Learning to Avoid Indoor Obstacles from Optical Flow

Toby Low and Gordon Wyeth
School of Information Technology and Electrical Engineering
University of Queensland
St Lucia, Australia.
tobylow@itee.uq.edu.au, wyeth@itee.uq.edu.au

Abstract

Optical flow (OF) is a powerful motion cue that captures the fusion of two important properties for the task of obstacle avoidance − 3D self-motion and 3D environmental surroundings. The problem of extracting such information for obstacle avoidance is commonly addressed through quantitative techniques such as time-to-contact and divergence, which are highly sensitive to noise in the OF image. This paper presents a new strategy towards obstacle avoidance in an indoor setting, using the combination of quantitative and structural properties of the OF field, coupled with the flexibility and efficiency of a machine learning system. The resulting system is able to effectively control the robot in real-time, avoiding obstacles in familiar and unfamiliar indoor environments, under given motion constraints. Furthermore, through the examination of the networks internal weights, we show how OF properties are being used toward the detection of these indoor obstacles.

1 Introduction

A fundamental stimulus of vision is formed by changing light patterns across the optic array producing an important vector field called optical flow. Optical flow (OF) fields contain spatio-temporal information about the environment that have been shown to be associated with control and timing actions within nature, such as bee flight control [Srinivasan, 1992], plummeting gannets timing [Lee and Young, 1985], and human balance control [Lee, 1980].

Many of the present visual-based obstacle avoidance methods rely on the spatial properties of a single image [Cheng and Zelinsky, 1998][Lorigo *et al.*, 1997], which although important, lack control, speed and timing information fused within visual-motion properties. Techniques utilising optical flow properties such as divergence [Camus *et al.*, 1996] and time-to-contact [Coombs *et al.*, 1995] [Low and Wyeth, 2005] take advantage of these benefits, but primarily focus on the quantitative aspect of the OF field, which is highly sensitive to OF field noise.

In this paper, we present a novel approach to mobile robot obstacle avoidance by training a neural network to avoid indoor obstacles using both quantitative and structural properties of the OF field. Structural cues of optical flow are less sensitive to noise and indicated by biology to hold important information in addition to its quantitative counterpart. Experiments by Kappers found that humans respond the optical flow spatial structures rather than divergence of optical flow fields [Kappers *et al.*, 1996]. Furthermore, the examination of the V5 or MT area in the visual cortex of a macaque monkey reveals a complete set of direction selective neurons [Albright, 1993]. How the OF structural cues are used for obstacle avoidance remains unclear.

The motivation for using a machine learning approach is four-fold: Firstly, it allows the robot to flexibly incorporate and efficiently process the vast amount of information produced from a visual-motion system. Secondly, visual machine learning systems have been successfully used for robot steering control based on road images [Pomerleau, 1993] and optical flow learning of camera motion vectors [Miyauchi *et al.*, 1993]. Thirdly, the inherent ability of neural networks to generalise system input is a important and desired system feature; as current optical flow algorithms are far from perfect and have many trade-offs between the performance, accuracy and density [Barron *et al.*, 1994]. Lastly, through the examination of the neural network weights, we reveal how the OF field is used within the network for the task of indoor obstacle avoidance.

The neural network training strategy developed comprises of three systems, laser navigation, optical flow generation, and robot heading direction control. Optical flow fields from robot camera images are correlated to a laser navigation heading directions to produce training

samples. OF fields are generated at a rate of 3Hz, with each field mirrored along its y-axis (with its corrected corresponding mirrored navigation velocity), in turn creating two training samples for every frame captured. We show the effectiveness of the system in controlling and timing robot actions to avoid obstacles in a smooth and natural manner, under given motion constraints. In addition, the weights analysis of the neural network shows how obstacle avoidance is being learnt using OF properties. It is acknowledged that solely using visual-motion information for obstacle avoidance has a fundamental flaw if the robot remains stationary. Instead, the premise of the proposed approach is to develop a visual-motion obstacle avoidance technique that can be used in conjunction with current visual-spatial systems, in order to create a robust vision-only obstacle avoidance system.

1.1 Outline

The study presented here consists of two distinct phases. The first phase involves the collection of laser-based navigation behaviours and concurrent measured optical flow patterns, used to train the neural network. The second phase evaluates the performance of the neural network by examining how well the network uses measured optical flow patterns to avoid obstacles in a office environment. Section 2 describes the robot laser-based navigation system. Section 3 shows the visual camera arrangements and optical flow creation process. In Section 4, the neural network architecture and training sample gathering is described. Section 5 shows the overall system framework and experimental setup. The training results and navigational performance of the system are then given in Section 6. To further examine the network, Section 7 performs a internal analysis of the neural network's weights and highlights flow contributions to obstacle avoidance. Lastly the conclusion and future development of this study is given in Section 8.

2 Navigation System

Since the goal of the navigation system is to provide robot heading velocities which are then correlated with an optical flow image, it is important to choose a navigation method that stays 'true' to the visual sensor space. The navigation system must also be reactive, be able to keep the robot moving at consistent speeds, and must maintain smooth and natural motions. To satisfy these constraints, a modified virtual-force-field (VFF) navigational approach was implemented using laser information.

