

Directions of Egomotion from Antipodal Points

John Lim^{1,2}

Nick Barnes^{1,2}

¹Department of Information Engineering, RSISE
Australian National University

john.lim@rsise.anu.edu.au

²NICTA
Canberra, Australia

Nick.Barnes@nicta.com.au

Abstract

We present a novel geometrical constraint on the egomotion of a single, moving camera. Using a camera with a large field-of-view (FOV), the optical flow measured at a single pair of antipodal points on the image sphere constrains the set of all possible camera motion directions to a subset region. By considering the flow at many such antipodal point pairs, it is shown that the intersection of all subset regions arising from each pair yields an estimate on the directions of motion. These antipodal point constraints rely on the geometrical properties of using a spherical representation of the image as well as the larger information content available from a large FOV. An algorithm using these constraints was implemented and tested on both simulated and real images. Results show comparable performance to the state of the art in the presence of noise and outliers whilst processing in constant time.

1. Introduction

Egomotion estimation from the image motion perceived by a single, moving camera has been extensively studied in the case of cameras with planar images, such as that found in traditional cameras. However, with the introduction of sensors such as catadioptric devices and fish-eye lenses, an extremely large field-of-view (FOV) can be obtained. A suitable space to represent such images is the *image sphere* and we wish to explore the geometry of this space and exploit its properties for egomotion estimation.

We focus on solutions using image correspondences or optical flow. The method developed uses only *directions* of flow (not magnitudes) and correspondingly, only *directions* of motion are found. Later, these can be re-substituted into the equations to find magnitude of the rotation angle. Scene depth is unknown and only rigid body motion occurs.

Whilst the equations governing motion and image projection are fundamentally nonlinear, a linear formulation is

possible via epipolar geometry. The recovery of the fundamental matrix (and thus motion) is possible with a minimum of 8 point correspondences [18, 8], or just 5, for calibrated cameras [15, 23]. The outputs may then be refined via local iterative techniques such as bundle adjustment [28]. Whilst these reflect the current state of the art, a wide range of alternatives exist, including linear subspace methods [11], non-linear optimization [5, 24], qualitative egomotion methods [6, 25], approaches using scene depth positivity [4], fundamental matrix from flow [12]; the approaches of [9], [21], [13], [10], [3], [26], [20] and many, many more.

We investigate a new approach that exploits the larger information content inherently present in images with large FOV and the geometry of representing the image on a sphere. Stability analyses by [7] suggest that a large FOV on a spherical image may be optimal for the recovery of self-motion. Furthermore, [4] suggests that two different rigid motions cannot give rise to the same motion/optical flow fields when the entire viewsphere is considered whereas such confusions may occur when a smaller FOV is used. We are also inspired by eye design in biology, where flying creatures such as insects have compound eyes that give them vision over nearly the entire viewsphere.

Prior work exploiting the large FOV of spherical images [22, 6, 25] often perform a 2D search. Our method is much more efficient compared to searching. Our approach is most similar to the method of [17], which studies in detail a constraint observed by [27] as a special case of linear subspace methods [11, 27]. This method takes the vector sum of optical flow at antipodes to constrain translation. The main difference with our method is that the constraint of [17, 27] recovers only translation. Rotation is later found by re-substituting translation, but this propagates errors in the translation estimate through to rotation. Our method directly recovers both the direction of translation (DOT) and axis of rotation (AOR) with the same constraints. Both the DOT and AOR can be recovered in parallel. Also, no summation of flow occurs and only directions of flow are used.

We discuss the antipodal point geometrical constraint in

Section 2 before presenting a complete algorithm using the constraint (Section 3). The algorithm is a ‘non-numerical’ approach in that the constraints are obtained without numerically solving equations. In fact, no mathematical operations more complex than that required for voting will be performed. Furthermore, run-time is constant with increasing noise and outliers, and that, together with the naturally parallelizable nature of the algorithm gives it potential for real-time egomotion estimation. Finally, the algorithm is tested and results are discussed in Section 4.

1.1. Background

For a spherical imaging surface, the optical flow equation is (refer Figure 1a):

$$\begin{aligned} \dot{\mathbf{r}} &= f_{tr}(\mathbf{r}) + f_{rot}(\mathbf{r}) \\ &= \frac{1}{|\mathbf{R}|} ((\mathbf{t} \cdot \mathbf{r})\mathbf{r} - \mathbf{t}) - \mathbf{w} \times \mathbf{r} \end{aligned} \quad (1)$$

where the flow at an image point \mathbf{r} corresponding to scene depth $|\mathbf{R}|$ is given by $\dot{\mathbf{r}}$ when the camera translates in direction \mathbf{t} whilst simultaneously rotating with rotation \mathbf{w} [4]. This equation does not model noise. Equation 1 consists of two terms - the term due to translational motion, $f_{tr}(\mathbf{r})$ and that of rotational motion, $f_{rot}(\mathbf{r})$.

