

Real Time Biologically-Inspired Depth Maps from Spherical Flow

Chris McCarthy, Nick Barnes and Mandyam Srinivasan

Abstract—We present a strategy for generating real-time relative depth maps of an environment from optical flow, under general motion. We achieve this using an insect-inspired hemispherical fish-eye sensor with 190 degree FOV, and a de-rotated optical flow field. The de-rotation algorithm applied is based on the theoretical work of Nelson and Aloimonos [10]. From this we obtain the translational component of motion, and construct full relative depth maps on the sphere. We examine the robustness of this strategy in both simulation and real-world experiments, for a variety of environmental scenarios. To our knowledge, this is the first demonstrated implementation of the Nelson and Aloimonos algorithm working in real-time, over real image sequences. In addition, we apply this algorithm to the real-time recovery of full relative depth maps. These preliminary results demonstrate the feasibility of this approach for closed-loop control of a robot.

I. INTRODUCTION

Essential to autonomous navigation is the ability to perceive depth. While absolute measures of distance are useful, they are not necessary for achieving most navigation tasks. Relative measures of distance to surfaces have been shown to be sufficient for autonomously navigating corridors [12], avoiding obstacles [13], [2], and docking with objects in the environment [9], [12]. In addition, obtaining depth maps across a wide field of view provides a means of perceiving environmental structure, which in turn may be used for higher level navigation tasks such as invoking appropriate navigational subsystems and mapping.

It is well known that biological vision systems perceive depth from a variety of cues, depending on the configuration and the geometry of the eye. Among the potential cues are: (a) stereo information; (b) depth from focus; (c) depth from convergence; and, (d) depth from optical flow [16]. Insects, with their immobile, fixed-focus eyes and low interocular separation, rely heavily on optical flow cues to obtain depth information [17]. Information from the optical flow that is generated in the eyes by the insects' motion in the environment is used to: (i) navigate safely through narrow gaps; (ii) detect and avoid collisions with objects; (iii) distinguish objects from their immediate backgrounds; and, (iv) orchestrate smooth landings [17]. Estimating depth from optical flow is simpler, computationally, than estimating depth from stereo, and is thus an attractive strategy for the relatively simple nervous systems of insects. Given many robotic vision systems are equipped with only a single camera, much attention has been given to those cues that

do not rely on two or more simultaneous views of the scene, such as optical flow.

In computer vision, the use of optical flow for scene reconstruction has been problematic. While a significant body of theoretical work exists [7], [1], [11], [5], in practice, these approaches generally lack the speed and robustness required for a real-time navigation system. One issue is the intolerance of such strategies to noisy flow estimates in local regions [18]. Optical flow estimation is notoriously noisy, and difficult to compute accurately under real-world conditions.

An additional issue for depth map recovery is the reliable extraction of the translational component of flow (*de-rotation*), from which relative depth can be inferred. Many current flow-based depth perception strategies either assume pure translational motion (*e.g.* [3]), or apply planar models to extract surfaces from the scene (*e.g.* [13], [14]). While de-rotation algorithms exist, these are largely constrained to a single rotation, or are not fast or robust enough for real-time depth mapping. To generate 3D depth maps from optical flow under general motion, a de-rotation strategy is required for each rotational component. To obtain workable depth maps for navigation from flow, this algorithm must be sufficiently accurate, and must run in real-time. There currently exists no system capable of achieving this in real-world conditions.

There is a growing body of theoretical work suggesting a spherical projection model, rather than a perspective model, may offer distinct advantages when inferring scene structure and self-motion from optical flow [4], [10]. Nelson and Aloimonos [10] highlight specific advantages gained through the existence of both a focus of expansion (FOE) and focus of contraction (FOC) in a single spherical image. From such observations, a potentially real-time de-rotation algorithm is derived for the complete recovery of rotational velocity components from the optical flow on a full view sphere. While some theoretical analysis of the algorithm's likely robustness in real-world conditions is provided, to our knowledge there exists no published results to date reporting the algorithm's performance in real-time, and over real image sequences. Given the potential of this algorithm to support a variety of navigation tasks, it is of interest to examine its feasibility for the recovery of 3D depth maps in real-time.

