

A Comparison of Canny and V1 Neural Network Based Edge Detectors Applied to Road Extraction

A.C. Hauptfleisch[†], F. van den Bergh[†], A.K. Bachoo[†], A.P. Engelbrecht[‡]

[†]Meraka Institute
CSIR, Pretoria
South Africa
{ahauptfleisch, fvdbergh,
abachoo}@csir.co.za

[‡]Department of Computer Science
University of Pretoria, Pretoria
South Africa
engel@cs.up.ac.za

Abstract

The Anti-parallel edge Centerline Extractor (ACE) algorithm is designed to extract road networks from high resolution satellite images. The primary mechanism used by the algorithm to detect the presence of roads is a filter that detects parallel edges with a specified distance between them. The success of the ACE algorithm thus depends critically on the quality of the edges that are extracted early on in the algorithm, typically using Canny’s edge detector. This paper investigates the viability of an ACE variant that uses a different edge detector, modelled on the primary visual cortex (V1). Considering the experimental evidence, it seems unlikely that the V1-based algorithm is able to produce better results than the original Canny-based algorithm.

1. Introduction

The fully-automatic extraction of road networks from satellite imagery is a problem that has not been solved to the desired level of accuracy yet. Current systems, usually using a combination of techniques (e.g. [1, 2]) manage an overall extraction quality of around 70%.

These algorithms can be classified by the number of assumptions they make regarding the input scenes, or by degree of training that they required. On this scale, the ACE algorithm [3] makes only one explicit assumption, namely that the road width is known in advance. This makes the ACE algorithm useful as a first algorithm that can be used to bootstrap more sophisticated algorithms.

Within the South African context the need for automated road network extraction is clear. In particular, an automated road network extraction system that can successfully extract roads from low-income and/or informal settlements would be invaluable to emergency services. With the pending launch of ZASAT-002 [4], recently renamed *SumbandilaSat*, meaning ‘Lead the way’ in Tshivenda, researchers in South Africa will soon have access to affordable high-resolution satellite imagery. ZASAT-002 will offer a 2m resolution panchromatic band, which will be ideal for road network extraction using the ACE algorithm.

This paper investigates the possibility of improving the performance of the ACE algorithm by replacing the Canny-based edge detector it relies on with a different, neural network-based algorithm.

Section 2 provides a brief overview of the two edge detectors used here, followed by a description of the road network

extraction process in Section 3. Some experimental results are presented in Section 4, followed by some concluding remarks in Section 6.

2. Edge Detectors

Edge detectors are algorithms that detect large changes in intensity within a relatively small area in digital images. A large number of edge detectors exist in literature, with Canny’s algorithm [5] being amongst the most popular.

A rather different edge detection algorithm, inspired by the primary visual cortex (V1) in primates, was proposed by McKinstry and Guest [6]. Brief descriptions of these algorithms follow.

2.1. Canny

Canny’s edge detector, first proposed in [5], is supported by a computational theory that explains why the technique works. Through this theory, Canny defined what an ‘optimal’ edge detector is, and subsequently showed that the first derivative of a Gaussian function serves as a good approximation to the optimal edge detector. In practice, the first derivative of the Gaussian function is implemented by first blurring the input image with a Gaussian filter, and then computing the finite difference derivative on the result. The algorithm can be summarised in the following steps:

1. Blur the image with a Gaussian filter with a specified variance σ^2 .
2. Compute the finite difference derivative in the x and y directions to yield the directional derivative approximations G_x and G_y .
3. Compute the edge magnitude as $G = \sqrt{G_x^2 + G_y^2}$, and the edge direction as

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right), \quad G_x \neq 0$$

4. The gradient direction of each pixel is then discretized into one of eight possible sectors.
5. Non-maximum suppression is performed on the magnitude G by suppressing all pixels of which their neighbours, along the normal to the discretized gradient direction, are not all strictly smaller than the current pixel. This effectively ‘thins’ the edge magnitude image to a single-pixel width.

