

# Globally Optimal Inlier Set Maximization With Unknown Rotation and Focal Length

Jean-Charles Bazin<sup>1</sup>, Yongduek Seo<sup>2</sup>, Richard Hartley<sup>3</sup>, Marc Pollefeys<sup>1</sup>

<sup>1</sup>Department of Computer Science, ETH Zurich, Switzerland

<sup>2</sup>Department of Media Technology, Sogang University, South Korea

<sup>3</sup>Australian National University and NICTA, Canberra, Australia

**Abstract.** Identifying inliers and outliers among data is a fundamental problem for model estimation. This paper considers models composed of rotation and focal length, which typically occurs in the context of panoramic imaging. An efficient approach consists in computing the underlying model such that the number of inliers is maximized. The most popular tool for inlier set maximization must be RANSAC and its numerous variants. While they can provide interesting results, they are not guaranteed to return the globally optimal solution, i.e. the model leading to the highest number of inliers. We propose a novel globally optimal approach based on branch-and-bound. It computes the rotation and the focal length maximizing the number of inlier correspondences and considers the reprojection error in the image space. Our approach has been successfully applied on synthesized data and real images.

**Keywords:** Consensus set maximization, branch-and-bound, inlier detection, RANSAC.

## 1 Introduction

Distinguishing inliers and outliers among data is a fundamental problem and constitutes a necessary step for model estimation, notably in computer vision. An efficient approach to identify inliers and outliers consists in estimating the underlying model in such a way that the number of inliers is maximized. The most popular technique must be RANSAC [8] and has been applied for numerous computer vision tasks ranging from 3D reconstruction to object recognition. Despite its popularity and that interesting results can be obtained, RANSAC is not guaranteed to maximize the number of inliers in a globally optimal way. This paper is dedicated to rotational homography with unknown focal length, i.e. models composed of rotation and focal length, which typically occurs in the context of panoramic imaging [5]. We propose a globally optimal approach that computes the camera rotation and the focal length so that the maximum number of inlier correspondences between two images is guaranteed to be obtained.

Previous work investigated how to maximize the number of inliers. First, several variants of RANSAC have been proposed, for example MLESAC, LO-RANSAC and preemptive-RANSAC [22, 17, 19, 7]. While they generally perform

better than the original RANSAC in terms of the number of identified inliers, they are not guaranteed to obtain the optimal result. In contrast to random sampling, Li [13] applied an optimal branch-and-bound technique in combination with convex and concave envelopes [15]. However this approach is limited to distance definitions and constraints that are strictly linear with respect to the sought model. Kahl et al. [12] proposed an optimal method also based on branch-and-bound in combination with  $L_1$ -norm to partially reduce the sensitivity to outliers. A post-validation step was proposed by Olsson et al. [18]. While this approach is useful to verify the optimality of a potential solution, it does not provide a mechanism to explicitly compute the optimal solution. Bazin et al. [2] proposed an approach to maximize the number of inliers under a pure rotational model. In contrast to their work, (i) we do not assume that the focal length is known in advance, (ii) we compute the focal length, in addition to the rotation, (iii) instead of the angular error, we consider the meaningful Euclidian distance in the image space in pixels [10], which requires deriving the reprojection bounds in the image, and (iv) we introduce a rotation parametrization that permits to reduce the correlation between the focal length and the rotation parameters. Yang et al. [23] recently proposed a globally optimal Iterative Closest Point (ICP) algorithm for rigid registration (rotation + translation) of two 3D point sets and can be applied to maximize the number of inlier correspondences between these two sets. This method is dedicated to 3D point sets (e.g. registration error in 3D space) and thus cannot be straightforwardly generalized for unknown focal length and used for our application in the image space.

## 2 Formulation

Let us note  $\mathbf{x}_i = (x_i, y_i)$  and  $\mathbf{x}'_i = (x'_i, y'_i)$  the  $i^{\text{th}}$  input pair of 2D feature points in correspondence (e.g. obtained by SIFT [14]), respectively in the first and second images, with  $i = 1 \dots N$  and where  $N$  is the number of matches. The two images are taken with a camera located at a fixed position and turning with any 3D rotation  $R$ . It is assumed that the camera is intrinsically calibrated (e.g. camera center is known), except for the focal length  $f$  which is unknown and same for the two images. In the following, all the measurements ( $\mathbf{x}_i, \mathbf{x}'_i$ ) are centered, i.e. the camera center's coordinates ( $C_x, C_y$ ) are subtracted to the points' coordinates in pre-processing, and thus the intrinsic calibration matrix is reduced to  $K_f = \text{diag}(f, f, 1)$  [10]. In the absence of noise and outliers, any two measurements ( $\mathbf{x}_i, \mathbf{x}'_i$ ) in correspondence verify:

$$\mathbf{x}'_i = T\mathbf{x}_i = K_f R K_f^{-1} \mathbf{x}_i \text{ and } d_{R,f}(\mathbf{x}_i, \mathbf{x}'_i) = 0 \forall i \quad (1)$$

where  $d_{R,f}(\mathbf{x}, \mathbf{x}') = \|K_f R K_f^{-1} \mathbf{x} - \mathbf{x}'\|_2$ , that is the Euclidian distance in pixels between the measurement in the right image  $\mathbf{x}'$  and the measurement of the left image  $\mathbf{x}$  projected into the right image by the transformation  $T$  that depends on the camera rotation  $R$  and the focal length  $f$ . In the following,  $d_{R,f}(\mathbf{x}, \mathbf{x}')$  is called the reprojection error.

Due to noise and outliers, these relations might not be verified for all the input matches. Following the “residual tolerance method” [8], we define a match  $(\mathbf{x}_i, \mathbf{x}'_i)$  as an inlier if the reprojection error is lower than a residual tolerance  $\delta$ , i.e.  $d_{R,f}(\mathbf{x}_i, \mathbf{x}'_i) \leq \delta$ . Otherwise the match is considered an outlier.

