Master Semester Project
Fall 2024

Enhancing 3D Reconstruction with Thermal Imaging

Intelligent Maintenance and Operations Systems - EPFL

Jean Cordonnier

Professor: Olga Fink
Supervisor: Chenghao Xu

Ecole Polytechnique Fédérale de Lausanne

Contents

1 Introduction

This project aims to address the challenges posed by multi-modal NeRF 3D reconstruction, with a focus on the initial image pose estimation process. To achieve this, a FLIR dataset containing RGB and thermal images was pre-processed to form a new dataset named ThermalScene. Then, using the ThermoScene dataset with the ThermoNeRF model [2], the challenge of NeRF reconstruction with a limited number of images was highlighted. Finally, the Mast3r [3] image matching method was compared with the COLMAP approach using ThermoNeRF to evaluate the results, including a sliding window method to enable larger dataset pose estimation, overcoming the initial Mast3r memory limitation. The results show that the proposed method effectively extends the Mast3r coverage of the scene and compare the results with COLMAP globally initialized poses. Finally, the project critically analyzes the results and proposes future improvements to enhance 3D Thermal reconstruction using the proposed pipeline.

2 Theoretical Background

2.1 COLMAP for Structure-from-Motion

COLMAP [5] [6] is an open-source, state-of-the-art tool utilizing the Incremental Structure-From-Motion (SfM) process for 3D reconstruction. It is widely used as it represents one of the best SfM tools in the state of the art, is very well-documented, compatible with large datasets, and is open-source.

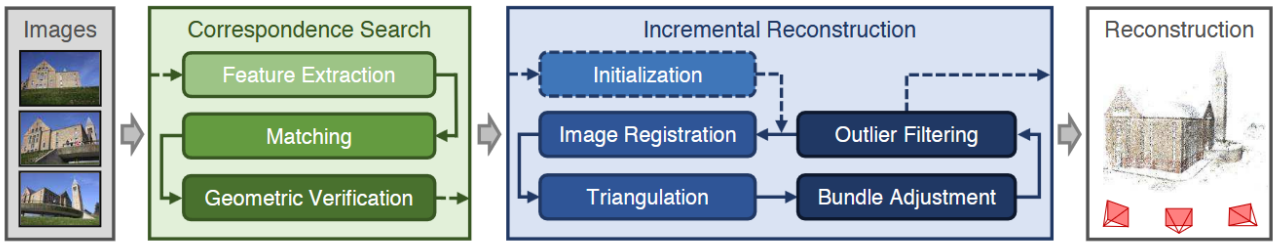


Figure 1: Typical incremental SfM reconstruction pipeline from a set of unposed images as presented in [5].

The method relies on image pairs created during the *Matching* and *Geometric Verification* steps, as shown in Figure 1. In order for two images to be paired together, they must share a large number of features (i.e., they look similar) that are geometrically verified (i.e., they indeed show a similar part of the scene). The conditions for two images to be paired are strict since a large number of requirements are needed. For large datasets, the percentage of images registered can vary from a small fraction to the entire scene, depending on the scene. Scenes with insufficient overlapping images or with low-quality features may lead to COLMAP struggling to register a high number of images.

2.2 Mast3r and Dust3r

Dust3r [8] is a novel method that utilizes transformers to create pointmaps, mapping each pixel of a pair of images to 3D coordinates. The Dust3r implementation also proposes a global alignment of every image pair, allowing for a global reconstruction of the scene. Mast3r [3] is built on top of Dust3r with the aim of enhancing matching capabilities through various adjustments (mainly with

the addition of local features output from their network). The two methods remain very similar, but Mast3r achieves overall better performances. It will be the model used throughout this project.

The two methods utilize the pointmap output (and the feature map for Mast3r) from the network to create a large amount of crucial information, such as 3D reconstruction, pixel correspondences, camera calibration, and camera pose estimation. The images pairs are then combined in a single, globally optimized, model using global alignement techniques. This novel technique achieves impressive results for challenging scenes and outperforms every other method it was compared to, including COLMAP.

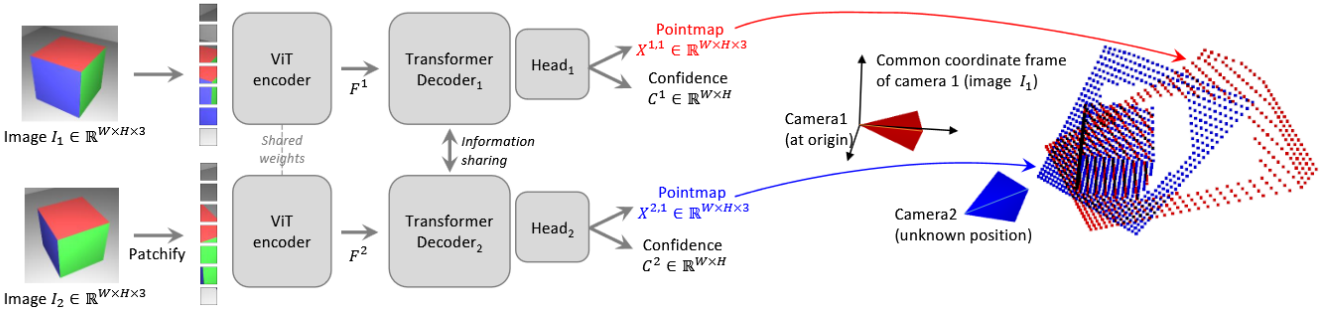


Figure 2: Dust3r pipeline from [8].

