Efficient Geometric Matching with Polar Bounds for Aligning Star Field Images

Ryan James Marker, Tat-Jun Chin, Garry Newsam
University of Adelaide, Australia
ryan.marker@student.adelaide.edu.au, {tat-jun.chin; garry.newsam}@adelaide.edu.au

Abstract

For many autonomous satellites, maintaining current knowledge of their location and pose is critical to mission success. Whilst there exist mature solutions for positioning and attitude estimation in space, computer vision approaches are emerging as viable and cheaper alternatives. At the core of a vision-based satellite motion estimation system is registering the successive images taken in space. This is a challenging task due to the lack of visually salient features in star field images. In this research, we propose a novel technique for registering star field images. We formulate the task as a geometric matching optimisation problem, which is solved using branch-and-bound. Our main contribution lies in devising a novel *polar* bounding function that provides tighter bounds than existing methods. Our bounding function is also easy to evaluate, requiring no more computations than available bounding functions. As demonstrated on actual space images, both factors contribute to speed up geometric matching by a large degree. The proposed technique further illustrates the promise of vision-based sensors for space missions.

1 Introduction

There is an ever increasing number of satellites being put into orbit to serve a wide range of services, such as telecommunications, broadcasting, environmental monitoring, and defence. Many of the satellites require autonomous or semi-autonomous decision making capabilities, due to the distance to the Earth's surface and the extreme velocity at which they travel. In order to correctly function, autonomous satellites require a host of measurements to aid in situational assessment, maintaining control, and reacting to unfolding events. As one of the most technologically advanced human endeavours, many sophisticated instruments have already been developed for satellite usage.

Recent years have seen a significant increase in proposals to use cameras to take many of these measurements. Unlike many other sensors, cameras are low-power and passive. The imaging sensor is also capable of capturing more general information about the surrounding environment, unlike other "dedicated" sensors. The widespread proliferation of cameras in consumer devices also means that cameras can be exploited as very low-cost general purpose sensors for space. The downside of using cameras is that they usually do not directly measure the quantity of interest. Some form of processing must be conducted on the images, before an understanding about the surroundings as well as the status of the satellite itself can be derived. To this end, computer vision algorithms can be exploited for space applications.

Recent examples of applying cameras and computer vision algorithms in space include the work done by Terui *et al.* [2006], Schnitzer *et al.* [2012] and Thomas *et al.* [2016], who utilise stereo imaging systems and structure-from-motion algorithms to reconstruct 3D objects in orbit (e.g., other satellites or debris) for proximity operations (e.g., manipulation, docking). Fasano *et al.* [2013] use a monocular camera on an orbiting satellite to measure its angular velocity. Riot *et al.* [2013] discusses the usage of an imaging system for space-based space surveillance, e.g., detecting space debris.

In this paper, we focus on a task that underpins many of the operation of an autonomous satellite in orbit - ego-motion estimation. Specifically, a sequence of images are recorded using a single camera as the satellite traverses its orbit. To characterise the rigid motion of the satellite, the image transformations between successive frames must be estimated. Equivalently, we must register the images such that the corresponding regions are aligned. The primary challenge in this task derives from the paucity of visually distinct features in star field images. Figure 1 shows typical star field images that were recorded in sequence. The lack of salient and dis-

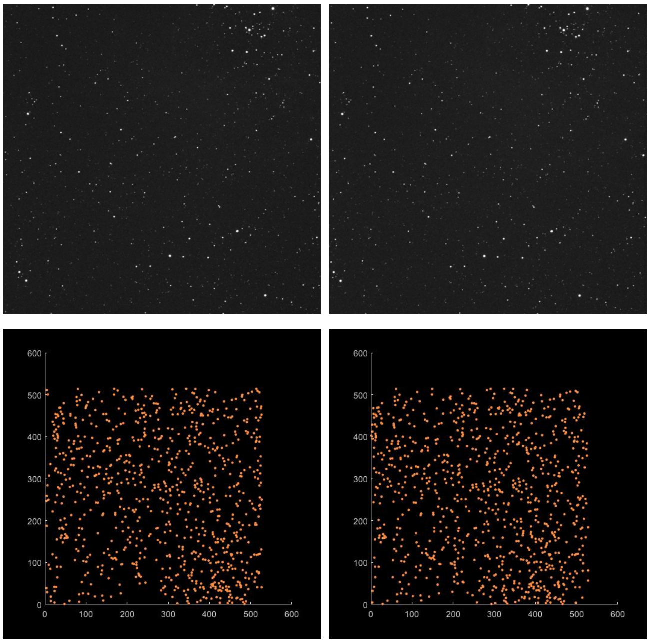


Figure 1: (Top row) Successive frames from an image sequence recorded in space (best viewed in closeup). For better presentation, these images differ only by a translation. In general, however, more complex motions occur. (Bottom row) Preprocessing reduces the images into point sets M and B.

criminative features precludes the usage of state-of-the-art feature-based estimation techniques [Se *et al.*, 2001; Mur-Atal *et al.*, 2015].

