Optimal Correspondences from Pairwise Constraints

Olof Enqvist   Klas Josephson   Fredrik Kahl

Centre for Mathematical Sciences, Lund University, Sweden

www.maths.lth.se/vision

Abstract

Correspondence problems are of great importance in computer vision. They appear as subtasks in many applications such as object recognition, merging partial 3D reconstructions and image alignment. Automatically matching features from appearance only is difficult and errors are frequent. Thus, it is necessary to use geometric consistency to remove incorrect correspondences. Typically heuristic methods like RANSAC or EM-like algorithms are used, but they risk getting trapped in local optima and are in no way guaranteed to find the best solution.

This paper illustrates how pairwise constraints in combination with graph methods can be used to efficiently find optimal correspondences. These ideas are implemented on two basic geometric problems, 3D-3D registration and 2D-3D registration. The developed scheme can handle large rates of outliers and cope with multiple hypotheses. Despite the combinatorial explosion, the resulting algorithm which has been extensively evaluated on real data, yields competitive running times compared to state of the art.

1. Introduction

We focus on two basic geometric problems. The first is 3D-3D registration, also known as 3D pose. Given two point sets in 3D space, the task is to estimate a rigid transformation aligning one point set with the other. What makes the problem challenging is that often the correspondences, that is, which points should be matched between the sets, are not known in advance. Hence we need to estimate both the correspondences and the transformation. See Fig. 1 for an alignment result based on our method. The second problem we consider is 2D-3D registration, also known as calibrated camera pose. As in the 3D-3D case, we need to estimate both the correspondences and the transformation simultaneously, but now the transformation is the perspective camera mapping, projecting 3D points to image points.

Given correct correspondences, it is straightforward to estimate the transformation, see [10]. For 3D-3D registration, it can even be done in closed form. On the other hand, given an estimate of the transformation, it is easy to deter-
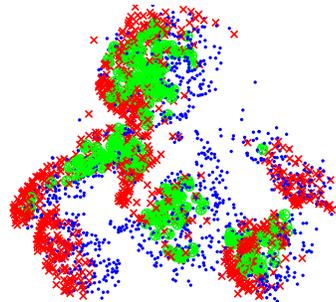

Figure 1. 3D-3D registration of two stereo reconstructions. Green circles and stars are matched 3D points in a maximal inlier set. Red x-marks and blue dots are unmatched points in the two models.

mine likely correspondences. So, a natural idea is to use an alternating minimization procedure, and there are many such algorithms in the literature ([8]), the most famous one being Iterative Closest Point (ICP) [1]. However, such approaches require a good initial estimate of the transformation and still there is no guarantee of getting a reasonable solution, especially when there are lots of outliers.

The aim of this paper is to find a more reliable and if possible faster method to deal with incorrect matchings. Our goal is to find the largest set of consistent correspondences. The key idea is to consider point correspondences and check whether pairs of such correspondences are consistent or not. This leads us to the vertex cover problem in graph theory. The bad news is that the problem is known to be NP-hard. The good news is that we are still able to solve instances of the problem for quite large data sets using a combination of branch-and-bound and approximation algorithms.

For the experiments we have used SIFT [18] to generate correspondences. The motivation is that SIFT is widely used which allows easier comparison to other work. However, as pointed out in [7], SIFT is actually not ideal for 3D problems with occlusions and perspective changes. Hence, we would like to compare the performance using other descriptors. Moreover, many feature detectors are optimized to give a high inlier-to-outlier ratio. Since our method can handle large amounts of outliers, it would be interesting to look more at the number of correct correspondences.

2. Problem Formulation and Related Work

Definition 1. *A* correspondence *is a pair of indices* (i, j) *indicating that point* x_i *in the first set corresponds to point* y_j *in the second set. A correspondence is an* inlier *with respect to a transformation* T *and a threshold* ε*, if* $d(y_j, T(x_i)) \leq \varepsilon$ *and otherwise it is an* outlier *with respect to* T.

Furthermore, two correspondences are inconsistent if they match the same point in one point set to different points in the other set. We want to solve the following problem.