6. Hysteresis is applied to the remaining non-maximum gradient magnitudes. If a pixel's magnitude exceeds the high threshold T_{high} , then the pixel is marked as an edge pixel. All neighbours of this edge pixel are then traced and labeled as edges, recursively, provided that their magnitudes remain above the lower threshold T_{low} .

The output of Canny's algorithm is thus a bitmap of single-pixel-width edges, along with the edge orientation, θ , for each of these edge pixels.

2.2. V1 Neural Network

McKinstry and Guest [6] suggested a Neural Network (NN) based edge detector that models V1 according to the most prominent characteristics. These characteristics were identified as:

Complex cells represent approximately 75% of the cells in V1 [6]. The complex-cells module comprises of a simple cell orientation filter layer and a complex-cell response layer.

Feature maps are modeled as an arrangement of complex cells in a two dimensional feature map.

Long-range connections were determined within the feature maps through the Self-Organizing Map Extended (SOME) algorithm [7].

The process involved with integrating these characteristics are described in the following steps at the hand of Figure 1:

1. Each pixel is represented by a neuron in the Input Layer. The neuron output is the gray-scale value of the corresponding image pixel.
2. Simple cells serve as an orientation filter, within the complex-cell module, and is modelled by Gabor functions. The simple cell response is computed as

$$s_{\phi} = f(g_{\phi} * I)$$

where I is the intensity image, $*$ a two dimensional convolution, g the Gabor function, and ϕ the phases 90° and -90° . The Gabor function is computed as

$$g(x, y) = e^{-\pi\sigma^2(x^2+y^2)} \cos(2\pi\omega[x \cos \theta + y \sin \theta] + \phi)$$

where θ is the orientation, ω the spatial frequency, and σ the Gaussian variance.

3. The Gabor Receptive Field (GRF) was developed to model complex-cell responses by computing the dot product of simple cell responses s_{ϕ} (difference of Gaussian filter) and the Gabor Function $g(x, y)$. The GRF output is therefore

$$c = f(g * s_{90} + g * s_{-90}).$$

4. Please see Section 2.1 for a description on edge thinning through non-maximum suppression.
5. The SOME algorithm is used to detect the long-range connections between the complex cells. Since it is not computationally feasible to connect each neuron to all the neurons within a neighbourhood, the number of connections is limited to 500. These connections are distributed at random within a circular annulus area with an inner, r_{inhib} , and outer, $r_{\text{horizontal}}$, radius.

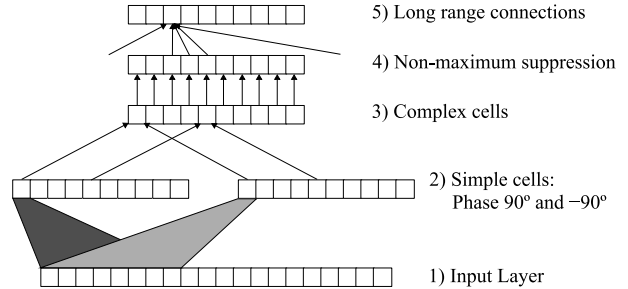


Figure 1: Integrated V1 model.

After every 500 iterations, each neuron's connections are evaluated and redistributed if a low connection strength is detected. New connections are formed with randomly chosen neurons from a set of all neurons adjacent to existing connected neurons. This results in groupings of connection where high synaptic activity occur, similar to those found in V1.

The reader is referred to [7, 6] for a more in-depth discussion of SOME.

The result of the V1 edge detector is a binary image, containing edges where neurons with high synaptic activity occurs. Due to the nature of the SOME algorithm, high synaptic activity occurs over long-range edges. This method is less susceptible to noise and tends to discard short edges, resulting on an emphasis on longer continuous lines, which is expected to lead to improved results on road network extraction.