In this paper, we report preliminary results obtained from the application of the Nelson and Aloimonos de-rotation algorithm to the task of generating 3D relative depth maps from a spherical sensor. In simulation, we show its application over a full view sphere undergoing general motion. We also present preliminary results over two real image sequences captured during the ground-based motion of a hemispherical view fish-eye camera. We show that the

C. McCarthy and N. Barnes are with the Vision, Science and Technology Programme, National ICT, Australia, and the Dept. Information Engineering, The Australian National University
M. Srinivasan is with the Centre for Visual Sciences, The Australian National University

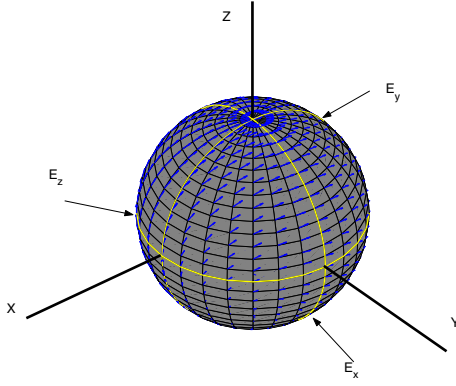


Fig. 1. Optical flow on the view sphere.

recovery of 3D relative depth maps can be achieved in real-time, without the need for camera calibration.

II. GENERATING DEPTH MAPS FROM SPHERICAL FLOW

We briefly outline the theory for recovering 3D depth maps from the spherical projection of optical flow. For the purposes of de-rotation, it is of interest to consider the component of flow in the direction of great circles about each axis of rotation [10].

We define the position of any point on a view sphere in terms of its angular location on three great circles lying in orthogonal planes perpendicular to each axis, such that:

$$\theta = [\theta_x \quad \theta_y \quad \theta_z], \quad (1)$$

where θ_x , θ_y and θ_z are angles in the direction of each orthogonal great circle, \mathbf{E}_x , \mathbf{E}_y and \mathbf{E}_z , in the range $[0, 2\pi]$, as shown in Figure 1. Using this representation, we may express the component of motion, $e(\theta)$, in the direction of these equators as (defined here for \mathbf{E}_x):

$$e_x(\theta) = \frac{v \cos(\alpha)}{R(\theta_x)} \sin(\phi_x - \theta_x) + \omega_x, \quad (2)$$

where ω_x is the rotational velocity component about the X axis, $R(\theta_x)$ is the radial depth to the scene point projecting to θ_x , v is the translational velocity of the sphere, α is the angle of the translation vector from the plane of \mathbf{E}_x , and ϕ_x is the direction of the projection of this vector on \mathbf{E}_x .

It is important to note that the above equation may be defined for any great circle, and is not limited to equators about the X, Y and Z axis. From this we note that flow in the direction of any great circle is effected by only a single rotation about the axis perpendicular to that great circle's plane. This observation has led to the development of a full de-rotation algorithm for optical flow on the sphere.

Algorithm 1 Nelson and Aloimonos De-rotation Algorithm [10]

```

1: for  $\omega_c = \omega_{min}$  to  $\omega_{max}$  do
2:    $D[\omega_c] = 0$ 
3:   for  $\theta_c = 0$  to  $2\pi$  do
4:      $a = e(\theta_c) - \omega_c$ 
5:      $\beta = \phi - \theta_c$ 
6:     if  $a < 0$  and  $0 \leq \beta < \pi$  then
7:        $result = -a$ 
8:     else if  $a > 0$  and  $\pi \leq \beta < 2\pi$  then
9:        $result = a$ 
10:    else
11:       $result = 0$ 
12:    end if
13:     $D[\omega_c] = D[\omega_c] + result$ 
14:  end for
15: end for
16:  $\omega = \text{min\_index}(D[\omega_{min} : \omega_{max}])$ 
17: return  $\omega$ 

```

A. De-rotating flow on the sphere

In the late 1980s, Nelson and Aloimonos [10] proposed an algorithm for recovering the full 3D rotational velocities of a view sphere, in real-time, by exploiting three key geometric properties of optical flow on the sphere:

- 1) the component of flow parallel to any great circle is effected only by the rotational component about its perpendicular axis, thus decoupling it from rotations about orthogonal axes.
- 2) under pure translation, both the FOE and FOC will co-exist at antipodal points on the sphere, and will evenly partition flow along any great circle connecting these two points, into two distinct directions of motion (*i.e.* clockwise and counter-clockwise).
- 3) the existence of any rotational motion along a great circle causes the FOE and FOC to converge, thus ensuring the two points will only lie at antipodal locations under pure translation.

