Toroidal Constraints for Two-Point Localization under High Outlier Ratios

Federico Camposeco[1] Torsten Sattler[1] Andrea Cohen[1] Andreas Geiger[1,2] Marc Pollefeys[1]
[1]Department of Computer Science, ETH Zürich
[2]Autonomous Vision Group, MPI for Intelligent Systems Tübingen
{federico.camposeco,torsten.sattler,andreas.geiger,marc.pollefeys}@inf.ethz.ch

Abstract

Localizing a query image against a 3D model at large scale is a hard problem, since 2D-3D matches become more and more ambiguous as the model size increases. This creates a need for pose estimation strategies that can handle very low inlier ratios. In this paper, we draw new insights on the geometric information available from the 2D-3D matching process. As modern descriptors are not invariant against large variations in viewpoint, we are able to find the rays in space used to triangulate a given point that are closest to a query descriptor. It is well known that two correspondences constrain the camera to lie on the surface of a torus. Adding the knowledge of direction of triangulation, we are able to approximate the position of the camera from two *matches alone. We derive a geometric solver*[1] *that can compute this position in under 1 microsecond. Using this solver, we propose a simple yet powerful outlier filter which scales quadratically in the number of matches. We validate the accuracy of our solver and demonstrate the usefulness of our method in real world settings.*

1. Introduction

Estimating the pose of a query image from a set of 2D-3D matches is a central task in image-based localization [4, 15, 21, 22, 26, 28], with many applications in, *e.g.*, Structure-from-Motion (SfM) [12, 24, 25, 27], simultaneous localization and mapping (SLAM) [3, 5], augmented reality [1, 17], or camera calibration [2]. Wrong 2D-3D correspondences are typically handled through RANSAC [9] as long as the percentage of such outliers among all matches is not too large.

Image-based localization approaches establish 2D-3D correspondences by matching descriptors extracted from the query image (*e.g.*, SIFT [16]) against descriptors associated with the 3D model points. However, at large scale or in complex scenes with many repetitive elements, establishing 2D-3D correspondences becomes a challenging task due to the inherent ambiguities of the local appearance [15]. As it becomes harder to distinguish between correct and incorrect matches based on local descriptors alone [15], localization algorithms must be able to cope with large outlier ratios in order to enable reliable pose estimation. This in turn creates a strong need for developing efficient outlier filtering strategies which are able to identify and remove wrong matches.

In this paper, we derive a new previously unused geometric constraint that can aid large-scale localization. Exploiting this constraint, we present a novel filtering strategy that can cope with arbitrary outlier ratios, while retaining a runtime that scales quadratically with the number of matches. Previous approaches either require knowledge about the gravity direction in combination with a prior on camera height [26, 28] or knowledge about the full camera orientation [13]. In contrast, our approach does not depend on such external information and can approximate the full 6 DOF pose from just two 2D-3D correspondences. For our application, however, we only use the position from the pose to efficiently prune outliers (*c.f.* Section 3.5). Whereas most reconstruction methods consider viewpoint variance [18] of descriptors as a weakness, we view it as a strength useful to infer the viewpoint of the camera.

Our main observation is that, during the SfM process used to generate a 3D model, each 3D point is associated with the descriptors of the image features from which it was triangulated. Thus, each of these descriptors is associated with the viewing direction under which the point was observed. Empirically, we found that the best matching descriptor of a 3D point for a given 2D feature provides a good approximation to the viewing direction of the query image, *c.f.* Section 3.3 and Fig. 1, typically being within 10° of the closest match. We call this viewing direction constraint the *triangulation constraint*. Additionally, given two 2D-3D matches, it is known that the camera will lie on the surface of a torus [7]. The toroidal constraint combined with the triangulation constraint allows us to formulate a solver that estimates the position of the camera from only

[1] MATLAB sample code available at www.cvg.ethz.ch/research/toroidal-constraints/

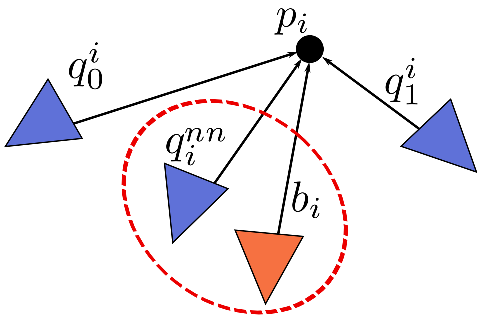


Figure 1. **Novel Geometric Information.** For each query descriptor, d_i^{query}, we find the closest descriptor inside the track that produced the 3D point p_i. We do this for each match in $\mathcal{S}$, yielding our augmented match set $\bar{\mathcal{S}}$ (see Section 3.3 for more details). Given a query measurement, b_i and its matched 3D point p_i we are able to find an estimate of its closest ray in 3D by matching against all images that triangulated p_i. Thus, we constrain the query camera position to lie near ray q_i^{nn}.

two matches. While the resulting positions are approximations, they are accurate enough to enable efficient outlier detection, which is a key task for pose estimation at large scale.

This paper makes the following contributions: *(i)* We derive a novel two-point formulation to estimate the absolute position of the camera which incorporates prior information on the viewing directions. To the best of our knowledge, ours is the first pose solver which directly incorporates this type of information into the pose estimation task. *(ii)* While computing an exact solution to this problem analytically is hard, we derive an approximate solver and empirically show that it is able to recover near-optimal solutions. As our solver only requires solving two quadratic problems, it is very efficient with run-times of around $\sim 1\mu$s. *(iii)* Based on our solver, we propose a novel outlier filter, whose run-time is independent of the outlier ratio. Compared to previous approaches, it does not make any assumptions on the availability of external information [13, 26, 28]. Besides, it is significantly simpler to implement than [26, 28] and has a lower computational complexity than [13, 26]. *(iv)* Lastly, we show that the novel constraint is indeed meaningful for the localization task by comparing its performance as an outlier filter to the state-of-the-art.

