Pixel-Accurate Epipolar Guided Matching

Oleksii Nasypanyi[1], Francois Rameau[2]
[1]Stony Brook University,USA [2]SUNY Korea, Incheon, Korea
oleksii.nasypanyi@stonybrook.edu francois.rameau@sunykorea.ac.kr

Abstract

Keypoint matching can be slow and unreliable in challenging conditions such as repetitive textures or wide-baseline views. In such cases, known geometric relations (e.g., the fundamental matrix) can be used to restrict potential correspondences to a narrow epipolar envelope, thereby reducing the search space and improving robustness. These epipolar-guided matching approaches have proved effective in tasks such as SfM; however, most rely on coarse spatial binning, which introduces approximation errors, requires costly post-processing, and may miss valid correspondences. We address these limitations with an exact formulation that performs candidate selection directly in angular space. In our approach, each keypoint is assigned a tolerance circle which, when viewed from the epipole, defines an angular interval. Matching then becomes a 1D angular interval query, solved efficiently in logarithmic time with a segment tree. This guarantees pixel-level tolerance, supports per-keypoint control, and removes unnecessary descriptor comparisons. Extensive evaluation on ETH3D demonstrates noticeable speedups over existing approaches while recovering exact correspondence sets. The project page is available here.

1. Introduction

Matching keypoints between image pairs is an essential component in 3D reconstruction pipelines, such as Structure-from-Motion (SfM) [31], and Simultaneous Localization and Mapping (SLAM) [21]. These tasks are widely used in applications ranging from robotic navigation [2], augmented reality [23], and implicit representation like Gaussian Splatting [15]. Traditional matching pipelines extract and compare local descriptors to identify tentative correspondences, followed by robust geometric verification to filter outliers [10]. While these approaches can operate without prior geometric knowledge, they become computationally prohibitive and unreliable in challenging scenarios involving repetitive structures, textureless regions, or wide-baseline viewpoints. When geometric relationships between views are known or can be reliably estimated, they provide powerful constraints that can significantly improve matching efficiency, density, and robustness [3, 29]. The most well-established example is rectified stereo vision, where calibrated cameras reduce correspondence search to efficient 1D scanline matching [24].

In the general case of unrectified images with known relative pose or fundamental matrix, one can similarly constrain the search to an *epipolar envelope* — a narrow band around the epipolar line. This idea, known as epipolar-guided matching, offers the same potential benefits as rectified stereo matching, but is harder to implement efficiently due to the arbitrary orientation and location of epipolar lines in the image. For that reason, existing methods attempt to approximate the epipolar envelope using angular bins, regular grids, or hashing schemes [3, 29]. While these strategies help reduce complexity, they suffer from two main limitations: they often require additional geometric checks to correct for coarse approximations, and they lack precise control over the envelope width in pixel units. As a result, they may either miss valid candidates or include spurious ones, which weakens both robustness and efficiency.

To cope with these limitations, we revisit the problem of epipolar-guided matching with a new formulation guaranteeing an exact and fast point retrieval within an epipolar envelope. For this task at hand, rather than computing the pixel-level distance between each keypoint in the second image and the epipolar line, we associate each of these keypoints with a tolerance circle (with a diameter equivalent to the desired epipolar envelope width). Viewed from the epipole, each circle defines an angular interval. A keypoint is retained as a match candidate if its angular interval contains the direction of the epipolar line defined by the query point. This reformulation avoids spatial discretization and enables efficient candidate selection in logarithmic time using a flat segment tree [9] (see Figure 1). Beyond avoiding approximation, our method is also faster, as it directly returns only the set of valid keypoints within the epipolar envelope, reducing unnecessary descriptor comparisons or additional geometric checks to eliminate false positive candidates (see Figure 2). Moreover, it offers greater flexibility,

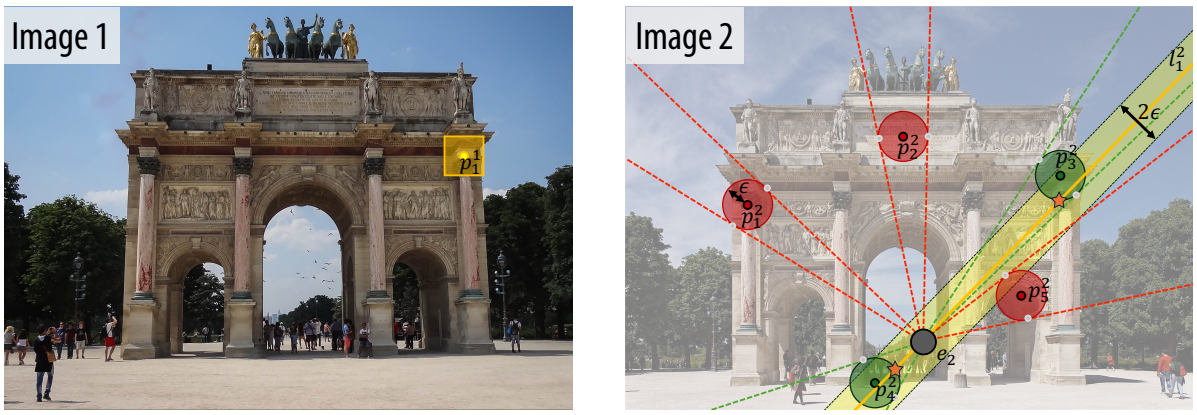

Figure 1. Overview of the guided matching process. Given a query point in image 1 $\mathbf{p}_1^1$, its corresponding epipolar line in image 2 $\mathbf{l}_2^1$ (yellow) restricts the search space to a narrow envelope of width 2ϵ. Any keypoint lying within this region (e.g., $\mathbf{p}_4^2$ and $\mathbf{p}_3^2$) is considered a valid candidate for matching. This can be determined by computing the orthogonal distance from each keypoint to the epipolar line, but such brute-force checks are computationally expensive. Instead, we draw circles of diameter 2ϵ around each keypoint in image 2 and test whether the epipolar line intersects any of them (orange stars). Interestingly, this geometric test can be reformulated as a fast 1D angular range query: for each circle, the angle interval formed by the two tangents from the epipole defines a valid angular interval, and a candidate is retained if the epipolar line direction lies within it. Green points indicate inliers, red points are outliers.

allowing per-keypoint tolerance control at pixel-level precision. Our strategy has been extensively evaluated against existing approaches [3, 29] across a wide range of keypoint densities and epipolar tolerance values, using the ETH3D dataset [28]. The results highlight better scalability, exactness, and flexibility compared to prior methods.