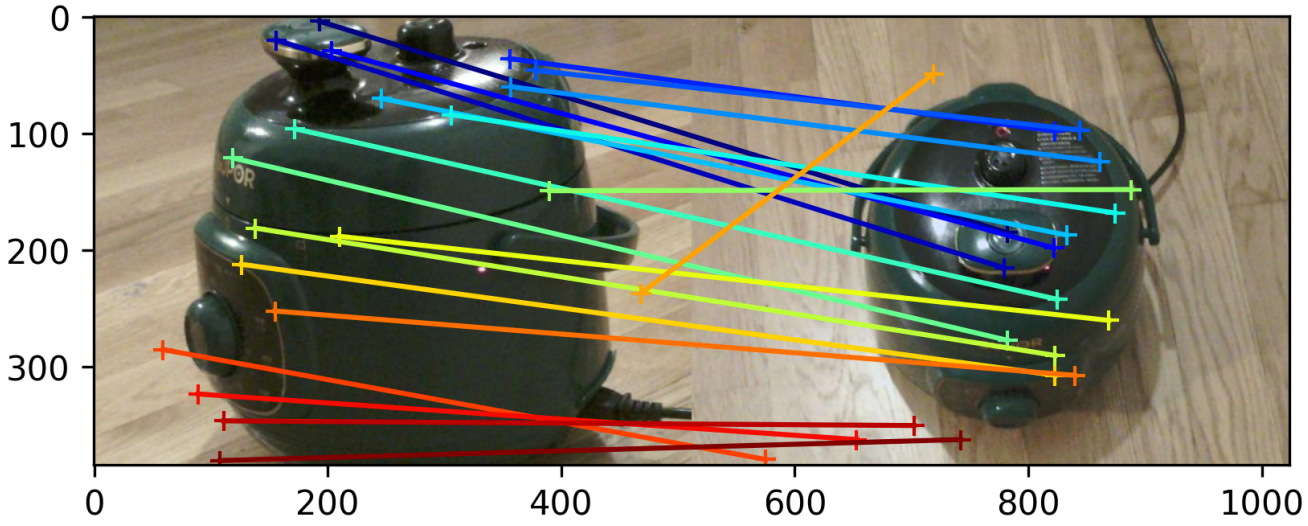


Figure 3: Example of pixel correspondence for 20 points using Mast3r on an image pair from the *Rice Cooker* scene.

2.3 NeRF and ThermoNeRF

Neural Radiance Fields (NeRF) [4] are a novel technique that utilizes neural networks to create novel view synthesis. This method yields extremely good results, achieving photo-realistic rendering.

The initial pipeline proposed in [4] works by creating rays (with known orientation and pose) for each pixel of the input images. Then, the loss between the output of the network and the true color and volume density (from the image) is minimized. Once training is complete, it is possible to input novel views as a set of new rays with a chosen position (X, Y, Z) and orientation (θ, ϕ).

For the method to work, it is extremely important for each ray's position to be very precisely known. This means that the camera poses and parameters need to be carefully computed before

reconstruction can begin. In the vast majority of cases, such data is not available, and these key factors are estimated using different tools (as described in Section 2.1 and Section 2.2).

Since the original paper [4] was published in 2021, the method has drawn significant attention, and many improvements have been proposed. One of these improvements is the augmentation of models with thermal rendering.

ThermoNeRF [2] is one such improvement. It effectively uses paired RGB and thermal images to overcome the challenges posed by thermal reconstruction. The benefit of pairing the images is that it allows the camera poses for the RGB images to be determined and the same poses to be applied to the equivalent thermal images. ThermoNeRF is built upon a model called Nerfacto, developed in Nerfstudio [7], which exploits the characteristics of different NeRF models into a single robust model. ThermoNeRF successfully creates high-quality RGB and thermal 3D scene reconstructions and will be used throughout this project.

2.4 The Absolute Orientation Problem

The absolute orientation problem involves determining the optimal transformation that aligns one point set to another, often using rotation, scaling, and translation. This problem and the methods proposed to tackle it are part of the Procrustes method family.

The Umeyama algorithm tackles the absolute orientation problem by minimizing the RMS error between the two sets of points. It requires knowledge of the point correspondences between the two sets. The goal is to find R (rotation), t (translation), and s (scaling) such that the following cost is minimized:

$$\mathcal{L}(\mathbf{R}, \mathbf{t}, s) = \sum_{i=1}^{n} \|s\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2 \tag{1}$$

with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$, the two sets of points.

The Umeyama solution to the absolute orientation problem effectively provides a solution with reasonable complexity (thanks to the SVD decomposition used). However, it is still very sensitive to outliers. For these reasons, it was used in this report to tackle camera alignment.

3 Design and Motivation

3.1 The Practical Limitations of Colmap

While Colmap excels in the reconstruction and pose estimation of scenes with many overlapping features and a large number of images, it can encounter difficulties when working with datasets where these criteria are not met.

The dataset described in Table 1 and downloadable here will be referred to as Thermalscenes in this report. As shown in Table 1, this dataset is extremely challenging for Colmap due to the sparsity of the images and the lack of prominent features in each scene.

Scene	Number of Images	Images placed by colmap
Building	13	2
Car	20	5
Kettle	26	2
Mug	24	2
Rice Cooker	25	2

Table 1: ThermalScene composition. Each image count refer to a RGB and thermal image covering the exact same field. The images where processed using the default COLMAP mapper parameters with the RGB as an input.

Scene Name	Number of Subscenes	Percentage of Images Placed [%]
Building	1	15.38
Building A Spring	7	25.74
Building A Winter	5	62.42
Car	1	15.00
Double Robot	5	26.35
Exhibition Building	8	45.41
Freezing Ice Cup	9	68.99
Heater Water Cup	6	70.86
Heater Water Kettle	5	100.00
Kettle	1	8.00
Melting Ice Cup	5	100.00
Mug	1	8.33
Raspberrypi	7	73.30
Rice Cooker	1	8.00
Trees	5	60.14

Table 2: Percentage of images placed for each scene with the default colmap mapper. The scenes from the ThermoScenes dataset are written in **bold**. Each subscene was created using the methodology described in Figure 5.

In contrast to Colmap, Mast3r can effectively place each of the images for each subscene described in Table 2, surpassing Colmap for these specific conditions.

3.2 The Mast3r RAM Memory Limitation

To overcome the challenges posed by the dataset, Mast3r was applied to initialize the pose and the camera parameters for each image. Due to the use of a neural network, this method is limited by the extensive memory usage required to process a simultaneous number of image pairs. A 32GB RAM NVIDIA Tesla V100 (Volta architecture) would encounter an *OutOfMemory* error with around 32 RGB images. This limitation makes the current Mast3r implementation unsuitable for many applications requiring a large number of images for a single GPU.