The rest of this paper is organized as follows: Section 2 provides an overview of the related work. Section 3.1 introduces the problem of large-scale image-based localization in the presence of large outlier ratios. Sections 3.2 and 3.3 provide a description of the toroidal and triangulation constraints. Section 3.4 describes our geometric solver which is then used to design an efficient outlier filter in Section 3.5. Section 4 validates our proposed method on both synthetic and real-world datasets.

2. Related Work

Recent work on scalable image-based localization has dealt with ambiguities in the matching stage by relaxing the matching criterion [15, 21, 26, 28]. They handle the resulting larger amount of wrong matches by detecting and filtering incorrect correspondences before pose estimation. These methods can be divided into approaches based on co-visibility [15, 21, 22] and geometric reasoning [26, 28]. The method proposed in this paper falls into the second category.
Visibility-based methods exploit the fact that the SfM process provides information about which 3D points can be observed together. This information is encoded in the bipartite visibility graph [14] and the 2D-3D matches determine sets of connected components in this graph. Sattler *et al.* [22] use only those correspondences falling into the largest connected component for pose estimation. Rather than using a single component, [21] computes poses from multiple subsets of matches and then select the pose with the largest number of inliers. Instead of deciding on a fixed subset before pose estimation, Li *et al.* adopt a RANSAC sampler to avoid computing a pose from points not co-visible together [15]. Visibility filtering usually reduces the number of RANSAC iterations required to ensure a good estimate is found. Yet, it does not remove the dependency of RANSAC's run-time on the outlier ratio.
Geometry-based approaches determine a subset of matches whose 3D points are geometrically consistent with their corresponding 2D features. The main motivation is to design an approach whose run-time depends only on the number of matches and not on the outlier ratio. One way to select such a subset are branch-and-bound algorithms [6, 8, 20], which often come with guarantees on the optimality of their solution. However, their algorithmic complexity only allows them to handle a relatively small number of matches given a limited computational budget.

Recently, more efficient outlier filters have been proposed that rely on additional information. Given the full orientation of a camera, the method from Larsson *et al.* rejects outliers in $\mathcal{O}(n^2 \log n)$ [13], where n is the number of matches. They obtain a bound on the maximum number of inliers for each single match. This, in turn, enables them to identify and remove correspondences that cannot be part of the maximum inlier set. Unfortunately, their method requires to repeatably determine the intersection between two cones, which is computationally involved. Similar to our approach, Larsson *et al.* thus employ an approximation algorithm that is more efficient to compute.

Svärm *et al.* [26] present an outlier filter based on a known gravity direction and an estimate of the camera's height above ground. As a result, they can model pose estimation as a 2D registration problem. Similar to Larsson *et al.*, the approach of [26] determines the maximum number of correspondences geometrically consistent with each match. Matches are rejected if this number falls below an adaptive threshold. While both, Svärm *et al.* and Larsson *et al.*, require $\mathcal{O}(n^2 \log n)$ steps, our approach has a computational complexity of $\mathcal{O}(n^2)$. At the same time, it is more general as it neither requires information about the full camera orientation, the gravity direction, nor the camera's height.

Following the same setup as [26], Zeisl *et al.* propose a filtering strategy based on voting that has an optimal asymptotic complexity of $\mathcal{O}(n)$ [28]. However, their voting strategy is rather involved, both in terms of implementation and constant factors contributing to the running time. In order to accelerate their method, they exploit additional constraints provided by the local feature geometry, *e.g.*, the scale and orientation of a 2D feature, as well as viewing direction constraints in order to reject matches before voting.

In contrast to Svärm *et al.* and Zeisl *et al.*, our proposed approach is both very efficient to compute and simple to implement. It does not require any additional assumptions. Instead, it leverages information that is readily available but has not been previously exploited. This new insight allows us to formulate new constraints that can be efficiently used in a simple setting, therefore avoiding more involved schemes as the ones introduced by Zeisl *et al.* Our experiments show that we can achieve performance similar to [26] and [28] while maintaining low computational complexity.

3. Localization Using Two Points And Their Directions of Triangulation

In the following, we first formulate the problem and provide a description of the toroidal and triangulation constraints. We then describe our geometric solver and the proposed outlier filter based on this solver.

3.1. Problem Formulation

Our aim is to localize an image of a scene given a SfM model. To this end, we assume a set of 2D-3D correspondences $\mathcal{S} = \{b_i \leftrightarrow p_i\}_i^N = \mathcal{O} \cup \mathcal{I}$, where $\mathcal{O} \cap \mathcal{I} = \emptyset$ denote the unknown sets of outlier and inlier matches. The set $\{b_i\}_i^N$ denotes the putative image projections of the matched landmarks $\{p_i\}_i^N$. Since we deal with calibrated image features only, we view b_i as a 3D unit vector in the camera frame of reference that emanates from its center. To identify the inlier set $\mathcal{I}$ and refine the pose from all inlier matches, the most popular approach is to use P3P plus RANSAC [9]. However, as noted before, a large outlier ratio $|\mathcal{O}|/|\mathcal{S}|$ greatly reduces the chances of finding a correct pose with such methods. This is because RANSAC-style methods are prone to getting stuck in local minima (*i.e.*, they do not converge to the optimal $|\mathcal{I}|$).

