Real Time Motion Recovery using a Hemispherical Sensor

Rebecca Dengate, Nick Barnes, John Lim, Chi Luu and Robyn Guymer
National ICT Australia, the Australian National University and the Centre for Eye Research Australia
nick.barnes@nicta.com.au

Abstract

This paper describes work in a new project based on a collaboration between experts in low vision and a computer vision research group. The focus of the project is to develop assistive devices for individuals with severe and profound vision impairment resulting from diseases such as Age-related Macular Degeneration and Retinitis Pigmentosa. We describe focus groups that are being conducted to understand such needs. To assist with tasks such as navigation and obstacle avoidance for an individual who is walking, knowledge of self-motion is essential. In this context we present a new implementation of a wide angle camera visual motion recovery algorithm suitable for use on a low cost, low power, light-weight wearable sensing device. For wearable sensing, camera paths are far more erratic than for ground based vehicles such as wheeled robots or cars. Also, weight from computing, cameras and batteries is a major issue.

1 Introduction

In this paper we present a new project that aims to develop assistive devices for the visually impaired including some early research results. The study brings two organisations together, a team of medical researchers whose aim is to improve living conditions and lifestyles of people affected by vision impairment, and a computer vision research group. The project aims specifically to produce wearable sensing and computing technologies that recover and interpret visual information to improve the ability of individuals with low vision to function in their everyday life, and maintain or enhance their ability to live independently.

The diseases we focus on for this project are Retinitis Pigmentosa and Age-related Macular Degeneration. Retinitis pigmentosa (RP) is an inherited degenerative retinal disease that results in death of photoreceptors leading to progressive vision loss. Worldwide, about 1.5 million people suffer from RP which makes it the leading cause of inherited blindness (visual acuity of less than 6/60 in the better eye) [Bunker et al., 1984]. Age-related macular degeneration (AMD), a progressive retinal disease associated with aging and affecting the central retina (the macular), is the major cause of blindness in Western countries [Taylor et al., 2005; Mitchell et al., 1995]. For example, in Australia AMD is responsible for 48% of blindness in persons over 40 years of age [Taylor et al., 2005]. At present, there is no effective treatment for most patients with RP and AMD. Thus many RP and AMD patients would need some level of visual aids to reduce their social restrictions.

This project aims to develop real-time image stream processing for low vision assistance. To support independent living, low vision assistive devices must travel with the individual, that is wearable sensing and computation. If a wearable device is to be used frequently it must be light-weight and able to operate for some time without charging. Consequently we are interested in developing algorithms that are low power and operate on light-weight devices. Further, moving around in the world is an important component of what people do in their everyday lives. In order to support any possible low vision devices, a system would be aided by an understanding of self motion, particularly for assisting in way-finding and obstacle avoidance. From wearable sensing, self-motion is complicated by factors such as the complexity of walking motion, and that people often do not take smooth paths. As such in this paper we describe a new implementation of a robust algorithm for six degree of freedom self-motion recovery that has low computational cost.

This paper first describes the focus groups that will be conducted by the study including the target groups for assistive technologies. We then describe an algorithm that is suitable for real-time motion recovery from wearable cameras. Due to the nature of the algorithm an er-
ror is introduced through the use of a wide field-of-view camera rather than a spherical camera. We present an analysis of these errors and the real-time motion recovery results obtained when the algorithm is implemented on a small light-weight wearable wide field-of-view camera. Importantly the technique gives constant time performance regardless of the number of outliers in the data it assesses.

2 Focus groups

Certain aspects of vision are more important to one person than another depending on the individual’s occupation, life style, type of blindness (loss of central or peripheral vision) and the age at the onset of blindness. To understand more from blind individuals about what aspects of vision are more important to their daily lives we will be conducting focus groups with individuals who have profound visual loss. The main objective of the study is to identify the most important visual cues needed from blind people to help perform their daily tasks. The information collected from this study will allow us to focus on the development of specific devices that meet the needs of the patients.

