Camera Pose Voting for Large-Scale Image-Based Localization

Bernhard Zeisl Torsten Sattler Marc Pollefeys
Department of Computer Science, ETH Zurich, Switzerland
{bernhard.zeisl,torsten.sattler,marc.pollefeys}@inf.ethz.ch

Abstract

Image-based localization approaches aim to determine the camera pose from which an image was taken. Finding correct 2D-3D correspondences between query image features and 3D points in the scene model becomes harder as the size of the model increases. Current state-of-the-art methods therefore combine elaborate matching schemes with camera pose estimation techniques that are able to handle large fractions of wrong matches. In this work we study the benefits and limitations of spatial verification compared to appearance-based filtering. We propose a voting-based pose estimation strategy that exhibits $\mathcal{O}(n)$ complexity in the number of matches and thus facilitates to consider much more matches than previous approaches – whose complexity grows at least quadratically. This new outlier rejection formulation enables us to evaluate pose estimation for 1-to-many matches and to surpass the state-of-the-art. At the same time, we show that using more matches does not automatically lead to a better performance.

1. Introduction

Estimating the camera pose from which a given image was taken is a fundamental problem for many interesting applications such as navigation of autonomous vehicles, Augmented Reality, and (incremental) Structure-from-Motion (SfM). Given a 3D model of the scene, the camera pose can be computed from 2D-3D matches between 2D measurements in the image and 3D points in the model by applying an n-point-pose solver [3] inside a RANSAC loop. Since RANSAC's run-time increases exponentially with the percentage of false matches, it is crucial to avoid accepting too many wrong matches. At the same time, distinguishing between correct and incorrect correspondences is an ill-posed problem for large datasets, as they contain many points with (nearly) identical local appearance. This is especially true for urban scenes which often possess repetitive structures. Consequently, Lowe's widely used ratio test [17] is too restrictive and thus often fails in these cases. Many current state-of-the-art localization algorithms therefore use rather complicated matching procedures that combine 2D-to-3D and 3D-to-2D search with filtering based on co-visibility information [7, 16, 22]. However, geometric verification is still only performed on a final, small set of correspondences.

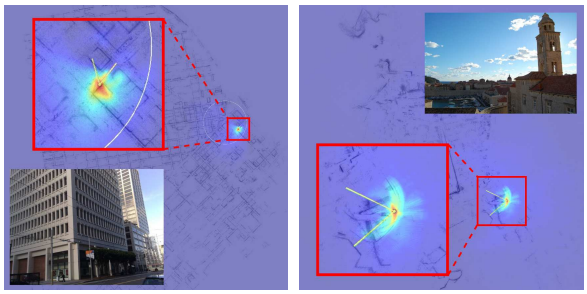


Figure 1: Given n 2D-3D matches, our approach makes extensive use of geometric filtering and votes for a 4 DoF camera pose (translation and rotation around the gravity direction) in $\mathcal{O}(n)$ while naturally integrating location priors if available (white circle). The heatmaps encode the number of geometrical correct matches for 2D positions.

In this work we propose to shift the task of finding correct correspondences from the matching stage to the pose estimation step, by leveraging geometric cues extensively, which are local and thus independent of the model size. First, instead of using 1st nearest neighbors and only retaining matches that are likely to be inliers, we simplify the matching problem and consider 1-to-many correspondences. This results in a large number of matches with a very small inlier ratio. Second, we aim to perform extensive spatial verification early on in the pose estimation procedure. As such the core questions we tackle is: *How can we make geometric verification scalable to thousands of tentative correspondences and what can we expect to gain from it?* To that end, we introduce a voting-based spatial verification process that exploits a known gravity direction and an approximate knowledge of the camera height using a setup similar to [27]. Exemplary results of our voting procedure are illustrated in Fig. 1. Our contributions are as follows:

– We formulate spatial verification as a Hough voting problem in pose parameter space, obtaining a run-time that grows only *linearly* in the number of matches.
– We show that we can detect a large fraction of wrong matches using simple but efficient filtering operations based on local (image) geometry.

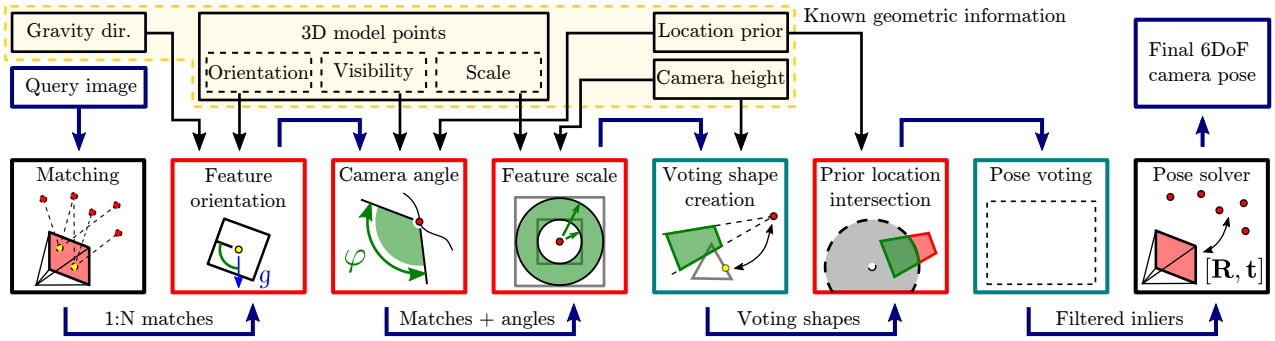


Figure 2: Overview of our linear filtering and location voting process (blue path) extensively utilizing spatial verification based on (known) properties. The cyan colored boxes denote steps of the voting procedure and are discussed in Sec. 2 and 3. Boxes marked in red correspond to the proposed filtering steps based on geometric constraints, which are explained in Sec 4.