To effectively register star field images, we formulate the task as a *geometric matching* problem [Grimson, 1990], which directly aligns the pattern of star distributions between images. An established approach for geometric matching is branch-and-bound (BnB) [Breuel, 2003], which recursively partitions and prunes the parameter space to find the optimal transformation. Our main contribution lies in devising a novel *polar* bounding function that provides tighter bounds than the standard "Euclidean" bounding function [Breuel, 2003]. By defining an efficient data structure in polar space, our bound is also as cheap to compute as the Euclidean bound. The net effect is a significant reduction in runtime of BnB for geometric matching, as we demonstrate on synthetic data and real star field images.

1.1 Related work

In the context of our target problem, a closely related work is by Fasano *et al.* [2013], who aimed to track the angular velocity of a satellite by analysing star field images. Similar to our setting, they place a camera on the satellite and take successive images of the stars. Their primary machinery to estimate the relation between successive images is via optical flow [Horn and Schunc, 1981], which is then used to determine the angular velocity of the satellite. It is well known, however, that optical flow computation can be brittle [Baker *et al.*, 2011]. It is also unknown if the accuracy of current flow techniques, which were developed based on more general scenes, can be achieved on visually sparse space images. In contrast to Fasano *et al.* [2013], our geometric matching approach directly estimates the image transformation based on a clear cut objective function and a BnB algorithm which guarantees global optimality.

The state-of-the-art technology for motion estimation (specifically, attitude determination) on spacecraft is star trackers [Birnbaum, 1996]. Most star trackers operate by identifying known stars in the image and comparing them with a star catalogue (a step called star identification [Spratling and Mortari, 2009]) to establish the camera pose at each frame. This obviates the need for frame-to-frame comparisons to estimate the motion. However, not all missions can host a dedicated star tracker, due to the high cost of the required optics, sensors, memory and processing [Rawashdeh *et al.*, 2012; Fasano *et al.*, 2013].

More broadly related are works by Terui *et al.* [2006], Schnitzer *et al.* [2012] and Thomas *et al.* [2016], who utilised stereo vision to reconstruct the 3D shapes of objects in space and/or their motion (recall that under the fundamental imaging constraints, structure and motion must be simultaneously estimated). Underpinning the estimation is again solving for the correspondences between overlapping or successive images, which are then subjected to triangulation for shape recovery. In these works, since the object of interest (debris, spacecraft, etc.) may contain visually interesting features, they can leverage on existing 2D keypoint matching techniques or heuristic pixel region matching to perform registration.

2 Problem Definition

2.1 Model

Let $\mathcal{M} \subset \mathbb{R}^2$ and $\mathcal{B} \subset \mathbb{R}^2$ represent two successive and overlapping space images taken in a sequence. A core operation to estimate the motion undergone by the camera between $\mathcal{M}$ and $\mathcal{B}$ is to first register the images. By this, we mean finding a transformation $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ such that *corresponding* pixels $m \in \mathcal{M}$ and $b \in \mathcal{B}$ in the images satisfy the equation $b = f(m)$. Once the alignment is achieved, further processing can be conducted to estimate the desired relative motion between the cameras, e.g., rigid transformation or angular difference as in Fasano *et al.* [2013].

For general scenes, the theory of epipolar geometry [Faugeras, 1992] tells us that no parametric function f can achieve the alignment (i.e., any f must be scene dependent and locally varying across $\mathcal{M}$). We argue,

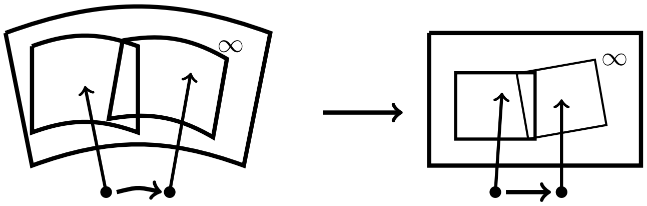


Figure 2: Illustrating our simplified alignment model for star field images. Since the stars in star field images are infinitely far away, the motion model can be approximated well by a 2D rigid transformation, which consists of a 2D translation and 1D rotation.

however, that for star-field images where the "objects" are at a vast distance away, a more restricted form for f is feasible. Specifically, the stars can be assumed to lie on a very distant plane, and the distances of the cameras to the plane are equal. Further, the relative orientation of the cameras can be approximated well by a rotation about the normal direction of the plane. Figure 2 illustrates the simplified model.

Under the assumed imaging conditions, f can be taken as a 2D rigid transformation, i.e., f is isometric and consists of a 2D translation and a 1D rotation

$$f(m|T) = R(\theta)m + [X, Y]', \tag{1}$$

where $T = [X, Y, \theta]$ are the parameters of f,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{2}$$

and m' indicates the transpose of the vector m.

2.2 Geometric matching

Our solution for estimating T for f would first involve preprocessing the images $\mathcal{M}$ and $\mathcal{B}$ to obtain two sets of discrete points $M = \{m\} \subseteq \mathcal{M}$ and $B = \{b\} \subseteq \mathcal{B}$. In this work, we used simple intensity thresholding on the images $\mathcal{M}$ and $\mathcal{B}$, followed by reducing each connected component in the thresholded image into its centroid to form M and B; see Figure 1. We emphasise that the proposed registration algorithm is independent of the specific preprocessing method, and any technique or existing software tool can be used to perform the preprocessing, such as SeExtractor [Bertin and Arnouts, 1996].

