Pattern Recognition 93 (2019) 36–54

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A family of globally optimal branch-and-bound algorithms for 2D–3D correspondence-free registration

Mark Brown[a], David Windridge[a,b], Jean-Yves Guillemaut[a,*]

[a] Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford GU2 7XH, United Kingdom
[b] School of Science and Technology, Middlesex University, London NW4 4BT, United Kingdom

ARTICLE INFO

Article history:
Received 17 August 2018
Revised 15 February 2019
Accepted 4 April 2019
Available online 4 April 2019

Keywords:
2D–3D registration
Multi-modal registration
Branch-and-bound
Global optimisation

ABSTRACT

We present a family of methods for 2D–3D registration spanning both deterministic and non-deterministic branch-and-bound approaches. Critically, the methods exhibit invariance to the underlying scene primitives, enabling e.g. points and lines to be treated on an equivalent basis, potentially enabling a broader range of problems to be tackled while maximising available scene information, all scene primitives being simultaneously considered. Being a branch-and-bound based approach, the method furthermore enjoys intrinsic guarantees of global optimality; while branch-and-bound approaches have been employed in a number of computer vision contexts, the proposed method represents the first time that this strategy has been applied to the 2D–3D correspondence-free registration problem from points and lines. Within the proposed procedure, deterministic and probabilistic procedures serve to speed up the nested branch-and-bound search while maintaining optimality. Experimental evaluation with synthetic and real data indicates that the proposed approach significantly increases both accuracy and robustness compared to the state of the art.

1. Introduction

This paper deals with the general problem of 2D–3D registration where given an image taken by a calibrated camera and a 3D model, the objective is to determine the pose of the camera with respect to the model. While there exist established solutions to this problem in the case where correspondences are known, there are many situations where it is not possible to reliably extract such correspondences across modalities, thus requiring the use of a correspondence-free registration algorithm. Existing correspondence-free methods rely on local search strategies and consequently have no optimality guarantee. In this paper we present a family of globally optimal solutions to the 2D–3D registration problem from points and lines without correspondences and in the presence of outliers. Fig. 1 illustrates how these solutions can be used within a 2D–3D registration pipeline. 2D–3D registration finds use in a range of tasks such as motion segmentation [1], object localisation and recognition [2], with practical applications in many areas including vehicle navigation [3], media visualisation [4], medicine [5,6] and forensics [7].

Despite considerable progress in feature extraction and single-modality registration (e.g. 2D–2D or 3D–3D), the general 2D–3D registration problem remains challenging. While there exist techniques to extract features in the 2D and 3D domains (e.g. corners [8], salient features [9] or lines [10,11]), it is an open problem to automatically establish correspondences between them. This may be explained by a variety of reasons. First, feature appearance can vary dramatically between 3D and its 2D projection due to the non-linear nature of the transformation; a 3D feature may be projected from a large range of viewpoints and perspective distortion may occur as well as view-dependent appearance variations if the material is non-Lambertian. Second, in the specific case of lines, there are many scenes where it is difficult to establish correspondences based on appearance, for example in highly repetitive man-made scenes or where low-width structures are present [12]. Finally, and more generally, correspondences of any feature type are particularly difficult to hypothesise when the 3D model is untextured, as is often the case if it is obtained by a laser range scanner.

The lack of feature correspondences renders traditional hypothesise-and-test approaches (e.g. RANSAC [13]) practically obsolete due to the very high computational complexity of the problem. State-of-the-art approaches e.g. [14,15] search over the transformation space and scale cubically with the number of features, but are not robust to the high rates of outliers required

* Corresponding author.
E-mail addresses: m.r.brown@surrey.ac.uk (M. Brown), d.windridge@mdx.ac.uk (D. Windridge), j.guillemaut@surrey.ac.uk (J.-Y. Guillemaut).

https://doi.org/10.1016/j.patcog.2019.04.002

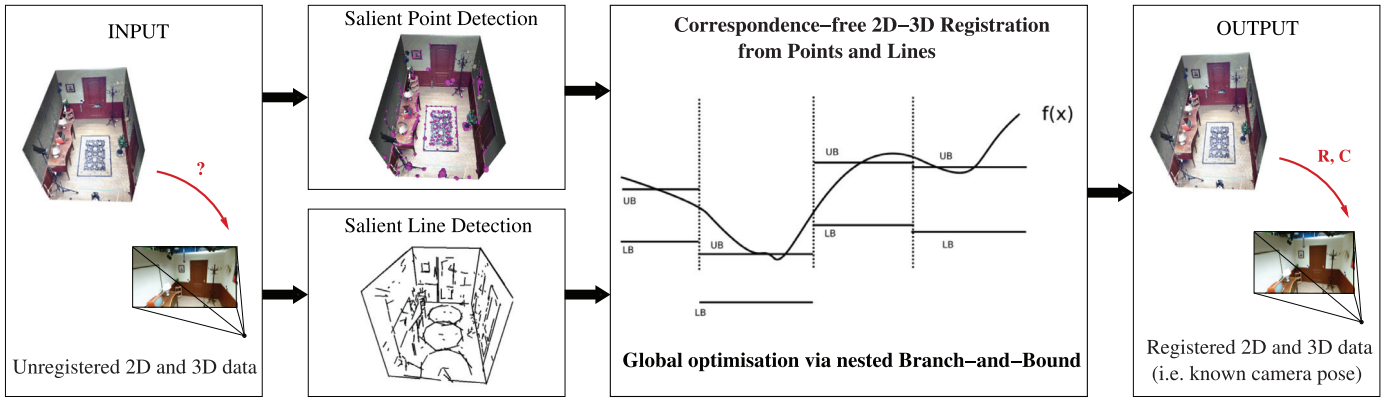

Fig. 1. Diagram illustrating the pipeline for correspondence-free 2D–3D registration. The proposed nested Branch-and-Bound algorithms are the central part of the pipeline, enabling global optimisation from point and line features extracted from 2D and 3D data.

for the problem at hand. However, existing approaches only search for local maxima and hence i) require a good initialisation and ii) are sub-optimal, particularly for higher rates of outliers.

In this paper we propose a globally optimal solution to this problem, achieved via a Branch-and-Bound (BnB) strategy. It recursively searches the transformation space, bounding the objective function at each stage and discarding parts of the transformation space for which it is impossible for the solution to lie in. Eventually, the remaining transformation space is tightly bounded and it may be concluded that transformations in the remaining space must be within ϵ of the globally optimal solution. Furthermore, the approach is not restricted to one feature type, but instead can be applied to the case where points, lines, or a mixture of each are present.

Within the proposed BnB algorithm a nested BnB structure is used (similarly to Yang et al. [16]), whereby an outer BnB searches over the rotation component, with an inner BnB searching for the camera centre at each stage. It is in general faster than searching the full 6D parameter space directly since large parts of the rotation space may be unconditionally discarded, and since evaluating each bound is faster as features are only rotated once for the outer BnB. We extend upon this idea by proposing two extensions to the nested BnB structure in order to speed up the convergence without compromising on the accuracy of the solution.

In the first instance, a deterministic annealing procedure is implemented that gradually increases the accuracy of the search as the algorithm progresses. As such, early regions of rotation space may be more quickly evaluated, and the algorithm can focus its search at the later stages where it is nearing convergence. Secondly, we propose a probabilistic variant, whereby the inner BnB of less promising areas of rotation space is evaluated to a lower accuracy compared to more promising areas of rotation space. Both approaches result in a significant speed-up to the algorithm as demonstrated across a range of experiments on synthetic and real data.

The paper makes the following contributions. Firstly, we propose a globally optimal solution to this problem, achieved via a Branch-and-Bound strategy. Its formulation readily allows for both point and line features to be used, allowing it to be applicable to a broader range of scenes and also exploiting the complementarity of these different types of features to improve registration accuracy and robustness. Secondly, we propose novel formulations that allow for the speed-up of nested BnB algorithms while preserving the optimality properties of the solution. The approach is evaluated against the state of the art where significant improvements are demonstrated: our approach is more accurate and significantly more robust to high rates of outliers compared to existing approaches.

The paper is based on our previous work [17] which it extends in several ways. Firstly, the formulation is generalised to simultaneously allow use of both point and line features for globally optimal registration. This broadens the applicability of the method over our previous approach and improves its performance. Secondly, the methodology is further developed to include deterministic and probabilistic nested BnB formulations, resulting in a significant performance speed up while preserving optimality. Finally, the experimental evaluation is considerably improved through consideration of a broader range of synthetic and real datasets, comparison against an additional RANSAC approach, and use of a more realistic evaluation protocol based on features obtained from recently proposed 2D and 3D salient feature detectors (as opposed to 2D features backprojected to the 3D domain which were used in our previous work).

The structure of this paper is as follows. Related work is discussed in Section 2. Section 3 formally defines the scope of the problem before the proposed Branch-and-Bound approach is detailed in Section 4. Section 5 describes the deterministic and probabilistic nested BnB formulations. The different approaches are then evaluated against the state of the art on synthetic and real dataset in Section 6. Finally, conclusions and avenues for future work are discussed in Section 7.

2. Related work

A traditional approach to the feature registration problem is the hypothesise-and-test RANSAC algorithm [13]. RANSAC relies upon hypothesising small sets of 2D–3D correspondences (of size 3 for the 6 parameter 2D–3D registration problem), determining the transformation parameters from the small set of correspondences, and verifying the transformation against the rest of the features. Assuming there are N 2D features and M 3D features, there are a total of $\binom{NM}{3}$ hypothetical sets of size 3 correspondences to choose from. Assuming there are only kN inlying feature correspondences (where k is the inlier ratio, $k<1$), there are a total of $\binom{kN}{3}$ sets of size 3 of inlying correspondences. As a result, the expected number of correspondences that must be hypothesised before finding an inlier set is $\mathcal{O}((\frac{M}{3})^3)$. However, the hypothesis verification stage requires projecting the 3D features onto an image plane and determining their nearest neighbours from the 2D features. Hence, for 2D–3D feature registration where correspondences are unknown, RANSAC has complexity $\mathcal{O}(\frac{M^4 \log N}{k^3})$.

The above analysis is too simple–in reality, a set of 3 inlying correspondences may not lead to the optimal transformation

due to noise. This was observed by Chum et al. [18] who propose an outer and an inner RANSAC loop, whereby whenever a new best solution is found the inner RANSAC locally searches from the smaller, inlying set of correspondences. It was more formally addressed by Imre and Hilton [19] who minimise the total number of iterations within each stage of such a two-stage RANSAC approach. Alternative extensions have been proposed to improve the speed of RANSAC e.g. WALDSAC [20] that evaluates the potential correspondences of a transformation in an optimal order. However, no RANSAC variant is able to reduce the very high complexity for this particular problem. The high complexity of RANSAC for this problem has led to more recent approaches e.g. [14,15] that search over the transformation space rather than potential correspondences leading to lower complexity of $\mathcal{O}(N^3)$.

Machine learning approaches have recently been applied to the 2D–3D registration problem. PoseNet [21] by Kendall et al. uses a CNN for 2D–3D registration of an outdoor scene, where the scene is obtained by Structure-from-Motion (SfM). Its accuracy is, however, somewhat limited–an issue later addressed by Kendall and Cipolla [22], where the authors specifically focus on applying a geometric loss function to the network, thereby improving the accuracy over their previous work. Conversely, a Random Forest approach has been proposed by Shotton et al. [23], however, this is for the slightly easier task of registering a 3D scene to an RGB-D image. ML approaches may also be applied to specific subcomponents of the 2D–3D registration problem, for example DSAC [24] who replace the deterministic RANSAC hypothesis with a smooth, differentiable objective function. However, RANSAC-based approaches fundamentally scale poorly where correspondences are unknown.

In the next two subsections, we review specific approaches that have been proposed in the cases of point features and line features respectively. The authors are not aware of any approach that explicitly uses points and lines within the same framework, therefore these two types of approaches are discussed separately. The section is concluded with a survey of branch-and-bound approaches that have been proposed to solve related geometry estimation problems.

2.1. Point-based methods

One of the best, early approaches to 2D–3D registration using points where correspondences are unknown is the *SoftPosit* algorithm [14]. It locally searches the transformation space while simultaneously determining the correspondences between 2D and 3D points. At each iteration, multiple, weighted correspondences are hypothesised based on the pose and points' nearest neighbours under the pose; and subsequently the pose is determined from the multiple, weighted correspondences. An annealing parameter is used within the weighting that ensures the algorithm converges towards hypothesising one-to-one correspondences as it progresses.

Moreno–Noguer et al. [15] have proposed a solution known as *BlindPnP*, by modelling an initial set of poses by a Gaussian Mixture Model and using each component to initialise a Kalman filter. Potential 2D and 3D points are considered in turn by the model to update the mean and covariance; eventually the algorithm determines a solution with high confidence. It performs comparably to *SoftPosit* in a similar amount of time except in large amounts of clutter, where *SoftPosit* is outperformed by *BlindPnP*.

An interesting solution has been proposed by Enqvist et al. [25] who compute pairwise constraints between pairs of potential correspondences. By creating a graph of all possible pairs of correspondences, the optimal solution is found by determining the largest set of pairwise consistent correspondences, formulated as a vertex cover problem. However, results were only given when correspondences were hypothesised and the problem was inlier set maximisation; it is unclear how it would perform if no correspondences could be known between the 2D and 3D points.

Other proposed solutions are a lot more restrictive, e.g. both [26,27] solve the problem where no outliers are present. Zhou and Zhang [26] use this to obtain global information e.g. that the mean of the 3D points should project onto the mean of the 2D points, and Marques et al. [27] view the problem as a correspondence permutation problem, which they solve by a convex relaxation procedure. The assumption however is unreasonable in many scenarios, where an algorithm that is robust to high outlier rates is required.

2.2. Line-based methods

An early solution to 2D–3D registration from correspondence-free lines is proposed by Beveridge and Riseman [28] who use a local search procedure to iteratively arrive at local optima. They investigate how easy the problem is; evaluating expected run-time as a function of the number of lines and amount of clutter. Bhat and Heikkilä [29] systematically sample and rank the space of potential poses however it is computationally inefficient for large numbers of lines. Alternatively, the *SoftPosit* algorithm has been extended to use lines [30]. At each iteration, the algorithm finds the nearest point of each 2D line for the endpoint of each 3D line. This point assignment enables it to adapt to the original *SoftPosit* algorithm for points.