Problem 1. *For a prescribed threshold* ε *find the transformation* T *that maximizes* $|I|$*, where* I *is the maximal consistent set of inliers to* T.

For 3D-3D registration, the transformation T is just a rigid transformation and the error residual is the geometric distance. For 2D-3D registration, T is a rigid transformation followed by a perspective mapping and the error residual is the angular reprojection error, that is, the angular difference between the measured image point and the reprojected 3D point.

Pairwise constraints. We approach Problem 1 by considering pairs of correspondences. For each problem we will show how to detect geometric inconsistency of such a pair. Two geometrically inconsistent correspondences cannot both belong to the optimal solution and hence we look for large sets of pairwise consistent correspondences. In general pairwise consistency is not sufficient, but we will show how to get around this problem.

Assume we have established pairwise consistencies. We then build a graph with all hypothetical correspondences as vertices and edges connecting inconsistent ones. Clearly a consistent subset according to Definition 1 cannot include any edges. Thus the maximal subset of pairwise consistent correspondences should be a good candidate for the optimal solution. Finding this set is equivalent to removing as few vertices as possible while *covering* all edges. This is known as the vertex cover problem and one of Karp's 21 NP-complete problems [14] presented in 1972.

Definition 2. *A* vertex cover *for an undirected graph, is a subset* S *of its vertices such that every edge of the graph has at least one endpoint in* S.

Problem 2. *The* vertex cover problem *is the problem of finding the smallest vertex cover for a given graph.*

Related work. The idea of using vertex cover for finding mutually consistent correspondences is not new in the vision literature. One of the earliest work we have found is [2] where it is used for 2D part location. Pairwise constraints are also discussed in [9]. In [22], an association

graph is built for 2D matching which results in a maximum clique problem. This is an alternative formulation of the same graph problem as ours. In [16], the registration problem was formulated in a graph setting, but solved using a non-optimal spectral technique. Graph matching has also been applied to non-rigid registration problems [21], which are not considered in this paper. In [3], a branch-and-bound algorithm over rigid transformations in the 2D plane is proposed. This works well since the transformations have only three degrees of freedom, but the approach becomes computationally infeasible for rigid transformations in 3D space.

For 3D-3D registration, the most popular class of methods are EM-type algorithms using alternating optimization, such as ICP [1] and SoftAssign [8]. Other non-optimal approaches include the Hough transform, geometric hashing and hypothesize-and-test algorithms like RANSAC [6]. Recently, in [17], the problem was solved globally using branch and bound in rotation space. The method requires that all points are matched and that the translation component is given, which are severe limitations so this is effectively not solving the complete problem. We have compared our algorithm to several of these competitors in the experimental section.

For 2D-3D registration, the literature is more diverse. There are several heuristic methods with no guarantee of optimality such as [12, 19, 4] where RANSAC is perhaps the most popular one [6]. In multiple view geometry, the camera pose problem is often solved with DLT [10], but the method cannot handle outliers among the correspondences. Another class of methods is based on subdividing transformation space and aiming for global solutions such as [13] using an affine camera model. Our approach is based on the global method in [5], but we present a more sophisticated tool for discarding outliers and show how to incorporate orientation constraints.

Unlike many of the above mentioned methods, by using pairwise constraints, it is possible to generate multiple hypotheses for a single point, but in the final solution, it is guaranteed that no conflicting hypotheses are included. Another advantage is that it is possible to use extra information from the feature extractor, for example the orientation.

3. Vertex Cover

We will review two classical approaches for solving the vertex cover problem, namely the linear-time factor-2 approximation and a branch-and-bound approach [23]. These methods will form the basis for our vertex cover algorithm, specially designed for registration problems.

Factor-2 approximation. A factor-k approximation for a minimization problem is an approximation that is guaranteed to give no worse than k times the optimal value. For

the vertex cover problem there is a well-known linear-time factor-2 approximation. It is obtained by repeatedly picking a random edge in the graph and then placing both its endpoints in the covering. Take for example the graph in Fig. 2. We start by picking the edge between vertices 1 and 2, add these vertices to our solution set and remove them from the graph. In the next step we pick the edge between 4 and 5 and remove these two vertices. We have found a vertex cover $\{1, 2, 4, 5\}$. This happens to be exactly twice as big as the minimal vertex cover $\{2, 5\}$. Note that starting with the edge between 2 and 5 would have yielded the optimal solution.
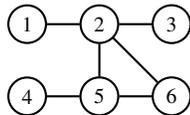

Figure 2. Example graph for vertex cover.