To summarize, the main contributions of this work are as follows: (i) A novel epipolar-guided matching technique based on fast angular filtering, (ii) Extensive evaluation across varying keypoint densities and tolerance levels, (iii) Open-source implementation.

2. Related Work

Feature matching has long been a fundamental problem in computer vision, with applications in 3D reconstruction [31], localization [23], and SLAM [21] among many others. As sparse matching scaled to large datasets [1] and moved into real-time settings [8], the need for faster, more efficient pipelines became critical. Efforts to accelerate feature extraction led to faster detectors like FAST [25] and AGAST [17], and more efficient descriptors like SURF [4] and binary alternatives to SIFT [26]. Still, the descriptor matching stage remains a bottleneck, especially as the number of keypoints increases. Initial approaches relied on brute-force matching [18], where each descriptor from one image is compared to all descriptors from the other using a chosen distance metric. This method guarantees that each match corresponds to the true nearest neighbor in descriptor space, but it scales poorly with the number of descriptors. To overcome this limitation, Approximate Nearest Neighbor (ANN) methods like FLANN [19], LSH for binary descriptors [20], and Product Quantization [14] were introduced to accelerate matching by organizing or compressing descriptors for fast lookup. Some other works have reformulated the matching problem as a global assignment task using optimal transport [12], which minimizes the total cost of pairing features under assignment constraints. These developments have inspired a new generation of learned matching pipelines, such as SuperGlue [27] and LoFTR [30] that rely on attention mechanisms and transformers for matching, achieving strong performance at higher computational cost. Regardless of the specific strategy, most of these pipelines incorporate geometric information only at the final stage, using RANSAC [10] to estimate the fundamental or essential matrix and remove outliers. However, in many real-world settings, geometric information between views is already available or can be estimated with reasonable accuracy. This prior can significantly improve matching in terms of robustness, accuracy, and efficiency. For example, SLAM systems often use predicted poses from motion models [13] or inertial sensors [22] to restrict the correspondence search around projected landmarks [21]. Similarly, in stereo vision, calibrated and rectified cameras reduce the problem to a 1D search along image rows [7], which improves both speed and reliability. These cases show that geometry can do more than just verify matches after the fact, but it can actively guide the matching process itself. All the approaches using prior camera pose information for this purpose can be considered geometric guided matching strategies. Several prior works have proposed geometry-guided matching strategies conceptually related to ours. One of the earliest geometry-guided approaches is [29], which samples points at regular intervals along each epipolar line. For each sam-

ple, nearby keypoints are retrieved from a spatial grid, and a local ratio test is applied to reject ambiguous matches in the epipolar envelope. This technique reduces the search space compared to brute-force matching, but still relies on multiple range queries per point. It also introduces several heuristic parameters and requires additional geometric checks, since grid-based retrieval is only an approximation. To further reduce complexity, Barath et al. [3] introduced epipolar hashing, which partitions keypoints into angular bins based on epipolar line orientations for fast candidate lookup. However, this approach assumes angular wedges from the epipole approximate the epipolar envelope, which fails when epipoles lie inside images or near borders. This leads to missed valid matches and inclusion of false candidates, requiring additional Sampson distance checks. Moreover, the tolerance is tied to bin size rather than pixel units, and boundary effects between bins can cause brittle matching. These limitations reduce robustness and adaptability across different camera configurations. Our method avoids these issues by directly retrieving keypoints within a user-defined pixel tolerance around the epipolar line, without relying on coarse discretization or multi-stage heuristics. It outputs the exact candidate set, so no additional geometric verification is needed. As a result, our approach is faster in practice, even compared to techniques such as epipolar hashing that claim constant-time performance. Figure 2 illustrates the approximation used in existing works and their implications in terms of returned candidates. Recent learned approaches like Patch2Pix [32] and Structured Epipolar Matcher [6] incorporate epipolar constraints into neural networks for accuracy-focused refinement. However, these methods prioritize accuracy over efficiency with different computational constraints, making them not directly comparable to our fast geometric filtering approach.

3. Background and Notations

In this work, we consider the classical setup of two calibrated cameras with known extrinsic parameters, — although knowing only the fundamental matrix is sufficient for epipolar guided matching. As illustrated in Figure 3, let $\mathbf{p}_i^1 = (x_i^1, y_i^1)^\mathsf{T}$ and $\mathbf{p}_i^2 = (x_i^2, y_i^2)^\mathsf{T}$ denote the projections of the same 3D scene point onto image 1 and image 2, respectively, with homogeneous coordinates $\tilde{\mathbf{p}}_i^1 = (x_i^1, y_i^1, 1)^\mathsf{T}$ and $\tilde{\mathbf{p}}_i^2 = (x_i^2, y_i^2, 1)^\mathsf{T}$.