Some approaches to registration using lines can make restrictive assumptions. It is not uncommon to assume a Manhattan World where all lines are orthogonal, which may be used to speed up the algorithm e.g. by restricting the search space [31]. Alternatively, detected lines may be viewed as edges on a graph, leading to a graph matching approach [32]. However the graph structure is typically not preserved under a projective transformation, and the approach is more suited to other tasks e.g. aerial image registration.

All existing approaches to 2D–3D correspondence-free registration are heuristic, with no guarantee of optimality. In contrast, here we present a globally optimal solution to the problem, achieved via a branch-and-bound approach. By solving the problem in a globally optimal manner, our approach is demonstrably more robust to high rates of outliers compared to the state of the art. Furthermore, the approach naturally allows for both points and lines to be used within the same framework, in contrast to the approaches reviewed above.

2.3. Branch-and-bound methods

Branch-and-bound solutions to geometry estimation in computer vision have been proposed for a number of different problems, typically requiring novel derivations of bounds in each case.

Many BnB approaches in registration rely on *linear programming* (LP) techniques to compute bounds, e.g. [33], whereby bounds may be computed as solutions to a LP. In a naive form they may only be applied to linear transformations, so to be more widely applicable nonlinear constraints are relaxed into linear convex and concave envelopes to compute upper and lower bounds respectively (e.g. [33,34]). The optima of each envelope are determined as the bounds for the region of space: as the size of the region decreases the difference between the optima decreases and so the algorithm converges. LP relaxation techniques have been developed for complex and highly non-linear problems e.g. [35], where it is used for inlier set maximisation where correspondences are unknown. With respect to the 2D–3D registration problem, Jurie [36] approximates perspective pose by orthographic pose (a linear transformation) to create a problem that may be solved by similar techniques without the need for convex or concave envelopes. However, its use of the

Gaussian error model results in an approach that is not robust to outliers.

Alternative BnB approaches compute bounds that are *geometrically meaningful*. The earliest approaches are due to Breuel [37] who focuses mainly on 2D–2D registration problems with up to 4 degrees of freedom. He derives geometrically meaningful bounds that describe the maximum distance a feature can move by under a bounded set of transformations. He also proposes the use of matchlists: potential correspondences are kept when searching new parts of the transformation space so as to speed up nearest neighbour searches. The geometrically meaningful approach to computing bounds has been used for more complex problems, e.g. two-view translation estimation [38] and relative orientation estimation [39]. Geometric bounds have been non-trivially derived for the group of 3D rotations by Hartley and Kahl [40] by considering rotations in their minimal axis-angle representation. This has allowed for globally optimal relative pose estimation [40], and 3D–3D registration [16]. In the latter case an outer BnB algorithm searches over the rotation space while an inner BnB searches for the translation.

Recent BnB approaches have focused on creating efficient search mechanisms. For example, Parra Bustos et al. [41] propose an efficient bounding mechanism for 3D rotations, based on the insight that a rotation leaves the magnitude of a point unchanged. Alternatively, a novel, efficient approach was proposed by Chin et al. [42]. Unlike the majority of other approaches that search over the transformation space, this explicitly searches over potential correspondences. Initially it hypothesises all correspondences, then runs a tree search to determine which correspondences are invalid. An A* algorithm is used to significantly speed up the search. While very good run-times are reported, it has not been tested for large numbers of outliers–this may be significantly more challenging, since the search tree becomes exponentially larger with the number of outliers. In [43], Paudel et al. use a sum-of-squares optimisation framework to determine whether a point is an inlier for point-to-plane registration and show how plane visibility conditions can be used to boost registration.

Very recently, in [44] Campbell et al. introduced an approach for optimal 2D–3D alignment from point features. Unlike our approach which minimises a continuous objective function measuring the misalignment between 2D and 3D features, Campbell et al. propose an inlier maximisation framework which solves for the camera pose maximising the cardinality of the set of 2D features that are within a set inlier threshold from a projected 3D feature. Their approach also follows a Branch-and-Bound formulation, introducing new bounds which are proved to be tighter than those used in our formulation. Similarly to our approach, theirs guarantees global optimality, albeit for a different metric to that considered in this paper. [44] presents the advantage of not requiring an estimate of the proportion of inliers as it does not require trimming. However, it relies upon a user-defined threshold, which controls whether or not a match is classified as an inlier.

Our approach, originally introduced in [17] and extended here, is the first globally optimal approach to 2D–3D registration using points and lines without correspondences. We use a similar search mechanism to the globally optimal 3D–3D registration algorithm *Go-ICP* [16], whereby an outer BnB searches over the rotation space, and an inner BnB searches over the camera centre. In contrast, our problem firstly requires the derivation of new bounds for the 2D–3D problem. Unlike our original formulation which considered either points or lines, the formulation is extended to simultaneously consider both types of features in the same optimisation framework. Secondly, we propose novel deterministic and probabilistic implementations that allow for the speed-up of nested branch-and-bound algorithms. Thirdly we propose a more general solution, extending the framework to use points and lines, allowing for broader scene applicability.

3. Problem formulation

Initially we give the problem definition for 2D and 3D *features* in general, before moving onto the specifics for points and/or lines. Let there be N 2D features $\{\Lambda_i\}_{i=1}^{N}$ and M 3D features $\{\Psi_j\}_{j=1}^{M}$, and denote the distance between a 3D and 2D feature as $d(\Psi_j, \Lambda_i)$. The objective is to determine the pose of the camera that optimally aligns the sets of features. The pose is an element of 3D motion space $\mathrm{SE}(3) = SO(3) \times \mathbb{R}^3$, composed of a 3D rotation and 3D translation. Hence, where no outliers are present, the objective is to find the rotation $\mathrm{R} \in SO(3)$ and camera centre $\mathbf{C} \in \mathbb{R}^3$ that minimise:

$$\sum_{i=1}^{N} \min_{j \in \{1 \ldots M\}} d(\mathrm{R}(\Psi_j - \mathbf{C}), \Lambda_i). \tag{1}$$

To make (1) robust to outliers, we use *trimming*: instead of minimising the sum over all 2D features it is minimised over the smallest k values, where k represents the expected number of inliers. Without loss of generality, assume the terms of the sum in (1) have been re-ordered in ascending order, yielding the *trimmed objective*: finding $\mathrm{R} \in SO(3)$ and $\mathbf{C} \in \mathbb{R}^3$ that minimise:

$$\sum_{i=1}^{k\ *} \min_{j \in \{1 \ldots M\}} d(\mathrm{R}(\Psi_j - \mathbf{C}), \Lambda_i), \tag{2}$$

where * denotes the sum rearranged in ascending order (note this depends upon R and $\mathbf{C}$). To apply (2) for points (denoted $\Lambda_i^{(P)}$ and $\Psi_j^{(P)}$) or lines (denoted $\Lambda_i^{(L)}$ and $\Psi_j^{(L)}$) simply requires the distance measure to be defined.

In the case of points, denote each 2D point by X_i and each 3D point by Y_j. It is initially tempting to use the Euclidean reprojection error as the most principled distance measure. However, such a distance measure may still not be perfect where there are potential errors in the location of both the 2D and 3D features, and it makes bound computation difficult (and hence more time consuming) due to how it changes non-linearly with respect to the pose of the camera. Instead, we use a more geometrically meaningful distance measure. For convenience, assume the 2D point has been reprojected onto the unit sphere i.e. $X_i \in \mathbb{R}^3$, $\|X_i\| = 1$ where $\|.\|$ denotes the ℓ_2 norm. Then we define the distance between a 2D point and 3D point as:

$$d(\Psi_j^{(P)}, \Lambda_i^{(P)}) = \angle(Y_j, X_i) = \arccos\left(\frac{Y_j \cdot X_i}{\|Y_j\|}\right). \tag{3}$$

In the case of lines, a suitable distance measure is less obvious. Approaches to pose estimation from line correspondences (e.g. [45]) often decouple the problem into the determination of the rotation by using the direction of the 3D line, then determine the camera centre by using an arbitrary point on a line. Inspired by this approach, our line distance measure is a weighted sum of two terms, where the first term is dependent solely on the rotation of the 3D line, and the second term is the distance of a point on the 2D line to the 3D line. With such a construction, the distance will be quite large when the rotation is incorrect regardless of the camera centre–this is of use within the subsequent nested BnB approach, where it can potentially allow for unpromising areas of rotation space to be discarded more quickly.

Our line distance measure is as follows: for each 3D line, denote its normalised direction vector as $\mathbf{d}_j$. For each 2D line, denote its midpoint as P_i, and backproject the line, denoting the normal to this plane as $\mathbf{n}_i$ (see Fig. 2 for an illustration of these terms). In the ideal, noiseless case, $\mathbf{d}_j$ will lie on the backprojected plane and

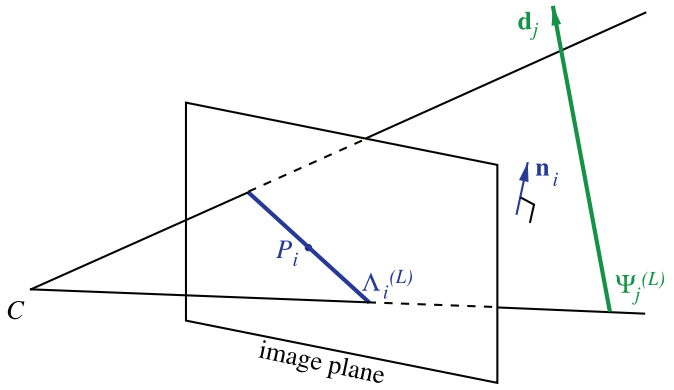

Fig. 2. An illustration of the terminology used in defining a distance measure for lines. $\Lambda_i^{(L)}$ denotes a 2D line, P_i its midpoint and $\mathbf{n}_i$ the normal to its backprojected plane. $\Psi_j^{(L)}$ denotes a 3D line and $\mathbf{d}_j$ its normalised direction vector.

P_i will lie on the projection of line $\Psi_j^{(L)}$. Hence, a suitable distance between the lines is defined as:

$$d(\Psi_j^{(L)}, \Lambda_i^{(L)}) = \lambda \left| \frac{\pi}{2} - \angle(\mathbf{d}_j, \mathbf{n}_i) \right| + \angle(\Psi_j^{(L)}, P_i), \tag{4}$$

where λ defines the relative weighting between the two terms. $\angle(\Psi_j^{(L)}, P_i)$ denotes the angle between P_i and the nearest point of the projected (finite) *line segment* $\Psi_j^{(L)}$; this point on $\Psi_j^{(L)}$ is either between the endpoints of $\Psi_j^{(L)}$ or is one of its endpoints, whichever is closest. This is low for lines that overlap slightly with endpoints that are not well aligned (to account for occlusion), but is higher when the lines are significantly further away. By using this we are implicitly considering 2D lines as infinitely long but 3D lines as finitely long. This assumption has been made elsewhere e.g. [46] due to the poor reliability of determining the endpoints of a 2D line.

In the case where both points and lines are present, we compute a weighted sum of the two objective functions. Assuming there are M_1 3D points and M_2 3D lines, the objective function becomes:

$$\mu \sum_{i=1}^{k_1{}^*} \min_{j \in \{1 \ldots M_1\}} d(\mathrm{R}(\Psi_j^{(P)} - \mathbf{C}), \Lambda_i^{(P)}) + \sum_{i=1}^{k_2{}^*} \min_{j \in \{1 \ldots M_2\}} d(\mathrm{R}(\Psi_j^{(L)} - \mathbf{C}), \Lambda_i^{(L)}), \tag{5}$$

where k_1 and k_2 represent the expected numbers of inlying points and lines respectively. For the relative weighting term we take $\mu = 2$. This is on the principle that the line distance (4) is composed of two equally weighted terms (after setting λ correctly). The second of these is an angular distance which is comparable to the point distance (3); hence, the line distance should be approximately twice that of the point distance.

4. Branch-and-bound

Branch-and-Bound (BnB) is a very general framework for global optimisation. Assume the objective is to minimise some function f over an N-dimensional bounded space $\Omega \subset \mathbb{R}^N$. Assume further that for any subset $\omega \subseteq \Omega$ (hereafter, known as a *branch*) a *lower bound* and an *upper bound* may be determined for the minimal value of f in this branch, and that these bounds converge as the size of the branch tends to zero. For example, the upper bound could simply be the value of the function at the midpoint of the branch, and the lower bound could be the upper bound minus some expression for how much the function can deviate in an interval of that size.

These assumptions allow for the determination of a solution to f whose value is within ϵ of the globally optimal solution, for *any* user-specified $\epsilon > 0$. It relies upon recursively subdividing the space, calculating upper and lower bounds for each branch. Initially the input to the algorithm is simply the branch Ω, and, at any stage in the algorithm, there is a set of branches that are subsets of Ω, each with a lower and upper bound to the minimum value f can take in that branch. At each stage of the algorithm the following two steps are performed:

1. Determine the distance between the *lowest* lower bound and *lowest* upper bound of the bounds in the set of branches. If this distance is less than ϵ the algorithm terminates, outputting the lowest upper bound and its branch.
2. Otherwise, consider the branch that has the lowest lower bound and subdivide it further, computing upper and lower bounds for each sub-branch.

The algorithm will converge because, eventually, the size of the branches considered will be sufficiently small that the distance between the upper bound and lower bound of a newly divided branch will be less than ϵ. When this occurs, the outputted value is within ϵ of the globally optimal solution because the entirety of Ω has been (recursively) searched and so it is known that any better solution is no more than ϵ better than the one returned. For the 2D–3D registration problem, optimisation takes place over the space $SE(3)$. This space is unbounded, so it is assumed the camera centre is known to lie within a bounded set Ω_C, which is typically a reasonable assumption when Ω_C encapsulates a suitably large space.

This section is structured as follows: in Section 4.1, we give *geometrically meaningful* bounds that describe how much the features can be transformed by within a given neighbourhood and in Section 4.2 how these are used to bound the objective function. Then we describe the nested BnB structure in Section 4.3. Finally, local refinement techniques are detailed in Section 4.4.