The lack of visual features prevents the establishment of prior correspondences between M and B. Thus, any attempt to estimate T must also involve estimating correspondences (either implicitly or explicitly). The geometric matching framework [Grimson, 1990] obtains T by solving the problem

$$T_{\max} = \arg\max_{T \in S} Q(T), \tag{3}$$

where $S = \mathbb{R}^2 \times [-\pi, \pi]$ is the space of all rigid transformations, and the objective function $Q(T)$ is

$$Q(T) = \sum_{m \in M} \max_{b \in B} \lfloor \|f(m|T) - b\| \leq \epsilon \rfloor. \tag{4}$$

Here, $\|\cdot\|$ indicates the Euclidean norm, and $\lfloor \cdot \rfloor$ is the indicator function which returns 0 if its input proposition is false and 1 otherwise. Intuitively, $Q(T)$ counts the number of points in M that have a valid match in B within distance ϵ under T. Problem (3) thus amounts to finding the T that aligns the most number of points.

Note that evaluating $Q(T)$ involves a proximity search, i.e., for each $f(m|T)$ we ask if there is a $b \in B$ that is within distance ϵ. Such a query can be conducted efficiently by indexing B in a kd-tree and querying the tree with $f(m|T)$. The max operator in (4) effectively associates each point in M with a point in B under T. Note, however, that due to the threshold ϵ, not all points will be matched - these behave as "outliers" to T. The usage of a threshold is vital to prevent spurious detections and stars in the non-overlapping regions of $\mathcal{M}$ and $\mathcal{B}$ from affecting the result.

3 Branch-And-Bound

A popular approach for registering point sets is ICP [Besl and MacKay, 1992]. However, since the method is based on the maximum likelihood criterion, it is not robust towards outliers (non-matching points), which may exist in large quantities in our images if the satellite speed is significant. Whilst robust variants of ICP exist, the algorithms are still only locally optimal, thus the quality of the solution depends heavily on the initialisation. The fact that the quality of the solution is not guaranteed may not be desirable in space missions.

Unlike ICP, BnB guarantees global optimality. The use of BnB for geometric matching was pioneered by Breuel [2003]. Algorithm 1 summarises the method. The algorithm is a form of tree search where every node in the tree represents a region of the solution space S. The search is initialised with the root node containing an initial rectangular region

$$s = [X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}] \times [\theta_{\min}, \theta_{\max}] \tag{5}$$

of S, where s contains the optimal solution $T_{\max}$ (s can be initialised conservatively to ensure that this happens). The method then recursively partitions s equally into eight smaller rectangular regions $\{s_i\}_{i=1}^{8}$.

Each subregion s_i is pruned if it is judged to not contain a better solution than the current best $\tilde{T}$. The pruning is accomplished by means of an upper bound function $\widehat{Q}(s)$, which must satisfy the property

$$\widehat{Q}(s) \geq \max_{T \in s} Q(T) \tag{6}$$

Algorithm 1 BnB for geometric matching (3).

```
Input: Point sets $M = \{m\}$ and $B = \{b\}$, initial rectangular region $s \subset S$, threshold $\epsilon$.
Output: The optimal rigid transformation $T_{\max}$.
$\tilde{Q} \leftarrow 0$, $\tilde{T} \leftarrow NULL$
queue.add($s$, $\widehat{Q}(s)$)
loop
    $s \leftarrow$ queue.best()
    $T_C \leftarrow$ centre($s$)
    if $Q(T_C) = \widehat{Q}(s)$ then
        return $T_{\max} = T_C$
    else if $Q(T_C) > \tilde{Q}$ then
        $\tilde{Q} \leftarrow Q(T_C)$, $\tilde{T} \leftarrow T_C$
    end if
    $\{s_i\}_{i=1}^{8} \leftarrow$ partition $s$ into 8 equal subregions
    for all $s_i$ do
        if $\widehat{Q}(s_i) > \tilde{Q}$ then
            queue.add($s_i$, $\widehat{Q}(s_i)$)
        end if
    end for
end loop
```

for all $s \subset S$. Specifically, if

$$\widehat{Q}(s_i) \leq Q(\tilde{T}), \tag{7}$$

then we can conclude that s_i does not contain a better solution than $\tilde{T}$. If s_i cannot be pruned, it is inserted into a priority queue to be further partitioned. The incumbent solution $\tilde{T}$ is updated by checking whether the centre point T_C of a region retrieved from the queue is of higher quality.

To ensure that the algorithm eventually converges to $T_{\max}$, the items s in the queue must be prioritised based on their upper bound value $\widehat{Q}(s)$, i.e., the most promising node is considered first. Another requirement is that

$$\lim_{s \to T} \widehat{Q}(s) = Q(T). \tag{8}$$

Intuitively, the above condition states that $\widehat{Q}(s)$ must converge to $Q(T)$ as s converges to a single point T.

