

# Optimal Relative Pose with Unknown Correspondences

Johan Fredriksson<sup>1</sup>, Viktor Larsson<sup>1</sup>, Carl Olsson<sup>1</sup> and Fredrik Kahl<sup>1,2</sup>

<sup>1</sup>Lund University, Sweden    <sup>2</sup>Chalmers University of Technology, Sweden  
 {johanf, viktorl, calle}@maths.lth.se    fredrik.kahl@chalmers.se

## Abstract

*Previous work on estimating the epipolar geometry of two views relies on being able to reliably match feature points based on appearance. In this paper, we go one step further and show that it is feasible to compute both the epipolar geometry and the correspondences at the same time based on geometry only. We do this in a globally optimal manner. Our approach is based on an efficient branch and bound technique in combination with bipartite matching to solve the correspondence problem. We rely on several recent works to obtain good bounding functions to battle the combinatorial explosion of possible matchings. It is experimentally demonstrated that more difficult cases can be handled and that more inlier correspondences can be obtained by being less restrictive in the matching phase.*

## 1. Introduction

The ability to compute the relative pose between two cameras is of fundamental importance in multiple view geometry. Almost all structure from motion systems rely on this subproblem for building an overall reconstruction. Efficient minimal solvers [17] as well as globally optimal branch and bound methods [10] exist. Like most other multiple view geometry problems, relative orientation is typically computed from a set of image point correspondences which are generated using image appearance alone. To obtain reliable correspondences the matching criterion is quite restrictive and will in practice only allow feature points with unique appearance. For example, [15] requires that the best matching score is significantly better than the second best one. As a consequence the number of available point correspondences can be small in scenes with ambiguous and/or repeated texture. Despite a restrictive matching criterion one has to expect that a portion of the feature points are incorrectly matched due to noise and appearance changes. Since these mismatches do not obey the Gaussian noise assumption they can be devastating to the reconstruction quality. To detect and remove such outliers, minimal solvers [17] in combination with RANSAC [6] are often employed. While it is very efficient in cases with a relatively low number of outliers RANSAC does not offer any optimality guarantees. Recently it has been shown that in many applica-

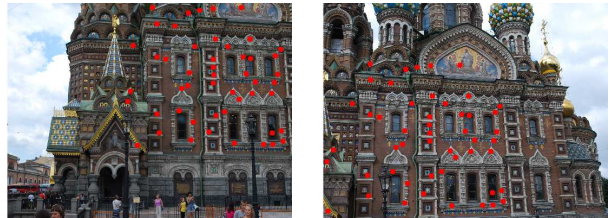


Figure 1: Example of two images of the same scene. The goal is to recover both the relative pose and the correct matching between the points.

tions RANSAC tends to fail when the number of outliers is high [13, 4, 7]. To address this problem researchers have started to consider methods that are guaranteed to find the largest possible inlier set. In [7] a method for determining the relative translation that maximizes the number of inliers is presented. The approach has a worst case complexity of  $\mathcal{O}(n^2 \log(n))$  where  $n$  is the number of putative matches. In [8] the same problem is addressed in a branch and bound framework. While the worst case complexity is exponential it is shown to outperform [7] in practice. For the full relative pose problem, [5] and [19] propose methods that also maximize the number of inliers.

While the above approaches all assume unique putative matches, in this paper, we take a step further and consider problems with multiple possible matches. In the extreme case, we consider the search space of all possible matching pairs. This gives a way to aid the matching in settings with ambiguous texture by incorporating geometric knowledge into the matching process. If the camera motion is known then the search for point correspondences is reduced to a search over known epipolar lines [16]. On the other hand, if the motion is unknown then we need to solve for geometry and point correspondences simultaneously, which is much harder. We propose a simple technique for solving this problem. To our knowledge it has previously not been attempted. Compared to regular inlier maximization there are two difficulties that have to be overcome. First, the space of matches to search grows quickly when we allow more than just the best match for each point. Second, with several feasible matches for each point we need to enforce one-to-one constraints. Our approach can be seen as a branch and bound method in two levels. Rather than searching over all

model parameters like in standard branch and bound, we first consider only relative translation estimation with unknown correspondences at the lower level. By solving a bipartite matching problem we create a bounding function that can be evaluated very rapidly. The above subproblem is then used at the higher level in order to estimate rotation parameters.

From a practical point of view the enlarged space of possible matches introduces solution ambiguities. Given an epipolar line any point lying on it is a potential match. Hence points can switch matches along the epipolar line without effecting the camera motion yielding multiple solutions.

