



Enhanced Edge Detection Technique for High-Resolution Satellite Imagery Using a Curvelet-Canny Approach

A. Elsharkawy^{*}, M. Elhabiby[†], N. El-Sheimy[‡]

Abstract: In this research paper, a new approach using the capability of the second generation curvelet transform together with the traditional canny operator for edge detection of high-resolution satellite imagery; the combined technique will be applied on WorldView-2 imagery. First, the curvelet coefficients will be generated in multi-scales and multi-directions using a forward discrete curvelet algorithm. Then, these coefficients will be sorted in each scale to generate the edge map using the larger coefficients for the coarser scales. Second, this edge map will be the input to the second stage where the three main steps of the traditional canny operator, gradient calculation, non-maximal suppression and hysteresis, will be applied. The first step results in removing noise, fine edges, from the image aiding the second step for better connecting the strong edges without the effect of weak edges coming from the noise.

The percentage of the utilised coefficients in the curvelet transforms step together with the weight for each scale are the tuning parameters the user has to adjust to get the desired level of edges detected. The results from the proposed approach were compared to the traditional canny edge detection algorithm. The results showed very good potentials for detecting elongated edges and also for generating more closed objects, which make this method a good alternative for the segmentation step for any further object-based classification algorithm.

Keywords: Curvelet transform, canny operator, edge detection, high resolution satellite imagery.

Introduction

Second generation curvelet transform provides optimally sparse representations of objects, which display smoothness except for discontinuity along the curve with bounded curvature (Candes et al., 2006). Some papers have investigated this technique for edge detection in high-resolution satellite imagery such as IKONOS or QuickBird, and microscopic imagery, (Geback and Koumoutsakos, 2009; Guha and Wu, 2010; Hui-qian et al., 2007; Xiao et al., 2008; Zhenghai and Jianxiong, 2009) which show great potentials of using curvelet transform in solving edge detection problems.

^{*} Corresponding Author, Egyptian Armed Forces, Egypt; ahmadshawky007@yahoo.com.

[†] Public Works Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt; (Dept. of Geomatics Engineering, University of Calgary, Calgary, Alberta, T2N 1N4,) mmelhabi@ucalgary.ca

[‡] Dept. of Geomatics Engineering, University of Calgary, Calgary, Alberta, T2N 1N4, elsheimy@ucalgary.ca .

Urban studies, coastal erosion, and agricultural surveys are a few examples where edge detection can be utilized. In the past few years, the development of edge detection techniques for the analysis of multi-temporal remote sensing imagery has been intensively growing. For many years, satellite based remote sensing has been a priceless tool for change detection. No other platform can constantly revisit an area, quantify and classify land cover or land use on such a broad scale. Satellite imagery are proving to be a cost-effective alternative to aerial photography, especially, for the acquisition of Land Cover information([Ouma et al., 2010](#)).

One of the most important characteristics in an image is the features' edges, which can be described as a discontinuity in the local domain of the image. These discontinuities may result as gray, colors and texture variations ([Zhenghai and Jianxiong, 2009](#)). Edge detection has broad applications in the domain of image processing, computer vision and so on. The influence of this process comes from the fact that it usually lies at the bottom of the classification process to serve as a base map for all other coming modules. Consequently, the more accurate this process is, the more accurate the whole classification results.

Direct thresholding of the edge map generated from the curvelet transforms was discussed in ([Elsharkawy et al., 2011](#); [Elsharkawy et al., 2012](#)), for building extraction for high-resolution satellite imagery. In this research paper, an implementation of the second generation curvelet transform followed by the three main steps for the canny operator, gradient calculation, non-maximal suppression and hysteresis, for edge detection of high-resolution satellite imagery. A comparison with the optimal edge detector operator, Canny, will be done. In the following two sections, a brief introduction about Curvelet and Canny operator will be introduced followed by the description of the data used and methodology section, then the results and analysis section and finally, the conclusions.