4.1. Geometric bounds

Bounds are considered separately for the rotation component and camera centre component. Firstly, the rotation bound is computed. Rotations are considered in the *axis-angle representation*: a rotation is represented by a vector $\mathbf{r} \in \mathbb{R}^3$ whose direction specifies the axis of rotation and whose magnitude specifies the angle (hence, $\|\mathbf{r}\| \leq \pi$). The rotation matrix that $\mathbf{r}$ represents may be computed via Rodrigues' rotation formula:

$$\mathrm{R} = \mathrm{I} + \sin(\|\mathbf{r}\|)[\hat{\mathbf{r}}]_\times + (1 - \cos(\|\mathbf{r}\|))[\hat{\mathbf{r}}]_\times^2, \tag{6}$$

where $\hat{\mathbf{r}} = \mathbf{r}/\|\mathbf{r}\|$. The notation $[\mathbf{v}]_\times$ for vector $\mathbf{v} \in \mathbb{R}^3$ denotes the skew-symmetric matrix representation of $\mathbf{v}$, defined as:

$$[\mathbf{v}]_\times := \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{7}$$

Note that $[\mathbf{v}]_\times \mathbf{x} = \mathbf{v} \times \mathbf{x}$ for any vector $\mathbf{x} \in \mathbf{R}^3$.

Lemma 1: *Let* R_0, R *be rotation matrices and* $\mathbf{r}_0$, $\mathbf{r}$ *their corresponding axis-angle representations. Then, for any point* $\mathbf{X} \in \mathbb{R}^3$:

$$\|\mathbf{r}_0 - \mathbf{r}\|_\infty \leq \delta_R \Rightarrow \angle(\mathrm{R}_0\mathbf{X}, \mathrm{R}\mathbf{X}) \leq \epsilon_R, \quad \textit{where} \quad \epsilon_R = \sqrt{3}\delta_R. \tag{8}$$

Proof. [40] has already established that $\angle(\mathrm{R}_0\mathbf{X}, \mathrm{R}\mathbf{X}) \leq \|\mathbf{r}_0 - \mathbf{r}\|$. Noting that

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{3} v_i^2} \leq \sqrt{3 \max_{i=1}^{k} v_i^2} = \sqrt{3} \max_{i=1}^{3} |v_i| = \sqrt{3}\|\mathbf{v}\|_\infty, \tag{9}$$

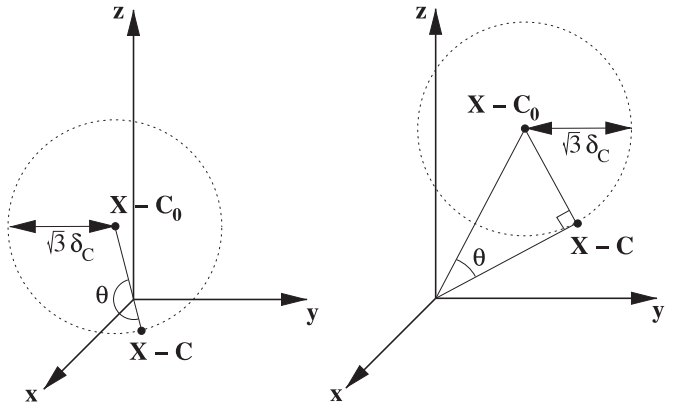

Fig. 3. Left: When $\sqrt{3}\delta_C \geq \|\mathbf{X}-\mathbf{C}_0\|$, the maximum angle is π by placing $\mathbf{X}-\mathbf{C}$ behind (or on) the origin. Right: Otherwise, the maximum angle obtained is when $\mathbf{X}-\mathbf{C}$ is at a right angle to $\mathbf{C}-\mathbf{C}_0$.

it follows that $\angle(\mathsf{R}_0\mathbf{X}, \mathsf{R}\mathbf{X}) \leq \sqrt{3}\|\mathbf{r}_0-\mathbf{r}\|_\infty$, which concludes the proof. □

In the context of BnB, if one considers a branch as a cube of rotations $\mathbf{r}$ in their axis-angle representation where the centre of the branch is $\mathbf{r}_0$ and the cube has half side-length δ_R, then we have $\|\mathbf{r}_0-\mathbf{r}\|_\infty \leq \delta_R$. By the above result, it follows that for any rotation (R) within the cube and for any point $\mathbf{X}$, $\angle(\mathsf{R}_0\mathbf{X}, \mathsf{R}\mathbf{X}) \leq \epsilon_R$.

Next, bounds on the camera centre are derived.

Lemma 2: *Let* $\mathbf{C}_0, \mathbf{C} \in \mathbb{R}^3$. *For any point* $\mathbf{X} \in \mathbb{R}^3$, *let* $\theta = \angle(\mathbf{X}-\mathbf{C}_0, \mathbf{X}-\mathbf{C})$. *Then:*

$$\|\mathbf{C}_0-\mathbf{C}\|_\infty \leq \delta_C \Rightarrow \theta \leq \epsilon_C^{\mathbf{X}-\mathbf{C}_0}, \tag{10}$$

where

$$\epsilon_C^{\mathbf{X}-\mathbf{C}_0} = \begin{cases} \pi & \text{if} \quad \sqrt{3}\delta_C \geq \| \mathbf{X}-\mathbf{C}_0 \|, \\ \arcsin\left(\dfrac{\sqrt{3}\delta_C}{\| \mathbf{X}-\mathbf{C}_0 \|}\right) & \text{otherwise.} \end{cases} \tag{11}$$

Proof. Lemma 2 can be intuitively understood by referring to Fig. 3. The bound is trivially satisfied in the case where $\sqrt{3}\delta_C \geq \|\mathbf{X}-\mathbf{C}_0\|$ since π is the largest possible value allowed under our axis-angle representation parametrisation. The rest of the proof therefore assumes that $\sqrt{3}\delta_C < \|\mathbf{X}-\mathbf{C}_0\|$. The proof is conducted by searching for the camera centre $\mathbf{C}$ that maximises the angle θ and verifying that the corresponding angle is no greater than the bound defined in (11).

Consider the triangle with sides of length $\|\mathbf{X}-\mathbf{C}_0\|$, $\|\mathbf{X}-\mathbf{C}\|$, and $\|\mathbf{C}-\mathbf{C}_0\|$ (e.g. the triangle in the right diagram in Figure 3). By the cosine rule one obtains

$$\|\mathbf{X}-\mathbf{C}\|^2 < 2\|\mathbf{X}-\mathbf{C}_0\|\,\|\mathbf{X}-\mathbf{C}\|\cos\theta, \tag{12}$$

hence $\cos\theta \geq 0$, i.e. $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Since $\sin\theta$ is a strictly increasing function in this interval, obtaining an upper bound on $\sin\theta$ will yield an upper bound on θ. By the sine rule:

$$\sin\theta = \frac{\|\mathbf{C}_0-\mathbf{C}\|}{\|\mathbf{X}-\mathbf{C}_0\|}\sin(\angle(\mathbf{C}_0-\mathbf{C}, \mathbf{X}-\mathbf{C})). \tag{13}$$

Without loss of generality $\mathbf{X}$ and $\mathbf{C}_0$ may be assumed to be constant (since we are searching for $\mathbf{C}$ maximising the angle), hence the expression is maximised when $\angle(\mathbf{C}_0-\mathbf{C}, \mathbf{X}-\mathbf{C}) = \frac{\pi}{2}$. Consequently

$$\sin\theta \leq \frac{\|\mathbf{C}_0-\mathbf{C}\|}{\|\mathbf{X}-\mathbf{C}_0\|} \leq \frac{\sqrt{3}\|\mathbf{C}_0-\mathbf{C}\|_\infty}{\|\mathbf{X}-\mathbf{C}_0\|} \leq \frac{\sqrt{3}\delta_C}{\|\mathbf{X}-\mathbf{C}_0\|} \tag{14}$$

and the result follows. □

4.1.1. A uniformly continuous bound

The function governing the bounds on the camera centre (11) is not uniformly continuous: the relationship between $\epsilon_C^{\mathbf{X}-\mathbf{C}_0}$ and δ_C is dependent on $\mathbf{X}$. This causes real difficulties for the algorithm: if precision ϵ_C is desired and a point $\mathbf{X}$ is arbitrarily close to $\mathbf{C}_0$, an arbitrarily small branch (δ_C) is required. Hence the algorithm will not converge in finite time.

To alleviate this we modify the objective function slightly so as to be uniformly continuous: when computing (2) we *only* take into account 3D features whose distance from the camera centre is larger than a specified threshold (γ). For a suitably small threshold this is sensible in practice: in general very few features will be located immediately in front of the camera.

Note that an alternative way of addressing this issue is to restrict the search space to prevent camera centres from being located within a very small distance from an existing 3D point as proposed by Campbell et al. in [44].

By enforcing $\|\mathbf{X}-\mathbf{C}_0\| \geq \gamma$, we ensure that:

$$\arcsin\left(\frac{\sqrt{3}\delta_C}{\|\mathbf{X}-\mathbf{C}_0\|}\right) \leq \arcsin\left(\frac{\sqrt{3}\delta_C}{\gamma}\right). \tag{15}$$

This now defines a uniformly continuous function since the relationship between δ_C and ϵ_C is independent of $\mathbf{X}$. More explicitly, if a precision of $\epsilon_C \in]0, \pi/2]$ is desired, one may set $\delta_C = \frac{\gamma}{\sqrt{3}}\sin\epsilon_C$ to guarantee a minimum branch size, hence guaranteeing the convergence of the algorithm.

By combining the above lemmas, the following result is obtained:

Theorem 1. *Let* R_0, R *be rotation matrices and* $\mathbf{r}_0$, $\mathbf{r}$ *their corresponding axis-angle representations. Further, let* $\mathbf{C}_0, \mathbf{C} \in \mathbb{R}^3$. *Then, for any point* $\mathbf{X} \in \mathbb{R}^3$:

$$\begin{aligned} &\|\mathbf{r}_0-\mathbf{r}\|_\infty \leq \delta_R \wedge \|\mathbf{C}_0-\mathbf{C}\|_\infty \leq \delta_C \\ &\quad \Rightarrow \angle(\mathsf{R}_0(\mathbf{X}-\mathbf{C}_0), \mathsf{R}(\mathbf{X}-\mathbf{C})) \leq \epsilon_R + \epsilon_C, \end{aligned} \tag{16}$$

where $\epsilon_R = \sqrt{3}\delta_R$ *and* $\epsilon_C = \arcsin\left(\frac{\sqrt{3}\delta_C}{\gamma}\right)$.

The proof follows by combining Lemmas 1 and 2 with the triangle inequality:

$$\begin{aligned} &\angle(\mathsf{R}_0(\mathbf{X}-\mathbf{C}_0), \mathsf{R}(\mathbf{X}-\mathbf{C})) \leq \angle(\mathsf{R}_0(\mathbf{X}-\mathbf{C}_0), \mathsf{R}(\mathbf{X}-\mathbf{C}_0)) \\ &\quad + \angle(\mathsf{R}(\mathbf{X}-\mathbf{C}_0), \mathsf{R}(\mathbf{X}-\mathbf{C})) \\ &\quad \leq \epsilon_R + \angle(\mathbf{X}-\mathbf{C}_0, \mathbf{X}-\mathbf{C}) \leq \epsilon_R + \epsilon_C. \end{aligned} \tag{17}$$

4.2. Function bounds

In this subsection, the bounds defined in Section 4.1 are related to the objective functions described in Section 3. Assume we are minimising the trimmed objective (2) with the angular distance measure for point features (3). It is required to determine *upper* and *lower* bounds for (2) when the pose space SE(3) is bounded. At each stage in the BnB algorithm, the pose space will be divided up into cubes, where we consider jointly a *rotation cube* centred at $\mathbf{r}_0$ of half side-length δ_R and a *camera centre cube* centred at $\mathbf{C}_0$ of half side-length δ_C.

To compute the *upper bound* for (2) using points (3) the objective function is simply evaluated at $(\mathsf{R}_0, \mathbf{C}_0)$. To compute the *lower bound* the expression is derived by evaluating the function at $(\mathsf{R}_0, \mathbf{C}_0)$ and subtracting the maximum amount by which the function may deviate within that branch. Denote $z(\epsilon) = \epsilon_R + \epsilon_C$ and hence, the lower bound is obtained as:

$$\sum_{i=1}^{k^*} \min_{j\in\{1\ldots M\}} \max\left\{0, \angle(\mathsf{R}_0(Y_j-\mathbf{C}_0), X_i) - z(\epsilon)\right\}. \tag{18}$$

The lower bound for lines (4) is derived in a similar way; the angles for each of the two terms in (4) are bounded in the same

manner (by $\epsilon_R + \epsilon_C$). Hence, the lower bound for (2) using lines (4) is obtained as:

$$\sum_{i=1}^{k} {}^{*} \min_{j \in \{1 \ldots M\}} \left(\max \left\{ 0, \lambda \left| \frac{\pi}{2} - \left(\angle(\mathrm{R}_0 \mathbf{d}_j, \mathbf{n}_i) - z(\epsilon) \right) \right| \right\} + \max \left\{ 0, \angle(\mathrm{R}_0 (\Psi_j^{(L)} - \mathbf{C}_0), P_i) - z(\epsilon) \right\} \right). \quad (19)$$

4.3. Nested branch-and-bound

In a similar manner to [16], we use a nested BnB structure for efficiency: an outer BnB searches over the rotation space $SO(3)$ and, for each rotation branch, the upper and lower bounds are solved by an inner BnB algorithm for the camera centre. In doing so, all features may be rotated at the beginning of an inner BnB, leaving only their translation component ($-\mathrm{R}\mathbf{C}$) to be added at each stage; this is more efficient than directly implementing a full 6D search. We shall now describe the computation of bounds in the inner BnB algorithm.

Firstly, the case for determining the *upper bound* of a *rotation cube* is considered. To do so, the rotation is considered at the centre of the cube ($\mathbf{r}_0$) and the aim is to determine the minimum value of (2) where $\mathbf{r}$ is fixed to $\mathbf{r}_0$ and $\mathbf{C}$ is allowed to vary. The upper bound used in the inner algorithm is simply the value of the function at that point, i.e. computed using (18) with $z(\epsilon) = 0$, with the lower bound computed using $z(\epsilon) = \epsilon_C$. There is an early bail-out condition: if the inner lower bound is greater than the outer upper bound then the inner BnB may terminate. This allows for speed-up of the algorithm if the outer upper bound is small (i.e. the algorithm is faster the closer it is to the optimal solution).