The literature dealing with geometric vision problems with unknown matches is sparse. A method for subspace clustering is proposed in [12]. In [3] a branch and bound framework is applied to estimate in-plane Euclidean transformations and globally optimal matching solutions. In [14] a method for estimating 3D-3D rigid transformation is presented. While the approach effectively searches a 6-dimensional space its applicability is somewhat limited since it requires that all points in the source and target point-clouds have a match. An integer programming approach for geometric problems with linear residuals is presented in [2]. The approach can in principle be applied to relative pose with affine cameras but with an algebraic error function. The unknown correspondence problem with known relative motion is dealt with in [16].

### 1.1. Introductory example

In this introductory example we study the extreme case where we use no image information and hence each point in one image can be matched to every point in the other. In Figure 1 we show two images of the same scene. Applying our method to this problem, we are able to recover the relative pose in an optimal manner, see Figure 2. Although the global optimum of the estimated relative motion corresponds to the expected (true) camera motion, two of the correspondence pairs are mismatched (red lines). We will later show that this is due to an inherent ambiguity in the problem, which cannot be resolved using geometric constraints alone.

### 1.2. Problem statement

In this work we use the spherical camera model where a 3D point  $\mathbf{X}$  is projected into a camera  $(R, \mathbf{t})$  by simply normalizing the transformed point, i.e.

$$\mathbf{v} = \frac{R(\mathbf{X} - \mathbf{t})}{\|R(\mathbf{X} - \mathbf{t})\|}.$$

For spherical cameras the most natural error metric is the angular reprojection error.

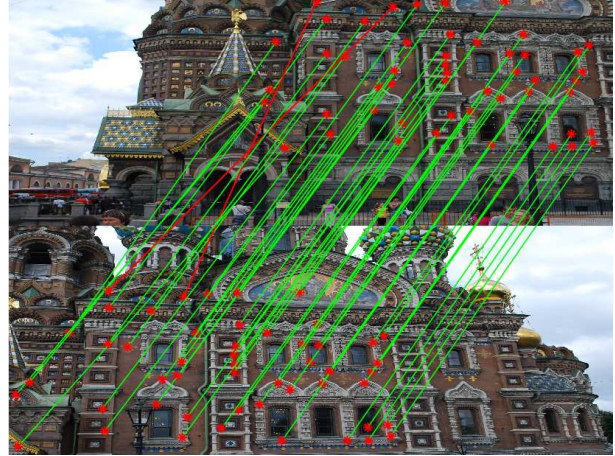


Figure 2: The correspondences found by the proposed method. The inliers have less than 0.1 degrees angular error. The red lines correspond to incorrectly matched points (compared to a manually created ground truth matching).

**Definition 1.** Given an error threshold  $\epsilon \in \mathbb{R}^+$ , relative rotation  $R$  and translation  $\mathbf{t}$ , the corresponding image point-pair  $(\mathbf{v}_1, \mathbf{v}_2)$  is called an **inlier** if there exists some  $\mathbf{X} \in \mathbb{R}^3$  such that<sup>1</sup>

$$\angle(\mathbf{v}_1, \mathbf{X}) \leq \epsilon \quad \text{and} \quad \angle(\mathbf{v}_2, R(\mathbf{X} - \mathbf{t})) \leq \epsilon. \quad (1)$$

The classical relative pose problem can then be stated.

**Problem 1.** Given a threshold  $\epsilon$  and a set of putative point matches between two images find the relative rotation  $R$  and translation  $\mathbf{t}$  which maximize the number of inliers.

Let  $I_1$  be the index set for the image points in the first image and  $I_2$  the index set for the second image. Define  $E \subseteq I_1 \times I_2$  to be the set of possible correspondences. If we allow every point in the first image to match to every point in the other image we have  $E = I_1 \times I_2$ .

**Definition 2.** A set of putative matches  $M \subset E$  is **feasible** if each  $i \in I_1$  appears in at most one pair  $(i, j) \in M$  and each  $j \in I_2$  appears in at most one pair  $(i, j) \in M$ .

Now we can state the problem we solve in this paper.

**Problem 2.** Given a set of possible correspondences  $E$  and  $\epsilon > 0$ , find the relative rotation  $R$  and translation  $\mathbf{t}$  which have a maximal set of feasible inlier matches  $M \subset E$ .

## 2. Known relative rotation

First we consider the case when the relative rotation is known and we only want to estimate the relative translation.

<sup>1</sup> $\angle(\mathbf{a}, \mathbf{b})$  denotes the angle between the vectors  $\mathbf{a}$  and  $\mathbf{b}$ .

