Testing a biologically-based system for extracting depth from brief monocular 2-D video sequences

John A. Perrone
School of Psychology
University of Waikato
Hamilton, New Zealand
john.perrone@waikato.ac.nz

Michael J. Cree
School of Engineering
University of Waikato
Hamilton, New Zealand
michael.cree@waikato.ac.nz

M. Hedayati
School of Engineering
University of Waikato
Hamilton, New Zealand
hedi.hedayati@waikato.ac.nz

Dale Corlett
School of Engineering
University of Waikato
Hamilton, New Zealand
dcorlett@waikato.ac.nz

***Abstract*—Knowledge of the 3-D layout in front of a moving robot or vehicle is essential for obstacle avoidance and navigation. Currently the most common methods for acquiring that information rely on 'active' technologies which project light into the world (e.g., LIDAR). Some passive (non-emitting) systems use stereo cameras but only a relatively small number of techniques attempt to solve the 3-D layout problem using the information from a single video camera. A single camera offers many advantages such as lighter weight and fewer video streams to process. The visual motion occurring in brief monocular video sequences contains information regarding the movement of the camera and the structure of the scene. Extracting that information is difficult however because it relies on accurate estimates of the image motion velocities (optical flow) and knowledge of the camera motion, especially the heading direction.**

We have solved these two problems and can now obtain image flow and heading direction using mechanisms based on the properties of motion sensitive neurones in the brain. This allows us to recover depth information from monocular video sequences and here we report on a series of tests that assess the accuracy of this novel approach to 3-D depth recovery.

***Index Terms*—visual odometry, monocular visual sensor, image motion, depth-from-motion**

I. INTRODUCTION

Humans are very good at extracting 3-D depth information from the visual motion in monocular, 2-D video sequences [1]. This ability is part of an amazing skill whereby we are able to extract information about the world in front of us from just the 2-D motion on the retinae of our eyes [2]–[6]. Currently the mapping out of the environment in front of a moving robot or vehicle requires active sensors (e.g., LIDAR) or binocular systems [7], [8]. Emulating the human ability to extract 3-D depth from 2-D motion in software would have many practical benefits such as sensors for robotics that are lighter and more compact than current systems. It would also reduce the need to process two (or more) video streams simultaneously, as is the case for binocular or multi-camera systems.

The theory behind our ability to determine self-motion and depth from 2-D motion has been known for a long time [2]–[4], [9] but it turned out to be a difficult problem when it is applied to actual video input sequences. A number of challenges such as the ability to accurately measure the image motion slowed progress in this area [10]. On the surface, the derivation of depth from motion looks straightforward and can be conceived as a basic trigonometric problem (Fig. 1a). If the image motion arises from a pure translation of the camera (i.e., no rotation is present) then knowledge of the heading direction (shown as a white circle in Fig. 1b) enables the relative depth of two points to be obtained from the relative magnitude of the two vectors.

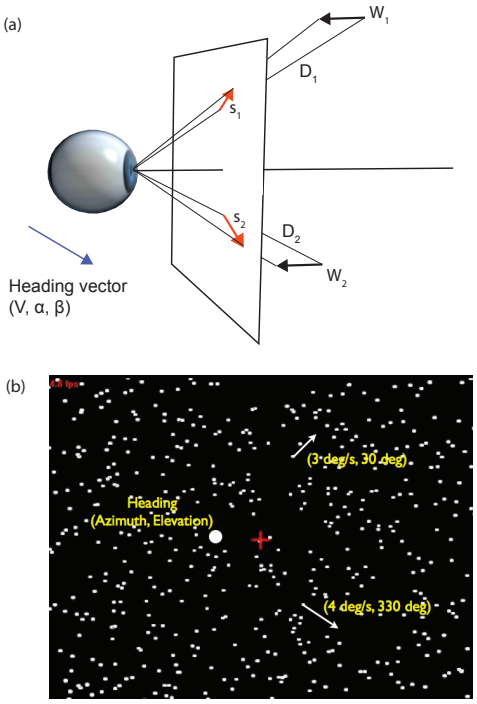

Figure 1. Deriving depth from optical flow. (a) If heading direction (α, β) is known, and w_1, w_2 are fixed in the world, D_1 and D_2 can be found from s_1 and s_2. If V is known, it is possible to obtain absolute values of D_1 and D_2. (b) Knowledge of the heading direction and the two vector magnitudes can provide information about the relative depths of the two dots.

