

# Improved Signal To Noise Ratio And Computational Speed For Gradient-Based Detection Algorithms

Nick Barnes <sup>1,2</sup>

<sup>1</sup> National ICT Australia, Canberra, ACT 2601, AUSTRALIA

<sup>2</sup> Dept of Information Engineering, Australian National University,  
Nick.Barnes@nicta.com.au

**Abstract**—Image gradient-based feature detectors offer great advantages over their standard edge-only equivalents. In driver support systems research, the radial symmetry detection algorithm has given real-time results for speed sign recognition. The regular polygon detector is a scan line algorithm for these features facilitating recognition of other road signs such as stop and give way signs. Radial symmetry has also been applied to real-time face detection, and the polygon detector is showing promising results as a feature detector for SLAM. However, gradient-based feature detection is more sensitive to noise than standard edge-based algorithms. As the total gradient magnitude at a pixel decreases, the component of the gradient at that point that arises from image noise increases. When a pixel votes in its gradient direction out to an extended radius, its position is more likely to be inaccurate if the gradient magnitude is low. In this paper, we analyse the performance of the radial symmetry and regular polygon detector algorithms under changes to the threshold on gradient magnitude. We show that the number of pixels correctly voting on a circle is not greatly reduced by thresholds that decrease the total number of pixels that vote in the image to 20%. This greatly reduces the noise component in the image, with only slight impact on the signal. This improves the performance, particularly for the regular polygon detector where the voting mechanism is complex and constitutes a large amount of the processing per pixel. This facilitates a real-time implementation, which is presented here.

## I. INTRODUCTION

A key technological goal in road vehicles today is to improve safety. One way this can be achieved is by creating systems within the vehicle that support the driver in reacting to changing road conditions. In our research we are particularly concerned with driver support systems. Systems that support the driver in controlling the car, but keep the driver in the loop. Within driver support systems, it is important to consider that roads are highly structured environments, designed to simplify the driving task where possible.

Sign recognition is an important task for a driver support system. Signs giving information that is relevant to the local conditions appear clearly in the environment, however, a driver may not notice a particular sign due to distractions or lack of concentration. In this case it may be helpful to make them aware of the information that they have missed. Previously, we have applied gradient-based detectors to sign recognition. We have shown real-time results for detecting speed signs [1], and demonstrated detection of signs that have a regular polygonal shape [2], for example stop signs, give ways signs, etc. These results are possible due to exploitation of directional gradient

information in shape detection. Gradient-based feature detectors also have shown advantages in other fields, including face-detection [3] as part of SeeingMachine's FaceLab<sup>TM</sup>, and as a feature detector for Simultaneous Localisation and Navigation [4]. The advantages of gradient information for efficient line finding has been known for many years [5].

Image gradient-based feature detectors offer important advantages over their standard edge-only equivalents. However, gradient-based feature detection is more sensitive to noise. As the total gradient magnitude at a pixel decreases, the component of the gradient at that point that arises from image noise increases. Thus, when a pixel votes in its gradient direction out to an extended radius, its position is more likely to be inaccurate if the gradient magnitude is low. In this paper, we analyse the performance of the radial symmetry algorithm and the regular polygon detector algorithm with respect to the threshold on gradient magnitude. We show that the number of pixels correctly voting on a circle is not greatly reduced by magnitudes that decrease the total number of pixels that vote in the image to 20%. This greatly reduces the image noise component, with only a slight impact on the signal. This results in significant improvements in performance, particularly for the regular polygon detector where the voting mechanism is complex and constitutes a large amount of the processing per pixel.

Previous discussion of orientation information from gradients for detection appears in [5]. Here gradient information was used to coarsely divide edge points by orientation. In this case, the authors suggested paying attention to the gradient information mostly to manage computation. They suggested having the threshold at about 2% of the range in the image. However, coarse quantisation based on orientation is not as subject to the noise that arises from the gradient as voting to a point that is a distance of the radius away. We can certainly expect stronger effects from noise, and thus would require a better signal-to-noise ratio, and hence a greater threshold. Our results show substantially different threshold performance. The discussion of thresholding in [3] is brief, and focuses on small radii for eyes where the effect of noise is not as pronounced, location error scales with radius. We examine particularly for radii that are typical for sign detection and larger features.

In this paper, we present an analysis of performance under variation of the gradient magnitude threshold. We show that controlling this threshold appropriately decreases the signal-

to-noise ratio of resulting detection images for the radial symmetry detector and the regular polygon detector. It also facilitates a significant improvement in the computational speed of the algorithms, particularly the polygon detector.