In this study, we will have about 5 focus groups. Each focus group will be made up of 8-10 volunteers with severe visual loss caused by a variety of conditions including RP and AMD. The information relevant to the study will be gathered through casual discussion amongst the members of the group. The discussion will be based around some set questions to keep the group on track. Questions relating to difficult daily tasks, the things individuals would trade off for sight, and the visual cues such as colour, size, movement that are most important to them will be asked. The discussion will be recorded on a tape and the information will be transcribed for qualitative data analysis.

3 Real-time motion recovery on wide field of view cameras

Many methods exist for estimating the self-motion of a monocular observer, such as those in reviews [Tian et al., 1996] and [Huang and Netravali, 1994]. Optical flow is often used as it is a computationally inexpensive way of estimating motion from a single camera. Most approaches consider flow over an image with a limited field of view (FOV). However, a spherical camera has advantages in estimating 3D motion [Fermüller and Aloimonos, 2000] as ambiguities arise from using a small FOV [Nelson and Aloimonos, 1988]. [Fermüller and Aloimonos, 1988; Nelson and Aloimonos, 1988] both utilize the properties of large FOV cameras to estimate self-motion, using search-based algorithms.

A faster method of exploiting the large FOV property is given by Lim and Barnes [2007]. Lim and Barnes use antipodal points on the view sphere to generate constraints on the direction of translation. The basis of the algorithm is somewhat similar to [Thomas and Simoncelli, 1994] but uses summation rather than subtraction of flow, so is more successful in some situations.

In this paper, we present a real-time implementation of this algorithm with a wide FOV camera. Stable results are obtained despite a limited area on the image where flow can be found. The error in direction of translation is dependent on the direction of translation, and an analysis demonstrates why this occurs. Preliminary results show that the direction of translation can be found at a rate of approximately 5 frames per second. Trials using image sequences generated from the InsectBot robot [Lim et al., 2006] demonstrate an error in direction of translation of around 5 degrees which is sufficient for obstacle avoidance. Importantly, this result is obtained on a $100 camera with a $250 lens, as opposed to a $10,000 Ladybug camera. A depth map is also obtained from a test sequence.

An analysis of the robustness of the algorithm is given in [Lim and Barnes, 2007] with simulation over noisy real image sequences involving independently moving objects, however to our knowledge this is the first demonstration of the algorithm in real world conditions.

3.1 Theory

We outline the theory for the recovery of motion, based on [Lim and Barnes, 2007]. Equation (1), from [Brodsky et al., 1998], gives the relation between the rigid motion of the camera and the image motion for some point \( r_1 \) on the sphere. Optical flow \( \dot{r} \) results from translational and rotational components of motion \( t \) and \( w \). See Figure 1.

\[
\dot{r}_1 = \frac{1}{|R(r_1)|}((t \cdot r_1)r_1 - t) - w \times r_1. \tag{1}
\]

To solve for translation, we must recover \( t \) from Equation (1). Given an image sequence, the camera centre of
the first image is related to the camera centre of the second image by some translation $t$ (recoverable only up to a scale [Hartley and Zisserman, 2000]). This means that the direction of translation is the epipole, as the epipole is defined as the intersection of the line joining the two camera centres with the image sphere.

Given a point $r_2$ that is antipodal to $r_1$, the optical flow at that point will be

$$r_2 = \frac{1}{|R(-r_1)|}((t \cdot r_1)r_1 - t) + w \times r_1. \tag{2}$$

Note $r_2 = -r_1$ due to geometric properties of antipodes. Summing Equations (1) and (2), we get

$$r_s = \left(\frac{1}{|R(r_1)|} + \frac{1}{|R(-r_1)|}\right)((t \cdot r_1)r_1 - t). \tag{3}$$

$r_1$, $r_s$, and $t$ are coplanar, as can be seen in Equation (3). $t$ lies on that plane, the normal of which is given by the cross product of $r_1$ and $r_s$. The intersection of that plane with the image sphere produces a great circle (a plane that cuts the sphere into two hemispheres). By repeating this process with another pair of antipodal points a second great circle is obtained. The intersection of the two great circles gives an estimate of $t$.