Two problems surface when attempting this exercise however. First, it is difficult to obtain accurate estimates of the image velocities and there is a long history of attempts to do so [11]–[14]. Second, it is difficult to estimate the heading direction from the flow field because it is rare to only have

978-1-7281-0125-5/18/$31.00 ©2018 Crown

a pure translation flow field; for the majority of self-motion scenarios some camera rotation also occurs. The image motion from the rotation ($\mathbf{R}$) gets added to the translation part of the motion ($\mathbf{T}$) and each vector in the flow field is now given by the vector sum ($\mathbf{T}+\mathbf{R}$). It is difficult to find the heading direction in these combined $\mathbf{T}+\mathbf{R}$ flow fields [6], [15].

We have now overcome both of these problems and have developed a system (based on the known properties of cells in the primate visual system) for measuring image motion [10] and for obtaining a pure translation flow field from a combined $\mathbf{T}+\mathbf{R}$ field [16], [17]. The image velocity estimation stage has been described in detail previously [10] and an overview can be found in [18]. One of us recently also described a technique for measuring and removing the R component of the motion during movement along curvilinear paths [16]. In this paper, we are assessing the depth estimation stage of our system and the test sequences contain no rotation components, so we will not specifically discuss the rotation compensation stage of the model. We demonstrate how 3-D depth information can be recovered from an eight frame, monocular video sequence obtained from a camera moving towards a scene with known object distances. The model successfully discriminates between objects at different distances from the camera.

II. MODEL DESCRIPTION

Fig. 2 shows the steps used to extract depth from an image sequence. Eight frames extracted from a video movie (see details below) are first input (step 1) through the velocity estimation algorithm [10]. This method for extracting a velocity vector field ('optical flow field') uses banks of spatiotemporal filters as an initial stage and combines the outputs in a unique way to generate filters that are tightly tuned to the speed of the image motion. The outputs of these speed and direction tuned filters are then combined across different spatial scales to derive a velocity estimate at (x, y) locations that form part of a diamond-based spatial array [10].

The output of the velocity estimation algorithm is referred to as 'raw' in this figure because some of the direction estimates are perturbed away from the correct direction because of the aperture problem. This problem arises where the stimulus patch moving across the motion sensors contains only one orientation and only the motion orthogonal to the edge can be detected [12], [19]. This is corrected at a later stage in our system, once the heading direction has been established.

If a known rotation of the camera relative to the moving platform has occurred, it would be removed (step 2) from the vector flow field at this stage [17]. No rotation was used in the tests reported in this paper and so no rotation compensation was required. The flow field is then passed (step 3) through a curvilinear rotation detection stage [16] which is designed to detect the rotation component of the vectors when rotation is introduced via motion of the camera along a curved path in the world. If curvilinear rotation is detected, it is subtracted from the flow field. In all tests reported in this paper, the curvilinear rotation detected was $0°\text{s}^{-1}$ and so no rotation compensation was applied.

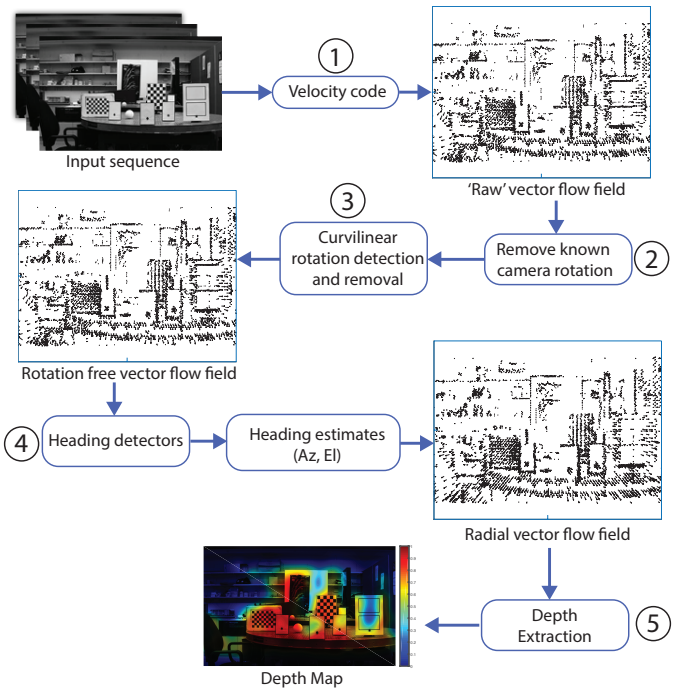

Figure 2. Overview of depth estimation system.