3. Road Extraction

3.1. ACE

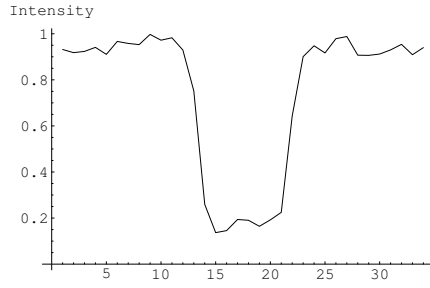
Simple edge detection methods are not suitable for high resolution images with a ground sampling distance¹ of less than 5 meters, since a typical road will appear as a feature several pixels wide. It is, however, possible to use edge information if an algorithm is constructed to consider pairs of parallel edges with opposing gradient directions. Consider the profiles of a cross section of a road, and its corresponding derivative, as shown for an idealised road in Figure 2.

Note that, in the gradient plot (Figure 2 b), the edges of the road are visible as two local extrema with opposing signs.

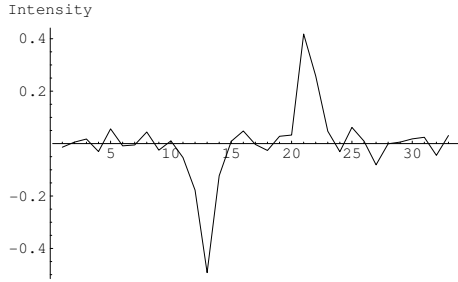
The ACE algorithm described in [8, 3] is one method that extracts candidate road segments by locating such edges with opposing gradients. First, a Canny edge detector is used to find significant edges within the image, which is followed by a 3×3 Sobel filter to determine the edge orientation. Thereafter, the edge image is scanned horizontally and vertically, for successive pixels p and q that satisfy the distance and gradient orientation conditions. The conditions can be summarised as follows (See Figure 3):

- The gradient at p must be approximately equal to the gradient at q plus 180° .
- If \vec{w} is the vector from p to q , and \vec{w}_{pq} represents a vector in the gradient direction at p with its magnitude equal to the expected road width, then the angle between \vec{w} and

¹a remote sensing term that corresponds roughly to the familiar notion of 'pixel size'



(a) Profile



(b) Derivative

Figure 2: Profile of an idealised road, with noise added

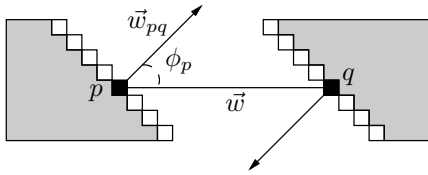


Figure 3: Anti-Parallel Edge detection, as used in ACE

\vec{w}_{pq} must be consistent with the distance between p and q , so that

$$\|\vec{w}\| \cos \phi_p \approx \|\vec{w}_{pq}\|$$

If these conditions are met, the midpoint between p and q is marked as a road centerline.

In [3], colour-infrared images were used as input to the ACE algorithm, with the best extraction result occurring when principal component 2 was used, rather than applying ACE to each band individually. For this paper, only panchromatic (i.e. grayscale) imagery was used.

3.2. Self-Organizing Road Map

The Self-Organizing Road Map (SORM) was developed by Doucette *et al.* [9, 3] as a topology verification and road network construction technique. SORM is a spatial clustering algorithm that links elongated neighbouring regions of similar orientations. The SORM algorithm is typically applied to line segments obtained from a low-level algorithm, such as ACE.

The main function of the SORM algorithm is to remove unwanted, isolated pixel clusters that do not form part of a road, while at the same time reinforcing the valid segments that are

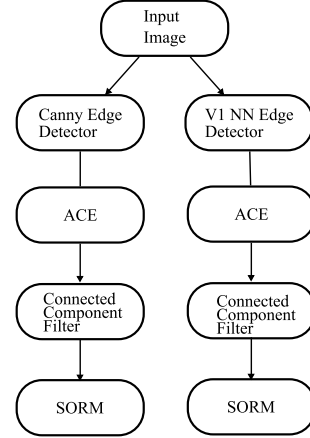


Figure 4: Data flow representation of road network extraction process

likely to form part of the road network. The latter is typically achieved by bridging small gaps between line segments that have similar orientations.