Equation 1 has seven unknowns - the scene depth, three unknowns for translation and another three for rotation. It is well-known that the translation can only be recovered up to a scale [8] and we can rearrange the equation such that we have six unknowns ($|\frac{\mathbf{t}}{\mathbf{R}}|$ is consolidated as one unknown). We wish to recover five of these unknowns - the DOT, the AOR and the magnitude of rotation. However, a straightforward least squares solution is impossible since scene depth, \mathbf{R} , is a function of \mathbf{r} and the system of equations is under-constrained. A standard method for overcoming this is to introduce an additional constraint via epipolar geometry.

We suggest an alternative approach using constraints on the directions of motion arising from the spherical geometry available from images with large FOV to disentangle the motion parameters of Equation 1.

2. Directions of Motion from Antipodal Point Constraints

Let us consider two antipodal points \mathbf{r}_1 and \mathbf{r}_2 on the image sphere. There exist infinitely many great circles that will pass through both points and we consider one of them, the circle C . The flow vector at a point \mathbf{r} inhabits the tangent plane to the image sphere at that point. Let us define a vector \mathbf{c} at \mathbf{r} that is tangent to the image sphere and that lies on the plane of the great circle C (refer Figure 1 (b)). Therefore, \mathbf{c} is both tangent to great circle C and also tangent to the image sphere. It lies on the same tangent plane

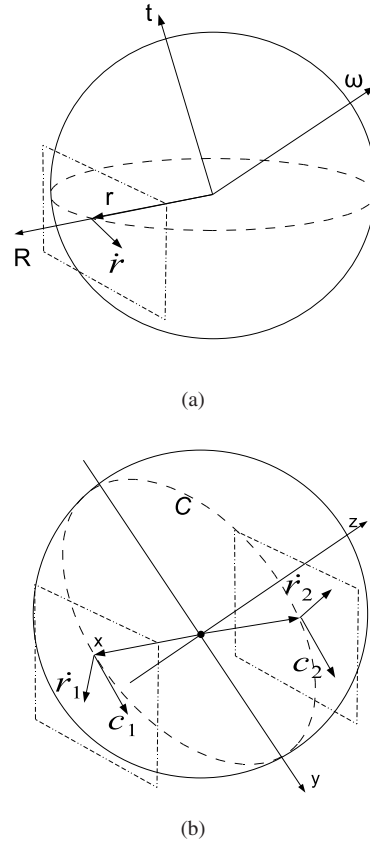


Figure 1. (a) Optical flow on the image sphere. (b) \mathbf{r}_1 , \mathbf{c}_1 and \mathbf{r}_2 , \mathbf{c}_2 lie on two parallel tangent planes if $\mathbf{r}_2 = -\mathbf{r}_1$ (i.e. they are antipodal).

as the flow vector $\dot{\mathbf{r}}$ at that point \mathbf{r} . Let the projection of the flow vector $\dot{\mathbf{r}}$ onto the vector \mathbf{c} be termed as *projected flow* and denoted as $proj(\dot{\mathbf{r}})$ where $proj(\dot{\mathbf{r}}) = \dot{\mathbf{r}} \cdot \mathbf{c}$.

From $proj(\dot{\mathbf{r}}_1)$ and $proj(\dot{\mathbf{r}}_2)$, we obtain constraints on egomotion as per Theorem 1 (illustrated by Figure 2):

Theorem 1 *If the projected flow vectors, $proj(\dot{\mathbf{r}}_1)$ and $proj(\dot{\mathbf{r}}_2)$, at two antipodal points \mathbf{r}_1 and \mathbf{r}_2 are in the same direction (same sign), then the DOT is constrained to lie in a hemisphere bounded by a great circle C_{bound} that passes through \mathbf{r}_1 and \mathbf{r}_2 and is perpendicular to the great circle C . The hemisphere is such that $proj(\dot{\mathbf{r}}_1)$ and $proj(\dot{\mathbf{r}}_2)$ point away from it.*

Likewise, if $proj(\dot{\mathbf{r}}_1)$ and $proj(\dot{\mathbf{r}}_2)$ are in opposite directions (opposite signs), the direction of the AOR is constrained to lie in a hemisphere bounded by C . Which side of C this hemisphere lies on is determined by the cross product in the rotational component of flow, $-\mathbf{w} \times \mathbf{r}$.

Without loss of generality, let us choose a Cartesian coordinate frame such that our antipodal points \mathbf{r}_1 and \mathbf{r}_2 are at $(1, 0, 0)$ and $(-1, 0, 0)$ and such that the great circle C lies on the x-y plane. Expanding Equation 1, substituting

