Image-Based Localization Using Hybrid Feature Correspondences

Klas Josephson Martin Byröd Fredrik Kahl Kalle Åström
{klasj, byrod, fredrik, kalle}@maths.lth.se
Centre for Mathematical Sciences,
Lund University, Lund, Sweden

Abstract

Where am I and what am I seeing? This is a classical vision problem and this paper presents a solution based on efficient use of a combination of 2D and 3D features. Given a model of a scene, the objective is to find the relative camera location of a new input image. Unlike traditional hypothesize-and-test methods that try to estimate the unknown camera position based on 3D model features only, or alternatively, based on 2D model features only, we show that using a mixture of such features, that is, a hybrid correspondence set, may improve performance.

We use minimal cases of structure-from-motion for hypothesis generation in a RANSAC engine. For this purpose, several new and useful minimal cases are derived for calibrated, semi-calibrated and uncalibrated settings. Based on algebraic geometry methods, we show how these minimal hybrid cases can be solved efficiently. The whole approach has been validated on both synthetic and real data, and we demonstrate improvements compared to previous work.

1. Introduction

Localization refers to the ability of automatically inferring the pose and the position of an observer relative a model [2]. We propose to solve the problem using an image-based approach. The model or the map of the environment can be anything from a single room in a building to a complete city. In general, one image will be used as a query image, but in principle several images can be used as input. No prior knowledge of the observer's position is assumed and therefore the problem is often referred to as global localization whereas local versions assume an approximate position. The mapping of the environment can be regarded as an off-line process since it is generally done once and for all. Such a mapping can be done using standard Structure from Motion (SfM) algorithms [7, 10], or by some other means.

In this paper, we demonstrate how a mixture of 2D and 3D features can be used simultaneously for localization. If one were to rely solely on 3D matches, one is restricting the set of possible correspondences to relatively few correspondences and a relatively rich 3D model would be required in order to be successful. On the other hand, using only 2D features requires relatively many correct correspondences to generate a single hypothesis. In addition, with existing methods such as the seven point algorithm of two views [7], one is limited to picking all the 2D correspondences from one single image in the model. Again, one is restricting the set of correspondences to a relatively small subset. Further, the absolute scale cannot be recovered solely from 2D correspondences of one query image and one model image.

Using hybrid correspondence sets for generating hypotheses gives a number of advantages. We can make use of all possible correspondences simultaneously, even from different 2D model images. Compared to approaches using only 2D correspondences, the scale relative to the 3D map can be recovered and, more importantly, the number of correspondences is smaller which is a good property when using RANSAC. One can argue that in most cases, traditional methods would work fine. However, we demonstrate that hybrid correspondence sets are indeed useful and there is simply no reason why this information should not be used as it leads to improvements.

The three main contributions of this paper are:
1. We demonstrate how hybrid feature correspondences can be used for improved image-based localization.
2. A complete list of minimal hybrid cases is given and for each case, we also give the number of possible solutions possible.
3. Algorithms for efficiently computing the solutions of the minimal cases are given. Further, the behavior and stability on synthetic data is evaluated for some cases.

1.1. Related Work

Localization and scene recognition are key components of any autonomous system. In robotics, (global) localization is also known as the *kidnapped robot problem*. Successful solutions have generally been achieved with laser, sonar or stereo vision range sensors and built maps for controlled robots moving in 2D, e.g., [12]. Another example

is the Deutches Museum Bonn tour-guide robot RHINO [3] where laser sensors are used. Another competing technique (at least, for some applications) is GPS. However, the accuracy is typically only in the order of 10-20 meters and no direction information is obtained.

Image-based localization using special landmarks is a common approach, e.g., [1], but this severely limits the flexibility and the applicability of the method. Similar to our approach, distinctive visual features were utilized in [14] to overcome this limitation. They also showed that RANSAC is an effective way of generating hypotheses. However, only 3D model features were used and this requires a rich 3D model to work well.