Branch and bound. Using branch and bound one can get guaranteed optimal solutions to difficult problems like vertex cover. Using the factor-2 approximation as bounding function we get an effective method to remove incorrect correspondences from our problem.

Assume that for some graph we want to determine whether there is a vertex cover with less than n elements. If a factor-2 approximation does not give a hard enough bound, we pick a vertex v and split the problem in the following way. Either (i) v belongs to the minimal vertex cover, or (ii) v does not belong to the minimal vertex cover.

In the first case we remove v from the graph as well as any edges to v. In the second case any vertices having an edge to v must lie in the minimal vertex cover so we remove these and update the graph. This gives us two smaller vertex cover problems that can be approached in a similar fashion. To quickly reduce the size of the graph we pick v to be the vertex with most edges.

Multiple hypotheses. An advantage of our approach to correspondence problems is that multiple hypotheses can easily be incorporated, that is, several hypothetical correspondences concerning the same point. To avoid getting a solution where one point is matched to several points, we simply add edges between any vertices (correspondences) matching the same point. The set of vertices matching a certain point will thus form a clique in the graph (Fig. 3).

3.1. An Algorithm for Vertex Cover

In general the vertex cover problem is very challenging, but what we have found is that the problems arising from registration are often much simpler. This section will describe an algorithm to solve such vertex cover problems
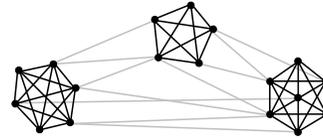

Figure 3. Example graph with multiple hypotheses. Correspondences matching the same point form a clique in the larger graph.

with good empirical performance. The algorithm is based on two classical approaches to the vertex cover problem - the effective factor-2 approximation and the guaranteed optimal branch and bound. By combining them, we get a method which is fast but still robust enough for difficult registration problems.

The algorithm has two objectives. One is to find better and better solutions and one is to remove outliers (incorrect correspondences). When all correspondences that have not been removed are elements of an optimal solution the algorithm has converged. Switching to graph terminology, the first objective is to find smaller and smaller vertex covers and the second to prove for more and more vertices that they must lie in the minimal vertex cover.

Algorithm 1. *Vertex Cover*

Let N be an upper bound for the minimal vertex cover.
Iterate until convergence:
1. Pick a vertex v from the graph.
2. Try to prove that v lies in the minimal vertex cover.
*   If this works*
*   - Remove v and update the graph.*
*   otherwise*
*   - Find a vertex cover S with $v \notin S$.*
*   - If $|S| < N$, update N.*

Removing vertices. Let N be the size of the smallest vertex cover that we have found so far. Naturally the minimal vertex cover S^* satisfies $|S^*| \leq N$. To prove that a certain vertex v is an element of S^*, we use the following scheme.

---

1. Assume that $v \notin S^*$. Then all vertices
*   having an edge to v must lie in S^*.*
2. Remove these and update the graph.
*   The solution to the reduced problem is the*
*   smallest vertex cover that does not include v.*
3. Find a factor-2 approximation for the reduced problem.
*   If this is $> N$, reject the hypothesis that $v \notin S^*$ and*
*   remove v from the graph.*

For many vertices it will not be necessary to use the factor-2 approximation in step 3. Simply counting the remaining vertices will be sufficient. Having multiple hypotheses for

each point it is even more effective to look at the number of unique points that these remaining vertices match. This also means that it is rarely necessary to actually set up the complete graph.

Finding smaller vertex covers. To find smaller and smaller vertex covers we use a simple heuristic that worked well in the experiments. Given a vertex v that we cannot prove to belong to the minimal vertex cover.