Ideally, one would exhaustively search through all triplets in $\mathcal{S}$ and vote for the pose with the highest consensus. However, this is prohibitively slow for most applications as up to $4\binom{N}{3} = {}^2\!/\!_3\, N^3$ poses (P3P returns up to 4 feasible solutions) need to be evaluated. The goal of our approach is to drastically reduce $|\mathcal{O}|$, so that pose recovery becomes an easier problem. We will now derive a solution which prunes the vast majority of outliers and scales quadratically with the number of matches.

3.2. Toroidal Constraints

Given two matches $m_0, m_1 \in \mathcal{S}$, our goal is to find the camera center location C represented in world frame coordinates. Let Π_0 denote the 3D plane defined by the two matched 3D points p_0 and p_1, as well as the camera center C. Since the angle θ between the rays of the features in the camera frame $\theta = \measuredangle(b_0, b_1)$ is known from their pixel position and the calibration parameters, the location of C is constrained to lie on a circle (*c.f.* Fig. 2a). However, this circular constraint is still fulfilled if we rotate C around the line connecting p_0 and p_1 by any angle u (*c.f.* Fig. 2b). Thus, C must lie on the surface of a torus $\mathbb{T}^2$, yielding

$$C(u, v) = \begin{bmatrix} (R + r\cos v)\cos u \\ (R + r\cos v)\sin u \\ r\sin v \end{bmatrix} \in \mathbb{T}^2 \,. \tag{1}$$

Here v is the angle which parameterizes the circle, R denotes the major radius of the torus (corresponding to the distance from the origin to the center of the circle) and r is the radius of the circle. As it can be seen from Fig. 2, in this particular setting the torus will always be self-intersecting (*i.e.* $r > R$) since the axis of revolution always includes two points on the circle. This yields sections of the torus on which the camera can lie; if $\theta > \pi/2$ the camera will be constrained to the *inner surface* of the torus, otherwise it will lie on the *outside surface* [7].

Without loss of generality, we make the implict assumption that p_0 and p_1 are aligned with the z-axis, and that their midpoint is at the origin. This can be trivially achieved by pre-rotating and translating p_0 and p_1 by a suitable amount. After the location of the camera on $\mathbb{T}^2$ has been found, we transform it back to the world frame of reference.

Without any additional constraints, the exact location of the camera center C on the two-dimensional manifold $\mathbb{T}^2$ is unknown. In the following, we make use of previously unutilized information from the matching process to find a likely location for C on the surface of the torus.

3.3. Triangulation-Ray Constraints

Any point p_i of the SfM model has been obtained by triangulating a set of $M > 1$ image measurements $\{q_j^i\}_j^M$ with associated descriptors $\mathcal{D}_i = \{d_j^i\}_j^M$, where we represent any q_j^i as a 3D vector of unit length. Alternatively, q_j^i can be regarded as a vector emanating from p_i towards

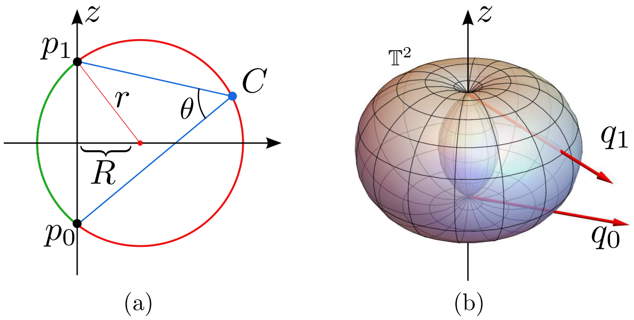


Figure 2. **Toroidal Constraints From Two Matches.** The circle in (a) describes the possible locations for C given two 2D-3D correspondences. The camera is located on the red arc for angles $\theta < \pi/2$ and on the green arc otherwise. The z-axis is the axis of revolution for this circular constraint, yielding $\mathbb{T}^2$, as seen in (b). The unit vectors q_0 and q_1 are the directions from the points p_0, p_1 to the cameras whose descriptor was matched to the query image descriptors.

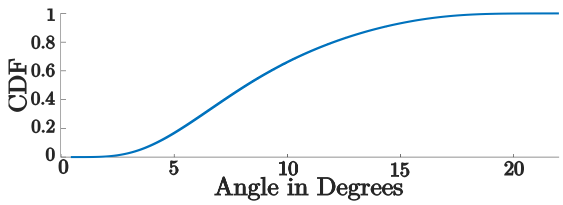


Figure 3. **Accuracy of the Matched Ray.** To validate our new constraint, we show the cumulative density function of the angular error between b_k and q_k for a real dataset with ground truth [23].

the center of camera j (*c.f.* Fig. 2b). For localization, most methods exploit a single descriptor derived from this set, *e.g.*, $D_i^{avg} = \text{mean}(\mathcal{D}_i)$ under the assumption that the variation in $\mathcal{D}_i$ is small enough for this to be a valid approximation. However, we make the observation that the variability within $\mathcal{D}_i$ can be exploited to obtain previously unused but geometrically meaningful information.

For a given match $(b_i \leftrightarrow p_i) \in \mathcal{S}$, we denote the associated query descriptor as d_i^{query}. We find the closest descriptor to d_i^{query} against all M descriptors that generated p_i, *i.e.*, we find $d_i^{nn} \in \mathcal{D}_i$ that is closest in descriptor space to d_i^{query} (*c.f.* Fig. 1). We empirically observed that for SfM ray measurements associated with d_i^{nn}, q_i^{nn} is close in space to the ray which has produced our query measurement b_i (*c.f.* Fig. 3). Thus, we obtain the direction of triangulation for each observed feature match $b_i \leftrightarrow q_i$ (where for simplicity $q_i = q_i^{nn}$). This augments the set of matches with noisy but informative orientation estimates $\bar{\mathcal{S}} = \{b_i \leftrightarrow (p_i, q_i)\}_i^N$.