Problems with known relative rotation occur naturally in applications such as robotics where rotation is either known to be fixed or can be estimated through other sensors.

This subproblem will also be used as a building block when performing branch and bound over the rotation space in Section 3. Since there is a scale ambiguity we can without loss of generality assume that the translation is of unit length. Similarly by rotating the image points in the second image we can assume the rotation to be identity, i.e.  $R = I$ .

### 2.1. Epipolar geometry under translation

The epipolar geometry for known relative rotation has been thoroughly investigated in prior works such as [10, 7, 8] and we only state the needed results here.

If the camera only has undergone a pure translation, then the epipolar constraint is reduced to the image points  $v_1$  and  $v_2$  and the translation  $t$  being coplanar with the origin. Since we assumed  $t$  to be of unit length this is equivalent to requiring that  $v_1, v_2$  and  $t$  lie on a great circle on the unit sphere. If we allow for a reprojection error  $\epsilon$ , then the translation must lie inside a wedge determined by the  $\epsilon$ -cones around  $v_1$  and  $v_2$ . This is illustrated in Figure 3.

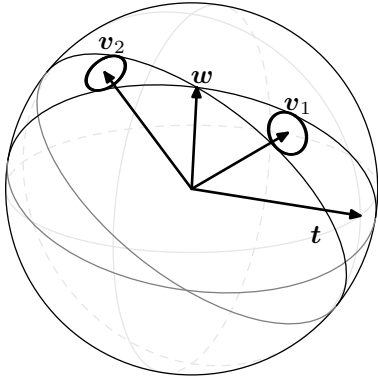


Figure 3: The image points  $v_1$  and  $v_2$  (arrows in the image) determine two wedges on the sphere. The wedges are formed by constructing great circles tangent to the  $\epsilon$ -cones around the image points. If the translation  $t$  lies inside the wedges it will have the image pair is an inlier. Only one of the wedges correspond to a reconstruction with the 3D point  $X$  in front of both cameras.

For a point pair  $(v_1, v_2)$  the normals of the two great circles which define the wedge are given by

$$n^\pm = \sin(\beta/2)(n \times w) \pm \cos(\beta/2)n \quad (2)$$

where  $\alpha$  is the angle between the image points and  $\beta$  satisfies

$$\sin(\beta/2) = \sin(\epsilon) / \sin(\alpha/2) \quad (3)$$

and

$$w = \frac{v_1 + v_2}{\|v_1 + v_2\|}, \quad n = \frac{v_1 \times v_2}{\|v_1 \times v_2\|}. \quad (4)$$

Checking if a translation  $t$  has the pair  $(v_1, v_2)$  as an inlier is then simply done by verifying that

$$n^+ \cdot t \geq 0 \quad \text{and} \quad n^- \cdot t \geq 0. \quad (5)$$

### 2.2. Branch and bound for translation estimation

In [8] the authors perform branch and bound on the unit sphere to solve the relative translation problem. In their approach they subdivide the unit sphere into spherical triangles. To compute upper bounds for any triangle on the sphere they count the number of wedges which either intersect the triangle or completely enclose it. This will of course be an upper bound for the number of wedges any translation  $t$  inside the triangle can satisfy. Finding the intersection of the wedges and a triangle is simple due to the fact that the edges of spherical triangles lie along great circles. The intersection points are easily computed by forming the cross products of the normals. To find a lower bound they simply evaluate the center point of the triangle.

### 2.3. Bounding the optimal feasible inlier set

In the case of multiple potential matches the lower bound computed in [8] is not valid since it does not guarantee one-to-one matches. The authors of [8] also presented a heuristic improvement to address unknown correspondences. However it only guarantees that points in one of the images are uniquely assigned.

Similarly to [8] we perform branch and bound on the sphere using spherical triangles representing a set of feasible translations. For each subdivision of the search space we need to compute an upper and lower bound for the size of the largest feasible inlier set any translation within the triangle can have. To compute the upper bound we find the largest feasible inlier set under the assumption that all wedges which either intersect the triangle or completely enclose it are inliers. Clearly this will be an upper bound for the largest feasible set for any translation inside the triangle. To find a lower bound, we check which wedges contain the center point of the triangle and compute the largest feasible set for these.

To ensure that we only search feasible matchings that fulfill the one-to-one constraint we construct a bipartite graph where the nodes in the graph correspond to the image points. For each possible inlier wedge we add an edge between the corresponding points. The problem is then reduced to finding a maximum-cardinality matching in this bipartite graph, see Figure 4.