The Basics of Curvelet Transforms

Initial introduction of Curvelet transforms technique was originally introduced by Candes and Donoho in 1999 as a result of the increasingly demand in the presence of effective multi resolution analysis that can overcome the drawbacks of wavelet analysis. The transform was designed to represent edges and other singularities along curves much more efficiently than traditional transforms, i.e. using many fewer coefficients for a given accuracy of reconstruction ([Donoho and Duncan, 2000](#)). This transform used a complex series of steps involving the ridgelet analysis of the radon transform of an image. However, the performance was considered slow.

Later and based on a frequency partition technique, the same authors proposed a considerably simpler second-generation curvelet transform. This second generation curvelet transform is meant to be simpler to understand and use. It is also faster and less redundant compared to its first-generation version([Ma and Plonka, 2009](#)). In the new version of curvelet the ridgelet transforms was discarded, thus reducing the amount of redundancy in the transform and increasing the speed considerably. Curvelet transform is defined in both continuous and digital domain. Moreover, it can be used for multi-dimensional signals. Since the image-based feature extraction requires only 2D FDCT, The discussion will be focused on only two-dimensional application and implementation([Candes et al., 2006](#)).

Continuous-Time Curvelet Transforms

The curvelet representation in two dimensions continuous space, i.e., \mathbb{R}^2 , will be through spatial variable x , with ω a frequency domain variable, and with r, θ polar coordinates in the frequency domain. Then, a pair of windows function $W(r)$ and $V(t)$ is introduced, the "radial window" and "angular window" respectively. These windows will obey the admissibility conditions:

$$\begin{cases} \sum_{j=-\infty}^{\infty} W^2(2^j r) = 1, & r \in (\frac{3}{4}, \frac{3}{2}) \\ \sum_{l=-\infty}^{\infty} V^2(t - l) = 1, & t \in (-\frac{1}{2}, \frac{1}{2}) \end{cases} \quad 1$$

where j is a radial variable and l is an angular variable. The frequency window in the Fourier domain is defined by:

$$2^{-\frac{3j}{4}} W(2^{-j} r) V\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right) \quad 2$$

where $\lfloor j/2 \rfloor$ is the integer part of $j/2$

In the spatial Cartesian domain, the scaling of the radial window introduces an angular window with short axis with 2^{-j} , and a long axis with $2^{j/2}$. Therefore, the effective length and width obey the anisotropy scaling relation $\text{width} \approx \text{length}^2$, and U_j is a polar wedge window, as show in Figure 1.

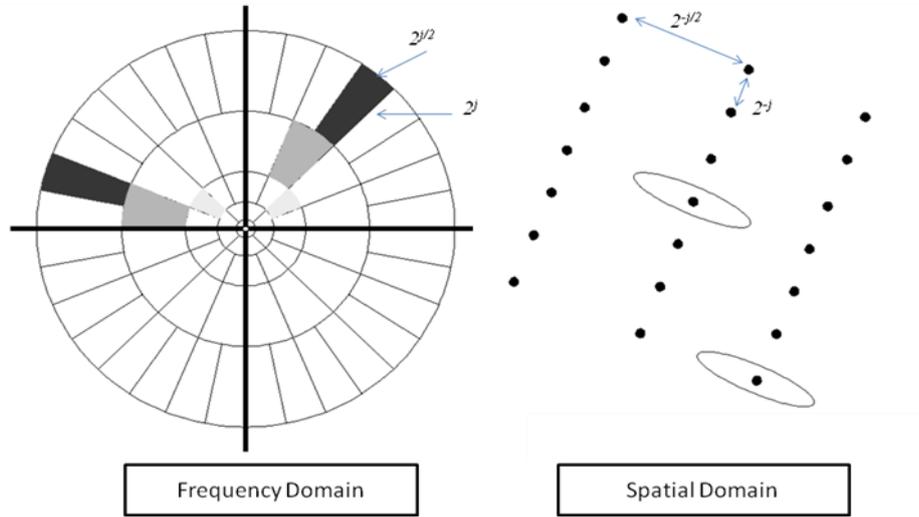


Figure 1 Curvelets in Fourier frequency (left) and spatial domain (right)
[\(Candes et al., 2006\)](#)