2.1 Virtual Force Field Implementation

VFF's operate by having obstacles conceptually exert forces onto the robot. Magnitude and direction of these forces depend on proximity and position of the obstacles relative to the robot. The sum of these forces produces a resultant force that directs the robot away from obstacles. The laser sensor field on the robot must be truncated to match the visual system FOV and divided into 11 polar-regions with equal degree of separation. The closest laser distance in each region is transformed into force vector using the elliptical potential field equation:

$$V_{ff}(\theta) = r \times \sqrt{\frac{1}{m^2} \times cos^2(\theta) + sin^2(\theta)} \qquad (1)$$

where r = distance to obstacle, θ = angle to obstacle relative to the robot's forward axis, and m = potential field elongation constant. The potential forces perpendicular to the robot forward axis are then summed and scaled, to produce an heading direction toward the region of most free space. To enhance correlation with visual-motion OF information, we further incorporate the robot's linear speed into the potential function, creating speed dependent path generation in the navigation routine:

$$\omega = \sum_{i=0}^{n} \left[k.v \times \left(1 - \frac{V_{ff}(\theta)}{d} \right) \right] \qquad (2)$$

where ω = angular heading direction, n = number of forces, k = constant gain, v = robot's forward speed, d = maximum limit of object distance and V_{ff} = potential force of the vector. The restriction of the VFF sensory space to the visual camera FOV introduces the problem of choosing local minima solutions for navigation, thus allowing trap conditions. That is, if the robot were to enter a corner of a room, the navigation routine will tell the robot to further move into the corner as it is currently has the most free space in the sensory FOV. To alleviate this issue, the lowest potential from left and right regions is identified and used to bias the left and right forces correspondingly. The bias strength is proportional to the square inverse of the object distance to the frontal region of the robot:

$$Bias = \frac{k}{r^2} \qquad (3)$$

where r = distance to object and k a constant gain. To ensure smooth navigation motions, limitations were placed on the forward speed and desired angular heading velocities. As quantitative optical flow information is translational velocity dependent, it is important for the VFF to maintain the same forward speed throughout all manoeuvres. These angular velocity and forward speed restrictions impose a minimum space requirement to perform crash avoidance manoeuvres – sometimes not possible in small spaces. To cope with these situations, a fail safe mechanism was implemented to stop and turn the robot until free space was seen.

Figure 1: The Pioneer robot including a SICK laser, sonar ring, two Basler cameras and the EiMU sensor.

2.2 Wall Following Perturbations

Perturbations in the sensor inputs are essential to improve the neural network's generalisation ability. The lack of varying sensory information makes it difficult for the network to classify new situations. Therefore, varying strength wall following perturbations are implemented within the navigation routine to generate fresh trajectories for training. Due to the flexibility of the VFF methods, wall following can be accomplished by simply adding a wall following gain to either the left or right potential forces depending on which wall was chosen.

3 Vision System

The role of the visual system is to supply real-time and consistent optical flow fields with adequate density, image spatial coverage, and sensor space coverage to the machine learning system.

3.1 Camera Arrangement

To improve the robot's overall image awareness and FOV, two cameras are employed and positioned on the robot with their optical axis pointing 30° outwards from the robot's forward axis. This configuration creates a wider FOV than stereo and monocular systems and a frontal blind spot region for objects within 0.5 m, adequate for obstacle avoidance. It also creates a small stereo overlap (although not used in this study) for distant obstacles located near the robot forward axis. Figure 1 shows a photo of the Pioneer mobile robot and the camera arrangement.

3.2 Optical Flow

Optical flow can be mathematically defined as a set of vectors with each vector describing the motion of features in image space. These vectors provide a 2D representation of the robot's motion and the environment's 3D structure under the correct conditions. The 3D motion field can be mapped onto a 2D image plane through the basic equations of motion [Trucco and Verri, 1998]:

$$\begin{aligned} v_x &= \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}. \\ v_y &= \frac{T_z y - T_y f}{Z} - \omega_x f + \omega_z x + \frac{\omega_y xy}{f} - \frac{\omega_x y^2}{f}. \end{aligned} \tag{4}$$

where $\mathbf{v}$ is the velocity vector in image space, $\mathbf{T}$ is the translational velocity vector in camera space, f is focal length, Z is the depth to the motion point in camera space and $\boldsymbol{\omega}$ is the rotational velocity vector in camera space. Equation (4) shows that the rotational or translation components are linearly separable and that depth information Z for a point only exists if there is translational motion in 3D space, $\mathbf{T}$.

There exists number of optical flow algorithms, with the most popular using either differential, feature-based or phase-based methods. From the study led by Barron et al. [Barron *et al.*, 1994], the Lucas and Kanade (LK) algorithm [Lucas and Kanade, 1981] was found to provide the best trade-offs between accuracy, computational time and density. We employed the LK algorithm with a further improved three-layer pyramidal implementation [Bouguet, 2000], essential when dealing with the expected low OF field frame rates in the system. Even so, real-time optical flow generation still proved difficult at high resolutions. Image resolutions were reduced to 128×96 for each camera, cropped to the bottom half of the image, undistorted using camera properties, and optical flow found for 1536 features (64×24 grid) per image. To help avoid any aperture problems, a window size of 15×15 was chosen for matching of optical flow points. The two 64×24 optical flow fields are then Gaussian filtered (5×5 kernel size, $\sigma = 1.25$), sub-sampled by two (32×12) and combined in their corresponding positions to produce a 64×12 optical flow vector matrix.

