Practical Robust Two-View Translation Estimation

Johan Fredriksson, Viktor Larsson, Carl Olsson
Centre for Mathematical Sciences
Lund University
{johanf,viktorl,calle}@maths.lth.se

Abstract

Outliers pose a problem in all real structure from motion systems. Due to the use of automatic matching methods one has to expect that a (sometimes very large) portion of the detected correspondences can be incorrect. In this paper we propose a method that estimates the relative translation between two cameras and simultaneously maximizes the number of inlier correspondences.

Traditionally, outlier removal tasks have been addressed using RANSAC approaches. However, these are random in nature and offer no guarantees of finding a good solution. If the amount of mismatches is large, the approach becomes costly because of the need to evaluate a large number of random samples. In contrast, our approach is based on the branch and bound methodology which guarantees that an optimal solution will be found. While most optimal methods trade speed for optimality, the proposed algorithm has competitive running times on problem sizes well beyond what is common in practice. Experiments on both real and synthetic data show that the method outperforms state-of-the-art alternatives, including RANSAC, in terms of solution quality. In addition, the approach is shown to be faster than RANSAC in settings with a large amount of outliers.

1. Introduction

Two-view relative pose estimation is one of the cornerstone problems in computer vision. Most structure from motion pipelines solve it as a subproblem in order to build an overall reconstruction. In this paper, we study the special case of translation estimation between two views. Either the camera is known not to rotate or the relative rotation can be estimated by other means. There are many situations where the camera undergoes pure translation, for example in robotic applications. In addition, known rotation formulations have been shown to be useful subproblems for the structure from motion pipeline. They were first addressed in an algebraic framework in [16], and later with reprojection errors in the L_∞-framework of [10]. However, these works assume that there are no outlier correspondences.

In this paper we are interested in estimation and outlier rejection in the presence of large amounts of outliers. For multiple view geometry problems, the predominant approach for dealing with outliers is RANSAC [5, 14, 9]. The method has been applied to related geometry problems such as relative pose estimation with rotation around one axis [6, 13]. This formulation is motivated by the fact that many modern cell phones are able to directly measure the gravitation vector. A disadvantage with these methods is the randomness of the results and that the solution space is restricted to the hypotheses given by minimal subsets of the data. Furthermore, in settings with a large amount of mismatches this approach becomes costly because of the need to evaluate a large number of random samples. The approach offers no optimality guarantees meaning that even if there is a good solution the algorithm is not guaranteed to find it.

In [7], Fredriksson *et al.* gave an optimal algorithm for the two-view translation problem in the presence of outliers. For a given error threshold ϵ the algorithm finds a translation with the maximal amount of inlier correspondences, with a theoretical time complexity of $\mathcal{O}(n^2 \log n)$. It was demonstrated that in settings with large amounts of outliers the approach not only produces better results but is also faster than RANSAC.

In this paper we present a branch and bound approach that is significantly faster than [7] with the same kind of theoretical guarantees. The branch and bound methodology has been applied to a variety of geometry problems. In [8], a branch and bound framework is introduced that searches over all rotations. As a subroutine, the translation of the camera is estimated using linear programming. It is assumed that no outliers are present in the data. The linear program could in principle be replaced with our approach to obtain an algorithm that works in the presence of outliers. A related algorithm for dealing with outliers is given in [3]. In [11] a method for uncalibrated reconstruction based on an algebraic cost function and a mixed integer formulation is proposed. While these methods guarantee optimality of the solution they are often based on computationally costly search schemes, with worst case exponential running time, resulting in prohibitively slow algorithms.

In this paper we present a branch and bound approach tailored for the two-view translation problem that outperforms Fredriksson et al. [7]. The key to designing an effective formulation is to find strong bounding functions that can be evaluated efficiently. We achieve this by a division of the parameter space S^2 using spherical triangles. Checking feasibility of a correspondence within a parameter triangle reduces to computing a few intersections between great circles, which can be done extremely efficiently using a simple cross-product. We show that the method outperforms RANSAC in terms of quality of the solution and can in some cases be faster, in particular if the portion of outliers is high. In addition, the approach outperforms [7] with orders of magnitude while providing the same optimality guarantees.