The first observation indicates that each component of rotation can be resolved independently, and thus each may be considered in turn. From the second and third observations, Nelson and Aloimonos propose an algorithm for recovering the rotational component of flow, ω , about any great circle of flow $e(\theta)$. For reference, we reproduce the algorithm in pseudo code (see Algorithm 1).

In words, the algorithm describes a simple search-based strategy for resolving rotation. For each discrete point, θ_c , on a circle of flow, $e(\theta)$, a range of rotations are searched through, where each candidate, ω_c , is used to de-rotate flow along the circle. After de-rotation, the sum of the residual flow on the circle is taken. Note that the sign of the flow indicates its direction on the circle, therefore, a perfect split of clockwise and counter-clockwise flow will yield a sum of 0. Accounting for noise and quantisation errors, the chosen rotation is therefore the one which yields the smallest sum

TABLE I
SIMULATION ERROR MEASURES

Gauss noise (std dev)	Rotational error			Trans dir error	Depth error
	ω_x	ω_y	ω_z		
0°	0.003	0.003	0.003	5.9°	8.9%
2°	0.003	0.003	0.004	9.0°	13.0%
4°	0.006	0.005	0.007	10.0°	25.3%
10°	0.009	0.008	0.012	16.6°	39.2%

of flow on the circle after de-rotation.

By applying this algorithm to great circles about each rotational axis, the complete recovery of the sphere's rotation is achieved. After de-rotation, the direction of translation is also given by the line passing through the FOE and FOC.

Notably, the algorithm's run time performance is dictated primarily by the quantisation of discrete locations on the great circle, and the range of possible rotations for each great circle. Given reasonable choices, the algorithm should provide fast execution [10].

B. Generating relative depth maps

After de-rotation, all flow vectors follow great circles passing through the FOE and FOC. Thus, we may express the magnitude of the residual translational flow at a discrete location on the great circle as (θ) as:

$$f(\theta) = \frac{v}{R(\theta)} \sin(\phi - \theta), \quad (3)$$

and from this, define the relative distance to any scene point:

$$\frac{R(\theta)}{v} = \frac{\sin(\phi - \theta)}{f(\theta)}. \quad (4)$$

If we know v , then we may obtain an absolute measure of distance, however, for most navigation tasks, a relative depth estimate is sufficient. Notably, (4) is only defined where optical flow exists (*i.e.* $f(\theta) \neq 0$). Thus, range cannot be reliably measured where a lack of texture exists, or where flow magnitude tends to zero such as at the FOE and FOC.

C. Simulation results

Simulation tests were conducted to examine the robustness of depth map estimates obtained using the strategy outlined above. For this, a model of a unit view sphere was defined, and immersed in a virtual 3D boxed space. A ground truth depth map was then obtained over the whole sphere, from which accuracy could be measured. Optical flow fields were computed on the view sphere for multiple sets of twenty randomly chosen translation directions, with randomly chosen components of rotation about each principle axis. Rotational velocity values were within the range $[-0.5, 0.5]$ (rad/unit time). To examine robustness, increasing levels of Gaussian noise were added to the direction of each flow vector for each set of twenty trials. On each equator, 112 discrete points, and 100 possible rotations were used for de-rotation. Table I provides mean errors obtained during each simulation run. Rotational errors are given as the mean of the absolute difference between the estimated and true rotation velocities for each simulation run. The translational direction error is given

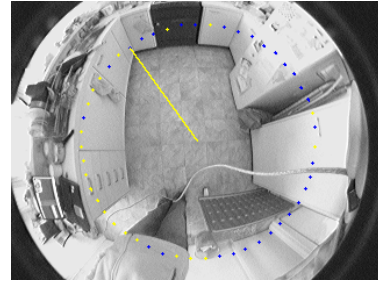


Fig. 2. Sample frame shows the circle used to de-rotate the flow. Blue pixels indicate clockwise flow and yellow counter-clockwise. The yellow line indicates the estimated direction of translation after de-rotation.

as the mean angular error between the estimated translational direction and ground truth. Depth map estimation errors are given as the mean of relative errors against ground truth.

From Table I it can be seen that rotational errors remain stable, despite increasing noise levels. Translational direction errors also appear to remain stable. It is important to note that the number of discrete points used on the de-rotation circle dictates the accuracy of translational direction estimates. Using 112 points, accuracy can only be achieved to within 3.2° of the actual direction, hence the presence of non-zero errors when no noise is introduced. This error was observed to reduce in proportion to an increase in the number of discrete points used on the de-rotation circle.