For large-scale models, an image search technique is required to speed up the process. This can be seen as a pre-processing step which produces a small number of hypothetical part-models that need further verification. Possible such pre-processing schemes are developed in [13].

The wealth of research in the SfM field is, of course, related to the present work, in particular, the work concerned with RANSAC [17] and wide baseline matching [18, 11]. The same approach as proposed in this paper can be used to solve the wide baseline matching problem to build up 3D models.

Understanding of the geometry and the number of solutions of minimal structure and motion problems has a long history. For the uncalibrated case, the minimal problem of seven points in two views (which has three solutions) was studied and solved already in 1855, cf. [4]. The corresponding calibrated case was in principle solved in 1913 [8]. The study of minimal cases has got increased attention with its use in RANSAC algorithms to solve both for geometry and correspondence in numerous applications [7].

2. Problem Formulation

To solve the localization problem we are interested in solving the following problem:

Problem 1 *Under the assumption that for a query image, there are m potential correspondences to image points in views with known absolute orientation and n potential correspondences to scene points with known 3D coordinates, find the largest subset of the correspondences that admits a solutions to the absolute orientation problem within a specified accuracy.*

The method that we will use to solve the localization problem is based on hypothesize-and-test with RANSAC [6] and local invariant features [9]. This involves solving minimal structure and motion problem with hybrid correspondence sets.

3. Minimal Hybrid Correspondence Sets

The classical absolute orientation problem (also known as camera resectioning) for calibrated cameras for three known points can be posed as finding the matrix $P = [\, R \, t \,]$, such that $\lambda_i u_i = P U_i$, $i = 1, 2, 3$. Here R is a 3×3 rotation matrix and t is a 3-elements translation vector. Thus, the camera matrix encodes six degrees of freedom of unknown parameters. Each point gives two constrains and therefore three points form a minimal case. In general there are four possible solutions [7].

We will study the absolute orientation problem both for calibrated cameras as above, for the case of unknown focal length and for the uncalibrated camera case. Furthermore we will consider both known 3D-2D correspondences (U_i, u_i) as above and 2D-2D correspondences (v_i, u_i) with features v_i in other views. Here we will assume that the camera matrices of the other views are known, so that a 2D-2D correspondence can be thought of as a 3D-2D correspondence where the unknown 3D point U_i lies on a line expressed in Plücker coordinates. In this paper **the (m,n) case** denotes the case of m 2D-2D correspondences and n 3D-2D correspondences. Notice that each 2D-2D correspondence imposes one constraint and each 2D-3D correspondence imposes two constraints.

Calibrated Cameras For calibrated cameras there are six degrees of freedom, three for orientation and three for position. One way of parameterizing the camera matrix is to use a quaternions vector (a, b, c, d) for rotation, i.e.

$$P = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd & x \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab & y \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 & z \end{pmatrix}. \quad (1)$$

Potential minimal cases are:

The (0,3) case. This is the well known resection problem, cf. [7] with up to four solutions in front of the camera.

The (2,2) case. This is solved in this paper. The algorithms works equally well if the 2D-2D correspondences are to the same or to different cameras. There are up to 16 solutions.

The (4,1) case. This is solved in this paper. The case of all four 2D-2D correspondences coming from the same model image can be solved by first projecting the 3D point in the known camera and then using the five point algorithm to solve for relative orientation (hence up to 10 solutions) and then fixing scale with the final 2D-3D correspondence.

The (6,0) case. This cannot be solved for absolute orientation if all points are from the same model view. However, if the correspondences come from different views, it is similar to the relative orientation problem for generalized cameras, cf. [16], which has up to 64 solutions.

Unknown Focal Length For calibrated cameras with unknown focal length there are seven degrees of freedom, three for orientation, three for position and one for the focal length. One way of parameterizing the camera matrix is as

$$P=\begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd & x \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab & y \\ 2f(bd-ac) & 2f(ab+cd) & f(a^2-b^2-c^2+d^2) & fz \end{pmatrix}. \quad (2)$$

Potential minimal cases are

The (1,3) case. This case is solved in this paper. There are 36 solutions.