We first present an overview of the radial symmetry, and regular polygon detection algorithms. Next we present a theoretical analysis of algorithm sensitivity to gradient image noise, and use a series of images to examine the practical effects of noise. Finally, we present results on the impact of the threshold on the signal-to-noise ratio, the image processing computation, and the implications for the results, and computational performance in the autonomous vehicle application.

## II. SIGN DETECTION IN DRIVER ASSISTANCE SYSTEMS

The advantages of separate recognition and detection stages have been observed by many authors (e.g., [6]). As recognition algorithms maybe computationally intensive per pixel, it is advantageous to use an initial detection stage that has low computational cost. Much of the research in this area uses colour based segmentation for detection. Typically, this is based on the assumption that the wavelength that arrives at the camera from a traffic sign is invariant to the intensity of incident light. This assumption usually manifests in the statement that HSV (or HSI) space is invariant to lighting conditions [7]. A great deal of the research in this area exploits a detection stage based on this assumption (e.g., [8], [6], [9]), either finding the signs, or eliminating much of the image from further processing. However, the camera image is not invariant to changes in the chromaticity of the incident light, and this can vary under different conditions such as direct sunlight, heavy cloud, or headlights at night. Thus, it is advantageous to use a method that is invariant all aspects of lighting variation. Shape detection shows such invariance.

We may eliminate the majority of pixels, by only finding a small number of candidates for recognition via shape detection. We have scale of these candidates from the radius returned by the shape detector. The system in [1] applies template matching to classify the resulting signs. The entire system was implemented in c++ and set up to run directly from a camera mounted where the rear-view mirror would be on the NICTA/ANU/CSIRO Autonomous Vehicle (see Figure 1). For a 320x240 image the full detection and classification of 40 and 60 signs was able to run at 20Hz.



Fig. 1. The ANU/NICTA/CSIRO Intelligent Vehicle. The cameras are mounted where the rear view mirror would be.

### A. Overview of the radial symmetry algorithm

The fast radial symmetry detector [3] is a variant on the circular Hough transform that executes in order  $kp$ , where  $p$  is the number of pixels, and  $k$  is the number of discrete radii that are searched. This is as opposed to the traditional circular Hough transform that executes in order  $kbp$ . The fast radial symmetry detector eliminates the factor  $b$  by taking the gradient of the edge point directly from the output of the Sobel edge detector. In this way, computation of the radial symmetry detector is reduced and the votespace is simplified by a dimension. This makes it suitable for real time use, with the application to speed sign recognition running at frame rate.

To better facilitate discussion, we include a description of the radial symmetry detector that is largely taken from [3], where full details appear. For a given pixel,  $p$ , the gradient,  $g$ , is calculated using an edge operator that yields orientation, such as Sobel. If this pixel lay on the arc of a circle, then its centre would be in the direction of the gradient, at the size of the radius. The location of a pixel that will gain a vote as a potential centre is defined:

$$\mathbf{p}_{+ve} = \mathbf{p} + \text{round} \left( \frac{g(\mathbf{p})}{\|g(\mathbf{p})\|} n \right), \quad (1)$$

where  $n \in N$  is the radius, and  $N$  is the set of possible radii. In application to sign detection, this is defined by expectations about the apparent sign size. A vote image is defined based on these orientation votes as:

$$O_n(\mathbf{p}_{+ve}) = O_n(\mathbf{p}_{+ve}) + 1 \quad (2)$$

The vote image is defined as:

$$\hat{F}_n(\mathbf{p}) = \text{sgn}(\tilde{O}_n(\mathbf{p})) \left( \frac{|\tilde{O}_n(\mathbf{p})|}{k_n} \right)^\alpha, \quad (3)$$

where  $\alpha$  is the radial strictness parameter, and  $k_n$  a scaling factor that normalises  $O_n$  across different radii. Also,

$$\tilde{O}_n(\mathbf{p}) = \left\{ \begin{array}{ll} O_n(\mathbf{p}), & \text{if } O_n(\mathbf{p}) < k_n, \\ k_n, & \text{otherwise.} \end{array} \right\} \quad (4)$$

To obtain the radial symmetry image,  $\hat{F}_n$  is convolved with a Gaussian. There are several images produced by the transform. Each radii of  $N$  votes into a separate image.