---

1. Assume that $v \notin S^*$.
2. Remove all vertices that have an edge to v.
3. Repeat until no edges remain:
   - Remove the vertex with most edges. Update the graph.
This yields a vertex cover for the graph.

---

4. Pairwise Constraints

In this section, the mechanisms for generating pairwise correspondence constraints are described.

4.1. 3D-3D Registration

Consider two hypothetical correspondences (i, j) and (m, n). If they are both correct then $|x_i - x_m| = |y_j - y_n|$ provided that there is no noise. This enforces that distances between corresponding points must be equal in the two point clouds. Thus two inliers in the optimal solution of Problem 1 must satisfy

$$\big| |x_i - x_m| - |y_j - y_n| \big| < 2\varepsilon, \tag{1}$$

for noise threshold ε.

Note that a set of correspondences can be pairwise consistent even though they are not consistent according to Definition 1. To verify that a given set of correspondences is consistent we instead use an approximate method to estimate the transformation and then check that all correspondences are consistent with this transformation. For the experiments we used the method from [11].

This leads to a simple and straightforward algorithm for registration, see Algorithm 2. In the experimental section, we have tested the empirical performance on numerous of real data sets and compared with other algorithms.

Algorithm 2. *3D-3D Registration*

1. Compute all paiwise distances.
2. Solve the vertex cover problem using Algorithm 1.
3. Compute the transformation for the obtained inlier set.
4. Verify consistency.

This approach will give bounds on the optimal number of outliers. The lower bound is the size of the minimal vertex cover and the upper bound is the number of outliers to the approximate transformation. As the experiments will show, the gap between these bounds is often very small. Even so, a method to reduce this gap is desirable.

Sufficiency. The error threshold of Problem 1 can be interpreted as an uncertainty in one of the 3D models. Instead of a discrete position each point has a sphere of possible positions. Let us first assume that this uncertainty is zero. We consider the inliers indicated by the minimal vertex cover. We know that they are all pairwise consistent, i.e. that all pairwise distances match. However, this means that the transformation between the two point sets can be described by a rotation, a translation and possibly a reflection.

This hints at a way of refining our search. By dividing some of the point uncertainty spheres into smaller spheres (or boxes) our bounds get harder. Each new smaller sphere yields a new hypothetical correspondence. Since we can already handle multiple hypotheses we can solve the refined problem using Algorithm 2. In the limit we approach the exact case when our constraints are sufficient - assuming that we also check for the reflection.

If many divisions of this type have to be made, execution will tend to get slow but the experiments suggest that this is rarely the case.

Remark 1. In many applications, for example, 3D-3D surface registration, there are also orientation constraints available. Surface normals at corresponding points should match. Given an angular threshold ε for consistency, such constraints are trivial to incorporate and will (generally) speed up the execution time of the algorithm since outliers are easier to discard in the vertex cover algorithm.

4.2. 2D-3D Registration

In [5] a method to solve the 2D-3D registration problem with outliers was presented. The approach was based on pairwise constraints combined with a branch-and-bound search over the possible camera positions. It was shown how this could be used to find the optimal camera pose from a calibrated camera. As in the 3D-3D case, the suggested pairwise constraints cannot guarantee convergence but [5] showed how this can be handled. This paper contributes two improvements to the method. Most importantly it is noted that the subproblems given by the pairwise constraints are instances of vertex cover and a method to efficiently handle these instances is presented. Furthermore, in the appendix it is shown how to incorporate orientation constraints in the search.

Let us first briefly go through the algorithm. For a thorough derivation, see [5]. Assume for a moment that we have no noise and no false correspondences. Then, given

two correspondences (i, j) and (m, n) with 3D points X_i and X_m and corresponding image vectors y_j and y_n, we have

$$\angle \left(X_i - C, X_m - C \right) = \angle \left(y_j, y_n \right). \qquad (2)$$

This yields a constraint on the camera position C. On the left hand side of this equation X_i and X_m are given by our 3D-model. On the right hand side we have the angle between two image vectors, which is simply a constant that can be calculated from the measured image coordinates. We denote it α.

