# Estimation of the Epipole using Optical Flow at Antipodal Points

John Lim[1,2]

Nick Barnes[1,2]

[1]Department of Information Engineering, RSISE
Australian National University
john.lim@rsise.anu.edu.au

[2]VISTA, NICTA
Canberra, Australia
Nick.Barnes@nicta.com.au

## Abstract

*This paper develops an algorithm for estimating the epipole or direction of translation of a moving monocular observer. To this end, we use constraints arising from two points that are antipodal on the image sphere. The antipodal point condition is necessary for decoupling rotation from translation. One such pair of points constrains the epipole to lie on a plane, and using two pairs of points, we have two such planes. The intersection of these two planes gives an estimate of the epipole. This means we require image motion measurements at two pairs of antipodal points to obtain an estimate. Repeating this will yield a set of possible solutions and a variety of methods could be applied to obtain a robust and refined estimate from this set. One robust and simple method is chosen for illustrative purposes and results on real images are shown. With real sequences, results of below 2° error in the estimate of the epipole can be obtained. Since antipodal points on an image sphere are required, this algorithm must use some kind of omnidirectional or large field-of-view (FOV) sensor.*

## 1. Introduction

A monocular observer moving in a scene of unknown depth undergoes rigid motion that is a combination of translational and rotational motions. The task of estimating these motions is a classical problem and a vast number of self-motion estimation methods that are based on image motion measurements or optical flow are available. These include the well-known epipolar geometry formulation of [15, 6] which leads to algorithms such as the 8-point [5], 6-point [13] and 5-point algorithms [14, 21]; methods involving nonlinear optimization [3, 23]; qualitative search methods [4, 25] and the recovery of a 'flow fundamental matrix' from optical flow [11]. Other approaches of note include [10], [8], [19], [12] and many others, which can be found in reviews such as [27] and [9].

However, most approaches assume the use of a planar image with a limited field-of-view (FOV) such as that found in traditional cameras. Although omnidirectional cameras have become widely available of late, few self-motion estimation algorithms actually exploit the large FOV property explicitly in order to aid or simplify the task. [20] and [4] are examples of some such prior work. However, the method proposed here is significantly more efficient compared to those algorithms, which solve the problem via a search.

Here, we present a method for estimating the direction of translation or the epipole of the camera based on the geometrical properties of points that are antipodal on the image sphere. The image sphere is simply a more natural surface for representing the images of omnidirectional cameras. From the optical flow at two antipodal points we are able to constrain the epipole to lie on a great circle on the image sphere. The rotational contribution to the measured optical flow can be geometrically eliminated. These constraints may then be used to estimate the location of the epipole. The method presented here does not directly estimate the rotational motion of the camera, but once direction of translation is found, rotation is not difficult to recover.

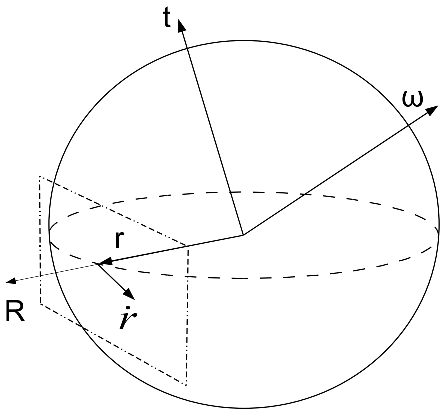
A somewhat similar antipodal point constraint was observed in [26]. However, significant differences in the theoretical derivation and in the resulting constraint exist between the method of [26] and the one presented here. Furthermore, as the authors of [26] noted, the lack of widely available omnidirectional sensors at that time (1994) meant that such a constraint was hardly of any practical use then. As a result, their constraint was merely observed as an equation and was not investigated further. The emergence of omnidirectional cameras as a popular tool in computer vision today warrants an investigation into methods that specifically exploit the large FOV of these sensors.

This paper presents a constraint that is somewhat simpler to derive and use compared to [26] and it succeeds in certain scene depth configurations where [26] fails. Both the constraint presented here and that of [26] may be thought of as special cases of the linear subspace methods investigated by [10]. A complete algorithm is provided based on the constraint and it is implemented and tested with simulations under noise, and on real images to show that it can robustly and accurately recover the epipole.