Matching in bipartite graphs is a well studied problem which can be efficiently solved. In the implementation we use the Hopcroft-Karp [11] algorithm but any standard solver can be applied. The Hopcroft-Karp algorithm has a worst case complexity of  $\mathcal{O}(|E|\sqrt{|V|})$  but has been shown for sparse graphs to have an average complexity of

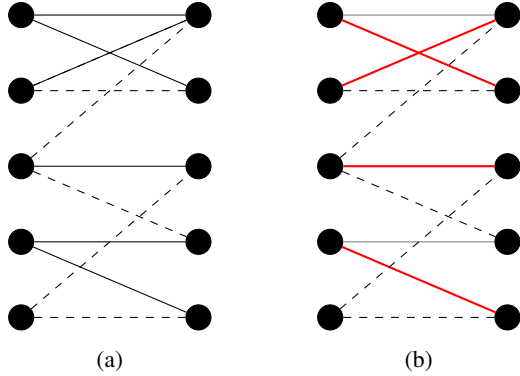


Figure 4: Computing bounds for a triangle. The left column of nodes correspond to points in the first image and the right column to the second image. For each subdivision of the search space a subset of the wedges are possible inliers. The possible inlier correspondences are illustrated as solid lines and the correspondences which can be ruled out are illustrated as dashed lines. For the solid lines a maximal set of independent edges are selected. These are illustrated by the red lines in (b).

$\mathcal{O}(|E| \log |V|)$  [1]. Typically for small spherical triangles the number of possible inlier wedges which are contradictory will be small and the matching problem can be solved extremely quickly.

## 2.4. Example with multiple matches

It is common practice in structure from motion to include a point pair in the set of possible correspondences if the point from the first image has no other good candidates [15]. This is often a good idea since it removes a lot of contradictory point pairs and decreases the amount of outliers. Unfortunately in scenes with repeated structures this often results in few matches between the images, which in turn could lead to low quality solutions. If we instead include all matches which are sufficiently good we do not need to discard possibly correct matches. While the problem becomes more difficult to solve, the optimal solution has the potential to contain more inliers.

In this example we compare matching with Lowe’s criterion<sup>2</sup> with simply including the 10000 best candidate matches. Note that this means that some points might have multiple matches in the other image. Figure 5 shows an example of an image pair with a lot of windows and the possible matches found using the two matching approaches. The inlier matches for the optimal solutions are shown in Figure 6.

<sup>2</sup>The threshold was set to 0.6.

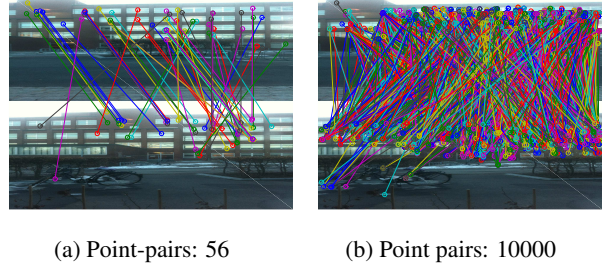


Figure 5: *Left*: Normal SIFT-matching with Lowe’s criterion 0.6. *Right*: The 10000 best candidate matches.

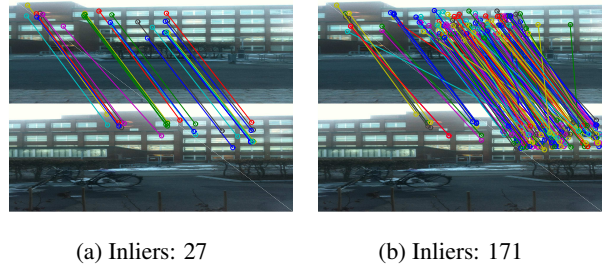


Figure 6: *Left*: The optimal number of inliers for the normal SIFT-matching, using [8]. The running time was 20 ms and Lowe’s criterion 0.6. *Right*: The optimal number of inliers when solving the problem using the proposed method on the 10000 best matches. The running time was 7 s. The  $\epsilon$  corresponds to 3 pixels. The incorrect inliers lie on the epipolar lines and in the model these are indistinguishable from the correct inliers.

## 2.5. Comparison to other methods

In this section we evaluate our method for translation estimation on the dataset of Fredriksson et al. [7]. The dataset contains 136 image pairs taken without any relative rotation. For each image pair we selected the best candidate matches. We ran the translation estimation multiple times with increasing amounts of matches included.