Given *two* of these augmented matches, $\bar{m}_0$ and $\bar{m}_1$, we aim to find $C(u,v) \in \mathbb{T}^2$ such that the angular distance to q_0 and q_1 is minimized. Notice that we do not strictly enforce b_i to be coincident with q_i, as this is only possible for the unrealistic noise-free case in which each query image was taken from exactly the same pose as a camera in the SfM model.

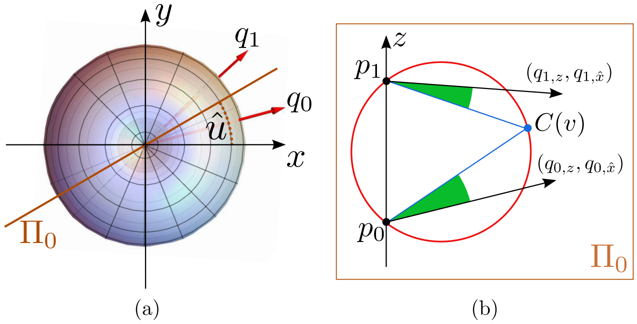


Figure 4. **Approximate Solution for Camera Position.** The cost optimization on $\mathbb{T}^2$ is approximated by a two step procedure: finding the angle parameterized by u, then locating the point $C(v)$ by solving two square roots. The solver we propose minimizes the angles highlighted in green (b).

Our goal is to find a point on the torus whose location is compatible with the triangulation directions q_0 and q_1. We model the variation in $\mathcal{D}_i$ as a product of viewpoint change solely. Thus, the most compatible location on $\mathbb{T}^2$ for C minimizes the *angular* distance, E, to both q_0 and q_1:

$$E(u,v) = \measuredangle\left(P_0(u,v), q_0\right)^2 + \measuredangle\left(P_1(u,v), q_1\right)^2, \qquad (2)$$

where $P_i(u,v) = C(u,v) - p_i$, *i.e.*, the vector from p_i to C and $C(u,v) \in \mathbb{T}$, *c.f.* Fig. 4b.

3.4. A Geometric Solver

Minimizing the cost in Eq. 2 results in multiple local minima. In particular, we are interested in two local minima, one located on the inside and the other on the outside of the self-intersecting torus. Of these two, we select the one which best fulfills the angular constraint of $\theta = \measuredangle(b_0, b_1)$ as described in Section 3.2 (*c.f.* Fig. 2a). Since Eq. 2 needs to be minimized for all $N(N-1)/2$ match pairs, we need an efficient solver which yields a feasible solution within a few microseconds. Extrema of the initial cost happen when the gradient of the cost equals zero, and we can use this to build a system of polynomial equations. We initially pursued a Gröbner basis approach that would solve the system of polynomial equations given by the gradient of the cost. Depending on the parameterization (Lagrangian or trigonometric), we found this to yield between 24 and 32 solutions by using Macaulay2 [10]. Therefore, a respective 24 to 32 square matrix needs to be eigen-decomposed, leading to runtimes of more than 2 milliseconds. Although this yields the true global optima of Eq. 2, the runtime and number of solutions to consider render this method unpractical.

We thus propose an approximate solution by solving the problem in two steps. By inspecting the projection of the torus onto the xy-plane (*c.f.* Fig. 4a), one can see that both triangulation rays emanate from the origin. Thus, a reasonable approximation $\hat{u}$ of the optimal angle u is the average of the angles formed by the projections of q_0 and q_1 onto the xy-plane.

Assuming the angle u around the z-axis to be known, Eq. 2 can be reduced to a one dimensional cost function

$$E(v) = \sum_{i=0,1} \left(\arctan\left(\frac{q_{i,z}}{q_{i,\hat{x}}}\right) - \arctan\left(\frac{P_{i,z}}{P_x}\right)\right)^2 \tag{3}$$

where $q_{i,\hat{x}}$ is the projection of the x coordinate of q_i onto the plane defined by $\hat{u}$ and the z-axis, denoted Π_0. $P_x = R + r\cos v$ and $P_{i,z} = r \sin v - p_{i,z}$.

We can further simplify Eq. 3 by leveraging the fact that

$$\arctan(\alpha_1) + \arctan(\alpha_2) = \arctan\left(\frac{\alpha_1 + \alpha_2}{1 - \alpha_1\alpha_2}\right). \tag{4}$$

Knowing that

$$X^* = \arg\min_x \arctan(x)^2 = \arg\min_x x^2\,, \tag{5}$$

and dropping the arctan yields

$$\hat{E}(v) = \sum_{i=0,1} \left(\frac{s_i - x_i(v)}{1 + s_i\, x_i(v)}\right)^2, \tag{6}$$

where $s_i = q_{i,z}/q_{i,\hat{x}}$ and $x_i(v) = P_{i,z}/P_x$ are the slopes of the rays and the unknown point on the circle, respectively. Setting the derivative of Eq. 6 w.r.t. v equal to zero we obtain an equation in terms of $\cos v$ and $\sin v$,

$$\begin{aligned} \frac{\partial \hat{E}}{\partial v} &= \sum_{i=0,1} \frac{\mu_i\ \rho_i\ \phi_i}{\xi_i} = 0\,,\text{ where} \\ \mu_i &= \left(r s_i^2 \cos v + r\cos v + R s_i^2 + R\right), \\ \rho_i &= \left(p_{i,z}\sin v + r + R\cos v\right), \\ \phi_i &= \left(r s_i \cos v - r \sin v + R s_i - p_{i,z}\right), \\ \xi_i &= \left(s_i p_{i,z} + r s_i \sin v + r\cos v + R\right)^3. \end{aligned} \tag{7}$$

Groebner basis analysis (*c.f.* supplementary document) shows that Eq. 7 has at most 12 solutions. This is a great improvement upon the 32 solutions of the initial problem.