1.1. Background

In this work we consider the spherical camera model which has previously been used in e.g. [8, 7]. For spherical cameras the projections are formed by first applying a Euclidean transformation which takes the 3D-points to the camera's coordinate system followed by normalization. The projection of a 3D-point $\boldsymbol{X}$ in a spherical camera $(R, \boldsymbol{t})$ is formed as

$$\boldsymbol{v} = \frac{R(\boldsymbol{X} - \boldsymbol{t})}{\|R(\boldsymbol{X} - \boldsymbol{t})\|}. \tag{1}$$

This means that the image points are on the unit sphere in $\mathbb{R}^3$. This is in contrast to a projective camera where the image points lie on a plane. This paper considers the problem of two-view estimation with known relative rotation. By rotating the image points we can w.l.o.g. assume that $R = I$, i.e. the projection of a 3D-point $\boldsymbol{X}$ in the two images is given by

$$\boldsymbol{v}_1 = \frac{\boldsymbol{X}}{\|\boldsymbol{X}\|} \quad \text{and} \quad \boldsymbol{v}_2 = \frac{\boldsymbol{X} - \boldsymbol{t}}{\|\boldsymbol{X} - \boldsymbol{t}\|}. \tag{2}$$

Now we can define what we mean by a point pair being an inlier.

Definition 1. *The corresponding point pair $(\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2)$ is called an **inlier** for the translation $\boldsymbol{t}$ if there exist some $\boldsymbol{X} \in \mathbb{R}^3$ such that*[1]

$$\angle(\hat{\boldsymbol{v}}_1, \boldsymbol{X}) \leq \epsilon \quad \textit{and} \quad \angle(\hat{\boldsymbol{v}}_2, \boldsymbol{X} - \boldsymbol{t}) \leq \epsilon \tag{3}$$

for a given error threshold $\epsilon \in \mathbb{R}^+$.

Next we state the problem we are interested in solving.

Problem 1. *Given a threshold ϵ and a set of putative point pairs between two images find the translation $\boldsymbol{t}$ which maximizes the number of inliers.*

Due to scale ambiguity we can restrict ourselves to translations of unit length.

2. Spherical epipolar geometry

For the case we consider, the epipolar constraint reduces to $\boldsymbol{t}$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ being coplanar. This means that there exists some $\boldsymbol{X} \in \mathbb{R}^3$ with projections $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ if and only if the translation $\boldsymbol{t}$ is coplanar with $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. If we allow for the reprojections to have an angular error of ϵ the translation must instead reside in one of two wedges determined by the point pair as is illustrated in Figure 1. Only one of the wedges corresponds to a valid reconstruction. The wedge can be described using two normals $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$, and contains all points such that the dot products with the normals are positive, i.e.

$$W = \{\boldsymbol{x} \in S^2 \mid \boldsymbol{n}^+ \cdot \boldsymbol{x} \geq 0,\ \boldsymbol{n}^- \cdot \boldsymbol{x} \geq 0\}. \tag{4}$$

The normals defining the wedge for the pair $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ are given by

$$\boldsymbol{n}^{\pm} = \sin(\beta/2)(\boldsymbol{n} \times \boldsymbol{w}) \pm \cos(\beta/2)\boldsymbol{n}, \tag{5}$$

where

$$\boldsymbol{w} = \frac{\boldsymbol{v}_1 + \boldsymbol{v}_2}{\|\boldsymbol{v}_1 + \boldsymbol{v}_2\|}, \quad \boldsymbol{n} = \frac{\boldsymbol{v}_1 \times \boldsymbol{v}_2}{\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\|}, \tag{6}$$

and α, β satisfies

$$\sin(\beta/2) = \sin(\epsilon)/\sin(\alpha/2), \quad \cos(\alpha) = \boldsymbol{v}_1 \cdot \boldsymbol{v}_2. \tag{7}$$

The complete derivations can be found in [7, 8].

3. Optimal estimation with branch and bound

To find the optimal translation we employ a branch and bound strategy. Branch and bound has been used extensively for geometric problems in the past [17, 8, 2, 15, 4] and is particularly suited for problems with low-dimensional solution spaces. The idea in branch and bound is to iteratively subdivide the search space and for each subdivision compute an upper bound for any solution candidate inside it. If the upper bound of any subdivision is lower than any solution found so far it can be safely discarded. Computing bounds and subdividing the space is then iterated until convergence.