The rotation free flow field is then used as input (step 4) to our heading detector stage [18], [20]. The heading detectors are designed to find the 'Focus of Expansion' (FOE) which is the point in the image out from which all the vectors radiate. This point $(x_{\text{FOE}}, y_{\text{FOE}})$ corresponds to the instantaneous heading direction of the camera [21]. For a set of candidate FOE locations (x_{f_i}, y_{f_i}) spanning $-50°$ to $50°$ azimuth and $-50°$ to $50°$ elevation in $2.5°$ steps, each vector in the flow field at (x_j, y_j) is projected (via a dot product) onto the radial line that joins (x_{f_i}, y_{f_i}) to (x_j, y_j). It is possible to sample a greater range of possible FOE locations [22] but we limited the range for the tests reported in this paper because the heading was constrained to be in the central region of the image. The sum of the dot products is assigned to each 'heading detector' in the map of (x_{f_i}, y_{f_i}) locations. This forms a 2-D distribution with the peak corresponding to the heading direction since the sum of the dot products is maximal when the vectors align with the radial direction out from the actual FOE [20]. In order to obtain greater precision in the heading estimates we threshold the distribution (all values below 0.95 of the peak are set to zero) and use a centroid operation (weighted vector sum) to find the heading vector from the remaining non-zero (x_{f_i}, y_{f_i}) values [16].

Once we have established the estimated heading direction $(\text{AZ}_{\text{est}}, \text{EL}_{\text{est}})$, we know that the correct direction of the vectors in the flow field should all align with the radial directions out from the image point (x_f, y_f) corresponding to $(\text{AZ}_{\text{est}}, \text{EL}_{\text{est}})$. Any perturbation of the vector direction away from the radial direction was most likely caused by the aperture problem and so the corrected magnitude (V_c) of each vector is estimated from $V_c = V/\cos(\alpha - \beta)$ where α is the radial direction and

β is vector direction. This transformation is only applied for values of $\alpha - \beta < 70°$. Vectors outside of these bounds are not passed to the next stage of processing.

Once the radial flow field has been calculated the vectors are passed (step 5) through a depth estimation stage (see Fig. 1a) using standard optic flow equations [22] to derive the absolute depth of the point (X, Y, Z) at image location (x, y) from the vector magnitude, the heading direction and the camera forward speed. The derived depth can be represented in the form of a heat-map (e.g., bottom of Fig. 2). Here we analyse the accuracy of the depth estimation stage by comparing the depth estimates of specific objects and image regions to the known depth of the objects in the scene.

III. TESTING METHODOLOGY

We used a computer-controlled camera (Basler acA1920-150um) mounted on a Pan-Tilt unit attached to an X-Y translation table (Newmark CS Series XY Gantry-1500-1500-1). The camera (field of view = 42° horizontal and 26.3° vertical) moved at 0.25 $\mathrm{m/s}$ towards a laboratory scene containing identifiable target objects (Fig. 3). A series of eight frames (the minimum required to implement the biphasic temporal filters used in the flow field estimation model [10]) of 1920×1200 pixels was extracted from the video stream at a 10 Hz sample rate. The large images were divided into 4×8 sub-movies (each 256×256 pixels) with a 16 pixel overlap, which is for convenience only as our model is currently set up to use 256×256 images. The 32 separate $256 \times 256 \times 8$ frame movies were run through the velocity estimation code individually and the resulting vector flow fields were then stitched together to produce a single flow field for the original full frame movie. The output of the velocity code model develops over the eight frame sequence and we use the output from the fourth frame as an estimate of the vector flow field. Each vector in the composite flow field has a magnitude (in pixels per frame) and direction (degrees). The current implementation of the model is based around 256×256 sized images running in Matlab and is not intended for 'real-time' analysis. The 32 separate movies could be run in parallel on suitable hardware or distributed systems for faster processing times.