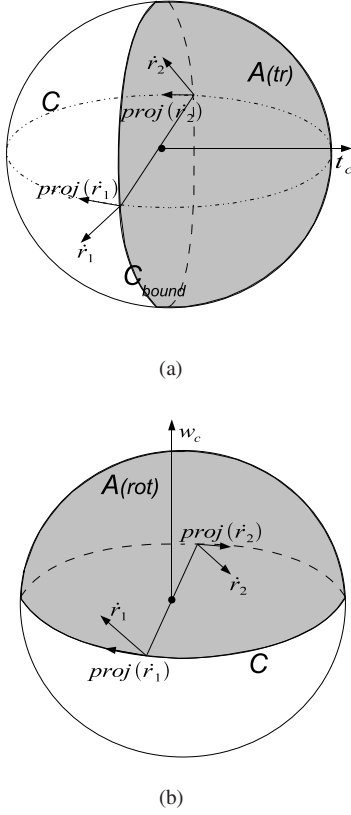


Figure 2. (a) If t_c exists, DOT is constrained to lie in the shaded hemisphere $A(tr)$. (b) The AOR is constrained to lie in the hemisphere $A(rot)$. Since $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ are in a clockwise configuration, the direction of w_c is upward due to the cross product, $-\mathbf{w} \times \mathbf{r}$ (the left hand grip rule).

for \mathbf{r}_1 and \mathbf{r}_2 and projecting onto \mathbf{c}_1 and \mathbf{c}_2 which in this case are both simply $(0, 1, 0)$, the projected flow is:

$$\begin{aligned} proj(\mathbf{r}_1) &= - \left| \frac{\mathbf{t}}{\mathbf{R}_1} \right| t_y - w_z \\ proj(\mathbf{r}_2) &= - \left| \frac{\mathbf{t}}{\mathbf{R}_2} \right| t_y + w_z \end{aligned} \quad (2)$$

where $|\mathbf{R}_1|$ and $|\mathbf{R}_2|$ are the depths of the scene at \mathbf{r}_1 and \mathbf{r}_2 and $|\mathbf{t}|$ is the magnitude of translation. The unknowns are reduced to t_y , and w_z , which are mutually orthogonal.

Here, we are interested in the directions (i.e. the sign) of $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$. Equation 2 shows that the projected flow consists of two components, the translational component, $\left| \frac{\mathbf{t}}{\mathbf{R}_1} \right| t_y$, and the rotational component, w_z . We find that the translational components of both $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ act in the same direction because they have the same sign whilst the rotational components have opposite signs and therefore, act in opposing directions.

Logically, we can conclude that if we observed $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ to have the same sign (same direction), then

some translational component of projected flow, $\left| \frac{\mathbf{t}}{\mathbf{R}_1} \right| t_y$, must exist. This is because the rotational component of projected flow, w_z , acts in opposite directions at \mathbf{r}_1 and \mathbf{r}_2 and, on its own, could never give rise to what we observe.

Similarly, if we observed $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ to be in opposite directions (opposite signs), then there must exist some rotational component of flow that is causing it as the translational component could not, on its own, give rise to the observed signs. These conclusions hold true for any general pair of antipodal points since the coordinate frame can always be chosen such that it is aligned conveniently in the above manner, leading us to the statements of Lemma 1:

Lemma 1 *If the projected flow vectors, $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$, at two antipodal points \mathbf{r}_1 and \mathbf{r}_2 are in the same direction (same sign), there must exist a component of translation, t_c , that is parallel and in the opposite direction to $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ (Figure 2a). Likewise, if $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$ are in opposite directions (opposite signs), a component of rotation, w_c , that is normal to the plane of the great circle C must exist. (Figure 2b)*

Furthermore, for t_c to be in the opposite direction to $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$, the DOT, \mathbf{t} , is constrained to lie in a hemispherical region, $A(tr)$, shown in Figure 2. If \mathbf{t} was outside $A(tr)$, then t_c would be parallel to but in the same direction as the projected flows $proj(\mathbf{r}_1)$ and $proj(\mathbf{r}_2)$; thus contradicting Lemma 1. This hemisphere, $A(tr)$ is bounded by the great circle, C_{bound} , obtained by the intersection of a plane with normal vector t_c with the image sphere. Similarly, for the AOR to have a component in the direction w_c , it must lie in a similar hemispherical region $A(rot)$.

2.1. The constraints from all great circles passing through a pair of antipodal points

In Theorem 1 we saw that a great circle passing through a pair of antipodal points will yield a hemispherical region, $A(tr)$, that constrains the DOT and another region, $A(rot)$, that constrains the AOR. Since there exist infinitely many great circles that will pass through these two antipodal points, an infinite set of hemispherical constraint regions, $A_1, A_2, \dots, A_i, \dots$ will result. As the motion must simultaneously satisfy all these constraints, the direction of motion must lie within the intersection of all these hemispherical constraint regions, $A_{res} = A_1 \cap A_2 \cap \dots \cap A_i \dots$.