The (3,2) case. This is solved in this paper. There are 40 solutions.

The (5,1) case. This is solved in this paper. For the case of the 5 2D-2D correspondences coming from the same model view, it can be solved using the six point algorithm to solve for relative orientation and focal length [15] and then fixing scale with the final 3D correspondence. There are then up to 15 solutions.

The (7,0) case. This cannot be solved for absolute orientation if all points are to the same view. However for the case of correspondence to different view it is an open problem.

Uncalibrated Cameras For the uncalibrated camera case there are 11 degrees of freedom. Each 2D-2D correspondence gives one constraint and each 2D-3D correspondence gives two constraints. Potential minimal cases are

The (1,5) case. This can be solved by hand-calculations as follows. Using the five 3D-2D correspondences, the camera matrix can be determined up to a one-parameter family $P = P_1 + \nu P_2$, where P_1 and P_2 are given 3×4 matrices and ν is an unknown scalar. The remaining 2D correspondence can be parameterized as a point on a line $U = C + \mu D$ for some unknown parameter μ. The projection equation gives $\lambda u = PU = (P_1 + \nu P_2)(U_1 + \mu U_2)$. Using resultants, it follows easily that there are two solutions for the unknowns λ, ν, μ.

The (3,4) case. There are eight solutions, unless all four 2D-2D correspondences are from the same model view, in which the standard seven-point-two-view algorithm can be used. There are then up to three solutions.

The (1+2k,5-k) case with $k = 2, 3, 4$. These cannot be solved for absolute orientation if all points originate from one model view. However, for the case of correspondences from different model views, there are $2^{(1+2k)}$ solutions. The solutions procedure is analogous to the (1,5) case above and can be obtained using resultants.

Summary We conclude this section by summarizing all the minimal cases for hybrid 2D and 3D feature correspondences, see Table 1. We state an upper bound on the number of physically realizable solutions. In general, as we shall see later in Section 5, the number of plausible solutions is much smaller. In the next section, we give the remaining justifications to these claims and this will also lead to efficient algorithms for computing the solutions.

2D-2D corresp.	2D-3D corresp.	number of solutions	camera setting
0	3	4	calibrated
2	2	16	calibrated
4	1	32 or 10^*	calibrated
6	0	64	calibrated
1	3	36	unknown focal
3	2	40	unknown focal
5	1	112 or 15^*	unknown focal
7	0	?	unknown focal
1	5	2	uncalibrated
3	4	8 or 3^*	uncalibrated
$1+2k$	$5-k$	2^{1+2k}	uncalibrated

Table 1. Minimal hybrid cases for structure from motion. The number of solutions indicates an upper bound of the number of physically realizable solutions. The solution numbers marked with asterisk "*" correspond to cases where all 2D-2D correspondences originate from a single (model) view, whereas for other cases, it is implicitly assumed that the correspondence set covers multiple views. Note that one case is still an open problem (marked with "?").

4. Solving Minimal Cases with Algebraic Geometry

Minimal structure and motion problems typically boil down to solving a system of polynomial equations in a number of unknowns. For a problem instance the structure of the problems is quite fixed. Thus the number of solutions to a structure and motion problem, typically depends only on the type of problem at hand.

There are several techniques for determining the number of solutions for a class of polynomial equations systems. The theory of mixed volumes [5] can be used to prove the number of solutions for a set of polynomial equations assuming general coefficients of the polynomial. The software package `phc` [19] is useful both for calculating mixed volume and for finding solutions with homotopy methods. Another method is to calculate the so called Gröbner basis. For problems that are not synthetic (where the coefficients are represented as floating point approximations) finding the Gröbner base is error prone, since it can be difficult to establish if a certain coefficient is zero in the reduction. One technique here is to synthesize examples with integer coefficients and then projecting the equations from $R[x]$ to $Z_p[x]$

Once the number of solutions and the structure of the Gröbner basis is known, the procedure of obtaining a

