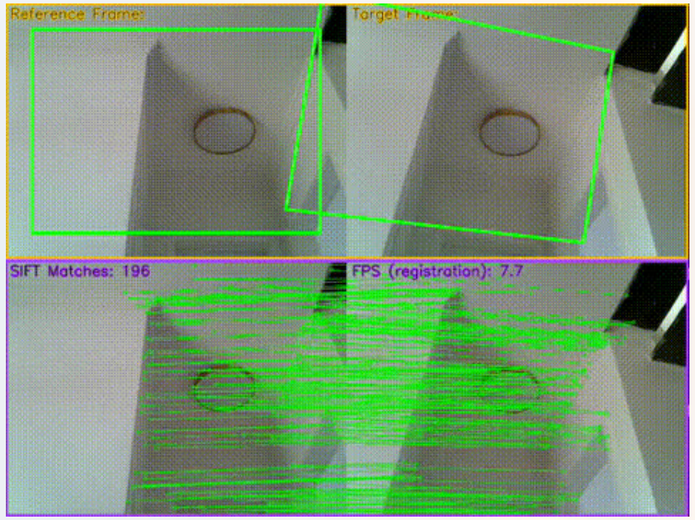
XFeat: Accelerated Features for Lightweight Image Matching



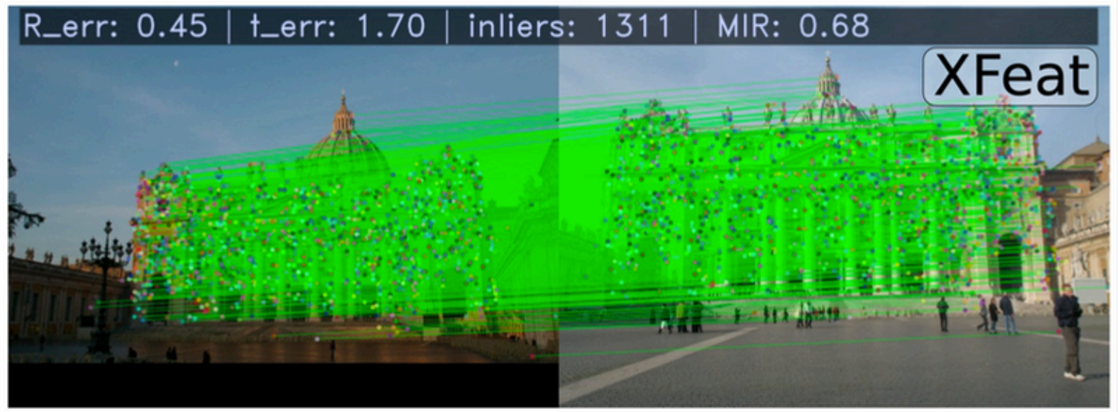
Advanced Computer Vision 2024/25
Roman Simone e Moiola Christian

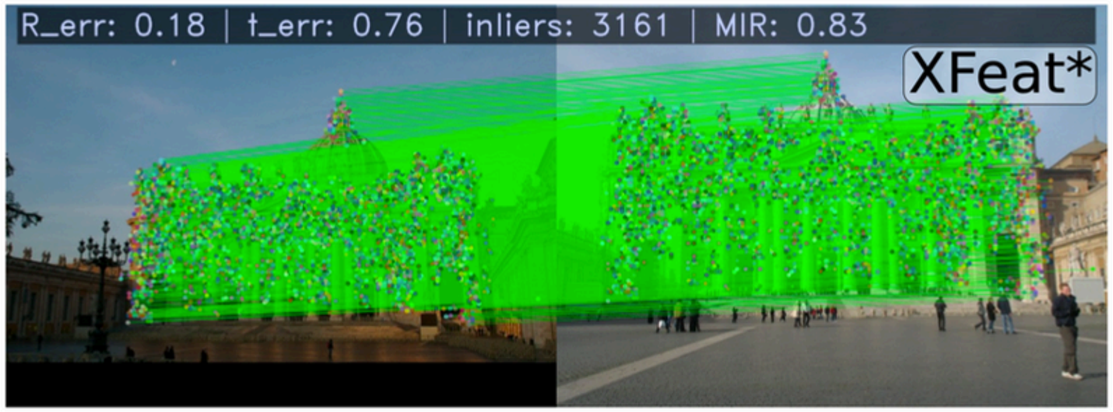
Introduction

XFeat Objective: lightweight CNN for detecting, extracting and matching local features.

Matching Types:

1. **Sparse Matching (XFeat):**
   - Identifies salient points (or keypoints)
   - Optimized for computational efficiency
2. **Semi-Dense Matching (XFeat*):**
   - Enhances sparse matches by refining at pixel-level
   - Balances precision and resource efficiency





State of Art

Modern image matching balances **accuracy and efficiency**, with methods divided into **standard** for precision and **fast** for speed on constrained devices.

Standard

- Traditional Approaches (ex SiFT)
- Middle-End Approaches (ex SuperGlue)
- End-to-End Approaches (ex DISK)

👍 **Performance**
👎 **Efficency**

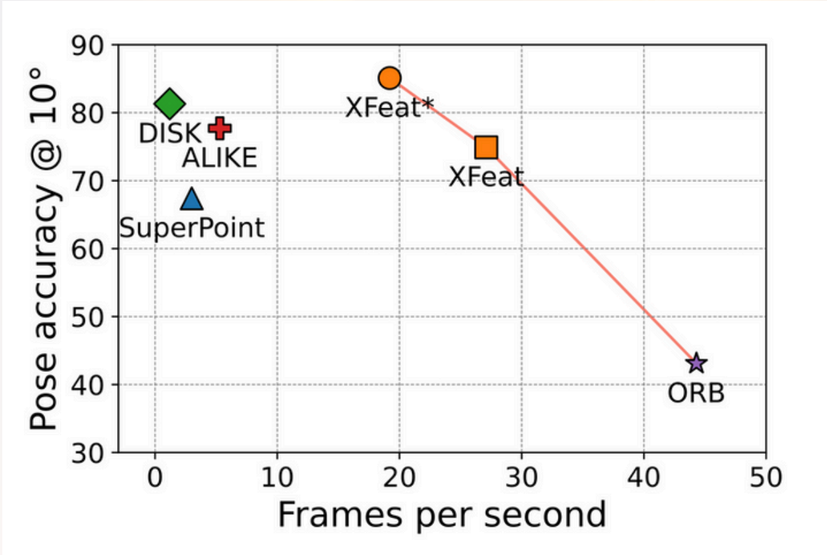
Fast

- ALIKE
- ORB

👍 **Efficency**
👎 **Performance**

State of Art

As we can see XFeat reach the **best trade off between efficency and performance**