The relative pose $(\mathbf{R}^{12}, \mathbf{t}^{12})$ induces a geometric relationship between image points, captured by the fundamental matrix $\mathbf{F}^{12}$, which encodes the epipolar geometry:

$$\mathbf{F}^{12} = \left(\mathbf{K}^2\right)^{-\top} [\mathbf{t}^{12}] \times \mathbf{R}^{12} \left(\mathbf{K}^1\right)^{-1}, \tag{1}$$

where $\mathbf{K}^1$ and $\mathbf{K}^2$ are the intrinsic calibration matrices of cameras 1 and 2, respectively, and $[\mathbf{t}^{12}]_\times$ is the skew-symmetric matrix associated with $\mathbf{t}^{12}$. The fundamental matrix enforces the epipolar constraint:

$$\left(\tilde{\mathbf{p}}_i^2\right)^\top \mathbf{F}^{12} \tilde{\mathbf{p}}_i^1 = 0, \tag{2}$$

which must hold for all true correspondences $\mathbf{p}_i^1 \leftrightarrow \mathbf{p}_i^2$. The epipolar line $\mathbf{l}_i^2 = \mathbf{F}^{12}\tilde{\mathbf{p}}_i^1$ in image 2 defines the locus of potential matches for point $\mathbf{p}_i^1$. The epipolar geometry also involves the epipoles $\mathbf{e}^1 = (x_e^1, y_e^1)^\top$ and $\mathbf{e}^2$, which correspond to the projections of the camera centers onto the opposite image planes. Note that, all epipolar lines $\mathbf{l}_i^2$ pass through $\mathbf{e}^2$.

Guided Matching. The goal of the matching process is to find correct correspondences between two sets of keypoints, $\mathscr{P}^1 = \{\mathbf{p}_1^1, \ldots, \mathbf{p}_m^1\}$ in image 1 and $\mathscr{P}^2 = \{\mathbf{p}_1^2, \ldots, \mathbf{p}_n^2\}$ in image 2, based on the similarity between their associated descriptors $\mathscr{D}^1 = \{\mathbf{D}_1^1, \ldots, \mathbf{D}_m^1\}$ and $\mathscr{D}^2 = \{\mathbf{D}_1^2, \ldots, \mathbf{D}_n^2\}$. A common approach is to compare each descriptor $\mathbf{D}_i^1$ against all descriptors in $\mathscr{D}^2$, which becomes computationally expensive as m and n grow. In the context of guided matching, where the relative pose is known, the search can be restricted geometrically: for each point $\mathbf{p}_i^1$, only candidate keypoints near its corresponding epipolar line $\mathbf{l}_i^2$ are considered. However, due to image noise, discretization, and calibration errors, the epipolar constraint cannot be enforced exactly. We therefore define a pixel-level tolerance ϵ and introduce the epipolar envelope as the region surrounding $\mathbf{l}_i^2$. Thus, a point $\mathbf{p}_j^2 \in \mathscr{P}^2$ is considered a potential match for $\mathbf{p}_i^1$ if:

$$\mathrm{dist}(\mathbf{l}_i^2, \mathbf{p}_j^2) = \frac{|a_i x_j^2 + b_i y_j^2 + c_i|}{\sqrt{a_i^2 + b_i^2}} \leq \epsilon, \tag{3}$$

where $\mathbf{l}_i^2 = (a_i, b_i, c_i)^\top$ is the epipolar line induced by $\mathbf{p}_i^1$.

All such points form the candidate set:

$$\mathscr{C}_i^2 = \left\{\mathbf{p}_j^2 \in \mathscr{P}^2 \;\middle|\; \mathrm{dist}(\mathbf{l}_i^2, \mathbf{p}_j^2) \leq \epsilon\right\}. \tag{4}$$

Each descriptor $\mathbf{D}_i^1$ from image 1 is then compared only against descriptors $\mathbf{D}_j^2$ associated with $\mathscr{C}_i^2$. This guided strategy reduces computational cost and improves robustness in challenging or repetitive environments.

Nevertheless, computing $\mathscr{C}_i^2$ for all m query points by checking every keypoint in image 2 results in a brute-force complexity of $\mathcal{O}(mn)$. The next section discusses strategies to accelerate this candidate search along epipolar lines.

4. Methodology

The core challenge in epipolar-guided matching is to efficiently identify all keypoints in the second image that lie within a pixel tolerance ϵ of a given epipolar line. While brute-force approaches suffer from $\mathcal{O}(mn)$ complexity and approximate methods risk missing valid correspondences, we propose an exact and scalable alternative based on a geometric reformulation of the problem. Our approach starts

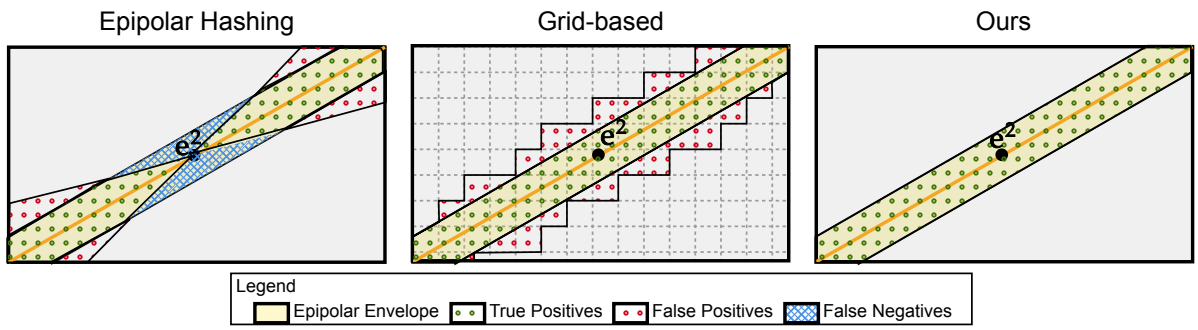

Figure 2. Comparison of three epipolar correspondence filtering strategies. Unlike our approach, Epipolar Hashing [3] and Grid-based [29] methods return matches outside the true epipolar region (false positives) and miss valid matches within it (false negatives).