3.1. Dividing the solution space

For translation estimation the solution space consists of the unit sphere S^2 in $\mathbb{R}^3$. For the first iteration we divide the unit sphere into eight equally sized spherical triangles

[1] $\angle(\boldsymbol{a}, \boldsymbol{b})$ denotes the angle between the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$

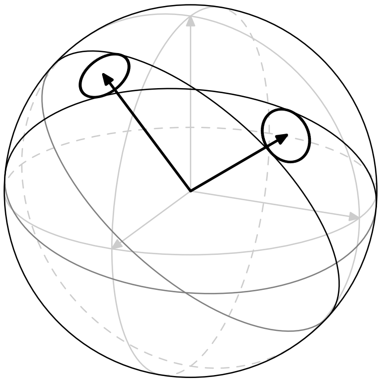


Figure 1: The pair $(\boldsymbol{v}_1,\ \boldsymbol{v}_2)$ determines two wedges formed by two great circles on the sphere. The great circles are tangent to the ϵ-cones around $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, respectively. Any translation candidate which lies inside the wedges will have the pair as an inlier. Only one of the wedges corresponds to a valid reconstruction.

covering the whole sphere. During the optimization we refine the partition of the search space by splitting triangles along the longest edge into smaller triangles, in a branch and bound fashion. Similar to a wedge, a spherical triangle on S^2 is bounded by the intersection of three planes,

$$T = \{\boldsymbol{x} \mid \boldsymbol{x} \in S^2,\ \boldsymbol{n}_k \cdot \boldsymbol{x} \geq 0, k = 1, 2, 3\}. \tag{8}$$

The subdivision of the search space is illustrated in Figure 2. In [4] the authors perform branch and bound on $S^2 \times S^2$ where they also subdivide the sphere using spherical triangles.

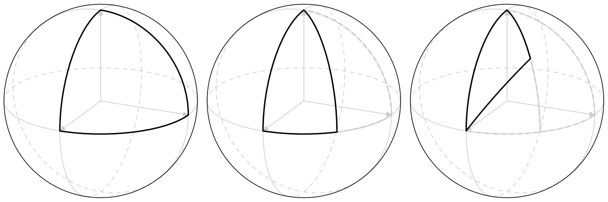


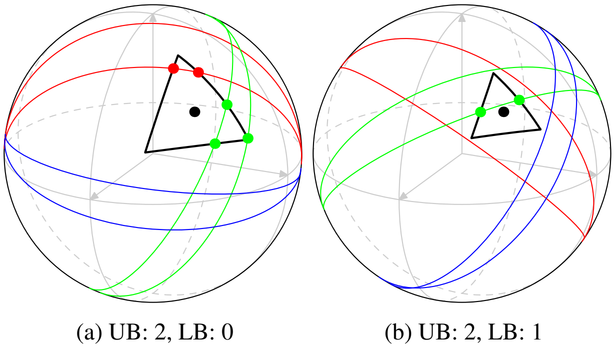
Figure 2: The search space is subdivided into triangles. Each triangle is recursively divided into smaller and smaller triangles.

3.2. Computing the bounds

In our branch and bound scheme we consider groups of translations represented by triangles on the sphere. Our goal is to bound the number of inliers the translations within a group can have in order to either discard it from further consideration or further subdivide it to search for an optimal solution. The idea for computing the bounds is similar to the one used in [1]. As described in Section 2 each point pair gives rise to a wedge on S^2. The point pair is an inlier to a translation candidate if the translation lies inside the wedge formed by the point pair. Therefore, if a wedge intersects the triangle there will be at least one translation in the group for which this point correspondence is an inlier. Thus, counting the number of wedges that either intersect the triangle or completely contains it gives an upper bound on the number of inliers each translation in the triangle can have.

Note that there may not be a single translation in the group that is contained withing all these wedges. To find a lower bound we also check how many wedges contain the center point of the triangle.

The lower and upper bounds for two triangles are illustrated in Figure 3.



(a) UB: 2, LB: 0

(b) UB: 2, LB: 1

Figure 3: Upper and lower bounds for the triangles. In (a) we see three wedges. Two of them intersect the triangle and none completely contain the triangle so the upper bound becomes two. Since the center point is not contained in any wedge the lower bound becomes zero. In (b) only the green wedge intersects the triangle while the red wedge contains the triangle completely. So the upper bound becomes two. Since the center point is contained in the red wedge the lower bound becomes one.

The subdivision of the space into triangles, where the edges belong to great circles, allow us to efficiently compute the intersections. To find the intersections between a triangle and a wedge we simply have to compute the six possible cross products between the plane-normals of the triangle and wedge respectively and check if they lie within the triangle. Some care has to be taken to ensure that the correct signs are chosen. To find wedges which have no intersections but completely contain the triangle we simply choose the center point of the triangle and check if it is contained in the wedge.