3.1 Bounding function

Evidently, the most crucial component of BnB is the bounding function $\hat{Q}$. To construct $\hat{Q}$, it is helpful to first bound the position of $f(m|T)$ under all $T \in s$. By the form of the 2D rigid transform (1), the effects of the translational and angular ranges in s can be analysed separately. Under the angular range $[\theta_{\min}, \theta_{\max}]$, the point m is rotated to lie on an arc, where the centre point of the arc is the point obtained by rotating m with the "centre" angle

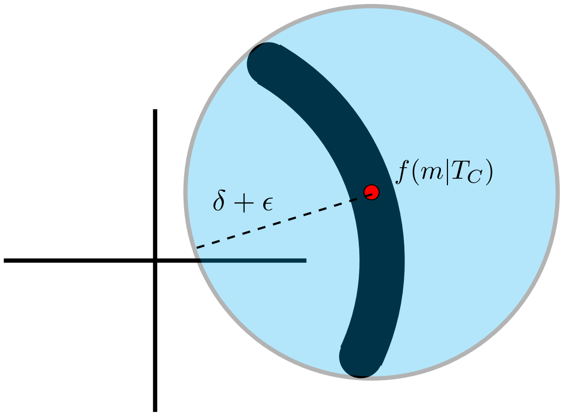


Figure 3: A RAS for point m arising from a region of parameters $s \subset S$. The original bounding function of Breuel bounds the RAS with the tightest bounding circle centred at the centroid of the RAS.

$$\theta_C = 0.5(\theta_{\min} + \theta_{\max}). \tag{9}$$

For the translational range $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}]$, let the centre of the range be

$$[X_C, Y_C] = 0.5[X_{\min} + X_{\max}, Y_{\min} + Y_{\max}], \tag{10}$$

and let

$$d = \frac{1}{2}\sqrt{(X_{\max} - X_{\min})^2 + (Y_{\max} - Y_{\min})^2} \tag{11}$$

be the distance of the farthest point in the range to the centre. Under all translations in the range, the position of m is bounded to within a disc with centre $m + [X_C, Y_C]'$ and radius d. Cumulatively, the location of $f(m|T)$ under the translational and rotational ranges in s is bounded within a *rounded annulus segment (RAS)*; see Figure 3. The geometric centre of the RAS is

$$f(m|T_C), \tag{12}$$

where $T_C = [X_C, Y_C, \theta_C]$ is the centre of s.

The original method of Breuel bounds the RAS corresponding to s with the tightest bounding disc centred at $f(m|T_C)$. This gives rise to the bounding function

$$\widehat{Q}(s) = \sum_{m \in M} \max_{b \in B} \lfloor \|f(m|T_C) - b\| \leq \delta + \epsilon \rceil, \tag{13}$$

where δ, computed as follows

$$\delta = \|f(m|T_C) - f(m|[X_C, Y_C, \theta_{\min}])\| + d \tag{14}$$

is the radius of the smallest disc centred on $f(m|T_C)$ that contains the RAS; see Figure 3. By construction, since for all $T \in s$, $f(m|T)$ must lie in this disc, (13) is a valid upper bounding function. It can be shown that as s converges into a point T, the bounding function (13) becomes $Q(T)$; see Breuel [2003] for details. Computationally, evaluating (13) again involves proximity operations as in the calculation of the quality function $Q(T)$. We can thus exploit kd-trees again for efficient evaluation.

Much of the later works on point cloud registration based on BnB can be viewed as derivations of Breuel's method. Whilst the later techniques were developed for different variants/settings of the problem such as pure rotational registration [Hartley and Kahl, 2009; Bazin *et al.*, 2013], full 6 DoF rigid registration [Olsson *et al.*, 2009; Yang *et al.*, 2016], and the incorporation of a priori determined correspondences, the "essence" of Breuel's idea, i.e., bounding the uncertainty region with the tightest bounding disc, remained.

4 Polar Bounding Function

Intuitively, however, Breuel's original bounding function is very conservative, in that it constructs a bound for the area of $f(m|T)$ under $T \in s$ that can be much larger than the actual area. There have been attempts [Chin *et al.*, 2014; Bustos *et al.*, 2014] to improve the bound developed by Breuel, however, they have dealt with only the pure rotation case. In particular, for the case of 1D rotation, Chin *et al.* [2014] bound the position due to angular uncertainty using an arc, then evaluates the bound by solving an interval stabbing problem. Their method, however, cannot be applied on 2D rigid transformations.

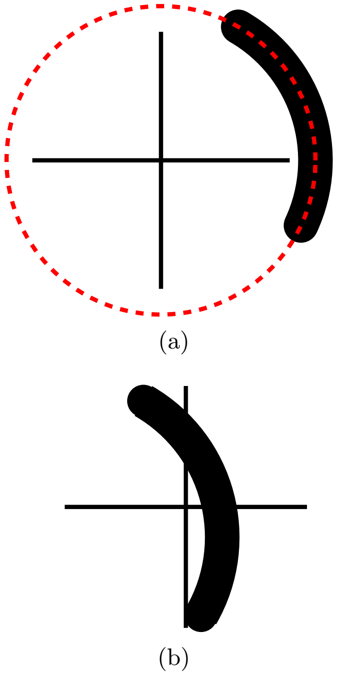
Our main contribution in this paper is a novel *polar* bounding function that is generally tighter than the original "Euclidean" bound (13). Our bounding function is also as efficient to evaluate, i.e., it too requires only a kd-tree query. To begin, we modify the "standard" rigid transformation (1) to the following equivalent form by changing the order in which the translation and rotation are applied

$$f(m|T) = R(\theta)\left(m + [X, Y]'\right). \tag{15}$$

This change is necessary to ensure a form of invariance for the uncertainty area given s. Specifically, by translating first before rotating, we ensure that the RAS arising from s is always concentric with a circle which is centred at the origin; see Figure 4(a). Contrast this with the RAS under the conventional rigid transformation (1), which can have an arbitrary position; see Figure 4(b).