Let C_i be the i^{th} great circle in the infinite set of great circles passing through \mathbf{r}_1 and \mathbf{r}_2 . Associated with every C_i , is the vector \mathbf{c}_i that is tangent to the sphere and lying in the plane of the circle C_i as defined earlier. The intersection of constraints can then be described in Theorem 2 and illustrated by Figure 3:

Theorem 2 *The optical flow \mathbf{r}_1 and \mathbf{r}_2 observed at any two antipodal points \mathbf{r}_1 and \mathbf{r}_2 on the image sphere will give rise to a constraint on the DOT and a constraint on the AOR.*

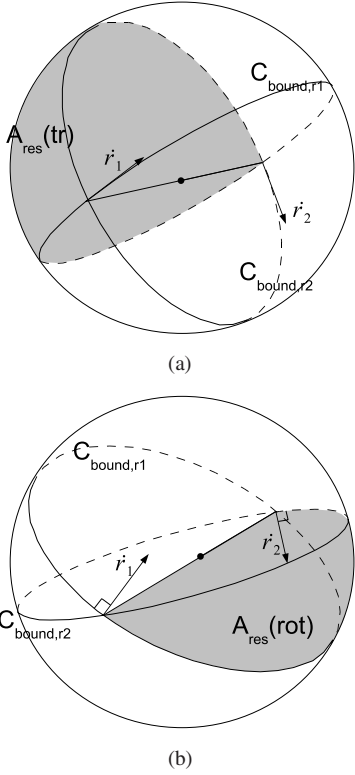


Figure 3. (a) Shaded lune $A_{res}(tr)$ is resultant constraint region on translation (b) $A_{res}(rot)$ is resultant constraint region on rotation

The DOT lies in a region on the sphere bounded by great circles $C_{bound,r1}$ and $C_{bound,r2}$ that pass through \mathbf{r}_1 and \mathbf{r}_2 and are such that the associated \mathbf{c}_i vectors are $\mathbf{c}_{bound,r1} = \hat{\mathbf{r}}_1$ and $\mathbf{c}_{bound,r2} = \hat{\mathbf{r}}_2$.

The direction of the AOR lies in a region on the sphere bounded by great circles $C_{bound,r1}$ and $C_{bound,r2}$ such that $\mathbf{c}_{bound,r1}$ is perpendicular to $\hat{\mathbf{r}}_1$ and $\mathbf{c}_{bound,r2}$ is perpendicular to $\hat{\mathbf{r}}_2$.

Theorem 2 can be shown using a geometrical argument. For simplicity, we consider only the vectors inhabiting the two tangent planes to the sphere at \mathbf{r}_1 and \mathbf{r}_2 . They are the flow vectors and the \mathbf{c}_i vectors that are associated with the great circles. As the two tangent planes are parallel in \mathbb{R}^3 , we can represent everything in just one plane, as is done in Figure 4. Every great circle passing through a point \mathbf{r} is characterized completely by \mathbf{r} and \mathbf{c}_i , so in Figure 4, great circles C_i and vectors \mathbf{c}_i are synonymous.

Recall that $proj_i(\hat{\mathbf{r}}) = \mathbf{c}_i \cdot \hat{\mathbf{r}}$. So, for the case of translation, only the great circles with \mathbf{c}_i satisfying the following condition will give constraints on translation:

$$sgn(proj(\hat{\mathbf{r}}_1)) = sgn(proj(\hat{\mathbf{r}}_2)) > 0 \quad (3)$$

That is to say, the acute angles between $\hat{\mathbf{r}}_1$ and \mathbf{c}_i and between $\hat{\mathbf{r}}_2$ and \mathbf{c}_i must both be less than $\pi/2$. Such \mathbf{c}_i vec-

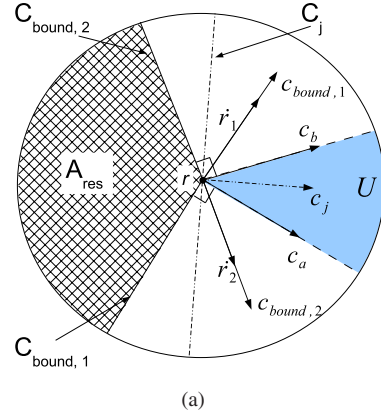


Figure 4. The two tangent planes containing $\hat{\mathbf{r}}_1$, $\hat{\mathbf{r}}_2$, \mathbf{c}_1 and \mathbf{c}_2 are parallel to the plane of the page. Intersection of all hemispherical constraints on translation is the lune A_{res}

tors lie in the shaded region U in Figure 4. Consider the two great circles C_a and C_b with \mathbf{c}_i 's exactly perpendicular to $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$. Let their \mathbf{c}_i vectors be \mathbf{c}_a and \mathbf{c}_b . From Theorem 1, we obtain two hemispherical regions constraining translation that are bounded by C_{bound1} and C_{bound2} which are again perpendicular to C_a and C_b . This means that the \mathbf{c}_i vectors for C_{bound1} and C_{bound2} are actually $\mathbf{c}_{bound,1} = \hat{\mathbf{r}}_1$ and $\mathbf{c}_{bound,2} = \hat{\mathbf{r}}_2$. The DOT must lie within the intersection of these two hemispherical regions - the region A_{res} of Figure 4.

In fact, C_{bound1} and C_{bound2} are the boundary cases. All other great circles that give valid constraints on translation - those with \mathbf{c}_i 's lying between \mathbf{c}_a and \mathbf{c}_b - will give rise to hemispherical regions that contain A_{res} . For example, see the great circle represented by the vector \mathbf{c}_j in Figure 4. It gives rise to the hemispherical region bounded by the great circle C_j and that hemisphere contains A_{res} . So the A_{res} bounded by C_{bound1} and C_{bound2} is the smallest such region that satisfies all the constraints. In geometry, these wedge-shaped constraint regions are termed *lunes*.