3.3. Implementation details

To greatly lower the computational cost for computing the upper bounds of the triangles we employ the *match-list* approach which was also used in [1]. The idea is that for any triangle the set of possible inliers is a subset of the possible inliers for the parent triangle. So for each triangle we keep track of which wedges it intersected and how many wedges it was completely contained in. When computing the bounds for a new triangle we only need to check the wedges which intersected the parent triangle.

The experiments were run on a computer with an Intel i7 3.4 GHz processor with 16 GB RAM. The algorithms where implemented in C++ and run on a single core.

The algorithm is summarized in Algorithm 1. In the experiments we will refer to this algorithm as BNB.

Algorithm 1: Branch and bound for two-view translation estimation.

```
Add initial triangles to queue
$lower\_bound = 0$
while queue not empty do
    $T$ = remove head from queue
    if $T.upper\_bound \leq lower\_bound$ then
        continue
    end
    $[T_1, T_2]$ =branch($T$)
    Compute bounds for $T_1$ and $T_2$
    if $T_1.lower\_bound > lower\_bound$ then
        $lower\_bound = T_1.lower\_bound$
        $current\_best = T_1.center$
    end
    if $T_1.upper\_bound > lower\_bound$ then
        Add $T_1$ to the queue.
    end
    /* Repeat the two previous If
       clauses for $T_2$.                  */
end
```

3.4. Comparison to Fredriksson et al.

We now present an experiment where we compare the performance of BNB to the performance of the method proposed in [7]. We consider the same 136 image pairs which were used in their experiment evaluation. The image pairs have no rotation between them and contain on average 7200 point pairs. In Figure 4 two of the images are shown. Since both methods solve the problem optimally we only compare the running times of the algorithms. In Figure 5 we can see a histogram of the running times for the two algorithms. The average running times were 0.8 s for BNB and 26 s for [7].

We also performed a running time comparison on synthetic data for problems of varying sizes. In the experiment we generate problem instances of varying sizes which contained approximately 10% inliers. The results can be seen in Figure 6. We can see that even for moderately sized problems the method of [7] becomes intractable while the proposed method has no problems solving large problems.

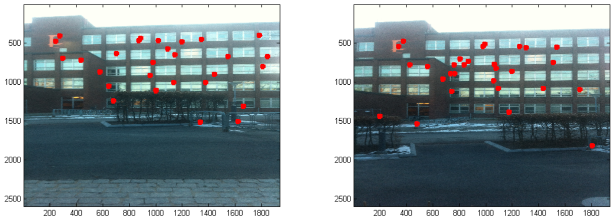


Figure 4: One image pair out of 136. The image points are shown in red.

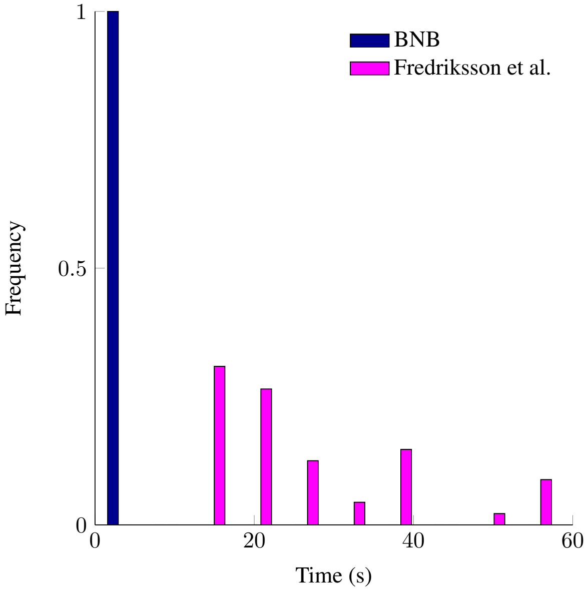


Figure 5: Histogram over the running times for the 136 image pairs. In blue BNB and in purple Fredriksson et al. [7]. The average running times are 0.8 s and 26 s.