With the above manipulation, we construct the following bounding function

$$\widehat{Q}_{polar}(s) = \sum_{m \in M} \max_{b \in B} \lfloor b \in \Omega(m|s,\epsilon) \rfloor, \tag{16}$$


(a)

(b)

Figure 4: (a) By defining the rigid transformation to consist of a translation followed by a rotation, the uncertainty region of $f(m|T)$ is always an RAS which is centred on the origin. (b) When the rigid transformation is defined conventionally, i.e., a rotation followed by a translation, the uncertainty region is an RAS which can be situated anywhere.

where $\Omega(m|s,\epsilon)$ is an *annulus segment (AS)* that bounds the RAS of m due to s; see Figure 5 (note that an AS has straight ends whilst an RAS has rounded ends). Specifically, the AS is defined as

$$\begin{aligned}\Omega(m|s,\epsilon) = \{p \in \mathbb{R}^2 \; | r_\ell(m,s,e) \le \|p\| \le r_h(m,s,e),\\ \theta_\ell(m,s,e) \le \alpha(p) \le \theta_h(m,s,e)\},\end{aligned} \tag{17}$$

where $\alpha(p)$ is the angle between p and the horizontal axis. To ensure that $\Omega(m|s,\epsilon)$ bounds the RAS, the boundary values are computed as follows:

$$\begin{aligned}r_\ell(m,s,e) &= \|m + [X_C, Y_C]'\| - d - \epsilon,\\ r_h(m,s,e) &= \|m + [X_C, Y_C]'\| + d + \epsilon,\\ \theta_\ell(m,s,e) &= \angle(R(\theta_{\min})(m + [X_C, Y_C]')) - \arctan\frac{d+\epsilon}{r_\ell},\\ \theta_h(m,s,e) &= \angle(R(\theta_{\max})(m + [X_C, Y_C]')) + \arctan\frac{d+\epsilon}{r_\ell}.\end{aligned} \tag{18}$$

By construction, $\widehat{Q}_{polar}(s)$ is a valid upper bounding function to $Q(T)$, since $\Omega(m|s,e)$ bounds the position of $f(m|T)$ under all $T \in s$. To establish the required convergence (8) for $\widehat{Q}_{polar}(s)$, observe that

$$\lim_{s \to T} d = 0, \tag{19}$$

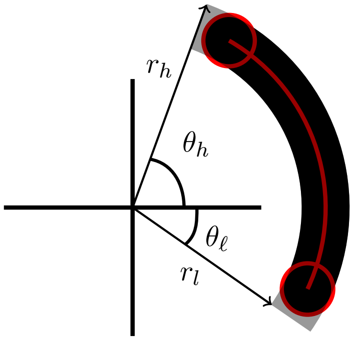


Figure 5: An annulus segment (AS) $\Omega(m|s,e)$ that bounds the RAS for m due to the region $s \subset S$.

which causes the bounding area $\Omega(m|s,\epsilon)$ to reduce to the single point $f(m|T)$, surrounded by the the threshold ϵ. This bounding area matches the definition of $Q(T)$ in (4).

4.1 Efficient evaluation

A tighter bound is impractical if it is costly to evaluate. The key principle which allows $\widehat{Q}_{polar}$ to be efficiently computed, is that $\Omega(m|s,\epsilon)$ is a rectangular region in polar coordinates. Specifically, under the mapping $\phi : \mathbb{R}^2 \mapsto \Theta$, where $\Theta = [0, 2\pi] \times \mathbb{R}_+$ indicates the polar space, and

$$\phi(m) = [\alpha(m), \|m\|]', \tag{20}$$

$\Omega(m|s,\epsilon)$ is "unwrapped" to become the rectangular region

$$\begin{aligned}\{[\alpha, r] \in \Theta \,|\theta_\ell(m,s,e) \leq \alpha \leq \theta_h(m,s,e),\\ r_\ell(m,s,e) \leq r \leq r_h(m,s,e)\}\end{aligned} \tag{21}$$

in Θ. Figure 6 illustrates this representation.

In polar space, the query

$$\max_{b \in B} \lfloor b \in \Omega(m|s,\epsilon) \rceil \tag{22}$$

can thus be answered efficiently using *orthogonal range search* [de Berg *et al.*, 2008, Chapter 5]. In more detail, the point set B is first mapped to Θ where it is then indexed in a *2D range tree* in $\mathcal{O}(n \log n)$ time (note that this can be done "offline" prior to the main loop in Algorithm 1). During BnB search, the query (22) is answered by searching the 2D range tree with the rectangular region (21) in $\mathcal{O}(\sqrt{n} + k)$ time, where k is the number of points from $\phi(B)$ that lie in the rectangular range. Note, in fact, that if the answer to (22) is affirmative, we need not retrieve all k points in the output and the range query can be terminated immediately as soon as a single point that lies in the range is found.