In the frequency domain, the curvelet coefficient, with the j scale, the l angle and the sequence of translation parameters $k = (k_1, k_2) \in \mathbb{Z}^2$, is defined as:

$$C(i, j, k) := \frac{1}{(2\pi)^2} \int \hat{f}(\omega) U_j(R_\theta \omega) e^{i(x_k^{(j,l)}, \omega)} d\omega$$

3

Figure 1 illustrates the result of partitioning the Fourier plane into radial (concentric circles) and angular divisions. The concentric circles are responsible for the decomposition of an image into multiple scales, j , while the angular divisions partition the band passed image into different angles or orientations l (Zhenghai and Jianxiong, 2009). For instance the light gray wedges represent the maximal support of the curvelet function $\hat{\varphi}(3,2,k)$ and $\hat{\varphi}(3,7,k)$, while the gray wedges represent $\hat{\varphi}(4,3,k)$ and $\hat{\varphi}(4,8,k)$ and the dark wedges represent $\hat{\varphi}(5,5,k)$ and $\hat{\varphi}(5,15,k)$. As a result, defining the scale j and angle l is the proper way to deal with a particular wedge. By noticing the spatial domain, Figure 1 right, it was found that each of the wedges corresponds to a specific curvelet, shown as ellipses, at a given scale and angle. This indicates that the curvelet coefficients for that scale and angle can be determined by the inverse FFT of this particular wedge (Guha and Wu, 2010). This is the main idea behind the implementation of curvelet transform. As shown in Figure 1, curvelets have a well localized, needle-shaped in higher scales, as the wedges are longer and thinner with scale growing (Ma and Plonka, 2009).

Discrete Curvelet Transform

Coronae and rotations, as in the continuous-time definition, are not especially adapted to Cartesian arrays, so it is convenient to replace these concepts by Cartesian equivalents; here, “Cartesian coronae” based on concentric squares (instead of circles) and shears, figure 2.

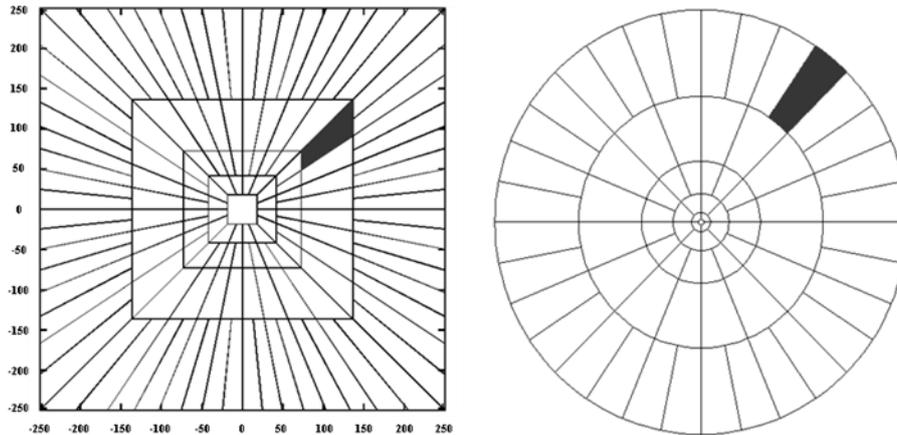


Figure 2 The transition from the continuous-time definition (left) to the discrete-time definition(right) (Candes et al., 2006).

The above figure (left) illustrates the basic digital tiling where, the windows \tilde{U}_j smoothly localize the Fourier transform near the sheared wedges obeying the parabolic scaling. The shaded region represents one such typical wedge.