The minimal case for translation estimation requires two points,

$$\mathbf{t} = (\mathbf{v}_1 \times \mathbf{v}_2) \times (\mathbf{v}'_1 \times \mathbf{v}'_2). \quad (6)$$

We compare our method against two different versions of RANSAC. In the first version we randomly select two point correspondences and evaluate the number of inlier matchings without regard for the feasibility requirement. In the other version (RANSAC-bpt) we compute the number of actual inliers by solving a bipartite matching problem in each iteration. For the two RANSAC algorithms we run 500 and 50000 iterations, which corresponds to a 99% chance of finding at least one minimal case where both point pairs are inliers in situations with 90% and 99% outliers respectively.

We also compare with the method from Fredriksson et

al. [8]. In [8] the authors also presented a heuristic improvement for dealing with unknown correspondences. We include results both with and without this heuristic. Table 1 shows the average inlier count for different number of matches included. Figure 7 shows the running times for all methods.

Matches	1k	5k	10k	20k	50k
Our	<b>143</b>	<b>289</b>	<b>385</b>	<b>511</b>	<b>724</b>
RANSAC-500	139	262	332	410	515
RANSAC-50k	142	284	375	469	559
RANSAC-500 (bpt)	140	267	336	434	617
RANSAC-50k (bpt)	<b>143</b>	287	383	507	716
[8] (without heuristic)	142	285	375	472	562
[8] (with heuristic)	<b>143</b>	288	383	509	715

Table 1: The average number of inliers over the 136 image pairs for different number of included matches. Note that our result is optimal.

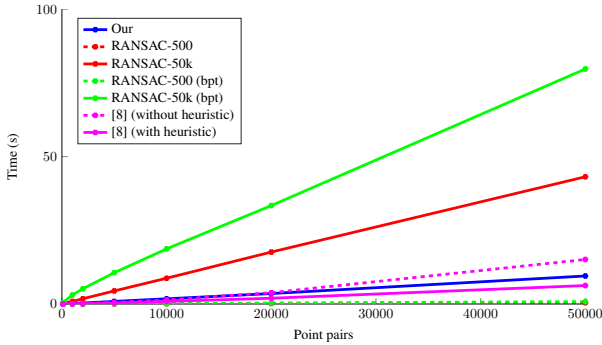


Figure 7: The average running times for different number of point pairs included.

### 3. Full relative pose

Now we consider the the full relative pose problem with unknown correspondences. The method we propose is based on performing a branch and bound over rotation space similarly to [10]. For each subdivision of rotation space we compute the bounds using the method presented in Section 2.

#### 3.1. Branch and bound over rotation space

To handle the full relative pose problem we use the rotation bounds derived by Hartley and Kahl in [10]. They optimally solve the relative pose problem without outliers by using branch and bound over the space of rotations.

Each rotation can be represented by the axis of rotation and the angle. This allows the set of rotations to be parameterized by the set  $\{\mathbf{r} \in \mathbb{R}^3, \|\mathbf{r}\| \leq \pi\}$  where the rotation

angle is represented by the length of the vector. For ease of implementation [10] uses this parametrization but searches over the cube  $[-\pi, \pi]^3$ . To obtain bounds the following result is used.

**Lemma 1.** Let  $\mathbf{r}_1$  correspond to the rotation  $R_1$  and  $\mathbf{r}_2$  to  $R_2$ . Then<sup>3</sup>

$$d(R_1, R_2) \leq \|\mathbf{r}_1 - \mathbf{r}_2\|. \quad (7)$$

This means that if we have a cube inside  $[-\pi, \pi]^3$  with side length  $\sigma$  centered at  $R_c$  then any rotation  $R$  inside the cube satisfies

$$d(R_c, R) \leq \frac{\sqrt{3}}{2}\sigma. \quad (8)$$

To find bounds for any relative pose with a rotation inside this cube [10] transforms the image points in the second image by  $R_c^T$  and search for the translation while allowing for a larger error in the second image,

$$\epsilon_2 = \epsilon + \frac{\sqrt{3}}{2}\sigma. \quad (9)$$

This additional error accounts for the uncertainty in the rotation. Note that this larger threshold is only used for the second image point. In [10] the bounds are computed by solving a linear program to determine the optimal translation. This is possible since correspondences are known with no outliers.

#### 3.2. Extension to unknown correspondences

We extend the framework of [10] to handle unknown correspondences. The idea is to perform branch and bound over rotation space similarly to [10] but to compute the bounds using the method presented in Section 2. This is possible since the translation estimation can be solved very quickly.