To get a robust estimate of $t$ the best intersection of many great circles is found, using a voting method similar to the Hough transform. The algorithm relies on coarse-to-fine voting, where in the fine stage gnomonic projection is used so the voting can be performed on a projection plane rather than the image sphere to increase efficiency [Lim and Barnes, 2009].

As observed in [Lim and Barnes, 2007], rotation can be recovered provided an estimate of the translation is known. To eliminate dependence on depth, the dot product of every term of Equation (1) with $t \times r$ is taken. This gives an equation linear in rotation:

$$(t \times r_1) \cdot r_1 = (t \times r_1) \cdot \left(\frac{1}{|R(r_1)|}((t \cdot r_1)r_1 - t) - (t \times r_1) \cdot (w \times r_1)\right) \tag{4}$$

$$(t \times r_1) \cdot r_1 = (t \times r_1) \cdot (w \times r_1).$$

Taking flow at $n$ points creates a system of $n$ linear equations that can be solved using linear least squares. In the fine voting stage of translation-finding antipodal flow vectors that did not intersect the epipole were discarded as outliers. This increases the robustness of the estimate of rotation, although it should be noted that some antipodal flow vectors that intersect the epipole may be outliers, even though they support the same translation.

Figure 2: The dark band indicates an area where antipodes can be found on the 190° view sphere. The $\phi$ value of both points must be between $-5^\circ$ and $5^\circ$ for a valid point pair to exist. The far right case demonstrates that if $r_1$ has a $\phi$ value of greater than $5^\circ$ its antipode does not exist on the hemisphere.

Given rotation and translation, finding depth is trivial. Equation (1) is rearranged as shown:

$$\frac{|R(r_1)|}{|t|} = \frac{(\hat{t} \cdot r_1)r_1 - \hat{t}}{r_1 + (w \times r_1)}. \tag{5}$$

It should be noted that this produces time-to-contact, or scaled depth, rather than actual depth. To obtain actual depth Equation (5) must be multiplied by the translation magnitude.

### 3.2 Implementation

The translation recovery algorithm was implemented for use with a single 190° FOV fish-eye camera, an Omni Tech Robotics Fire-i BCL 1.2 lens with a Unibrain CCD array. Since antipodal point pairs were needed, the area in which optical flow could be used was constrained to a 10° band as shown in Figure 2.

Camera calibration was required for the selection of point pairs to implement the algorithm. The perspective projection method is unsuitable for calibrating fish-eye lenses, so a radially symmetric projection model was used. [Kannala and Brandt, 2006] present a method that models the inherent distortion of the fish-eye lens.

In addition to the 10° band constraint the Omni Tech lens extends over the edge of the CCD array, leading to cropping at the top and bottom of the image. This reduces the area to that shown in Figure 10. The use of this reduced area introduces an error that is based on the direction of translation.

The algorithm was implemented in C++ using code from the open source computer vision library OpenCV [Intel, 2008]. The Lucas-Kanade algorithm [Lucas and Kanade, 1984] was used to calculate optical flow in a sparse iterative pyramidal based implementation. The algorithm was chosen on the basis of strong performances in a comparison of optical flow methods for real-time robust operation [McCarthy and Barnes, 2004].
Figure 3: The voting table produced during translation recovery over a pair of images, for $z$ translation

Figure 4: The point pairs chosen to generate the great circles, whose intersection will define the direction of translation

3.3 Analysis

Modelling error based on direction of translation

In the fine stage of voting for the egomotion algorithm, finding the direction of translation amounts to finding the best intersection of many lines. This can be seen in the voting table generated by a pair of frames, shown in Figure 3. An error in direction of translation is induced because the best intersection of many lines in the presence of noise is significantly worse if the lines are all within a small angle of each other. The maximum angle between the lines is governed by the geometry of the cropped band containing the antipodal point pairs at which flow can be found. This results in worse errors in the $x$ and $y$ directions than the $z$ direction.