Gröbner basis for the ideal I and a basis for the quotient space $C[x]/I$ can be obtained using numerical linear algebra. For problems with say less than 50 solutions, this procedure often has reasonable numerical stability. The solution process is briefly as follows. Given a set of polynomials $(p_1, \ldots, p_n)$ that have been obtained from the problem. Multiply each polynomial by a set of monomials. This set is obtained from the analysis above. This gives a larger set of polynomials. Represent these polynomials as a matrix product between the coefficient matrix C and a monomial vector m. By row reduction high order monomials can be expressed in terms of lower order monomials. If the set of product monomials is chosen large enough it is possible to express all polynomials in terms of a fixed number d of basis monomials. These form a basis for $C[x]/I$. By expressing the linear mapping $C[x]/I \ni p(x) \mapsto xp(x) \in C[x]/I$ in this basis an action matrix M is obtained. The eigenvectors of M^T gives the solutions to the polynomial equations. For further details, see [5].

4.1. Calibrated Cameras

A calibrated camera can be parameterized using quaternions as shown in (1) Assume that we have two correspondences between image points and scene points

$$u_1 \sim PU_1, \quad u_2 \sim PU_2.$$

Since there is a freedom in choosing coordinate systems both in the scene and in the images, these can be transformed into

$$U_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \; u_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \; U_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \; u_2 = \begin{pmatrix} 1 \\ 0 \\ u \end{pmatrix}.$$

This gives us the following constraints

$$x = 0, \;\; y = 0, \;\; ad = -bc,$$
$$z = u(a^2 + b^2 - c^2 - d^2) - 2bd + 2ac.$$

As the overall scale of the camera matrix is irrelevant, one can set $a = 1$ and eliminate d according to $d = -bc$. This makes it possible to parameterize the camera matrix as

$$P = \begin{pmatrix} (1+b^2)(1-c^2) & 4bc & 2c(1-b^2) & 0 \\ 0 & (1-b^2)(1+c^2) & -2b(1+c^2) & 0 \\ -2c(1+b^2) & 2b(1-c^2) & (1-b^2)(1-c^2) & z \end{pmatrix}.$$

By putting $a = 1$ two things happen. First the scale of the camera matrix is fixed, hence the left-hand 3×3 submatrix in (1) will only be a rotation matrix up to scale. This will not have any further impact on the problem since the measurement equations are homogeneous. The second consequence is that solutions with $a = 0$ will not be included. Since $a \in \mathbb{R}$ the probability for this is zero, but there might be problems if a is close to zero. However, as the synthetic experiments will show this is no serious problem.

Assume now that we have two correspondences between image points and points that have been seen in only one other model image. This gives two points on the viewing line C_i and D_i associated to a point v_i in the query image. If the line is represented with Plücker coordinates [7] and the camera is converted to the correspondent Plücker camera the constraints above converts to a single equation. It is further on easy to see that every nonzero element in the Plücker camera has a common factor of $1+b^2$. After removing the common factor, the constraint polynomials (p_1, p_2) are of order 2 in b and order 4 in c.

The dimension of the quotient space $C[b, c]/I$ is 16 with $I = (p_1, p_2)$ which can be checked with computer algebra [19]. By multiplying the polynomials with $(1, b, c, bc)$ we obtain 8 constraints in 24 monomials. It is then possible to express 8 of the monomials in terms of the remaining 16 monomials

$$(bc^4, b^3c^2, c^4, bc^3, b^2c^2, b^3c, c^3, bc^2, b^2c, b^3, c^2, bc, b^2, c, b, 1)$$

which then form a basis for the quotient space $C[b, c]/I$. From this it is straightforward to construct the 16×16 action matrix M for the linear mapping $C[b, c]/I \ni p(c) \mapsto cp(c) \in C[b, c]/I$. From the eigenvalue decomposition of the matrix M the 16 (some possibly complex) solutions are obtained. Similar calculations give that there are 32 solutions for the (4,1) case.