Now the Cartesian window \tilde{U} is defined as:

$$\tilde{U}_{j,l}(\omega) := \tilde{W}_j(\omega)\tilde{V}_j(S_{\theta_l}\omega) \quad 4$$

where:

$$\begin{cases} \tilde{W}_j(\omega) = \sqrt{\phi_{j+1}^2(\omega) - \phi_j^2(\omega)}, \\ \tilde{V}_j(\omega) = V\left(\frac{2^{\lfloor \frac{j}{2} \rfloor} \omega_1}{\omega_2}\right) \end{cases} \quad 5$$

Φ is defined as the product of low-pass one dimensional windows:

$$\phi_j(\omega_1, \omega_2) = \phi(2^{-j}\omega_1)\phi(2^{-j}\omega_2) \quad 6$$

And S_θ is the shear matrix:

$$S_\theta := \begin{pmatrix} 1 & 0 \\ -\tan\theta & 1 \end{pmatrix} \quad 7$$

Hence, the discrete curvelet coefficients are defined as:

$$C(i, j, k) := \int \hat{f}(\omega)\tilde{U}_j(S_{\theta_l}^{-1}\omega)e^{i(S_{\theta_l}^T \cdot \omega)} d\omega \quad 8$$

According to ([Candes et al., 2006](#)), there are two different digital implementations of FDCT:

- Curvelets via **USFFT** (Unequally Spaced Fast Fourier Transform)
- And Curvelets via **Wrapping**.

Both the variants are linear and they are taken as input a Cartesian array to provide an output of discrete coefficients. The only difference is in the choice of the spatial grid where curvelets at each scale and angle are translated. As the FDCT wrapping is the fastest curvelet transform currently available ([Candes and Donoho, 2005](#)), the wrapping version of curvelet transform, will be used in the implementation. The FDCT wrapping algorithm may be summarized as follow:

1. Take FFT of the image
2. Divide FFT into collection of Digital Corona Tiles as in (Figure 2)
3. For each corona tile do the following:
 - Translate the tile to the origin as in (Figure 3-a)
 - Wrap the parallelogram shaped support of the tile around a rectangle centered at the origin as in (Figure 3-b).
 - Take the Inverse FFT of the wrapped support
 - Add the curvelet array to the collection of curvelet coefficients.

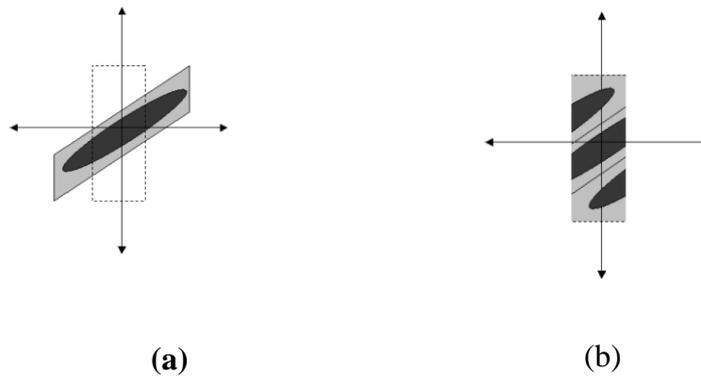


Figure 3 Support of wedge before and after wrapping

The values of curvelet coefficients are depending on how they are aligned in the real image. One can expect higher coefficients values when the curvelet is accurately aligned with a given curve in an image. A very clear explanation is provided in Figure 4. The curvelet named 'c' in the figure is almost perfectly aligned with the curved edge and therefore has a higher coefficient value. Curvelets 'a' and 'b' will have coefficients close to zero as they are quite far from alignment with the curve ([Guha and Wu, 2010](#)).

From the previous discussion it is clear that the curvelet transform provide a distinguished characteristic for the signals where they are better localized in both frequency and spatial domain compared to wavelet or any other transform.