It is clear that depth estimation errors increase with noise, and with errors in the estimated translational direction. Notably, large errors were also found to occur around the FOE and FOC, where flow magnitudes approach zero. Under less precise, real-world conditions, this issue is less likely to influence accuracy due to noise levels preventing flow magnitudes from diminishing to such small values.

III. REAL-WORLD EXPERIMENTS

The 3D depth map algorithm was implemented for use with a single fish-eye camera (Unibrain Fire-i BCL 1.2¹) undergoing ground-based motion. Given motion was approximately planar, only a single hemispherical view was required to account for rotation on the ground plane. The fish-eye camera has a 190 degree FOV, and thus provides a suitable approximation to a hemispherical projection of the scene.

In all experiments, full estimates of optical flow were acquired using Lucas and Kanade's gradient-based method [6], in combination with Simoncelli's multi-dimensional paired filter [15] for pre-filtering and gradient estimation. This combination was chosen on the basis of strong performances in a recent comparison of optical flow methods for real-time robot navigation [8].

Two image sequences were constructed for the depth map experiments. In both experiments, the camera's forward velocity was kept approximately constant, while rotation about the axis perpendicular to the ground plane was introduced. No camera calibration was performed or post-filtering of flow

¹Omni-tech robotics.

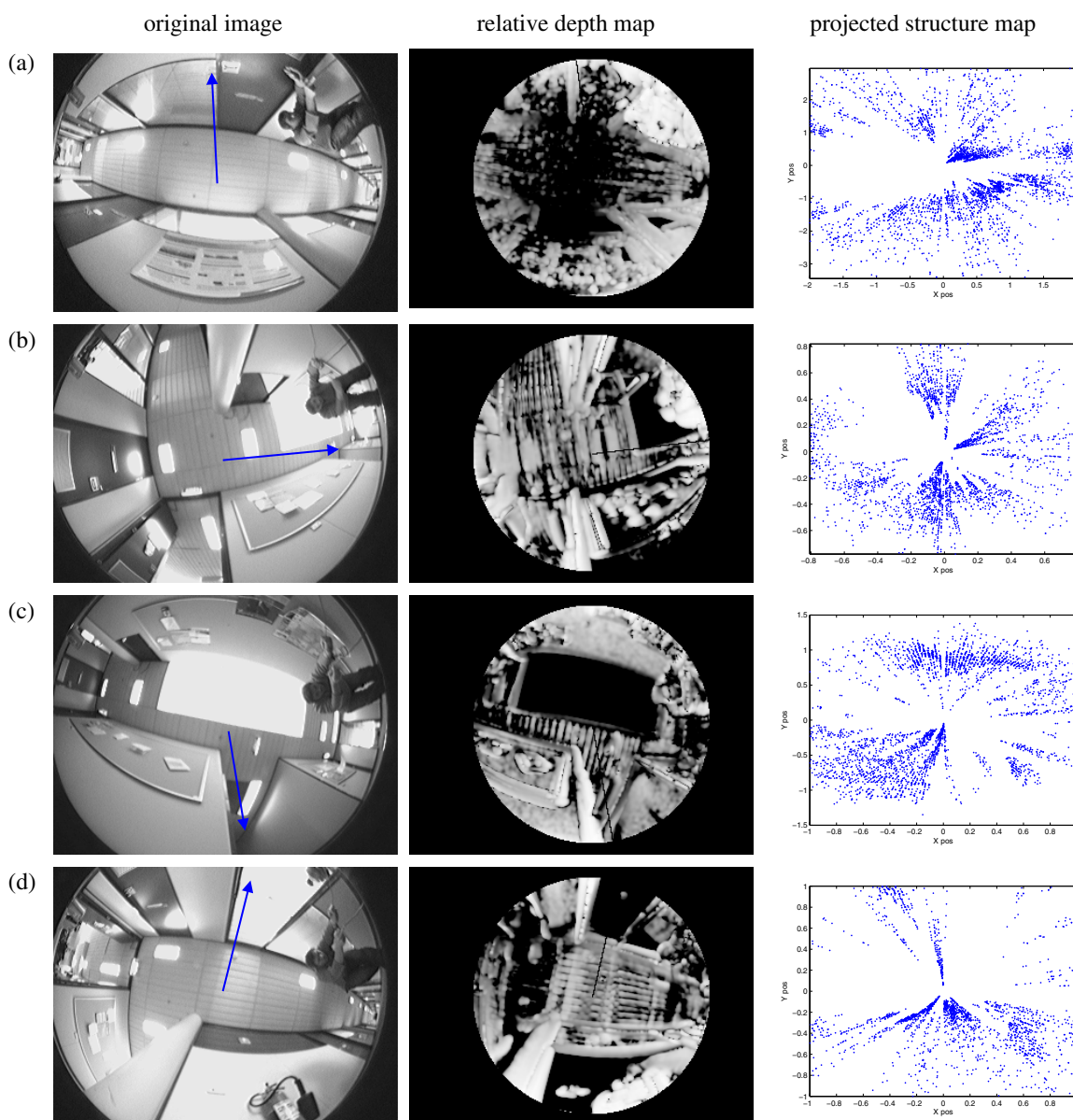