4.2. Unknown Focal Length

For the case of unknown focal length we have one additional unknown. Thus we need one extra constraint. There are several interesting minimal cases: (1,3), (3,2) and (5,1). However for the last case (assuming that all the five points were in correspondence with the same view) one could solve the relative orientation problem using the six point algorithm [15] and then fix the scale using the known 3D correspondence.

Using (2) as parameterization for the camera matrix and assuming that two of the 3D point correspondences are with

$$U_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \; U_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \; u_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

it is possible to eliminate $y = 0$ and $x = zf = g(b, c, d, f)$. We fix the scale by setting $a = 1$. For both the $(1, 3)$ case and the $(3, 2)$ case we get polynomial constraints in the five remaining unknowns (b, c, d, z, f). Calculations with computer algebra [19] suggests that there are 36 solutions for the $(1, 3)$ case, 40 solutions to the $(3, 2)$ case and 112 in the $(5, 1)$ case.

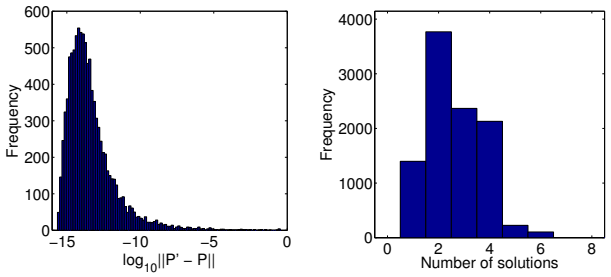


Figure 1. Statistics from the evaluation of the solver for the (2,2) case for calibrated cameras. The solver was run on 10.000 randomly generated cases. Left: Histogram over the error in matrix norm between the estimated camera P' and the true camera P. The error is plotted on a logarithmic scale. Right: Histogram over the number of real valued solutions yielding positive depths.

5. Validation on Synthetic Data

The purpose of this section is to evaluate the stability of the algorithm for solving the $(2, 2)$ case introduced in Section 4. To this end we use synthetically generated data in the form of randomly generated cameras and points. This allows us to measure the typical errors and the typical number of plausible solutions, over a large range of cases.

The point features are drawn uniformly from the cube ± 500 units from the origin in each direction. The cameras (two known and one unknown) are generated at approximately 1000 units from the origin pointing roughly in the direction of the center of the point cloud.

The algorithm has been run on 10.000 randomly generated cases as described above. To evaluate the accuracy of the solution we take the minimal error (over the plausible solutions) of the standard matrix 2-norm $\|P' - P\|$ of the difference between the estimated camera P' and the true camera P. The cameras were normalized by setting the last element to one. The result is illustrated in Figure 1. As can be seen, the error typically stays as low as 10^{-15} to 10^{-10}, but occasionally much larger errors occur. However, since the solver is used as a subroutine in a RANSAC engine, which relies on solving a large number of different instances, these very rare cases with poor accuracy are not a serious problem.

As shown in Section 4 the (2,2) calibrated case in general has 16 solutions. Since obviously only one of these solutions is the correct one it is interesting to investigate how many plausible solutions are typically obtained. With plausible solutions we mean real valued camera matrices which yield positive depths for all four problem points. In Figure 1 a histogram which shows the typical number of plausible solutions is given. As can be seen the most common situation is one to four plausible solutions. In one of the 10.000 cases, the algorithm was unable to find a real solution with positive depths for all points. This is probably due to numerical problems when the points and/or cameras are unfortunately positioned (two or more real solutions irrespective of the sign of the depths were found in all cases). In three of the cases seven solutions were found and in one case eight plausible solutions were found. The average number of plausible solutions was 2.6 and the average number of real solutions was 6.4. In some of the cases all 16 solutions were real.

Figure 2. Examples of experiment images. The left scene is called office and the right scene is called pump.

6. Application: Localization

The localization algorithm based on hybrid feature correspondences was tested on two data sets, one with images of offices and one set taken of a steel pump. Examples of the data sets can be seen in Figure 2. The sizes of the office and pump images are 1600×1200 and 2048×1360, respectively. In the following, the data sets are referred to as office and pump. The images used to test the localization performance have not been included when building the 3D model. For the office images, the test and model images were shot on separate days resulting in considerable differences.