Let  $\mathcal{S}$  represent the set of input matches:  $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}'_i), i = 1 \dots N\}$ . The set  $\mathcal{S}$  is partitioned into an inlier-set  $\mathcal{S}_I \subseteq \mathcal{S}$  containing the inlier matches and an outlier-set  $\mathcal{S}_O \subseteq \mathcal{S}$  with  $\mathcal{S}_O = \mathcal{S} - \mathcal{S}_I$ . The cardinality of  $\mathcal{S}_I$  corresponds to the number of inlier matches. Maximizing the number of inliers with unknown rotation and unknown focal length can now be formulated as:

$$\max_{\mathcal{S}_I, R, f} \text{card}(\mathcal{S}_I) \quad (2a)$$

$$\text{s.t. } d_{R,f}(\mathbf{x}_i, \mathbf{x}'_i) \leq \delta, \forall i \in \mathcal{S}_I \subseteq \mathcal{S} \quad (2b)$$

$$R \in SO(3) \quad (2c)$$

Solving System 2 in a globally optimal way is a challenging task mainly due to the non-linearity, the non-convexity and the rotation constraint. System 2 can also be considered as a typical chicken-and-egg problem: if the inliers are known, then the underlying model (rotation and focal length) can be computed [6], and reciprocally, if the model is known, the inliers can be retrieved (simple check of the inlier constraint at Eq. (2b)). Unfortunately neither the inliers nor the model is known apriori. A method would be to test all the possible combinations of inliers/outliers. While the number of combinations is finite ( $2^N$ ), it is generally untractable in practice. Another naive method would be to test all the possible models but the model search space has an infinite cardinality and thus is untractable. In practice, a popular method to solve System 2 is to apply RANSAC in combination with a 2-point algorithm [4], but as explained above, RANSAC is not guaranteed to return the globally optimal solution.

### 3 Proposed approach

This section presents the proposed approach to solve System 2 in a globally optimal way. We start by introducing a particular parametrization of the transformation  $T$ , then study the projection bounds when the parameters of the transformation lie in a given range, and finally explain how to use these bounds in the framework of branch-and-bound.

#### 3.1 Parametrization

Rotation can be parameterized in several ways such as Euler angles, quaternion and axis-angle. In our work, we propose to use a parametrization of the form

$$R = R_z(\theta) R_r R_z(\phi) \quad (3)$$

where  $R_z(\theta)$  is a rotation about the  $z$ -axis (optical axis) by an angle  $\theta$  and  $R_r$  is a rotation about the  $y$ -axis to be explained presently.

One key advantage of this parametrization is the way it interacts with  $K_f$  in the definition (1) of  $T$ . One may easily verify that  $K_f$  commutes with a rotation  $R_z$ , i.e.  $K_f R_z = R_z K_f$ . This important observation provides a simple form for  $T$ :

$$\begin{aligned} T &= K_f R K_f^{-1} = K_f R_z(\theta) R_r R_z(\phi) K_f^{-1} \\ &= R_z(\theta) K_f R_r K_f^{-1} R_z(\phi) . \end{aligned} \quad (4)$$

We define  $R_r$  as the rotation about the  $y$ -axis with the property that  $K_f R_r K_f^{-1}$  takes origin  $(0, 0, 1)^\top$  to the point  $(r, 0, 1)^\top$ . The mapping  $K_f R_r K_f^{-1}$  is of the form

$$K_f R_y(\alpha) K_f^{-1} = \begin{bmatrix} \cos(\alpha) & 0 & f \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha)/f & 0 & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} 1 & 0 & f \tan(\alpha) \\ 0 & 1/\cos(\alpha) & 0 \\ -\tan(\alpha)/f & 0 & 1 \end{bmatrix} \quad (5)$$

up to (irrelevant) scale. In order to take  $(0, 0, 1)^\top$  to  $(r, 0, 1)^\top$ , it follows that  $\tan(\alpha) = r/f$ , so  $\cos(\alpha) = f/\sqrt{f^2 + r^2}$ . For easier notation, let us consider the reciprocal focal length  $g = 1/f$  instead of the focal length. After some mathematical manipulations, we finally obtain the mapping

$$U(g, r) = K_f R_r K_f^{-1} = \begin{bmatrix} 1 & 0 & r \\ 0 & \sqrt{1 + r^2 g^2} & 0 \\ -r g^2 & 0 & 1 \end{bmatrix} . \quad (6)$$

Finally, the complete transformation  $T$  gets simplified to:

$$T = R_z(\theta) \begin{bmatrix} 1 & 0 & r \\ 0 & \sqrt{1 + r^2 g^2} & 0 \\ -r g^2 & 0 & 1 \end{bmatrix} R_z(\phi) . \quad (7)$$

**Interpretation** The two rotations  $R_z(\phi)$  and  $R_z(\theta)$  carry out rotations about the image origin. One may think of these as being rotations in the left and right hand images respectively. Thus, the way a point is transformed by  $T$  is as follows:

1. The point in the left image is rotated around the origin by an angle  $\phi$  by  $R_z(\phi)$ .
2. Then it is mapped into the right image by  $U(g, r)$  (see (6)), parameterized by  $g$  and  $r$  (i.e.  $f$  and  $\alpha$ ).
3. Finally, it is rotated around the origin of the right image by an angle  $\theta$  by  $R_z(\theta)$ .