As camera rotations cause difficultly in extracting information from optical flow, we remove the majority of rotational components using the robot rotation angle – provided by a complementary filtered wheel encoders and inertial measurement unit (EiMU [Roberts *et al.*, 2004]) information [Low and Wyeth, 2005]. As a final step, a simple error metric that records the number of features not found in the optical flow matching process is calculated.

4 The Neural Network

Neural networks can be described as a group of simulated neurons with weights connecting them. These variable weights allow the network to learn complex functions given a complete training set. Having obtained the optical flow field and correlated navigation heading direction,

a neural network is required to process 768 optical-flow vectors and output a control heading direction for obstacle avoidance.

4.1 Network Architecture

To learn the optical flow fields, a three-layer feed-forward fully interconnected neural network was implemented. Optical flow vectors are fed into the network with outputs chosen to represent heading velocities. All neurons in the network use the *tanh* activation function. A block diagram of the architecture is shown in Figure 2 with detailed layer information given below.

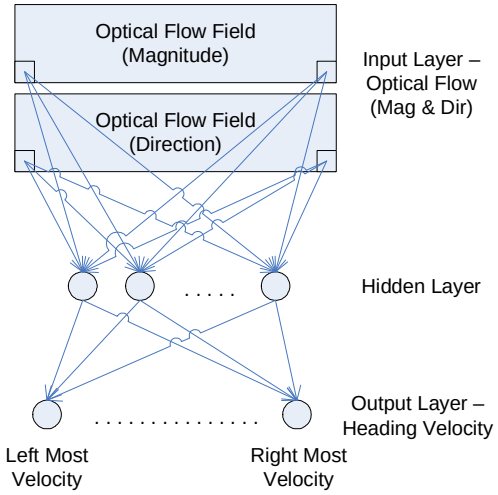


Figure 2: Neural Network Architecture.

Input Layer: Optical Flow Field - The 64x12 vector matrix of optical flow points (representing the collective image fields seen by the cameras) is generated by the visual system, transformed to polar coordinates and transformed into a 1536-dimensional vector. The input layer of the network uses this single 1536-dimensional vector to represent the optical flow vector field. The use of polar coordinates helps the weight examination of directional properties presented later.

Hidden Layer: 8 Units - The number of hidden layer neurons depend on several factors such as image size, number of input/output neurons and the optical flow field patterns (e.g., the larger the number of inputs increases the number of pattern possibilities, thus requiring a larger number of hidden neurons). Through trial, error and visual analysis of NN weights, eight (8) neurons were examined to perform well with regard to generalisation and over-learning trade-offs.

Output Layer: Heading Direction - The output layer consists of 31 neurons, each representing the desired heading direction angular velocities with 1°/s separation. Angular velocities are limited to ±15°/s to help keep motions smooth and flow fields error-free. Note that the neural network does not encompass the robot's forward speed, as it was set to a constant value during robot navigation. Thus during neural network evaluation, the robot speed is set to the same speed chosen in the VFF navigation training routine.

4.2 Network Training

A neural network's ability to learn relies on a training algorithm. The popular Back-propagation training algorithm can solve a multitude of complex functions by comparing the network's output response from a input, to the correct output response. The resultant error is then used to modify weights to reduce this error. For this study, an improved Back-propagation algorithm called R-prop is employed.

Back-Propagation Vs. R-Prop

Back-propagation trains the neural network by using gradient descent on the network's error surface. The magnitude and direction in weight space is determined from the partial derivative of the error with respect to the weight. The R-prop algorithm [Riedmiller, 1994] is similar to Back-prop but eliminates the possibly harmful influence of the partial derivative size. The R-prop algorithm only uses the sign of the derivative together with a weight-specific update value Δ_k. New updates values Δ_k are calculated using a learning rate adaption process that speeds up convergence in shallow error regions and slows down learning when jumping over minima.

Training Sample Details

Input and output training vectors are obtained from the optical flow image and corresponding VFF laser navigation output velocity respectively. Optical flow input is represented in a 1536-dimensional vector, whilst the correct training output is specified in a 31-long output array (representing -15 to 15°/s). For example, for an output heading direction of 0°/s, a 31-long null array is generated and a '1' placed at the 16th array position. To improve and aid the network learning process, a Gaussian filter was employed to smooth the heading output array. The final 1536-dimensional optical flow vector together with its correlated 31-dimensional Gaussian smoothed output vector is used as a single training sample.

Since optical flow fields are spatially consistent when mirrored about their vertical axis, an extra training sample is gathered using the new mirrored optical flow field and the correspondingly mirrored output velocity array. This effectively doubles the number of situations the network learns as well as helping reduce any learning bias

towards the training environment. Optical flow's fundamental flaw is that it requires adequate motion in order to be useful. Training samples obtained in either low velocity, high rotation or low-light situations can affect convergence of the neural network and the overall performance of the robot system. Thus, the robot was forced to maintain a consistent speed of 200mm/s during all manoeuvres. To ensure good training performance, training samples gathered at forward velocities less than 150 mm/s and erroneous rotation rates greater than 30°/s are removed. Moreover, if the combined optical flow vector field has greater then 150 unmatched features points, the training sample is flagged and ignored.