Fig. 3 shows the first and last frames of the eight-frame movie sequence with some of the objects that will be used to assess the depth extraction stage of the model. The true camera direction was intended to be $(0°, 0°)$ azimuth and elevation but because of a small misalignment with the X-Y system rails the actual heading location was determined to be $(0.9°, 1.5°)$ using visual calibration (zooming in on the central image region and finding the location with zero radial motion). This offset was removed from the model heading estimates prior to the depth estimation stage.

IV. RESULTS

The raw vector flow field output from the velocity estimation stage is shown in Fig. 4a. Note that because of the diamond spatial array used to sample the image motion [10], slower speeds are represented on a more finely sampled spatial

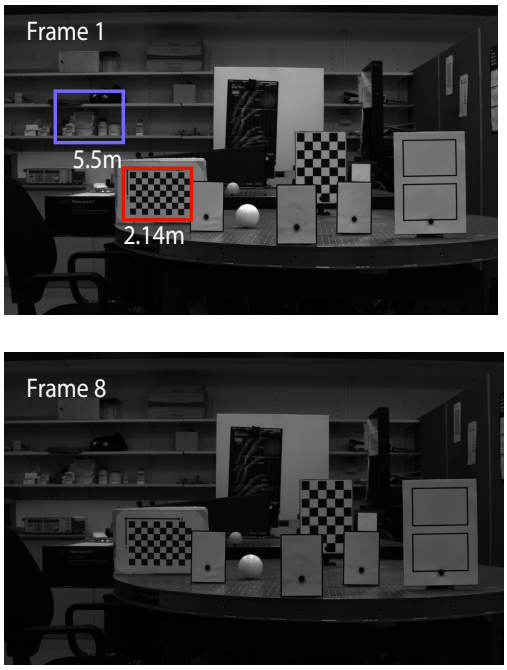

Figure 3. First and last frames from video input sequence used as input to the depth estimation model. The red rectangle indicates an object at a known distance from the camera as well as a region that is some distance away (blue). The ability of the model to discriminate the depth of such regions was tested for a range of separation distances.

array than faster speeds and so the output does not look like that from some other optical flow algorithms that use dense uniform sampling. The first stage flow field was passed through the heading detector array (step 4 in Fig. 2) and the activity distribution from the array is plotted in Fig. 4b. The estimated heading (corrected for the camera misalignment error) was $(0.1°, 0.5°)$ which is very close to the true heading $(0°, 0°)$.

Given the estimated heading direction (and associated expansion point in the image), the actual radial direction of each vector was determined and the vector magnitude was corrected (see Model Description above). The resulting radial flow field is shown in Fig. 5.

The radial flow was used to estimate the distance to each point occupied by a vector in the output field (see Model Description above). Usually knowledge of the forward speed of the camera is an unknown and only the relative distances can be found. For the following tests, we assume knowledge of the camera velocity so that the estimates can be compared to the actual (known) distances of the objects in the scene. The resulting overall cloud of (X, Y, Z) depth estimates occupy a frustum that projects out from the image plane and are difficult to visualise. We therefore clipped the X and Y values within particular window regions of the image (see Fig. 3) and examined the depth values within each zone.

A. Test 1. Three-metre separation in depth

Fig. 6 shows a plan view of the depth (Z) estimates for the two image regions shown in Fig. 3. The red circles correspond