Fig. 3. Sample depth maps obtained on-board the mobile platform (camera facing up). The left column shows the original image, and estimated direction of translation obtained from de-rotation. The middle column shows grayscale relative depthmaps computed from the translational flow. The right column shows structure maps, obtained by projecting relative depth estimates into 3D space, and then flattened out.

estimates, translational direction or relative depth estimates. On a 2.1 GHz computer, depth maps were generated at a rate of 1.2 updates per second over circular image regions of radius 110 pixels, with an input frame rate of 15Hz.

A. De-rotation

Given ground-based motion, the Nelson and Aloimonos de-rotation algorithm was implemented for a single rotation, using a circle of evenly distributed points (radius 110 pixels) around the estimated projective centre of the camera. Due to limitations imposed by the size of the camera's image plane, the true great circle could not be used. For planar motion, any concentric circle about the projective centre is sufficient.

Figure 2 provides a sample frame showing the output

of the de-rotation algorithm over a real image sequence (described later). The blue and yellow points indicate the locations on the circle used for de-rotation. Blue points indicate clockwise flow, and yellow, counter-clockwise. The yellow line signifies the estimated direction of translation after de-rotation is applied. The imbalance of partitioned flow indicates the presence of rotation.

B. Corridor navigation experiment

The first image sequence was obtained by mounting the camera on-board a mobile robot platform, with omnidirectional motion. The camera was fixed as close as possible to the robot's rotation axis, facing upwards. Figure 3 shows sample depth maps obtained during the corridor experiment.