5 Experimental Setup

Neural network learning experiments are conducted offline using the FANN library package [Nissen, 2006] on a Pentium 4, 2.8GHz computer. Real robot experiments are conducted using a upgraded P2DXE Pioneer Robot. Sensors include a sonar ring, wheel encoders, two Basler colour cameras, a SICK laser with an 180° FOV and an inertial measurement unit (EiMU). The sonar sensors are not used and are disabled on the robot. The robot is able to capture, compute optical flow and interpret neural network outputs at a rate of 3Hz.

5.1 Overall System Framework

The complete system framework is shown in Figure 3. The system operates to either collect training information for the network (SW1 - on, SW2 - right) or control the robot using neural network once trained (SW1 - off, SW2 - left). The laser VFF navigation technique generates motor velocity commands for the robot to avoid obstacles. Optical flow fields are then created from the camera images. The flow field is processed by the neural network for either learning or control purposes.

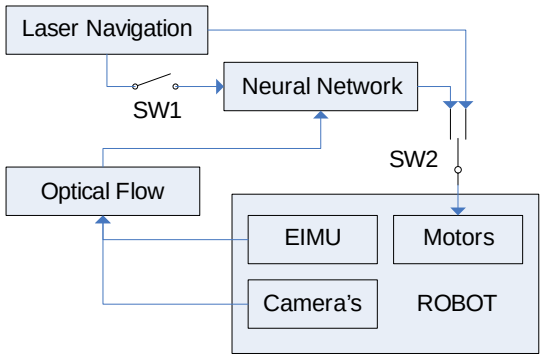


Figure 3: System Framework. SW1 - on, SW2 - right, for training and vice versa for neural network control.

5.2 Training Collection and Setup

A training sample set is generated from the robot navigating around an indoor office environment. Figure 4 shows an example of the robot vision (two camera images stitched side by side) of the office environment before rectification and optical flow processing.

Figure 4: Office Image - robot view.

The robot was systematically perturbed by 12 different strength wall-following gains. After sample filtering and mirroring, approximately 90,000 useful training samples were presented to the neural network. An additional test set of ∼17000 samples was also generated from the same office space using random wall-following perturbations. This test set is essential for determining the networks generalisation ability. Network weights are initialised using Widrow-Nguyen's algorithm [Nguyen and Widrow, 1990].

6 Experimental Results

In this section, we present the results of the training conducted for the neural network, as well as real robot tests in a office and lab environment.

6.1 Training Results

The trained network converged to MSE of approximately 1.8 in around 5 epochs, with further epochs only showing very minor improvements. Performance of network is evaluated by comparing the chosen heading velocities against their corresponding correct heading velocities in the training samples. The chosen heading velocities were calculated from the highest activated neuron.

A detailed error graph comparing the training and test set classification results is shown in Figure 5. The majority of errors were found in the ±1°/s output velocity error band. Further velocity errors were seen to decrease in a linear fashion, to a max error of 19°/s. In an error margin of ±5°/s, a total of 71.5% of training samples were classified correctly.

Training performance results were slightly better with the sample test set data, with 79.6% classifications found within a ±5°/s output error margin. This demonstrates that the network has not over-learnt training samples and has the ability to generalise new inputs well. In comparing the positive and negative output errors, a slight bias is seen towards negative output velocities in both training and test sample sets.

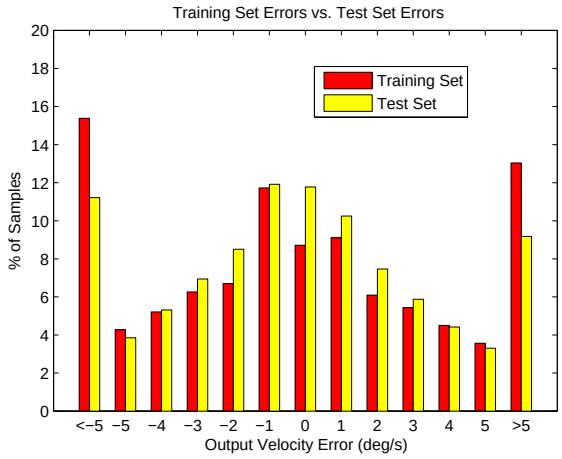


Figure 5: Training Vs. Test Set Error Graph.

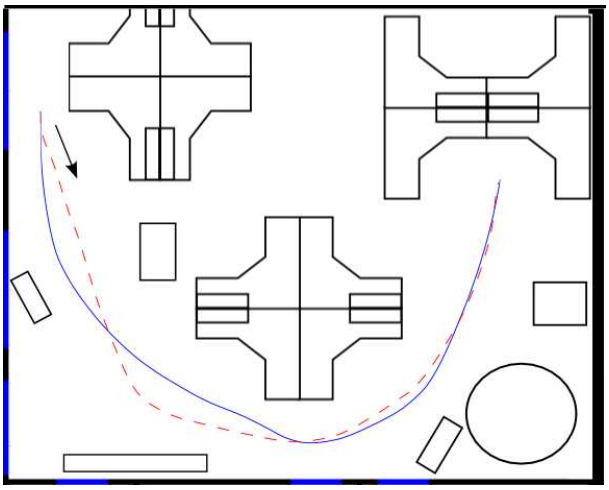


Figure 6: Office Trial 1: VVF Navigation Path - Dotted/Red, NNOA Path - Solid/Blue.