An interesting observation is that in order to test a correspondence between points  $\mathbf{x}_i$  (in the left image) and  $\mathbf{x}'_i$  (in the right image), one may do a pre-rotation of  $\mathbf{x}_i$  by  $\phi$  and  $\mathbf{x}'_i$  by  $-\theta$ . In this operation, the circular inlier neighbourhoods (of radius  $\delta$ ) remain circular and the radius is unchanged.

All this is possible thanks to the rotation parametrization of (3) that permits  $R_z$  to commute with  $K_f$ , and in turn, to “separate”  $R_z$  and  $f$ . This reduces the correlation of  $f$  and the rotation parameters, which is desirable in the reduction of the error bound, and also simplifies the derivations of the bounds in pixel error (more details in the next sections).

**Additional properties** The mapping  $U(g, r)$  applied to a point  $\mathbf{x} = (x, y, 1)^\top$  provides a point  $\mathbf{x}'$  given by

$$\mathbf{x} = (x, y, 1) \rightarrow \mathbf{x}' = (x + r, y\sqrt{1 + r^2g^2}, 1 - rxg^2)^\top. \quad (8)$$

After dehomogenizing, this becomes

$$(x, y) \rightarrow (x', y') = \left( \frac{x + r}{1 - rxg^2}, \frac{y\sqrt{1 + r^2g^2}}{1 - rxg^2} \right). \quad (9)$$

An important note is that  $x'$  depends only on  $x$ , and not on  $y$ . In particular, this mapping takes a (vertical) line of the form  $x = c$  to a new line  $x' = (c + r)/(1 - rxg^2)$ . This will be used for a quick-and-easy way of checking whether a match is a potential inlier given a certain transformation.

Another important property is the “trajectory” of the point  $(x, y)^\top$  mapped into the right image as  $r$  varies. From (9), one can show that this trajectory is a hyperbola with the equation

$$(1 + g^2x_0^2)(y')^2 - (g^2y_0^2)(x')^2 = y_0^2 \quad (10)$$

where  $(x_0, y_0)$  represents the initial point when  $r = 0$ . The initial point  $(x_0, y_0)$  is the same as  $(x, y)$ , but written with the subscript so as to indicate that this point is constant here and  $(x', y')$  are the variables.

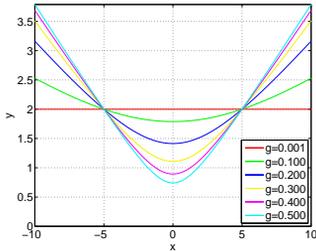
### 3.2 Intervals of $r$ and $g$

This section investigates where a point  $\mathbf{x} = (x, y)$  is mapped to under a transformation  $U(g, r)$ , when the parameters  $r$  and  $g$  lie in a range. Let us suppose that  $r_{\min} \leq r \leq r_{\max}$  and  $g_{\min} \leq g \leq g_{\max}$ . With these intervals, the mapped point  $U(g, r)\mathbf{x}$  must lie inside some bounded region in the right image and in the following, we compute the bounds on its  $x$  and  $y$  coordinates.

**Bounds on  $y$ .** Based on the above observations and derivations,  $U(g, r)\mathbf{x}$  must lie between the two hyperbolas defined by (10) for  $g = g_{\min}$  and  $g = g_{\max}$ . This gives a very convenient way to determine whether the mapping of  $\mathbf{x}$  into the right image passes “close” to (i.e. up to  $\delta$ ) its putative corresponding point  $\mathbf{x}'$  for any value of  $r$ . The hyperbolas are illustrated in Figure 1. This figure shows that for values of  $g$  close to zero (very large focal lengths), the point moves almost horizontally, whereas for larger values of  $g$  (shorter focal lengths) the trajectory becomes more curved.

**Bounds on  $x$ .** As shown previously, where a point  $\mathbf{x} = (x, y)$  maps to does not depend on  $y$ , only on  $x$  (see (9)). As  $y$  varies in the point  $(x, y)$ , the mapped point  $(x', y')$  varies along a vertical line as well. Thus,  $U$  maps vertical lines to vertical lines. Taking derivatives of the line  $x' = (x + r)/(1 - rxg^2)$  with respect to  $r$  and  $g$  gives

$$\frac{\partial x'}{\partial r} = \frac{1 + g^2x^2}{(-1 + g^2rx)^2} \text{ and } \frac{\partial x'}{\partial g} = \frac{2grx(r + x)}{(-1 + g^2rx)^2}. \quad (11)$$



**Fig. 1.** Hyperbolic trajectories of a point transformed by  $\mathcal{U}(g, r)$  with different values of  $g$ , given the initial point  $(x_0, y_0) = (5, 2)$  and for any values of  $r$ .

The derivative with respect to  $r$  is always positive. Therefore, the minimum value  $x_{min}$  of  $x'$  will be achieved at  $r = r_{min}$  and the maximum  $x_{max}$  at  $r = r_{max}$ .

The sign of the derivative with respect to  $g$  corresponds to the sign of  $2grx(r+x)$ , which depends on the values of  $x$  and  $r$  (note that  $g = 1/f \geq 0$  since focal length is always positive). Let us study the minimum of  $x'$ . As explained right above, it is achieved with  $r = r_{min}$ . Let note  $s = \text{sign}(r_{min}x(r_{min} + x))$ . Thus the sign of the derivative is the same as  $s$ : if  $s \geq 0$  (rec.  $s \leq 0$ ) then the derivative is positive (rec. negative), then the minimum of  $x'$  is obtained at  $g = g_{min}$  (rec.  $g = g_{max}$ ). Concretely, the minimum value of  $x'$  is achieved at one of the two parameter values  $(r, g) = (r_{min}, g_{min})$  or  $(r_{min}, g_{max})$  depending on the sign of  $r_{min}x(r_{min} + x)$ . A similar derivation can be followed for the maximum of  $x'$ : the value at which  $x'$  takes its maximum value is at either  $(r, g) = (r_{max}, g_{min})$  or  $(r_{max}, g_{max})$  depending on the sign of  $r_{max}x(r_{max} + x)$ .