### 1.1. Background

For an image sphere, the equation relating the rigid motion of the camera with image motion is given in Equation 1 [2]. At an image point $\mathbf{r}$ (refer Figure 1), the optical flow $\dot{\mathbf{r}}$ resulting from the translational motion $\mathbf{t}$ and the rotational motion $\mathbf{w}$ is given by:

$$\dot{\mathbf{r}} = \frac{1}{\mid \mathbf{R}(\mathbf{r}) \mid}((\mathbf{t} \cdot \mathbf{r})\mathbf{r} - \mathbf{t}) - \mathbf{w} \times \mathbf{r} \tag{1}$$



(a)

Figure 1. For some translational motion, $\mathbf{t}$ and some rotation about the axis $\mathbf{w}$, the optical flow at point $\mathbf{r}$ on the image sphere is $\dot{\mathbf{r}}$ and it lies on the tangent plane to the sphere at that point.

The self-motion estimation task attempts to recover the translational and rotational motions, where it is well-known that the translation can only be recovered up to a scale [6]. A least squares solution is not possible since the scene depth, $\mathbf{R}$, is a function of $\mathbf{r}$ and the system of equations is under-constrained. Without any knowledge of depth, recovering the five unknown motion parameters is difficult.

In this work, we attempt to recover the direction of translation or the epipole. The epipole can be defined as the intersection of the line joining the two camera centres with the image sphere [6]. The second camera centre is related to the first camera centre via some translation, so the direction of translation and the epipole are equivalent.

Note that in this paper, we use the term ‘optical flow’ to refer to any measurement or approximation of image motion. This includes image velocities obtained from feature correspondences such as SIFT or Harris corners, as well as image velocity fields obtained using methods like Lucas-Kanade [17] or Horn-Schunk [7]. In the former, point correspondences are found and matched for two images and the velocity vector transforming one point to the other calculated. The method presented here works with both classes of image motion measurements.

## 2. Removing Rotation by Summing Flow at Antipodal Points

On the image sphere, the optical flow, $\dot{\mathbf{r}}_1$ and $\dot{\mathbf{r}}_2$, at two antipodal points, $\mathbf{r}_1$ and $\mathbf{r}_2$, can be written as:

$$\dot{\mathbf{r_1}} = \frac{1}{\mid \mathbf{R}(\mathbf{r_1}) \mid}((\mathbf{t} \cdot \mathbf{r_1})\mathbf{r_1} - \mathbf{t}) - \mathbf{w} \times \mathbf{r_1} \tag{2}$$

$$\begin{aligned} \dot{\mathbf{r_2}} &= \frac{1}{\mid \mathbf{R}(\mathbf{r_2}) \mid}((\mathbf{t} \cdot \mathbf{r_2})\mathbf{r_2} - \mathbf{t}) - \mathbf{w} \times \mathbf{r_2} \\ &= \frac{1}{\mid \mathbf{R}(-\mathbf{r_1}) \mid}((\mathbf{t} \cdot \mathbf{r_1})\mathbf{r_1} - \mathbf{t}) + \mathbf{w} \times \mathbf{r_1} \end{aligned} \tag{3}$$

since $\mathbf{r}_2 = -\mathbf{r}_1$ if they are antipodal. By summing Equations 2 and 3, we have an expression that arises purely from the translational component of motion. The rotational components cancel out.