6.2 Navigation Performance

The numerous differences between navigational techniques sensory input information and their dependence upon a wide variety of situations makes it very difficult to compare their performances. In the previous section, the neural network based obstacle avoidance (NNOA) system was only evaluated offline against a sample test set gathered using the laser VFF technique. Although a number of errors were found within the offline tests, it is hard to quantify these error with regards to system performance, as these errors may not necessarily lead to failures in real robot tests.

Training Environment

To further demonstrate the NNOA system abilities and performance, we conducted a number of online indoor office navigational trials. Despite the optical flow generation difficulties within this domain, the network was seen to perform successful obstacle avoidance and free-space seeking manoeuvre in a variety of situations. For example, if the robot was found travelling parallel to a wall with free space on the other side, the robot would adjust its heading towards the free space. Figure 6 shows a successful navigational trial, with the solid blue and red dotted lines showing the paths taken by the robot using the laser VFF navigation and NNOA system respectively.

Comparing the two paths, a number of conclusions can be drawn. Examining the NNOA path, it can be seen that the reactions to obstacles are delayed compared to the VFF path. This can be attributed to the visual range limitations imposed from the inherent properties of optical flow generation from 2D images. Optical flow information about an object degrades significantly with respect to the objects distance. The objects further away contain very subtle optical flow information that the neural network cannot learn, thus the system can only react to obstacles within a proximity that generates decent optical flow. Examining the VFF path, as expected, the greater range of the laser technique ensures the robot takes a safer path between the surrounding obstacles.

Figure 7 shows another successful trial run performed by the NNOA system. The robot starts on a trajectory initially heading towards left-side obstacles and manoeuvres slightly to the right to just miss these objects. Consequently, the robot is put on a collision course with a right side obstacle, which the NNOA system was able to avoid by making significant turn towards the left. In this trial it is interesting to note a general bias favouring the avoidance of right side obstacles, consistent with offline training results.

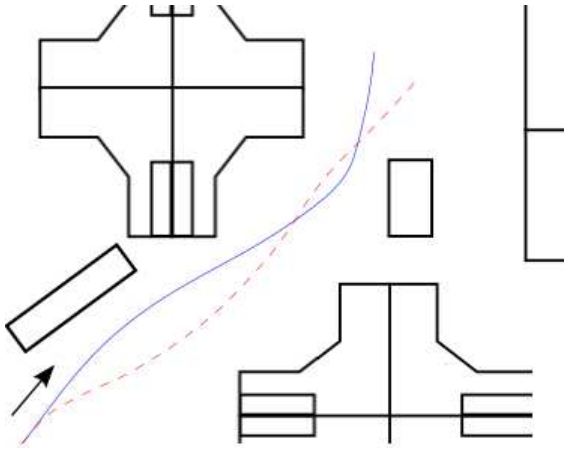


Figure 7: Office Trial 2. Refer to Figure 6.

In the robot navigation trials conducted, we examined a few conditions that the NNOA system could not handle. Direct head-on paths to obstacles were one such

condition – mainly due to the fact that these paths were not encompassed within the training set. The network would generalise a head on collision into the closest related flow field, that of a forward heading direction, thus causing a failure. In addition, low incident approach angles to obstacles resulted in collisions due to the avoidance manoeuvres being unable to be executed in time. Figure 8 shows a crash example, where the robot starts off towards a number of obstacles, veers right to position itself away from the desk, putting the robot on a sharp incident approach angle to obstacles lined against a wall. The NNOA system detects these obstacles and starts to veer left, but is unable to avoid collision in time given the motion constraints imposed on the robot.

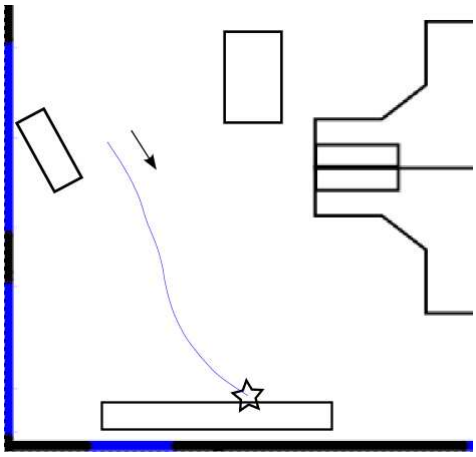


Figure 8: Office Crash Trial. Refer to Figure 6.

Unseen Environment

The key motivation behind using optical flow coupled with a machine learning system is to take advantage of the strong generalisation abilities inherent in both techniques. To test these abilities, online robot tests using the NNOA system were also conducted in an unseen lab environment. Figure 9 shows an image representing the robot's vision of the lab. There are clear differences in floor textures (vinyl versus carpet) and the surrounding objects.

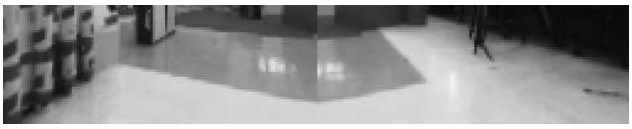


Figure 9: Lab Image - robot view.