To overcome this limitation, the Umeyama algorithm, enhanced with outlier detection, can be applied using a sliding window on a set of reference images. This method allows multiple sets of 30 images from the same scene to be combined elegantly. The structure for this approach is developed in Section 4.

3.3 NeRF and the Number of Images for Image Reconstruction

Since this report focuses on challenging datasets (i.e., those with a limited number of images), it is interesting to note how the model used for evaluation, ThermoNerf, behaves with limited images as input.

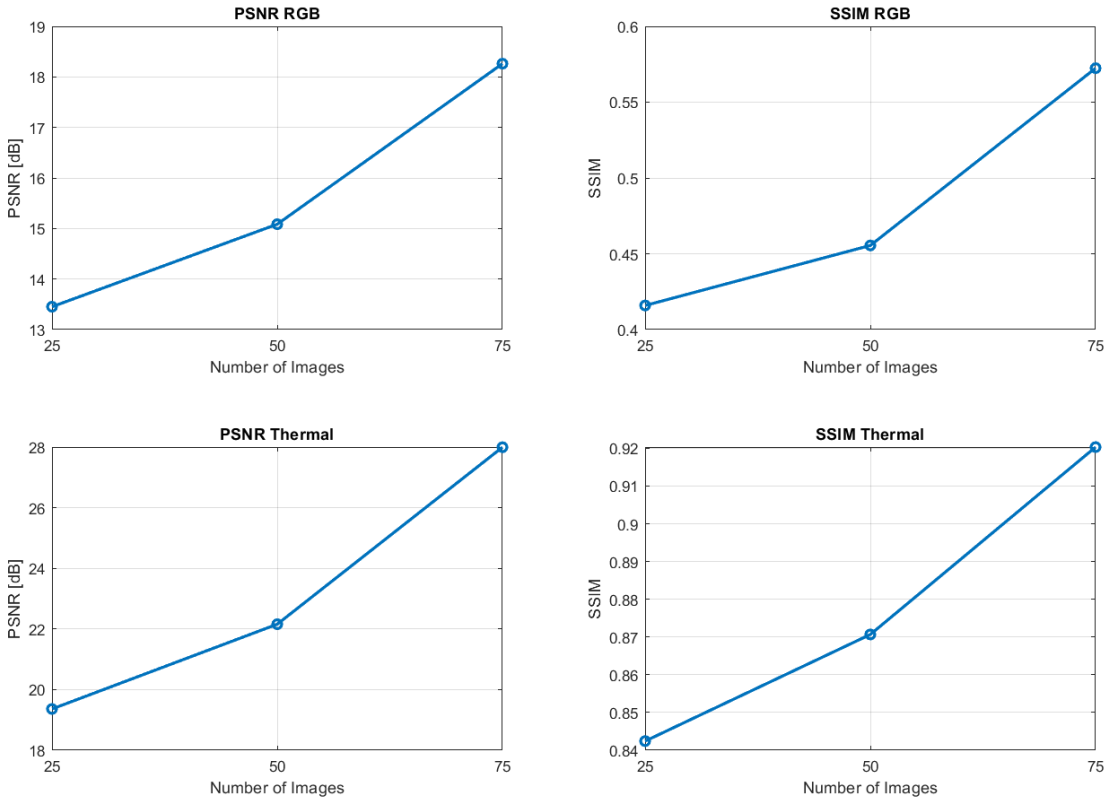


Figure 4: Average performance of ThermoNerf for the 10 scenes of ThermoScenes. The image poses used were globally initialized with every image of the scene, thus excluding the impact of pose estimation on the results.

The small experiment shows a direct link between the quality of the rendering of the evaluation images and the number of images used for the ThermoNerf model. It allows to assume the validity of the model with around 25 images in ThermoScenes, as the metrics indicate generally lower performance for fewer images, but still within a reasonable quality range for this application. This lower performance can mainly be attributed to the coverage of the scene, which is significantly impacted when reducing the number of images. This results in more isolated evaluation images with poorer rendering.

4 Methods

Each topic aborded in this section is implemented in this github repository.

4.1 General Pipeline

The general pipeline of the proposed sliding window method is described in Figure 5. It allows to separate a larger dataset in different sub-scenes, estimate its relative poses, use Mast3R while

avoiding the *OutOfMemory* error and combine each subscene together.

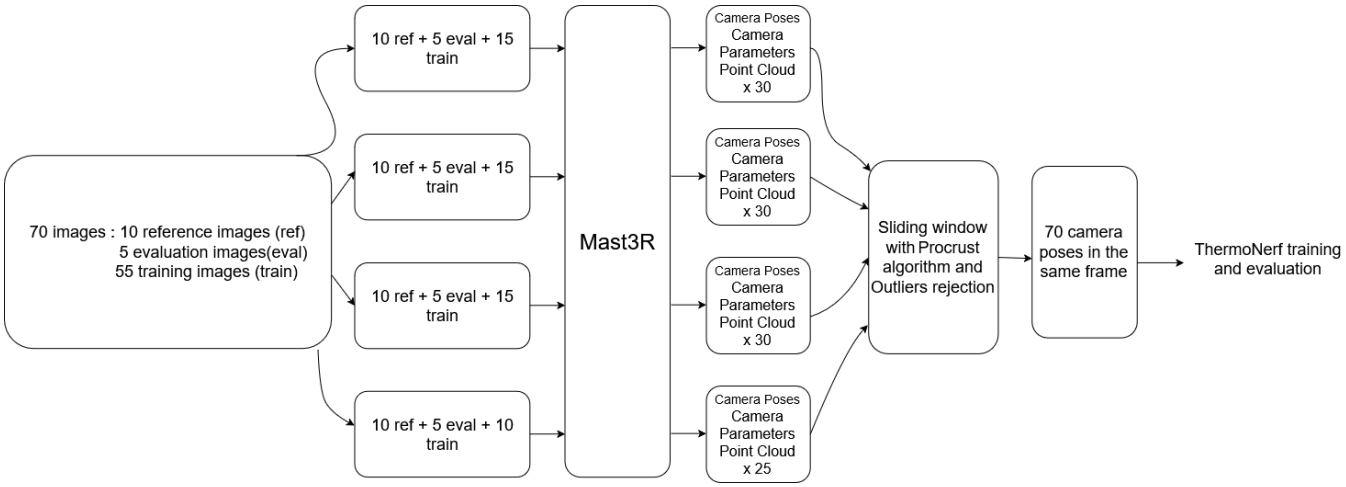


Figure 5: Flowchart for the sliding window operation. The algorithm separates each large dataset in several several subset of images sharing the same references. It then estimates independantly the poses and camera parameters with Mast3r and reproject each sub-results in the same frame using Umeyama and a sliding window algorithm.