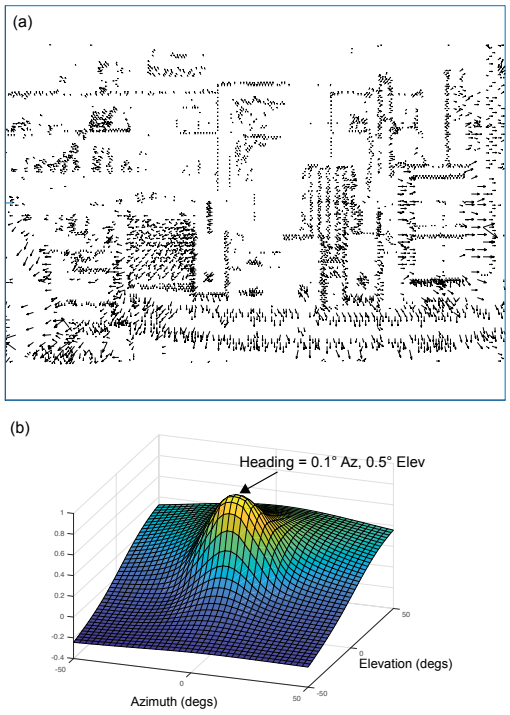

Figure 4. (a) Vector flow field output from the velocity estimation stage of the model in response to the eight frame test sequence. The blank zones at the top and bottom of the field are regions that were not sampled because of the subdivision into 256×256 pixel movies. (b) Output of heading estimation stage of the model. The graph shows the activity of each heading detector (tuned to a particular azimuth and elevation value) in response to the vector flow field shown in (a).

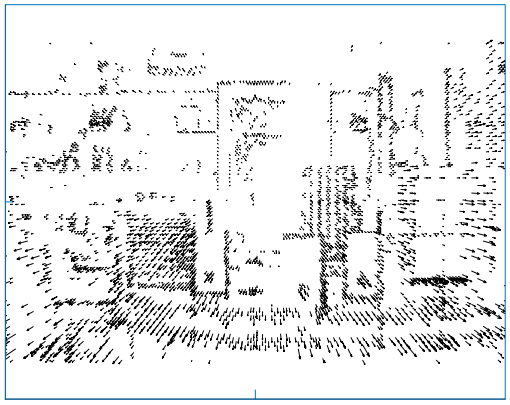

Figure 5. Radial vector flow field output from the velocity estimation stage of the model in response to the eight-frame test sequence.

to the estimates from the closer checkerboard pattern and the blue circles are from the region containing the far wall of the laboratory (approximately 3 m further from camera than the red region). The horizontal dashed lines show the actual object distances from the camera (2.1 m and 5.5 m) for the red and blue regions respectively.

The estimated point cloud is noisy because slight variations in the vector magnitudes can result in large depth variations given the small projection angles involved. However, it is

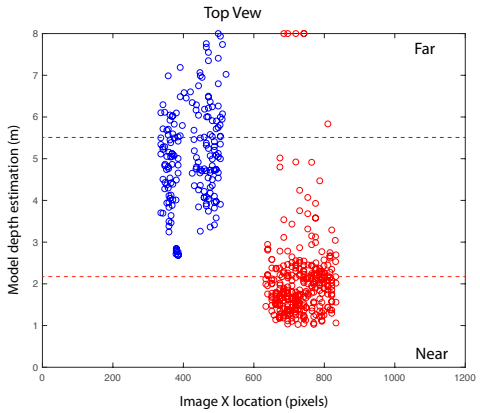

Figure 6. Top view of point cloud depth estimates. X = horizontal, Y = vertical (out of page) and Z is distance from camera and is plotted on the y-axis of the graph. Each circle is a depth estimate for each of the image velocity vectors within the rectangular zones shown in Fig. 3. The horizontal dashed lines show the true distance from the camera for the objects located in the zones. Only the left half of the image is represented in order to provide an enlarged view of the point cloud.

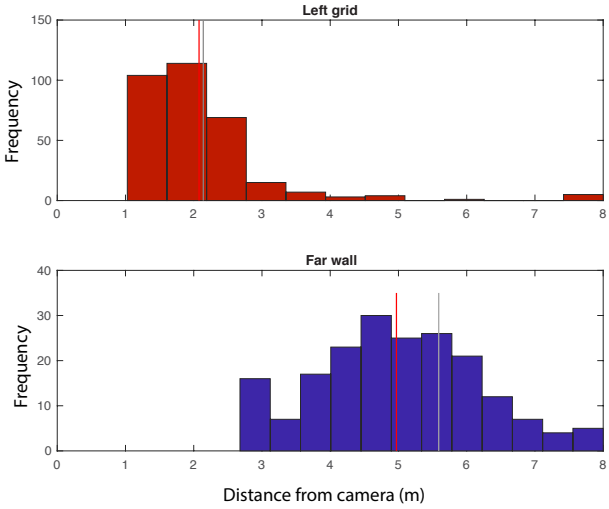

Figure 7. Frequency histograms of the estimated distances (x-axis) found in each of the two image zones (top-red zone, bottom blue-zone). The red vertical lines are the means of the distributions and the grey lines are the actual distances to the objects.

apparent that the red region is closer on average than the blue, far wall region. We binned the estimates along the Z dimension (using the histogram function in Matlab) and the resulting frequency histograms are shown in Fig. 7.