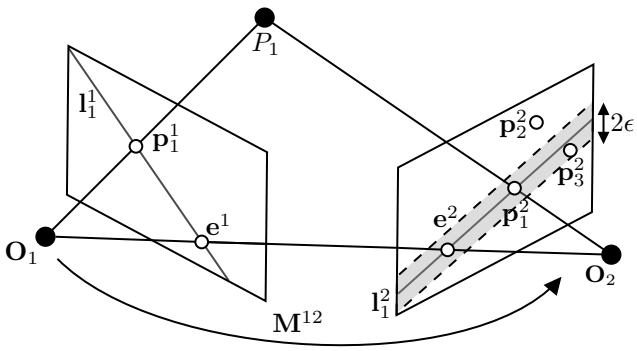

Figure 3. Epipolar geometry and envelope-based match filtering. Given a point $\mathbf{p}_1^1$ in the first image, its corresponding epipolar line $\mathbf{l}_1^2$ is shown in the second image. The grey band illustrates an epipolar envelope of width 2ϵ, used to retain only candidate matches such as $\mathbf{p}_1^2$ and $\mathbf{p}_3^2$.

by revisiting the epipolar constraint from an angular perspective, where each keypoint in the second image defines a tolerance region that can be expressed as a 1D angular interval (Sec. 4.1). This enables casting candidate search as a fast angular query using a segment tree data structure (Sec. 4.2). We then show how this geometric filtering integrates with standard descriptor-based strategies, such as Lowe's ratio [16] test or GMS [5] filtering (Sec. 4.3). Finally, we address implementation considerations, including angular discontinuities and near-epipole cases, to ensure correctness and robustness across configurations (Sec. 4.4).

4.1. Problem Reformulation

Given a query point $\mathbf{p}_i^1$ in the first image, the corresponding epipolar line $\mathbf{l}_i^2$ in the second image defines a constraint where any valid match must lie within distance ϵ of this line. Rather than computing the perpendicular distance from each keypoint $\mathbf{p}_j^2$ to $\mathbf{l}_i^2$, we reformulate this constraint using 1D angular intervals viewed from the epipole. The key geometric insight transforms the question from "which points are close to the epipolar line?" to "which epipolar line directions pass close to each point?". To do so, for any keypoint $\mathbf{p}_j^2$ in the second image, we define a tolerance circle of radius ϵ centered at $\mathbf{p}_j^2$. An epipolar line intersects this circle if and only if the corresponding point $\mathbf{p}_i^1$ represents a potential match under the distance tolerance ϵ. Viewed from the epipole $\mathbf{e}^2$, each tolerance circle subtends an angular interval. Any epipolar line whose direction falls within this interval will intersect the circle, indicating a valid candidate match. This reformulation transforms the matching problem into a 1D angular range query: given an epipolar line direction α, we seek all keypoints whose angular intervals Θ contain α. Figure 4 and Algorithm 1 summarize the main steps of the angular filtering and range query process.

4.2. Angular Range Query via Segment Tree

Each keypoint $\mathbf{p}_j^2 = (x_j, y_j)^\top$ in image 2 is associated with an angular interval that captures its visibility from the epipole $\mathbf{e}^2 = (x_e, y_e)^\top$, under a pixel tolerance ϵ. This interval corresponds to the directions from the epipole to the two tangents of a circle of radius ϵ centered at the keypoint, as illustrated in Figure 4. We first compute the angle from the epipole to the keypoint:

$$\theta_j = \operatorname{atan2}(y_j - y_e,\ x_j - x_e), \tag{5}$$

and the angular radius of the circle seen from the epipole is

$$\delta_j = \arcsin\left(\frac{\epsilon}{\|\mathbf{p}_j^2 - \mathbf{e}^2\|}\right). \tag{6}$$

The resulting angular interval is defined as $\Theta_j = [\theta_j - \delta_j,\ \theta_j + \delta_j]$. Once computed, all angular intervals Θ_j are stored in a flat segment tree structure [9], which supports efficient intersection queries.

For each query point $\mathbf{p}_i^1$ in image 1, we compute the direction α_i of its corresponding epipolar line $\mathbf{l}_i^2$ in image 2. This direction is measured as the angle from the epipole $\mathbf{e}^2$ to a reference point $\mathbf{q}_i$ on the epipolar line, empirically chosen as the point on the line closest to the image center:

$$\alpha_i = \operatorname{atan2}(y_{q_i} - y_e,\ x_{q_i} - x_e). \tag{7}$$

Note that while angles span $[0, 2\pi)$, epipolar lines are undirected: directions α and $\alpha + \pi$ are equivalent. We therefore

work in the reduced domain $[0, \pi)$, mapping all angles via $\alpha' = \alpha \bmod \pi$.

The segment tree is then queried to retrieve all keypoints whose angular intervals Θ_j contain α_i: $\alpha_i \in \Theta_j$. This filtering step outputs a candidate set $\mathscr{C}_i^2$ of matches for $\mathbf{p}_i^1$, with complexity $\mathcal{O}(\log n + k)$, where k is the number of retrieved candidates. Unlike other methods, our approach does not require computing the epipolar distance during querying, leading to faster performance while guaranteeing pixel-level precision and per-point tolerance control, without any approximation. A complete analysis of time and space complexity for all operations is provided in Table 1, with the full algorithmic procedure detailed in Algorithm 1. Note that, in practice, $k \gg \log n$, so the total matching complexity becomes $O(\bar{k}m)$, with $\bar{k}$ the average number of points returned per query. Unlike other methods, this time is spent on simple Boolean filtering rather than computing geometric distances.