4.2 Umeyama algorithm implementation

The Umeyama algorithm was implemented, with an illustration of the implementation being proposed in Figure 6.

Algorithm 1 Umeyamaes Alignment (`ralign`)

Require: Point sets X and Y of size m (dimension) $\times n$ (number of points)
Ensure: Rotation matrix R, scale factor c, translation vector t

$\mu_X \leftarrow \text{mean}(X)$ ▷ De-mean the data
$\mu_Y \leftarrow \text{mean}(Y)$
$X_c \leftarrow X - \mu_X$
$Y_c \leftarrow Y - \mu_Y$
$s_X \leftarrow \text{sum}(X_c^2)/n$ ▷ SVD
$S_{XY} \leftarrow Y_c X_c^T/n$
$[U, D, V] \leftarrow \text{SVD}(S_{XY})$
$S \leftarrow I_m$
if rank(S_{XY}) >m - 1 and det(S_{XY}) <0 **then** ▷ Remove reflection
  $S[m, m] \leftarrow -1$
else if rank(S_{XY}) = m - 1 and det(U) * det(V) <0 **then**
  $S[m, m] \leftarrow -1$
end if
$R \leftarrow U \cdot S \cdot V^T$ ▷ Extract R,c,t
$c \leftarrow \text{trace}(D \cdot S)/s_X$
$t \leftarrow \mu_Y - c \cdot R \cdot \mu_X$
Return R, c, t

Using R, c and t, X can be projected in the frame of Y using:

$$X_{proj} = cRX + t \tag{2}$$

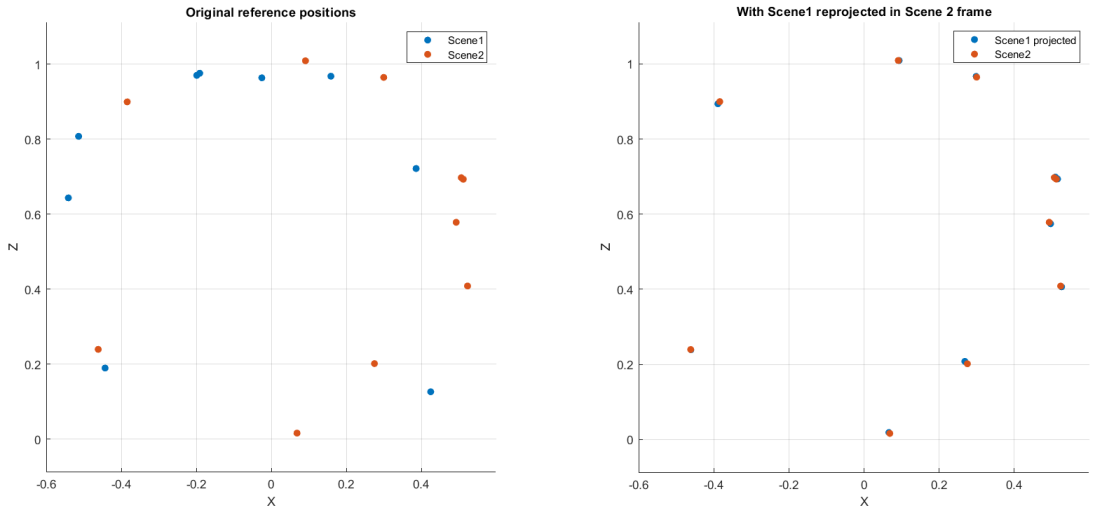


Figure 6: Application of the Umeyama Alignment on 2 equivalent set of points generated by Mast3r for two sub-scenes on the *Heater water Cup* Scene. The 3D points were projected on the XZ plane.

In addition to the position, the orientation of each camera was also rotated. By first transforming R to its quaternion form q_R, each quaternion q_X(corresponding to the orientation of a respective position in X) can be rotated to obtain $q_{X,rotated}$ using simple quaternion multiplication. :

$$q_{X,rotated} = q_R q_X \tag{3}$$

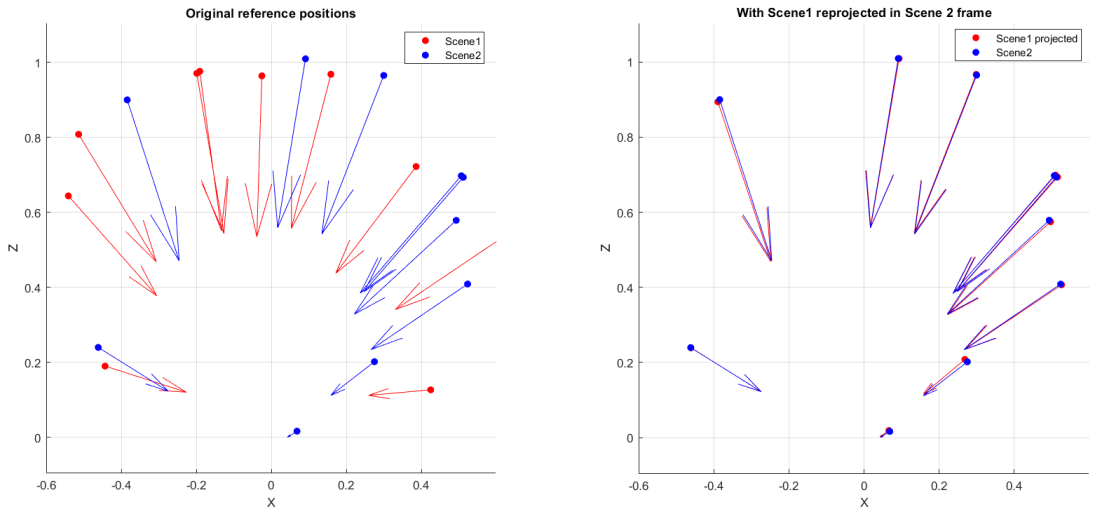


Figure 7: Example of application of the Procurst alignment with the cameras orientation, projected on the XZ plane. The set of points is the same than Figure 6.