To show the case of constraining rotation, simply reiterate the previous argument, but now let $\mathbf{c}_{bound,1}$ be perpendicular to $\hat{\mathbf{r}}_1$ and $\mathbf{c}_{bound,2}$ be perpendicular to $\hat{\mathbf{r}}_2$.

2.2. Motion estimation with constraints from N antipodal point pairs

Thus far, we have shown that once the direction of flow $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$ at antipodal points are computed, finding the boundaries of the lune shaped constraint regions is trivial. N pairs of antipodal points on the image sphere will give rise to N constraint regions for the DOT and N regions for the AOR. The intersection of these will give some estimate of the directions of motion. By sampling sufficiently many antipodal points from all over the image sphere, this will

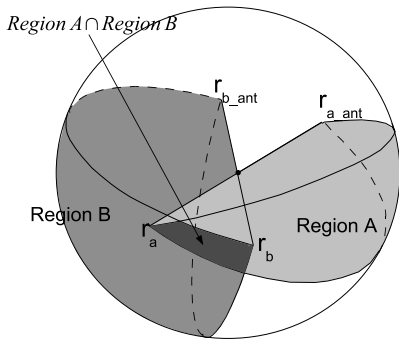


Figure 5. Point r_a and its antipode $r_{a,ant}$ gave rise to constraint region A. Then, point r_b was chosen such that it was inside region A. r_b and $r_{b,ant}$ gave rise to region B. Region A \cap region B is guaranteed to be convex and smaller in size than regions A and B.

give a global estimate of motion.

However, does this method converge to the true directions of motion as N becomes large? We argue that if the next antipodal point pair is always chosen such that one of the points lie inside the current intersection of all previously obtained constraint regions, the intersection of the new constraint with the current intersection region will be a convex region smaller than the current region (Figure 5).

Lunes are convex regions. Since the intersections of convex regions are themselves convex, the intersection of the lunes must also be convex. Let the region of intersection from all previous lunes be A and the constraint region from a newly chosen antipodal point pair be B . One of the points in this pair is P , so $P \in B$. Let us pick P subject to two conditions. Firstly, it is inside region A , that is, $P \in A$ and therefore $P \in (B \cap A)$. Secondly, P is not on the boundary of A , that is, there is an open ball, C , around P such that $C \subset A$. Since P , the antipodal point, lies on the boundary of B there exists some point Q in the neighborhood of P such that $Q \in A$ but $Q \notin B$, so $Q \notin (B \cap A)$. Therefore, $(A \cap B) \subset A$, i.e. the intersection of the original constraint region and the lune is smaller than the original region.

Using such a scheme to choose the antipodal point pairs, as N becomes large, the constraint region will converge towards the true motion direction. Thus, for the noise-free case of dense optical flow, we have a convergent algorithm.

In practice, we need to incorporate Hough-like voting, simulated annealing or other such methods to account for noise, outliers and independently moving objects that would give erroneous constraint regions. Furthermore, once the solution has been constrained to some region smaller than a hemisphere, a gnomonic projection can map great circles on the sphere onto straight lines on a plane and intersecting lunes onto polygons. This reduces the convex problem into a linear one which can be efficiently solved by robust linear programming methods.

Once the DOT and AOR have been recovered, estimating the magnitude of rotation is easy and several methods are possible. Here is one suggestion: $\dot{\mathbf{r}} = f_{tr}(\mathbf{r}) + f_{rot}(\mathbf{r})$ (Equation 1) is a vector sum triangle. We now know the directions of $f_{tr}(\mathbf{r})$ and $f_{rot}(\mathbf{r})$, that is, we know the angles between the sides of that triangle. Thus, magnitude of $f_{rot}(\mathbf{r})$ can be found by the sine rule of triangles. Repeating for several points on the image, an overconstrained system of linear equations is obtained and a least squares solution to rotation angle may be found.

3. Algorithm and Implementation

The quickly converging method earlier described could be used if speed was critical, but here, we use a simple but very robust vote-based algorithm. N antipodal point pairs were randomly picked on the image sphere. The constraint regions from each pair contributed to votes in a voting table in a coarse-to-fine scheme. The centre of mass of the bin(s) with the highest votes gives the Maximum Likelihood estimate of motion direction. We chose a vote-based approach because it is highly robust to noise and outliers in the flow. Its disadvantage is that accuracy is limited by voting bin resolution. Errors were measured based on the angle between the recovered motion direction and the true motion.

Matlab simulations: We compared performance with the 5-point algorithm of [15] (code supplied by author) and the other existing antipodal point method of [17]. The camera simultaneously underwent rotation and translation randomly generated from the range of all possible directions (azimuth $\in [0, 360^\circ]$, elevation $\in [-90^\circ, 90^\circ]$). Baseline was 2 units and rotation angle, 0.2 radians. 500 pairs of random antipodal scene points were uniformly distributed in all directions with depth ranging from 10 to 15 units. Results were averaged over 100 trials. 5-point used point match inputs, whilst flow for antipodal point algorithms was generated from Equation 1. To simulate noise, the ray of a matched point was perturbed by some angle modeled as Gaussian noise with standard deviation σ . Outliers were simulated by randomly replacing matches with errors.