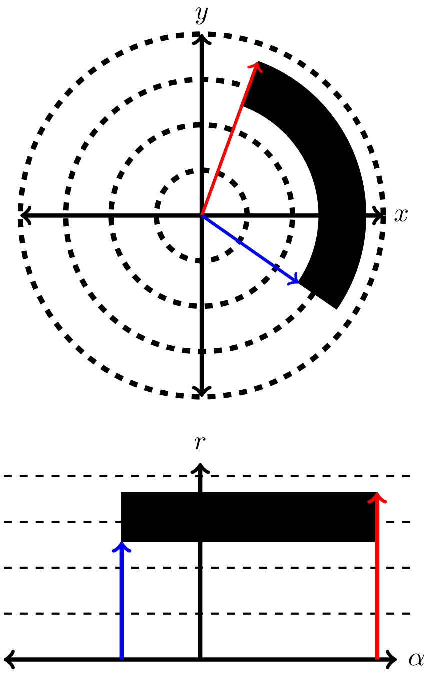


Figure 6: When converted to polar coordinates, an annulus segment becomes a rectangular region.

4.2 Boundary cases

Conducting an inherently Euclidean operation such as (22) in polar space Θ will inevitably incur boundary issues, since the angular axis in Θ does not automatically "wrap around". In the following, we describe the boundary issues that affect (22) and their remedies.

Case 1

If the angular range $[\theta_\ell(m,s,e), \theta_h(m,s,e)]$ in $\Omega(m|s,\epsilon)$ extends beyond the standard range $[0, 2\pi]$ in the horizontal axis of Θ (this is the case depicted in Figure 6), the rectangular region $\Omega(m|s,\epsilon)$ in Θ should be wrapped around to ensure the correctness of the query (22). We accomplish this by simply tiling the point set $\phi(B)$ before building the 2D range tree. Specifically, two copies of $\phi(B)$ are created, $\phi(B) - [2\pi, 0]'$ and $\phi(B) + [2\pi, 0]'$, and the 2D range tree is constructed using all the points

$$\phi(B) - [2\pi, 0]' \cup \phi(B) \cup \phi(B) + [2\pi, 0]' \tag{23}$$

over the bounding range

$$[-2\pi, 4\pi] \times [0, \max_{b \in B}(\|b\|)] \tag{24}$$

in the expanded Θ. The fact that

$$\theta_h(m,s,e) - \theta_\ell(m,s,e) \leq 2\pi \tag{25}$$

ensures that no "double counting" which leads to a false positive outcome to query (22) will occur.

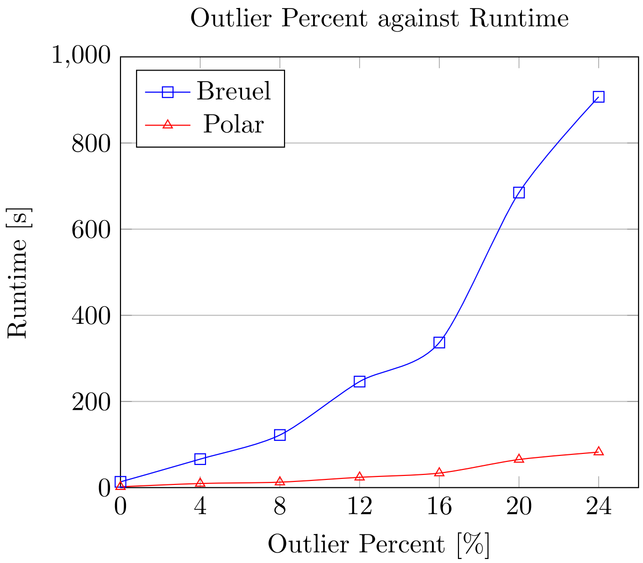


Figure 7: Runtime results for the synthetic data.

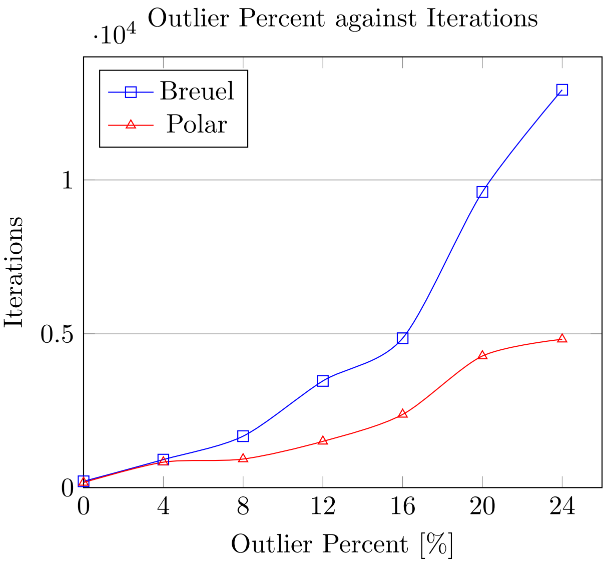


Figure 8: Iteration results for the synthetic data.