Secondly the *lower bound* of a rotation cube is considered. The same computation is performed as for the upper bound, but takes into account the maximum amount the objective function can deviate within the rotation branch. Hence, the upper bound used in the inner algorithm in this case is computed using (18) with $z(\epsilon) = \epsilon_R$; the lower bound with $z(\epsilon) = \epsilon_R + \epsilon_C$.

At this point we should point out some minor differences between our nested BnB implementation and that of Yang et al. [16]. In [16] the authors compute the inner BnB to the same accuracy as the outer BnB and return the (inner) upper bound as the bound for that rotation branch. However, if the *lower* bound of a rotation branch is being considered, clearly the inner *lower* bound will be desired rather than the inner upper bound. Subsequently, for the outer BnB to be calculated to an accuracy of ϵ the inner BnBs will need to be computed to accuracy ϵ/τ, where $\tau > 2$; this will ensure the difference between the outer upper and lower bounds is less than ϵ, hence guaranteeing the convergence of the algorithm. Detailed descriptions of the algorithms are provided in Algorithms 1 and 2 and a proof of the convergence of the algorithm is provided in a supplementary report [47].

4.4. Local refinement

Similarly to other BnB approaches (e.g. [16]) we locally optimise the solution whenever a promising part of the search space is found. If the output of the local optimisation results in a new best solution (according to (2)), the upper bound is updated with the new solution. In our case, we use two refinement algorithms: one with a large basin of convergence that does not assume correspondences between features are known, and a more precise refinement requiring known correspondences. The first refinement is called whenever a solution is within 50% of the current best solution *and* a local refinement has not been called in a neighbourhood of this point. The second refinement is called whenever a new best solution is found (similarly to [16]) and uses the correspondences given by the trimmed nearest neighbours.

Algorithm 1: Nested BnB algorithm to compute optimal rotation and camera centre.

Input: 2D and 3D feature sets, initial rotationand camera centre cubes Ω_R and Ω_C, desired accuracy ϵ.
Output: Optimal rotation $\mathbf{r}^{\text{res}}$ and camera centre $\mathbf{C}^{\text{res}}$.
Set $U_O = +\infty$, $L_O = 0$.
Insert Ω_R with priority L_O into priority queue Q_R.
while $(U_O - L_O > \epsilon)$ **do**
  Remove rotation cube with lowest lower bound from Q_R and sub-divide into 8 sub-cubes.
  foreach *sub-cube* ω_R **do**
    Compute upper bound U_I and corresponding optimal $\mathbf{C}_I^{\text{res}}$ by calling Algorithm 2with $\mathbf{r}_0$ at centre of ω_R, rotation uncertainty $\epsilon_R = 0$, current best error U_O andinner bound accuracy ϵ_I.
    Compute lower bound L_I by calling Algorithm 2with $\mathbf{r}_0$ at centre of ω_R, rotation uncertainty $\epsilon_R = \sqrt{3}\delta_R$, current best error U_O andinner bound accuracy ϵ_I.
    if $U_I < U_O$ **then**
      Set $U_O = U_I$, $\mathbf{r}^{\text{res}} = \mathbf{r}_0$ and $\mathbf{C}^{\text{res}} = \mathbf{C}_I^{\text{res}}$.
      Run local refinement (see Section 4.4) and update U_O, $\mathbf{r}^{\text{res}}$ and $\mathbf{C}^{\text{res}}$ if better solution found.
    end
    if $L_I \leq U_O$ **then**
      Insert ω_R with priority L_I into Q_R.
    end
  end
  Set L_O to lowest lower bound value in Q_R.
end

Algorithm 2: BnB algorithm to compute optimal camera centre given rotation.

Input: 2D and 3D feature sets, initial camera centre cube Ω_C, rotation $\mathbf{r}_0$, rotation uncertainty ϵ_R, current best error U_O, desired accuracy ϵ_I.
Output: Lower and upper bounds L_I and U_I on error and corresponding optimal camera centre $\mathbf{C}_I^{\text{res}}$.
Set $U_I = U_O$, $L_I = 0$.
Insert Ω_C with priority L_I into priority queue Q_C.
while $(U_I - L_I > \epsilon_I)$ **do**
  Remove cube with lowest lower bound from Q_C and sub-divide into 8 sub-cubes.
  foreach *sub-cube* ω_C **do**
    Compute upper bound $\overline{U}_I$ using (18) with $z(\epsilon) = \epsilon_R$.
    Compute lower bound $\underline{U}_I$ using (18) with $z(\epsilon) = \epsilon_R + \arcsin\left(\frac{\sqrt{3}\delta_C}{\gamma}\right)$.
    if $\overline{U}_I < U_I$ **then**
      Set $U_I = \overline{U}_I$ and $\mathbf{C}_I^{\text{res}} = \mathbf{C}_0$ (centre of ω_C).
    end
    if $\underline{U}_I \leq U_I$ **then**
      Insert ω_C with priority $\underline{U}_I$ into Q_C.
    end
  end
  Set L_I to lowest lower bound value in Q_C.
end

For the first local refinement algorithm with a large basin of convergence we use *SoftPosit* in the case of either points, lines, or both [14,30]. In the case where both points and lines are used, we modify the existing *SoftPosit* algorithm; specifically, the assignment matrix is adjusted to account for both points and lines such that it is impossible to assign any weighting to a point-line

correspondence. For the second algorithm we use *EPnP* [48] for points and the approach by Kumar and Hanson [49] for lines. For both points and lines we use the approach by Dornaika and Garcia [50] that is based on the Posit algorithm [51].

It should be noted that none of these algorithms directly minimise the objective function defined in (2) and if local refinement does not result in a better function value the algorithm will not update its best solution. Furthermore, it is not necessary to perform local refinement since the approach will still eventually find the optimal solution without it. Despite this, these refinement techniques allow the BnB algorithm to more efficiently find and discard local optima and concentrate on finding the global optimum.

5. Deterministic and probabilistic nested branch-and-bound methods

In Section 4.3 a nested BnB was proposed, where the outer BnB is computed to an accuracy of ϵ by computing the inner BnBs to an accuracy of $\epsilon_I = \epsilon/\tau$ with $\tau > 2$. However, it is not necessary to always compute the accuracy of an inner BnB to ϵ/τ and there is a trade-off here: calculating the inner BnBs to a high degree of accuracy (low ϵ_I) will result in tighter upper and lower bounds meaning the outer BnB will converge in fewer iterations, however each inner BnB will take more iterations.

In this section, we shall present two variants of the algorithm that take advantage of the above insight by computing the inner BnBs to different degrees of accuracy (ϵ_I). Under an appropriate choice of ϵ_I, both variants retain the global optimality of the approach by ensuring the outer BnB converges to within ϵ. A detailed analysis proving the convergence of both variants is provided in our supplementary material [47].

For proposed variants of the algorithm, the accuracy of the inner BnBs (ϵ_I) is a function of ϵ, the current outer upper and lower bounds (U_O and L_O), and the current inner upper and lower bounds (U_I and L_I) at that stage of the algorithm. For the first proposed variant the accuracy is computed in a deterministic way; ϵ_I is large at the beginning of the algorithm and gradually decreases as it progresses. For the second variant, ϵ_I is computed probabilistically whereby branches that look promising are evaluated to a higher degree of accuracy (lower ϵ_I) than those that do not.

5.1. Deterministic BnB

The deterministic BnB that we propose initially computes inner BnBs to a large ϵ_I, and gradually decreases it to ϵ/τ with $\tau > 2$ as the algorithm progresses. Hence, it terminates to the same accuracy as the original algorithm, despite computing many previous branches to a worse accuracy.

At any stage in the algorithm the outer upper bound (U_O) and outer lower bound (L_O) are known. Then we deterministically take $\epsilon_I = \frac{U_O - L_O}{\tau}$ as the accuracy to use for the inner BnB. This is for two reasons: firstly, it guarantees the difference between U_O and L_O to decrease as better parts of the search space are explored, i.e. the algorithm will continue to converge. Secondly, it naturally leads to a final accuracy of ϵ/τ, guaranteeing the same accuracy as the original algorithm. To begin with, ϵ_I is set to an arbitrarily large number, hence U_O is quickly set to a reasonable value after the first inner BnB.

5.2. Probabilistic BnB

We furthermore propose a probabilistic BnB formulation that, informally, calculates an inner BnB to a high degree of accuracy if it looks promising (e.g. it will lead to a new best solution) and a low degree of accuracy otherwise. More formally, we shall determine the trade-off between the amount of time taken evaluating an inner BnB and the expected benefit of taking that amount of time.

We shall assume for simplicity that there are two outcomes of interest for evaluating an inner BnB. When using the inner BnB to determine an upper bound the outcome of interest is whether or not it leads to a new global upper bound–if it does, this will speed up the algorithm (since there is an early bail-out condition, see Section 4.3) or the algorithm may potentially converge (i.e. terminate). When using the inner BnB to determine a lower bound the outcome of interest is whether the lower bound is high enough such that the branch may be discarded as this will further narrow the search space.

The probabilities for the outcomes of interest vary depending on the accuracy ϵ_I that is desired. Denote the probability that the outcome of interest occurs as $p(\epsilon_I)$ and the time taken to evaluate the inner BnB to an accuracy of ϵ_I as $t(\epsilon_I)$. If the outcome of interest occurs, assume the rest of the BnB algorithm takes time t_1 and time t_2 otherwise (hence $t_1 < t_2$ is assumed). Hence, the expected amount of time taken is:

$$T = (t(\epsilon_I) + t_1)p(\epsilon_I) + (t(\epsilon_I) + t_2)(1 - p(\epsilon_I)). \tag{20}$$

To determine ϵ_I that minimises the expected amount of time taken we set the derivative of (20) to zero to give:

$$(t_1 - t_2)p'(\epsilon_I) + t'(\epsilon_I) = 0. \tag{21}$$

We shall assume that $t(\epsilon_I) \propto 1/\epsilon_I$ because all bounds derived are first order bounds that scale linearly with respect to the branch size. Hence, (21) may be re-written to give $\frac{1}{\epsilon_I^2} \propto p'(\epsilon_I)$. Integrating both sides gives the relationship

$$\epsilon_I = \frac{a}{p(\epsilon_I) + b} \tag{22}$$

for constants a and b. To guarantee the convergence of the algorithm, we constrain the maximum value of ϵ_I to be $\frac{U_O - L_O}{\tau}$ with $\tau > 2$. Furthermore, we set a minimum value of ϵ_I as ϵ/τ for all inner BnBs so that the algorithm does not spend too much time in an inner BnB.

These conditions may be substituted into (22) such that, when $p(\epsilon_I) = 1$, ϵ_I takes its minimum value of ϵ/τ ; and similarly when $p(\epsilon_I) = 0$, ϵ_I takes its maximum value of $\frac{U_O - L_O}{\tau}$. These allow for the constants a and b to be determined, yielding the relationship:

$$\epsilon_I = \frac{(U_O - L_O)\epsilon}{\tau((U_O - L_O - \epsilon)p(\epsilon_I) + \epsilon)}. \tag{23}$$

Computation of $p(\epsilon_I)$: If the inner BnB is being used to determine the outer upper bound (UB), we are interested in the probability that the inner BnB will find a new outer UB, i.e. $p(\epsilon_I)$ is the probability that the inner UB is found to be less than U_O when evaluated to accuracy ϵ_I. Conversely, if the inner BnB is being used to determine the outer lower bound (LB), we are interested in the probability that the inner BnB will lead to this branch being discarded, i.e. $p(\epsilon_I)$ is the probability that the inner LB is found to be greater than $U_O - \epsilon$ when evaluated to accuracy ϵ_I.

In either case, the estimate is determined by firstly considering the optimal value of the objective function in the inner BnB, denoted g. At this stage, all that can be said is that g lies between L_I and U_I. However, we have observed it to have a tendency to lie significantly closer to U_I than L_I, i.e. L_I is a very pessimistic lower bound.

The reason for this is that the lower bound computation in (18) or (19) are computed as the sum of minima, but it is very unlikely that the summands simultaneously obtain their minimum at the same point in space. It is however true that any one of the

summands obtains its minimum value within the branch, reducing the inner UB by an approximate value r where $r = \frac{U_I - L_I}{k}$, since the difference between the inner UB and inner LB is split between the k summands. The other $k-1$ summands are very unlikely to also obtain their minimum value at this point in the branch, and we assume each summand to reduce the inner UB by a uniformly distributed amount from the interval $[-r, r]$ at this point in the branch.

As a result, we assume the expected value of g to be $U_I - r$, and its variance is that of sum of $k-1$ uniformly distributed variables from the interval $[-r, r]$. Using the central limit theorem, we approximate the distribution of g as:

$$g \sim \mathcal{N}\left(U_I - r, \frac{k-1}{3}r^2\right). \tag{24}$$

To use the distribution of g to estimate $p(\epsilon_I)$ where the inner BnB is being used to determine the outer UB, we use the approximation:

$$p(\epsilon_I) = P\left(g < U_O - \frac{\epsilon_I}{k}\right). \tag{25}$$

(25) may be computed using the error function. To determine ϵ_I requires solving (23) and (25) simultaneously–we use an iterative approach to this with initial condition $p(\epsilon_I) = 0.5$. Where the inner BnB is being used to determine the outer LB, we take $p(\epsilon_I) = P(g > U_O - \epsilon - \frac{(k-1)\epsilon_I}{k})$ and proceed in a similar manner.

6. Experimental evaluation

We compare between the three proposed approaches: *BnB, BnB-D* for the deterministic BnB, and *BnB-P* for the probabilistic BnB. They are furthermore compared to existing methods for 2D–3D feature matching without correspondences. Specifically, we compare against the traditional RANSAC [13] algorithm, *SoftPosit* [14], and the state-of-the-art *BlindPnP* [15] approaches.

The structure of this section is as follows. In Section 6.1 we give implementation details for all approaches evaluated in this section, and in Section 6.2 the evaluation measures (accuracy and speed) are described. Subsequently results are presented, in Section 6.3 for synthetic data and in Section 6.4 for real data.