Summarizing this, the minimum and maximum of  $x'$  are achieved at two of the four corners of the rectangle defined by the range of the parameters  $r$  and  $g$ . Thus, we see that we can bound the range of the point  $\mathbf{x}' = (x', y')$  to lie between two vertical lines  $x' = x_{min}$  and  $x' = x_{max}$  and to lie between the two hyperbolas corresponding to the values of  $g_{min}$  and  $g_{max}$ . This gives a simple test to see whether the point  $\mathbf{x}$  can transform to the target point  $\mathbf{x}'$  within a suitable radius (i.e. up to  $\delta$ ). Concretely, the test can be conducted as follows:

1. In terms of the initial point  $\mathbf{x} = (x_0, y_0)$  compute the bounds  $x_{min}$  and  $x_{max}$  and test if  $x'$  lies between them.
2. Test if  $\mathbf{x}'$  lies between the two hyperbolas defined for  $g_{min}$  and  $g_{max}$  and with the initial point  $(x_0, y_0)$ .

If either of these two tests fails, then it is sure that  $\mathbf{x}'$  is not within range. Otherwise,  $\mathbf{x}'$  *might* be within range:  $(\mathbf{x}, \mathbf{x}')$  is a potential inlier match but this has to be confirmed with further investigations, as detailed in the next section.

### 3.3 Branch-and-bound

To efficiently deal with bounds, we follow the branch and bound algorithm (noted BnB in the following). BnB is a general framework for global optimization [11] that recently gained popularity in the field of computer vision [21, 24, 9]. Given

a search space of the model to estimate, BnB iteratively subdivides this search space into smaller subspaces, identifies and removes the subspaces that do not contain the optimal solution via a feasibility test and refines the remaining subspaces, i.e. the subspaces that can potentially contain the optimal solution.

Our search space is composed of the rotation space  $SO(3)$  parameterized by the angles  $(\theta, \phi, \alpha)$  (see Section 3.1) and the focal length range. We call “cube” a delimited part of that space. We now ask the question: how to estimate the lower and upper bounds of the number of inliers that can be obtained by any models in a given cube?

*Lower bound.* Let us start with the lower bound. The center of a given cube  $\mathcal{C}$  corresponds to a specific point in the search space and thus corresponds to known rotation angles  $(\theta, \phi, \alpha)$  and focal length  $f$ , which in turn provides  $R$  and  $K_f$ . Then we can simply count the number of matches  $(\mathbf{x}_i, \mathbf{x}'_i)$  verifying the inlier constraint (2b). One may note that this number does not necessarily correspond to the lowest number of inliers that could be obtained in the cube  $\mathcal{C}$ , but provides a practical lower bound of the maximum number of inliers that can be obtained in the cube  $\mathcal{C}$ .

*Upper bound.* Computing the upper bound is more challenging. We note  $[\theta_l, \theta_u]$ ,  $[\phi_l, \phi_u]$ ,  $[\alpha_l, \alpha_u]$  and  $[f_l, f_u]$  the definition ranges of these parameters in a given cube  $\mathcal{C}$ . The rotation of a point  $\mathbf{x}_i$  (in the left image) by  $R_z(\phi)$  with  $\phi \in [\phi_l, \phi_u]$  defines a circular arc centered at the origin and of length  $\phi_u - \phi_l$ . The  $x$  and  $y$  bounds of this arc can be easily obtained analytically and we note them  $([x_i], [y_i]) = ([x_{i,l}, x_{i,u}], [y_{i,l}, y_{i,u}])$  where  $l$  and  $u$  stand for lower and upper values. Similarly, let us note  $([x'_i], [y'_i]) = ([x'_{i,l}, x'_{i,u}], [y'_{i,l}, y'_{i,u}])$  the bounds of the point  $\mathbf{x}'_i$  (in the right image) rotated by  $R_z(\theta)$  with  $\theta \in [\theta_l, \theta_u]$ .

In Section 3.2, we investigated how the ranges of  $r$  and  $g$  (i.e.  $\alpha$  and  $f$ ) influence the mapping  $U$  and we derived the  $x$  and  $y$  bounds of the point  $\mathbf{x}$  mapped by  $U$ . This approach can be easily generalized to an interval of point  $([x_i], [y_i])$  instead of a given point  $\mathbf{x}$ . The “extended”  $x$  and  $y$  bounds of the interval  $([x_i], [y_i])$  mapped by  $U$  can be computed in a similar way, analytically or by simple interval arithmetics [16]. This provides the interval of  $\mathbf{x}_i$  mapped by  $U$  in the right image (obtained by the definition ranges of  $\phi$ ,  $\alpha$ ,  $f$ ) and we compare this interval to the interval of  $([x'_i], [y'_i])$  (obtained by the definition range of  $\theta$ ). If the intervals intersect (up to  $\delta$ ), then the match  $(\mathbf{x}_i, \mathbf{x}'_i)$  is a potential inlier under a model contained in  $\mathcal{C}$ . If they do not intersect, then it is definitively an outlier for the cube  $\mathcal{C}$ . We conduct this procedure for all the input matches and count the number of potential inliers, which in turn provides an upper bound of the number of inliers for  $\mathcal{C}$ .

*Search strategy.* We explained above how to compute the lower and upper bounds of the number of inliers for a given cube. We now discuss how to iteratively discard the non-feasible cubes and conduct the search strategy.