– Our approach naturally integrates and profits from pose priors, *e.g.*, from GPS data, inertial measurements or vanishing point information, when those are available.
– Our formulation yields a multi-modal distribution over possible camera poses without any additional cost and thus is well suited to handle repetitive scenes.
– We study the applicability of different matching strategies and the influence of allowing 1-to-many matches.

The resulting method localizes considerably more images than current state-of-the-art techniques [1,6,16,28] on large scale datasets and processes tens of thousands of matches with an inlier ratio below 1% in a few seconds – which is well beyond what current methods [16, 27] can handle. Interestingly, while our results demonstrate that geometric constraints are well suited for outlier filtering, they also clearly indicate that simply using more matches does not automatically lead to a better localization. Thus, one intention of this paper is to stimulate further research on defining the quality of matches and how to find good correspondences.

The rest of the paper is structured as follows. The remainder of this section discusses related work. Sec. 2 outlines our voting method, while Sec. 3 explains the computation of spatial votes from matches. Sec. 4 shows how to exploit local geometric constraints to filter wrong matches. Finally, Sec. 5 discusses our experimental evaluation. Additional material is available from our project website `www.cvg.ethz.ch/research/location-voting`.

Related Work: There exist two possible approaches to obtain the 2D-3D matches need for pose estimation. Methods based on *direct matching* perform approximate nearest neighbor search in descriptor space and apply Lowe's ratio test [17] for outlier filtering. While 2D-to-3D search is inherently more reliable than 3D-to-2D matching [21], state-of-the-art approaches use the latter to recover correspondences missed or rejected during the former [7,16,22]. This enables them to better counter the problem that the ratio test rejects more and more correct matches for larger datasets due to the increased descriptor space density [16, 23]. Recently, alternatives [16, 27] to aggressive outlier filtering during the matching stage have been proposed. These works are most related to our approach, as they can handle significantly lower inlier ratios. Li *et al*. [16] use covisibility information to guide RANSAC's sampling process, enabling them to avoid generating obviously wrong camera pose hypotheses. Following a setup equivalent to ours, Svärm *et al*. [27] derive a deterministic outlier rejection scheme based on a 2D registration problem. The runtime of their method is $\mathcal{O}(n^2 \log n)$, where n is the number of matches, which severely limits the number of correspondences that can be processed in reasonable time. In contrast, the method proposed in this paper runs in time $\mathcal{O}(n)$, enabling us to solve significantly larger matching problems.

Location recognition and *indirect localization* methods apply image retrieval techniques [4, 8, 19, 26] to restrict correspondence search to the 3D points visible in a shortlist of retrieved database images. In order to improve the retrieval performance, [13, 24] remove confusing features, [28] explicitly handle repetitive structures, and [11] generates synthetic views to increase the robustness to viewpoint changes. Most relevant to this paper are the methods from [1,6,28,29]. Chen *et al*. [6] show how to exploit GPS information and viewpoint normalization to boost the retrieval performance. Similar to us, Zamir *et al*. [29] consider multiple nearest neighbors as potential matches, while [1] adapt Hamming embedding to account for varying descriptor distinctiveness. We show that our approach achieves superior localization while providing the full camera pose.

Finally, Quennesson *et al*. [20] find a density of camera viewpoints from high level visibility constraints in a voting like procedure. Similar to us, Baatz *et al*. [2] verify geometric consistency early on in their voting for view directions.

2. Voting-Based Camera Pose Estimation

In this paper we relax the matching filter and aim to exploit geometric cues instead. To handle the massive amount of outliers there exists the need for a fast and scalable outlier filter. To this end, we borrow a setup from Svärm *et al*. [27] which facilitates geometric constraints on the camera (gravity direction and approximate height) and transform it to a voting procedure. In addition we augment the voting with

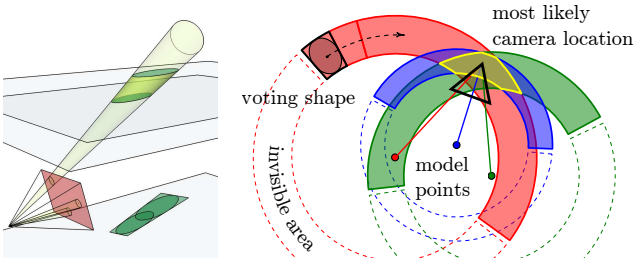


Figure 3: (Left) 2D error shape generation from a 3D reprojection error cone. (Right) Voting shapes are rotated around their 3D points, thus intersecting in the most likely camera location (visualized as marginalization over all rotations).

other filters utilizing global geometric constraints from the 3D model (feature orientations, visibility and scale of 3D points) and, if available, a positional prior for the camera location. An overview of our linear outlier filter is visualized in Fig. 2. It reduces the problem of finding a camera pose that maximizes the number of inliers to several 2D voting problems, one for each distinct camera orientation. In the following we will explain the voting procedure in more detail, while Sec. 4 covers the proposed geometric filters.