6.1. Building the model

For the two data sets, the first step is to build a model of the 3D structure and to find the camera locations. The models consist of the location of the cameras used when building the model and points in the images that are localized with maximally stable extremal regions (MSER) [11]. SIFT descriptors [9] are also associated with all these areas. To do these calculations binaries from Oxford[1] were used with default parameter settings. The model also includes information about which points that correspond between images, and thus become potential 3D matches and which points that are just located in one image and hence become 2D correspondence candidates.

Office The office images were taken with a calibrated stereo rig. The cameras' inner parameters were calculated with a camera calibration toolbox available on the web[2].

[1] http://www.robots.ox.ac.uk/∼vgg/research/affine/index.html
[2] http://www.vision.caltech.edu/bouguetj/calib_doc

With the aid of this toolbox the distortion of the images was also calculated and all images were rectified. The stereo setup made it possible to compute epipolar geometry between the images pairwise. This was used to get a 3D structure between the two first images in the sequence. The next step was to add a new image from another stereo pair. This was done by first establishing correspondences between the new image and the calculated 3D points. Then with the three point solver and RANSAC, a first localization was estimated. After that bundle adjustment was applied for a local optimization of the location. The next step, again, was to use the known stereo data to triangulate new 3D points. This procedure was repeated until the model was finished. In the model for one office, the number of 2D points is typically ten times the number of 3D points.

Pump The pump scene had a known camera calibration and the location of the cameras was also given. In this sequence, two images were chosen to build the model with and the remaining images were used for testing. The known motion served as ground truth for the localization experiment (see result section). Because of the characteristics of these images (many specularities due to steel), it is difficult to get stable descriptors between images which makes it hard to find correct correspondences. As a result, there are not so many 3D correspondences to use for the further localization. In fact, from our automatic matching algorithm, there are only six 3D points and two of these are actually false. However, the number of 2D points is 493 and 802, respectively, in the two images.

6.2. Localization Method

The same localization scheme has been used for both image sets. As a comparison, the three point solver has also been used similar to the approach in [14]. The first step is to establish correspondences between the query image and the model. This is done by first locating the best matching correspondences to points that are located in 3D by exhaustive search. The 50 best matches are then stored. After that the 50 best matches to correspondences in the model images are calculated and stored. The 2-norm is used to measure the distances between two sift descriptors after they have been normalized according to [9].

When the 50 best 2D and 3D correspondences are fixed (or fewer if there not that many model points), the correspondence set is used in a RANSAC engine with 500 iterations. When one localization step is applied the next thing in the pipe is to decide which of the points that are inliers. This is done for the 3D points by measuring the reprojection error and if this is less then 10 pixels, the point is assumed to be an inlier. In the case of 2D, the point is first triangulated. After that, again, the reprojection error is measured but in this case the threshold is set to one pixel.

(a) (b)

Figure 3. Reprojected points from the estimated localization. Image (a) shows the points projected when the camera is placed according to the (2,2) solver and (c) when the camera is located as the three point solver proposed.

The camera location with the greatest number of inliers is then chosen. To improve the result further bundle adjustment is applied to the inlier data to get higher precision of the localization.

To compare the results, the three point solver for 3D-2D correspondences is applied [14]. The methodology used is the same as for the (2,2) solver. The localization step is executed and then the inliers are counted. Also in this algorithm we have chosen to iterate this 500 times. And as before when this is finished we make a bundle adjustment to improve the result.

6.3. Results

Office In most of the cases, the result of the localization is very similar, independent of which method used. This is occurs especially when the model includes many correctly located 3D points. A typical example of that is shown in Figure 3. In those images it is hard to see the differences between the two results when the three point solver and the (2,2) solver are used.