The means (and standard deviations) for the two depth distributions were 2.1 m (1.0) and 4.9 m (1.2). A t-test indicated that these two distributions are significantly different, $t(1, 513) = 29.1$, $p < .001$. Therefore, the model was able to extract depth from the monocular video sequence and successfully identified that the two zones were at different distances from the camera. Because the camera speed was known, we were also able to derive estimates of the absolute distance estimates (in metres) from the camera. The mean estimates for the two actual object distances (see red and grey

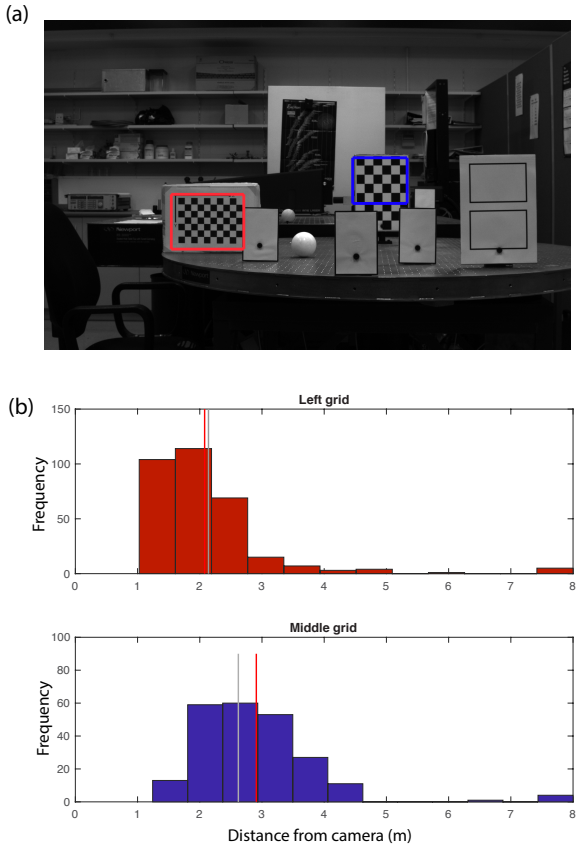

Figure 8. Depth test 2 for objects separated by 0.5 m. (a) The region marked by the blue rectangle was further away from the camera than the red region. (b) Frequency histogram of depth estimates from model for the two regions.

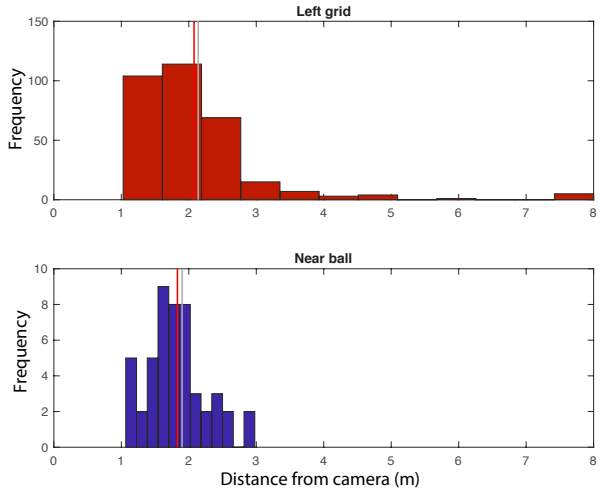

Figure 9. Results from depth test 3 for objects separated by 0.2 m. The test used the nearest ball in the scene (see Fig. 8a) and it was 0.2 m closer than the left grid (red zone in Fig. 8a).

vertical lines in Fig. 7) were close to the true values, especially for the near object with the mean estimate equal to 2.08 m (only a 6 cm error).

This was a reasonably easy test because the separation distance of the two regions was close to 3 m. We therefore selected another object in the scene that was closer to the red region and performed a similar test.

B. Test 2. 0.5 m object separation

Fig. 8 shows the new blue zone for an object located at 2.6 m from the camera with the difference in depth from the red zone object equal to just under 0.5 m.