3.5. Comparison to non-optimal methods

We also compared our method to a RANSAC based approach. The minimal solver for translation estimation requires two point pairs,

$$\boldsymbol{t} = (\boldsymbol{v}_1 \times \boldsymbol{v}_2) \times (\boldsymbol{v'}_1 \times \boldsymbol{v'}_2). \tag{9}$$

Since a RANSAC two point solver gives no guarantee of finding the optimal solution we plot the number of inliers of the best solution found by both methods against time. The result can be seen in Figure 7. Since RANSAC is a randomized algorithm we repeated the experiment multiple times and estimated a 95% confidence interval for its performance. The dotted lines in Figure 7 show this confidence interval. This experiment was constructed using the first im-

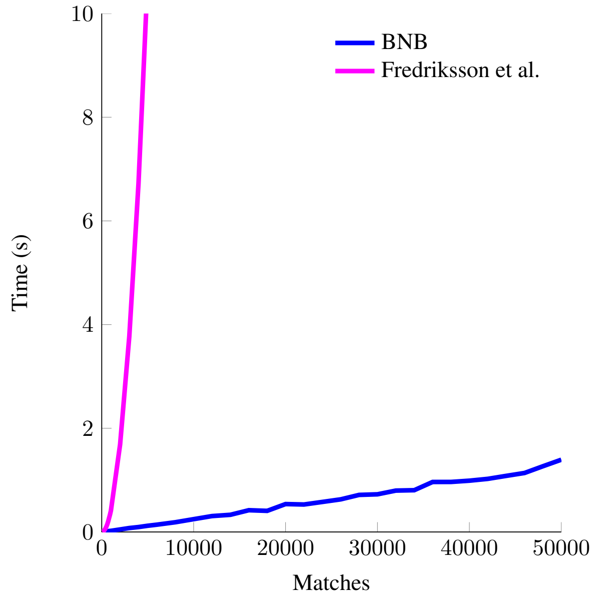


Figure 6: Comparisons of the running times on synthetic data for Fredriksson et al.[7] (purple) and BNB (blue). The number of inliers are kept at 10% and the number of point pairs vary between 50 and 50000.

age pair from the experiment in Section 3.4. The image pair contained about 5000 point pairs.

3.5.1 Synthetic data

Now we compare the running time for BNB against the two point RANSAC solver for synthetic data. The number of point pairs is set to be 100000 and we vary the outlier ratio. Since the two point solver never can be sure of finding a good solution, the number of iterations is chosen so that it has 99% chance of selecting at least one outlier-free sample. This results in about 100 iterations for 80% outliers and about 50000 iterations for 99% outliers. The running times can be seen in Figure 8.

4. Extension to one-to-many matchings

Scenes which contain repeated structures pose a difficult problem for reconstruction algorithms since it can give rise to ambiguous point pairs. Unfortunately repetition is a frequent occurrence in real world scenes, e.g. windows on a house facade. To deal with ambiguous point pairs it is common to require that an image points corresponding candidate is significantly better than the next best candidate [12]. In scenes with repeated structure this can lead to a greatly reduced number of point pairs.

One approach to deal with this is to allow each image point to belong to several point pairs. This is what we call the one-to-many matching problem. A naive approach would be to simply add all these point pairs. This however allows a single point to count as an inlier multiple times possibly leading to sub-optimal solutions.

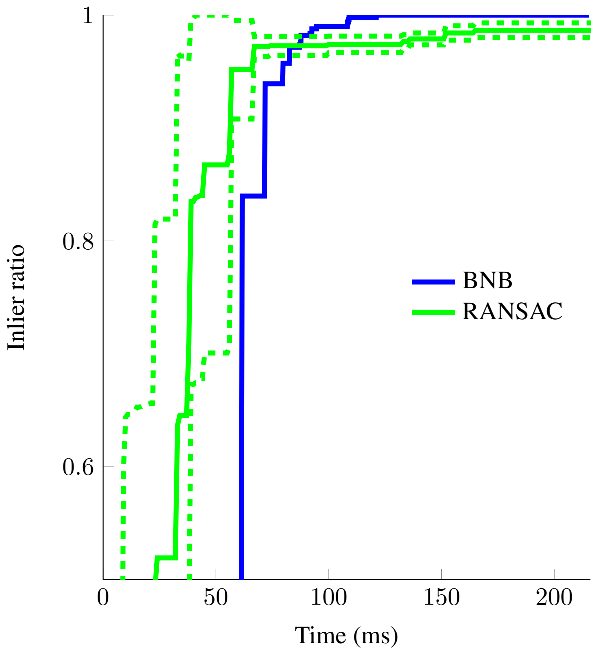


Figure 7: Comparisons of the running times for RANSAC (green) and BNB (blue). The figure is generated using the first image pair, with about 5000 point pairs. The dotted green lines are the 95% confidence intervals for the RANSAC solution. The blue curve shows the evolution of the lower bound.