In some cases the proposed algorithm makes a better job than the three point solver. An example of that can be seen in Figure 4. In part (b) of that figure, the black crosses represent the reprojections of the 3D points in the model when the three point solver is used. If those are compared with the reprojections in one of the model images in Figure 4a the incorrect placement is obvious. The red triangles, the reprojection with the (2,2) solver, in Figure 4b however are close to the correct places, so in this case the (2,2) solver obtains a superior result compared to the state of the art, three point solver. In the model built for this example, many missmatches were created during the making of the 3D model and even more occurred in the matching of the model image, so there were not that many correct matches to use. One reason for the high number of incorrect matches was that the blackboard the day the model images were taken was filled, but when the test images were taken the blackboard was clean.

(a) (b)

Figure 4. Reprojected points from the estimated localization. Image (a) shows the points projected down in one of the model images, and hence, is a correct image to compare with. The red triangles in image (b) is the points projected when the camera is placed according to the (2,2) solver. These are close to correctly placed whereas the black crosses, placed according to the three point solver, are incorrectly located.

Out of 28 query images tested, there were 22 (79%) correctly localized with the (2,2) solver and there were 21 (75%) correctly classified with the three point solver. The reason for localization failure was mainly due to insufficient number of potential matches or a great number of incorrect located 3D points in the model. This shows that the (2,2) solver gives a slightly improved result already on its own and in combination with the three point solver, it is expected to further improve the accuracy of the localization.

Pump In the pump sequence, there are not that many stable points in the images to build the 3D model from and hence it can be useful to use hybrid features. When the model was built with the two model images only four 3D points were correctly determined. Additionally, two points are false matches and are thus incorrectly placed in the 3D space.

In the localization step, the matching phase makes additional miss-matches and then only two correct matches remains. This makes it impossible to use the three point solver, but two points is still enough to use the (2,2) solver. In Figure 5, the camera location and some scene points are plotted. To the left, the unknown camera is placed according to the (2,2) solver and to the right the unknown camera is placed as the three point solver proposes. The (2,2) solver places the camera correctly, as can be seen as the two close placed cameras, whereas the three point solver is not even close, as can be seen in the right part of Figure 5. In Figure 6 two of the model points are reprojected to one of the images. If the same points are reprojected according to how the (2,2) solver places the camera the result is as shown in Figure 7a. As can be seen, the reprojection is correct. The difference gets obvious if the result is compared to the three point solver. The result from the reprojection of the same points according to that algorithm is shown in Figure 7b.

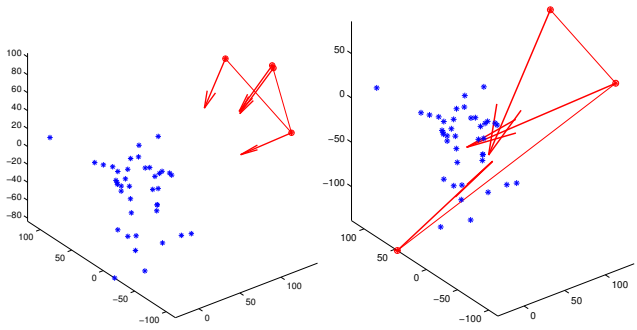


Figure 5. The estimated camera positions and the model cameras position in the pump scene. In the left image the result of the (2,2) solver and the ground truth is showed. The two close placed arrows are the localization with the (2,2) solver and the ground truth. In the right image the three point solver is used, in that image it is obvious that the result is not even close to the ground truth.

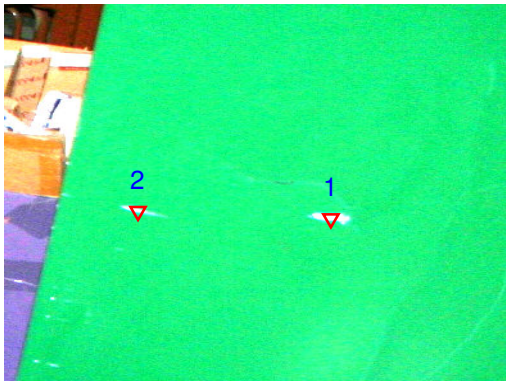


Figure 6. Model points reprojected down in one of the model images. These points are correctly placed and can thus be used to visually judge the results of the localizations illustrated in Figure 7