Xfeat (sparse matching)

Keypoint Detection

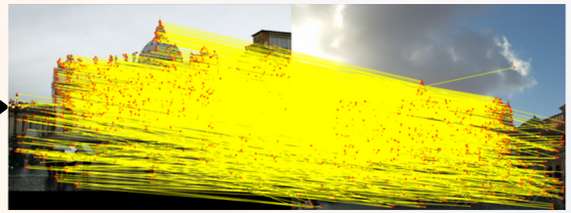
- Identifying **salient points** in the image
- Pixel level

Description & Reliability extraction

- **Description:** feature descriptors map for matching.
- **Reliability:** quantifies the confidence of each descriptor.

Matching

- Mutual Nearest Neighbor
- **Select** top keypoint
- **Match** keypoint by **cosine similarity** using their descriptors

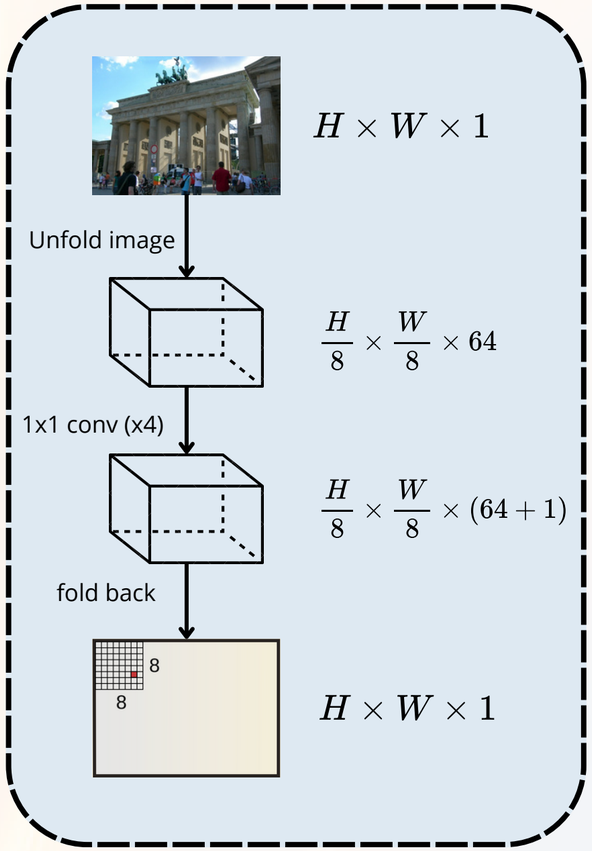


Keypoint Detection

Purpose: Find *sparse features* with minimal computational cost

Procedure:

1. **Divide the input image** in a 2D grid composed:
   - **8 × 8 pixels** on each grid cell
   - Reshape each cell into **64 channels**
2. **Apply 1×1 convolutions (x4)** to regress keypoint embeddings
   - Obtain $\mathbf{K} \in \mathbb{R}^{H/8 \times W/8 \times (64+1)}$
3. **Classify keypoints** within each cell or assign them to a "dustbin" if no keypoint is detected.
4. **Fold back** to image and they obtain the map of keypoints.



Description & Reliability extraction

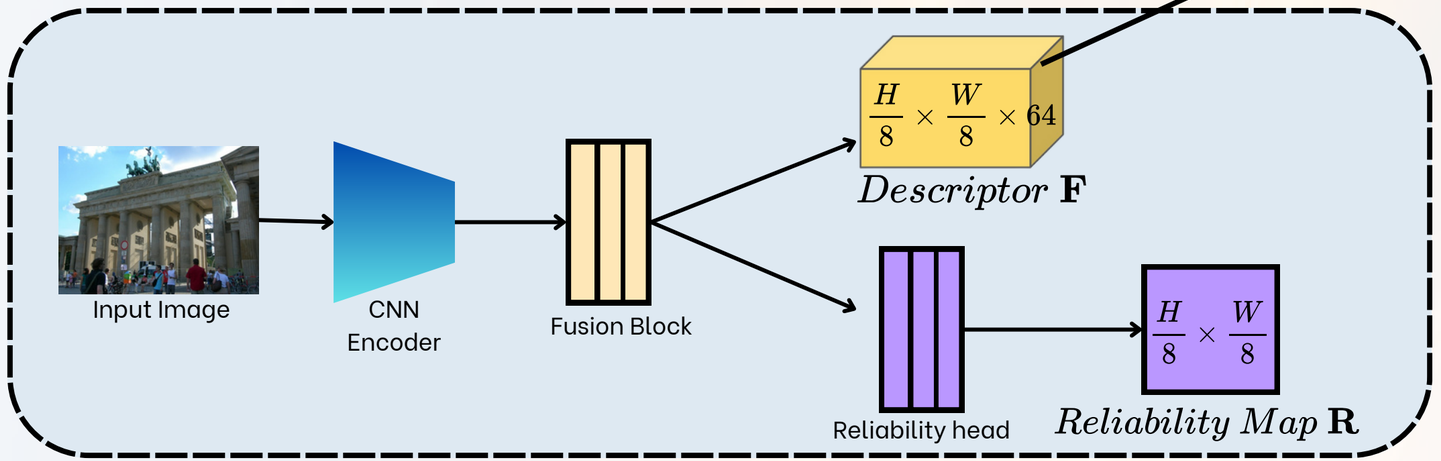
Purpose:

- Provides reliable **local descriptors** and **confidence measures** for efficient image matching.