2.1. Pose Estimation as a Registration Problem

Given the camera gravity direction, we can define a rotation matrix $\mathbf{R}_g$ that transforms the local camera coordinate system into a coordinate frame which is gravity-aligned (w.l.o.g. we assume that the 3D scene model is gravity-aligned as well). This reduces the pose estimation problem from 6DoF to finding a rotation $\mathbf{R}_\varphi \in \mathbb{R}^{2\times 2}$ around the gravity direction and a translation $\mathbf{t} \in \mathbb{R}^3$. A 2D-3D match between a 3D point $\mathbf{X}$ and a 2D image observation $\mathbf{x}$ is defined to be an inlier if $\mathbf{X}$ is projected within r pixels next to $\mathbf{x}$. This is equivalent to the transformed 3D point falling into the 3D error cone

$$\mathbf{c}(\mathbf{x}, r) = \nu \cdot \mathbf{r}(\mathbf{x} + \mathbf{u}), \;\; \forall \mathbf{u} \in \mathbb{R}^2, \|\mathbf{u}\| = r, \nu \in \mathbb{R}_{\geq 0} \quad (1)$$

defined by the reprojection error r and $\mathbf{x}$. Here, $\mathbf{r}(\mathbf{x}) = \mathbf{R}_g \mathbf{K}^{-1} (\mathbf{x}\ 1)^\mathsf{T}$ is the viewing ray corresponding to $\mathbf{x}$ transformed into the gravity-aligned coordinate frame. We require that the intrinsic calibration $\mathbf{K}$ is known. Assuming that we know the height h of the camera above the ground plane, the problem of registering the 3D point with the cone simplifies to estimating a 2D translation $\mathbf{t}'$ such that

$$\begin{bmatrix} \mathbf{R}_\varphi & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X} + \begin{bmatrix} \mathbf{t}' \\ -h \end{bmatrix} \in \mathbf{c}(\mathbf{x}, r), \quad \mathbf{t} = [\mathbf{t}', -h]^\mathsf{T} \;. \quad (2)$$

As a result the registration problem gets further restricted to the conic section at offset $X_z - h$, *i.e.*, $\mathbf{c}_z(\mathbf{x}, r) = X_z - h$ and thus is fully described on a 2D plane.

Obviously we do not know the camera height exactly upfront. However, we can often approximate the ground plane by interpolating the positions of the cameras represented in the model. At the same time, the height of the query camera position is usually close to the ground plane within a certain interval, *e.g.* $\pm$5m. Centering the inverted error cone at the matching 3D point $\mathbf{X}$ and rotating it around gravity direction defines a space in which the camera has to lie (see the following Sec. 2.2 for an explanation of the inversion). Intersecting it with the ground plane thus allows us to estimate the height interval $[h_{min}, h_{max}]$ for the camera pose. This uncertainty in camera height corresponds to intersecting the error cone $\mathbf{c}(\mathbf{x}, r)$ by two horizontal planes. As shown in [27], we can project these capped error cones onto the ground plane and thus reduce the camera pose estimation to a 2D registration problem between projected error cones (Fig. 3 left) and projected 3D point positions.

Definition 1. *A 2D error shape for a given 2D-3D point correspondence is the union of all projected conic sections between the reprojection error cone* $\mathbf{c}(\mathbf{x}, r)$ *and heights in the interval* $[h_{min}, h_{max}]$.

Hence, the uncertainty in camera height is propagated to the camera location, reflected by the larger area covered by the error shape. In case the cone does not intersect the height interval, the correspondence is immediately invalidated.

2.2. Linear Time Pose Voting

So far the error shapes are defined in the local, gravity-aligned coordinate system of the *camera*. As such, the registration problem can also be imagined as translating and rotating the camera in 2D space and for each unique transformation (discretized in location and angle) count the number of projected 3D points that fall into their voting shape. This procedure is not optimal since the 2D space is unbounded and we would need to test an infinite number of translations.

Thus, we propose to view the problem from a different perspective and to transform the error shapes into the *global* coordinate system; *i.e.*, for a given correspondence we set the projected 3D point position as fixed and by this transform the uncertainty to the camera location. The locations of the transformed error shapes – called voting shapes in the following – thereby also depend on the orientation of the camera. We exploit this fact to design a linear time camera pose estimation algorithm (cf. Fig. 3 right): Iterating over a fixed set of rotations, each 2D-3D match casts a vote for the region contained in its voting shape. Accumulating these votes in several 2D voting spaces, one per camera orientation, thus enables us to treat every match individually. As a result we obtain a (scaled) probability distribution in the 3-dimensional pose parameter space. The best camera pose is then defined by the orientation and position that obtained most votes. The final 6DoF pose is computed with a 3 point solver inside a RANSAC loop on the voted inlier set. In case of similar structures in the scene, our voting creates a multi-modal distribution. We obtain its modes via non-maximum suppression and verify each of them separately, accepting the pose with most support.

The ideal voting space would be concentric wrt. each matching 3D point (cf. Fig. 3 right), but this complicates in-

tersection computation significantly. Instead we use a uniform sampling to guarantee $O(n)$ runtime. During voting we account for the quantization by conservatively considering each bin contained in, or intersected by a voting shape.

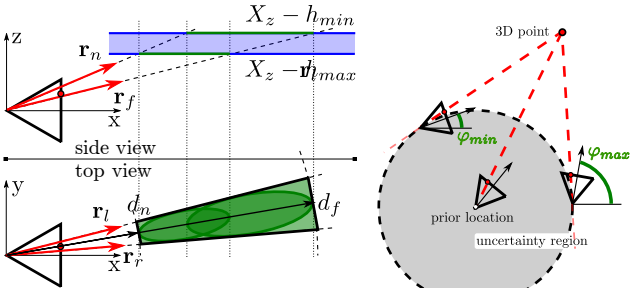
Proof of Coordinate System Transformation: Consider the error shape M of a match $m = (\mathbf{x}, \mathbf{X})$. Setting the reprojection error for this match to 0 is equivalent to adding uncertainty to the camera position (which is at the camera frame origin $\mathbf{0}$). With $\bar{\mathbf{M}}$ being the center of M, the error shape for the camera location is given by the Minkowski difference $\mathrm{M}_C(m) = \{\mathbf{0} - \mathbf{p} + \bar{\mathbf{M}} \mid \mathbf{p} \in \mathrm{M}\}$. If the match m is correct, the translation from the camera coordinate system to the global world coordinate system (both gravity- and thus axis-aligned) is given as $\mathbf{t}' = \mathbf{X}' - \bar{\mathbf{M}}$, where $\mathbf{X}'$ is the 2D position of the projected point $\mathbf{X}$. Therefore, the camera center in world coordinates has to fall into the global voting shape $\mathrm{V}(m) = \mathrm{M}_C + \mathbf{t}'$, which was obtained without altering the orientation of the camera. For a different orientation, we simply rotate the local camera coordinate system by an angle ϕ before performing a translation $\mathbf{t}'_\phi = \mathbf{X}' - \mathbf{R}_\phi \bar{\mathbf{M}}$. Hence, a rotated voting shape is obtained via