During the outer branch and bound (over rotations) we need to compute the bounds for a given rotation cube in  $[-\pi, \pi]^3$ . To find the upper bound we perform an inner branch and bound over translation space where we account for the uncertainty in rotation. Figure 8 shows a wedge where we allow for a larger error in the second image point (as in (9)). In [10] it is shown that the normals to the wedges where the image points have different  $\epsilon$  are given by

$$\mathbf{n}^\pm = \sin(\beta/2)(\mathbf{n} \times \mathbf{w}) \pm \cos(\beta/2)\mathbf{n} \quad (10)$$

where  $\alpha$  is the angle between the image points,

$$\mathbf{w} = \frac{\sin(\epsilon_2)\mathbf{v}_1 + \sin(\epsilon_1)\mathbf{v}_2}{\|\sin(\epsilon_2)\mathbf{v}_1 + \sin(\epsilon_1)\mathbf{v}_2\|}, \quad \mathbf{n} = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{\|\mathbf{v}_1 \times \mathbf{v}_2\|} \quad (11)$$

and  $\beta$  satisfies

$$\sin^2(\beta/2) = \frac{\sin^2(\epsilon_1) + 2\sin(\epsilon_1)\sin(\epsilon_2)\cos(\alpha) + \sin^2(\epsilon_2)}{\sin^2(\alpha)}.$$

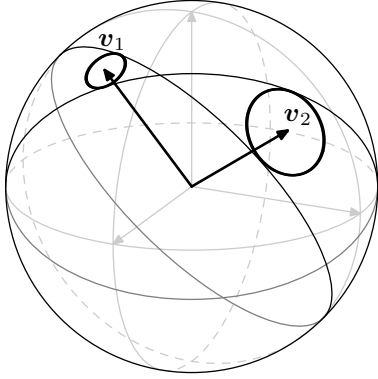


Figure 8: The wedges of possible translations when we allow for larger reprojection error in the second image point.

To find a lower bound we solve the translation problem again but with the original wedges (same  $\epsilon$  for both images). This corresponds to finding the optimal translation for the rotation in the center of the cube.

### 3.3. Evaluation on synthetic data

In this section we evaluate the method for full relative pose on synthetic data. In these experiments we see how different error thresholds affect the quality of the solution. We also compare stopping the algorithm when some specific optimality gap (difference between upper bound and lower bound) is achieved. The optimality gap gives a bound for the maximum difference in inliers between the returned solution and the optimal solution.

#### 3.3.1 Narrow field of view

In this experiment we test the method on problems with narrow field of view. Since the image points are closer together, the narrow fields of view are more difficult to solve using branch and bound based algorithms. We synthesize 50 different 3D points, located uniformly in a cube of side length four around the origin. The cameras are placed randomly at distance four to the origin, resulting in about  $60^\circ$  field of view. We add Gaussian noise with standard deviation  $0.033^\circ$  to the image points. In Table 2 and Table 3 the results can be seen for  $\epsilon = 0.5^\circ$  and  $\epsilon = 0.25^\circ$ .

#### 3.3.2 Narrow field of view: Planar scene

In this experiment we use the same experimental setup as above, but all the points are now in the same plane, defined by the square  $(\pm 1, \pm 1, \sqrt{6})$ . The first camera is located in the origin and the second at the unit circle parallel to the plane. We use 50 synthesized 3D points, randomly

<sup>3</sup>Here  $d(\cdot, \cdot)$  denotes the angle between the rotations.

$\epsilon = 0.5^\circ$				
Gap	Rotation	Translation	Mean time	Median time
5	$81^\circ$	$35^\circ$	71s	66s
3	$20^\circ$	$13^\circ$	128s	68s
0	$3.6^\circ$	$3.4^\circ$	686s	68s

Table 2: The results for the proposed method on 10 synthetic narrow field of view instances. The synthetic data contains 50 3D points. The error threshold  $\epsilon$  is  $0.5^\circ$ .

$\epsilon = 0.25^\circ$				
Gap	Rotation	Translation	Mean time	Median time
5	$3.7^\circ$	$2.6^\circ$	879s	189s
3	$2.9^\circ$	$1.8^\circ$	881s	189s
0	$2.2^\circ$	$1.5^\circ$	1015s	201s

Table 3: The results for the proposed method on 10 synthetic narrow field of view instances. The synthetic data contains 50 3D points. The error threshold  $\epsilon$  is  $0.25^\circ$ .

distributed in the plane. Noise are added to the image points as in the previous case. In Table 4 and Table 5 the results can be seen for  $\epsilon = 0.5^\circ$  and  $\epsilon = 0.25^\circ$ .

$\epsilon = 0.5^\circ$				
Gap	Rotation	Translation	Mean time	Median time
5	$31^\circ$	$28^\circ$	43s	45s
3	$31^\circ$	$28^\circ$	42	45s
0	$10^\circ$	$12^\circ$	83s	45s

Table 4: The results for the proposed method on 10 synthetic narrow field of view instances, where all points are located in the same plane. The synthetic data contains 50 3D points. The error threshold  $\epsilon$  is  $0.5^\circ$ .