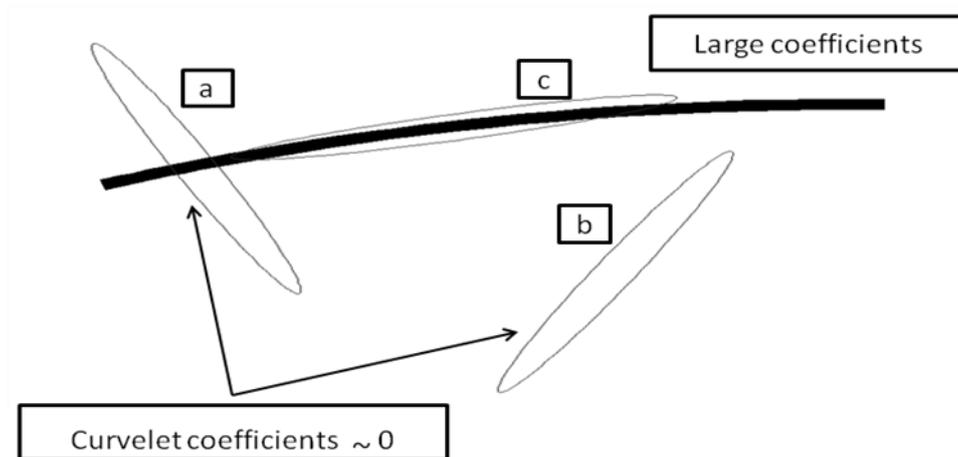


Figure 4 Alignment of curvelets along curved edges ([Guha and Wu, 2010](#))

The unique mathematical property to represent curved singularities in a non-adaptive manner makes the Curvelet transform as a higher dimensional generalization of wavelets. The main advantage of the curvelet transform over wavelet is that the edge discontinuity is better approximated by curvelets than wavelets. Curvelets can provide solutions for the limitations the wavelet transform suffers from, which can be summarized as follow:

1. Curved singularity representation,
2. Limited orientation (Vertical, Horizontal and Diagonal)
3. And absence of anisotropic element (isotropic scaling)

If an image function f is approximated by largest m coefficients as \hat{f}_m , then the approximation errors are given by:

$$\|f - \hat{f}_m^F\|^2 \propto m^{-1/2}, m \rightarrow +\infty \quad \text{Fourier transforms} \quad 9$$

$$\|f - \hat{f}_m^W\|^2 \propto m^{-1}, m \rightarrow +\infty \quad \text{Wavelet transforms} \quad 10$$

$$\|f - \hat{f}_m^C\|^2 \propto m^{-2} \log(m^3), m \rightarrow +\infty \quad \text{Curvelet transforms} \quad 11$$

Canny Edge Detector

Canny edge detection is an optimal method for **step edges**' detection in the spatial domain. Canny used three criteria to design his edge detector. First, a reliable detection of edges with low probability of missing true edges, and a low probability of detecting false edges must be achieved. Second, the detected edges should have a minimum distance to the true location along the edge. Third, there should be only one response to a single edge (thin lines for edges).

Based on these criteria, the Canny edge detector first **smooths** the image to eliminate any noise, then it finds the **image gradient** to highlight regions with high derivatives. The regions with high derivatives are tracked by the algorithm to suppress any pixel that is not at the maximum (non-maximum suppression). The remaining pixels are further reduced by two thresholds T1 and T2. If the magnitude is below T1, it is set to zero (none edge), if the magnitude is above T2, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2 ([Canny, 1986](#)).

Step 1: Gaussian Filtering to Remove Noise

The first step of canny edge detection is to filter out any noise in the original image before trying to locate and detect any edges. The Gaussian filter is used to blur and remove unwanted detail and noise. By calculating a suitable 5 X 5 mask, Gaussian smoothing can be performed using the standard convolution method. The larger the width of the Gaussian mask, the lower the detector's sensitivity to noise. By increasing the standard deviation the intensity of the noise is either reduced or blurred. An example of 2D Isotropic Gaussian equation is given below:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad 12$$

Step 2: Gradient Calculation

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image –there are many ways and masks to perform the gradient calculation. One of these ways is to find the difference between the intensity value of the two consecutive pixels in both directions (x and y). When finding edges we are looking for the steepest descent as well as the steepest ascent since both represent a high change in the intensity of the image. Figure 5 depicts the gradient and orientation process.