The branch and bound method presented in Section 3 can easily be adapted to the one-to-many matching problem. The extension allows us to find solutions which are optimal in the sense that they maximize the unique number of point inliers in one of the images. While this is not enough to guarantee to finding a true one-to-one matching we will show that it is a good heuristic.

In the one-to-many matching problem we can have multiple wedges which correspond to a single point in the first image. To accurately compute the upper bound in a triangle we simply compute how many unique points that have wedges which intersect or embed the triangle. The modified algorithm is exactly the same as BNB except that we compute the bounds differently. In the experiments we will refer to this algorithm as BNBu.

4.1. Comparison to naive formulation

In this experiment we compare the modified branch and bound (BNBu) method with solving the naive formulation (BNB) where we simply add more point pairs for every point. To evaluate the solutions we count the number of

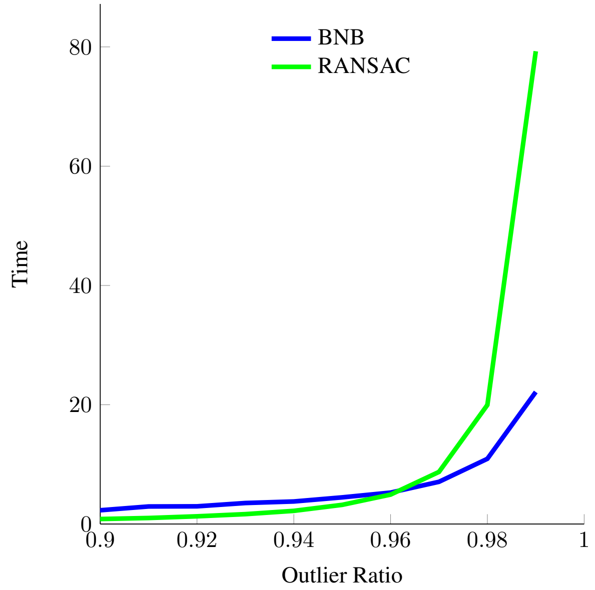


Figure 8: Comparisons of the running times on synthetic data for RANSAC (green) and BNB (blue). The number of point pairs are kept at 100000 and the outlier ratio varies between 90% and 99%.

Size	7.2k	72k	144k
Inliers	362	600	693
Percent inliers	5%	0.8%	0.5%

Table 1: Average number of inliers found by BNBu, depending on the problem size of the 136 image pairs. Every image has about 7200 feature points.

inlier point pairs which are uniquely matched in both images.

We consider the image pairs from [7], but we allow for the image points in one of the images to be a member of multiple point pairs. Table 1 shows how the number of inliers found by BNBu varies when we add more point pairs.

We compare the two proposed algorithms to [7] (up to 36k points) and RANSAC using the two-point solver. Since the RANSAC approach has no convergence guarantee we use 500 and 50000 iterations, corresponding to more than 99% chance of a good solution for problems with 90% respectively 99% outliers. The results for the different methods are in Table 2 and Table 3. For the larger instances the method from Fredriksson et al. [7] was not able to solve the problem in reasonable time (more than one hour).

Size	7.2k	72k	144k
Fredriksson et al. [7]	360	-	-
BNB	360	584	666
BNBu	**362**	**600**	**693**
RANSAC 500	316	441	536
RANSAC 50000	355	574	655

Table 2: Average number of unique (in both images) inliers between the 136 image pairs. Out of an average 7200 feature points.

Size	7.2k	72k	144k
Fredriksson et al. [7]	26s	-	-
BNB	0.8s	28s	97s
BNBu	1.3s	39s	128s
RANSAC 500	0.060s	0.76s	1.5s
RANSAC 50000	5.9s	68s	113s

Table 3: Average running time for the 136 image pairs for the one-to-many matching experiment.