$\epsilon = 0.25^\circ$				
Gap	Rotation	Translation	Mean time	Median time
5	$12^\circ$	$16^\circ$	91	94s
3	$8^\circ$	$12^\circ$	137s	103s
0	$5^\circ$	$9^\circ$	218s	161s

Table 5: The results for the proposed method on 10 synthetic narrow field of view instances, where all points are located in the same plane. The synthetic data contains 50 3D points. The error threshold  $\epsilon$  is  $0.25^\circ$ .

### 3.3.3 Omni-directional

In this synthetic experiment we instead use omni-directional cameras. We again use 50 points, this time they are randomly distributed on the sphere of radius two. The camera centres are randomly distributed on the unit sphere. The added noise here is the same as in the narrow field of view. The result for  $\epsilon = 1^\circ$  and  $\epsilon = 0.5^\circ$  can be seen in Table 6 and Table 7. Note that for the same error threshold and gap, the solution to the omni-directional problem has much higher quality than the narrow field of view.

$\epsilon = 1^\circ$				
Gap	Rotation	Translation	Mean time	Median time
10	72°	31°	121s	125s
5	3.2°	3.4°	167s	130s
3	2.5°	2.1	167s	133s
0	1.7°	2.0°	168s	133s

Table 6: The results for the proposed method on 10 synthetic omni-directional instances. The problem has 50 points.

$\epsilon = 0.5^\circ$				
Gap	Rotation	Translation	Mean time	Median time
10	1.6°	2.5°	266s	241s
5	1.2°	1.4°	264s	241s
3	1.1°	1.0	264s	241s
0	0.9°	1.0°	266s	249s

Table 7: The results for the proposed method on 10 synthetic omni directional instances. The problem has 50 points.

### 3.4. Ambiguities along epipolar lines

We now again consider the image pair shown in Figure 1 (Section 1.1). For this image pair there exist multiple optimal solutions and Figure 9 shows another optimal solution with the same relative rotation and translation but with different point matches. This ambiguity occurs since there can be multiple possible matches along the epipolar lines for the optimal rotation and translation. See Figure 10.

### 3.5. Evaluation on the KITTI benchmark dataset

Next we evaluate our method on the KITTI dataset [9]. The dataset consists of images taken from a camera mounted on the roof of a car. Since the car typically only moves in a plane we can restrict our search to rotations around the  $y$ -axis.

The dataset also contains ground truth poses obtained through other sensors. We consider four image sequences



Figure 9: Another optimal solution. Note that the solution has the same rotation and translation as in Figure 2 but different point matches.

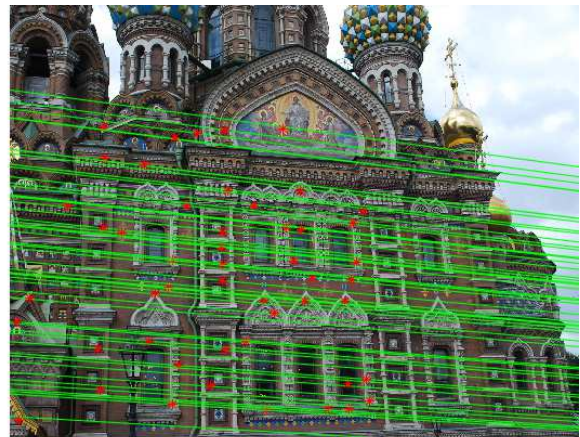


Figure 10: The epipolar lines in the second image. The ambiguous matches in Figure 9 are along the epipolar lines.

which were taken 2011-09-26. Some example images from one of the sequences can be seen in Figure 11.

For each sequence all image pairs in the sequence which differed by ten frames were used. We consider two types of matching strategies.

The first strategy is a standard symmetric matching strategy where for each point we find the best match in the other image. The match is kept if it is sufficiently good<sup>4</sup>, it is sufficiently better than the next best match (Lowe's criterion) and the match was symmetric. The symmetry requirement ensures that there are no ambiguous matches.

If there is repeated texture in the scene a point might have multiple good matches in the other image. Due to the ambiguous matching such points are discarded in the first

<sup>4</sup>Determined by a threshold on the angle between the SIFT descriptors.



Figure 11: Example images from the KITTI vision dataset. For the experiment the grayscale images were used.

matching strategy. In the second strategy we instead allow points to have multiple matches in the other image. For each point we include all matches that are sufficiently good and such that the worst of the included matches satisfy Lowe’s criterion to the next best match. In Table 8 some statistics for the sequences and matching strategies can be found.