We seek the points C in space that form exactly the angle α with X_i and X_m. It turns out (see [5]) that they all lie on a surface obtained by rotating a circular arc, see Fig. 4. Moreover, any $\bar{C}$ such that $X_i \bar{C} X_m > \alpha$ lies in the set enclosed by this surface. We define

$$M_\alpha(X_i, X_m) = \{ C \in \mathbb{R}^3 : X_i C X_m > \alpha \}. \qquad (3)$$

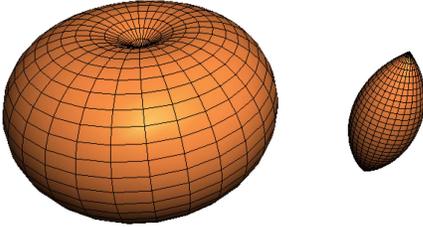If $\alpha < \pi/2$, this set will be non-convex and shaped like a pumpkin.

Figure 4. M_α for angles less than (*left*) and larger than (*right*) $\pi/2$ respectively.

Now suppose (X_i, y_j) and (X_m, y_n) are two inliers each having an angular reprojection error less than ε in the optimal solution, cf. Problem 1. Consider a box of $\mathbb{R}^3$. If there exists a camera position C in this box such that the correspondences are correct, then by the spherical triangle inequality

$$\alpha - 2\varepsilon \leq \angle \left(X_i - C, X_m - C \right) \leq \alpha + 2\varepsilon. \qquad (4)$$

Checking whether there exists such a camera position C in the box amounts to checking whether the pumpkins $M_{\alpha \pm 2\varepsilon}$ intersect the box. If the intersection is empty, an inconsistency is obtained.

For a candidate box in $\mathbb{R}^3$ of camera positions, it is not required that we solve the complete vertex cover problem every time. Since we are performing branch and bound, it is enough if we can get a bound on the number of possible inliers. Algorithm 1 does indeed keep track of this bound. The success of the registration algorithm is dependent on discarding whole boxes at an early stage without having to

subdivide it many times. Another crucial factor is if we are able to discard individual outliers early, because then we will get smaller vertex cover problems when we subdivide the box. See Algorithm 3.

Algorithm 3. *2D-3D Registration*

Initialize to get a bounded set in $\mathbb{R}^3$.
Iterate until desired precision is reached:
    1. Pick a box from the queue.
    2. Set up a graph with all pairwise consistencies for this box.
    3. Use the vertex cover techniques to discard the box.
    4. If the box cannot be discarded:
        - Use vertex cover techniques to remove outliers
        - Divide the box and update the queue.
        - Try to update the bound on the optimum.
    6. Remove the box from the queue.

Remark 2. The constraints of the last section use the positions of image features to restrict on the camera position. However, often the orientation of feature points is also available (e.g. from SIFT) and enforcing consistent orientations as well will give harder constraints on the camera position and make it easier to find the correct correspondences.

In Appendix A we derive an algebraic constraint to enforce orientation consistency. To incorporate this in our optimization we reformulate Problem 1 slightly considering the reprojection errors of the feature orientations as well.

5. Results

Simple C implementations of the algorithms were made for the experiments. Execution times are for a computer with a 3.0 GHz Intel DualCore CPU and 3 GB RAM.

5.1. 3D-3D Registration

We tried our algorithm on a 500-point 3D model of the Stanford bunny (see Fig. 5). The same data set was used in [17], but there the translation was assumed known and only the rotation was estimated. To mimic the experiments from [17] we computed a random rotation and translation and added uniform noise of magnitude 0.1 to the transformed points. The error threshold of Problem 1 was set to 0.3 (approximately 1% of the bunny length). We then set the algorithm to seek a solution matching at least 90 % of the points. In all cases the optimal solution was found with the basic Algorithm 2 without splitting any hypotheses.

In this experiment the total number of hypothetical correspondences was $250\,000$ and the rate of true correspondences 0.2%, since no correspondences were given and hence all points were matched against all points.[1] Our mean

---

[1]This means that a basic RANSAC algorithm would have required on average 125 million iterations to converge.