The degree to which the SORM algorithm can repair a noisy centerline image (as produced by ACE) is limited, and as a result the quality of the output of SORM is significantly affected by the quality of the input image. To this end, it is advisable to remove all connected component smaller than some threshold, cc_{min} before the image is processed with SORM.

Implementation details of the SORM algorithm can be found in [9, 3].

3.3. Road extraction pipeline

Figure 4 illustrates how the various components that make up the road network extraction algorithms fit together.

In Section 3.2 it was mentioned that the SORM algorithm is sensitive to the size of the smallest connected component in its input. To investigate the impact of this parameter, and thereby optimise the performance of the overall system, the connected component filter minimum component size will be treated as a system parameter in the experiments of Section 4.

4. Experiment

4.1. Quality Metric

The output of the two variants of the ACE algorithm presented here will be compared using the quality metric suggested by Heipke *et al.* [10, 11], which compares extracted centerline vectors to a reference set.

To simplify the calculation of the quality metrics, a raster-based approximation was implemented to produce the results presented here.

In order to measure the quality of the extracted road centerlines, a reference set must be available, giving the true position of the road centerlines, known as the ground truth. For the images used in this experiment, the ground truth was obtained by manually tracing out the road centerlines, and saving the resulting bitmap images. An example of such a ground truth image is presented in Figure 6.

Heipke's quality metrics require the calculation of 'buffer zones' around the ground truth centerlines. In the raster implementation, this requires the morphological dilation [12] of the

ground truth images with a disc of radius 3 as structuring element.

The following symbols are defined:

Symbol	Meaning
L_e	length of extracted line segment
L_r	length of reference line segment

Pixels in the extracted centerline images that are covered by the dilated ground truth image are counted as true positives (TP). Pixels in the extracted centerline images that are not covered by the dilated ground truth images are counted as false positives (FP). Lastly, the difference between the number of on-pixels in the ground truth image, and the number of true positive pixels is called the false negatives (FN).

Using these quantities, the following metrics are defined:

Completeness is a measurement of the percentage of the reference road network that has been extracted.

The completeness metric is calculated as:

$$\begin{aligned} \text{completeness} &= \frac{\text{matched } L_e}{L_r} \\ &\approx \frac{TP}{TP + FN} \end{aligned}$$

where completeness $\in [0, 1]$. A completeness value of 1 indicates that the reference network is entirely covered by the buffer area around extracted network.

Correctness is the percentage of the extracted road network that is also present in the reference set.

The correctness metric is calculated as:

$$\begin{aligned} \text{correctness} &= \frac{\text{matched } L_e}{L_e} \\ &\approx \frac{TP}{TP + FP} \end{aligned}$$

where correctness $\in [0, 1]$. A correctness value of 1 indicates that the extracted network is entirely covered by the buffer area around the reference network.

Quality is a measurement that combines completeness and correctness.

The quality metric is calculated as:

$$\begin{aligned} \text{quality} &= \frac{\text{matched } L_e}{L_e + \text{unmatched } L_e} \\ &\approx \frac{TP}{TP + FP + FN} \end{aligned}$$

where quality $\in [0, 1]$. A quality value of 1 indicates that the extracted network lies within the buffer around the reference network and vice versa.

4.2. Parameter selection

Table 1 presents the results of applying the two ACE variants to a set of 12 images, all with a dimension of 512×512 pixels. The values reported are median value over all 12 images, for each of the metrics defined in Section 4.1.

Based on these results, a cc_{\min} value of 7 was selected for the Canny ACE variant. This corresponds to the setting with a good compromise between obtaining the high quality rating, but also a good correctness value. For the V1 neural network ACE variant, a cc_{\min} value of 6 was selected, as this again corresponds to the best combination of quality and correctness.