Algorithm 1 Epipolar-Guided Matching

Require: $\mathcal{P}_1, \mathcal{P}_2, F_{12}$, tolerance $\epsilon > 0$
Ensure: Matches $\mathcal{M}$

Preprocess:
$e_2 \leftarrow \text{null}(F_{12}^\top)$; $\quad \mathcal{I} \leftarrow \varnothing$
for $p_j^2 = (x_j, y_j)^\top \in \mathcal{P}_2$ **do**
  if $\|p_j^2 - e_2\| \leq \epsilon$ **then**
    add $[0, \pi]$ to $\mathcal{I}$
  else
    $\theta_j \leftarrow \text{atan2}(y_j - y_e, x_j - x_e)$
    $\delta_j \leftarrow \arcsin(\epsilon / \|p_j^2 - e_2\|)$
    constrain θ_j to $[0, \pi]$ range
    add interval $[\theta_j - \delta_j,\ \theta_j + \delta_j]$ to $\mathcal{I}$ ▷ split at $0/\pi$ if it wraps
  end if
end for
build balanced segment tree $\mathcal{T}$ over $\mathcal{I}$ ▷ up to $2n$ intervals
Match:
for $p_i^1 \in \mathcal{P}_1$ **do**
  $l_2^i \leftarrow F_{12}\tilde{p}_1^i$; $\alpha_i \leftarrow \text{AngleFromEpipole}(e_2, l_2^i)$
  constrain α_i to $[0, \pi]$ range
  $\mathcal{C}_i \leftarrow \text{Query}(\mathcal{T}, \alpha_i)$ ▷ expected $O(\log(n) + k)$
  select best by descriptor; validate; add to $\mathcal{M}$
end forreturn $\mathcal{M}$

4.3. Descriptor Matching and Integration

Once geometric filtering has narrowed down the set of potential matches to $\mathcal{C}_i^2 \subset \mathcal{P}^2$, descriptor comparison is restricted to this subset. Specifically, the descriptor $\mathbf{D}_i^1$ of point $\mathbf{p}_i^1$ is matched only against $\mathbf{D}_j^2$ for which $\mathbf{p}_j^2 \in \mathcal{C}_i^2$. This matching strategy reduces computational cost and improves robustness by avoiding comparisons with unrelated descriptors. Our method is compatible with standard matching strategies, including simple nearest-neighbor matching and the classical Lowe's ratio test [16]. However, guided matching introduces a challenge for traditional ratio-based filtering: the Lowe ratio test relies on comparing the best and second-best matches across the entire descriptor space to detect ambiguous correspondences. In our case, the candidate set $\mathcal{C}_i^2$ may contain relatively few keypoints, potentially making the ratio test less discriminative. To address this limitation, Barath et al. [3] propose adaptive ratio thresholds that adjust based on the size of the candidate set, though this approach requires careful parameter tuning and may not generalize across different scenes. As an alternative, GMS [5] is well-suited to guided matching scenarios. By enforcing local motion consistency across image regions without relying on global descriptor distributions, GMS remains effective even when the candidate set is limited.

Table 1. Computational Complexity Analysis

Operation	Time Complexity	Space Complexity
Angular interval computation	$O(n)$	$O(n)$
Segment tree construction	$O(n \log(n))$	$O(n)$
Single epipolar query	$O(\log(n) + k)$	$O(1)$
Total matching (all queries)	$O(m(\log(n) + \bar{k}))$	$O(n)$

m: number of keypoints in image 1, n: number of keypoints in image 2, k: number of candidates per query, $\bar{k}$: average candidates per query

4.4. Implementation Details and Special Cases

While the proposed angular filtering method is general and efficient, specific geometric configurations require dedicated handling.

Segment Tree Structure We use a centered segment tree, adapted to the circular domain. The tree is built recursively, at each node we choose a split angle c — the median of all midpoints of current intervals. The node stores a bucket of all intervals whose span contains c, together with the minimum start and maximum end angles among those intervals. Intervals lying entirely to the left ($end < c$) and right ($start > c$) form the left and right child nodes, respectively. During the querying angle α, at each visited node, if α lies between the bucket's min-max bounds, we scan the bucket and return intervals that contain α.

Discontinuity at $0/\pi$ When the angular interval $\Theta_j = [\theta_j - \delta_j,\ \theta_j + \delta_j]$ crosses the $[0, \pi]$ boundary, we split it into two valid sub-intervals:

$$\Theta_j^{(1)} = [\theta_j - \delta_j, \pi], \quad \Theta_j^{(2)} = [0, \theta_j + \delta_j],$$

so that both parts can be correctly indexed and queried within the segment tree, this duplication increases the number of intervals stored in the data structure, which may raise the worst-case query complexity to $\mathcal{O}(\log 2n + k)$. However, this has a negligible impact on runtime performance.

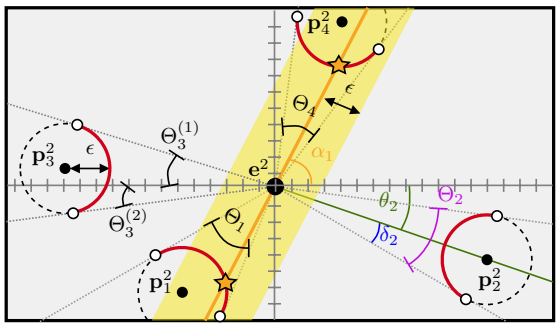

Figure 4. Illustration of the angular interval query strategy. Given an epipolar line $\mathbf{l}_1^2$ (orange), inlier points lie within the epipolar envelope (yellow). All angular intervals Θ_j that contain the epipolar line angle α_1 correspond to valid match candidates, forming the set $\mathscr{C}_1^2$. In this example, the points $\mathbf{p}_1^2$ and $\mathbf{p}_4^2$ have intervals Θ_1 and Θ_4 that include α_1, and are therefore considered inliers. For the point $\mathbf{p}_2^2$, the construction of its interval Θ_2 is illustrated via its central direction θ_2 and angular radius δ_2. The point $\mathbf{p}_3^2$ exemplifies a wrap-around case, where the interval spans the $0/\pi$ boundary. It is therefore split into two sub-intervals $\Theta_3^{(1)}$ and $\Theta_3^{(2)}$ in the segment tree.