5-point was implemented in a RANSAC framework from [14]. Sampling was adaptive with probability $p = 0.99$ and Sampson distance threshold of 0.01 (see [8] for details). We chose 5-point with RANSAC as a comparison since it is well-known, widely used and code is widely available. For the baselines used in our experiments, we found 5-point to work reasonably well (sub-pixel accuracy for zero noise and $< 1^\circ$ error for small noise).

Translation estimates were also compared with the existing antipodal point method of [17] (special case of linear subspace methods [11, 27], see Section 1). Briefly, the method sums antipodal flow to constrain DOT on a great circle. It does not find rotation. For better comparison with our work, we adapted [17] into a voting scheme which votes

along great circles (our method votes on lunes). In this implementation, both our method and that of [17] use all available flow (no sampling).

Real videos: Real sequences were captured with a Ladybug camera [2] which returns 5 images positioned in a ring. Scale Invariant Feature Transform (SIFT) feature matching (code from [19]) was used as input to 5-point and also used to calculate sparse optic flow for our method. In the pure translation case, the camera translated along the ground with baseline 2cm per frame in the direction of the x-axis, which is parallel to the ground plane. In the simultaneously translating and rotating case, it also rotated 5° per frame about the z-axis, which is perpendicular to the ground plane. Our algorithm ran on consecutive frames whilst for 5-point, we skipped a frame for every estimate, to afford it larger baselines and rotation. Two points within 0.01 radians of being antipodal were considered antipodal. In the real images of cluttered indoor scenes, experiments using SIFT matches obtained flow at typically 300 to 800 of these near antipodal points. Increasing this margin finds more antipodes, but at the expense of accuracy. We also experimented with using dense flow calculated by iterative Lucas-Kanade in a pyramidal, multi-scale hierarchy (code from [1]) as input to our algorithm. With this method, antipodal flow is dense and more plentiful, but noisier than SIFT. Experiments randomly picked flow at 400 antipodes.

4. Results and Discussion

Outliers: Figures 6A and 6B show averaged errors for translation and rotation as outliers increase. Our method is labeled ‘our antipodal’ and the method of [17], which sums antipodal flow, is labeled ‘sum antipodal’.

Unsurprisingly, 5-point with RANSAC estimates translation most accurately for low noise since voting bin resolution limits our method in this implementation. Interestingly, both our method and [17] outperformed 5-point-RANSAC as outliers grew. Here, errors in RANSAC arise because some outliers fall within the model distance threshold used (0.01). Reducing RANSAC distance threshold improves performance; but in general, both antipodal methods tended to outperform RANSAC for very large outlier proportions. This is because antipodal methods require only 2 points for a constraint. If q is the probability of an inlier, then the probability of obtaining a good constraint is q^2 for antipodal methods and q^5 for 5-point. Therefore, antipodal methods are always more likely to obtain more good constraints.

However, 5-point with RANSAC outperformed our method in finding rotation. Compared to estimating translation, our method performed the same but 5-point was much more accurate in estimating rotation. This is probably because 5-point estimates rotation more easily than translation, as observed by [23]. The important point is that our method degrades at a *comparable rate* to 5-point for rota-

tion errors, implying similar robustness to increasing outliers in the two methods.

Gaussian Noise: Figures 6C and 6D show the translational and rotational errors under noise. All three methods degrade comparably for translation and both our method and 5-point show similar robustness for rotation estimation. Once again, for large noise, the trend is for antipodal methods to perform better than 5-point.

Timing: We have seen that our vote-based antipodal algorithm shows comparable robustness with 5-point and RANSAC. Figure 6E further demonstrates that our algorithm achieves this with *constant* processing time compared to RANSAC, which runs in approximately cubic time here as outliers increase. Our algorithm uses all available flow (although sampling is an option), yet it is still faster than 5-point with RANSAC which samples points adaptively according to $N = \log(1-p)/\log(1-(1-\epsilon)^s)$ (refer [8]). The numbers in 6E are obviously implementation dependent but the trend will remain the same. Our algorithm has minimal mathematical complexity (voting and some dot products). Furthermore, it is quite naturally parallelizable. Theoretically, some parallel machine could simultaneously cast all votes from all constraints in *one* time step - potentially making this an ‘instant’ time egomotion method.

Real sequences: Comparisons with measured ground truth and 5-point-RANSAC estimates show accurate and robust egomotion recovery for simultaneous translation and rotation sequences and for pure translation sequences. Errors for the former are summarized in Figures 6G and 6H. Experiments using SIFT inputs for our method (marked ‘SIFT’) showed little difference in performance when compared with 5-point-RANSAC. Average errors over the whole sequence range from 2° to 3° for both methods, with the average slightly lower for 5-point (1° lower for translation and 0.5° lower for rotation axis). Experiments with pyramidal Lucas-Kanade flow inputs (marked ‘LK’) for our algorithm showed that our method is robust to the considerably noisier inputs. Results were, on average, about 1 degree worse than those obtained using matched SIFT points as inputs. The pure translational sequence gave similar good results. Supplementary videos are included.