Table 1: Extraction results of ACE variants for various minimum connected component filters. The column headings are as follows: cc_{\min} is the minimum connected component size, **cor** denotes *correctness*, **com** denotes *completeness*, and **qual** denotes *quality*.

cc_{\min}	Canny			V1 Neural Net		
	cor	com	qual	cor	com	qual
5	0.732	0.606	0.498	0.864	0.357	0.302
6	0.839	0.562	0.488	0.932	0.342	0.303
7	0.868	0.540	0.480	0.927	0.318	0.294
8	0.851	0.518	0.470	0.895	0.306	0.298
9	0.863	0.498	0.442	0.899	0.265	0.258
10	0.893	0.492	0.430	0.912	0.268	0.256
11	0.909	0.488	0.442	0.915	0.252	0.250

Table 2: Explanation of image classes. The value in parentheses indicates the number of scenes in each class.

	Good contrast	Poor contrast
Trees present	class C (13)	class B (5)
No/few trees present	class D (12)	class A (5)

4.3. Comparison

The ACE algorithm makes certain assumptions regarding the nature of the input image. The first assumption is that the road widths are known in advance — this assumption is readily satisfied for remote sensing images, since pixel sizes are fully specified via projection metadata. The four classes presented in Table 2 were selected to reflect some of the implicit assumptions that the ACE algorithm makes.

The second of these assumptions is that sufficient contrast exists between roads and their immediate surroundings. This assumption is satisfied when looking at well-developed suburban areas, with newly tarred roads. Image classes C and D represent such areas where the contrast is (for most of the roads in the scene) sufficient for the edge detection algorithms to detect strong edges.

The third assumption that ACE makes regards the continuity of visible road edges. In established suburban areas it is not uncommon to have large trees growing next to the road, often occluding a significant fraction of the road surface. This results in one (or even both) of the road edges being fragmented, thereby resulting in the failure of ACE to extract a centerline. Images in classes C and B include examples of partial to near complete occlusion of the road surface.

Table 3: Stratified comparison of extraction quality: Canny ACE + SORM vs V1 NN + ACE + SORM. The column headings are as follows: **cor** denotes *correctness*, **com** denotes *completeness*, and **qual** denotes *quality*.

Class	Canny			V1 Neural Net		
	cor	com	qual	cor	com	qual
class A	0.066	0.013	0.011	0.329	0.025	0.024
class B	0.811	0.341	0.314	0.760	0.272	0.249
class C	0.900	0.368	0.345	0.915	0.261	0.252
class D	0.789	0.500	0.442	0.880	0.339	0.322



Figure 5: Sample scene_03-D.

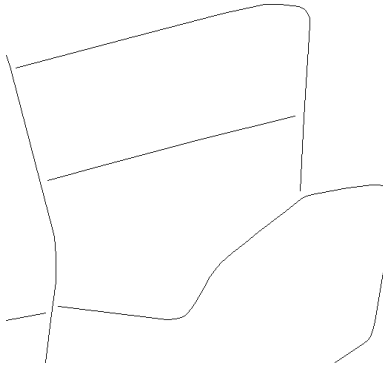


Figure 6: Ground truth for scene_03-D.



(a) V1 NN edges

(b) V1 NN ACE



(c) V1 NN ACE $cc_{min}=7$

(d) V1 NN SORM (final)

Figure 8: V1 NN + ACE + SORM applied to scene_03-D



(a) Canny edges

(b) Canny ACE



(c) Canny ACE $cc_{min}=7$

(d) Canny SORM (final)

Figure 7: Canny + ACE + SORM applied to scene_03-D

4.4. Sample outputs

Sample output for the two ACE variants have been provided in Figures 7 and 8. For reference, the quality metric values for this scene (scene_03-D) are as follows:

Metric	Canny	V1 NN
correctness	0.985407	0.925106
completeness	0.628306	0.532321
quality	0.622513	0.510329

5. Discussion

Based on the results presented in Sections 4.3 and 4.4 it would appear that the Canny-based algorithm yields extracted road networks of superior quality. In particular, looking at Table 3, it appears as if the Canny-based algorithm delivers a fairly substantial increase of about 10% in performance over the V1 NN-based algorithm on all image classes, except class A.