For the new blue region, the mean of the depth estimates was 2.9 m (1.0). This was significantly different from the red zone, $t(1, 548) = 9.5$, $p < .001$. The model was therefore able to separate out two objects in the scene with a separation of close to 0.5 m. The estimate for the absolute distance of the far grid was also quite accurate (0.28 m error). For our final test, we selected an object that was only separated by 0.2 m from the red object.

C. Test 3. −0.2 m object separation

The mean depth estimate for the blue zone was 1.83 m (0.4) which is closer than the red zone mean (2.1 m). However the spread of the two point cloud distributions are too high for this difference to be detected reliably and the separation was not significant for this condition, $t(1, 369) = -1.63$, $p = 0.1 > 0.05$. It should be noted however that the separation between the ball and the left checkboard pattern is impossible to discern visually when viewing the eight frame video sequence (while fixated at the centre of the frame) and so the model appears to be failing under the same conditions that challenge the human visual system.

V. DISCUSSION

We have shown that 3-D depth information can be recovered from an eight frame, monocular video sequence using a model based on the properties of neurons in the primate visual system. For effective obstacle avoidance, the detection of objects along the path of travel needs to occur very quickly if evasive action is to be executed in time. The temporal filters in the early stage of our velocity detection algorithm have an epoch of around 200 ms [23] and the timeline for the extraction of depth is not much longer than this because the later stages mainly involve integration of the first stage motion signals. Our approach has been motivated by the need to extract depth rapidly and the knowledge that humans can extract depth very quickly from image motion [1]. Our whole system is feedforward only and we would argue that this gives it an advantage over schemes that rely on iterative searches for solving the velocity or odometry stages [7], [8].

Despite our preference for this current feedforward implementation, we do recognise that the output from the later stages of the model could be used to refine the depth estimates over time. For example, we could use the depth distributions (see Fig. 7) to refine future estimates of the extracted depth signals and implement some form of Kalman filtering [24].

Given the difficulty of measuring optical flow accurately [10] the vectors are always going to have a certain amount of noise associated with them. For example, variations in contrast across the scene will affect the speed estimates

and the aperture problem will perturb the direction estimates depending on the orientation of edges in the scene. These errors propagate through to the depth estimation stage and so the depth estimates will always have a reasonable amount of noise associated with them. This is particularly so for locations close to the heading direction where small errors in the heading vector get magnified to large depth errors because of the small angles involved. Therefore, the depth estimates derived using flow fields and monocular video sequences will always have a certain amount of uncertainty associated with them. We have shown, though, that separations of as little as $0.5\,\mathrm{m}$ can be readily distinguished (at least for the camera speed we used) and that such object depth separation could in theory be derived in around $200\,\mathrm{ms}$ (the integration time of the motion filters).

Humans extend their ability to extract depth from self-motion by moving their eyes continuously around a scene. We tend to fixate and follow objects as they get nearer with our eyes [25]. This offers two advantages: first, it slows down the overall image motion and secondly, the eye fixation allows part of the scene to fall on the receptor dense area of our eyes (the fovea). This higher resolution and slower overall image speed means that greater depth discrimination is possible in the fixated region as well as providing a finer spatial sampling of the depth points. Slower speeds are sampled using smaller filters that are sampled more finely. We could emulate this skill by having the camera track particular objects in our laboratory scene and zoom in on the region of interest. Of course, this introduces a rotation component to the motion (since the camera is now tracking a moving object) and this rotation would need to be removed prior to the heading estimation stage of the model (steps 2 and 3 in Fig. 2).

The tests of the model reported in this paper do not include any rotation of the camera which simplified the heading estimation stage (step 4 in Fig. 2). We are currently testing depth extraction in scenarios that include a camera rotation (through both local rotations and via motion along curved paths). This paper represents a first step in demonstrating that a neural-based architecture can extract depth from monocular 2-D video sequences.

VI. CONCLUSION

We have shown that information about the relative depth of objects in front of a single moving monocular camera can be derived from the camera's video stream using a system based on the known properties of motion sensitive cells in the primate visual system. Our algorithm replicates these properties in software and we have demonstrated that such a system is able to correctly identify the relative depth of objects in a scene. For our camera moving at $0.25\,\mathrm{m/s}$ we were able to distinguish two objects separated by $0.5\,\mathrm{m}$ at an average distance of $2.6\,\mathrm{m}$ (19% depth difference). This was achieved using a fully feedforward system and an integration time of around $200\,\mathrm{ms}$. We were thus able to mimic the human ability to rapidly extract depth from monocular video sequences.