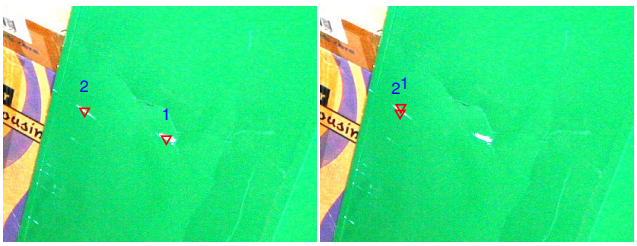
There, both points are reprojected to almost the same location and hence one of them is incorrectly placed.

As for the office images, 25 possible query images were tested. There were 12 (48%) correctly localized with the (2,2) solver and there were only 3 (12%) correctly classified with the 3-point solver. The reason for localization failure was mainly, in this sequence, due to insufficient number of potential matches. This shows that the (2,2) solver gives a considerably improved result in a setup like this.

7. Conclusions

In this paper we have shown new ways to use both 2D and 3D correspondences to solve the localization problem. Several minimal configurations have been classified and the number of solutions has been derived.

In the case of calibrated cameras with two 2D and two



(a) (b)

Figure 7. Reprojected points from the estimated localization. Image (a) shows the points projected when the camera is placed according to the (2,2) solver. These results are hard to distinguish from the correct places in Figure 6. In (b) the camera is located as the three point solver proposed. If this result is compared with Figure 6 it is obvious that the estimated location of the camera is incorrect.

3D correspondences experiments have been preformed. The synthetic experiments show that by using Gröbner basis methods, the calculations can be done numerical stable and it is known from before that they can be done fast. It is also shown that the number of feasible solutions in average is less than three even if the polynomial equations to solve give 16 solutions, in some cases all real.

In the experiments on real data it is shown that in some cases the three point solver, which is state of the art, is not capable of establishing a correct localization when the (2,2) solver is. This especially occurs when the model includes few correct 3D points or when the number of incorrectly placed 3D points is high. In most cases both algorithms give a similar result.

The logical way to extend the work of this paper is to test the remaining cases with semi calibrated (unknown focal length) and uncalibrated cameras. Another line of future work is to investigate how the proposed methods would be able to improve a complete structure and motion system.

Acknowledgment

This work has been funded by the Swedish Research Council through grant no. 2004-4579 'Image-Based Localisation and Recognition of Scenes', grant no. 2005-3230 'Geometry of multi-camera systems', SSF project VISCOS II and the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot.

References

[1] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Trans. on Robotics and Automation*, 13(2):251–262, 1997.

[2] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[3] W. Burgard, A. Cremers, D. Fox, D. Hhnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *National Conference on Artificial Intelligence (AAAI'98)*, Madison, Wisconsin, USA, 1998.

[4] M. Chasles. Question 296. *Nouv. Ann. Math.*, 14(50), 1855.

[5] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.

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

[8] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitz-Ber. Akad. Wiss., Wien, math. naturw. Kl. Abt*, IIa(122):1939–1948, 1913.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal Computer Vision*, 2004.

[10] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conf.*, pages 384–393, Cardiff, UK, September 2002.

[12] P. Newman, J. Leonard, J. Neira, and J.Tardos. Explore and return: Experimental validation of real time concurrent mapping and localization. In *Int. Conf. Robotics and Automation*, pages 1802–1809, 2002.

[13] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Conf. Computer Vision and Pattern Recognition*, volume II, pages 2161–2168, New York City, USA, 2006.

[14] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[15] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Conf. Computer Vision and Pattern Recognition*, volume 2, pages 789–794, San Diego, USA, 2005.

[16] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, Beijing China, Oct. 2005.

[17] P. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *Int. Conf. Computer Vision*, pages 727–732, Mumbai, India, 1998.

[18] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. Journal Computer Vision*, 59(1):61–85, 2004.

[19] J. Verschelde. Algorithm 795: Phcpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999.