4.2. All-to-all matching

All-to-all matching problems can arise in situations where the image points locally are indistinguishable from each other or in other situations where descriptor based matching fails. Examples of this can be e.g. images taken at night where the only visible objects are point shaped light sources from stars or street lights. The all-to-all problems are very hard to solve using the naive approach since algorithms often find translations with inlier point pairs that contradict each other. These ambiguous point pairs can not be counted and hence the solution fulfills few unique wedges. We compare BNB and BNBu in synthetic experiments. The number of image points vary from 40 to 100, resulting in 1600 and 10000 possible point pairs. In the experiments a solution is defined as valid if the angle to the ground truth translation is less than five degrees. The results can be seen in Table 4 and the running times is in Figure 9.

5. Conclusion and future work

In this paper we have presented an efficient branch and bound method for translation estimation in the presence of outliers. The method is guaranteed to find the optimal solution and in experiments we show that the proposed method can outperform the heuristic random sampling methods in both speed and solution quality. While most optimal methods trade speed for optimality, the proposed algorithm has competitive running times on problem sizes well beyond

Size	40	60	80	100
BNBu	**0.50 ± 0.1**	**0.49 ± 0.1**	**0.73 ± 0.1**	**0.49 ± 0.1**
BNB	0.25 ± 0.06	0.01 ± 0.09	0.02 ± 0.03	0.01 ± 0.02

Table 4: All-to-all experiments. All image points in one image are matched to all image points in the corresponding image. The table contains the mean together with the 95% confidence intervals of finding a good solution for BNBu and BNB.

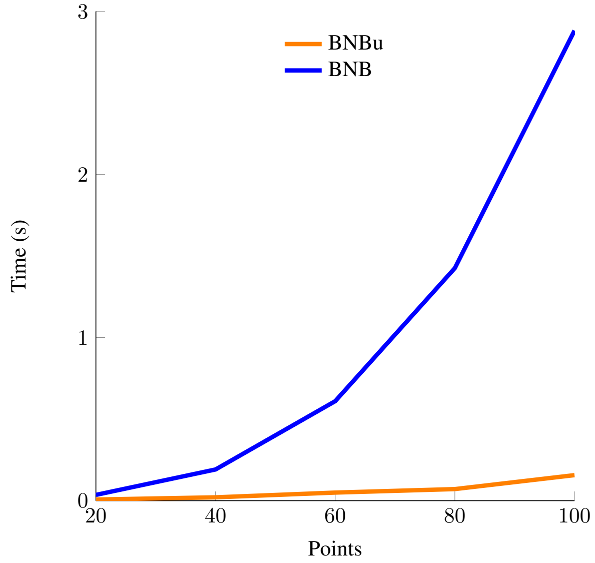


Figure 9: Number of image points plotted against running time for the branch and bound methods BNB (blue) and BNBu (orange) in the all-to-all experiments.

what is common in practice.

We also considered the problem of ambiguous point pairs by extending the method to handle instances where each image point in one image can have many possible matches in the other. This problem is solved optimally in the sense that we maximize the number of unique inliers in one of the images. Through experiments on real data we show that this can increase the number of inlier point pairs substantially.

Future work includes solving the multiple match problem optimally in both images simultaneously.

6. Acknowledgments

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

References

[1] T. M. Breuel. Fast recognition using adaptive subdivisions of transformation space. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 445–451. IEEE, 1992. 3

[2] T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 90(3):258–294, 2003. 2

[3] O. Enqvist and F. Kahl. Two view geometry estimation with outliers. In *British Machine Vision Conf.*, London, UK, 2009. 1

[4] O. Enqvist and F. Kahl. Two view geometry estimation with outliers. In *BMVC*, 2009. 2, 3

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1

[6] F. Fraundorfer, P. Tanskanen, and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *European Conf. Computer Vision*, Crete, Greece, 2010. 1

[7] J. Fredriksson, O. Enqvist, and F. Kahl. Fast and reliable two-view translation estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2, 4, 5, 6

[8] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009. 1, 2

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. Second Edition. 1

[10] F. Kahl and R. Hartley. Multiple view geometry under the L_∞-norm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1603–1617, 2008. 1

[11] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Int. Conf. Computer Vision*, Kyoto, Japan, 2009. 1

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

[13] O. Naroditsky, X. Zhou, J. Gallier, S. Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(4):812–824, 2012. 1

[14] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, June 2004. 1

[15] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):783–794, 2009. 2

[16] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *Int. Journal Computer Vision*, 49(2/3):117–141, 2002. 1

[17] J. Yang, H. Li, and Y. Jia. Optimal essential matrix estimation via inlier-set maximization. In *Computer Vision–ECCV 2014*, pages 111–126. Springer, 2014. 2