4.3 Outliers Rejection

The outliers rejection layer was added by reprojecting the X poses using Equation 2 and measuring the L2 distance between each pair of corresponding points. Then, the points with the greatest distances were set aside, and the transformation was recomputed. This ensures that at least two outliers in the reference set of images do not influence the transformation computed by the Umeyama alignment. In case of the rejection of two perfectly valid points, there would be little to no impact on the transformation.

4.4 Sliding Window

To incorporate each sub-scene into one, one of them was arbitrarily chosen to act as the reference frame for the others. Using the 10 references, the poses of the images for training and evaluation were re-projected in the same frame using the Umeyama algorithm and then merged together. The camera parameters did not need to be changed from the initial Mast3r estimation, as the scale remains the same for each sub-scene computed by Mast3r.

One of the practical challenges was the appropriate transformation from the Mast3r output format to the ThermoNerf and NerfStudio format. Mast3r creates a Colmap-like output with a sparse folder containing the images.bin, cameras.bin, and points3D.ply files. Attempts to manually convert them to a transforms.json were unsuccessful. Instead, the method from NerfStudio, *ns-process-data* with the *–skip-colmap* flag, containing the reference frame change from Colmap to NerfStudio, was used on each sub-scene. The sliding window was then applied to the transforms.json files, recombining them into a single file.

To transform the Mast3r output into a Colmap-like format, the baseline from InstantSplat [1], which effectively incorporates the results into a suitable format, was used.

5 Results and Discussion

Sliding Window

The metrics for the results of the sliding window algorithm, presented in Table 3, Figure 8, and Figure 9, indicate that its performance is comparable to the sub-scene separated results. While the algorithm successfully incorporates the unique information from each sub-scene, it does not significantly enhance the overall scene reconstruction when merged. This approach effectively extends the Mast3r implementation's coverage of a scene but does not improve the overall NeRF representation. The metrics do not show a significant improvement brought by the outliers rejection layer.

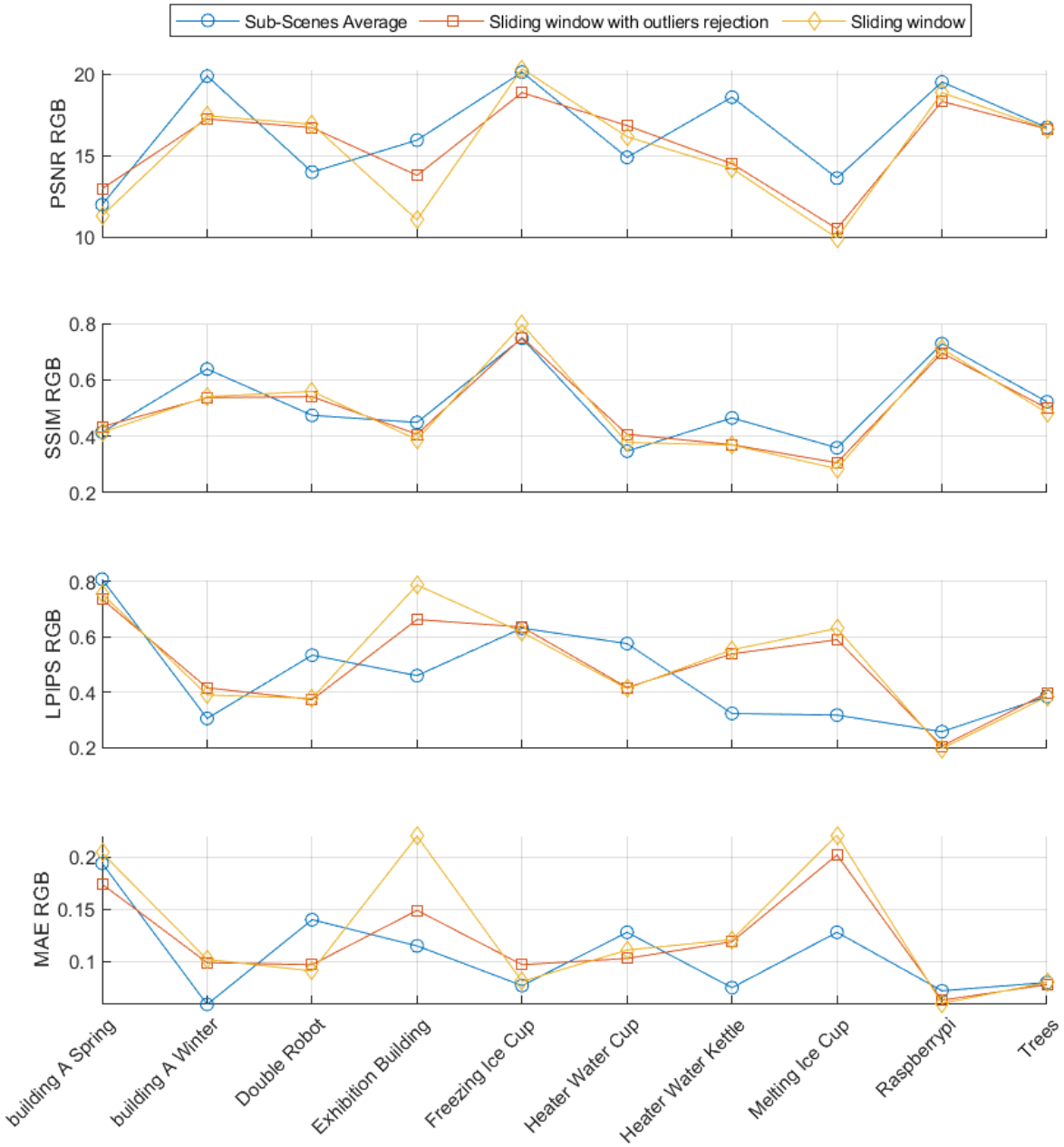


Figure 8: Metrics for the ThermoNerf evaluation of the poses for the RGB images. It was made using 5 evaluation images per scene. The sub-scenes average represents the average of the 5 evaluation images over each sub-scene. Each model was trained for 30 000 steps