Results of the robot lab tests were similar to those of the office tests. Figure 10 shows a successful lab trial where the robot displayed similar manoeuvres to those seen the office test 2. The laser VFF method again chose the path of optimum space between the obstacles, whereas the NNOA system avoided obstacles in reactive manner, relying solely on the build up of optical flow information for navigation and control. Although the NNOA trajectories do not match up with the laser-based VFF navigation paths, the NNOA system produces a path that of least energy in performing the prescribed goal of driving straight forward and avoiding obstacles.

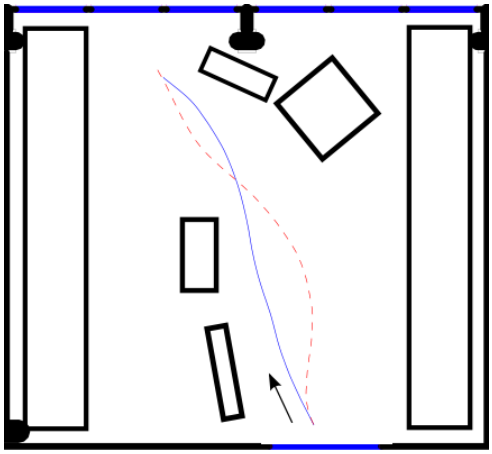


Figure 10: Lab Trial. Refer to Figure 6.

The reproduction of these results within a completely new environment demonstrates the NNOA strength in generalisation. The smooth obstacle trajectories also show the systems ability to use optical flow information for control of the mobile robot. Furthermore, these results demonstrate the excellent ability of neural network system to cope with inherent optical flow sensor limitations for robot obstacle avoidance and control.

7 Weights Analysis

The successful obstacle avoidance control manoeuvres demonstrated in a number of situations in real robot tests spawns the question – how is optical flow field being learnt for the task of obstacle avoidance within this 'black box' system. To answer this question, we performed qualitative weight analysis on the networks hidden units. Weight analysis is important in achieving understanding of systems limitations and in particular, gives us insight into the use of qualitative optical flow cues for the task of obstacle avoidance. To aid analysis, weight diagrams were created for each of the eight (8) hidden neurons. Weights from the optical flow inputs into these hidden neurons are mapped according to joined image space seen by the robot's two cameras. Weights from the hidden unit to output neurons are also mapped in relation to the image space steering directions. These two mappings give a snapshot of the hidden units favoured inputs and outputs within the neural net-

work. When examining these mappings, it is important to note the camera arrangement as it will affect the way flow vectors are learnt within the neural network.

Weight diagrams revealed a number of distinct hidden units, each with a specialised heading output, such as a forward or left/right heading. Table 1 shows the output heading contribution classes and the number of neurons found in that class.

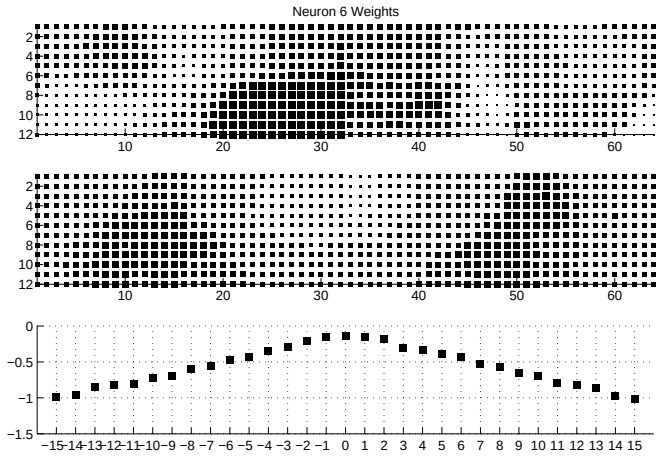
Table 1: Hidden Neurons Contribution.

Neuron Contribution	No. of Neurons
Forward	2
Strong Left/Right	2
Weak Left/Right	3
Scattered	1

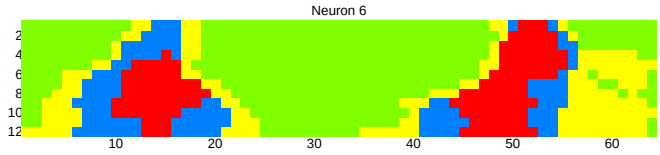
7.1 Forward Contributing Hidden Neuron

Figure 11(a) presents the weight diagram for hidden neurons 6. The top graph shows the weights into the hidden unit from the magnitude input of optical flow, spatially arranged to reflect the image seen by the robot. The second graph from the top shows the weights for the direction input of optical flow in the same arrangement as magnitude weights. Last graph shows the weights from the hidden neuron to the network outputs, spatially arranged according to the image space (that is, leftward heading = negative output heading). The weight sign is colour coded; light/yellow neurons = negative weights, dark/black neurons = positive weights. Weight values are reflected by the size of the squares, larger squares represent a larger weight value and vice versa. In all neurons, we examined that directional weight values were approximately half the weight of magnitude weight values. Thus these directional weights look to play the vital role in providing neuron strengthening and noise reduction, especially when flow magnitude information is inadequate.

As can be seen from Figure 11, hidden neuron 6 is a forward biased neuron. The magnitude weights of this neuron lacks clear structures but still give a general idea behind the processing behind optical flow magnitudes. The first feature noticed in the magnitude weights is a region of large excitatory weights found at the centre of the joined image. Large magnitude flows at the centre vision (such as those from close forward obstacles) will cause a high activation of this forward heading neuron, which is incorrect in regards to obstacle avoidance. These inconsistent central weights are hypothesised to be the outcome of the lack of head-on collision training. With the lack of these head-on collisions, the neuron only learns samples where central region magnitudal flows are small. Head-on collision samples were not included as



(a) Hidden Neuron Internal Weights.