6.1. Implementation details

BnB/BnB-D/BnB-P: Few parameters need to be set for our globally optimal approaches, and we use the same parameters for all experiments with the exception of k (the expected number of inliers). For the synthetic data, k is set to the exact number of inliers (unless otherwise stated); for real data it is fixed to 25% of the total number of 2D features. In (4) we use $\lambda = 0.3$, and, for the uniformly continuous bound, we take $\gamma = 0.1$. We set $\epsilon = 0.0025k$ for where only point features are used, and $\epsilon = 0.006k$ for when line features, or both point and line features, are used (with the exception of in Fig. 4, where ϵ is a free parameter). For all approaches, the parameter τ setting the inner bound accuracy ϵ_I was set to the limit case $\tau = 2$. Experiments confirmed the converge of all the BnB variants under this setting.

RANSAC: The RANSAC algorithm [13] relies upon hypothesising transformations from minimal subsets and determining how many inliers there are with respect to the hypothesised transformation. In our case there is no inlier threshold as trimming is used, therefore the transformation that minimises (2) is taken. Since minimal samples of inlying features typically do not produce optimal transformations in the presence of noise, we use the LO-RANSAC algorithm [18]. Alternative, more efficient variants of RANSAC are inapplicable in our case. For example, PROSAC [52] relies upon the similarity of feature descriptors to obtain a better evaluation order, however we assume no feature descriptors are used. Alternatively, WALDSAC [20] evaluates the potential correspondences of a transformation in an optimal order–this is difficult to apply in our case where a trimmed objective function is used. To determine the transformation from minimal samples we use the approach by Kneip et al. [53] in the case of points, and the approach by Dhome et al. [54] in the case of lines. We do not test against RANSAC in the case where both points and lines are present.

SoftPosit: The *SoftPosit* algorithm has been implemented for points [14] and lines [30]. We extend it to the case where both points and lines are present by adjusting the assignment matrix used, such that it is impossible to assign any weighting to a point-line correspondence. It is run from a number of random starting points in SE(3) covering the same space the proposed BnB algorithms search from.

BlindPnP: The *BlindPnP* algorithm [15] has only been proposed for points. Furthermore, it is observed that *BlindPnP* relies upon the ability to use *pose priors* on where the possible camera pose may be – represented by a Gaussian Mixture Model (GMM) of typically 20 components. In their experiments the pose is constrained such that the camera lies on a torus around the 3D scene. However, it is often unrealistic to assume such prior knowledge can be obtained, and it is difficult to alter their approach to work with a significantly larger number of priors over a greater space of SE(3). Therefore, for a fair comparison, our approach was altered to use these pose priors for some of the synthetic experiments.

6.2. Evaluation measures

Throughout the experiments we aim to measure the accuracy of the available algorithms, and the speed of the approaches, where possible.

Accuracy: The accuracy is defined as the proportion of experiments from which an inlying solution is produced by an algorithm. A solution is deemed an *inlier* if the distance between the ground truth and estimated rotation, and ground truth and estimated camera centre, are both less than a given threshold.

For the rotation, the angle between the ground truth and estimated rotations is required to be less than 0.1 radians to be deemed an inlier. The angle between two rotations R_a and R_b is computed by constructing $\mathrm{R}_c = \mathrm{R}_a^T \mathrm{R}_b$, and computing the angle of rotation of R_c in its axis-angle form [40].

For the camera centre, the relative error between the two camera centres (expressed as $\|\mathbf{C}_{true} - \mathbf{C}\|/\|\mathbf{C}\|$) is required to be less than a threshold of 0.1 to be deemed an inlier, the same as in [15]. However, we note that the relative error between camera centres is coordinate system dependent, therefore we also use the absolute error between the camera centres ($\|\mathbf{C}_{true} - \mathbf{C}\|$). It will be made clear which error on the camera centres is used in each case.

Speed: Timings are obtained by running the algorithms on servers with 2 $\times$ 10 core CPUs running at 2.6 GHz. Note that timings should be interpreted with care as they can only provide a coarse estimate of algorithm performance being influenced by server load at the time of the experiments. To complement this, we also include information on the number of iterations as this provides a more meaningful basis for comparing the different BnB algorithms proposed. Both run-times and numbers of iterations have high variance, hence we report the three quartiles, and give the proportion of experiments that converged within an iteration limit (denoted T_I).

6.3. Synthetic data

In this subsection, we compare against existing approaches for synthetically generated data. However, to fairly compare against

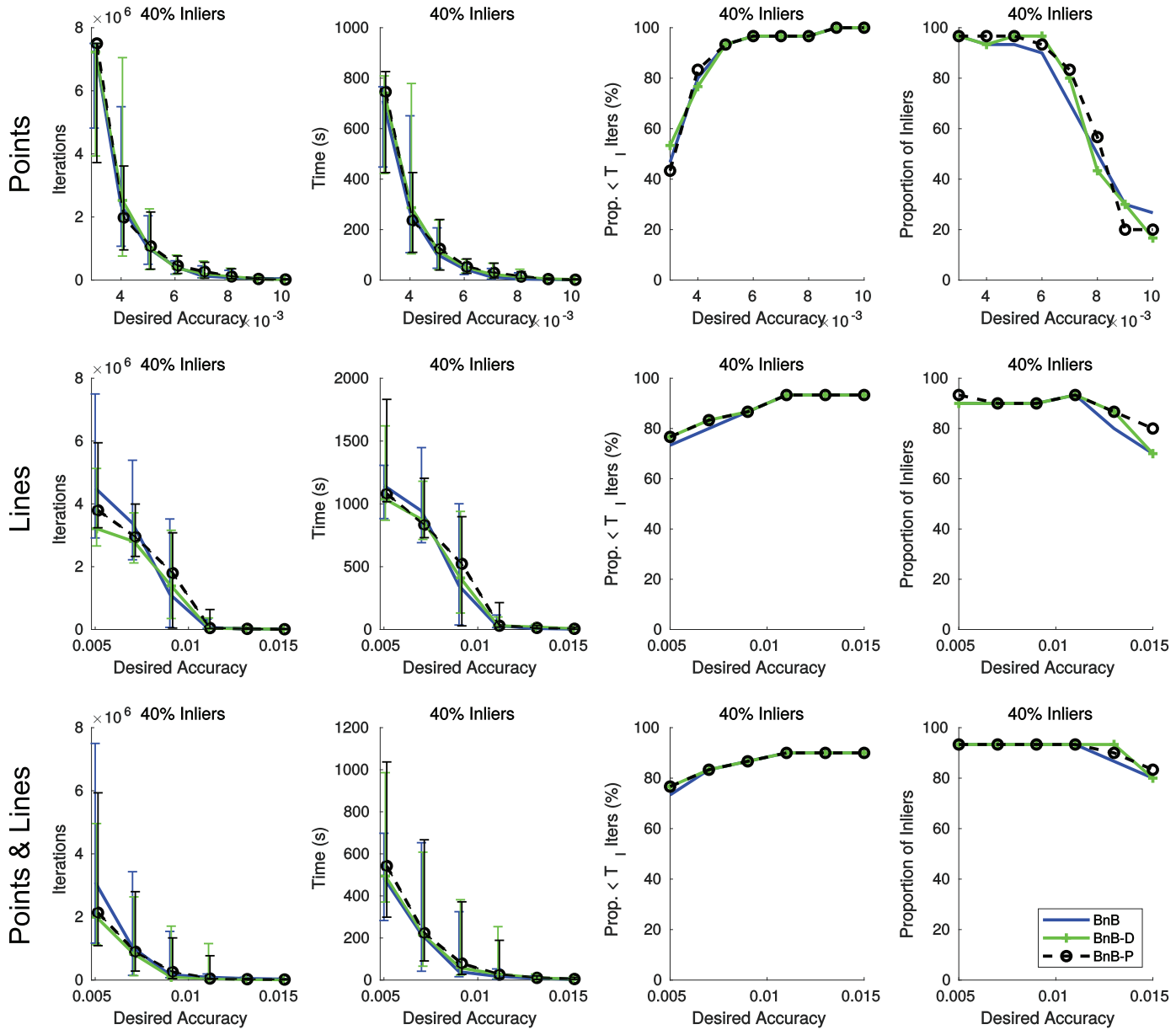

Fig. 4. Median number of iterations (first column) and run-time (second column) with bar showing first and third quartiles, proportion of trials that converged within the iteration limit (third column) and proportion of inlying solutions (fourth column) for *BnB*, *BnB-D*, and *BnB-P* across different levels of desired accuracy, for a feature size of 50. No local refinement is used here. Each experiment was terminated after $T_I = 7.5 \times 10^6$ inner BnB iterations if it had not already converged by then.

different approaches requires certain assumptions be placed on the data for each approach. For example, *BlindPnP* places a prior on the camera pose; assuming it to lie on a torus around and facing the 3D scene, represented by a GMM of 20 components. However, RANSAC searches all potential correspondences regardless of pose priors, hence, to give a fair comparison against RANSAC there should be very little prior placed on the camera pose. Therefore, this section is split into two subsections; the first where pose priors are used, and the second where significantly fewer assumptions are placed on the camera pose.

6.3.1. Pose priors

Throughout this subsection the accuracy and speed of the approaches are tested where pose priors are assumed. For a fair comparison with [15], the pose priors are generated in the same way as in [15]; and the relative error between camera centres will be used in this section to determine whether a solution is an inlier. Our algorithms are modified to use pose priors in the following way: the input to our algorithm is a set of branches corresponding to each pose prior. Hence, each pose prior is defined by an initial rotation branch (centred at the prior) with each branch initiating its own camera centre branch (centred at the prior). Due to the potentially large running times, the proposed approaches are terminated after $T_I = 7.5 \times 10^6$ inner BnB iterations if they had not already converged by then. For a fair comparison, RANSAC is also terminated after T_I iterations. *SoftPosit* and *BlindPnP* are run 20 times; from the centre of each of the GMM components.

We generate the 2D and 3D features in a similar manner to [15]: firstly, we randomly generate a set of 3D features (points or lines) and randomly choose a camera position in SE(3) from the torus. A proportion of these 3D features are deemed *inliers* and are projected onto the image. Noise is added to their position (the endpoints in the case of lines) of variance 2 pixels. A number of outlying 2D features are then randomly generated on the image

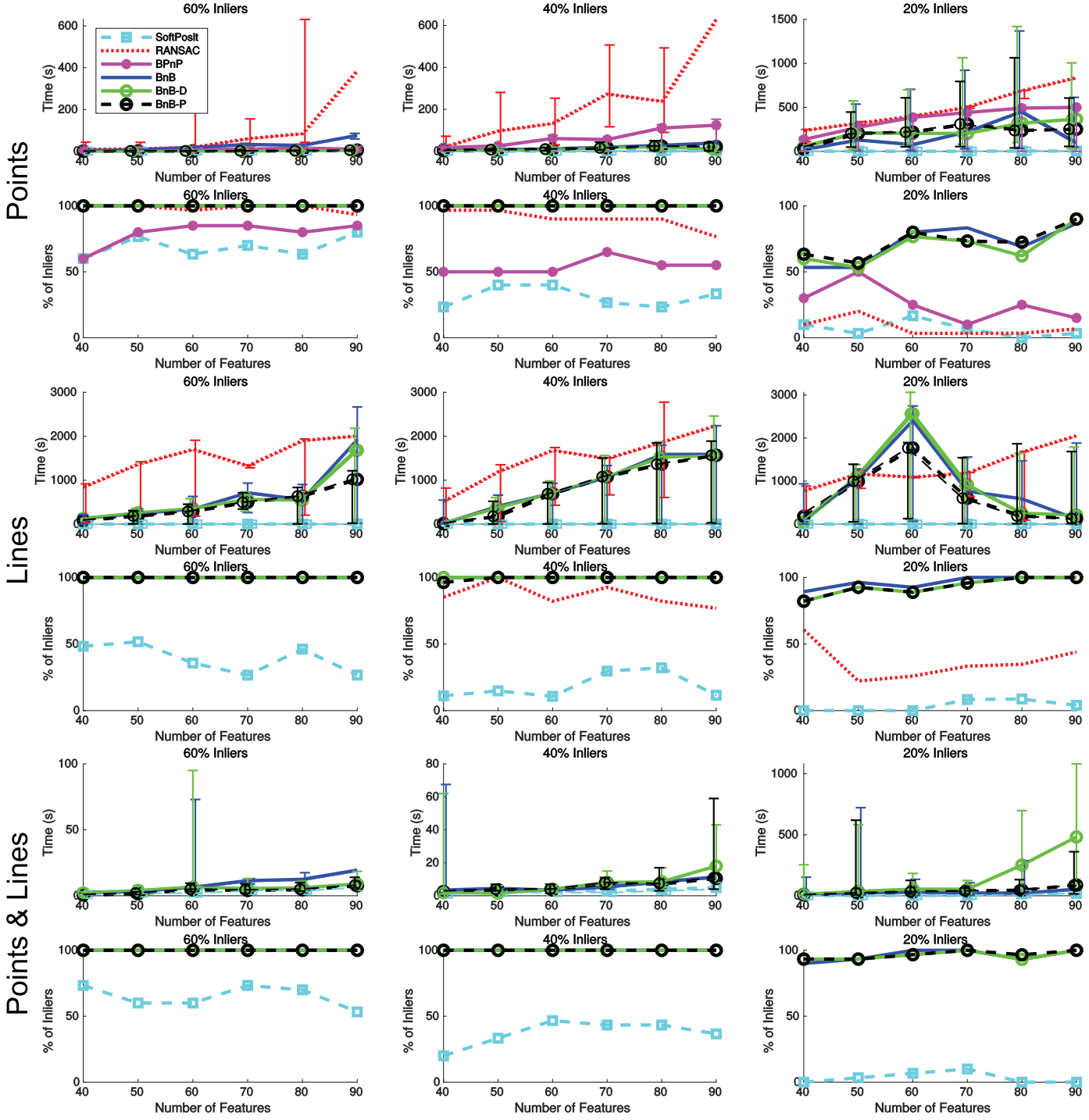

Fig. 5. Proportion of inlying solutions for all algorithms tested. From top to bottom: using points, using lines, using both. From left to right: 60% inliers, 40% inliers, 20% inliers.

such that the number of 2D and 3D features is equal (none of the algorithms *require* the two feature sets to be of equal size–we simply test in this way for simplicity).