Further, it turns out that 6 of the obtained 12 solutions correspond to a repeated root, namely a root for an invalid solution. This repeated root, $r\cos(v) = R$, corresponds to the solution coincident with the points p_0 or p_1 and arises from the fact that at that point, P_x is zero and thus the angle is undefined. Manipulating the equations algebraically allows us to factor out $(r\cos(v) - R)^3$, effectively removing 6 roots from the 12 algebraically possible ones (note that for this factor, v and $-v$ are both feasible solutions). Using the half-angle tangent substitution (*i.e.*, $t = \tan(v/2)$), Eq. 7 factors as

$$\begin{aligned} 0 = &\overbrace{(r + R + t^2(R - r))^3}^{\text{Invalid Roots}} \cdot \overbrace{(\lambda_1 + \lambda_2 t + \lambda_3 t^2)}^{\text{Valid roots: } v_1, v_2} \cdot \\ &\underbrace{((s_1 + s_0 + t(4s_0s_1 - 2) - t^2(s_0 + s1))}_{\text{Valid roots: } v_3, v_4} \cdot \underbrace{(f_c(t))}_{\text{Complex}} \end{aligned} \tag{8a}$$

where

$$\begin{aligned} \lambda_1 &= \kappa - \tau, \\ \lambda_2 &= 2r(s_0 + s_1)(p_{1z}(s_1 - s_0) + R s_0 s_1 + R), \\ \lambda_3 &= \kappa + \tau \end{aligned} \tag{8b}$$

and

$$\begin{aligned} \kappa = &R^2\left(s_0^2\left(s_1^2 - 1\right) + 4s_0s_1 - s_1^2 + 1\right) + \\ &r^2\left(s_0^2\left(2s_1^2 + 1\right) + 2s_0s_1 + s_1^2 + 2\right) - \\ &2Rp_{1z}\left(s_0^2 s_1 - s_0 s_1^2 + s_0 - s_1\right) \\ \tau = &r(s_0s_1 - 1)(p_{1z}(s_1 - s_0) + R s_0 s_1 + R)\,. \end{aligned} \tag{8c}$$

The factor $f_c(t)$ in Eq. 8a is a term quadratic in t whose roots are always complex. For details on the derivation, we refer the reader to the supplementary document.

This approximate closed form solution is several orders of magnitude faster w.r.t. the former methods, since it only requires the solution of two quadratic polynomials in t. We obtain a single solution by taking the minimum cost solution satisfying the inside/outside constraint of the camera. Even though the proposed solver finds only an approximate solution, albeit a pretty accurate one (*c.f.*, Fig. 6), this is acceptable since we are only interested in a rough and fast estimate of the camera position to efficiently prune outliers. The complete procedure for computing an estimate of the camera position from two matches is given in Algorithm 1.

Algorithm 1 Compute C given $b_i \leftrightarrow (p_i, q_i)$

```
Require: Points $p_0$, $p_1$, rays $q_0$, $q_1$, measurements $b_0$, $b_1$
Compute rotation and translation $\mathtt{T}$ s.t. $\mathtt{T}\hat{p}_i = p_i$ are on the $z$-axis and their midpoint is the origin.
$\hat{u} \leftarrow (\arctan(q_{0,y}/q_{0,x}) + \arctan(q_{1,y}/q_{1,x}))/2$
$\theta \leftarrow \arccos(b_0 \cdot b_1)$
Compute $\mathcal{V} = \{v_i\}_{i=1}^4$ using Eq. 8a.
for all $v_i \in \mathcal{V}$ do
    if $v_i$ on the side of the circle constrained by $\theta$ (Fig. 2) then
        Compute cost: $cost_i \leftarrow E(\hat{u}, v_i)$ using Eq. 2
    end if
end for
$v^* \leftarrow$ solution with minimum $cost_i$
return $\mathtt{T}^{-1}C(\hat{u}, v^*)$
```

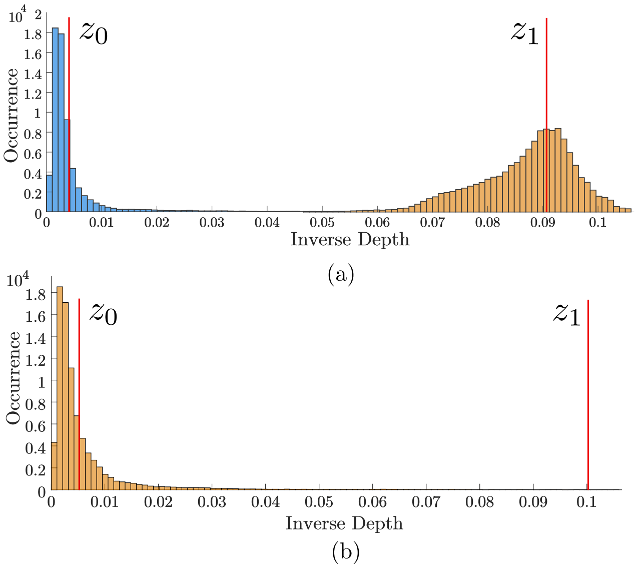
3.5. The Outlier Filter

For any pair of matches, we can follow the efficient procedure outlined in the previous section in order to obtain an estimate of the camera position. Since our aim is to deal with cases that have very low inlier ratios, we want to use every possible match pair in order to fully explore the set of possible camera poses. A naïve approach would simply consider all $Q = N(N-1)/2$ matching pairs, compute a position estimate, and later cluster on the space of 3D positions. However, this method has a few drawbacks. First of all, for images with a very high number of feature detections, Q might be larger than 10 million pairs. Thus, clustering such a dense population of position hypotheses could prove difficult. Second, clustering in 3D space would not identify the inlier set required for pose refinement. In this paper we therefore take a different approach by individually scoring each 2D-3D match.