(b) Direction Weights Contour Image.

Figure 11: **Forward Contributing Hidden Neuron:** Weight Diagrams for Hidden Neuron 6 (N6). (a) *Top & Middle* - input weights into the hidden neuron from magnitude and direction arrays of optical flow respectively. Black/dark and yellow/light colour represent $+ve$ and $-ve$ weight values respectively. Larger squares represent a larger magnitude and vice versa. *Last* - output hidden neuron weights to the heading neurons. The top three graphs are organised in an array corresponding to the input optical flow joined image plane. (b) Contour image of direction weights, highest to lowest = red,blue,yellow,green (best viewed in colour).

the recovery manoeuvres at the detected ranges are not possible given the robot motion constraints. The robot can only execute a stop and turn to recover, thus losing optical flow information for guidance and recovery (optical flow's key flaw).

We can further see weight changes along the verticals located at the x-axis points 15 and 47. These weight changes subtlety highlight object boundaries seen in the image space. The weights beyond these boundaries (left of point 15, right of point 47) quickly regain strength, thus any large magnitudes generated from obstacles seen on far left and right sides (beyond these gradient boundaries) will activate the forward neuron/

The direction weights also seem to have a distinct pattern in weight gradient changes. To aid in the changes seen, we constructed a direction weights contour image shown in Figure 11(b). Here weights are sorted into four

ranges and are assigned a colour. Largest to smallest weights are shown in red, blue, yellow and green respectively. The contour image clearly shows two triangular shaped regions of interest positioned at (13,8) and (50,8). These direction weight gradient structures are examined to strengthen flow directions contained within these triangular patterns. Thus it highlights the important object boundaries in image space where key optical flow directional changes are detected (that is, flow changes seen between obstacles and the ground plane). The subtle changes in flow directions along these direction weight gradient lines helps strengthen this forward heading neuron accordingly.

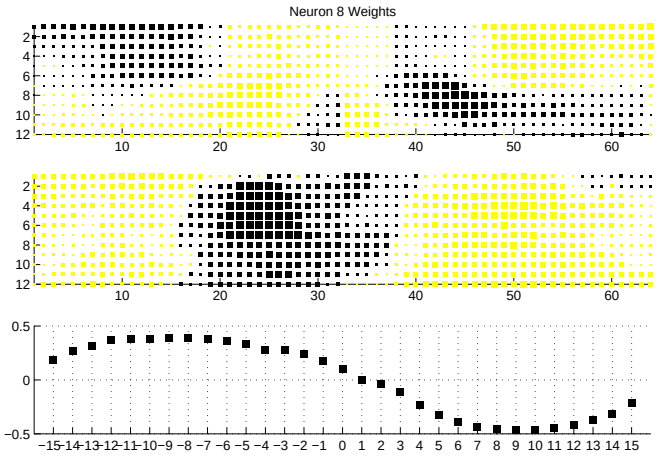
Another subtle aspect is examined within the directional weights, which undermines the functional simplicity of this neuron. As examined before, we can attribute the inner gradients to the detection of left and right obstacle boundaries. On the other hand, we can further attribute the outer gradients seen at the peripheries of the image to the robot driving between long obstacles, such as walls or office desks. The outer weight structures outline the boundaries between the wall and the floor to form the outer line of the triangular structure seen in the contour image. The dual or multiple purpose of neuron weight structures is something examined throughout all the NNOA system's hidden neurons.

7.2 Left/Right Contributing Hidden Neuron

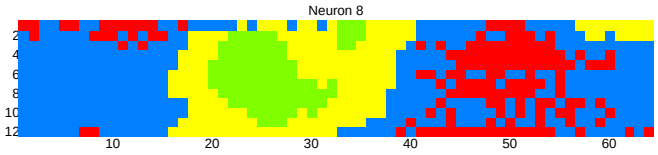
Figure 12 shows the weight diagram for hidden neuron 8, used in determining left or right heading outputs. The left/right biased hidden neuron is much more complex than the previously presented forward neuron. The neuron weight structure includes positive and negative weights that determine its function as a left or right heading biasing neuron.

Here magnitude weights show a structure that highlights a number of important features. Firstly, inverted symmetry can be seen between the left and right image weight space as well as in the output weight space. Secondly, magnitude weight changes are again seen to correspond to obstacle boundary structures. Given a obstacle approaching within the far right (55,4) image region, the large flows seen (compared to the ground plane seen across the signed weight boundary) will excite this hidden neuron outputs to have a leftward heading bias. Due to the inverted symmetry displayed in the weight structure, a rightward heading neuron can be produced in a similar fashion. Inhibiting or negative weights within the left obstacle region (10,4) will negate the hidden neurons output weights when large flows are present, thus creating a rightward heading bias.

The left/right direction weight contour image again highlight the important obstacle boundary positions required for the hidden unit activation. If large changes in flow directions (that is, obstacle boundaries located at those positions) are seen along the left side gradient or the right side gradient, the neuron will change its output bias to the right side or left side heading respectively.



(a) Hidden Neuron Internal Weights.