$$\mathrm{V}(m, \phi) = \mathbf{R}_\phi \mathrm{M}_C + \mathbf{t}'_\phi = \{\mathbf{X}' - \mathbf{R}_\phi \mathbf{p} \mid \mathbf{p} \in \mathrm{M}\} \ . \quad (3)$$

Eq. (3) reveals that changing the camera orientation results in the voting shape being rotated around the 2D position $\mathbf{X}'$ of the matching point (cf. Fig. 3 right). □

Time Complexity $\mathcal{O}(n)$: First, given a rotation angle, a *single* iteration over all n correspondences is sufficient to aggregate votes for the 2D camera location. Second, to obtain the full distribution and by this the best inlier set also wrt. a discriminative camera angle, the procedure needs to be performed separately for k discretized angles. Third, for a large variation in the camera height, the propagated uncertainty leads to less discriminative votes. To avoid this property we also quantize the considered height range into l smaller intervals and test for each of them. Consequently, the number of used angle-height pairs is constant, *i.e.*, our method performs $kl \cdot n$ iterations. The size of the voting shapes is bounded as well, meaning that we cast a constant number of votes for each shape (In principle a conic section can be unbounded, *e.g.*, a parabola; however, as will be introduced in Sec. 4, we leverage the feature scale to constrain its extent). As a result, our approach has an overall computational complexity of $\mathcal{O}(n)$. We will show later that the constant is significantly reduced by our filters, *e.g.*, on average only 8% of the camera orientations need to be tested and as few as 15% of the matches survive the geometric tests.

Implementation Details: Since the size of each voting shape can vary drastically, we use a hierarchical voting approach. For each shape, we select the level in the hierarchy such that all shapes cast at most a fixed number of votes (*e.g.*, for 100 bins). On the finest level, the size of each bin is 0.25m^2. For each level the 2D voting space is implemented as a hash-map (indexed via discretized camera locations) and due to its sparse structure not bounded in space. The height interval is typically $\pm$5m and discretized in 1m steps. For the angular resolution we chose 2° degrees.



(a) Voting shape construction (b) Camera orientations

Figure 4: (a) Voting shape construction: A quadrilateral is modeled by the rays $\mathbf{r}_{n|f}$ intersecting the height interval and the bounding rays $\mathbf{r}_{l|r}$. (b) The location prior constrains the camera rotation for a match to the interval $[\varphi_{min}, \varphi_{max}]$.

3. Efficient Voting Shape Computation

In the following, we present an efficient computation of the voting shapes and show how to account for the errors introduced by the voting space quantization and gravity direction inaccuracy. In Sec. 2.2 we pointed out that a voting and error shape only differ by a proper rigid transformation. Thus, we base our derivation on Def. 1 and approximate an error shape via its bounding quadrilateral (cf. [27]) for efficiency. The quadrilateral can be described via its near and far distance $d_{n|f}$ to the camera center and the two bounding rays $\mathbf{r}_{l|r}$, as illustrated in Fig. 4a. A quadrilateral for a particular camera orientation is then efficiently computed by rotating the projected rays $\mathbf{r}_{n|f}$ as derived in Eq. (3).

W.l.o.g. let us define that the projected optical axis points in the x direction in gravity aligned camera coordinates. The left- and rightmost rays have extremal y value; *i.e.*, we are looking for stationary points of the y-component of $\mathbf{c}(\mathbf{x}, r)$. The cone parameterization from Eq. (1) for $\nu = 1$ describes points on the image plane with reprojection error r. Therefore, $\mathbf{r}_{l|r}$ intersect the image plane at keypoint offsets

$$\begin{aligned} \mathbf{u}^*_{l|r} &= \arg\min_{\mathbf{u},\lambda} c_y(\mathbf{x}, r) + \frac{\lambda}{2}(\mathbf{u}^\mathsf{T}\mathbf{u} - r^2) \\ &= \mp r \begin{pmatrix} r_{21} & r_{22} \end{pmatrix}^\mathsf{T} \Big/ \left\| \begin{pmatrix} r_{21} & r_{22} \end{pmatrix}^\mathsf{T} \right\| \ , \end{aligned} \quad (4)$$

with $r_{ij} = \mathbf{R}_g(i, j)$. In a similar manner the offsets corresponding to the near and far rays are derived as

$$\mathbf{u}^*_{n|f} = \mp r \begin{pmatrix} r_{31} & r_{32} \end{pmatrix}^\mathsf{T} \Big/ \left\| \begin{pmatrix} r_{31} & r_{32} \end{pmatrix}^\mathsf{T} \right\| \ . \quad (5)$$

They are orthogonal to $\mathbf{u}^*_{l|r}$. As one would expected, all offsets are *independent* of the particular feature location $\mathbf{x}$. To account for the bounded heights, $\mathbf{r}_{n|f} = \mathbf{r}(\mathbf{x} + \mathbf{u}^*_{n|f})$ is intersected at heights $h_{n|f} = \{X_z - h_{max}, X_z - h_{min}\}$, resulting in the distances of the error shape to the camera, *i.e.*, $d_i = \|\mathbf{r}_{i_{x:y}} h_i / \mathbf{r}_{i_z}\|, \forall i \in \{n, f\}$. To account for the

discretization in angles, $\mathbf{r}_{l|r} = \mathbf{r}(\mathbf{x} + \mathbf{u}^*_{l|r})$ is rotated apart around the z-axis by half the angular resolution.