Case 2

If the lower radial limit $r_\ell(m, s, e)$ of $\Omega(m|s, e)$ computed according to (18) is negative, the rectangular region occupied by $\Omega(m|s, e)$ in Θ is no longer "concentric" at the origin, which disallows the corresponding query (22) to be solved using orthogonal range search. For such $\Omega(m|s, e)$, we simply set $r_\ell(m, s, e) = 0$, and extend the angular range $[\theta_\ell(m, s, e), \theta_h(m, s, e)]$ to cover the full range $[0, 2\pi]$, to yield an expanded rectangular region. The resulting region includes the original region, as $r_\ell(m, s, e) \geq -d - \epsilon$. Thus the answer from (22) allows $\widehat{Q}_{polar}$ to remain a valid upper bound. Since these boundary issues occur mainly near the root of the BnB search tree, the loosening of the bound does not significantly impact the optimisation efficiency in practice.

5 Results

Several experiments were conducted to empirically test and benchmark the proposed algorithm compared to the original method of Breuel [2003]. The main BnB algorithm for both methods (henceforth referred to as, *Breuel* and *Polar*) were implemented in Matlab. For objective function and upper bound evaluations the relevant computations were implemented using Mex C++. Note that since Breuel and Polar optimise the same objective function and both are globally optimal, a direct comparison of their computational efficiency is possible.

5.1 Performance metrics

Two performance metrics were used for the comparison. The first is the runtime of the algorithms, as recorded on a standard machine with Intel Core i7-4770 3.40 GHz processor. The runtime difference gives arguably a more "practical" assessment of the benefits of the proposed method. The second performance metric is the iteration count of BnB for the respective bounding functions. Specifically, this is the number of nodes in the BnB search tree which had to be expanded before the globally optimal solution was discovered. This metric provides an insight into the relative tightness of the upper bound functions, which directly influences their ability to prune the search space.

5.2 Synthetic data tests

The first experiments conducted involved creating synthetic test problems. Each method was tested with several randomly generated problems in order to compare their performance. In order to generate the problems, first a random set of points, M and B, would be created. Next a random transformation would be generated consisting of a translation and rotation. The set of points would then be transformed by this transformation, producing a second set of points which is related to the initial set by the transformation. Then noise would be introduced by shifting the points in the second set by a random translation. Finally in order to represent outliers, a proportion of both point sets, M and B, would be removed and replaced with random points. These two point sets would then be passed to the matching algorithms for it to find the transformation to best match the point sets.

The synthetic experiments were run with one thousand points in each image set, over an area of 500 by 500 units. The noise added to the problem was uniform in both x and y in the range of 0 to 2 units. To correspond with this level of noise the threshold ϵ for the algorithms was set to 3. The algorithms were tested with differing levels of outliers, and for every value tested five trial problems were run, which were averaged together. The runtime data collected can be viewed in Figure 7 and the iteration data can be viewed in Figure 8.

It can be seen that for both metrics that the polar algorithm proposed outperforms the upper bound used by Breuel. While both algorithms have a similar performance for when there are no outliers, the difference between them increases as the problem becomes harder. The ability to handle more noisy problem instances can be important if this approach is to be used in real world applications where such instances would be expected.

Image 1	Image 2	Breuel	Polar
1	2	196.21	108.59
2	3	500.35	121.94
3	4	237.21	147.68
4	5	745.2	156.03
5	6	262.02	65.912
6	7	356.4	168.63
7	8	244.81	81.328
8	9	255.9	140.97
9	10	4777.2	1367.7
10	11	128.68	41.979
11	12	1067.9	469.08
12	13	505.81	222.34
13	14	130.72	104.24
14	15	10329	2783.8
15	16	168.31	199.71
16	17	209.62	86.657
17	18	2301.7	655.49
18	19	302.39	277.72
19	20	257.56	120.89

Table 1: The runtime data generated when running the Breuel and Polar algorithm on the space images.

5.3 Star field images

The algorithms were also tested using real star field images. This allows the algorithms to be analysed to see how they perform on a more realistic setting. An example of a pair of images which were tested can be viewed in Figure 1. These single frame images were taken by a camera-equipped telescope pointed towards orbit and rotated, taking the images at intervals a few seconds apart. First a threshold was passed over the image to transform it into a binary image. All of the connected components in the image were then identified and all of the centroids of these connected components formed the point set for the image. For this process a threshold value of $\frac{50}{255}$ was chosen because it created point sets of approximately 1000 points. Note that this processing was specifically chosen for these images for testing, and that more robust methods would be utilised which are directly suited for the specific task being accomplished.