We use 3D occupancy information to discard gross outliers by using an octree, which is particularly suited for images with a very large number of matches. To this end, we store all Q position hypotheses in an octree of fixed depth. Because of its fixed depth, the structure itself helps us discard position hypotheses far from a principal cluster. After generating all hypotheses, we traverse the tree once to find the most populated voxel, denoted V^*. For all remaining operations, we only use point hypotheses that lie inside V^*.

After computing all camera hypotheses, for each match $m_i \in \mathcal{S}$ we have a set of N_V position hypotheses in which match m_i participated and that lie inside V^*, denoted as $\mathcal{C}_i = \{C_j^i\}_j^{N_V}$. Using these putative camera positions, we compute N_V inverse depth measurements $\mathcal{W}_i = \{\|p_i - C_j\|^{-1}\}_j^{N_V}$ (*c.f.* Fig. 5) and use this to derive an inlier score for each match *individually*. Given that outlier matches will produce camera positions that are *not* clustered around any particular point in space, the inverse depth measurements they produce will not cluster around the true inverse depth of the feature p_i. Rather, the inverse depths produced by outliers will cluster around zero (since many outliers will result in positions very far from the true position). On the other hand, an inlier match m_i will necessarily present two peaks, one near zero z_0 (as produced when m_i was paired with an outlier match), and a peak around the true inverse depth value z_1 (*c.f.* Fig. 5).

Our goal is therefore to find the number of inverse depth measurements that are part of the true depth cluster, z_1. If the support of the z_1 cluster $|\mathcal{W}_i^{z_1}|$ is high, then there is high evidence that the point p_i is geometrically sound, *i.e.*, it is an inlier. We thus define the score of the i-th match as the ratio $|\mathcal{W}_i^{z_1}|/|\mathcal{W}_i|$. To produce the scores we then only need to partition $\mathcal{W}_i$ into two clusters. We use k-means with $k = 2$ for this purpose. For such simple one-dimensional two-class clustering problems, k-means can be approximated with linear complexity by setting a fixed number of iterations (equal to 20 for our case), and will converge in approximately 1μs per match.



(a)

(b)

Figure 5. **Inverse Depth Distribution for Inliers and Outliers.** Shown in blue are the inverse depths that were classified as outliers and in orange the inliers, with their respective centroids z_0 and z_1. For inliers, the support for z_1 will be high (a). On the other hand, the support of z_1 for outliers will be very low (b). The data used for this visualization was produced from a single image from [23].

Algorithm 2 Produce a score for match $m_i \in \mathcal{S}$

```
Require: A set of $N_V$ positions $\mathcal{C}_i = \{C_j^i\}_j^{N_V}$
for all $C_j^i \in \mathcal{C}_i$ do
    $W_j^i \leftarrow \|p_i - C_j^i\|^{-1}$
end for
Initialize cluster centroids: $z_0 \leftarrow 0$ and $z_1 \leftarrow 1$
Run $k$-means over all $W_j^i$
$\mathcal{W}_i^{z_0} \leftarrow$ members of cluster $z_0$
$\mathcal{W}_i^{z_1} \leftarrow$ members of cluster $z_1$
return $|\mathcal{W}_i^{z_1}|/|\mathcal{W}_i|$
```

The output of the procedure outlined above and summarized in Algorithm 2 is thus a score for each input match. Notice that the position computation and the inverse depth clustering are fully parallelizeable. Having obtained a set of scores, we can use a threshold $t_{id} \in [0, 1]$ to discard outliers, or leverage the statistics of the scores to keep a prescribed number of matches. From the remaining set, we run RANSAC with P3P [11] to get a final pose estimate. Since the outlier ratio of the remaining filtered set of matches is much lower compared to the initial set, RANSAC converges quickly to the correct solution.

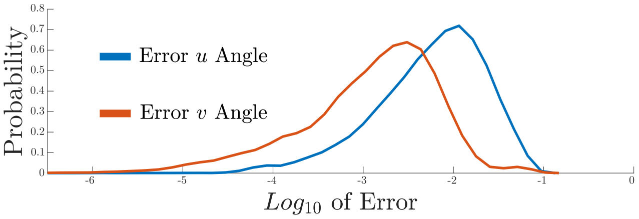


Figure 6. **Accuracy of the Geometric Solver.** The $\log_{10}$ angular error in degrees of our approximate geometric solver against the ground truth method, *i.e.*, solving the cost in Eq. 2 using gradient descent and using multiple initializations on a 100×100 grid. The strongest approximation we make, that of the u angle, suffers most from error. However, v has a very low approximation error.