### B. Overview of the regular polygon detection algorithm

The regular polygon detection algorithm is a generalisation of the radial symmetry algorithm. Full details of the algorithm can be seen in [2], a summary is included here to facilitate analysis. A regular polygon can be thought of as a circle, with its edge represented by a number of linear segments of uniform length (three segments for a triangle or eight for an octagon). Consider a point on the boundary,  $p$ . The centre of the shape is along a line,  $l$ , parallel to the edge at the distance of the radius,  $r$ , of the corresponding circle from  $p$ . Let the closest point on  $l$  to  $p$ , be  $cp$ , and let  $el$  be the length of the edge segments of the shape for the considered radius. The centre

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

|    |    |    |
|----|----|----|
| 1  | 2  | 1  |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

Fig. 2. The x and y 3x3 Sobel masks

|     |     |     |
|-----|-----|-----|
| k   | k   | k   |
| 10k | 10k | 10k |
| 10k | 10k | 10k |

|     |     |      |
|-----|-----|------|
| k   | k   | 1.3k |
| 10k | 10k | 13k  |
| 10k | 10k | 13k  |

Fig. 3. A horizontal edge without noise, and with 30 % noise

of the shape is constrained in its distance from  $cp$  on  $l$  to be less than or equal to  $el$  away. Thus, we can say that the likely location is anywhere in this set of pixels, and vote for them accordingly. At each of these pixels, votes accumulate for each image. Further, regular polygons are equiangular, thus for an  $n$ -sided polygon, their sides are separated by  $360/n$  degrees. To increase detection rates we define an angle  $\gamma = n\theta$ , where  $\theta$  is the gradient angle of an edge. All points vote also with a vector in the direction of this angle. At a possible centre location, these angles are accumulated so that if two edge points voted with opposite direction values for  $\gamma$  they would cancel each other out. This means that the expected shape will accumulate a large tally for  $\gamma$  at its centre. The direct voting image, and this angle voting image are combined to form the regular polygon detection image for a particular radius and number of sides.

### III. ANALYSIS OF SENSITIVITY OF VOTING

In this section we consider the impact of noise in a discrete image on the orientation detected by the Sobel operator. We also examine the effect this has on the voting accuracy of the radial symmetry and regular polygon detection algorithms.

For simplicity, consider the 3x3 Sobel masks (Figure 2), convolved with a local area that has noise. Specifically, consider a straight line in x, where the third column is all increased by 30%. With no noise, and with 30% noise, the window will appear as shown in Figure 3. Without noise the result would be a total gradient of 36k in the y direction and zero in x, and so a vertical normal orientation. With noise, the y gradient at the centre would be 39.7k, and 9.3k in x, resulting in a gradient direction of 13 degrees, or 0.23 pixels for every unit pixel of radius. This means at a radius of nine (a typical radius for voting in sign detection), the error would be greater than two pixels and so is unlikely to have much of an impact on the votes on the centre point.

Note that if we were just using magnitude, this would be quite a strong edge, and so would not greatly decrease performance. Further, if we were using the gradient to coarsely quantise edge direction, it is unlikely that our quantisations would be used to this level, and so may not impact on performance (certainly not in the case discussed in [5], with orientation divided into eight quanta, and so of 45 degrees).

Thus, given a constant amount of noise in an image, we can expect that regions of low intensity gradient will not lead to

accurate votes in radial symmetry based algorithms.

#### A. What is noise here?

For the discussion here, noise is any effect that changes pixel values from being a ideal step edge. This could be variation in the CCD elements or that the colour at the edge itself is not actually perfectly consistent (this can be quite exaggerated in faded road signs). Alternatively, it could be partial pixel effects, that at the border of areas, pixel intensities will be a mixture of the separate colours that fall on the CCD (e.g., Figure 4). All these effects lead to a significant level of image noise. For an incoming image stream it is not easy to characterise the noise, so deriving an analytic formulation of the threshold would not be helpful. Instead, we directly consider the impact of the noise on the radial symmetry image.

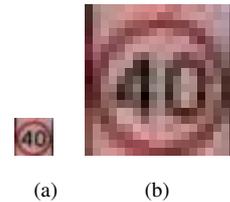


Fig. 4. A typical detected candidate sign (a) at input image size, and (b) close-up. The circle is not clearly of a single colour, and the edges fall within pixels. Also, the edge on the right is poorly defined due to its proximity to the edge of the sign.