That being said, a randomisation test at a 5% confidence level detected no statistically significant differences between the two algorithm when applied to each of the classes individually, which would thus lead to the acceptance of the null hypothesis stating that both the algorithms have similar performance.

When this randomisation test is repeated at a 10% confidence level, then a statistically significant difference is detected in class D (for both completeness and quality) and in class C (only for completeness). Given the large confidence interval, it is probably safer to accept the overall null hypothesis recommended by the 5% confidence interval test, and concede that the algorithms perform similarly.

Thus, even though Canny achieved a fairly substantial in-

crease in performance over the V1 flavour, it is not a statistically significant increase.

Additional test datasets, especially different sensor data, would help to confirm if indeed a statistically significant difference between the two ACE variants exists.

6. Conclusion

Despite the evidence in the literature that the V1 NN-based edge detector has superior edge coherence [6] compared to Canny's algorithm, it does not appear to translate into superior road network extraction quality.

A few factors that might influence the performance of the V1 NN-based algorithm have been identified:

1. The V1 algorithm might be more sensitive to the contrast between roads and background.
2. The detected edge positions might differ from the edge position reported by Canny's algorithm, thus affecting the apparent width of the road. This effect was compensated for to some degree by adjusting the width parameter in the ACE algorithm, however, no systematic study of this effect has been performed yet.
3. The ACE algorithm requires fairly accurate edge orientation information. Since the V1 NN-based algorithm implementation used in the experiments did not provide edge orientation information, it was estimated by computing the gradient direction under the detected edge pixels. Given the point 2 above, this could result in significant edge orientation errors. The influence of this has not been studied yet.

Based on the experimental evidence it seems unlikely that the V1 NN-based road network extraction algorithm is able to produce better results than the original Canny-based algorithm.

7. References

- [1] X. Jin and C. H. Davis, "An integrated system for automatic road mapping from high-resolution multi-spectral satellite imagery by information fusion," *Information Fusion*, vol. 6, pp. 257–273, 2005.
- [2] J. B. Mena and J. A. Malpica, "An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery," *Pattern Recognition Letters*, vol. 26, pp. 1201–1220, 2005.
- [3] P. Doucette, P. Agouris, and A. Stefanidis, "Automated road extraction from high resolution multispectral imagery," *Photogrammetric Engineering & Remote Sensing*, vol. 70, pp. 1405–1416, Dec. 2004.
- [4] A. Schoonwinkel, G. W. Milne, and S. Mostert, "Opportunities in satellite based earth observation, telecommunication and navigation for sustainable development in southern africa," *Acta Astronautica*, vol. 53, pp. 749–759, 2003.
- [5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, 1986.
- [6] J. L. McKinstry and C. C. Guest, "Long range connections in primary visual cortex: A large scale model applied to edge detection in gray-scale images," in *Proceedings of IJCNN*, (Washington, DC.), 2001.
- [7] J. L. McKinstry and C. C. Guest, "Self-organizing map develops v1 organization given biologically realistic input," in *Proceedings of the IEEE International Conference on Neural Networks*, 1997.
- [8] P. Doucette, *Automated Road Extraction from Aerial Imagery by Self-Organization*. PhD thesis, Department of Spatial Information Science and Engineering, University of Maine, 2002.
- [9] P. Doucette, P. Agouris, A. Stefanidis, and M. Musavi, "Self-organized clustering for road extraction in classified imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 55, no. 5–6, pp. 347–358, 2001.
- [10] C. Heipke, H. Mayer, and C. Wiedemann, "Evaluation of automatic road extraction," in *Proceedings of the ISPRS Conference*, 1997.
- [11] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet, "Empirical evaluation of automatically extracted road axes," in *Empirical Evaluation Techniques in Computer Vision*, 1998.
- [12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (2nd Edition)*. Prentice Hall, 2002.