We examined this phenomenon using a simplified version of the algorithm. In these tests, random rotation of magnitude 0.2 rad was generated. Two pairs of antipodal points $r_F$ (r first) and $r_S$ (r second) were chosen from the cropped $10^\circ$ band, as shown in Figure 4. Flow was generated at $r_F$ using Equation 1. Noise was added by generating random vectors in a Gaussian distribution around the flow vector and reprojecting them on to the tangent plane, then adding the result to the flow vector. A Gaussian standard deviation of 0.001 $^\circ$ was used, which was appropriate relative to the size of the flow vectors. By generating flow that simulated motion in the axis directions, the error in direction of translation for $x$, $y$ and $z$ translation could be compared, as shown in Figure 5. It is consistently high in the $y$ direction, with outliers in $x$ that have a greater error than $y$. $z$ has the smallest error. The variation in the error within the different directions of translation is due to the locations of points $r_F$ and $r_S$ within the band.

To see the relationship between point position and error, $r_F$ and $r_S$ were plotted on a table of $\theta$ versus $\phi$ for point pairs causing errors of less than $20^\circ$ in $x$ (Figure 6 i)), point pairs causing errors of greater than $140^\circ$ in $x$ (Figure 6 ii)), and point pairs causing errors of greater than $120^\circ$ in $y$ (Figure 6 iii)). These tables represent the $10^\circ$ band at the base of the hemispherical camera, with the $\theta$ value wrapping horizontally. The length between $r_F$ and $r_S$ as well as between $r_F$ and the antipode of $r_S$ was found, and the two points that had the minimum distance between them were plotted and connected.

In Figure 6 translation in the $x$ direction is examined, where the top figure shows point pairs that cause errors of less than $20^\circ$ and the middle figure shows point pairs that cause errors of greater than $140^\circ$. It can be seen that the points that pass through $\theta = 0^\circ$ or $180^\circ$ and have $\phi$ values that are roughly equal and opposite seem to cause errors. Even points that do not pass through $\theta = 0^\circ$ or $180^\circ$, but would if the line connecting them was extended, cause errors. This is because the points are essentially antipodal. The two planes generated are almost equal, producing a worst case error as the solution is ill-conditioned. Generally, directions of translation generated by point pairs with good angular separa-
Figure 6: Top: i) In the top figure, point pairs that produce a low error for translation in the $x$ direction are shown. Middle: ii) Point pairs that lead to inaccurate recovery of motion for $x$ translation. Bottom: iii) Point pairs that lead to inaccurate recovery of motion for $y$ translation.

Figure 7: Point pairs with similar $\phi$ values cause larger error by reducing the angle between the planes.

Translation in the $y$ direction is less accurate if the point pairs have similar $\phi$ values (Figure 6, bottom) as these point pairs produce similar planes so the angle between the plane normals is small. This is illustrated in Figure 7. In the case of $x$ translation, similar $\phi$ values will still produce a large angle between the two plane normals.

In summary, the error may be large for $x$ or $y$ translations depending on the point pairs chosen to generate the two planes. In the $z$ direction, the ratio of point combinations producing large errors to point combinations producing small errors is small, so the algorithm is unlikely to fail.

Building an improved sensor

In practice, the errors that occur are not as large as this simplified model predicts due to the robustness of the Lim and Barnes [2007] algorithm, where multiple point pairs - around 80 - are used in every estimation of translation. This is more helpful for translations in the $x$ direction than the $y$ direction, as due to the location of the cropping, many point pairs that would produce a more accurate estimation of translation are not possible. This is shown in Figure 8.

If a navigational system requires motion in the $x$ or $y$ direction, a new Unibrain 1/3” CCD firewire camera is available that eliminates the errors due to cropping. In the case of $z$ translation however the new camera would not offer significant advantages.

In addition to the constraints discussed, having a limited area over which to find flow causes problems if flow cannot be found in that area, for example if there is insufficient texture. This problem could be eliminated by finding flow over an entire viewsphere.