### IV. THE IMPACT OF GRADIENT MAGNITUDE ON THE RADIAL SYMMETRY IMAGE

We now consider some real images, and the effectiveness of votes from pixels of varying gradient magnitude. Figure 5 shows images with different sized signs and visibility conditions. In the radial symmetry algorithm, for a given radius, only a set group of pixels could vote on a particular centre. As we convolve the final radial symmetry results with a Gaussian, any votes that fall within a few pixels of the centre may contribute. For any particular radius, any pixel that is a radius distance from any of the pixels within the region of support can effectively contribute to the vote. For the sample set of images, consider all points that may contribute to a centre vote by falling close enough to the centre to make a significant contribution. We set this limit to a two pixel distance of the centre (that is Euclidean distance, rounded). Figure 6 shows a graph of the number of pixels that would have voted within two pixels of the correct centre versus log of squared magnitude of pixel gradient, summed across all four images. This graph shows that a very small percentage of the pixels that vote correctly actually have a low gradient threshold. Secondly, there is a point in the graph at about 9, or actually 11299 in magnitude squared, above which the points become dense. Below this point, the votes are more sparse. If we set the threshold on gradient squared magnitude to this value, we would lose 28% of the votes onto the centre point.

However, on close examination, many of the pixel votes are actually erroneous, and have just fallen close to the centre by accident. Another large group have only fallen that way due to surrounding structure. For example, in at the left-hand edge of the sign in 5(b), it can be seen that the circle falls close to the edge of the sign. At this point, the edge of the sign may vote towards the centre, even if the actual gradient of the sign circle did not properly support this. This 28% appears to include much of the noise.

In terms of applying this to new images in an online system, we need to allow for changes in overall image intensity and contrast. At first inspection, it would be consistent to set a threshold to use votes from the top 20% of the gradient magnitudes. However, for many images there are large regions of self-similar pixels, such as sky. This will have no gradient, but would make a strong difference to where the threshold is acting, rather than getting a fixed percentage of the image, a varying amount would be taken up with regions of constant intensity, effectively reducing the threshold. To adapt for this, we throw away zero gradient magnitudes, and take the top 25% of those remaining. 25% allows these pixels to be found by two partitions of the data, and gives approximately similar results for the sequences we have examined. This creates an additional sort that must be performed, however, this does not have to be performed every frame, as the total image contrast will not change quickly in comparison to frame rate. It may be updated in the background every few frames.

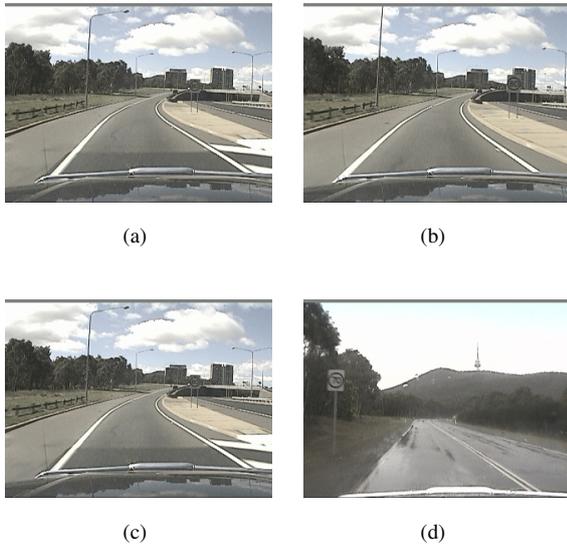


Fig. 5. Four images showing speed signs. (a) - (c) show the same road scene from a range of distances, with the sign appearing at larger radii. (d) shows a different road scene where it is raining heavily and visual conditions vary quite substantially from the other images.

What we have found is a threshold from a set of images that supports our analysis that low pixels with low intensity gradients will not produce reliable radial symmetry votes. There are a series of questions that we must now address in the results. In practice does the application of such a threshold

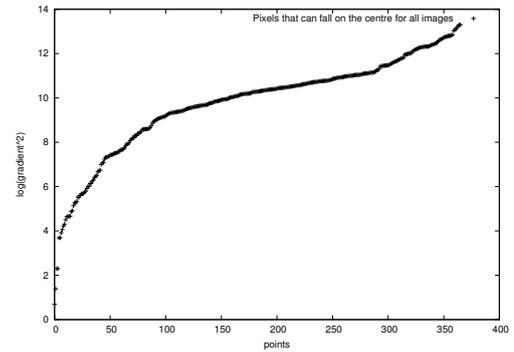


Fig. 6. Cumulative histogram of the gradient magnitude of pixels that fell within a two pixel distance of the centre of the circle of the signs for all the images of Figure 5.

reduce the signal excessively in the radial symmetry image? Does this greatly reduce the noise of the image (i.e., does it reduce the total number of points in the image that are not on the circle that vote)? Will this aid with computation speed? And finally, is this the same for the regular polygon detector, and will its computation speed be aided (it is currently slower than we would prefer)?