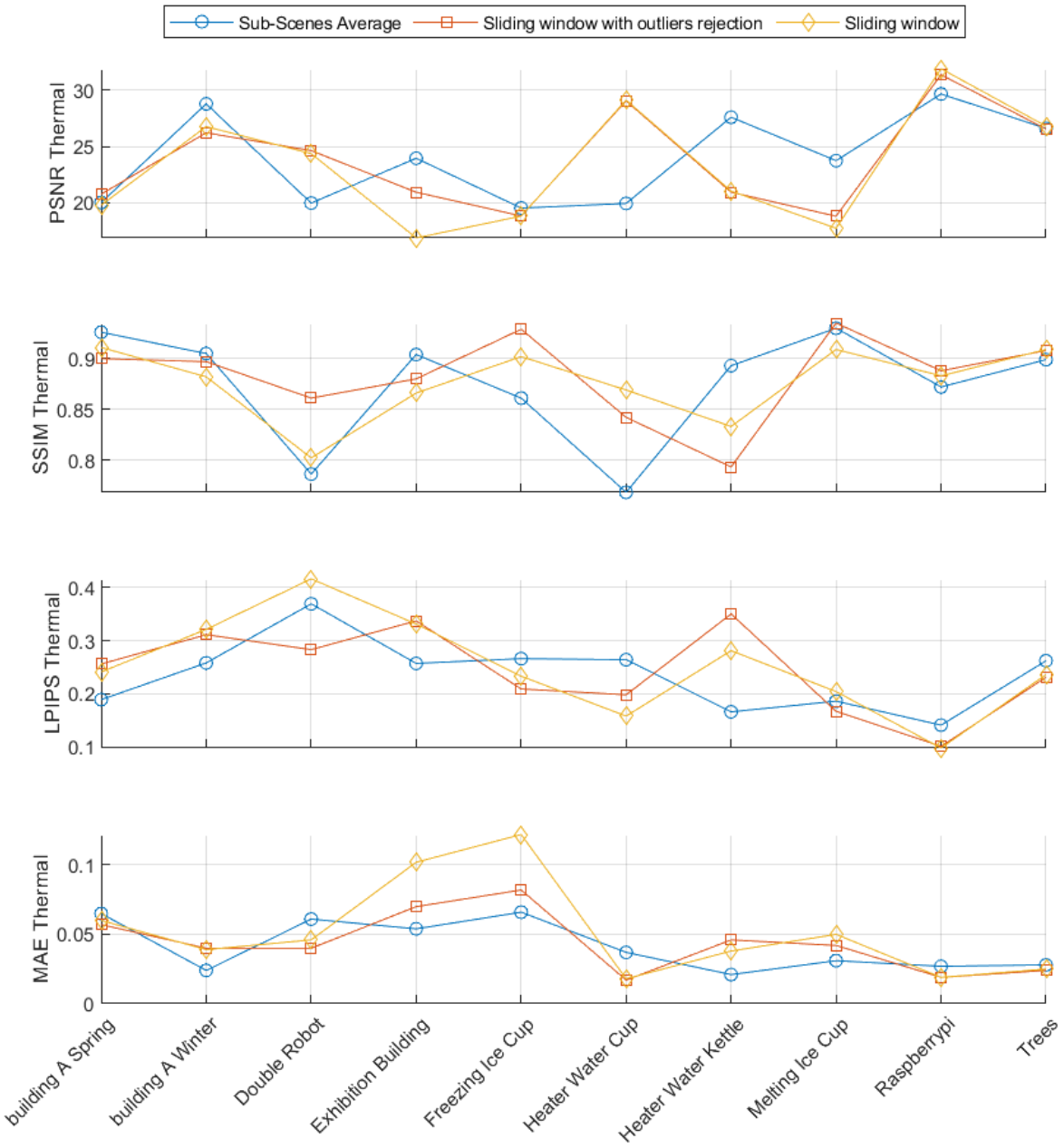
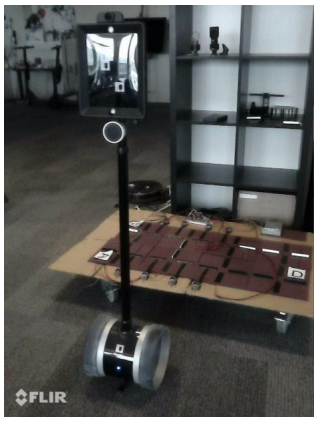