Figure 5. Left: The Stanford Bunny used in the 3D-3D experiments. Right: One of the images of the shopping street used in the 2D-3D experiments.
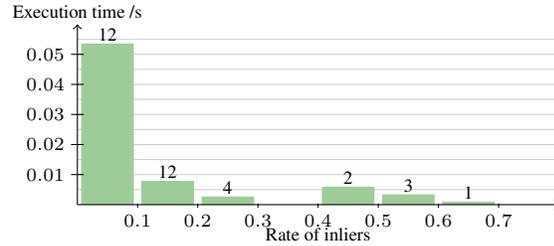

Figure 6. Execution times for 3D-3D registration with 34 stereo pairs from [15] versus the ratio of inliers. The bars show the mean execution time and the numbers give the number of examples in each interval. The optimal solutions have between 6 and 100 inliers.

execution time was 0.77 seconds. Essential to get this low average is that thanks to the bounding discussed in Section 3 we never have to set up the complete vertex cover graph.

The algorithm in [17] obtains similar results on this data set as our algorithm, but they assume that the translation is known and reported execution times are much higher. The ICP and the SoftAssign algorithms [1, 8] work only when the initial rotation is within an angle of less than 50 degrees of the correct solution.

To get more realistic examples, the 3D-3D registration algorithm was also tested on the data introduced in [15]. An example of those images is shown in Fig. 10. Three-dimensional models were constructed from each stereo pair in the training set. Then the registration was performed to the two neighboring stereo pairs for each model. The bear, ball, vase and Bournvita images were used.

To generate the 3D models the training images of the data set were downsampled a factor four. Then SIFT features [18] were extracted and matched for each stereo pair. The matching was performed by comparing the ratio between the best and the second best match, as proposed by Lowe, with a threshold at 0.6. To construct the actual 3D-model, points were triangulated using the optimal method described in [10]. Points with a reprojection error larger than two pixels were removed.

In the matching between two stereo models all SIFT features with a dot product greater than 0.75 were considered to be hypothetical matches. This results in many multiple hypotheses. The algorithm was set to seek a solution with 20 inliers. If none was found this number was divided by two and the algorithm restarted. In all cases an optimal solution was found and verified, but in a few examples we did not prove uniqueness. This could have been done using the refined algorithm discussed in Section 4.1.

In Fig. 1 a concatenation of two stereo models of the bear is shown. In Fig. 6 the execution times for the model concatenation is given for different ratio of outliers.

5.2. 2D-3D Registration

Recognition. Again, we worked with the data sets introduced in [15], in this case, the bear and the Bournvita objects. Our aim was not a complete recognition system but rather to show that our algorithm is robust enough to work with challenging real world data. It is also an example of how to use an optimal algorithm to evaluate features for recognition. Our goal is again to find the largest consistent set of correspondences. The size tells how well the 3D model matches the image.

First SIFT features were extracted from the test images. The features were then matched to the stereo model with most overlapping view as well as three other models, one of the same object in a different view and two of another object. The threshold of the ratio in the feature matching was set to 0.8. Since we were interested in the recognition problem, we ran the algorithm until we had tight bounds on the optimal inlier set. Figure 9 shows the lower and upper bounds in two examples. For all but one image the difference between lower and upper bound was ≤ 5 and in this image there were between 252 and 270 inliers. The error threshold used was 0.0015 radians. To initialize Algorithm 3 we put a lower bound on the size of object in the image yielding a bound on the distance from the camera to the object.

The sizes of the inlier sets are presented in Figs. 7–8. In 2 out of 22 bear images, there were less than 20 correspondences in the optimal set when matched against the bear stereo models. For the Bournvita images, all test images had optimal sets larger than 20 correspondences. There was not a single case with more than 20 correspondences in other cases. In Fig. 10 the 3D model is re-projected for one of the test images in the data set.