## V. RESULTS

We derived the threshold 11299 based on the histograms of the four images. To evaluate the effectiveness of this, we apply it back to these four images and analyse the effect on noise and computation speed for these images. We then examine the effect on the full sequence of 31 images that Figure 5(a) forms a part, and consider some other images.

### A. Impact on radial symmetry signal strength

For the images of Figure 5 with a low threshold (10) for the squared gradient, the radius of the best voting result was taken as the selected radius, in the same manner as the algorithm. We consider the votes on the circle centre. We then applied the threshold derived above of 11299, and compared the votes. The results are shown in Figure 7, it can be seen that only a small reduction in the number of votes for the detected circle has occurred. Sample images with the threshold at the default of 10, and with the new value are shown in Figure 8, we can see that a large reduction in the total number of votes, and the apparent noise. Indeed, in Figure 8 (a), the peak associated with the sign was not the largest value for the low threshold case, whereas it was for the higher threshold.

### B. Threshold effect on image noise and computation speed

Figure 9 shows a cumulative histogram of gradient magnitude for pixels in all four images of Figure 5, excluding the image edges where gradient is undefined. Comparing with Figure 6, we see a much smoother distribution of gradient magnitudes. An interesting point of contrast is at around 9, or 11299 in squared gradient magnitude. 80% of the pixels lie

| image | thr = 10 | thr = 11299 |
|-------|----------|-------------|
| a     | 9        | 8           |
| b     | 10       | 8           |
| c     | 11       | 10          |
| d     | 10       | 10          |

Fig. 7. Total votes for circle centre for threshold of 10, versus derived threshold of 11299 for the images shown in Figure 5.



(a)



(b)

Fig. 8. Radial symmetry results results for the rain image of Figure 5(d): (a) detector vote image with a threshold of 10, and (b) detector image with threshold 11299.

below this value. This is as opposed to 28% for Figure 6. With a small decrease in the signal at the detected feature centre, we see a large decrease in total votes, and so of total noise. This is stronger again when one considers that a significant portion of the 28% are made up of erroneous, or just lucky votes, and thus also may be regarded as noise.

With an 80% reduction in total pixels to be voted on, we can expect up to a factor of five decrease in the execution time of the voting part of the computation, which is a large component of overall computation. As can be seen from Table 7, this has not lead to a significant decrease in the detection of candidates, and may improve it by reducing the number of erroneous candidates reported.

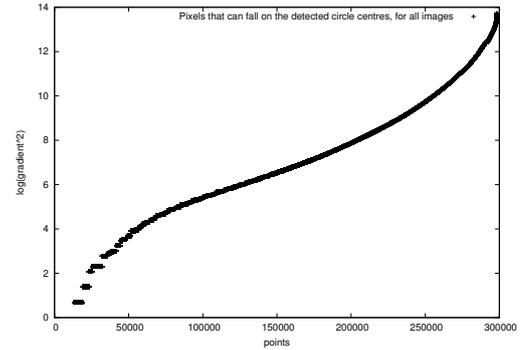


Fig. 9. Cumulative histogram of the gradient magnitude for all of the pixels for all of the images above (excluding a boundary around the edge of the image where the gradient is invalid).

### C. Performance of the regular polygon detector

We have implemented the regular polygon detector in C++. Previously, in running it, for a 320x240 image, searching at 4 separate radii, it ran at 0.8 seconds per frame. As the voting is more complex for the regular polygon detector, it will take up a significant proportion of the total computation time. For the image shown in 10(a), the processing time was reduced to 0.3 seconds per frame, thus the detector will run at more than 3 Hz. This means that a car moving at 60 kph will move only around 5 metres between frames, making this sufficient for real-time processing, as generally, a sign will clearly visible for more processed frames.

Figure 10 shows a giveaway sign and corresponding polygon detector output (direct vote image only) given the standard threshold of 10, and the new derived threshold of 11299. A major reduction in the noise is quite apparent. With a threshold of 10, the maximal vote was 98. At the same location with the increased threshold, the maximal vote was 87, however, at a neighbouring location it was 93, so the centre shifted marginally between the two images. Both centre locations were approximately correct. It can be plainly seen that there is significantly less total votes in Figure 10(c) with the raised threshold. The only slight decrease in the votes at the peak indicates a significant improvement in the signal-to-noise ratio. This is in addition to significant improvement in computational speed. That this image was not included in the set from which the threshold was derived, and was taken by a different camera under different conditions.