To create a complete viewsphere, two 190° FOV cameras can be placed back to back. The camera centers
cannot be coincident because each lens is attached to a CCD array. A baseline is introduced between the camera centers that prevents points from being antipodal as $\phi$ decreases. In practical terms, if the baseline is small compared to the scene depth the error term vanishes. The magnitude of the error term relies linearly on scene depth.

3.4 Real-time translation recovery

On a PC with an AMD Athlon 64 bit processor 3200+ (2.2GHz) and 512MB RAM, computer frames were processed at a rate of 5.02 frames per second (0.199 seconds per cycle). A video can be found at http://users.rsise.anu.edu.au/~nmb/RT_Trans.avi that shows a sequence captured in real-time, with the direction of translation indicated. The vector overlaid shows the direction of translation. The mean error for this 75 frame sequence is 18°. See Figure 9 for a histogram of the error. It is worth noting that the error is based on the assumption that the camera was moved in exactly the negative x direction whereas in reality the frame to frame direction may be up to 10° away from the negative x direction.

3.5 InsectBot testing

Tests were performed where the camera was attached to the lifting platform on the InsectBot robot described in [Lim et al., 2006]. This allowed translation in the near z direction, minimizing the error-inducing effects of the cropped 10° band. The sequence was recorded in a corridor with textured walls and ceiling to maximize the chance of optical flow being available at every point. The video sequence transSeqB.avi is found at http://users.rsise.anu.edu.au/~nmb/transSeqB.avi.

Figure 11 shows flow found at a frame pair in the InsectBot sequence, where the flow vectors have been plotted on the image. Flow has been found consistently around the antipodal point band.
It can be seen in Figure 12 that the translation recovery is consistent. The error in the sequence is plotted in Figure 13 where the direction of translation is assumed to be $[0, 0, 1]$, which shows consistent $20^\circ$ deviation from the $z$ axis. This suggests a systematic error, which can be accounted for by the unstable mounting of the camera on the robot. The systematic error is apparent in the image sequence transSeqB.avi, included here. Since the systematic error is not necessarily constant or linear, the exact error is still unknown. To compensate for the unknown actual translation, the error is plotted with respect to the median of the translation sequence in Figure 14. The mean error from the median was $5.3^\circ$. The mean recovered rotation magnitude was $0.21^\circ$ over the whole sequence, which seems correct given that rotation was not intentionally introduced.

### 3.6 Depth map generation

Figure 15 shows a depth map generated from a frame in an $x$ translation sequence. Light areas represent objects closer to the camera. The red areas represent places on the image where flow is absent - for example the light fixture on the ceiling where there is no texture. The recovered depth is less accurate around the focus of contraction (FOC) and focus of expansion (FOE). This is expected as the flow found in these directions is caused by mostly by divergence.

### 4 Conclusion

In this paper we gave an overview of a new collaborative project to develop assistive devices for individuals with
impaired vision. Focus groups have begun to assess the needs of individuals with RP and AMD. We anticipate that key needs will be in the areas of motion recovery and object detection, particularly on light-weight wearable cameras.

The Lim and Barnes algorithm [2007] was implemented on such a camera. This allowed the exploration of errors arising from constraints on point pair location that meant flow could not be used at every point on the image. The translation recovery was accurate up to about 5° which is sufficient for obstacle avoidance. This result was obtained in a highly textured environment in which flow could be found over the entire image, which represents a best case scenario. As most environments will be less textured, using a complete viewsphere rather than a wide field-of-view camera could reduce errors. Results in [Lim and Barnes, 2007] which tested the algorithm on the Ladybug camera had smaller errors, which supports using a full viewsphere. A cheaper way to create a viewsphere would be to place two fish eye cameras back to back. It should be noted that the analysis performed also holds for a catadioptric camera, for example that described in [Geyer et al., 2001]. Such a camera could also be used to create a low-cost system.

Acknowledgments

The authors are grateful to Luke Cole for his assistance with the InsectBot during the testing phase.

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