The same setup was used in a RANSAC engine where three points were randomly chosen to estimate the camera position. After 10,000 iterations for each test image, 3 out of 22 bear images did not get more than 20 inliers and 1 out of 9 images of the Bournvita objects. So RANSAC may give lower number of correspondences in some cases but more importantly, no bounds on the solutions are given.
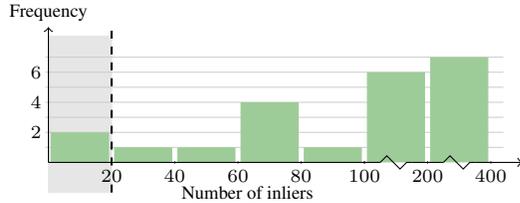
Figure 7. Size of the inlier sets for 22 bear images from [15]. Note the axis. Matching these images to stereo models of the other objects gave inlier sets exclusively in the shaded area.
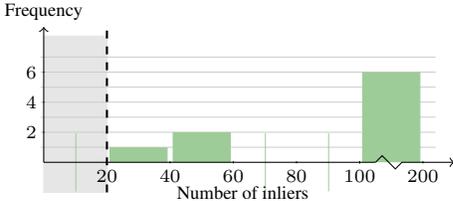

Figure 8. Size of the inlier sets for the 10 Bournvita images from [15]. Note the axis. Matching these images to models of the other objects gave inlier sets exclusively in the shaded area.
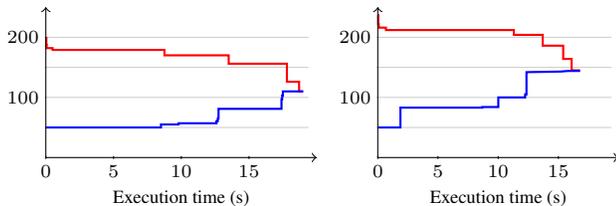

Figure 9. Upper and lower bounds for the solution as a function of time in vertex cover algorithm. (Test images 15 and 16 from [15].)

Figure 10. A bear with the 3D model reprojected onto the image.

Shopping street.   The 2D-3D registration was also tested on a data set consisting of 94 images of a shopping street covering approximately 100 meters. An example of these images is shown in Fig. 5. On this data set both registration with and without the use of orientation constraints were tested. To perform these experiments a model consisting of 3D points and camera matrices were constructed using Photo Tourism [20]. Then one image at the time was removed from the model and the proposed method was used to find the largest possible inlier set for that image. The
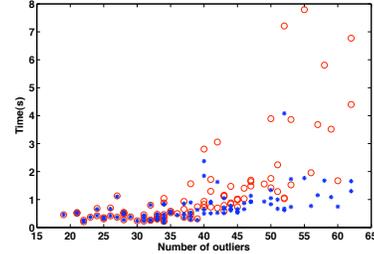

Figure 11. Comparison with the method from [5]. Red rings show execution times of [5] and blue stars show our result.
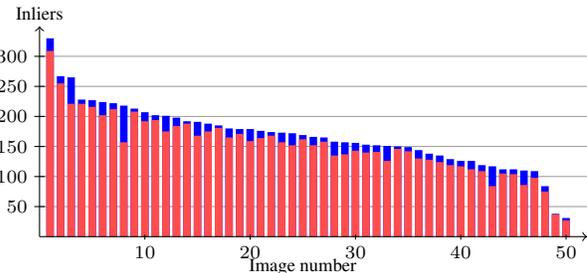

Figure 12. Blue bars give the number of inliers when the angle constraint is not used. The red bars give the number of inliers for the same problem when the angle constraint is used.

method from [5] was also implemented and tested for comparison. Fig. 11 shows that our method scales in a much more favorable way when the number of outliers increases.

The method using orientation constraints was also tested on this data set. For each pair of images in the model the points visible in both images were located. For these points the orientation vector given by SIFT was reconstructed in 3D by intersecting planes. We then ran our 2D-3D registration algorithm with and without enforcing orientation consistency. Figure 12 shows the number of inliers in the two cases. It shows that the orientations given by SIFT are indeed stable enough to use in such a setting. The error thresholds used in this experiments were 0.005 for the positions and 0.2 for the less stable orientation vectors.