4.1. Discussion

Are antipodal points practical? Crucial to our algorithm is the availability of useful antipodal points. Useful means that the constraint arising from an antipodal pair should serve to further disambiguate the current constraint region. Also, the Ladybug multi-camera system used is not exactly single-viewpoint, leading to points being less than ideally antipodal. Nevertheless, the results demonstrate that with real images, flow can be found at sufficient numbers of antipodes, with sufficiently strong constraints on motion, such that accuracies comparable to 5-point are achievable.

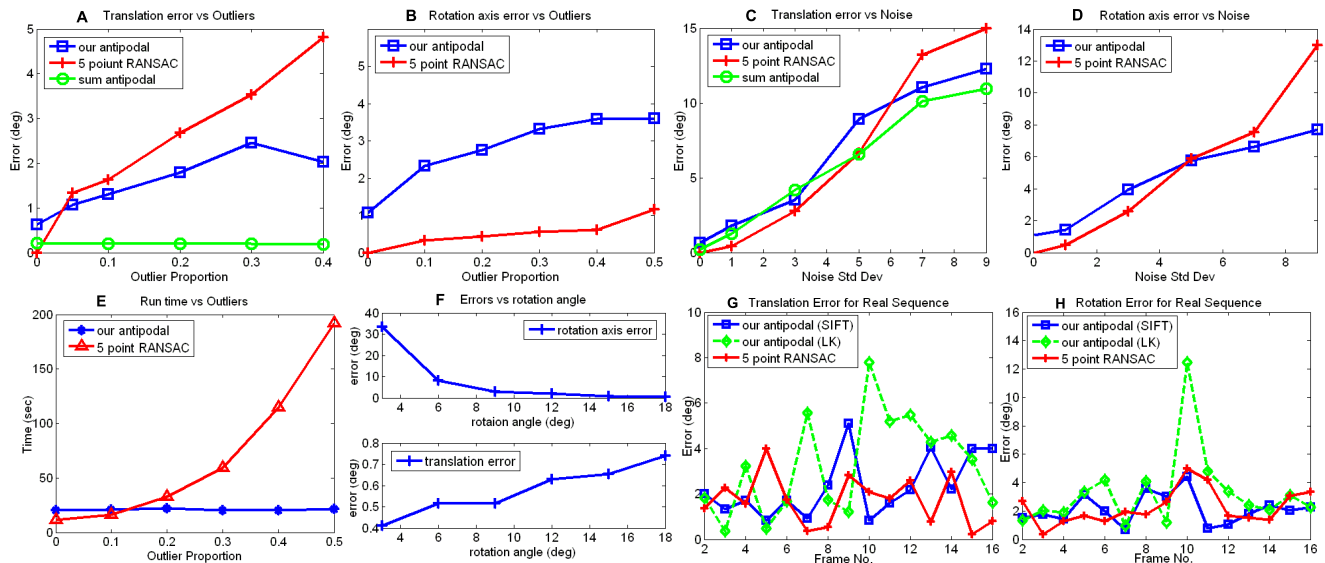


Figure 6. The method of [17] is labeled ‘sum antipodal’. (A-B) Translation and rotation axis errors in simulations with increasing outliers. (C-D) Translation and rotation axis errors with increasing Gaussian noise. (E) Run time comparison with 5 point RANSAC. (F) Errors as rotation angle increases. (G-H) Errors for real sequences. Videos attached.

Complex scenes. In general, the algorithm is suitable for feature rich environments. In environments such as unstructured, dynamic outdoor scenes with many features but also a multitude of moving shadows, fluttering leaves, moving objects and many other sources of noise and outliers, the simulations suggest that this algorithm would be at least as accurate as 5-point with RANSAC, while running much faster (in fact, in constant time). The results also suggest combining the method of [17] with our algorithm. Such a hybrid can find translation with [17] since it gives stronger constraints on translation and is very robust to outliers, whilst rotation is recovered with our method.

Fail-soft. Insufficient antipodal points does not break the algorithm - the motion only becomes more ambiguous. Voting returns a probability map (eg. Figure 7) where regions of higher votes indicate higher probability of the motion being in that direction. Ambiguity occurs when constraints fail to further disambiguate motion, and a patch of directions has tied votes. The true solution lies within that patch with maximum probability given the available antipodes. This is better than many other motion methods that return a nonsensical number when they fail.

Doesn't voting resolution limit accuracy? Voting is an approach that works well here and in other motion estimation methods (eg. [20]). However, alternative implementations are possible, such as linear programming, as suggested in Section 2.2. Nevertheless, the real image experiments demonstrate that under practical levels of noise and calibration errors, voting is comparable to exact methods like 5-point-RANSAC. For further refinement, methods like 5-

point or antipodal point can obtain initial estimates which are then refined by bundle adjustment [28]. Also, voting allows variable resolutions, which is suited to the multi-scale nature of real world tasks like robot navigation.