In this subsection, three sets of experiments are performed. The first is without local refinement (Section 4.4), and is to test the proposed deterministic and probabilistic BnB algorithms in isolation without being affected by the other aspects of the algorithm. Secondly, experiments are performed with local refinement, across a range of feature quantities and proportion of inliers. Finally, we present results of varying the expected number of inlier features (k) from their ground truth, since this cannot be assumed to be known in practice.

Without Local Refinement: Initially we test the three proposed approaches (*BnB*, *BnB-D*, and *BnB-P*) without local refinement. In this case, we test for a feature size of 50 (either points, lines, or 25 of each) for 40% inliers, for varying levels of accuracy (ϵ). 30 trials were performed in each case. The results are shown in Fig. 4. Due to the high variance of timings obtained, the quartiles of the number of iterations and run-time are shown, along with the proportion of trials to converge within the iteration limit. Based on

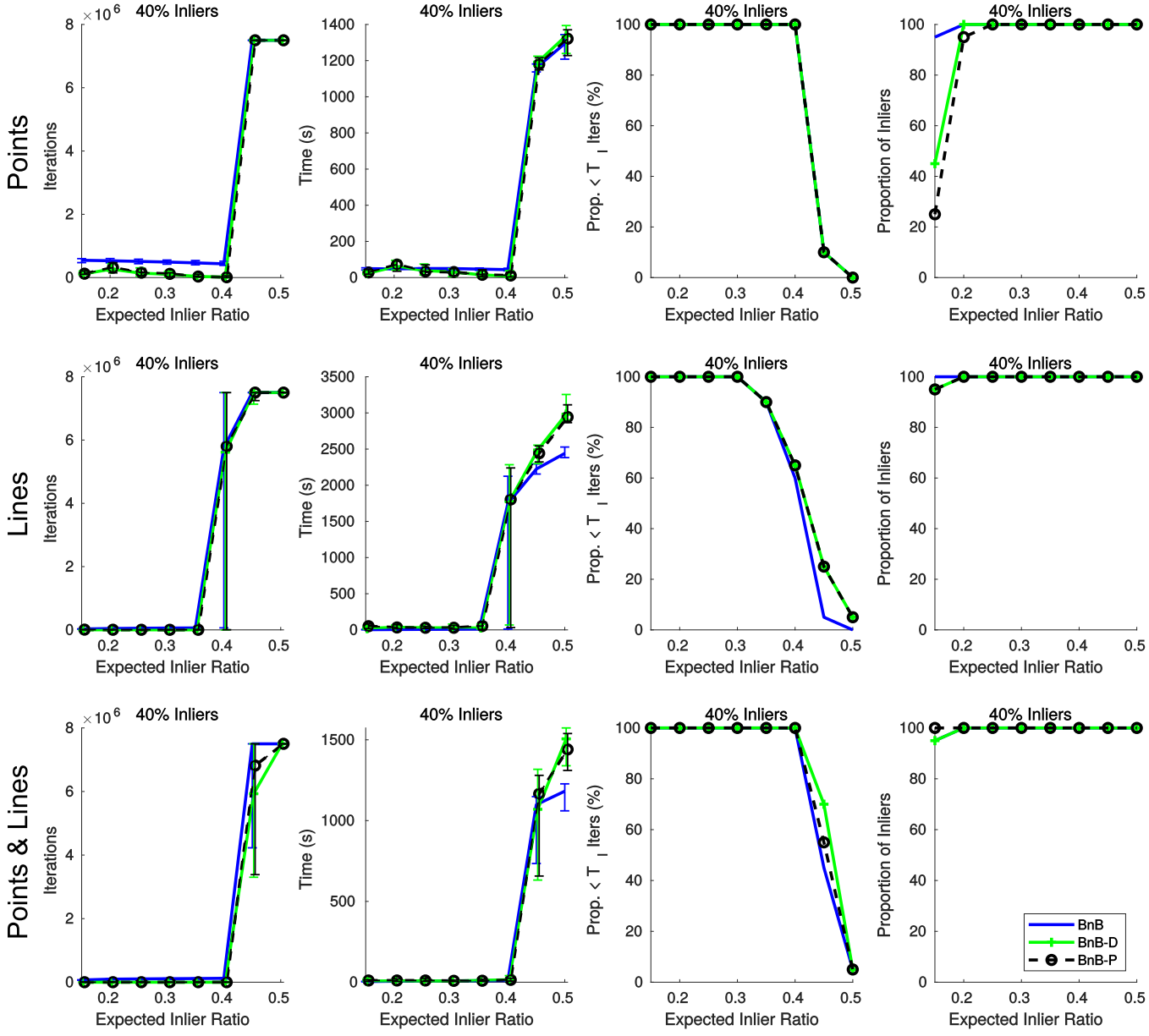

Fig. 6. Median number of iterations (first column) and run-time (second column) with bar showing first and third quartiles, proportion of trials that converged within the iteration limit (third column) and proportion of inlying solutions (fourth column) for *BnB*, *BnB-D*, and *BnB-P* across different assumed inlier ratios, for a feature size of 90. Each experiment was terminated after $T_I = 7.5 \times 10^6$ inner BnB iterations if it had not already converged by then.

the median number of iterations, *BnB-P* performs the fastest, however its iteration count has higher variance than *BnB-D*. Both proposed approaches (*BnB-D* and *BnB-P*) use fewer iterations than the original *BnB*. All methods perform similarly well in terms of the quantity of inlying solutions obtained. As could be expected; all algorithms converge faster where a lower accuracy (higher ϵ) is desired, often to the detriment of the quality of the solution.

With Local Refinement: Next we test with local refinement (Section 4.4), against all other algorithms. The feature sizes range from 40 to 90, with inlier rates at 60%, 40%, and 20%. 30 trials were performed in each case. Results are shown in Fig. 5. From these graphs it is seen that our approaches are consistently more accurate than the state of the art. Interestingly, our approach sometimes does not get the right solution with 20% inliers, despite being globally optimal. It is in fact observed that, in some cases, it obtains a solution whose function value (by (2)) is *lower* than the function value of the ground truth solution, despite being an outlying solution! This is indicative of the intrinsic difficulty of the problem, and the capacity of noise to redefine the global minimum. Also of note is the observation that RANSAC performs better than the state-of-the-art approaches *SoftPosit* and *BlindPnP*. This is largely due to the fact that it is run for a very large number of iterations–the same number that the BnB approaches are run for–in order to compare it against BnB. However, in doing so it performs a much larger search than *SoftPosit* and *BlindPnP*, both of which search locally from one of the 20 pose priors and are orders of magnitude faster than RANSAC. Furthermore, RANSAC has a much higher complexity ($O(N^4 \log(N))$ than *SoftPosit* and *BlindPnP* ($O(N^3)$).

Varying Expected Inlier Ratio: A potential point of concern is that the expected number of inliers, k, cannot be known beforehand. We therefore run experiments to test how the proposed algorithms cope when k is varied away from the true inlier ratio. In Fig. 6 we show results for a feature size of 90, for 40% inliers, for

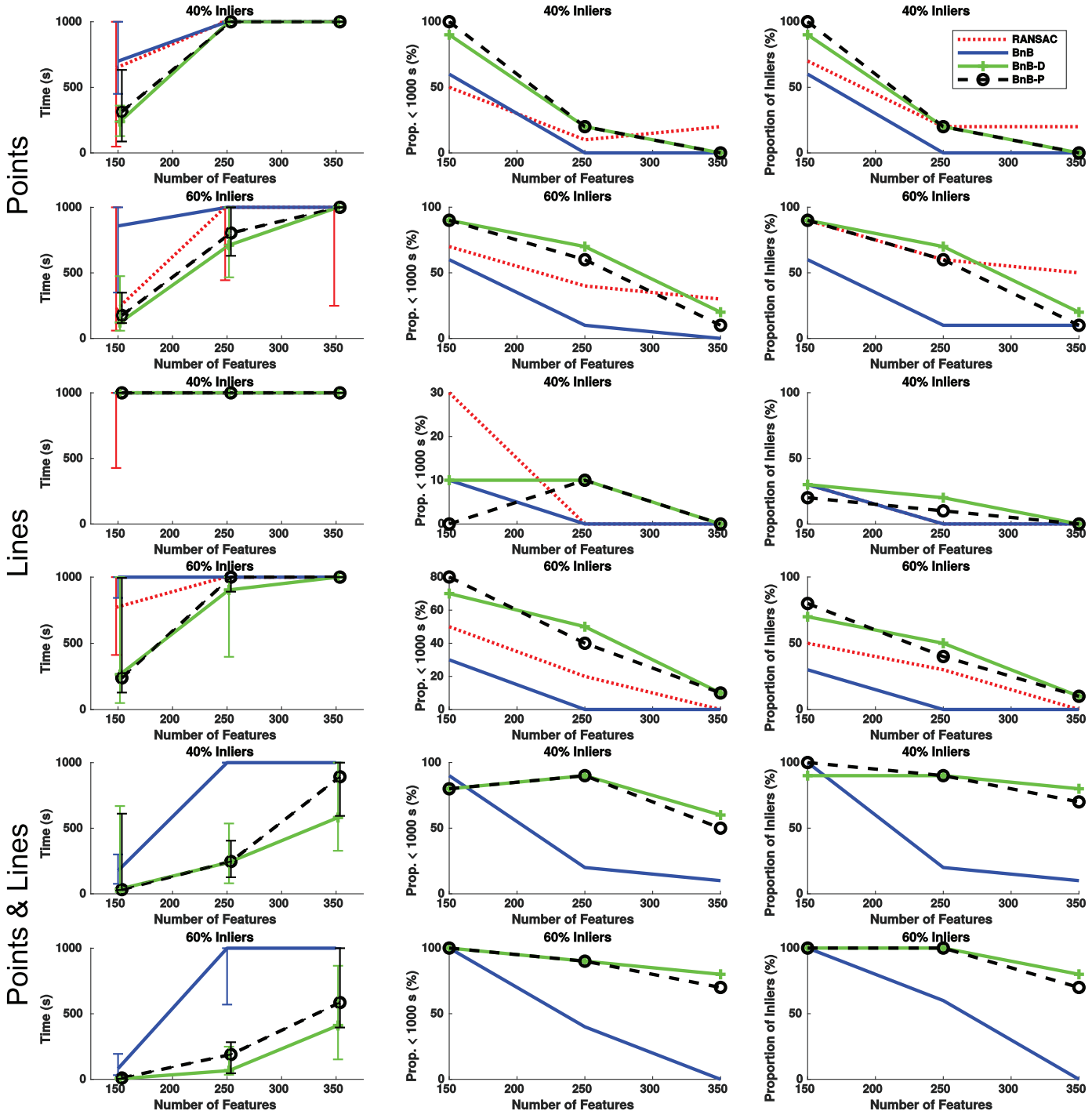

Fig. 7. Median run-time with bar showing first and third quartiles (left column), proportion of trials that converged within the iteration limit (middle column) and proportion of inlying solutions (right column) for *RANSAC, BnB, BnB-D*, and *BnB-P* across different feature sizes for inlier ratios of 40% (rows 1, 3, and 5) and 60% (rows 2, 4, and 6), where no pose priors are used. Results are given for points (top two rows), lines (middle two rows), and both points and lines (bottom two rows). Each trial was terminated after 1000 s if it had not already converged by then. All experiments were performed on the same machine.

varying expected inlier ratio (15%–50%). 20 trials were performed in each case. From these graphs it appears that varying the expected inlier ratio has little effect on the accuracy of the results, with the vast majority of trials converging on the correct answer for the expected inlier ratio anywhere between 25% and 50% (compared to the true ratio of 40%). It is only when the expected inlier ratio is very small (15% – 20%) that the algorithms fail, and this is mostly in the case of point features. However, the number of iterations the approaches take can vary drastically with respect to the expected inlier ratio. In particular, when the expected inlier ratio is higher than the true ratio, the number of iterations significantly increases. This could be due to the fact that, when the expected inlier ratio is at or less than the true ratio, the ground truth pose estimate is sufficiently small to warrant the algorithm to terminate (i.e. the ground truth pose estimate has objective function value in (2) less than ϵ). Thus, the algorithm may only have to find a solution with function value less than ϵ, without explicitly verifying it. This is not always the case, particularly where only line features

Fig. 8. Top: The 3D models used in the 2D–3D dataset. **Bottom**: An example image from each model used in the dataset. From left to right: *Cathedral, Courtyard, Reception, Room, Studio*.