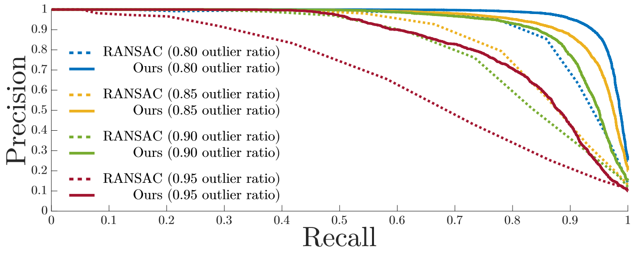


Figure 7. **Synthetic Evaluation of the Filter.** Precision and recall curves of the filter for different outlier ratios compared against RANSAC. Precision is computed as number of correctly classified inliers (CCI) over number of classified inliers. Recall is measured as CCI over total number of true inliers.

4. Experiments

4.1. Synthetic Evaluation of the Solver

In order to validate the solver described in Section 3, we compare the positions computed by our solver to the exact solution provided by an exhaustively initialized Gauss-Newton minimization. For this, we generate 10^6 random synthetic scenes in the following manner. We sample two 3D points, p_0 and p_1, from a uniform distribution inside the cube $[0, 10]^3$. Afterwards, a random camera location, C is chosen from the same distribution and translated 10 units away of the cube in the Z direction. We then simulate noisy triangulation-ray matches by adding Gaussian noise to the unit vectors $q_i = (C - p_i)/\|C - p_i\|, i = 0, 1$. The noise was added to a plane perpendicular to q_i with a standard deviation of 0.5. We have empirically observed that, for inlier matches, these are adequate simulation parameters.

Fig. 6 shows the error between the ground truth estimator and our solver, which is statistically negligible when compared to the angular distance from q_i to b_i.

Additionally, in order to validate the efficacy of our filter as an outlier filter, we generate 3000 2D-3D matches and then inject a varying amount of outliers. The performance of our filter and of simple RANSAC were then compared to validate the usefulness of the filter as an outlier rejection scheme. As it can be seen in Fig. 7, the filter is quite efficient for high outlier ratios.

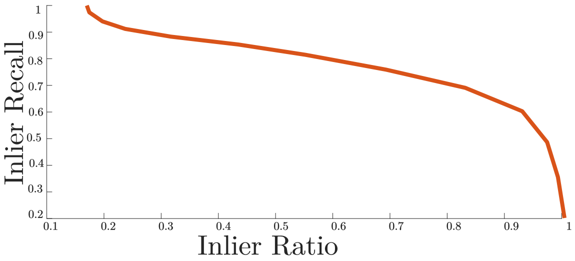


Figure 8. **Efficacy of our Method as an Outlier Filter.** For a prescribed threshold t_{id}, the inlier ratio can be increased although some number of true inliers might be discarded. This graph shows such trade-off when averaging inlier-ratio and recall for the dataset in [23]. *E.g.*, given a threshold we can get 80% of inliers with over 50% inlier ratio. A high threshold will reject the vast majority of outliers but might reject true inliers, and conversely for a low t_{id} threshold.

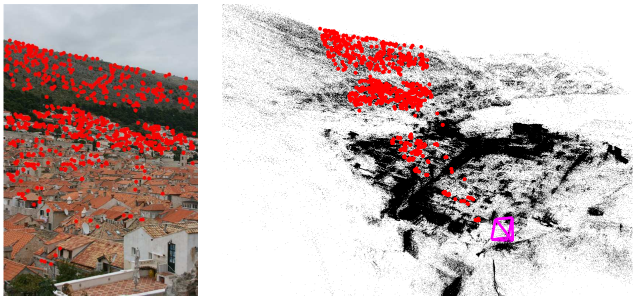


Figure 9. **Depth Range.** Left: View with inlier matches in red. Right: 3D model with inlier points (red) and camera in pink.

4.2. Real-World Evaluation

In order to further validate our method and assess its performance w.r.t. the state of the art, we conduct experiments on a publicly available real-world dataset. The dataset we used is the Dubrovnik dataset from [25], which consists of 800 query images with SIFT [16] features that are matched against an SfM model with 1.89 million landmarks. This dataset is scaled in meters and each of the 800 query images have bundle adjusted ground-truth poses. We chose this dataset since it is widely used for comparing localization methods, as it is a challenging dataset with ground truth poses. Furthermore, this dataset is particularly challenging for our filter. Since the depth variation per view in this dataset can be quite large (*c.f.* Fig. 9), this dataset allows us to validate our method under such difficult conditions.

For each query image, we extract ground-truth inlier matches by using the provided ground-truth pose. We first evaluate the effect of different score thresholds $t_{id} \in [0,1]$ (*c.f.* Fig. 8). For this, we vary t_{id} to obtain different performance points, where high score thresholds increase the inlier ratio but may prune out true inliers. We observe that

Method	Assumptions			Registration Statistics			Error Quartiles [m]			$\|S\|$	Time [s]
	V	**S**	**R**	$\|I\|>11$	$\epsilon<18.3$	$\epsilon>400$	1^{st}	2^{nd}	3^{rd}		
Setting 1				800	739	8	0.22	1.07	2.99	6210	9.7
Setting 2				797	731	8	0.50	1.16	3.42	8415	9.1
Setting 3			●	793	720	13	0.81	2.06	6.27	4766	3.2
RANSAC+P3P			●	634	601	11	1.20	5.06	8.11	4766	12.9
Zeisl [28]	●	●		798	725	2	0.75	1.69	4.82	11265	3.78
Zeisl BA [28]	●	●		794	749	13	0.18	0.47	1.73	49	-
Svärm [26]	●		●	798	771	3	-	0.56	-	4766	5.06
Sattler [22]			●	797	704	9	0.50	1.3	5.0	$\leqslant$ 100	0.16

Table 1. **Results for the Dubrovnik Dataset.** We compare registration metrics for our method against the state-of-the-art. Here ϵ and $|I|$ represent the translation error (in meters) and the number of inliers of the final image registration, respectively. Further, **V** denotes known vertical direction, **S** known scale and **R** methods relying on a good SIFT ratio (*i.e*. relying on the discriminative power of features, which is dataset-dependent).

the filter operates as expected; for a wide range of thresholds we can increase the inlier ratio while discarding only very few inliers.