3.1. Accounting for Gravity Direction Uncertainty

The measurement of the camera gravity direction is likely to exhibit a certain amount of noise, which we want to account for during voting. The introduced uncertainty will lead to a roll and tilt of the camera and hence rotate a feature point ray and reprojection error cone. Therefore, he union of all conic sections of rotated cones now defines the error shape, which is again approximated by a quadrilateral. For a fixed gravity orientation the keypoint offsets $\mathbf{u}^*$ have been computed before. What remains is to derive the extremal image plane positions in dependence of the camera tilt and roll. We will mainly present the results of our derivation, while more details are found in the supplementary material.

All possible rays for a feature point $\mathbf{x}$ are given by $\tilde{\mathbf{r}}(\mathbf{x}, \mathbf{a}) = \mathbf{R}_\alpha(\mathbf{a})\,\mathbf{r}(\mathbf{x})$, where the rotation matrix is parameterized via the angle α and an axis $\mathbf{a}$ (which lies in the horizontal plane). First, for the near and far extremal position stationary points of the z-component of rays are of interest, such that the two extremal rotation axes are

$$\begin{aligned}\mathbf{a}^*_{n|f} &= \arg\min_{\mathbf{a},\lambda}\ \tilde{r}_z(\mathbf{a}) + \frac{\lambda}{2}(\mathbf{a}^\mathsf{T}\mathbf{a} - 1) \\ &= \mp\big(-r_y,\, r_x,\, 0\big)^\mathsf{T} \Big/ \big\|\big(-r_y,\, r_x\big)\big\| \quad . \end{aligned} \tag{6}$$

Second, for the left and right positions the optimization problem wrt. the extremal y-components of rays reads as $\mathbf{a}^*_{l|r} = \arg\min_{\mathbf{a},\lambda}\ \tilde{r}_y(\mathbf{a}) + \frac{\lambda}{2}\,(\mathbf{a}^\mathsf{T}\mathbf{a} - 1)$. It's derivative wrt. $\mathbf{a}$ forms a 2×2 linear system $A(\lambda)\mathbf{a} = \mathbf{b}$. Solving for $\mathbf{a}$ and evaluating the norm constraint on $\mathbf{a}$ results in a fourth order polynomial in λ. We compute its roots as the Eigenvalues $\lambda_{1\ldots4}$ of the 4×4 companion matrix. The two rotation axes are obtained by evaluating the original function wrt. its minimum and maximum value:

$$\mathbf{a}^*_{l|r} = \left\{ \arg\min_{\lambda_{1\ldots4}} \tilde{r}_y(\mathbf{a}(\lambda)),\ \arg\max_{\lambda_{1\ldots4}} \tilde{r}_y(\mathbf{a}(\lambda)) \right\} \quad \text{with} \quad \mathbf{a}(\lambda) = \mathbf{A}(\lambda)^{-1}\mathbf{b} \tag{7}$$

Compared to the case with fixed gravity direction, now the extrema positions are *dependent* on the feature position $\mathbf{x}$. This is intuitive, since the further a keypoint is located from the principal point, the more influence a camera tilt and roll will have. To account for the reprojection error, results from Eq. (4) and (5) are added and we obtain the extremal positions of a cone under gravity uncertainty as

$$\mathbf{c}(\mathbf{x}, \mathbf{u}^*, \mathbf{a}^*) = \tilde{\mathbf{r}}(\mathbf{x}, \mathbf{a}^*) + \mathbf{R}_\alpha(\mathbf{a}^*)\mathbf{R}_g \begin{pmatrix} \mathbf{u}^*/f \\ 0 \end{pmatrix} \quad . \tag{8}$$

4. Filtering Based On Geometry Constraints

In the following, we present a set of filters that can individually be applied to each match. They are based on geometric relations between properties of the 3D model and local descriptors and aim to reduce the total number of votes to cast. The advantages are twofold. First, the consideration of different camera orientations introduces a constant factor in the time complexity. By applying some simple filters we can decrease both the number of relevant matches and the constant factor and thus gain a considerable speedup. Second, eliminating false votes upfront boosts the recall rate of our method by up to 20% as will be shown in Sec 5.

Relative Feature Orientation: Usually, local descriptors are defined relative to a feature orientation. Similar to [2, 12], who use orientations to improve image retrieval, we can thus use the local feature orientations to reject matches. Given the known gravity direction, we express the query feature orientation in a fixed reference frame and compare it to the feature orientations from the database images. The latter typically form an interval of possible feature orientations. A match is rejected if the query orientation differs by more than a fixed threshold from the orientations in the interval belonging to the matching 3D point. Notice that this filtering step works similar to upright-normalized descriptors, only that we do not need to warp the query image. Moreover, our filtering works on established correspondences and allows for a weaker rejection via a conservative, experimentally evaluated threshold of 30° degrees.

3D Point Visibility from SfM Model: Local descriptors are not invariant to viewpoint changes. For each 3D point in the scene model, the set of viewpoints under which it was observed is known. This enables us to determine the minimum and maximum rotation angle under which a 3D point is visible. It is used to bound the interval of camera rotations per correspondence for which voting is performed. To account for the viewpoint robustness of feature descriptors, we extend the bounding camera angles for a match by conservative $\pm 60^\circ$ degrees in each direction[1].