### D. Image sequence results

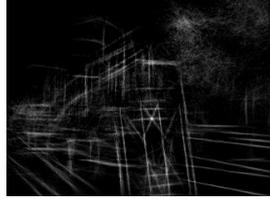
For the 31 frames of the image sequence, we set the threshold for each image to include only the top 25% of gradient magnitudes, after zero magnitude pixels were excluded. The number of pixels that vote on the predicted centre was reduced by 1-3 votes. However, in all cases where the correct peak was the strongest in the low threshold image, it was also the strongest peak in the high threshold image. Indeed, in several cases when the peak was the single best count in the high



(a)



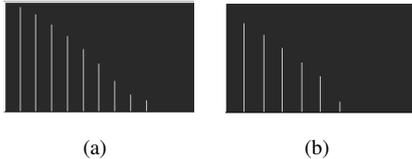
(b)



(c)

Fig. 10. The regular polygon detector: (a) original image, (b) detector vote image with a threshold set to 10, and (c) the detector image with the threshold set to 11299 as found from the set of radial symmetry algorithm results.

threshold image, there were other points at other positions that were not signs, with the same vote count. Thus, the new threshold always preserved algorithm performance and improved it in some cases. Further, the average percentage of pixels processed per image across the sequence for the low threshold was 94% and for the high threshold was 19%. Figure 11 shows the histograms of two typical images randomly selected from the sequence. This shows the better signal to noise performance of the algorithm with this larger threshold.



(a)

(b)

Fig. 11. Histograms of a sequence image. (a) low threshold, (b) taking only the top 20% of pixels. Signal to noise performance is improved with the new threshold. Here maximum scale for (a) is 12, and 10 for (b).

Finally, Figure 12 shows a heavily shadowed scene, to demonstrate performance with different contrast and a different camera, but the same car setup. This was processed with both a low threshold, and the revised 20% threshold. The system demonstrated similar performance to the cases above, with still one clear leading peak in both images at the correct position, and greatly decreased signal to noise apparent, and a large decrease in the number of pixels processed.

## VI. CONCLUSION

We have dramatically improved the performance of the radial symmetry and regular polygon detection algorithm by



Fig. 12. A heavily shadowed scene taken with a different camera.

reducing the noise in the detection images. This has helped improve the quality of detection, and significantly improved the computation speed of the algorithms. Although previous research has considered the accuracy of orientation estimation under noise, the radial symmetry and regular polygon detector algorithms are more sensitive due to their voting at a distance. In this paper we provided a theoretical and a practical analysis of voting in the algorithms under noise. The improvement has been sufficient to make the polygon detection algorithm fast enough for real-time operation.

These algorithms have previously been demonstrated to be highly effective as detectors for road sign recognition, also for face detection, and more recently for Simultaneous Localisation and Navigation. In all of these areas their real-time performance and efficiency is crucial to their operation.

## VII. ACKNOWLEDGMENTS

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

## REFERENCES

- [1] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," in *Proc IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004.
- [2] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS2004*, 2004, in press.
- [3] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, Aug. 2003.
- [4] D. Shaw and N. Barnes, "Regular polygon detection as an interest point operator for slam," in *Proc 2004 Australasian Conference on Robotics & Automation*, 2004, www.araa.asn.au/acra/acra2004.
- [5] P. Kahn, L. Kitchen, and E. M. Riseman, "A fast line finder for vision-guided robot navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, pp. 1098–1102, November 1990.
- [6] P. Paclik, J. Novovicova, P. Somol, and P. Pudil, "Road sign classification using the laplace kernel classifier," *Pattern Recognition Letters*, vol. 21, pp. 1165–1173, 2000.
- [7] L. Priese, J. Klieber, R. Lakmann, V. Rehrmann, and R. Schian, "New results on traffic sign recognition," in *Proceedings of the Intelligent Vehicles Symposium*. Paris: IEEE Press, Aug. 1994, pp. 249–254. [Online]. Available: citeseer.nj.nec.com/priese94new.html
- [8] G. Piccioli, E. D. Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image and Vision Computing*, vol. 14, no. 3, pp. 209–223, 1996.
- [9] C. Y. Fang, C. S. Fuh, S. W. Chen, and P. S. Yen, "A road sign recognition system based on dynamic visual model," in *Proc IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 750–755.