Keypoints near the epipole If the distance between a keypoint $\mathbf{p}_j^2$ and the epipole $\mathbf{e}^2$ is less than ϵ, the angular radius δ_j becomes undefined. In this case $\mathbf{p}_j^2$ is treated as visible in all directions, with interval $[0, \pi]$, and is always returned during queries since any epipolar line intersects its tolerance region

5. Experimental Results

5.1. Dataset and Experimental Setup

Our method is assessed on the ETH3D dataset [28], which consists of 13 high-resolution sequences totaling several hundred images captured in diverse indoor and outdoor environments. ETH3D provides high-resolution images up to 6048×4032 pixels, dense depth maps, and accurate ground-truth poses, covering scenes with varying levels of texture. Aside from these considerations, ETH3D is also well-suited due to its diverse camera motions. For comparison, autonomous driving datasets such as KITTI [11] are largely dominated by forward motion, which produces an in-image epipole and would favor our approach.

Hardware Setup All experiments are conducted on a desktop computer running Ubuntu 24.04 LTS with an AMD Ryzen 7 7700X processor and 32GB RAM. To ensure fair comparison, all baseline methods are reimplemented in C++ with identical preprocessing and parallelization.

Baselines We compare against four prior methods: Brute-Force matching, FLANN [19], Epipolar Hashing [3], and Grid-Guided Matching [29]. All hyperparameters are set as specified in the original works, unless stated otherwise. All methods use identical descriptor filtering strategies such as Lowe's ratio test ($\tau = 0.8$), adaptive Lowe ratio [3], or GMS [5] (threshold fixed to 6 neighbors) for a fair comparison. GMS is used by default unless otherwise specified. Note that for Epipolar Hashing and Grid-Guided Matching, we compute line–point distances instead of the Sampson distance for consistency across methods. Also, to ensure a fair comparison, descriptor-only techniques undergo a final epipolar filtering step that removes matches outside the epipolar envelope, allowing us to incorporate the known geometry.

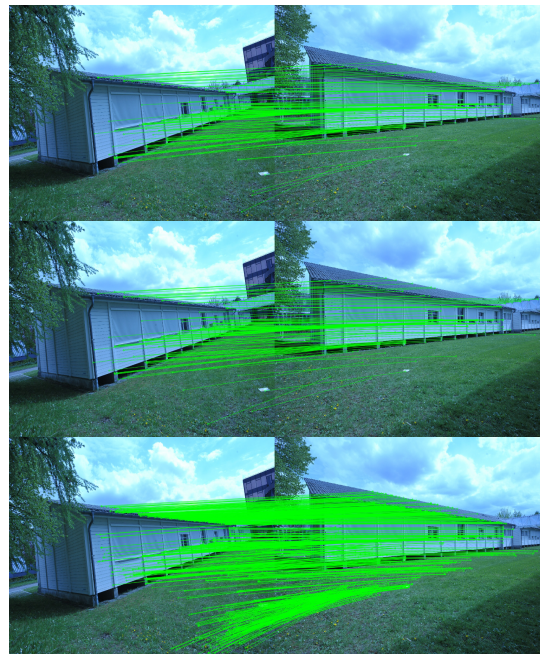

Figure 5. Comparison of the number of correctly matched points in a challenging scene with many repetitive structures, such as grass, rooftops, and building facades. (Top) Brute-Force (BF) matching. (Middle) FLANN-based matching. (Bottom) Our epipolar-guided matching. A match is considered correct if the 3D distance between correspondences is below 0.1 meters.

Ground-Truth Correspondences We use SIFT [16] for keypoint detection and description with identical parameters across all methods. As ETH3D provides sparse and unevenly distributed triangulated points, we generate denser ground-truth correspondences by associating each detected keypoint with the closest projected 3D point (within 5 pixels) from the provided cleaned scans and camera poses, discarding those without a nearby projection or falling in the occlusion mask. A match is considered correct if the 3D distance between the associated points is below 0.1 m (corresponding to less than 1% of the depth uncertainty even for close objects). Unless stated otherwise, we use the top 50,000 keypoints ranked by response. Only image pairs with at least 500 triangulated keypoints in the dataset ground truth are considered matchable and used for our

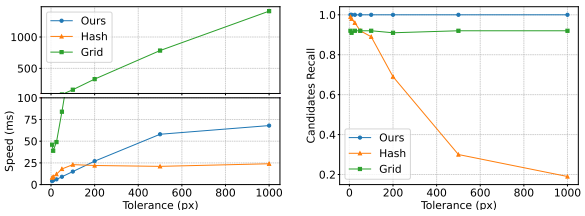

Figure 6. Impact of epipolar tolerance on candidate generation with 25,000 keypoints per image. (Left) Candidate generation execution time. (Right) Candidate set recall.

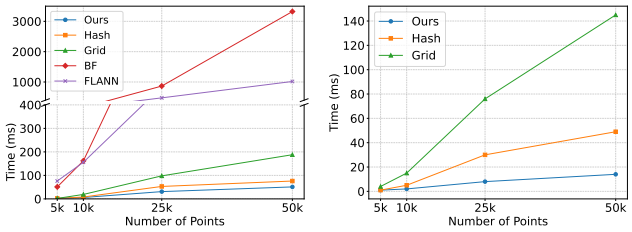

Figure 7. Impact of the number of keypoints on execution latency. (Left) Total matching time. (Right) Candidate set generation time (descriptor matching time is not included).

evaluation. In the case, the final number of matched pairs is lower than 10 (as for "office" or "meadow" sequences), we lowered the threshold to 100 points.

Evaluation Metrics We report the following metrics: (i) Total execution time, measured from matching start to final output, excluding feature extraction. (ii) Candidate generation time and descriptor matching time, reported separately for geometry-guided methods to distinguish the cost of geometric filtering from descriptor comparison. (iii) Candidate Recall, the fraction of ground-truth points within the epipolar envelope retrieved by the candidate selection stage (not applicable to methods without geometric filtering). (iv) Matching Recall, the fraction of final matches that are correct according to the 3D ground-truth correspondences.