Feature Scale: We also utilize the scale at which a feature was detected in the image to reason about the feasibility of a correspondence. Given a database image with focal length f observing a feature belonging to the 3D point p with scale s_I, we use the concept of similar triangles to obtain the scale $s_{3\mathrm{D}}$ of the 3D point as $s_{3\mathrm{D}} = s_I \frac{d}{f}$, where d is the depth of p in the local camera coordinate system. All observations of p thus form an interval of 3D scales. Following the same formula, we can use this interval to derive the interval $[d_{min}, d_{max}]$ of possible (camera to 3D point) depth values such that the 3D scales projected into the query image are similar to the scale of the matching feature. As derived in Sec. 3, the camera height interval defines the near and far distance ($d_{n|f}$) between camera and matching 3D point. We thus limit the extent of the voting shape to the intersection of both distance intervals, rejecting the match if it is empty.

Positional Prior: Besides orientation information, mobile devices often also provide location information (*e.g.*,

[1] We found that a threshold of 30° degrees (cf. [18]) is too restrictive.

Figure 5: Exemplary voting results for query images from the San Francisco (left) and Dubrovnik (right) dataset. Without usage of GPS information (top left, and most right column) votes are cast in the entire map. With GPS (white circle) the voting is restricted to the uncertainty region. In case of repetitive scenes (2nd column), *e.g.*, similar buildings or symmetric structures, our voting procedure returns a multi-modal distribution. In addition the localization accuracy, *e.g.*, depending on the distance to the scene, is reflected by the size of the returned distribution. The image framed in red shows a failure case.

network-based cell tracking, GPS, etc.). We represent the measured location and an upper bound to its uncertainty as a circular area in the voting space. For each match, we then only need to consider the intersection of its voting shapes with this prior region, usually enabling us to reject many wrong matches early on. This is achieved by our voting formulation in global world coordinates. It allows to directly filter based on the expected camera location for each correspondence individually, rather than restricting the part of the model to consider (*e.g.*, [6]) – which we believe is a much more natural way to include a pose prior. In comparison, [27] operate in local camera coordinates where a global location prior is not applicable. In addition there is a strong relation between the orientation of the query camera and its possible locations, which is explained visually in Fig. 4b. Using pose priors to limit the set of feasible camera locations thus also restricts the set of feasible rotations for each matching 3D point falling outside the uncertainty region.

5. Experiments and Results

To evaluate our approach we have conducted experiments on two real-world datasets which are summarized in Tab. 1. Exemplary voting results are visualized in Fig. 5.

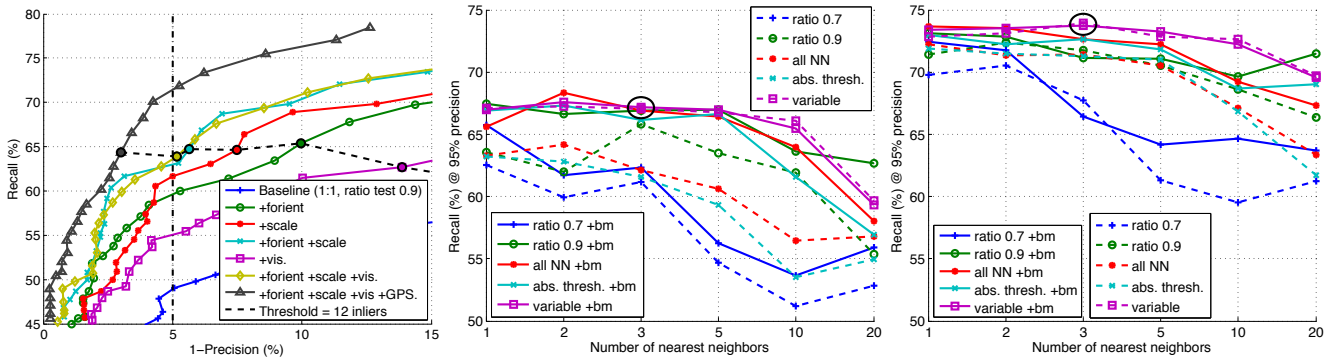
The *San Francisco* dataset [6] contains street-view like database images, while query images were captured on mobile devices and provided with (coarse) GPS locations. It is the most challenging dataset for image localization published so far, thus we base our analysis mostly on it. The datasets comes in four different types. Our evaluation is based on SF-0, which has the smallest size and thus represents the most challenging case for localization (unfortunately we could not obtain the SF-1 model). For each query image, its gravity direction is derived from the vertical vanishing point; thereby considering an uncertainty of 2° degrees in the voting procedure (cf. Sec 3.1). As in [6], we evaluate the performance of our method as the recall rate given a fixed precision of 95%. An image is considered to be correctly localized if it registers to points of the correct building ID according to the ground truth annotation; this is the same evaluation criterion as in [16]. Note that for SF-0, there exists an upper bound on the recall rate of 91.78%, since for 66 query images the corresponding building IDs are missing in the reconstructed model.

Second, we evaluate on the *Dubrovnik* [15] dataset which is a typical example for a 3D model build from image collections and has been widely used in the literature. As such database and query image follow a similar spatial distribution, which makes pose estimation easier. Consequently localization can be regarded as solved on the dataset, which especially [16] has shown recently.

Correspondence Generation: Similar as others [16, 22, 27] we use SIFT features for keypoint matching where de-

Dataset	San Francisco				Dubrovnik
	SF-0	SF-1	PCI	PFI	
DB images	610k	790k	1.06M	638k	6k
3D points	30.34M	75.41M	-	-	1.96M
Query images	803	803	803	803	800

Table 1: Characteristics of the datasets used for evaluation. PCI and PFI are sets of images used for retrieval tasks. SF-0 and SF-1 use parts of PCI to reconstruct a SfM model.