Figure 9: Same as Figure 8 but for the thermal images

Scene	Field	PSNR		SSIM		LPIPS		MAE	
		RGB	Thermal	RGB	Thermal	RGB	Thermal	RGB	Thermal
Building A Spring	Sub-scenes average	12	20.02	0.41	**0.93**	**0.19**	**0.81**	0.194	0.065
	Full scene	11.33	19.79	0.41	0.91	0.24	0.76	0.204	0.06
	Full scene outliers	**12.98**	**20.77**	**0.43**	0.9	0.26	0.74	**0.174**	**0.057**
Building A Winter	Sub-scenes average	**19.89**	**28.79**	**0.64**	**0.91**	**0.26**	**0.31**	**0.059**	**0.024**
	Full scene	17.45	26.78	0.54	0.88	0.32	0.39	0.102	0.039
	Full scene outliers	17.26	26.23	0.54	0.9	0.31	0.42	0.099	0.04
Double Robot	Sub-scenes average	14	19.98	0.47	0.79	0.37	0.53	0.14	0.061
	Full scene	**16.94**	24.37	**0.56**	0.8	0.42	0.38	**0.091**	0.046
	Full scene outliers	16.72	**24.64**	0.54	**0.86**	**0.28**	**0.37**	0.097	**0.04**
Exhibition Building	Sub-scenes average	**15.96**	**23.96**	**0.45**	**0.9**	**0.26**	**0.46**	**0.115**	**0.054**
	Full scene	11.07	16.88	0.39	0.87	0.33	0.79	0.22	0.102
	Full scene outliers	13.8	20.94	0.41	0.88	0.34	0.66	0.149	0.07
Freezing Ice Cup	Sub-scenes average	20.13	**19.53**	0.75	0.86	0.27	0.63	**0.077**	**0.066**
	Full scene	**20.34**	18.82	**0.8**	0.9	0.23	**0.62**	0.081	0.122
	Full scene outliers	18.88	18.83	0.75	**0.93**	**0.21**	0.64	0.097	0.082
Heater Water Cup	Sub-scenes average	14.9	**19.95**	0.35	0.77	0.26	0.58	0.128	0.037
	Full scene	16.17	29.12	0.38	**0.87**	**0.16**	**0.41**	0.111	0.018
	Full scene outliers	**16.85**	29.06	**0.41**	0.84	0.2	0.42	**0.103**	**0.017**
Heater Water Kettle	Sub-scenes average	**18.6**	**27.61**	**0.46**	**0.89**	**0.17**	**0.32**	**0.075**	**0.021**
	Full scene	14.22	21.04	0.37	0.83	0.28	0.55	0.121	0.038
	Full scene outliers	14.51	20.94	0.37	0.79	0.35	0.54	0.119	0.046
Melting Ice Cup	Sub-scenes average	**13.63**	**23.75**	**0.36**	**0.93**	0.19	**0.32**	**0.128**	**0.031**
	Full scene	9.96	17.74	0.29	0.91	0.2	0.63	0.221	0.05
	Full scene outliers	10.53	18.81	0.31	0.94	**0.17**	0.59	0.202	0.042
Raspberry Pi	Sub-scenes average	**19.53**	29.67	**0.73**	0.87	0.14	0.26	0.072	0.027
	Full scene	18.87	**31.92**	0.71	0.88	**0.1**	**0.2**	**0.06**	**0.019**
	Full scene outliers	18.34	31.38	0.7	**0.89**	**0.1**	**0.2**	0.063	**0.019**
Trees	Sub-scenes average	**16.74**	26.61	**0.52**	0.9	0.26	**0.38**	0.08	0.028
	Full scene	16.67	**26.81**	0.48	**0.91**	0.24	**0.38**	0.08	0.025
	Full scene outliers	16.65	26.56	0.5	**0.91**	**0.23**	0.4	**0.078**	**0.024**
Average	Sub-scenes average	**16.53**	**23.98**	**0.51**	0.87	**0.23**	**0.46**	**0.106**	**0.041**
	Full scene	15.3	23.32	0.49	**0.88**	0.25	0.51	0.127	0.051
	Full scene outliers	15.65	23.82	0.49	**0.88**	0.24	0.49	0.142	0.043

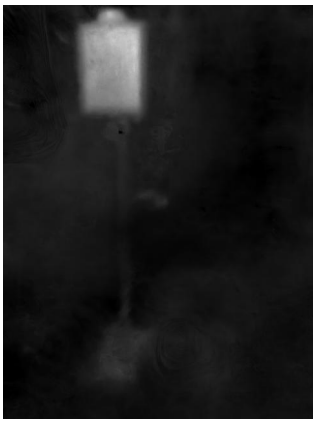
Table 3: Summary of the results for the sliding windows with or without outliers rejection method. This Table is plotted in Figure 8 and 9. The best score for each scene and metrics are **highlighted in bold**.



(a) Ground Truth for RGB

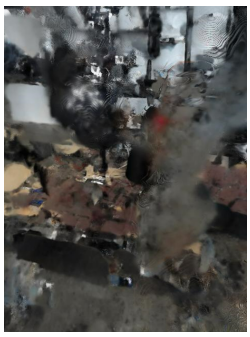
(b) RGB full sliding window rendering, PSNR = 16.94

(c) Ground truth for thermal



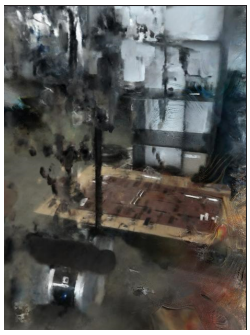
(d) Thermal full sliding window rendering, PSNR = 28.11

(e) RGB rendering for subscene 1, PSNR = 15.73



(f) RGB rendering for subscene 2, PSNR = 12.51

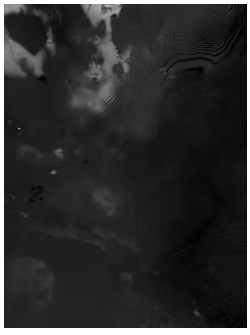
(g) RGB rendering for subscene 3, PSNR = 14.76



(h) RGB rendering for subscene 4, PSNR = 13.80

(i) RGB rendering for subscene 5, PSNR = 15.29

(j) Thermal rendering for subscene 1, PSNR = 21.77



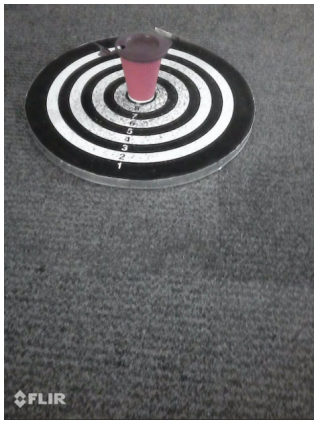
(k) Thermal rendering for subscene 2, PSNR = 17.84

(l) Thermal rendering for subscene 3, PSNR =24.77

(m) Thermal rendering for subscene 4, PSNR = 15.88

(n) Thermal rendering for subscene 5, PSNR = 19.93

Figure 10: Evaluation of one of the evaluation images across different steps of the sliding windows algorithm for the scene *Double Robot*. Despite the lack of propice camera poses for the sub-scene 2 and 4, the final output correctly uses the more suitable camera poses from the others sub-scenes. The model and training parameters are the same as in Figure 8. This is a typical example but depending on the scene or the seed, the full sliding window results can slightly vary compared to the other sub-scenes



(a) Ground truth for RGB

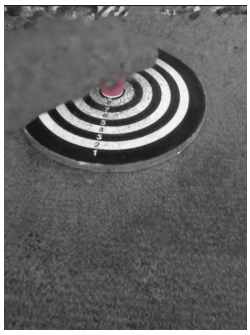
(b) RGB full sliding window rendering, PSNR = 17.66

(c) Ground truth for thermal

(d) Thermal full sliding window rendering, PSNR = 33.63

(e) RGB rendering for subscene 1, PSNR = 14.38

(f) RGB rendering for subscene 2, PSNR = 14.91



(g) RGB rendering for subscene 3, PSNR = 18.98

(h) RGB rendering for subscene 4, PSNR = 17.31

(i) RGB rendering for subscene 5, PSNR = 20.53

(j) Thermal rendering for subscene 1, PSNR = 25.36

(k) Thermal rendering for subscene 2, PSNR = 21.35

(l) Thermal rendering for subscene 3, PSNR =20.75

(m) Thermal rendering for subscene 4, PSNR = 20.27

(n) Thermal rendering for subscene 5, PSNR = 27.12

Figure 11: Same as Figure 10 but for the *Heater Water Cup* Scene.