5.2. Overall Evaluation

To provide an overview of our method's performance, we evaluate it on all sequences of the ETH3D dataset. In this experiment, we assume perfect ground-truth camera poses and intrinsics, which allows us to use an epipolar envelope of 50 pixels, approximately corresponding to 1% of the image size. Qualitative results are available in Figure 5 and Table 2 reports, for each sequence, the median values over all its image pairs. In terms of speed, our method is systematically faster overall. Most of this advantage comes from reduced candidate retrieval time, thanks to the efficiency of our data structure. Interestingly, hashing achieves a lower descriptor matching time, which can be explained by its lower candidate recall: it misses many true candidates, thereby reducing the number of descriptor comparisons. The grid technique is consistently slower due to the large number of queries required to scan the epipolar line. It also shows slightly higher descriptor matching time, as it tends to include false positives that remain for descriptor comparison; these false positives are also reflected in the candidate recall values. Finally, descriptor-only techniques remain significantly slower, and this gap would be even greater with a smaller epipolar envelope. Regarding matching quality, geometry-guided approaches demonstrate superior recall compared to descriptor-only methods, consistent with prior work [3, 29]. While our approach and the grid-based technique exhibit similar recall, hashing sometimes appears to perform better. Hashing operates with implicitly smaller search regions due to its fixed angular binning constraints, not because it respects the specified pixel tolerance. This creates artificially favorable conditions that break down when poses are imperfect, or epipoles lie near image boundaries, as demonstrated in subsequent experiments in Sec . 5.3 and Sec . 5.5.

5.3. Impact of Epipolar Tolerance

We evaluated the influence of the epipolar tolerance on matching performance using the entire "courtyard"sequence, as it yields the highest number of detected keypoints. For consistency, we limited the number of keypoints to 25,000 per image. We report the *candidate generation execution time* and the *candidate set precision* as the main metrics, as these directly reflect the efficiency of the geometric filtering step and the quality of the resulting candidate set. The results are shown in Fig. 6. Note that our approach remains the most effective up to a tolerance of 200 px (roughly 4% of the image size). Beyond this point, Epipolar Hashing appears faster, but only because its epipolar bin size is fixed. As tolerance increases, its bins fail to cover the full requested epipolar envelope, which is clearly visible in the recall curve (the method cannot reliably retrieve all valid points). This lack of adaptability can be problematic when the camera pose is approximate and a larger tolerance is needed, as illustrated in Sec . 5.5. The grid-based technique does not suffer from this limitation but scales significantly worse with tolerance.

5.4. Scalability

We evaluated the scalability of all methods by conducting experiments with 5,000, 10,000, 25,000, and 50,000 keypoints on the "courtyard" sequence with epipolar threshold of 50 pixels. This scene was chosen because the majority of its images contain over 50,000 points due to highly textured surfaces. The results are summarized in Fig. 7. Our method is over 3 times faster than Hashing in candidate generation and about 70% faster overall, since descriptor comparison dominates the runtime and is comparable across methods.

Sequence	Latency (Cand. Gen. / Desc. Match.) ms.↓					Candidates Recall ↑			Matching Recall ↑					Matches Num.				
	BF	FLANN	Ours	Hash	Grid	Ours	Hash	Grid	BF	FLANN	Ours	Hash	Grid	BF	FLANN	Ours	Hash	Grid
courtyard	0/2961	0/957	**34/101**	76/87	145/106	**1.00**	0.94	0.91	0.19	0.18	**0.25**	0.24	**0.25**	4891	4661	28900	28917	28807
delivery	0/231	0/212	**2/7**	11/6	22/7	**1.00**	0.98	0.89	0.34	0.33	0.46	**0.47**	0.46	3704	3537	6691	6691	6677
electro	0/27	0/50	**2 / 2**	2 / 3	4 / 2	**1.00**	0.85	0.91	0.21	0.19	0.38	0.39	**0.40**	977	935	5880	5879	5901
facade	0/2808	0/856	**35/99**	73/83	362/107	**1.00**	0.89	0.93	0.25	0.22	**0.35**	**0.35**	0.34	20335	19752	38282	38350	38265
kicker	0/894	0/321	**7/14**	13/12	53/15	**1.00**	0.91	0.92	0.23	0.23	0.54	**0.55**	0.54	2589	2539	10296	10103	10269
meadow	0/1426	0/262	**5/16**	6/16	52 /15	**1.00**	0.93	0.89	0.07	0.07	**0.10**	**0.10**	**0.10**	1223	1177	8930	9082	9049
office	0/29	0/50	**2/2**	**2/2**	6/2	**1.00**	0.94	0.93	0.11	0.10	**0.21**	**0.21**	**0.21**	552	543	3012	3077	3011
pipes	0/204	0/203	**2/5**	8/5	20 / 6	**1.00**	0.98	0.91	0.26	0.24	0.36	**0.37**	0.36	593	524	6426	6425	6438
playground	0/1323	0/553	**13/34**	28/31	113/38	**1.00**	0.89	0.93	0.22	0.18	**0.34**	0.33	**0.34**	5976	5027	14321	14354	14292
relief	0/464	0/334	**3/7**	9/7	29/8	**1.00**	0.96	0.91	0.24	0.20	**0.48**	**0.48**	**0.48**	1817	1726	9156	9068	9090
relief_2	0/357	0/244	**3/7**	11/7	33/8	**1.00**	0.97	0.91	0.32	0.31	**0.57**	**0.57**	**0.57**	7815	7709	8089	8089	8154
terrains	0/1695	0/691	**13/46**	65/45	160/45	**1.00**	0.99	0.88	0.35	0.32	**0.67**	**0.67**	**0.67**	13333	12370	23203	23194	23185
terrace	0/2051	0/744	**45/146**	157/ 137	335/ 156	**1.00**	0.96	0.90	0.10	0.09	**0.24**	**0.24**	**0.24**	6850	6306	24587	24634	24690

Table 2. Median results across all ETH3D [28] scenes using GMS (Lowe's ratio results are reported in the supplementary material). Best results are highlighted in **bold**. For latency, results are compared based on the total time for candidate generation and descriptor matching.