(a) Filter influence (1:1 matches, ratio = 0.9) (b) Precision over num. matches *without* GPS (c) Precision over num. matches *with* GPS

Figure 6: (a) Ablation study for the proposed filters on the SF-0 dataset with matching scheme (A) without back-matching (Legend for filters: forient = feature orientation, scale = feature scale, vis = 3D point visibility, GPS = location prior). (b-c) Recall rate at 95% precision for localization on the SF-0 dataset with different matching strategies (bm = matches additionally verified via back-matching). The marked data points denote the results of Tab. 2. Please see the supplementary material for additional evaluation curves.

scriptor entries are in the range 0-255. Matching is performed by approximated nearest neighbor search in a kd-tree structure, which is build from the descriptors belonging to all model observations. For each query feature up to N nearest neighbors are retrieved. To avoid biasing towards a particular matching strategy, we leverage and evaluate several of them. The studied matching schemes are:

(A) A *ratio test* on descriptor distances for retrieved nearest neighbors with a threshold of 0.7 (baseline) and 0.9 (used by [16, 27]). For 1:N matches the ratio test is performed wrt. the N+1$^{\text{th}}$ neighbor (cf. [29]).

(B) Retrieval of a *constant number of nearest neighbors*.

(C) *Absolute thresholding* on the descriptor distance of nearest neighbors to suppress wrong correspondence generation in sparsely populated feature space regions. A threshold of 224 was experimentally obtained from the model by evaluating corresponding descriptors of 3D points (similar to [5]), such that 95% of all correct matches survive.

(D) A *variable radius search*, where the search radius is defined by 0.7 times the distance to the nearest neighbor in the query image itself.

Methods (B) and (C) follow the idea to be independent of the model feature space density, while for (D) an adaptive threshold is estimated via the descriptor density in the query image, which serves as an approximation to the model characteristics. The latter typically returns many correspondences, except for query images containing repetitive structures. All methods can be augmented with a *back-matching* step which verifies that the retrieved 3D point shares the query feature as nearest neighbor in image descriptor space.

Influence of filters: First, we would like to study the influence of our proposed filters. We chose matching (A) with 1 nearest neighbor and a ratio test threshold of 0.9 as our baseline, making it comparable to [16, 27]. Then we apply the different filters individually and sequentially, see Fig. 6a. Most impact is observed by constraining the voting shape size to accord to the feature scale, followed by the consistency check on feature orientations. Restricting camera orientations has only limited influence if applied as last filter; however, it successfully serves the purpose of accelerating the voting procedure (the average angular range results in only 28° degrees) without any degradation of the pose estimation results. If a location prior is employed another performance boost of approx. 7% is noticed. In total only 15% of all correspondences survive the filtering steps. Often 12 inliers are used as measure to indicate a correct pose [16]. Employing this threshold, one can notice that recall stays at about 65%, while precision drops significantly without the filters. Summarizing, employing filters based on geometric constraints can lead to a performance increase of more than 20% and a speedup of up to factor 80. For other matching methods a similar influence of filters can be demonstrated.

Scalability: In our second experiment we want to study the influence of the different matching procedures providing the input to our algorithm. Results are illustrated in Fig. 6b and Fig. 6c with varying application of back-matching and a GPS prior. We run our algorithm with 1-3,5,10 and 20 nearest neighbors per keypoint. In the extreme case this accounts for up to 50k correspondences for a single query image. Due to the linear time complexity of our voting procedure the worst case runtime is still only at 16 seconds (cf. Fig 7). The obtained results show that the standard ratio test of 0.7 results in considerably worse performance. While a relaxed ratio test of 0.9 is doing significantly better, no significant difference can be noticed towards the other matching schemes. This suggests that the ratio test is useless in large-scale localization scenarios and strong geometric filtering is superior by a large margin (cf. Fig 6a). For matching strategies (B)-(D) and considering different numbers of nearest neighbors, we can notice that the performance is roughly constant up to 5 neighbors and starts to drop only beyond. This proofs the effectiveness of our algorithm; *e.g.*,

Method	avg #matches	Registration #imgs	time[sec]	Errors, Quartiles [m] median	1^{st}	3^{rd}	#imgs with error <18.3m	>400m
Voting	11265	798	3.78	1.69	0.75	4.82	725	2
RANSAC	56	796	-	0.56	0.19	2.09	744	7
Robust BA	49	794	-	0.47	0.18	1.73	749	13
Svärm [27]	4766	798	5.06	0.56	-	-	771	3
Sattler [22]	≤100	795.5	0.25	1.4	0.4	5.3	704	9

Table 3: Registration performance on the Dubrovnik dataset (no location prior, matching (D) with 3 nearest neighbors). Li *et al*. [16] register 800 images, but use a guided 3D-2D correspondences search if the initial 2D-3D matching fails.

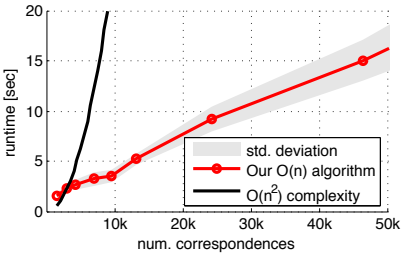


Figure 7: Runtime of our voting algorithm for different number of correspondences, showing our $\mathcal{O}(n)$ complexity.

2000 query keypoints and a required minimum of 12 inliers relate to an inlier ratio as low as 0.12%. However, it is also an interesting result, as it suggest that not necessarily more matches are better, but that there exists a trade-off between rejecting correspondences early on in matching and introducing too much noise in the pose estimation stage.