Description:

- Use of a CNN backbone to obtain:
  - **Feature map** $F \in \mathbb{R}^{H/8 \times W/8 \times 64}$
  - **Reliability map** for feature matching confidence, $R \in \mathbb{R}^{H/8 \times W/8}$

Associated to a Cell 8x8 pixels, not to a single pixel

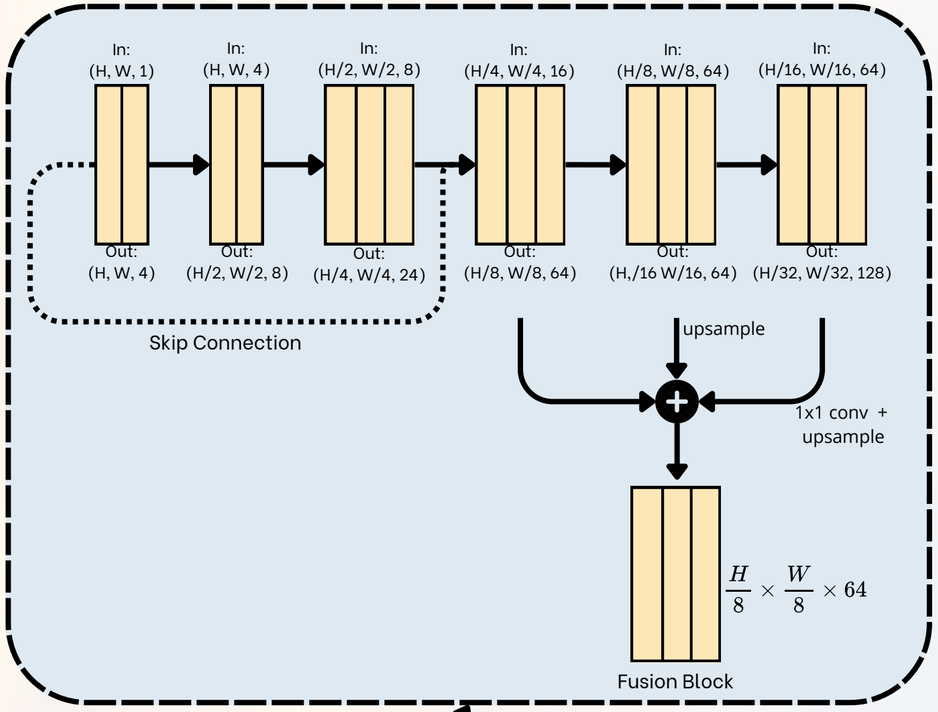


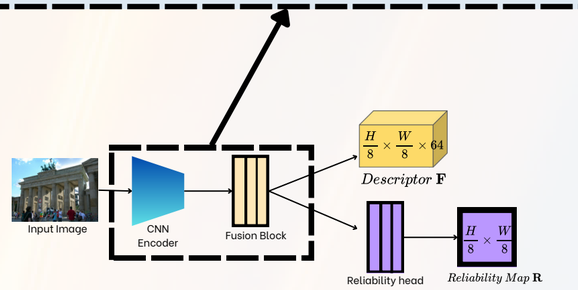
Description Head - Backbone

Design:

The main bottleneck in CNN is ***H*W*C*** so to alleviate the problem, the backbone is implemented with these features:

- **Progressive Downsampling:** Reduces spatial resolution step-by-step
- **Channel Depth Management:** Dynamically adjusts channel depth
- **Multiscale Integration:** Merges features across multiple resolutions(1/8, 1/16, 1/32) to a common resolution (H/8×W/8)





Pipeline - xfeat* (semi- dense matching)

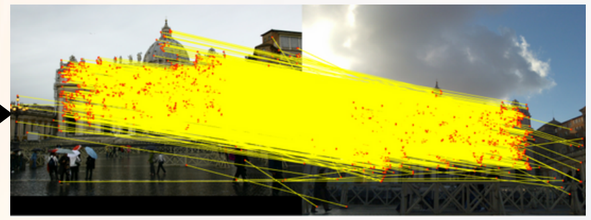
Description & Reliability extraction

- **Description:** feature descriptors map for matching.
- **Reliability:** quantifies the confidence of each descriptor.

Matching
(Mutual Nearest Neighbor)

Match Refinement

- Enhances at pixel-level precision for **semi-dense matching**.



N.B. NOT PIXEL LEVEL!!
CELL LEVEL 64 pixels

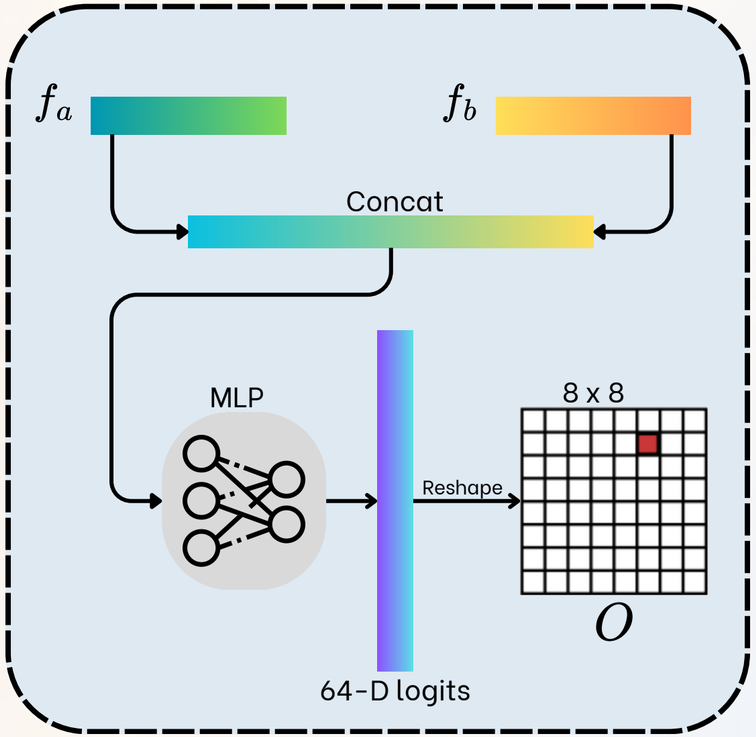
Refinement Module

Purpose:

- Achieve pixel level precision on semi-dense image matching. This due to descriptors matching are at cell level (64 pixels)

Procedure:

1. **Initial Matching**: Use mutual nearest neighbor techniques to pair features and from image pair .
2. **Concat the descriptors** that are matched
3. **Offset Calculation**: Employ an MLP to predict pixel offsets, so the right pixel to match.