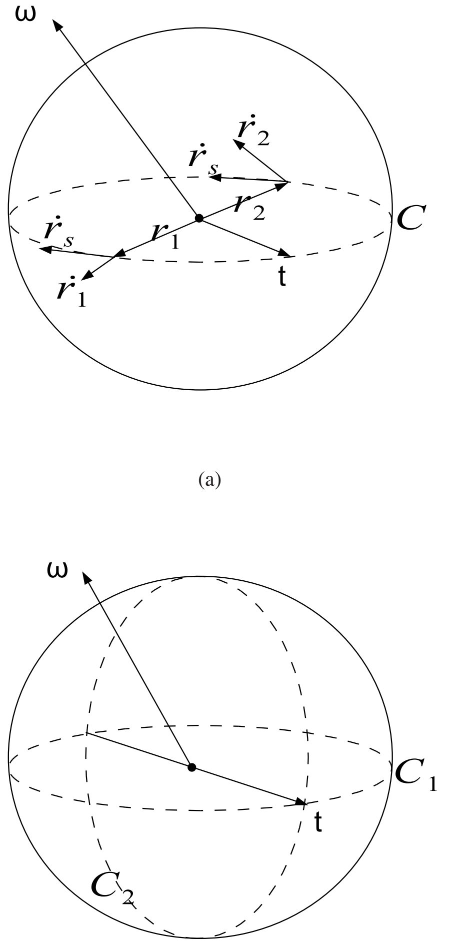
$$\begin{aligned} \dot{\mathbf{r}}_\mathbf{s} &= \dot{\mathbf{r_1}} + \dot{\mathbf{r_2}} \\ &= (\frac{1}{\mid \mathbf{R}(\mathbf{r_1}) \mid} + \frac{1}{\mid \mathbf{R}(-\mathbf{r_1}) \mid})((\mathbf{t} \cdot \mathbf{r_1})\mathbf{r_1} - \mathbf{t}) \\ &= K((\mathbf{t} \cdot \mathbf{r_1})\mathbf{r_1} - \mathbf{t}) \end{aligned} \tag{4}$$

From Equation 4, we see that vectors $\mathbf{r_1}$, $\dot{\mathbf{r}}_\mathbf{s}$ and $\mathbf{t}$ are coplanar. The normal of that plane is given by $\mathbf{r_1} \times \dot{\mathbf{r}}_\mathbf{s}$ where $\times$ denotes the vector cross product. Obviously, the epipole or direction of translation, $\mathbf{t}$, lies on that plane. The intersection of that plane with the image sphere gives a great circle. See Figure 2(a) for an illustration.

By picking another pair of antipodal points (that do not lie on the first great circle) and repeating, we obtain a second such plane or great circle. The intersection of the two planes or great circles gives an estimate of the epipole, $\mathbf{t}$. See Figure 2(b) for an illustration.

Of course, the intersection of the two great circles actually yields two points corresponding to $\mathbf{t}$ and $-\mathbf{t}$. To disambiguate between the two, one may pick one of the two points and calculate the angle between it and the vector $\dot{\mathbf{r}}_\mathbf{s}$. If the angle is larger than $\pi/2$ radians, then that point is $\mathbf{t}$. Otherwise, we have $-\mathbf{t}$.

The underlying principle here was first observed in [26]. However, major differences differentiate this work from that of [26]. The approach used there defined *angular flow*,



(a)

(b)

Figure 2. (a) Summing the flow $\dot{\mathbf{r_1}}$ and $\dot{\mathbf{r_2}}$ yields the vector $\dot{\mathbf{r_s}}$, which, together with $\mathbf{r_1}$ (or $\mathbf{r_2}$) , gives rise to a plane on which $\mathbf{t}$ is constrained to lie. The intersection of that plane with the sphere is the great circle $C$. (b) From two pairs of antipodal points, two great circles $C_1$ and $C_2$ are obtained. $\mathbf{t}$ is the intersection of these two circles.

which is obtained by taking the cross product of optical flow at a point with the direction of that point. In effect, a dual representation of flow is obtained and it was stated that if the angular flow at two antipodal points was *subtracted* from each other, the rotational component would vanish.

Both the constraint developed here and the one in [26] probably stem from a similar geometrical property inherent to antipodal points but the methods by which they were derived and the final results are quite different. The constraint in this paper does not require the transformation of optical flow into angular flow as in [26], a step which incurs an additional cross product.

Furthermore, the method of [26] requires a *subtraction* of angular flow at antipodal points and thus, that method fails for certain environments, such as when the antipodal points are equally far away (subtracting the angular flow in that case yields zero) - a problem observed by the authors of that paper. The method presented in this paper *sums* the optical flow at antipodal points and therefore, does not encounter such a problem.