For the next evaluation, we draw several descriptive statistics from our pose results in order to compare it against previously existing methods on the Dubrovnik dataset (*c.f*. Table 1). Additionally, we compare to a simple RANSAC+P3P method as a baseline. We run three different settings, the first one uses the octree method discussed in Section 3.5 to discard gross outliers, while the second and third do not.

Setting 1 For this setting, we use all available matches as computed by FLANN [19] and set t_{id}=0.55.

Setting 2 Here we do not use an octree but still use all available matches.

Setting 3 Here we discard the matches for which the ratio test [16] (the ratio of the first SIFT nearest neighbor over the second neighbor) was more than 0.9 (this is the setting that Svarm *et al*. use for their filter). Since now the number of matches is lower, we do not use an octree.

For Settings 2 and 3, we set t_{id}=0.35. Notice that we may set a lower threshold for these two settings since a lot of outliers have already been removed by the octree or the ratio test.

Since many true inliers are discarded using a ratio test and our filter is designed to handle as many outliers as needed, we perform better under the first setting. However, this means that the number of average matches per image to be considered is much higher ($|S|$ in Table 1), and as such the runtime of our filter is higher. In many hard cases, there may be a high number of *approximate inlier* matches. These are matches that would not be inliers to a final P3P position, but are not effectively discarded with our approximate positions. Thus, these matches are very close to being geometrically sound, since their true projection is only a few pixels away, passing the proposed outlier filter. This results in the final RANSAC solver retrieving a wrong local minimum leading to a bad localization.

However, since our filter enforces matches to be constrained by a strong geometric restriction, we observe better median positional errors w.r.t. other methods that do not refine their final solution, such as [28] with voting only. Further, [26] propose an optimal pose estimation strategy under low-outlier conditions that could also be used in conjunction with our filter. This would further improve our positional accuracy. Under Setting 2, we do not achieve such high performance since the inverse depth data points are too contaminated with outliers, rendering the choice of threshold more challenging. Finally, Setting 3 has lower recall since we clearly prune out valuable inliers using the ratio test. In terms of positional accuracy and successful localizations, we are able to produce competitive results for all of the settings used. Note that, in contrast to [26, 28], we do not make *any* assumptions about the data: no assumption of known vertical, known scale, known ground plane, *etc*. Instead, we use *geometrically meaningful* information latent in the 2D-3D matches that was previously untapped.

5. Conclusions

In this paper we have presented a new 2D-3D geometric constraint and its application to visual localization. Albeit our outlier filter based on toroidal constraints is simple, it effectively removes many outliers and performs on par with more involved approaches. The proposed solution makes no strong assumptions on the data and thus is widely applicable. Therefore, it can be used as a drop-in solution in any localization pipeline to greatly increase the performance for difficult cases. Furthermore, even other pipelines that already employ a tailored approach for city-wide localization [22, 26, 28] can potentially benefit from this newly derived constraint. The constraint presented can thus be regarded as a new additional and meaningful geometric insight useful for the localization task.

Acknowledgements. This research was funded by Google Tango.

References

[1] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-Time Self-Localization from Panoramic Images on Mobile Devices. In *ISMAR*, 2011.

[2] F. Camposeco, T. Sattler, and M. Pollefeys. Non-Parametric Structure-Based Calibration of Radially Symmetric Cameras. In *ICCV*, 2015.

[3] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal Solvers for Generalized Pose and Scale Estimation from Two Rays and One Point. In *ECCV*, 2016.

[4] S. Choudhary and P. J. Narayanan. Visibility Probability Structure from SfM Datasets and Applications. In *ECCV*, 2012.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *PAMI*, 29(6):1052–1067, 2007.

[6] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust Fitting for Multiple View Geometry. In *ECCV*, 2012.

[7] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *ECCV*, pages 141–153. Springer, 2008.

[8] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection. In *CVPR*, 2014.

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

[10] D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002.

[11] R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994.

[12] M. Havlena and K. Schindler. VocMatch: Efficient Multiview Correspondence for Structure from Motion. In *ECCV*, 2014.

[13] V. Larsson, J. Fredriksson, C. Toft, and F. Kajl. Outlier Rejection for Absolute Pose Estimation with Known Orientation. In *BMVC*, 2016.

[14] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *ECCV*, 2010.

[15] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *ECCV*, 2014.

[18] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615 –1630, 2005.

[19] M. Muja and D. G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *VISAPP*, 2009.

[20] F. Pfeuffer, M. Stiglmayr, and K. Klamroth. Discrete and geometric Branch and Bound algorithms for medical image registration. *Annals of Operations Research*, 196(1):737–765, 2012.

[21] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In *ICCV*, 2015.

[22] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI*, 2016 (accepted).

[23] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012.

[24] J. L. Schönberger and J.-M. Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, 2006.

[26] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *PAMI*, 2016 (accepted).

[27] C. Sweeney, T. Sattler, M. Turk, T. Hollerer, and M. Pollefeys. Optimizing the Viewing Graph for Structure-from-Motion. In *ICCV*, 2015.

[28] B. Zeisl, T. Sattler, and M. Pollefeys. Camera Pose Voting for Large-Scale Image-Based Localization. In *ICCV*, 2015.