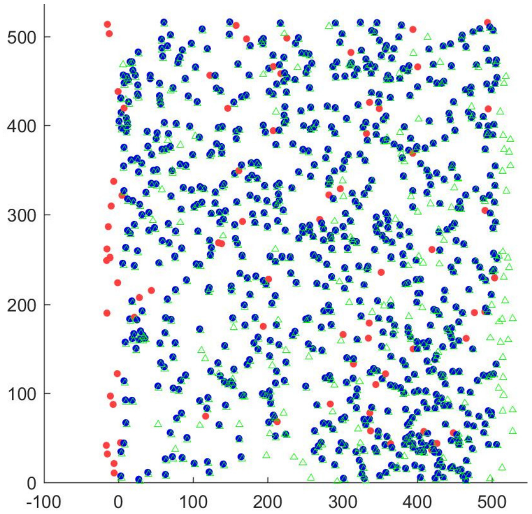


Figure 9: The solution when matching images one and two (best viewed in closeup). The green triangles are the points extracted from image two. The circles are the points extracted from image one after they have been transformed by the solution found. Blue means that it is considered a match while red indicates an outlier.

When finding the transformation between two images the point sets for both would be extracted and then would be passed to the branch and bound algorithm in order to find the correlation between them. The algorithms were tested on 19 image pairs and the results can be viewed in Table 5.2. It can be seen that the polar algorithm completes in less time than Breuel, similar to the synthetic problems. A solution that was found for one of the image pairs is shown in Figure 9.

6 Conclusions

This research has explored the application of utilising an optimal branch and bound algorithm for satellites. Keeping track of the position and motion of a satellite is an important task which is required in order to ensure that missions are carried out in an effective manner. It is believed that geometric matching can be used with images taken of space in order to determine the relative motion of the satellite. By assuming that stars lie on a plane infinitely far away, they should be able to be matched successfully, allowing the camera transformation to be determined efficiently.

Branch and bound provides a method which can be used to optimally solve this matching problem. The main issue preventing its use is that the algorithm may not be computationally efficient enough for the application. It has been proposed that by changing the way that the upper bound is calculated, that the solution space can be more efficiently searched. The next step in this research would be to investigate the suitability of the proposed solution in practical settings to obtain specific information on the algorithm's performance in realistic scenarios.

References

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92:1–31, 2011.

Jean-Charles Bazin, Yongduek Seo, and Marc Pollefeys. Globally Optimal Consensus Set Maximization through Rotation Search. In *Computer Vision ACCV 2012*, pages 539–551. Springer Berlin Heidelberg, 2013.

E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astronomy & Astrophysics Supplement*, 117:393–404, 1996.

P. J. Besl and N. D. MacKay. A method for registration of 3-D shapes. *IEEE TPAMI*, 14(2):239–356, 1992.

M. M. Birnbaum. Spacecraft attitude control using star field trackers. *Acta Astronautica*, 39(9–12):763–773, 1996.

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

A lvaro Joaquin Parra Bustos, Tat Jun Chin, and David Suter. Fast rotation search with stereographic projections for 3D registration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3930–3937, 2014.

Tat-Jun Chin, Álvaro Parra Bustos, Michael S. Brown, and David Suter. Fast rotation search for real-time interactive point cloud registration. In *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '14*, pages 55–62, New York, New York, USA, 2014. ACM Press.

Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry*. Springer Berlin Heidelberg, Berlin, Heidelberg, third edition, 2008.

Giancarmine Fasano, Giancarlo Rufino, Domenico Accardo, and Michele Grassi. Satellite angular velocity estimation based on star images and optical flow techniques. *Sensors (Switzerland)*, 13(10):12771–12793, 2013.

O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV*, 1992.

E. Grimson. *Object recognition by computer*. MIT Press, 1990.

Richard I. Hartley and Fredrik Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009.

B. K. P. Horn and B. G. K. Schunc. Determining optical flow. *Artif. Intell.*, 17:185–203, 1981.

R. Mur-Atal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-Bound Methods for Euclidean Registration Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.

S. Rawashdeh, J. E. Lumpp, J. Barrington-Brown, and M. Pastena. A stellar gyroscope for small satellite attitude determination. In *26th AIAA/USU Conference on Small Satellites*, 2012.

Vincent Riot, Willem de Vries, Lance Simms, Brian Bauman, Darrell Carter, Don Phillion, and Scot Olivier. The Space-based Telescopes for Actionable Refinement of Ephemeris (STARE) mission. *Proceedings of the AIAA/USU Conference on Small Satelites*, (Technical Session XI: Around the Corner), 2013.

Frank Schnitzer, Klaus Janschek, and Georg Willich. Experimental results for image-based geometrical reconstruction for spacecraft Rendezvous navigation with unknown and uncooperative target spacecraft. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5040–5045, 2012.

S. Se, D. G. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *ICRA*, 2001.

B. J. Spratling and D. Mortari. A survey on star identification algorithms. *Algorithms*, 2:93–107, 2009.

Fuyuto Terui, Heihachiro Kamimura, and ShiN'Ichiro Nishida. Motion estimation to a failed satellite on orbit using stereo vision and 3D model matching. In *9th International Conference on Control, Automation, Robotics and Vision, 2006, ICARCV '06*, 2006.

Dylan Thomas, Scott Kelly, and Jonathan Black. A monocular SLAM method for satellite proximity operations. In *2016 American Control Conference (ACC)*, pages 4035–4040. IEEE, jul 2016.

Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.