Finally, because omnidirectional and panoramic sensors were not widely available then, the method of [26] was not fully developed into an algorithm and no implementations or experiments were ever conducted. Here a full algorithm is presented for the first time and tested on simulations as well as demonstrated on real sequences.

## 3. Obtaining robust estimates of t

We now have a method for obtaining some estimate of the direction of $\mathbf{t}$ given the flow at any pair of antipodal points. In this section, we outline a simple method for doing this *robustly*. There are two approaches to the problem.

Firstly, since the flow at every pair of antipodes yields a great circle which must intersect $\mathbf{t}$, the epipole could be estimated by trying to find the best intersection of many of these great circles. This is very similar to the problem of trying to find the best estimate of a vanishing point given many vanishing lines and many methods have been proposed in literature for doing this, for example [22], [18] and [24].

Alternatively, the flow at two such pairs of antipodal points gives a unique (up to a scale of course) solution for the location of the epipole from the intersection of two great circles. We can repeat the process for $N$ pairs of antipodes to get around ${}^{N}C_2$ possible solutions where each solution in the set is a point on the image sphere. This is a convenient representation, since many point clustering methods, mode finding methods, outlier rejection schemes and such exist, and may be used to robustly estimate the best answer.

We choose a method from the latter class of approaches. We attempt to find the support of all points in the set of possible solutions. The support of a point is defined here as the number of points within a certain threshold distance from the point. Since the points lie on the (unit) sphere, the geodesic distance between points is the merely the angle between them, which can be found by a dot product. Based on empirical results, we choose the threshold to be 0.01 radians. All points with the maximum support (or with large supports) are found and the arithmetic mean of the points is found. This is summarized in Algorithm 1.

This method is simple but it is very robust. Certainly, more efficient methods could be devised if processing time is critical, however, this paper attempts to evaluate the performance of the geometrical constraint derived and its usefulness in real sequences rather than the algorithm that uses the constraint. Therefore, a robust but slow algorithm is more useful for this purpose than a fast, efficient but less robust one. In any case, the number of flow vectors at antipodal points found in the images used are typically in the order of 300 to 800, so processing time is not unreasonably large.

**Algorithm 1** Estimating the epipole location

```
1: for $i = 1$ to $N$ do
2:    Select a pair of antipodes $\mathbf{r_1}$, $\mathbf{r_2}$ with optical flow $\dot{\mathbf{r_1}}$, $\dot{\mathbf{r_2}}$.
3:    $\dot{\mathbf{r_s}} = \dot{\mathbf{r_1}} + \dot{\mathbf{r_2}}$
4:    Take cross product $\mathbf{n_1} = \dot{\mathbf{r_s}} \times \mathbf{r_1}$
5:    Repeat steps 2 to 4 for a different pair of antipodes to obtain another normal vector $\mathbf{n_2}$ such that $\mathbf{n_1} \neq \mathbf{n_2}$
6:    Take cross product $\hat{\mathbf{t_i}} = \mathbf{n_1} \times \mathbf{n_2}$
7: end for
8: for $i = 1$ to $N$ do
9:    for $j = 1$ to $N$ do
10:       Find the angle $\theta_{i,j} = acos(\hat{\mathbf{t_i}} \cdot \hat{\mathbf{t_j}})$
11:       if $\theta_{i,j} < \theta_{threshold}$ then
12:          support for point $\hat{\mathbf{t_i}}$ , $supp_i = supp_i + 1$
13:       end if
14:    end for
15: end for
16: The estimated epipole is the $\hat{\mathbf{t_i}}$ with maximum support, $supp_i$. Alternatively, find all $\hat{\mathbf{t_i}}$ with $supp_i > threshold$ and take the arithmetic mean of their angles.
```

## 4. Experiments and Results

The goal of these experiments is to investigate the accuracy of the method in simulations under noise as well as to demonstrate its practicability with real sequences.

The simulation results are as shown in Table 1, showing high levels of accuracy. Depth, rotations and translations were randomly generated. Directions of rotation and translation were varied randomly over the set of all possible directions. Scene depth was in the range of 1 to 2, the magnitude of rotation was in the range of 0 to 3 and the translation was in the range of 0 to 1. Since this algorithm estimates translation, the larger the rotational component of motion is, the more challenging the problem. Optical flow was generated as per Equation 1. The angular noise in the optical flow was modeled as a Gaussian distribution with standard deviation $\sigma$.