6. Conclusions

The vertex cover formalism is not a new idea for correspondence problems, but it has previously only been used with limited success. Even though it is an NP-complete problem, we have shown that it is possible to achieve competitive results with sophisticated use of approximation algorithms in combination with today's fast computers. For several state-of-the-art methods, we have compared the performance to our approach in terms of number of inliers and running times with good results. For example, at high rates of outliers we have demonstrated significant speedup com-

pared to [5] on real world data.

A. Orientation Constraints

In this section we derive the constraints for orientation consistency in 2D-3D registration. This makes it possible to incorporate orientation consistency into Algorithm 3.

Let X_i be a 3D point with orientation given by V_i and and x_i be the corresponding image point with orientation given by v_i. Ideally, the image orientation vector is the projection of the corresponding 3D orientation vector. Thus the angle $\phi = \angle\big(v_i, x_j - x_i\big)$ in the image is the projection of $\Phi = \angle\big(V_i, X_j - X_i\big)$.

To get a simple algebraic constraint, we place the origin at X_i and choose a coordinate system such that V_i is parallel to $(0, \sin\Phi, \cos\Phi)$ and $(X_j - X_i)$ parallel to $(0, -\sin\Phi, \cos\Phi)$. In this coordinate system the orientation consistency yields an amazingly simple constraint on the camera position. For a unit vector (x, y, z) parallel to $(C - X_1)$, it follows

$$y^2 + x^2 \cos^2\Phi - \sin^2\Phi = |x| \sin(2\Phi) \cot\phi. \quad (5)$$

Given a box in the branch and bound, (x, y, z) is known within some angular uncertainty. This gives us,

$$c_1 x + c_2 y + c_3 z \geq 1. \quad (6)$$

We are mostly interested in the boundary of this area. Using the norm constraint on (x, y, z) we get

$$c_3^2 x^2 + c_3^2 y^2 + (c_1 x + c_2 y - 1)^2 = c_3^2 \quad (7)$$

which is a quadratic equation in x and y. Equations (5) and (7) can be combined to one fourth degree polynomial in x. Using this equation we can determine if any camera in the current branch and bound box is consistent with this pair of features.

References

[1] P. Besl and N. McKay. A method for registration two 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] R. Bolles and R. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. Journal Robotics Research*, 1(3):57–82, 1982.

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

[4] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. SoftPOSIT: Simultaneous pose and correspondence determination. *IJCV*, 59(3):259–284, 2004.

[5] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *European Conf. Computer Vision*, pages 141–153, Marseille, France, 2008.

[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.*, 24:381–395, 1981.

[7] P.-E. Forssén and D. Lowe. Shape descriptors for maximally stable extremal regions. In *IEEE International Conference on Computer Vision*, volume CFP07198-CDR, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society.

[8] S. Gold, A. Rangarajan, C. Lu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31:957–964, 1998.

[9] E. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990.

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

[11] B. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using ortonormal matrices. *Journal of the Optical Society of America A*, 5, 1988.

[12] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *Int. Journal Computer Vision*, 5(2):195–212, 1990.

[13] D. Jacobs. Matching 3-d models to 2-d images. *Int. Journal Computer Vision*, 21(1–2):123–153, 1997.

[14] R. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations (Symposium Proceedings)*, 1972.

[15] A. Kushal and J. Ponce. Modeling 3d objects from stereo views and recognizing them in photographs. In *ECCV*, pages 563–574, Graz, Austria, 2006.

[16] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, Rio de Janeiro, Brazil, 2007.

[17] H. Li and R. Hartley. The 3D – 3D registration problem revisited. In *ICCV*, Rio de Janeiro, Brazil, 2007.

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

[19] C. Olson. A general method for geometric feature matching and model extraction. *Int. Journal Computer Vision*, 45(1):39–54, 2001.

[20] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM SIGGRAPH*, 25(3):835–846, 2006.

[21] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *European Conf. Computer Vision*, pages 596–609, Marseille, France, 2008.

[22] P. Tu, T. Saxena, and R. Hartley. Recognizing objects using color-annotated adjacency graphs. In *Lecture Notes in Computer Science; Shape, Contour and Grouping in Computer Vision*, pages 246–263, 1999.

[23] V. Vazirani. *Approximation Algorithms*. Springer, 2001.