First of all, the search starts with a cube list  $\mathcal{L}$  that is initialized with one cube covering the entire search space (see above). During the subdivision of a

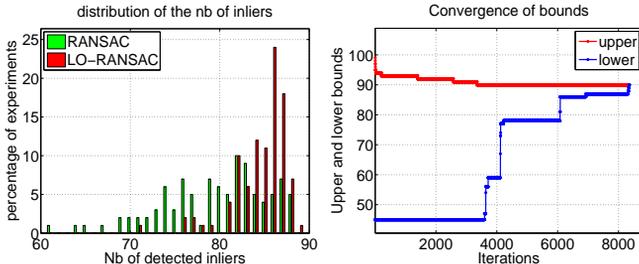
cube  $\mathcal{C}_j$ , each dimension of this cube is split into two intervals of equal length, which provides, in total,  $2^4 = 16$  disjoint smaller cubes whose side length is half of the side length of the original cube  $\mathcal{C}_j$ . The cube  $\mathcal{C}_j$  is removed from  $\mathcal{L}$  and replaced by its subdivided cubes.

Let  $l^*$  be the highest lower bound obtained so far or the number of inliers computed beforehand by any existing methods. Let us note  $l_j$  and  $u_j$  the lower and upper bounds of the number of inliers of a cube  $\mathcal{C}_j$ . If  $u_j < l^*$  then it is sure that the optimal model (i.e. the model leading to the highest number of inliers) is not contained in  $\mathcal{C}_j$  because  $\mathcal{C}_j$  does not contain any models that can lead to at least or more than  $l^*$  inliers. Thus the cube  $\mathcal{C}_j$  is considered non-feasible and can be removed from the search space. On the contrary (i.e. if  $u_j \geq l^*$ ) then the cube  $\mathcal{C}_j$  is subdivided for further investigation. In case  $l_j > l^*$  then  $l^*$  is updated by  $l_j$ . This procedure is applied iteratively for all the cubes contained in the cube list  $\mathcal{L}$ . We define the maximum upper bound  $u^*$  as the highest upper bound among the feasible cubes currently present in the list  $\mathcal{L}$ . Along the BnB iterations, the non-feasible cubes are discarded (the search space reduces), the size of the cubes decreases (by subdivision), the gap between the lower and upper bounds  $l_j$  and  $u_j$  computed for each cube  $\mathcal{C}_j$  diminishes, and  $l^*$  and  $u^*$  converge. The search stops when the list  $\mathcal{L}$  contains at least one cube  $\mathcal{C}_j^*$  whose lower bound  $l_j$  equals the maximum upper bound  $u^*$  because it means that the model at the center of  $\mathcal{C}_j^*$  (i.e. used to compute  $l_j$ ) provides the maximum number of inliers  $u^*$ . Finally the BnB procedure simply returns the model  $(\mathbf{R}^*, f^*)$  associated to the center of  $\mathcal{C}_j^*$  and this model leads to the maximum number of inliers  $u^*$  that can be obtained inside the search space.

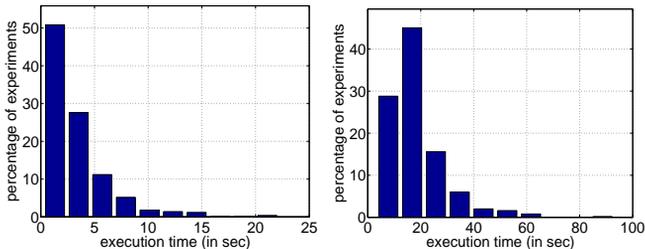
## 4 Results

This section presents some experimental results obtained for synthesized data and real images. Implementation details, additional algorithmic explanations and supplementary results are available on the authors' website. Our branch-and-bound approach has been implemented in C++ and ran on a computer equipped with an Intel Core i7 CPU 2.8GHz (a single core is used) and 12GB RAM. We run BnB on the whole rotation search space  $SO(3)$  and a conservative realistic range for the focal length  $[200, 4500]$ , unless otherwise stated. Our approach takes between a few seconds and a few minutes depending mainly on the number of points and the search space size.

We compare our approach to the conventional RANSAC and the optimized LO-RANSAC [7], both being referred to as RANSACs. They both embed the state-of-the-art minimal solution approach of Brown et al. [4], that we refer to as the 2-point algorithm. We consider LO-RANSAC in addition to the conventional RANSAC, because it is known to perform very well in practice. Among the versions of LO-RANSAC, we apply the *inner RANSAC with iteration* because it has been shown that it provides the best results [7]. The number of RANSAC iterations is automatically computed with the true outlier ratio (if not available, e.g. for real data, set to 80%), a guaranteed accuracy of 99% and a minimal sampling of 2 points for the 2-point algorithm [10]. Since different runs of RANSACs might



**Fig. 2.** Comparison between RANSACs and the proposed approach. Left: distribution of the number of inliers obtained by RANSAC and LO-RANSAC. Their best run leads to 88 and 89 inliers respectively. Right: convergence of the BnB bounds to 90 inliers.



**Fig. 3.** Distribution of the execution time of our approach for  $N = 30$  (left) and  $N = 300$  data points (right) with 70% of outliers.

lead to different results (because of the random data sampling), we repeat each experiment over 1000 runs with the same input data and parameters. Finally, the algorithms are compared with respect to the number of inliers detected.

#### 4.1 Synthesized data

We randomly generate a set of  $N = 300$  correspondences of 2D points between the left and right images. To reproduce realistic settings, we corrupt the  $x$  and  $y$  coordinates of the points by a Gaussian noise ( $\text{std}=0.5\text{pixels}$ ) and create a percentage  $p = 70\%$  of outliers with  $\delta = 2\text{pixel}$ . Figure 2 compares the number of inliers obtained by the proposed approach and RANSACs. Our BnB approach obtains 90 inliers, which corresponds to the number of synthesized inliers. On the contrary, RANSAC spans between 61 and 88 inliers and never obtains the number of synthesized inliers, that is 90. As expected, LO-RANSAC performs better than RANSAC: results span between 71 and 89 inliers. While it does not obtain the number of synthesized inliers neither, the distribution is clearly improved: RANSAC obtained more than 84 inliers in 26% of the 1000 runs, whereas it occurred in 72% of the runs for LO-RANSAC.