Sequence	images	I		II
		matches	matches	amb. matches
0001	108	65.4	82.2	9.1
0005	154	62.4	79.3	9.3
0015	297	61.9	83.2	12.9
0048	22	137.8	179.6	33.4

Table 8: The number of images and the average number of matches generated by the two strategies. For the second strategy we show also the average number of ambiguously matched points.

Using the proposed method we estimated the rotation and translation for the matches generated by both strategies. The resulting errors can be seen in Table 9 and Table 10. For comparison we show the result from 100 iteration of RANSAC using the 3 point solver from [18].

Note that for the second sequence (0005) we see a significant improvement when allowing for multiple matches. For the other sequences the results are comparable. The average running time for our method for the two strategies were 1.3 seconds and 2.1 seconds.

Sequence	I		II	
	Our	RANSAC	Our	RANSAC
0001	1.97	0.53	1.20	<b>0.53</b>
0005	5.31	5.23	<b>2.87</b>	4.62
0015	1.71	<b>0.45</b>	0.97	0.48
0048	0.62	0.75	<b>0.61</b>	0.81

Table 9: The average rotation error in degrees for the two matching strategies for the different sequences.

Sequence	I		II	
	Our	RANSAC	Our	RANSAC
0001	7.79	2.16	4.77	<b>2.10</b>
0005	13.81	21.61	<b>9.99</b>	20.21
0015	9.99	<b>4.00</b>	5.87	4.23
0048	2.17	4.19	<b>2.13</b>	3.75

Table 10: The average translation error in degrees for the two matching strategies for the different sequences.

## 4. Conclusions

In this paper, we have demonstrated that it is possible to optimally compute the relative pose with unknown correspondences. We also show that keeping ambiguous point matches can improve the accuracy of the estimated camera motion and increase the number of inliers. We have also shown that problem instances can be very different in nature, ranging from very hard instances with ambiguous solutions and long running times, to well-posed situations where our algorithm returns a solution within a couple of seconds.

## 5. Acknowledgments

This work has been funded by the Swedish Research Council (grant no. 2012-4213) and the Crafoord Foundation.

## References

- [1] H. Bast, K. Mehlhorn, G. Schafer, and H. Tamaki. Matching algorithms are fast in sparse random graphs. *Theory of Computing Systems*, 2006. 4
- [2] J. Bazin, H. Li, I. S. Kweon, C. Démonceaux, P. Vasseur, and K. Ikeuchi. A branch-and-bound approach to correspondence and grouping problems. *Pattern Analysis and Machine Intelligence (PAMI)*, 2013. 2
- [3] T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding (CVIU)*, 2003. 2
- [4] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Tractable algorithms for robust model estimation. *International Journal of Computer Vision (IJCV)*, 2015. 1



- [5] O. Enqvist and F. Kahl. Two view geometry estimation with outliers. In *British Machine Vision Conference (BMVC)*, 2009. [1](#)
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 1981. [1](#)
- [7] J. Fredriksson, O. Enqvist, and F. Kahl. Fast and reliable two-view translation estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. [1](#), [3](#), [4](#)
- [8] J. Fredriksson, V. Larsson, and C. Olsson. Practical robust two-view translation estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. [1](#), [3](#), [4](#), [5](#)
- [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. [7](#)
- [10] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision (IJCV)*, 2009. [1](#), [3](#), [5](#)
- [11] J. E. Hopcroft and R. M. Karp. An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 1973. [3](#)
- [12] P. Ji, H. Li, M. Salzmann, and Y. Dai. Robust motion segmentation with unknown correspondences. In *European Conference on Computer Vision (ECCV)*. 2014. [2](#)
- [13] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *International Conference on Computer Vision (ICCV)*, 2009. [1](#)
- [14] H. Li and R. Hartley. The 3d-3d registration problem revisited. In *International Conference on Computer Vision (ICCV)*, 2007. [2](#)
- [15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004. [1](#), [4](#)
- [16] J. Maciel and J. Costeira. Robust point correspondence by concave minimization. *Image and Vision Computing (IVC)*, 2002. [1](#), [2](#)
- [17] D. Nistér. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence (PAMI)*, 2004. [1](#)
- [18] C. Sweeney, J. Flynn, and M. Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 483–490. IEEE, 2014. [8](#)
- [19] J. Yang, H. Li, and Y. Jia. Optimal essential matrix estimation via inlier-set maximization. In *European Conference on Computer Vision (ECCV)*. 2014. [1](#)