Dual-Softmax and Reliability Losses

Xfeat is trained in a supervised manner with pixel-level ground truth correspondences.

1. **Descriptors -> Local descritors Loss** (L_{ds})
   - **Purpose:** Optimize local descriptor matching
   - **Key Steps:**
     - Compute similarity matrix $S = F_1 F_2^T$
     - Match features bidirectionally (forward + reverse)

$$L_{\text{ds}} = -\sum_i \left[\log(\text{softmax}_r(S))_{ii} + \log(\text{softmax}_r(S^T))_{ii}\right]$$

2. **Reliability ->Reliability Loss** (L_{rel})
   - **Purpose:** Train reliability map to quantify feature confidence
   - **Key Features:**
     - Used dual-softmax probabilities as supervision
     - Use of sigmoid activation function

$$L_{\text{rel}} = |\sigma(R_1) - (\overline{R}_1 \odot \overline{R}_2)| + |\sigma(R_2) - (\overline{R}_1 \odot \overline{R}_2)|$$

$$\bar{R}_1 = \max_r(\text{softmax}_r(S))$$

Keypoints Loss, Pixel Offset and Final Loss

3. Refinement Module-> Pixel offsets Loss (L_{fine})

- **Supervision:**
  - Ground-truth correspondences $M_{I_1 \leftrightarrow I_2}$
- **Objective:**
  - Refine matches to pixel-level precision using **NLL** loss

$$\mathcal{L}_{\text{fine}} = -\sum_i \log(\text{softmax}(\mathbf{o}_i))_{\overline{y}_i, \overline{x}_i}$$

4. **Keypoint-> Keypoints Loss** (L_{kp})

- **Supervision:**
  - Knowledge distillation form ALIKE's tiny backbone
- **Process:**
  - Map keypoints to linear indices:
    $tidx = t_x + t_y \cdot 8, t_{idx} \in \{0, 1, \ldots, 64\}$
  - Use **NLL loss** for detection

$$\mathcal{L}_{\text{kp}} = -\sum_k \log(\text{softmax}(\mathbf{k}_{i,j}))_{t_{\text{idx}}}$$

Final Loss: $L = \alpha L_{\text{ds}} + \beta L_{\text{rel}} + \gamma L_{\text{fine}} + \delta L_{\text{kp}}$

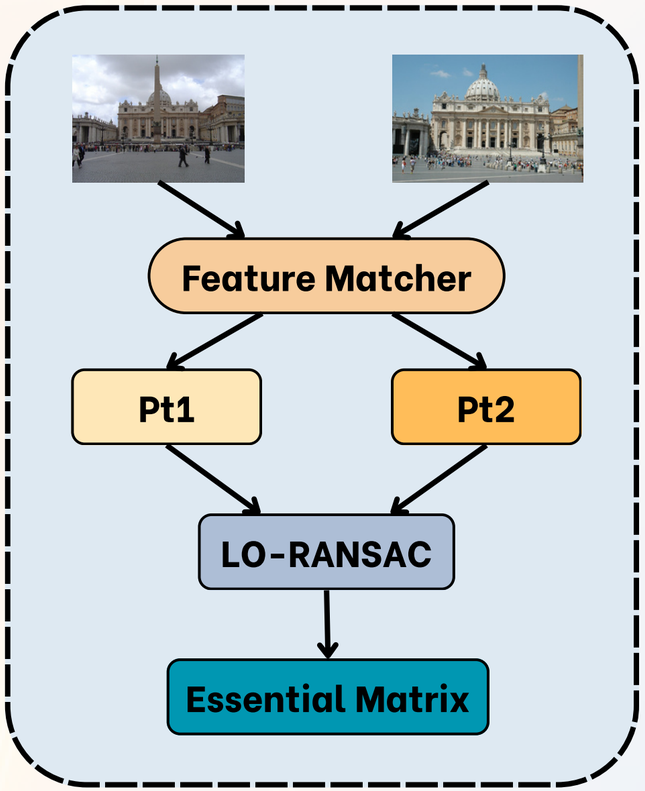
Experiments: Relative Pose Estimation

Definition:

- Determines the spatial relationship between two camera poses by estimating the rotation and translation that align their views

Metrics:

- **AUC@{5°,10°,20°}:** Measure accuracy across angular threshold
- **Acc@10°:** Proportion of poses with angular error below 10 degrees
- **Mean Inlier Ratio (MIR):** Ratio of matching points that comply with the esimated model



	Method	AUC@5°	AUC@10°	AUC@20°	Acc@10°	MIR	#inliers	dim	FPS
Standard	SiLK [9]	14.7	21.5	29.3	31.9	0.17	235	**32-f**	2.8 ± 0.08
	SiLK* [9]	16.2	23.2	31.8	34.7	0.14	478	**32-f**	2.9 ± 0.12
	SuperPoint [7]	37.3	50.1	61.5	67.4	0.35	495	256-f	**3.0 ± 0.07**
	DISK [42]	53.8	65.9	75.0	**81.3**	**0.72**	1231	128-f	1.2 ± 0.01
	DISK* [42]	**55.2**	**66.8**	**75.3**	**81.3**	0.71	**1997**	128-f	1.2 ± 0.01
Fast	ORB [34]	17.9	27.6	39.0	43.1	0.25	288	**256-b**	**44.3 ± 1.18**
	ZippyPoint [13]	23.6	34.9	46.3	51.8	0.23	192	**256-b**	+1.8 ± 0.06
	ALIKE [46]	49.4	61.8	71.4	77.7	0.47	333	64-f	5.3 ± 0.33
	XFeat	42.6	56.4	67.7	74.9	0.55	892	64-f	27.1 ± 0.33
	XFeat*	**50.2**	**65.4**	**77.1**	**85.1**	**0.74**	**1885**	64-f	19.2 ± 1.12

Table 1: Relative camera pose estimation results for the Megadepth-1500 dataset,

Experiments: Homography Estimation

Definition:

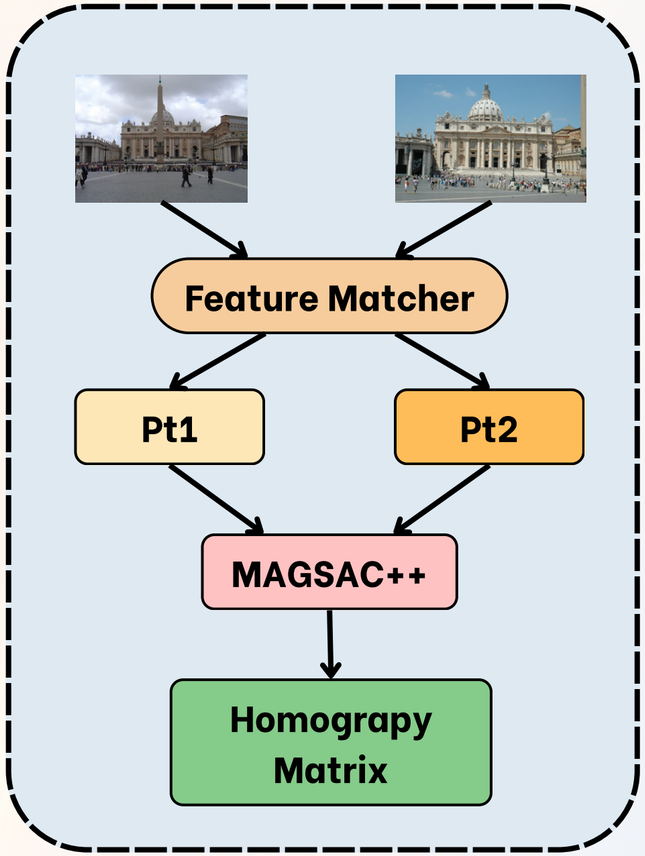
- Computes a transformation matrix mapping points from one image plane to another under planar assumptions

Metrics:

- **Mean Homography Accuracy (MHA):** Average accuracy at threshold of {3px, 5px, 7px} pixel error

Method	Illumination MHA			Viewpoint MHA		
	@3	@5	@7	@3	@5	@7
SiLK	78.5	82.3	83.8	48.6	59.6	62.5
SuperPoint	**94.6**	98.5	98.8	**71.1**	**79.6**	**83.9**
DISK	**94.6**	**98.8**	**99.6**	66.4	77.5	81.8
ORB	74.6	84.6	85.4	63.2	71.4	78.6
ZippyPoint	94.2	96.9	98.5	66.1	76.8	80.7
ALIKE	94.6	**98.5**	**99.6**	68.2	77.5	81.4
XFeat	**95.0**	98.1	98.8	**68.6**	**81.1**	**86.1**

Table 2:Homography estimation on HPatches results.



Our Improvement

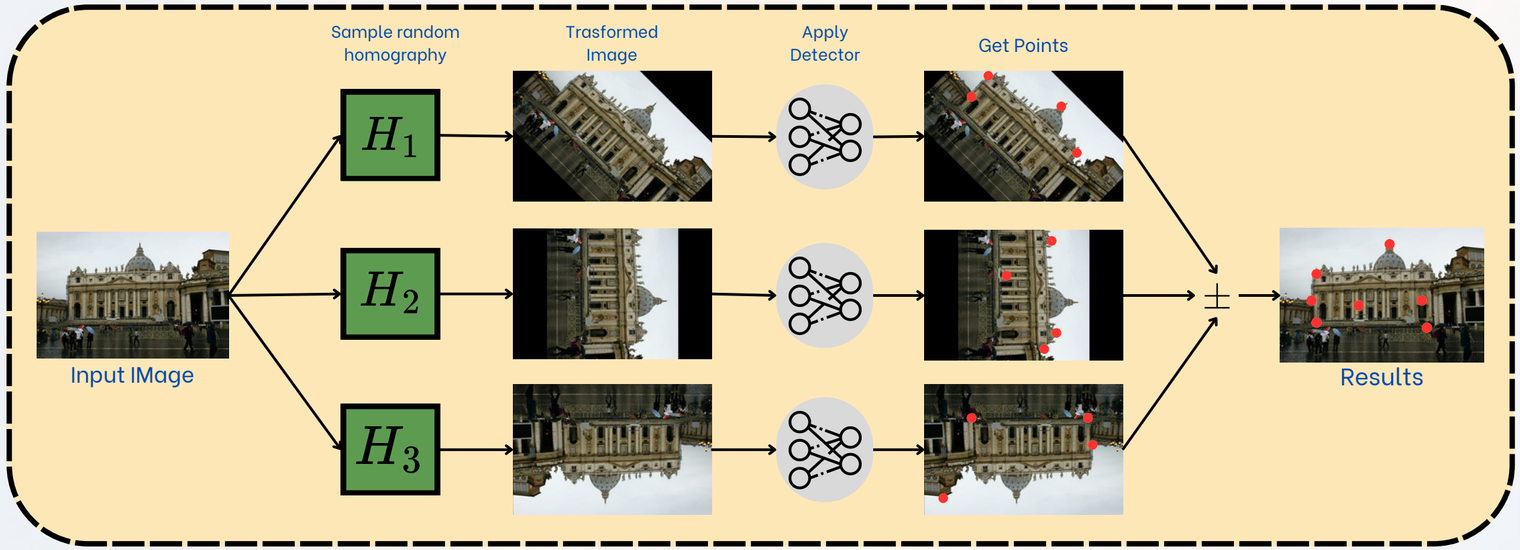
XFeat trasformed

Idea:

- Search to improve the selection of features with somo homography trasformation

Procedure:

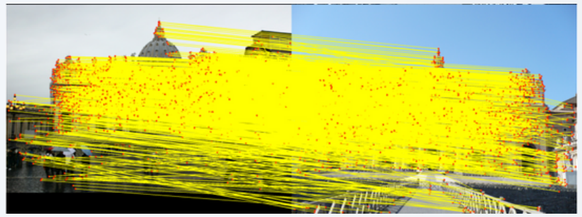
1. Sample random homography
2. Apply trasnformation
3. Apply detector
4. Unify/Intersect points



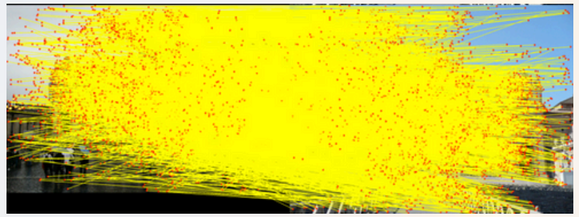
XFeat trasformed - Results

Qualitative:

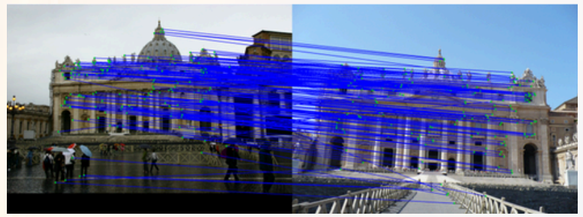
Original



XFeat original
1293 points



XFeat Trasformed unified
3810 points (x3 original)



XFeat Trasformed intersected
181 points (÷7 original)

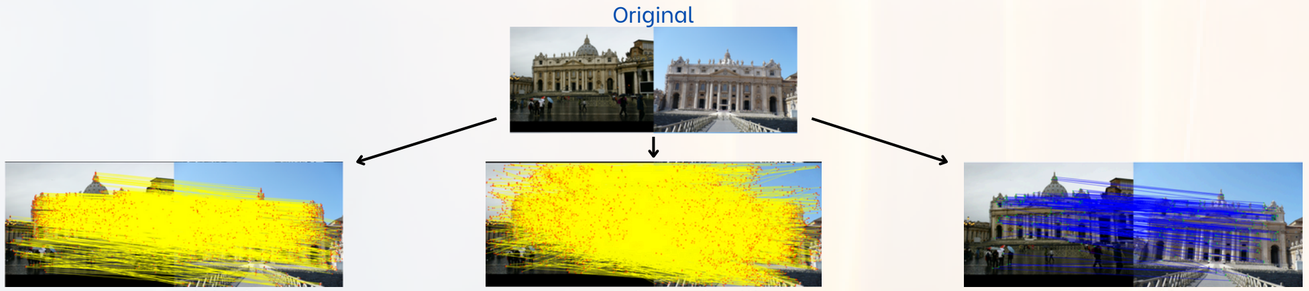
Quantitative:

Method	N trasnformation	AUC@5°	AUC@10°	AUC@20°	ACC@10°	FPS
XFeat original	0	**40.4**	54.4	66.2	73.5	**6.1±0.2**
XFeat trasformed unify	1	33.5	46.4	57.5	63.9	2.3±0.2
XFeat trasformed unify	2	25.0	37.0	49.2	55.1	1.4±0.2
XFeat trasformed unify	3	23.1	34.4	46.6	51.9	1.0 ±0.2
XFeat trasformed intersect	1	**40.4**	**54.8**	**66.5**	**73.9**	3.1±0.2
XFeat trasformed intersect	2	33.8	49.0	62.6	70.5	2.1±0.2
XFeat trasformed intersect	3	29.3	43.9	58.0	66.1	1.7 ±0.2

Table 4: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

XFeat trasformed - Results

Qualitative:



Quantitative:

Method	N trasnformation	AUC@5°	AUC@10°	AUC@20°	ACC@10°	FPS
XFeat original	0	**40.4**	54.4	66.2	73.5	**6.1±0.2**
XFeat trasformed unify	1	33.5	46.4	57.5	63.9	2.3±0.2
XFeat trasformed unify	2	25.0	37.0	49.2	55.1	1.4±0.2
XFeat trasformed unify	3	23.1	34.4	46.6	51.9	1.0 ±0.2
XFeat trasformed intersect	1	**40.4**	**54.8**	**66.5**	**73.9**	3.1±0.2
XFeat trasformed intersect	2	33.8	49.0	62.6	70.5	2.1±0.2
XFeat trasformed intersect	3	29.3	43.9	58.0	66.1	1.7 ±0.2

Table 4: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

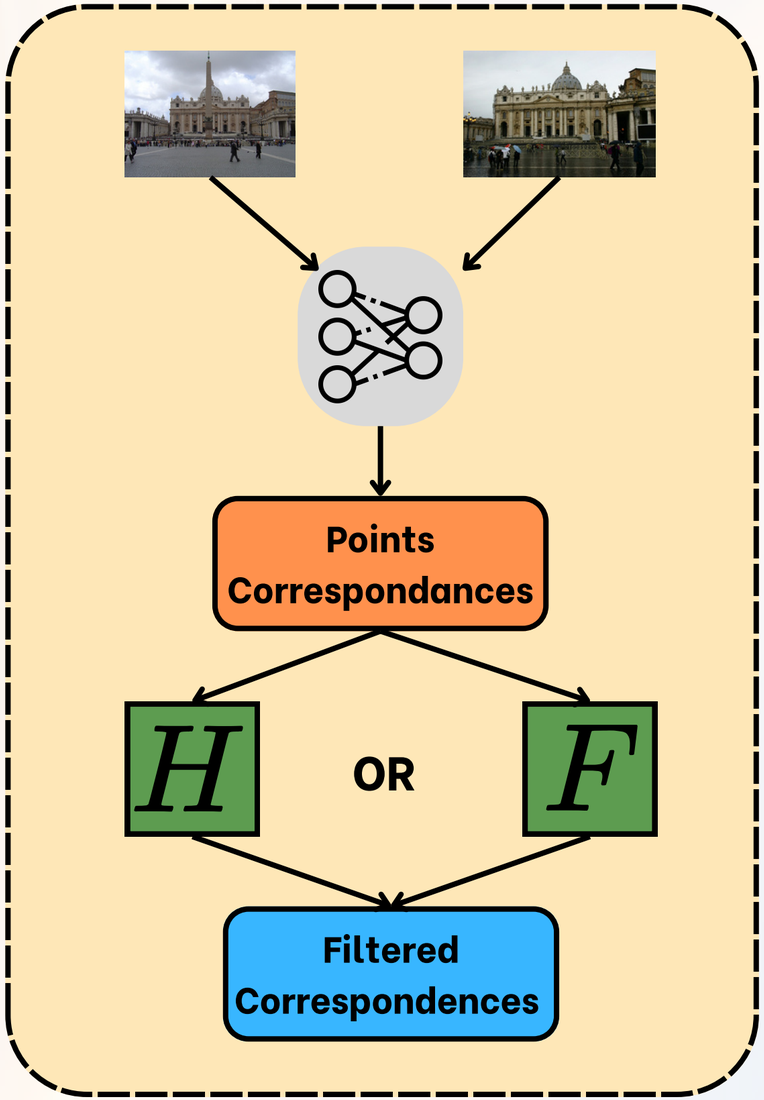
XFeat refined

Idea:

- Refine the points correspondences using Homography or Fundamental Matrix

Procedure:

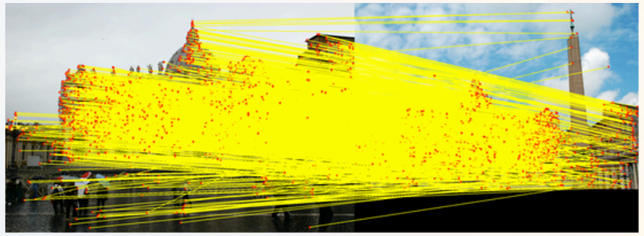
1. **Extract Point Correspondences:** Use XFeat to match features between two images.
2. **Compute Transformation Matrix:** Estimate the **Homography** or **Fundamental** Matrix.
3. **Filter Correspondences:** Refine the point correspondences by keeping only the inliers that are consistent with the estimated matrix.



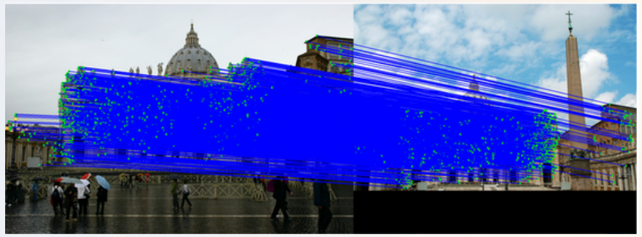
XFeat refined - results

Qualitative:

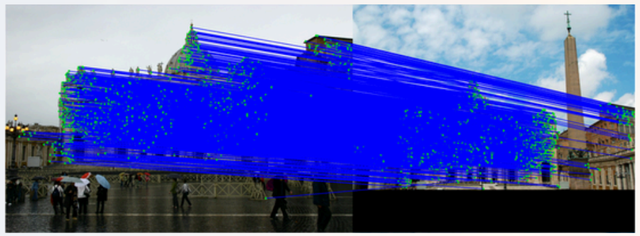
Original



XFeat original
1566 points



XFeat refined (Homography)
1193 points



XFeat refined (Fundamental)
1317 points

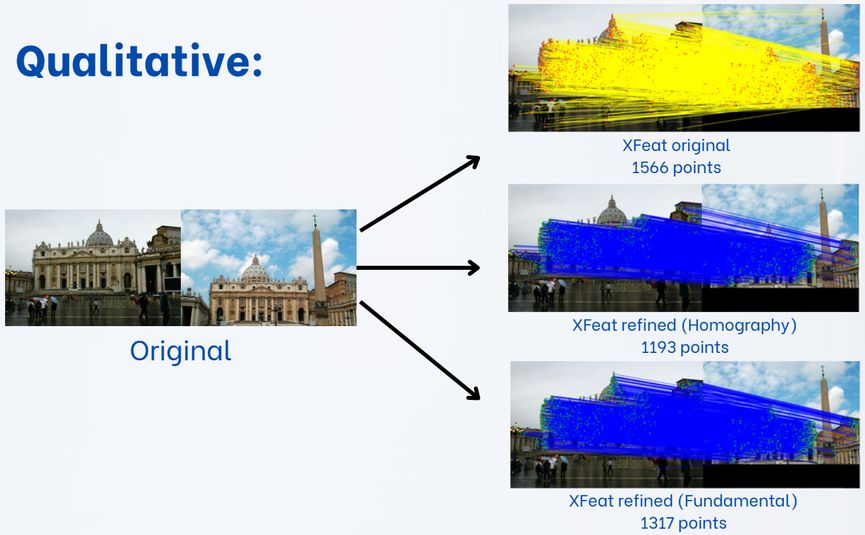
Quantitative:

Method	AUC@5°	AUC@10°	AUC@20°	Acc@10°	FPS
XFeat original	40.4	54.4	66.2	73.5	**6.1±0.2**
XFeat refined (H)	**41.8**	**56.2**	**68.0**	**75.7**	6.0 ±0.2
XFeat refined (F)	40.5	54.7	66.8	74.6	6.0 ±0.2

Table 6: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

XFeat refined - results

Qualitative:



Quantitative:

Method	AUC@5°	AUC@10°	AUC@20°	Acc@10°	FPS
XFeat original	40.4	54.4	66.2	73.5	**6.1±0.2**
XFeat refined (H)	**41.8**	**56.2**	**68.0**	**75.7**	6.0 ±0.2
XFeat refined (F)	40.5	54.7	66.8	74.6	6.0 ±0.2