5.5. Noise Sensitivity

Geometry-guided methods are inherently sensitive to the accuracy of the fundamental matrix. To evaluate their robustness, we introduced varying levels of noise to the relative pose used in Eq. 1. A noise level of 1 corresponds to a randomly perturbed rotation matrix with each axis perturbed within $[-1°, 1°]$, and a translation vector with each component sampled from $[-0.25, 0.25]$ meters — representative of common camera pose estimation errors. To better simulate roughly calibrated systems, we increased the epipolar threshold to 200 pixels. All experiments were conducted on the "relief" sequence, which exhibits a wider diversity of camera motions. The results are summarized in Fig. 8. At noise level 0, the Grid and Hash approaches outperform the proposed method, which we attribute to their smaller candidate sets (as discussed in Sec. 5.3). However, as the noise increases, the performance of the Hash method deteriorates rapidly, indicating limited robustness to pose uncertainty because its fixed bin discretization cannot adapt to the enlarged epipolar envelope, causing it to miss a growing fraction of valid correspondences that are located away from their epipolar lines. In contrast, the Grid-based and Ours maintain stable performance and match the BF baseline at a noise level of 3.3 - approximately corresponding to a translational error of 0.8m and a rotational error of $3°$.

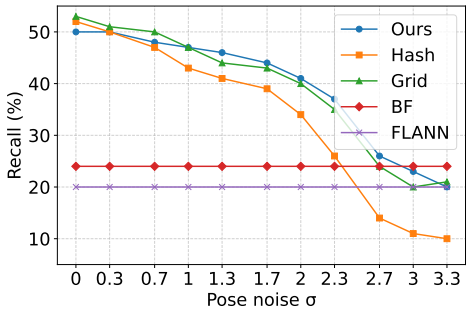

Figure 8. Matching recall under varying pose noise. Larger noise values correspond to greater perturbations in relative camera pose.

6. Conclusion & Future Work

We presented a pixel-accurate epipolar-guided matching method that avoids the discretization errors of existing approaches. By reformulating the problem as a 1D angular interval query from the epipole and solving it with a segment tree, our approach retrieves the exact set of candidates within a user-defined pixel tolerance in logarithmic time. This design offers fine-grained control at the keypoint level, eliminates unnecessary descriptor comparisons, and integrates seamlessly into standard matching pipelines.By supporting a per-point matching tolerance, our formulation allows adaptive matching strategies based on available priors, such as pose uncertainty. While not investigated in this paper, this represents a promising direction for future research. Experiments on ETH3D demonstrate that our method achieves consistent speedups over prior geometric-guided techniques while preserving all valid correspondences. While effective in practice, our method has two main limitations: it assumes a finite epipole (epipoles at infinity have not been witnessed during our experiments) and cannot reuse a segment tree for larger tolerances. Since rebuilding adds less than 10% to execution time, the overhead is minor. Future work will address these cases while also integrating our geometric filtering into learned matching approaches and extending the framework to handle three-view matching.

7. Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT)(IITP-2025-RS-2023-00259678).

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 2

[2] Basheer Al-Tawil, Thorsten Hempel, Ahmed Abdelrahman, and Ayoub Al-Hamadi. A review of visual slam for robotics: Evolution, properties, and future applications. *Frontiers in Robotics and AI*, 11:1347985, 2024. 1

[3] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14546–14555, 2021. 1, 2, 3, 4, 5, 6, 7

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 2

[5] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4181–4190, 2017. 4, 5, 6

[6] Jiahao Chang, Jiahuan Yu, and Tianzhu Zhang. Structured epipolar matcher for local feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6177–6186, 2023. 3

[7] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018. 2

[8] Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1403–1410. IEEE, 2003. 2

[9] Mark De Berg, Otfried Cheong, Marc Van Kreveld, and Mark Overmars. *Computational geometry: algorithms and applications*. Springer, 2008. 1, 4

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 6

[12] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 18(4):377–388, 2002. 2

[13] Hyowon Ha, François Rameau, and In So Kweon. 6-dof direct homography tracking with extended kalman filter. In *Image and Video Technology*, pages 447–460. Springer, 2015. 2

[14] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1): 117–128, 2010. 2

[15] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 1

[16] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 4, 5, 6

[17] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *European conference on Computer vision*, pages 183–196. Springer, 2010. 2

[18] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005. 2

[19] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009. 2, 6

[20] Marius Muja and David G Lowe. Fast matching of binary features. In *2012 Ninth conference on computer and robot vision*, pages 404–410. IEEE, 2012. 2

[21] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 1, 2

[22] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018. 2

[23] Francois Rameau, Oleksandr Bailo, Jinsun Park, Kyungdon Joo, and In So Kweon. Real-time multi-car localization and see-through system. *International Journal of Computer Vision*, 130(2):384–404, 2022. 1, 2

[24] Francois Rameau, Jinsun Park, Oleksandr Bailo, and In So Kweon. Mc-calib: A generic and robust calibration toolbox for multi-camera systems. *Computer Vision and Image Understanding*, 217:103353, 2022. 1

[25] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006. 2

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2

[27] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2

[28] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 8

[29] Rajvi Shah, Vanshika Srivastava, and PJ Narayanan. Geometry-aware feature matching for structure from motion

applications. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 278–285. IEEE, 2015. 1, 2, 4, 6, 7

[30] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2

[31] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 1, 2

[32] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4669–4678, 2021. 3