One might wonder why neither RANSAC nor LO-RANSAC managed to obtain the true number of inliers. This actually happened in about 80% of our experiments for RANSAC and 65% for LO-RANSAC. This can be explained by two reasons. First, the performance of RANSACs depends on the data noise:

for completeness, we conducted several experiments with noise-free data, but still corrupted by outliers, and the true number of inliers was obtained in a percentage of experiments similar to the selected guaranteed accuracy of 99%. Second, RANSAC can hypothesize only models that are directly supported by the selected points. The concrete consequence is that RANSAC cannot return the optimal number of inliers if the associated model cannot be hypothesized by the minimal points. Thanks to its local optimization steps, LO-RANSAC performs better than RANSAC, but still relies on the data points support to hypothesize models, and therefore this limitation is removed only partially.

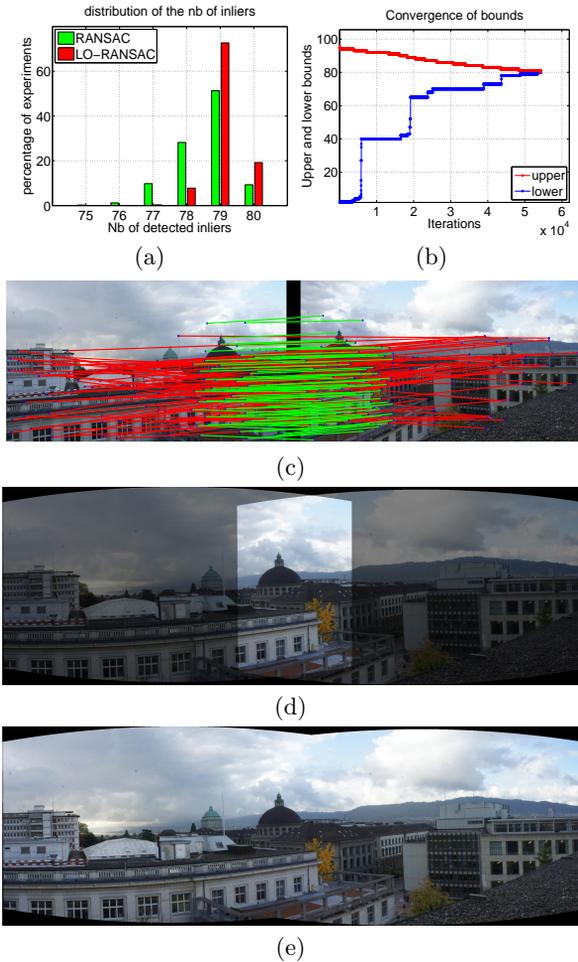
For a complete comparison of RANSACs and our BnB approach, we conducted more than 100,000 experiments with various data amounts ( $N = 10 \sim 600$  points), proportions of outliers ( $p = 0\% \sim 90\%$ ) and focal length ranges ( $[500, 700]$  to  $[10, 5000]$ ). In *all* the experiments, the BnB bounds always converged to the true number of inliers, and the number of inliers obtained by our approach was always higher than or equal to the number obtained by RANSAC and LO-RANSAC.

We conducted 1000 different experiments with newly randomly generated data and measured the execution time of our BnB approach for each experiment. The distribution of the execution time for  $N = 30$  and  $N = 300$  data points (70% of outliers) is illustrated in Figure 3. As expected, the execution time increases with the number of points since there are more data to process, and the figure shows that the approach is scalable with the data amount. RANSACs definitely run faster than the BnB approach but we do not aim to compete with RANSACs in terms of speed: our key goal is to obtain the globally optimal solution.

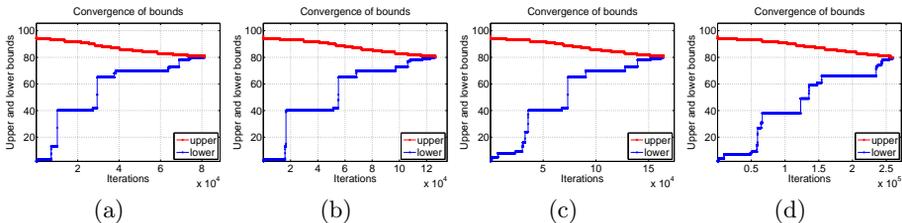
## 4.2 Real data

We now present results obtained by our approach on real images. We perform the intrinsic calibration of the camera by Bouguet’s toolbox [3], and the camera center is then applied to the point measurements to center their coordinates. Putative correspondences between two images are obtained by extracting and matching SIFT features [14]. The number of RANSAC iterations is computed with a very conservative outlier ratio of 80% and we run RANSACs 1000 times for each experiment. In the case the focal length and its limits are totally unknown, we use a conservative focal length range  $[200, 4500]$ . This focal length range covers most of the practical cases and can be enlarged if needed. Optionally, a rough focal length estimation and/or its limits can be obtained via EXIF tags like in  $[1, 20]$  or known approximately from the camera device.