Field of View: As Equation 1 shows, flow is an entanglement of translational and rotational components. The flow in certain parts of the viewsphere are strongly influenced by translation or rotation. For example, if the camera rotates about the z-axis, flow near the equator (x-y plane) gives strong constraints on rotation. Also, if either translation or rotation is large relative to the other, it becomes harder to estimate the weaker motion and easier to find the dominant one. Figure 6F shows that for baseline of 2 units, as rotation angle varies from 3° to 18° , rotation becomes easier to estimate and translation harder to estimate.

Why use only directions of flow? [17, 27] use the magnitude information of antipodal flow and its constraint on translation is therefore stronger. However, by using only directions of flow, we are able to extend antipodal constraints to rotation as well. Other benefits include improved robustness since it is independent of noise in the magnitude of flow vectors - only noise in the direction of flow affects it. It is also theoretically interesting because our constraint regions (the lunes) arise immediately from the flow vectors without further mathematical processing, be it addition, multiplication or more complex operations.

5. Conclusion

We presented a geometric constraint on egomotion from directions of antipodal point flow. Simulated and real

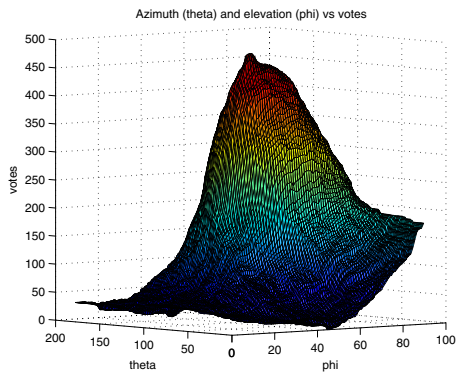


Figure 7. Voting on the sphere.

experiments show that this constraint works in practice, and that by combining many pairs of such points we are able to gain robust and accurate egomotion estimates. **Supplementary videos included** (also available at our website [16]).

Acknowledgements: We thank H. Li for providing assistance with his 5-point solver [15]. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

- [1] Intel Open Source Computer Vision Library, 2007. Available from: <<http://sourceforge.net/projects/opencv/>>.
- [2] Point grey research, 2007. <http://www.ptgrey.com/>.
- [3] A. Agrawal and R. Chellappa. Ego-motion estimation and 3d model refinement in scenes with varying illumination. In *IEEE Workshop on Motion and Video Computing, MOTIONS '05*, volume 2, pages 140–146, 2005.
- [4] T. Brodsky, C. Fermuller, and Y. Aloimonos. Directions of motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 26:5–24, 1998.
- [5] A. R. Bruss and B. K. Horn. Passive navigation. In *Computer Vision, Graphics and Image Processing*, volume 21, pages 3–20, 1983.
- [6] C. Fermuller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.
- [7] C. Fermuller and Y. Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *International Journal of Computer Vision*, 28:137–154, 1998.
- [8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [9] B. Horn and E. Weldon. Direct methods for recovering motion. *Int. Journal on Computer Vision*, 2(1):51–76, 1988.
- [10] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272, 1997.
- [11] A. Jepson and D. Heeger. Subspace methods for recovering rigid motion i: algorithm and implementation. *International Journal of Computer Vision*, 7(2), 1992.
- [12] K. Kanatani, Y. Shimizu, N. Ohta, M. Brooks, W. Chojnacki, and A. van den Hengel. Fundamental matrix from optical flow: optimal computation and reliability evaluation. *Journal of Electronic Imaging*, 9(2):194–202, April 2000.
- [13] J. Koenderink and A. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.
- [14] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia. Available from: <<http://www.csse.uwa.edu.au/~pk/research/matlabfns/>>.
- [15] H. Li and R. Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 630–633, 2006.
- [16] J. Lim, 2008. <http://users.rsise.anu.edu.au/~johnlim/>.
- [17] J. Lim and N. Barnes. Estimation of the epipole using optical flow at antipodal points. In *OMNIVIS*, 2007.
- [18] H. Longuet-Higgins. A computer algorithm for reconstruction of a scene from two projections. *Nature*, 293:133–135, 1981.
- [19] D. Lowe, 2007. <http://www.cs.ubc.ca/~lowe/keypoints/>.
- [20] A. Makadia, C. Geyer, S. Sastry, and K. Daniilidis. Radon-based structure from motion without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.
- [21] S. Negahdaripour and B. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.
- [22] R. Nelson and J. Aloimonos. Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.
- [23] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 756–770, 2004.
- [24] J. W. Roach and J. K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):55–62, 1980.
- [25] C. Silva and J. Santos-Victor. Direct egomotion estimation. In *Proc. 13th International Conference on Pattern Recognition*, volume 1, pages 702–706, 1996.
- [26] G. P. Stein and A. Shashua. Model-based brightness constraints: On direct estimation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):992–1015, 2000.
- [27] I. Thomas and E. Simoncelli. Linear structure from motion. Technical report, IRCS, University of Pennsylvania, 1994.
- [28] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms '99*, pages 298–372, 1999.