Colmap and Mast3r comparison

The comparison between master and colmap initization is displayed below. The proposed method struggles to match colmap globally initialized pose standards.

Scene	Field	PSNR		SSIM		LPIPS		MAE	
		RGB	Thermal	RGB	Thermal	RGB	Thermal	RGB	Thermal
Building A Spring	Colmap poses	17.76	25.35	0.565	0.912	0.33	0.2	0.089	0.035
	Sliding window	12.98	20.77	0.43	0.9	0.26	0.74	0.174	0.057
Building A Winter	Colmap poses	19.47	29.34	0.6	0.88	0.36	0.278	0.072	0.023
	Sliding window	17.26	26.23	0.54	0.9	0.31	0.42	0.099	0.04
Double Robot	Colmap poses	19.59	31.1	0.69	0.94	0.278	0.14	0.061	0.019
	Sliding window	16.72	24.64	0.54	0.86	0.28	0.37	0.097	0.04
Exhibition Building	Colmap poses	22.14	34.98	0.63	0.97	0.25	0.07	0.044	0.013
	Sliding window	13.8	20.94	0.41	0.88	0.34	0.66	0.149	0.07
Freezing Ice Cup	Colmap poses	23.72	26.5	0.83	0.98	0.56	0.04	0.046	0.024
	Sliding window	18.88	18.83	0.75	0.93	0.21	0.64	0.097	0.082
Heater Water Cup	Colmap poses	17.4	30.57	0.5	0.9	0.32	0.122	0.104	0.015
	Sliding window	16.85	29.06	0.41	0.84	0.2	0.42	0.103	0.017
Heater Water Kettle	Colmap poses	17.79	25.56	0.48	0.89	0.46	0.19	0.085	0.023
	Sliding window	14.51	20.94	0.37	0.79	0.35	0.54	0.119	0.046
Melting Ice Cup	Colmap poses	19.75	36.31	0.6	0.99	0.17	0.03	0.059	0.01
	Sliding window	10.53	18.81	0.31	0.94	0.17	0.59	0.202	0.042
Raspberry Pi	Colmap poses	21.59	34.1	0.79	0.96	0.15	0.05	0.041	0.012
	Sliding window	18.34	31.38	0.7	0.89	0.1	0.2	0.063	0.019
Trees	Colmap poses	19.8	32.34	0.65	0.95	0.32	0.17	0.05	0.014
	Sliding window	16.65	26.56	0.5	0.91	0.23	0.4	0.078	0.024
Average	Colmap poses	**19.91**	**30.62**	**0.63**	**0.95**	**0.37**	**0.13**	**0.065**	**0.017**
	Sliding window	15.65	23.82	0.49	0.88	0.24	0.49	0.142	0.043

Table 4: Colmap vs Sliding window comparison (with outliers rejection). The proposed method under performs compared to the initialized poses from Colmap proposed in ThermoScenes. The results for Colmap were recomputed for this project and match the results from the original paper. And example of the rendering of the *Heater Water Cup* is proposed in Supplementary material 1-2, downloadable here

From the results, several factors could explain the difference in performance between the two models.

1. First of all, when comparing poses computed globally versus locally several times, one could expect the former to perform better. A global implementation has the advantage of being able to optimize the entire scene simultaneously. This difference in approach is detrimental to the proposed method, which lacks proper global optimization for each pose in large datasets, a feature not implemented here. This would most likely bring more consistency across each camera pose and reduce the amount of noise created.

2. Second, the outliers rejection method in the Umeyama algorithm seems to be too light compared to what the method aims to achieve. In comparison to other classical methods that incorporate incremental adjustments of the poses (such as Mast3r and Colmap), it seems beneficial to add such a method here. The goal would be to add redundancy to ensure that single outliers in the reference set of training images could be eliminated. A method like this could,

for example, compute each image several times in different sub-scenes and fuse the results in an appropriate way.

3. Another potential improvement could be to increase the number of reference images for the subscenes. This approach would ensure greater consistency across the different sub-scenes at the cost of increased computational demands, which would ideally result in improved performance.

ThermalScenes Evaluation

The Mast3r initialization did not yield satisfying results for the challenging scenes in ThermalScenes. The renderings were blurry and/or contained multiple objects. Interestingly, other models utilizing Gaussian Splatting (GS) produced much better results with consistent rendering using the Mast3r pose initialization. The main reason for these positive results compared to traditional NeRF models could be explained by the difference in pose optimization. In this specific case, GS is more accurate and produces better results. To the author's best knowledge, there is currently no open-sourced GS model or code equivalent to ThermoNerf. The results of the InstantSplat rendering is proposed in Supplementary Material 3-4, downloadable here.

6 Conclusion

This project compared Mast3r and Colmap for initial pose estimation and analyzed their impact on the performance of a multi-modal novel view model like ThermoNerf. Additionally, it was shown that while the sliding window approach successfully merges several sub-scenes to compensate for Mast3r's memory limitations, it underperforms compared to the globally implemented Colmap poses, despite systematically outputting camera poses. The project highlights the potential of using Mast3r initialization in situations where Colmap fails to output poses.

References

[1] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2(3):4, 2024.

[2] Mariam Hassan, Florent Forest, Olga Fink, and Malcolm Mielle. Thermonerf: Multimodal neural radiance fields for thermal novel view synthesis. *arXiv preprint arXiv:2403.12154*, 2024.

[3] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2025.

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[5] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[7] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

[8] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

7 Annex