Table 6: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

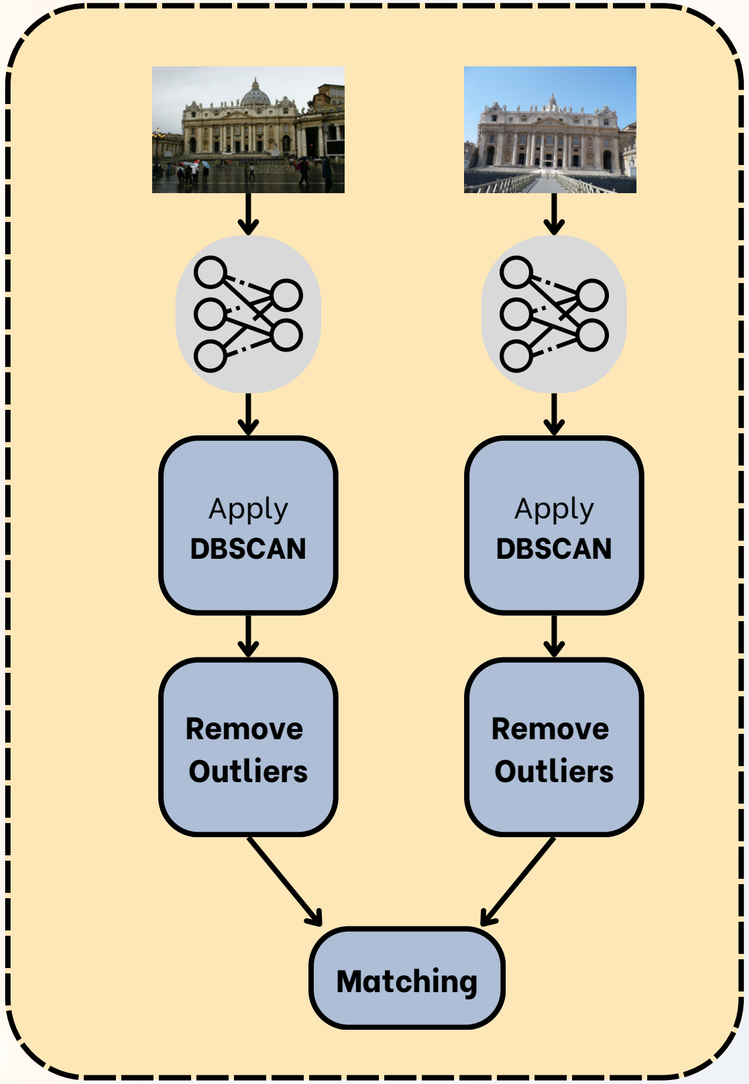
XFeat* clustering

Idea:

- Improve the semi-dense matching by removing outliers

Procedure:

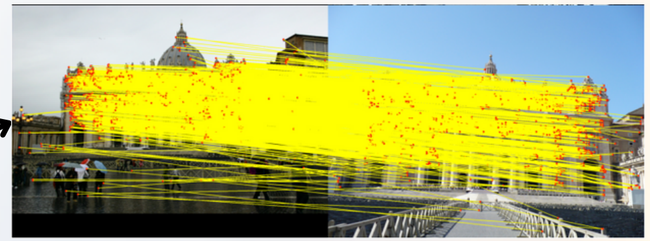
1. **Extract** the top 10,000 points
2. Use **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*) to group points into meaningful clusters based on spatial proximity
3. **Identify and remove outliers** (points that do not belong to any cluster).
4. Perform **semi-dense feature matching**



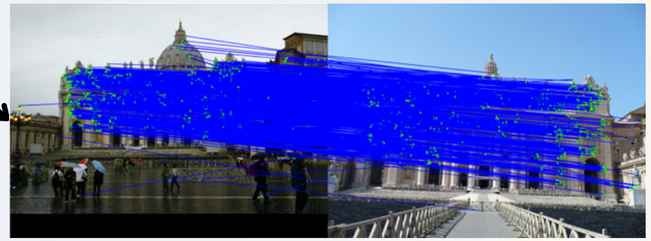
XFeat* clustering - results

Qualitative:

Original



XFeat* original
1293 points



XFeat* clustering
1126 points

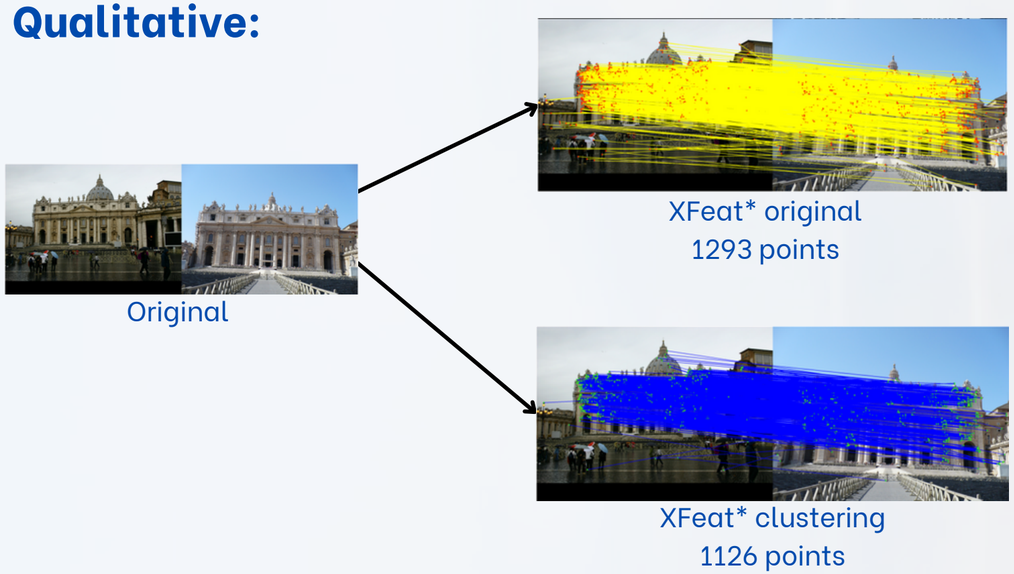
Quantitative:

Method	AUC@5°	AUC@10°	AUC@20°	ACC@10°	FPS
XFeat* original	50.2	65.4	77.1	85.1	**5.7±0.2**
XFeat* clustering	**50.7**	**66.0**	**77.6**	**85.9**	3.8 ±0.2

Table 7: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

XFeat* clustering - results

Qualitative:



Quantitative:

Method	AUC@5°	AUC@10°	AUC@20°	ACC@10°	FPS
XFeat* original	50.2	65.4	77.1	85.1	**5.7±0.2**
XFeat* clustering	**50.7**	**66.0**	**77.6**	**85.9**	3.8 ±0.2

Table 7: Relative camera pose estimation results for the Megadepth-1500 dataset, evaluated exclusively using a Mac M1 CPU.

Conclusion

- **XFeat Innovation:**
  - Lightweight CNN for efficient **keypoint detection** and local **feature extraction**.
  - Dual modes: **Sparse Matching (XFeat)** for speed and **Semi-Dense Matching (XFeat)*** for precision.
- **Key Contributions:**
  - Balances **computational efficiency** and matching accuracy.
  - Enhanced performance through innovative architectural design and optimization.
- **Our Improvements:**
  - **Homography Transformations:** Search to increase the quality of keypoints.
  - **Point Refinement**: Applied RANSAC-based filtering for robust feature matching.
  - **Clustering with DBSCAN**: Effectively removed outliers, boosting semi-dense match reliability.