Fig. 9. Qualitative result for solutions returned using all methods on an image from *Reception* dataset. Blue features are 2D, green are 3D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

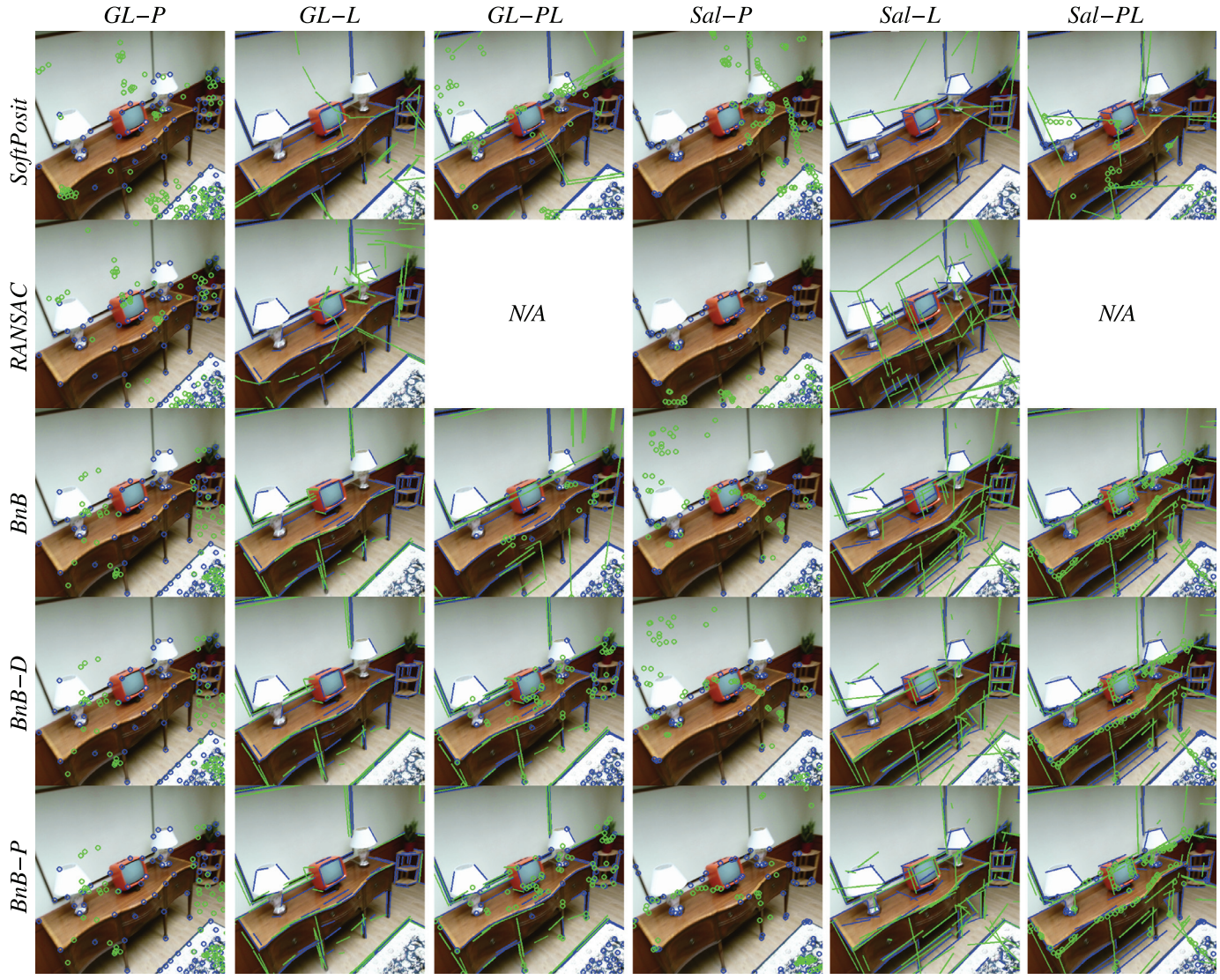

Fig. 10. Qualitative result for solutions returned using all methods on an image from *Room* dataset. Blue features are 2D, green are 3D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

are used and the number of iterations increases more gradually with the expected inlier ratio, however it is a contributing factor.

6.3.2. No pose priors

For a fair comparison against RANSAC that searches over the correspondences (and therefore searches over a large volume of SE(3)), we test our approaches over a much larger prior volume. Specifically, the data is generated by first constructing a random camera pose from the space $\Omega := SO(3) \times [-0.25, 0.25]^3$. The inlying 3D features are generated such that they project onto the camera and lie in $[-1.5, 1.5]^3 \setminus [-0.75, 0.75]^3$. The outlying 3D features are uniformly generated in $[-1.5, 1.5]^3$ and the outlying 2D features are uniformly generated on the image plane.

In this case, we do not test against *BlindPnP* since we are unable to adjust their approach to operate over a significantly larger prior search space. We also do not test against *SoftPosit*: it is observed that *SoftPosit* performs very poorly when the 3D features are so close to the camera centre since it relies upon approximating perspective projection by an orthographic projection. For this reason, *SoftPosit* is also not used as a subroutine for the BnB approaches in this section.

Experiments are performed for larger numbers of features $(150 - 350)$ for 40% and 60% inliers. Each trial is terminated after 1000 seconds if it has not already converged by then. Therefore, all experiments were performed on the same machine, and as such, only 10 trials were recorded in each case. Due to the high variance of timings, the median, and lower and upper quartiles of the time taken are recorded, along with the proportion of trials to converge within the time limit, as shown in Fig. 7. The proportion of inlying solutions is also shown on the right of Fig. 7, where an inlying solution is defined such that the angle between the hypothesised rotation and ground truth rotation is less than 0.1 and the *absolute error* between the two camera centres is less than 0.05. The absolute error is used here due to the camera centres lying in a neighbourhood of the origin (where the relative error is not meaningful).

Based on these results it can be seen that our approaches perform favourably to RANSAC, particularly in the case of lines. RANSAC performs better using points than for lines; this may be due to the different minimal solvers used in each case. The proposed approaches *BnB-D* and *BnB-P* result in a significant speed-up over the original *BnB*, particularly when both points and lines are used.

	GL-P	*GL-L*	*GL-PL*	*Sal-P*	*Sal-L*	*Sal-PL*	Average
SoftPosit	0	0	0	0	0	0	0
RANSAC	0	0	-	0	28.6	-	7.15
BnB	14.3	28.6	28.6	14.3	0	14.3	**16.7**
BnB-D	14.3	14.3	28.6	0	0	**42.9**	**16.7**
BnB-P	0	14.3	28.6	0	0	**42.9**	14.3
Mean	5.72	11.4	21.4	2.86	7.15	**25**	-

(a) *Cathedral*

	GL-P	*GL-L*	*GL-PL*	*Sal-P*	*Sal-L*	*Sal-PL*	Mean
SoftPosit	0	0	0	0	0	0	0
RANSAC	0	0	-	0	25	-	6.25
BnB	0	12.5	0	0	**50**	0	**10.4**
BnB-D	0	0	0	0	**50**	0	8.33
BnB-P	0	0	0	0	**50**	0	8.33
Mean	0	2.5	0	0	**35**	0	-

(b) *Courtyard*

	GL-P	*GL-L*	*GL-PL*	*Sal-P*	*Sal-L*	*Sal-PL*	Mean
SoftPosit	0	0	0	0	0	0	0
RANSAC	0	0	-	9.09	0	-	2.27
BnB	0	0	18.2	0	18.2	18.2	9.08
BnB-D	0	0	9.09	0	**36.4**	27.3	**12.1**
BnB-P	0	0	27.3	9.09	9.09	27.3	**12.1**
Mean	0	0	13.6	3.64	12.7	**18.2**	-

(c) *Reception*

	GL-P	*GL-L*	*GL-PL*	*Sal-P*	*Sal-L*	*Sal-PL*	Mean
SoftPosit	0	0	0	0	0	0	0
RANSAC	0	14.3	-	0	42.9	-	14.3
BnB	14.3	**100**	57.1	0	71.4	28.6	45.2
BnB-D	28.6	**100**	**100**	0	**100**	71.4	**66.6**
BnB-P	14.3	71.4	**100**	0	85.7	42.9	52.3
Mean	11.4	57.2	**64.3**	0	60	35.6	-

(d) *Room*

	GL-P	*GL-L*	*GL-PL*	*Sal-P*	*Sal-L*	*Sal-PL*	Mean
SoftPosit	0	0	0	0	0	0	0
RANSAC	0	0	-	0	0	-	0
BnB	0	0	0	0	0	0	0
BnB-D	0	0	**42.9**	0	0	0	**7.15**
BnB-P	0	0	28.6	0	0	0	4.76
Mean	0	0	**17.9**	0	0	0	-

(e) *Studio*

Fig. 11. Proportion of inlying solutions (%) returned per 3D model and per feature type for each scene.

6.4. Real data

In this section, we evaluate the performance of the registration algorithms on real data. Specifically, we are interested in using the real dataset (as shown in Fig. 8) that comprises five models with between 7 and 11 images with known projection matrices per model. The features used here are salient points (*Sal-P*) by [55] and salient lines (*Sal-L*) by [56]; denoted *Sal-PL* where a mixture of the two are present. State-of-the-art feature detectors are also used here as *GFT* and *Harris* for 2D and 3D points and *LSD* for lines; referred to as *GL-P, GL-L*, and *GL-PL*. The top-120 features are used in 2D and the top-240 used from 3D; except where both points and lines are used where we take the top-80 points and top-80 lines in 2D, alongside the top-160 points and top-160 lines in 3D.

The BnB methods proposed here require priors to be placed on the camera pose. This should not be seen as a significant barrier to the method; indeed, Moreno–Noguer et al. [15] assume the camera to lie on a torus around the object, and Svarm et al. [57] assume the 3D ground plane is known, and the orientation of the image with respect to the ground plane. In our case, we assume the camera centre to lie in a cube of diameter 1.5 m in the case of the indoor models (*Reception, Room*, and *Studio*), and a cube of diameter 5 m for the outdoor *Cathedral* and *Courtyard* models. We place no assumption on the rotation parameters. Each trial is run for a maximum of $T_I = 5 \times 10^6$ iterations for *RANSAC, BnB, BnB-D*, and *BnB-P. SoftPosit* is run for a maximum of 1000 iterations from random starting locations in the prior so as to take a similar amount of time to the other tested methods.

Firstly qualitative results are presented. Figs. 9 and 10 show estimated poses obtained from all five approaches, using the six types of features. The globally optimal approaches all perform better than the sub-optimal *RANSAC* and *SoftPosit*, particularly in the case of lines. Furthermore, *Sal-L* is more robust in comparison to *GL-L* due to the tendency of *GL-L* to detect multiple lines in a similar location, whereas the *Sal-L* features are more representative of the scene. The problem is mitigated for *GL-PL* where the complementarity of the two feature types results in a more robust objective function.

Secondly, quantitative results are presented, where we firstly measure the proportion of inlying solutions returned. For this we use a threshold of 0.1 radians for the angle between the rotations, and for the camera centre threshold we use a function of the prior size of the camera centre, so as to obtain a fair evaluation between models. For the prior camera centre to have volume d, we take threshold t such that $\frac{4}{3}\pi t^3 = 0.025d$, i.e. there is only a 2.5% chance of obtaining the correct camera centre by chance. For the outdoor *Cathedral* and *Courtyard* with prior camera centre over a volume 125 m^3 the inlier threshold is about 0.91 m, whereas for the indoor *Reception, Room*, and *Studio* it is 0.27 m. Fig. 11 shows the proportion of inlying solutions returned for the sets of 2D–3D features for each model. Note that we compare against the different registration approaches outlined in this paper, and the different feature types.

It is observed that the proposed globally optimal approaches perform significantly better than *SoftPosit* and *RANSAC*. In particular, *SoftPosit* never gets the correct solution–this is in part due to the high rates of outliers, and partly due to *SoftPosit* performing poorly whenever 3D features are close to the camera. *BnB-D* and *BnB-P* generally perform better than *BnB* since they are able to search the transformation space more quickly and are more likely to find the correct solution within the iteration limit. Lines are significantly more robust than points, with some improvement when considering both types of features in the *GL-PL* case. The *Room* model sees the highest proportion of inliers, where all images were registered correctly using certain approaches.

In contrast to the previous quantitative results in this paper, we also measure the quantity of inlying solutions when varying the inlier threshold. In doing so, we jointly measure the accuracy of the proposed approaches and the accuracy of the detected features. The rotation and camera centre inlier thresholds are both varied, where the camera centre threshold is based on the ratio of the inlier volume to the total volume of the camera centre prior (where the ratio was 0.025 for results presented in Fig. 11). The results

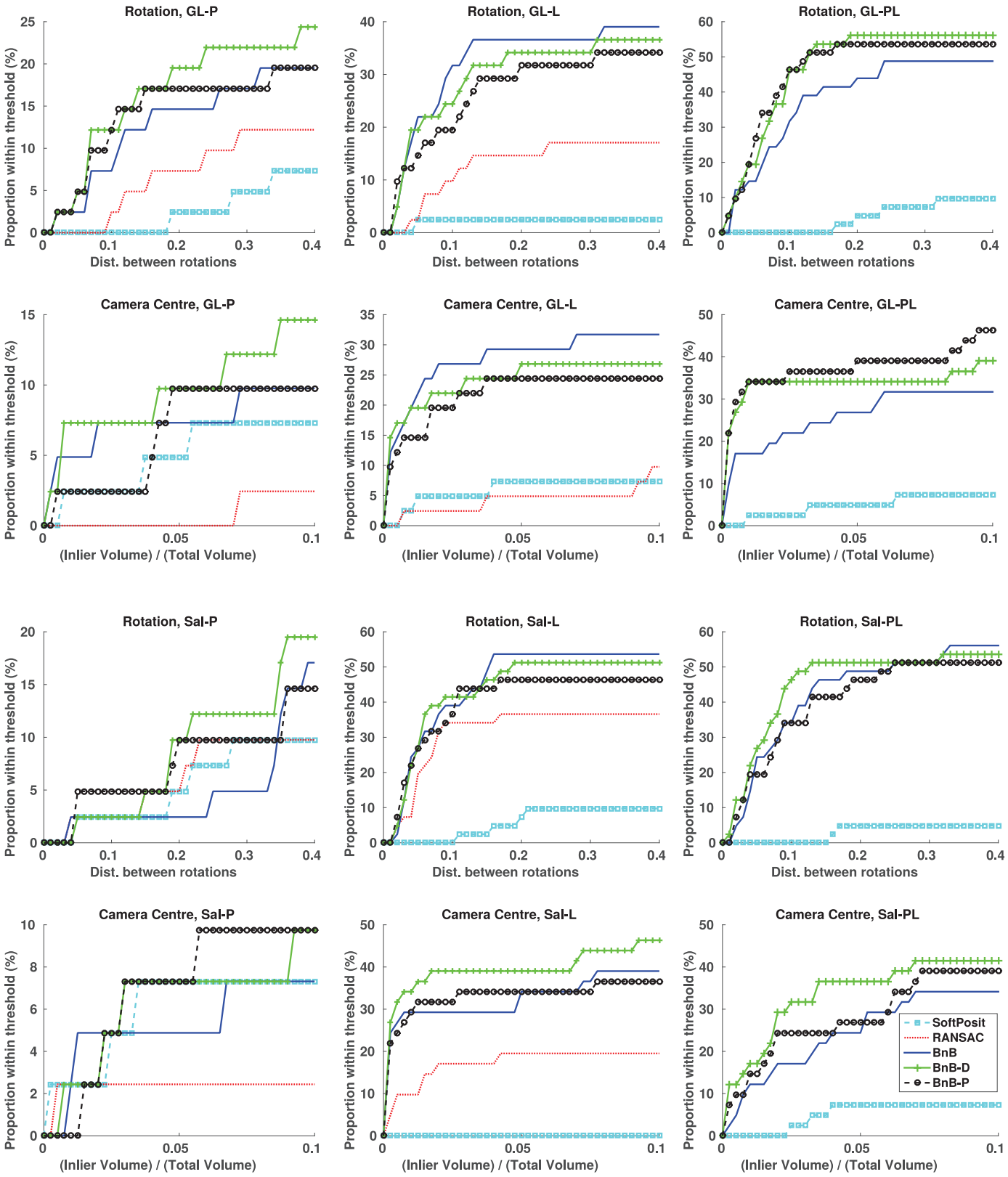

Fig. 12. Proportion of inlying solutions obtained when varying the inlier threshold. There are two graphs for each feature, for the rotation inlier threshold and camera centre inlier threshold.

are given in Fig. 12 where results are given per feature type, and averaged across all datasets.