The experiments on real image sequences were carried out by moving a camera along a trajectory that was marked out on the ground such that it underwent translation along the x-axis (parallel to the ground) and rotational motion about the z-axis (perpendicular to the ground). The method itself is suitable for general motions and not just motions on the ground plane, but for practical reasons, the experiments involved motion on the ground plane. The site of the experiment was a cluttered indoor scene such as that shown in Figure 3 and in the supplementary videos.

| $\sigma$(degrees) | average error(degrees) |
|---|---|
| 0 | $4.84\times10^{-6}$ |
| 5 | 2.49 |
| 10 | 5.66 |
| 20 | 10.14 |
| 30 | 13.12 |
| 60 | 19.79 |

Table 1. Simulated sequence: Average error in the estimated direction of motion as noise level increases.

The real sequences were captured with a Ladybug camera from Point Grey Research [1]. The Ladybug consists of 6 cameras positioned to capture three quarters of the view-sphere - 5 cameras are positioned in a ring whilst a single camera points upwards. However, we only use the 5 cameras in the ring since we require antipodal points for our work. The Ladybug camera system does not strictly have a single viewpoint although one is assumed, leading to some small errors in the calibration which can carry forward to the motion estimates.

In our implementation, SIFT features were extracted, matched and used as inputs to this algorithm. This was done using the SIFT keypoint detector software provided by [16].

Table 2 shows results for a purely translational sequence (zero or negligible rotation). Excellent accuracy and robustness is observed. The 'ground truth' against which the estimates are compared is translation along the x-axis. However, the ground truth is open to errors of up to several degrees due to practical measurement issues. Therefore the results are within the measurement accuracy of the experiment.

Table 3 shows the error in the estimation of the epipole location for a camera undergoing both translation and rotation. It can be seen that the algorithm performs robustly in real image sequences. With a small rotation coupled with translation, errors of below 2° are obtained, and with a larger rotation, the errors in the estimate increase slightly.

## 5. Discussion

It should be noted that this method is useful for estimating the translational component of motion in the presence of rotation, although it does not directly estimate the rotation of the camera. However, knowing translation, rotation is not difficult to recover. For example, in [10], with known translation, an algebraic manipulation to remove the translational component in Equation 1 is possible, thus making the equation linear in rotation. A least squares approach would then recover rotation.

| Frame No. | Error Seq (deg) |
|---|---|
| 2 | 0.83 |
| 4 | 0.38 |
| 6 | 0.57 |
| 8 | 0.79 |
| 10 | 1.38 |
| 12 | 0.97 |
| 14 | 1.41 |
| 16 | 0.52 |

Table 2. Typical errors in the estimation of the epipole for real scenes involving purely translational motion (2cm per frame). See enclosed video for full sequence.

| Frame No. | Error Seq 1(deg) | Error Seq 2(deg) |
|---|---|---|
| 2 | 0.45 | 3.42 |
| 4 | 0.54 | 5.43 |
| 6 | 0.08 | 1.04 |
| 8 | 0.41 | 7.56 |
| 10 | 1.49 | 1.44 |
| 12 | 0.57 | 0.82 |
| 14 | 1.55 | 0.73 |
| 16 | 2.23 | 2.27 |

Table 3. Typical errors in the estimation of the epipole for real image sequences. Sequence 1 involves translation by 2cm and rotation by 2° per frame whilst Sequence 2 involves translation by 2cm and rotation by 5° per frame. A video of sequence 1 included.

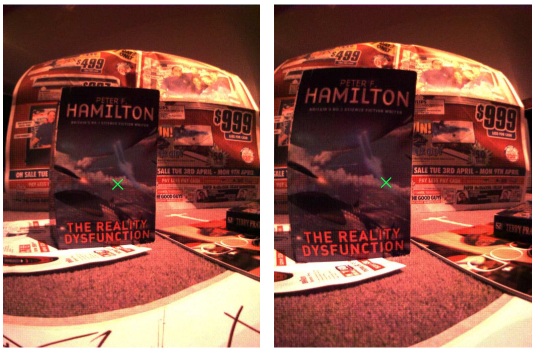


Figure 3. Two frames showing a typical image sequence (camera undergoing translation and rotation). Estimated epipole marked with a cross. Full video sequence included.