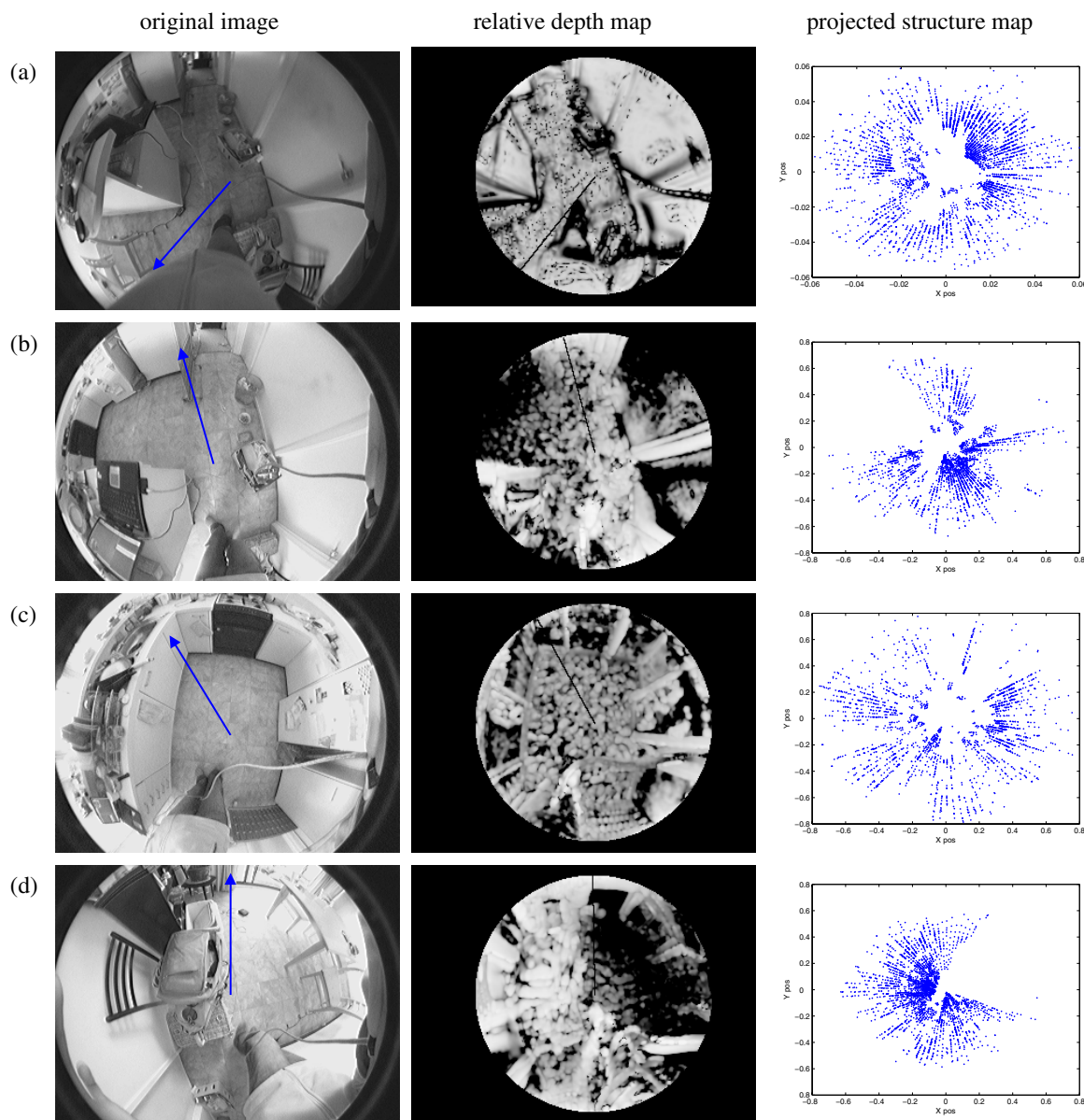


Fig. 4. Sample frames and depth maps from "Belco" kitchen sequence (hand held camera facing towards ground plane). The left column shows the original image, and estimated direction of translation obtained from de-rotation. The middle column shows grayscale relative depthmaps computed from the translational flow. The right column shows structure maps, obtained by projecting relative depth estimates into 3D space, and then flattened out.

The first column shows the central image (320×240 pixels) from the buffered frames used to compute the optical flow for the corresponding depth map. The second column shows a grayscale map of the relative depths of objects in the scene (brighter being closer) estimated from the de-rotated flow field. The third column provides a top-down view of the relative depths of scene points, projected onto the ground plane (we refer to these as structure maps). The centre of the structure map gives the location of the camera. For this, thresholding was applied to extract only the closest surfaces in the scene (and thus omit depth estimates from the ceiling).

The relative depth maps obtained over the corridor navigation sequence provide a good qualitative representation of the environment. An abundance of clear structural cues resulting

from the motion of surface boundaries such as corners, doorways and windows can be seen. This is particularly evident in Figures 3(c) and (d) where the wall edge in (c), and column and fire hydrant in (d) are the brightest objects.

The structure maps in Figure 3 further support the accuracy of the relative depth measures for inferring scene structure. Most evident is the extraction of free space in the local area about the robot. Notably, some surface areas have only a few depth measures associated with them due to a lack of measurable flow in the area.

C. Cluttered environment experiment

A second sequence was constructed depicting the camera's motion through a cluttered kitchen environment. For this, the

camera was hand-held, facing toward the ground plane as it was walked through the kitchen. Figure 4 shows four samples from the results obtained for this sequence.

The depth maps obtained exhibit less structural definition than the corridor sequence. This is unsurprising given the unstructured nature of the environment, and the greater abundance of objects in close proximity to the camera. The camera's orientation towards the ground plane appears to significantly improve the extraction of free space from obstructed space due to the abundance of visual texture. In the corridor sequence, fluro lights, and significantly more distant surfaces reduced the amount of visual texture available. While the improvement is evident in the grayscale depth maps, it is made particularly clear in structure maps like that shown in Figure 4(a), where a detailed map of free space is provided over the viewing area.