Similar conclusions may be made as from the tables in Fig. 11: the proposed globally optimal approaches significantly outperform the sub-optimal *RANSAC* and *SoftPosit*; and lines are much more robust than points. *Sal-L* features are registered more accurately than *GL-L* - this may be due to *GL-L* detecting repetitive lines in a similar location and causing ambiguity in determining a correspondence, while *Sal-L* detects a more representative set. On the whole however, *GL-PL* appears to perform the best, despite features that have lower 2D–3D repeatability. Due to the fact that *GL-P* outperforms *Sal-P* we are led to believe this may be due to the proposed salient points being less suited to registration than corners.

7. Conclusions and future work

This paper presented the first globally optimal framework for 2D–3D registration where feature correspondences are unknown. This framework introduced a family of methods, covering both deterministic and non-deterministic formulations, which are applicable to points, lines or a combination of the two, thereby maximising the use of available scene information and broadening the range of practical registration problems that can be tackled. Being based on BnB optimisation, the approaches have intrinsic guarantees on global optimality. Furthermore, the proposed deterministic annealing and probabilistic formulations of nested BnB algorithms have the advantage of allowing for greater efficiency without loss of optimality. This has resulted in algorithms that are significantly better than the state of the art, both in terms of accuracy and robustness to high outlier rates. These advances have been demonstrated and experimentally evaluated on a range of synthetic and challenging real datasets, where significant improvements can be observed.

An interesting avenue for future work would be to explore different ways to apply a BnB algorithm to the problem and their effects on performance. Bazin et al. [35] solve geometry estimation problems using BnB by relaxing non-linear constraints into linear convex and concave envelopes from which upper and lower bounds may be computed by linear programming techniques. Chin et al. [42] explicitly search over feature correspondences; initially hypothesising all correspondences and running a tree search to determine which are invalid. It is unclear at this stage which class of globally optimal method is preferable. However, we have presented the first globally optimal approach to the 2D–3D registration problem that is significantly better than the state of the art for the specific problem.

Research Data

The authors confirm that the indoor and outdoor 2D–3D datasets used for this research are freely available under the terms and conditions detailed in the license agreement enclosed in the data repositories. Details of the data and how to obtain access are available for the *Room* dataset at [58]; and for the *Cathedral, Courtyard, Reception*, and *Studio* datasets at [59].

Declaration of Interest

None.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (grant number EP/K503186/1), the European Commission FP7 IMPART project (grant number 316564) and the EU 2020 project Dreams4Cars (grant number 731593).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.04.002

References

[1] C.F. Olson, A general method for geometric feature matching and model extraction, Int. J. Comput. Vision 45 (1) (2001) 39–54.
[2] D.P. Huttenlocher, S. Ullman, Object recognition using alignment, in: Proc. IEEE Int. Conf. Comput. Vis., 1987, pp. 102–111.
[3] M. Guislain, J. Digne, R. Chaine, G. Monnier, Fine scale image registration in large-scale urban LIDAR point sets, Comput. Vis. Image Underst. 157 (2017) 90–102.
[4] H. Kim, A. Evans, J. Blat, A. Hilton, Multimodal visual data registration for web-based visualization in media production, IEEE Trans. Circuits Syst. Video Technol. 28 (4) (2018) 863–877.
[5] S. Miao, Z.J. Wang, R. Liao, A CNN regression approach for real-time 2D/3D registration, IEEE Trans. Med Imaging 35 (5) (2016) 1352–1363.
[6] W. Yu, M. Tannast, G. Zheng, Non-rigid free-form 2d3d registration using a b-spline-based statistical deformation model, Pattern Recognit. 63 (2017) 689–699.
[7] O. Gmez, O. Ibez, A. Valsecchi, O. Cordn, T. Kahana, 3d-2d silhouette-based image registration for comparative radiography-based forensic identification, Pattern Recognit. 83 (2018) 469–480.
[8] C. Harris, M. Stephens, A combined corner and edge detector, in: Proc. 4th Alvey Vision Conference, 1988, pp. 147–151.
[9] M. Brown, J.-Y. Guillemaut, D. Windridge, A saliency-based approach to 2D-3D registration, in: Proc. International Conference on Computer Vision Theory and Applications (VISAPP), 2014.
[10] R.G. von Gioi, J. Jakubowicz, J.M. Morel, G. Randall, LSD: A fast line segment detector with a false detection control, IEEE Trans. Pattern Anal. Machine Intell. 32 (4) (2010) 35–55.
[11] T. Chen, Q. Wang, 3d line segment detection for unorganized point clouds from multi-view stereo, in: Proce. Asian Conference on Computer Vision, volume 2, 2011, pp. 400–411.
[12] M. Hofer, A. Wendel, H. Bischof, Incremental line-based 3d reconstruction using geometric constraints, in: Proc. British Machine Vision Conference, BMVA Press, 2013.
[13] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[14] P. David, D. DeMenthon, R. Duraiswami, H. Samet, Softposit: Simultaneous pose and correspondence determination, in: Proc. European Conference on Computer Vision, 2002, pp. 698–714.
[15] F. Moreno-Noguer, V. Lepetit, P. Fua, Pose priors for simultaneously solving alignment and correspondence, in: Proc. European Conference on Computer Vision, ECCV '08, 2008, pp. 405–418.
[16] J. Yang, H. Li, D. Campbell, Y. Jia, Go-ICP: A globally optimal solution to 3d ICP point-set registration, IEEE Trans. Pattern Anal. Machine Intell. 38 (11) (2016) 2241–2254.
[17] M. Brown, D. Windridge, J.-Y. Guillemaut, Globally optimal 2D-3D registration from points or lines without correspondences, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 2111–2119.
[18] O. Chum, J. Matas, J. Kittler, Locally optimized RANSAC, in: Pattern recognition, in: Proc. 25th DAGM Symp., volume 2003, Magdeburg, Germany, 2003, pp. 236–243.
[19] E. İmre, A. Hilton, Order statistics of RANSAC and their practical application, Int. J. Comput. Vision 111 (3) (2015) 276–297.
[20] J. Matas, O. Chum, Randomized RANSAC with sequential probability ratio test, in: Proc. IEEE Int. Conf. Comput. Vis., 2005, pp. 1727–1732.
[21] A. Kendall, M. Grimes, R. Cipolla, Posenet: A convolutional network for real–time 6-DOF camera relocalization, in: Proc. Int. Conf. Comput. Vision, 2015, pp. 2938–2946.
[22] A. Kendall, R. Cipolla, Geometric loss functions for camera pose regression with deep learning, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6555–6564.
[23] J. Shotton, B. Glocker, Z.C. S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in rgb-d images, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2013, pp. 2930–2937.
[24] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, C. Rother, DSAC – differentiable RANSAC for camera localization, in: Proc. Conf. Comput. Vis. Pattern Recognit, 2017, pp. 2492–2500.
[25] O. Enqvist, K. Josephson, F. Kahl, Optimal correspondences from pairwise constraints, in: Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 1295–1302.
[26] H. Zhou, T. Zhang, W. Lu, Vision-based pose estimation from points with unknown correspondences, in: IEEE Trans. Image Process., 23, 2014, pp. 3468–3477. 8
[27] M. Marques, M. Stosic, J. Costeira, Subspace matching: Unique solution to point matching with geometric constraints, in: Proc. IEEE Int. Conf. Comput. Vis., 2009, pp. 1288–1294.
[28] J. Beveridge, E.M. Riseman, Optimal geometric model matching under full 3D perspective, Comput. Vis. Image Underst. 61 (3) (1995) 351–364.

[29] S. Bhat, J. Heikkilä, Line matching and pose estimation for unconstrained model-to-image alignment, in: Proc. International Conference on 3D Vision, 2014, pp. 155–162.
[30] P. David, D. DeMenthon, R. Duraiswami, H. Samet, Simultaneous pose and correspondence determination using line features, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2003, pp. 424–431.
[31] P. David, D. DeMenthon, Object recognition in high clutter images using line features, in: Proc. IEEE Int. Conf. Comput. Vis., 2005, pp. 1581–1588.
[32] W.J. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, IEEE Trans. Pattern Anal. Machine Intell. 17 (8) (1995) 749–764.
[33] C. Olsson, F. Kahl, M. Oskarsson, Branch and bound methods for euclidean registration problems, IEEE Trans. Pattern Anal. Machine Intell. 31 (5) (2009) 783–794.
[34] Y. Zheng, S. Sugimoto, M. Okutomi, A branch and contract algorithm for globally optimal fundamental matrix estimation, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2011, pp. 2953–2960.
[35] J.C. Bazin, H. Li, I. Kweon, C. Demonceaux, P. Vasseur, K. Ikeuchi, A branch-and-bound approach to correspondence and grouping problems, IEEE Trans. Pattern Anal. Machine Intell. 35 (7) (2013) 1565–1576.
[36] F. Jurie, Solution of the simultaneous pose and correspondence problem using gaussian error model, Comput. Vis. Image Underst. 73 (3) (1999) 357–373.
[37] T.M. Breuel, Implementation techniques for geometric branch-and-bound matching methods, Comput. Vis. Image Underst. 90 (2003). 294
[38] J. Fredriksson, V. Larsson, C. Olsson, Practical robust two-view translation estimation, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2015.
[39] O. Enqvist, F. Kahl, Two view geometry estimation with outliers, in: Proc. British Machine Vision Conference, 2009.
[40] R. Hartley, F. Kahl, Global optimization through rotation space search, J. Comput. Vision 82 (1) (2009) 64–79.
[41] A.P. Bustos, T.J. Chin, D. Suter, Fast rotation search with stereographic projections for 3D registration, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2014.
[42] T.J. Chin, P. Purkait, A. Eriksson, D. Suter, Efficient globally optimal consensus maximisation with tree search, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2015, pp. 2413–2421.
[43] D.P. Paudel, A. Habed, C. Demonceaux, P. Vasseur, Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes, Proc. Int. Conf. Comput. Vision (2015) 2048–2056.
[44] D. Campbell, L. Petersson, L. Kneip, H. Li, Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence, in: Proc. Int. Conf. Comput. Vision, 2017.
[45] A. Ansar, K. Daniilidis, Linear pose estimation from points or lines, IEEE Trans. Pattern Anal. Machine Intell. 25 (5) (2003) 578–589.
[46] B. Kamgar-Parsi, B. Kamgar-Parsi, Matching 2D image lines to 3D models: Two improvements and a new algorithm, Proc. Conf. Comput. Vis. Pattern Recognit. (2011) 2425–2432.
[47] M. Brown, D. Windridge, J.-Y. Guillemaut, A family of globally optimal branch-and-bound algorithms for 2D-3D correspondence-free registration – convergence analysis, supplementary report, 2019.
[48] V. Lepetit, F. Moreno-Noguer, P. Fua, Epnp: An accurate O(n) solution to the pnp problem, Int. J. Comput. Vision 81(2)
[49] R. Kumar, A. Hanson, Robust methods for estimating pose and a sensitivity analysis, CVGIP: Image Understand. 60 (3) (1994) 313–342.
[50] F. Dornaika, C. Garcia, Pose estimation using point and line correspondences, Real-Time Imaging 5 (3) (1999) 215–230.
[51] D.F. Dementhon, L.S. Davis, Model-based object pose in 25 lines of code, Int. J. Comput. Vision 15 (1-2) (1995) 123–141.
[52] O. Chum, J. Matas, Matching with PROSAC - progressive sample consensus, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2005, pp. 220–226.
[53] L. Kneip, D. Scaramuzza, R. Siegwart, A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation, Proc. Conf. Comput. Vis. Pattern Recognit. (2011) 2969–2976.
[54] M. Dhome, M. Richetin, J. Lapresté, G. Rives, Determination of the attitude of 3D objects from a single perspective view, IEEE Trans. Pattern Anal. Machine Intell. 11 (12) (1989) 1265–1278.
[55] M. Brown, D. Windridge, J.-Y. Guillemaut, A generalised framework for saliency-based point feature detection, Comput. Vis. Image Underst. 157 (2017) 117–137.
[56] M. Brown, D. Windridge, J.-Y. Guillemaut, A generalisable framework for saliency-based line segment detection, Pattern Recognit. 48 (12) (2015) 3993–4011.
[57] L. Svarm, O. Enqvist, M. Oskarsson, F. Kahl, Accurate localization and pose estimation for large 3D models, in: Proc. Conf. Comput. Vis. Pattern Recognit., 2014.
[58] M. Klaudiny, M. Tejera, C. Malleson, J.-Y. Guillemaut, A. Hilton, SCENE digital cinema datasets, 2015. doi: 10.15126/surreydata.00807665.
[59] H. Kim, IMPART multi-modal dataset, 2014. doi: 10.15126/surreydata.00807707.

Mark Brown received his BSc in Mathematics from the University of Bath in 2012 and his PhD in Electronic Engineering from the University of Surrey in 2016, with his thesis entitled Saliency Based Framework for Multi-Modal Registration His research interests include feature detection and geometry estimation.

David Windridge is Associate Professor in Computer Science at Middlesex University and leads the University's Data Science activities. His research interests centre on machine learning, cognitive systems and computer vision. He has authored and played a leading role on a number of large-scale machine learning projects in academic and industrial research settings, and has also won various interdisciplinary research grants.

Jean-Yves Guillemaut is Senior Lecturer in 3D Computer Vision at the University of Surrey. His main areas of expertise include 3D reconstruction, multi-modal registration, camera calibration and 3D video applications. His current research focuses on developing novel video-based modelling techniques for the reconstruction of outdoor scenes and scenes with complex surface reflectance properties.