In general, this method takes as inputs any measurement of the movement of image points and should work for optical flow obtained by both feature correspondences (such as SIFT, Harris corners) as well as correspondence-free methods (such as the spatio-temporal derivatives of Lucas-Kanade and Horn-Schunck). Our implementation used optical flow from feature correspondences (SIFT) since they provided more accurate measures of image motion compared to the latter, which are noisier. In our experiments, the number of antipodal SIFT features were found to be quite plentiful. However, in settings where this may not be the case, the use of dense optical flow from spatio-temporal derivatives may be more useful in providing input measurements.

Another advantage is that the method works for both large and small baselines. The baseline size would affect which method of optical flow measurement is used (for example, image gradient optical flow methods typically require small baselines whereas SIFT point matching works well for large baselines). Certainly, in the case of very small baselines and a large, dominating rotation, there would be greater ambiguity in the estimated epipole but the method does not break down.

## 6. Conclusion

In summary, we have presented a constraint on the epipole arising from the optical flow at two antipodal points. We demonstrated the validity of the constraint and devised a simple yet robust algorithm utilizing it to estimate the location of the epipole. We have shown that with noisy, real images this method does work robustly and accurately.

**Supplementary videos:** Two video sequences are included. In one the camera undergoes pure translation, and in the other, it undergoes both translation and rotation. The epipole is extracted and marked with a green cross.

**Acknowledgements:** NICTA is funded by the Australian Department of Communications, Information Technology and the Arts and the Australian Research Council through *Backing Australia's ability* and the ICT Centre of Excellence Program.

## References

[1] Point grey research, 2007. http://www.ptgrey.com/. 4

[2] T. Brodsky, C. Fermuller, and Y. Aloimonos. Directions of motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 26:5–24, 1998. 2

[3] A. R. Bruss and B. K. Horn. Passive navigation. In *Computer Vision, Graphics and Image Processing*, volume 21, pages 3–20, 1983. 1

[4] C. Fermuller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995. 1

[5] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, 1995. 1

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 2

[7] B. Horn and B. Schunck. Determining optical flows. *Artificial Intelligence*, 17:185–203, 1981. 2

[8] B. Horn and E. Weldon. Direct methods for recovering motion. *International Journal on Computer Vision*, pages 51–76, 1988. 1

[9] T. Huang and A. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):253–268, February 1994. 1

[10] A. Jepson and D. Heeger. Subspace methods for recovering rigid motion i: algorithm and implementation. *International Journal of Computer Vision*, 7(2), 1992. 1, 2, 4

[11] 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. 1

[12] 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. 1

[13] H. Li. A simple solution to the six-point two-view focal-length problem. In *European Conference on Computer Vision*, pages 200–213, 2006. 1

[14] H. Li and R. Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 630–633, 2006. 1

[15] H. Longuet-Higgins. A computer algorithm for reconstruction of a scene from two projections. *Nature*, 293:133–135, 1981. 1

[16] D. Lowe, 2007. http://www.cs.ubc.ca/ lowe/keypoints/. 4

[17] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981. 2

[18] M. Magee and J. Aggarwal. Determining vanishing points from perspective images. In *Computer Vision, Graphics and Image Processing*, volume 26, pages 256–267, 1984. 3

[19] S. Negahdaripour and B. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987. 1

[20] 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. 1

[21] 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. 1

[22] L. Quan and R. Mohr. Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters*, 9(4):279–286, 1989. 3

[23] 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. 1

[24] J. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, 1999. 3

[25] C. Silva and J. Santos-Victor. Direct egomotion estimation. In *Proc. 13th International Conference on Pattern Recognition*, 1996. 1

[26] I. Thomas and E. Simoncelli. Linear structure from motion. Technical report, IRCS, University of Pennsylvania, 1994. 1, 2, 3

[27] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 315–320, 1996. 1