ACKNOWLEDGEMENT

This work was funded by MBIE Endeavour Smart Ideas.

REFERENCES

[1] H. Ono and N. J. Wade, "Depth and motion in historical descriptions of motion parallax," *Perception*, vol. 34, no. 10, pp. 1263–73, 2005.

[2] J. 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*, vol. 22, no. 9, pp. 773–791, 1975.

[3] D. N. Lee, *Visual information during locomotion.* Ithaca: Cornell U. Press, 1974, pp. 250–267.

[4] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of moving retinal images," *Proceedings of the Royal Society of London B.*, vol. B 208, pp. 385–387, 1980.

[5] K. Nakayama and J. M. Loomis, "Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis," *Perception*, vol. 3, pp. 63–80, 1974.

[6] W. Warren, *Optic flow*. Cambridge, Massachusetts: Bradford, 2003, vol. 2, pp. 1247–1259.

[7] F. Fraundorfer and D. Scaramuzza, "Visual odometry Part II: Matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[8] D. Scaramuzza and F. Fraundorfer, "Visual odometry Part I: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[9] J. H. Rieger and D. T. Lawton, "Processing differential image motion," *J Opt Soc Am A*, vol. 2, no. 2, pp. 354–60, 1985.

[10] J. A. Perrone, "A neural-based code for computing image velocity from small sets of middle temporal (MT/V5) neuron inputs," *Journal of Vision*, vol. 12, no. 8, 2012.

[11] C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Comput. Graph. Image Process.*, vol. 9, pp. 301–315, 1979.

[12] E. C. Hildreth and C. Koch, "The analysis of visual motion: From computational theory to neuronal mechanisms," *Ann. Rev. Neurosci.*, vol. 10, pp. 477–533, 1987.

[13] B. K. P. Horn and B. G. Schunk, "Determining optic flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[14] K. Nakayama, "Biological image motion processing: A review," *Vision Res.*, vol. 25, no. 5, pp. 625–660, 1984.

[15] D. Regan and K. I. Beverley, "How do we avoid confounding the direction we are looking and the direction we are moving?" *Science*, vol. 215, no. 8, pp. 194–196, 1982.

[16] J. A. Perrone, "Visual-vestibular estimation of the body's curvilinear motion through the world: A computational model," *J Vis*, vol. 18, no. 4, p. 1, 2018.

[17] J. A. Perrone and R. Krauzlis, "Vector subtraction using visual and extraretinal motion signals: A new look at efference copy and corollary discharge theories," *Journal of Vision*, vol. 8, no. 14, pp. 1–14, 2008.

[18] M. J. Cree, J. A. Perrone, G. Anthonys, A. C. Garnett, and H. Gouk, "Estimating heading direction from monocular video sequences using biologically-based sensors," *Proceedings of the 2016 International Conference on Image and Vision Computing New Zealand (Ivcnz)*, pp. 116–121, 2016.

[19] S. Wuerger, R. Shapley, and N. Rubin, ""on the visually perceived direction of motion" by Hans Wallach: 60 years later," *Perception*, vol. 25, no. 11, pp. 1317–1367, 1996.

[20] J. Perrone, "Model for the computation of self-motion in biological systems," *Journal of the Optical Society of America*, vol. 9, pp. 177–194, 1992.

[21] J. Gibson, *The perception of the visual world*. Boston: Houghton Mifflin, 1950.

[22] J. Perrone and L. Stone, "A model of self-motion estimation within primate extrastriate visual cortex," *Vision Research*, vol. 34, pp. 2917–2938, 1994.

[23] J. A. Perrone, "A visual motion sensor based on the properties of V1 and MT neurons," *Vision Res*, vol. 44, no. 15, pp. 1733–1755, 2004.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[25] C. Busettini, G. S. Masson, and F. A. Miles, "Radial optic flow induces vergence eye movements with ultra-short latencies," *Nature*, vol. 390, no. 6659, pp. 512–5, 1997.