission errors.

Final pick:

Approach 3 (ALIKED + LightGlue + DINOv2 shortlist), tuned to achieve score 33.44.