(b) Direction Weights Contour Image.

Figure 12: **Right/Left Contributing Hidden Neuron:** Weight Diagrams for Hidden Neuron 8 (N8). Refer to Figure 11 for graph details.

Overall, the neurons weight structure demonstrates two possible processing techniques learnt by the neuron; obstacle region finding and obstacle boundary detection. Similar but less obvious region and boundary outlines can be seen in the remaining hidden neurons magnitude and direction weight structures. A few neurons are seen to merge a number of varying obstacle boundaries and influence a corresponding shift in output heading velocities, thus hypothesised to be transitory neurons to help manoeuvre the robot away from obstacles in the smooth and natural manner. Furthermore, we also examined the leftward bias seen within training results and robot results, with three left/right hidden neurons having output weights slightly biased towards leftward motions.

8 Conclusion and Future Work

We have presented a novel approach to obstacle avoidance that relies on optical flow and a machine learning system. Office and lab robot trials demonstrated the NNOA systems ability to perform smooth, reactive and natural motions for navigation and obstacle avoid-

ance, under given motion constraints. A key advantage of our approach is the ability to generalise and perform obstacle avoidance in new environments. In particular, the NNOA system encompasses the complete perception/action cycle, thus further uses optical flow to provide smooth and reactive motions on the robot. In addition, we presented an analysis of hidden neuron weight structure, which highlighted optical flow region finding and obstacle boundary detection processes occurring within the neural networks hidden neurons. In particular, the analysis showed that obstacle avoidance relied on important structural cues of obstacles and their location within the image space.

8.1 Future Work

The nature of metrics and methods employed in the proposed approach enables the possibility of expanding the system to cope with outdoor and dynamic environments. Furthermore, the neuron weight analysis highlighted system flaws and findings that can used to help further improve the overall system. As a final step, the combination of the approach with existing spatial-image obstacle avoidance techniques is necessary, in order to create a fully robust visual obstacle avoidance system.

References

[Albright, 1993] Thomas D. Albright. *Cortical processing of visual motion.*, volume 1, chapter 9, pages 177–201. Elsevier, 1993.

[Barron *et al.*, 1994] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *IJCV*, 1994.

[Bouguet, 2000] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Technical report, Intel Corperation, 2000.

[Camus *et al.*, 1996] Ted Camus, David Coombs, Martin Herman, and Tsai-Hong Hong. Real-time single-workstation obstacle avoidance using only wide-field flow divergence. In *Pattern Recognition*, volume 3, pages 323 – 330, August 1996.

[Cheng and Zelinsky, 1998] Gordon Cheng and Alexander Zelinsky. Goal-orientated behaviour-based visual navigation. *International Conference on Robotics and Automation.*, pages 3431–3436, May 1998.

[Coombs *et al.*, 1995] David Coombs, Martin Herman, Tsai Hong, and Marilyn Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. *IEEE*, 1995.

[Kappers *et al.*, 1996] Astrid M. L. Kappers, Susan F. te Pas, and Jan J. Koenderink. Detection of divergence in optical flow fields. *Optical Society of America.*, 1996.

[Lee and Young, 1985] David N. Lee and David S. Young. *Brain mechanisms and spatial vision.* Dordrecht and Boston and Nijhoff., 1985.

[Lee, 1980] D. N. Lee. The optic flow field: The foundation of vision. *Philosophical Transcations of the Royal Society of London. Series B, Biological Sciences.*, 290(1038):169 – 178, 1980.

[Lorigo *et al.*, 1997] Liana M. Lorigo, Rodney A. Brooks, and Eric L. Grimson. Visually-guided obstacle avoidance in unstructured environments. *IROS*, 1997.

[Low and Wyeth, 2005] Toby Low and Gordon Wyeth. Obstacle detection using optical flow. *Australasian Conference on Robotics and Automation*, pages 1–10, 2005.

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

[Miyauchi *et al.*, 1993] Minami Miyauchi, Masatoshi Seki, Akira Watanabe, and Arata Miyauchi. Interpretation of optical flow through complex neural network. *International Workshop on Artificial Networks.*, 686:654–650, June 1993.

[Nguyen and Widrow, 1990] Derrick Nguyen and Bernard Widrow. Improving the learning speed of 2-layer neural networks by choosinginitial values of the adaptive weights. *International Joint Conference on Neural Networks*, vol.3:21–26, June 1990.

[Nissen, 2006] Steffen Nissen. Fast artificial neural network library (fann), 2006.

[Pomerleau, 1993] Dean A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance.* Kluwer Academic Publlishers, 1993.

[Riedmiller, 1994] Marin Riedmiller. Rprop - description and implementation details. Technical report, Institut fur Logik, Komplexitat und Deduktionssyteme, University of Karlsruhe, January 1994.

[Roberts *et al.*, 2004] Jonathan Roberts, Peter Corke, and Leslie Overs. *EiMU - User Manual Version 0.48 (Firmware version 00.21).* CSIRO Manufacturing and Infrastructure, PO Box 883, Kenmore, QLD 4069, Australia, January 2004.

[Srinivasan, 1992] Mandyam V. Srinivasan. How bees exploit optic flow: Behavioural experiments and neural models. *Philosophical Transactions: R. Soc. Lond. B*, 1992.

[Trucco and Verri, 1998] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.