A first representative result with a small overlap is shown in Figure 4. The images are acquired with a Sony NEX-3 camera. A rough estimate of the focal length is available from the EXIF tag and we use a range of  $\pm 200$  around this value (we will show results without any EXIF tag information in Figure 5). 163 putative correspondences are obtained by SIFT. The number of inliers obtained by RANSAC and LO-RANSAC span between 75 and 80, and between 77 and 80 respectively, as shown in Figure 4-(a). Their maximum number of inliers, that is 80, is obtained during only 9% of the 1000 runs by RANSAC, and 19% by

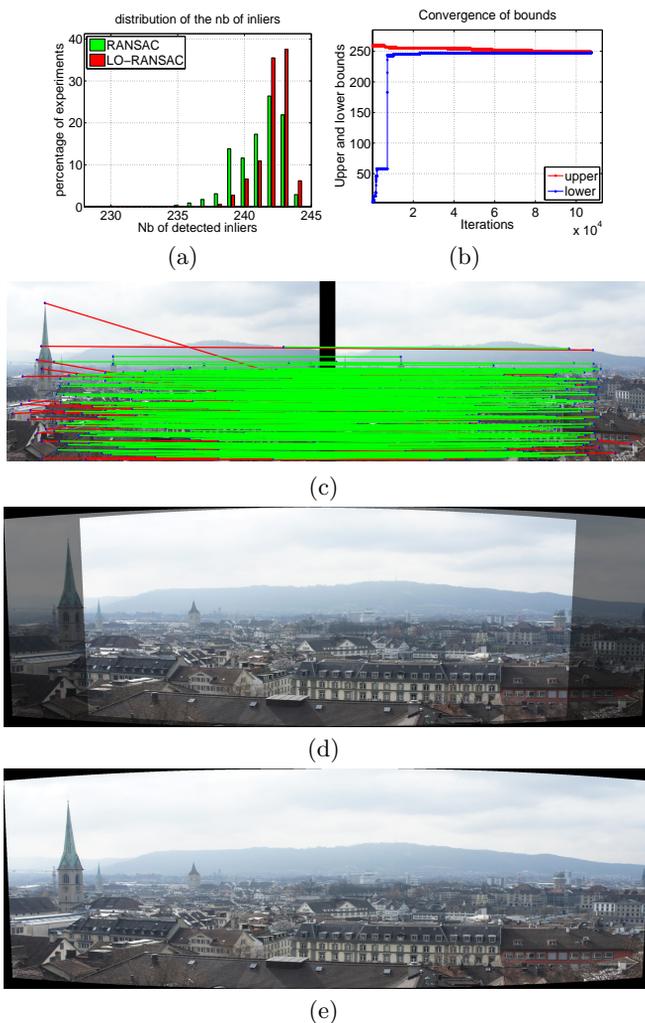


**Fig. 4.** (a) Distribution of the number of inliers obtained by RANSAC and LO-RANSAC. (b) Convergence of the BnB bounds to 80 inliers. (c) Inlier (green) and outlier (red) matches detected by our approach. (d) Highlighted overlap of the input images on the resulting panoramic view. (e) The final panoramic view.

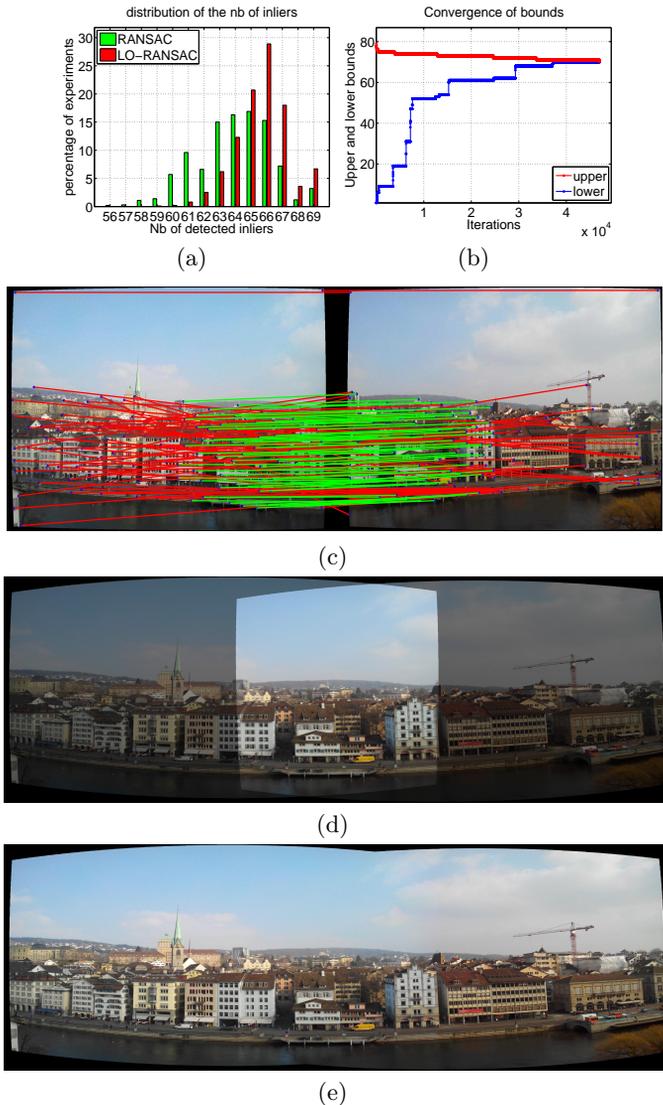


**Fig. 5.** Convergence of the BnB bounds to 80 inliers with a focal length range of  $\pm 300$  (a),  $\pm 500$  (b),  $\pm 1000$  (c) around a rough focal length estimation, and with a very conservative focal length range of  $[200, 4500]$  (d).