Comparison to state-of-the-art: Finally, we compare our results to the state-of-the-art in image-based localization and retrieval. Tab. 2 lists the evaluation of various forms of our algorithm with and without the usage of GPS information. As can be seen the geometric filters have a significant impact and our approach considerably improves over state-of-the-art. In particular the final P3P pose solver does not improve the localization performance, but provides a refined 6DoF pose. The average inlier ratio was at 0.9%, where RANSAC sample generation and hypothesis evaluation is obviously infeasible. Retrieval methods of [1,28] list their recall results without considering precision; *e.g*., 78% recall also relates to only 78% precision, as each query returns a positive result. For comparability to our method, we leverage the scores after geometric verification as computed by [1]. For [28] no scores are provided. Thus, we establish matches for their top 20 candidates (also contained in the SF-0 model) and run geometric verification with a 3-point-pose solver. The estimated camera pose supported by most inliers is then used to compute the precision-recall rate.

Method	SF-0 No GPS	SF-0 GPS	SF-1 No GPS	Retrieval (PCI) No GPS	Retrieval (PCI) GPS
Voting w\o filters	31.0				
— ” — + P3P	50.3				
Voting w\ filters	68.4	73.7			
— ” — + P3P	**67.5**	**74.2**			
Li et al. [16]	54.2		62.5		
Chen et al. [6]				41 (59)	49 (65)
Torii et al. [28]	50.9			63	
Arandjelović [1]	56.5			78	

Table 2: Recall rate on the SF-0 dataset for a precision between 94.5% and 95.2%, using matching (D) with at most 3 nearest neighbors. Results of comparing methods are on the SF-1 model and for image retrieval (using histogram equalization and upright features). For [6], results on PCI+PFI are given in brackets. For [1, 28], retrieval results denote the top ranked image *without* taking precision into account. We compute results for 95% precision via geometric verification on the top 20 candidate images.

For Dubrovnik, our evaluation criterion is equivalent to [27]; *i.e*., an image is considered correctly registered if the estimated pose is supported by 12 or more inliers under a reprojection error of 6 pixel. Tab. 3 lists the results and shows that we achieve state-of-the-art performance. The slightly better numbers of [27] wrt. registered images and on the error bounds stem from the fact that they use an optimal pose solver, while we leverage standard 3-point-pose RANSAC [9]. We also perform final 6DoF pose estimation directly via bundle adjustment on the voted inlier set with a robust Cauchy cost function. The results are inspiring: We achieve the smallest median location error and quartile errors reported on the dataset so far. This suggests that the inlier votes reflect a close upper bound on the true inlier set.

6. Conclusion

In this paper, we have proposed a novel camera pose estimation technique based on Hough voting, including a set of simple filtering operations, all employing strong geometric constraints. The run-time of our method grows linearly with the number of matches, and thus allows to handle huge amounts of correspondences in a reasonable time. Consequently, we have been able to study the influence of spatial filtering vs. aggressive rejection during matching and have shown the advantages of the former by achieving superior localization performance compared to state-of-the-art. Even though we are able to handle thousands of matches, our results also demonstrate that simply using more matches does not necessarily lead to a better localization performance. Further improvements could be achieved by using additional filters [10] or better descriptors [25]. Other interesting directions for future work are the generalization of voting shapes, *e.g*., via kernel or soft voting [14], and to construct a voting space that does not require quantization.

Acknowledgements: This work was supported by the EU grant FP7-269916 (V-Charge) and the CTI Switzerland grant #13086.1 PFES-ES (4D Sites).

References

[1] R. Arandjelović and A. Zisserman. DisLocation : Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, 2014.

[2] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In *European Conference on Computer Vision (ECCV)*, pages 517–530. Springer, 2012.

[3] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[4] S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[5] S. Cao and N. Snavely. Minimal scene descriptions from structure from motion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 461–468. IEEE, 2014.

[6] D. Chen, G. Baatz, K. Köser, S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[7] S. Choudhary and P. J. Narayanan. Visibility Probability Structure from SfM Datasets and Applications. In *European Conference on Computer Vision (ECCV)*, 2012.

[8] O. Chum, J. Philbin, J. Sivic, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[9] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[10] W. Hartmann, M. Havlena, and K. Schindler. Predicting Matchability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9–16, 2014.

[11] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[12] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision (ECCV)*, 2008.

[13] J. Knopp, J. Sivic, and T. Pajdla. Avoding Confusing Features in Place Recognition. In *European Conference on Computer Vision (ECCV)*, 2010.

[14] H. Li. A Simple Solution to the Six-Point Two-View Focal-Length Problem. In *European Conference on Computer Vision (ECCV)*, 2006.

[15] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *European Conference on Computer Vision (ECCV)*, 2010.

[16] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *European Conference on Computer Vision (ECCV)*, 2012.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[18] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *Int. Journal of Computer Vision*, 60:63–86, 2004.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[20] K. Quennesson and F. Dellaert. Rao-blackwellized importance sampling of camera parameters from simple user input with visibility preprocessing in line space. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 893–899. IEEE, 2006.

[21] T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-Based Localization using Direct 2D-to-3D Matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[22] T. Sattler, B. Leibe, and L. Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *European Conference on Computer Vision (ECCV)*, 2012.

[23] T. Sattler, B. Leibe, and L. Kobbelt. Towards fast image-based localization on a city-scale. In *Outdoor and Large-Scale Real-World Scene Analysis*, volume 7474 of *Lecture Notes in Computer Science*, pages 191–211. Springer Berlin Heidelberg, 2012.

[24] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(8):1573–1585, 2014.

[26] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[27] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate Localization and Pose Estimation for Large 3D Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[28] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual Place Recognition with Repetitive Structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[29] A. R. Zamir and M. Shah. Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(8):1546–1558, 2014.