D. Discussion

The quality of depth maps obtained in both these preliminary experiments is encouraging. While more quantitative testing is needed, these experiments show that basic 3D scene structure can be reliably inferred from optical flow in real-time. At the very least, these results suggest clear distinctions between free and obstructed space can be obtained, thus supporting the feasibility of this strategy for closed-loop control.

For a ground-based mobile robot, the entire viewing angle may not be needed to generate depth maps sufficient for navigation. In the case of the corridor sequence, only the peripheral view area need be examined to avoid obstacles around the robot. Consideration of both the robot's physical height, and constraints on its motion maybe exploited to improve the speed and accuracy (through higher resolution images) of the depth maps generated.

These results suggest the Nelson and Aloimonos de-rotation algorithm is performing well over real-world images. Both sequences depict significant rotations, yet no ill-effects of this appear evident in the depth maps obtained. While a thorough examination of the algorithm's accuracy over real-image sequences is still needed, it is evident from these experiments, that the algorithm provides sufficient accuracy to facilitate the real-time recovery of both the direction of ego-motion, and complete 3D relative depth maps.

IV. CONCLUSION

In this paper, we have presented a strategy for generating 3D relative depth maps from optical flow, in real-time. In so doing, we have demonstrated for the first time to our knowledge, the use of the Nelson and Aloimonos de-rotation algorithm in real-time, over real images, depicting real-world environments. Results from simulated full general motion of a sphere, and from real-world experiments suggest this strategy may be a useful base for many navigational sub-systems. In addition, these results further support theoretical arguments in favour of a spherical projection when attempting to infer scene structure and self-motion from optical flow.

ACKNOWLEDGEMENT

National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council

REFERENCES

- [1] G. Adiv, "Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 477–489, 1989.
- [2] T. Camus, D. Coombs, M. Herman, and T.-H. Hong, "Real-time single-workstation obstacle avoidance using only wide-field flow divergence," in *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austr., 1996, pp. 323–30.
- [3] J. S. Chahl and M. V. Srinivasan, "Range estimation with a panoramic visual sensor," *Journal of the Optical Society of America A*, vol. 14, no. 9, pp. 2144–50, sep 1997.
- [4] C. Fermüller and Y. Aloimonos, "Geometry of eye design: Biology and technology," in *Multi Image Search and Analysis, Lecture Notes in Computer Science*, T. H. R. Klette and G. Gimelfarb, Eds. Springer Verlag, Heidelberg, 2000.
- [5] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proceedings of the Royal Society of London*, vol. B208, pp. 385–397, 1980.
- [6] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of DARPA Image Understanding Workshop*, 1984, pp. 121–30.
- [7] S. Maybank, *Theory of Reconstruction from Image Motion*. Springer, Berlin, 1993.
- [8] C. McCarthy and N. Barnes, "Performance of optical flow techniques for indoor navigation with a mobile robot," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 5093–8.
- [9] —, "A robust docking strategy for a mobile robot using flow field divergence," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (in press)*, 2006.
- [10] R. C. Nelson and J. Aloimonos, "Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head)," *Biological Cybernetics*, vol. 58, pp. 261–73, 1988.
- [11] J. H. Rieger and H. T. Lawton, "Processing differential image motion," *Journal of the Optical Society of America A*, vol. 2, no. 2, pp. 354–60, 1985.
- [12] J. Santos-Victor and G. Sandini, "Divergent stereo in autonomous navigation: From bees to robots," *International Journal of Computer Vision*, vol. 14, no. 2, pp. 159–77, 1995.
- [13] —, "Uncalibrated obstacle detection using normal flow," *Machine Vision and Applications*, vol. 9, no. 3, pp. 130–37, 1996.
- [14] —, "Visual behaviors for docking," *Computer Vision and Image Understanding: CVIU*, vol. 67, no. 3, pp. 223–38, 1997.
- [15] E. P. Simoncelli, "Design of multi-dimensional derivative filters," in *First Int'l Conf on Image Proc.*, vol. I. Austin, TX USA: IEEE Sig Proc Society, 1994, pp. 790–3.
- [16] M. V. Srinivasan, "How insects infer range from visual motion," in *Visual Motion and its Role in the Stabilization of Gaze*, F. Miles and J. Wallman, Eds. Elsevier, Amsterdam, 1993, pp. 139–156.
- [17] M. V. Srinivasan and S. W. Zhang, "Visual motor computations in insects," *Annual Review of Neuroscience*, vol. 27, pp. 679–696, 2004.
- [18] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 490–498, 1989.