LO-RANSAC. The BnB bounds converge to 80 inliers, as shown in Figure 4-(b). The inlier and outlier matches identified by our BnB approach are shown in Figure 4-(c), and the resulting panoramic view in Figure 4-(d,e). For completeness, Figure 5 illustrates the evolution of the BnB bounds with different focal length ranges for the image pair of Figure 4. First it shows that the bounds always converged. Also, a focal length range of  $\pm 200$  took about  $5.4 \times 10^4$  iterations (see Figure 4-(b)) and a range of  $\pm 1000$  took about  $16 \times 10^4$  iterations (see Figure 5-(c)), that is about 2.96 times more iterations while the search space is 5 times larger. This shows that the method is scalable with the focal length range. Figure 5-(d) shows the convergence of the BnB bounds with a very conservative focal length range, which shows that our method can be applied for practical cases when the focal length and its limits are totally unknown.



**Fig. 6.** Same legend as in Figure 4. The best run of RANSAC and LO-RANSAC leads to 244 inliers and our BnB bounds converge to 248 inliers.



**Fig. 7.** Same legend as in Figure 4. The best run of RANSAC and LO-RANSAC leads to 69 inliers and our BnB bounds converge to 70 inliers.

An additional representative result with a large overlap is shown in Figure 6. 275 putative correspondences are obtained by SIFT. The number of inliers obtained by RANSAC and LO-RANSAC span between 229 and 244, and between 238 and 244 respectively. Their maximum number of inliers, that is 244, is obtained during only 3%, by RANSAC, and 6%, by LO-RANSAC, of the 1000 runs. In contrast, our BnB bounds converge to 248 inliers, which is 4 additional inliers than the best result of RANSAC and LO-RANSAC among the 1000 runs. For this image pair, the lower bound increases very quickly to 243 inliers, and

the gap between the lower and the upper bounds reduces slowly. This is because the reprojection errors of some correspondences were close to the inlier threshold and thus the size of the cubes (i.e. the uncertainty of the model) needs to be sufficiently small to be able to decide whether such correspondence is inlier or outlier, and this cube size is continuously reduced along the BnB iterations.

We also apply our approach on images acquired by a smartphone. Figure 7 shows a representative result with a conservative focal length range: the RANSAC distribution, the convergence of the BnB bounds, the inlier/outlier correspondences obtained by our approach and the resulting panoramic view. Some additional results for different scenes and cameras are available on the authors' website. Similarly to the experiments with synthesized data, the number of inliers obtained by our approach on *all* our experiments with real images was always higher than or equal to the number of inliers obtained by RANSAC and LO-RANSAC, which confirms the validity of our approach.

## 5 Conclusion

This paper faced the problem of inlier set maximization in the image space with unknown rotation and focal length. The most popular approach to solve this problem is RANSAC but it has several limitations especially the lack of optimality. We proposed a new approach based on branch-and-bound that maximizes the inlier set in a globally optimal way: it returns the rotation and the focal length leading to the highest number of inlier correspondences in the image space. The validity of the approach has been confirmed by experiments on synthesized data and real images.

An interesting direction for future work would be to estimate, in addition to the focal length, some extra intrinsic parameters, such as the camera center. We also plan to investigate the generalization of our approach of inlier set maximization in the image space beyond pure-rotation motion and for other models. Especially, in the context of fundamental matrix estimation, the goal would be to compute the rotation and translation of the camera, as well as the focal length, in such a way that the number of inlier correspondences in the image is maximized, up to an inlier threshold in pixels (e.g. with respect to the distance to the epipolar line).

## Acknowledgments

This research, which has been partially carried out at BeingThere Centre, was supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. The work was also supported by the Sogang University Research Grant of 201210029. NICTA is funded by the Australian Government through the ARC. Yongduek Seo is the corresponding author.

## References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: ICCV (2009)
2. Bazin, J.C., Seo, Y., Pollefeys, M.: Globally optimal consensus set maximization through rotation search. In: ACCV (2012)
3. Bouguet, J.Y.: Camera calibration toolbox for Matlab
4. Brown, M., Hartley, R., Nister, D.: Minimal solutions for panoramic stitching. In: CVPR (2007)
5. Brown, M., Lowe, D.: Automatic panoramic image stitching using invariant features. IJCV (2007)
6. Choi, K., Lee, S., Seo, Y.: A branch-and-bound algorithm for globally optimal camera pose and focal length. *Image and Vision Computing* (2010)
7. Chum, O., Matas, J.: Optimal randomized RANSAC. TPAMI (2008)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* (1981)
9. Hartley, R., Kahl, F.: Global optimization through rotation space search. IJCV (2009)
10. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edn. (2004)
11. Horst, R., Tuy, H.: *Global optimization: deterministic approaches*. Springer Verlag (2006)
12. Kahl, F., Agarwal, S., Chandraker, M., Kriegman, D., Belongie, S.: Practical global optimization for multiview geometry. IJCV (2008)
13. Li, H.: Consensus set maximization with guaranteed global optimality for robust geometry estimation. In: ICCV (2009)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2003)
15. McCormick, G.: Computability of global solutions to factorable nonconvex programs: part I - convex underestimating problems. *Mathematical Programming* (1976)
16. Moor, R.: *Interval Analysis*. Prentice-Hall (1966)
17. Nistér, D.: Preemptive RANSAC for live structure and motion estimation. In: ICCV (2003)
18. Olsson, C., Enqvist, O., Kahl, F.: A polynomial-time bound for matching and registration with outliers. In: CVPR (2008)
19. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: ECCV (2008)
20. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH (2006)
21. Sun, M., Telaprolu, M., Lee, H., Savarese, S.: An efficient branch-and-bound algorithm for optimal human pose estimation. In: CVPR (2012)
22. Torr, P., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* (2000)
23. Yang, J., Li, H., Jia, Y.: Go-ICP: solving 3D registration efficiently and globally optimally. In: ICCV (2013)
24. Yu, C., Seo, Y., Lee, S.W.: Global optimization for estimating a BRDF with multiple specular lobes. In: CVPR (2010)