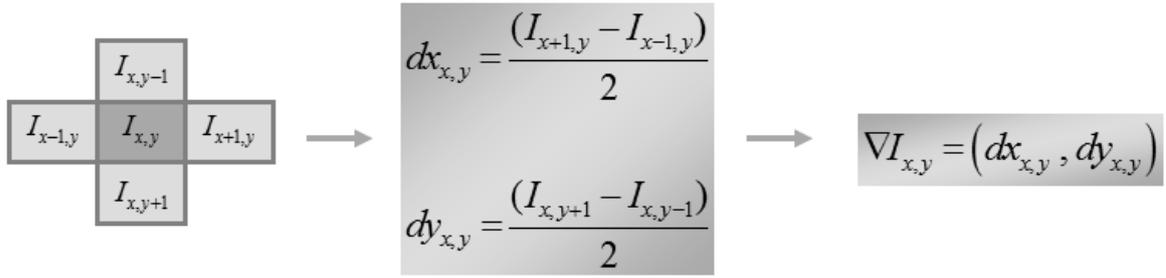


Figure 5 Illustration of gradient calculation in canny operator

Once we have the gradient value for each pixel we can get the **magnitude** of the gradient by:

$$|\nabla I_{x,y}| = \sqrt{(dx_{x,y})^2 + (dy_{x,y})^2} \quad 13$$

The main purpose of doing this is to highlight regions with high spatial derivatives. The **orientation** of the edge can be determined by the next equation:

$$\theta = \arctan \left(\frac{dy_{x,y}}{dx_{x,y}} \right) \quad 14$$

Step 3: Non-Maximal Suppression

This step works with the magnitude and orientation of the gradient of the pixel under consideration and creates one pixel-width edge. However, we need to interpolate the values of the pixels found at the neighborhood around the point under analysis. The pixel that has no local maximum gradient magnitude is now eliminated and the comparison is made between the actual pixel and its neighbors along the direction of the gradient as in Figure 6.

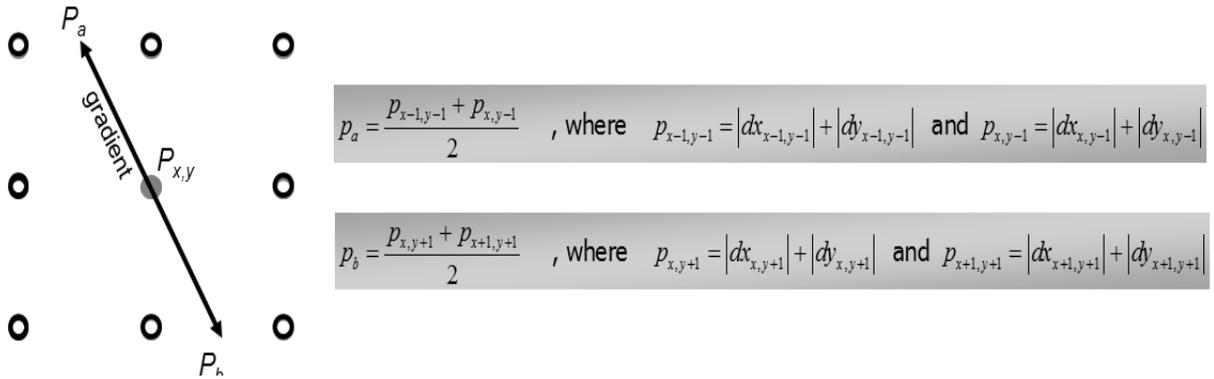


Figure 6 Non-maximal suppression procedure

Step 4: Hysteresis

The problem now is that we may have some pixels that, despite being a local maximum, represent noise. At this stage, most of the edge detectors apply a threshold process. The threshold is then defined so that each pixel with a value below it is eliminated. Two levels of thresholds are selected; high threshold (TH) and low threshold (TL). For a given pixel, if the gradient magnitude is below TL, it is unconditionally set to zero. If the gradient is at least

TH, the pixel is left alone. And finally, if the gradient is between these two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above TH. An example is in Figure 7.

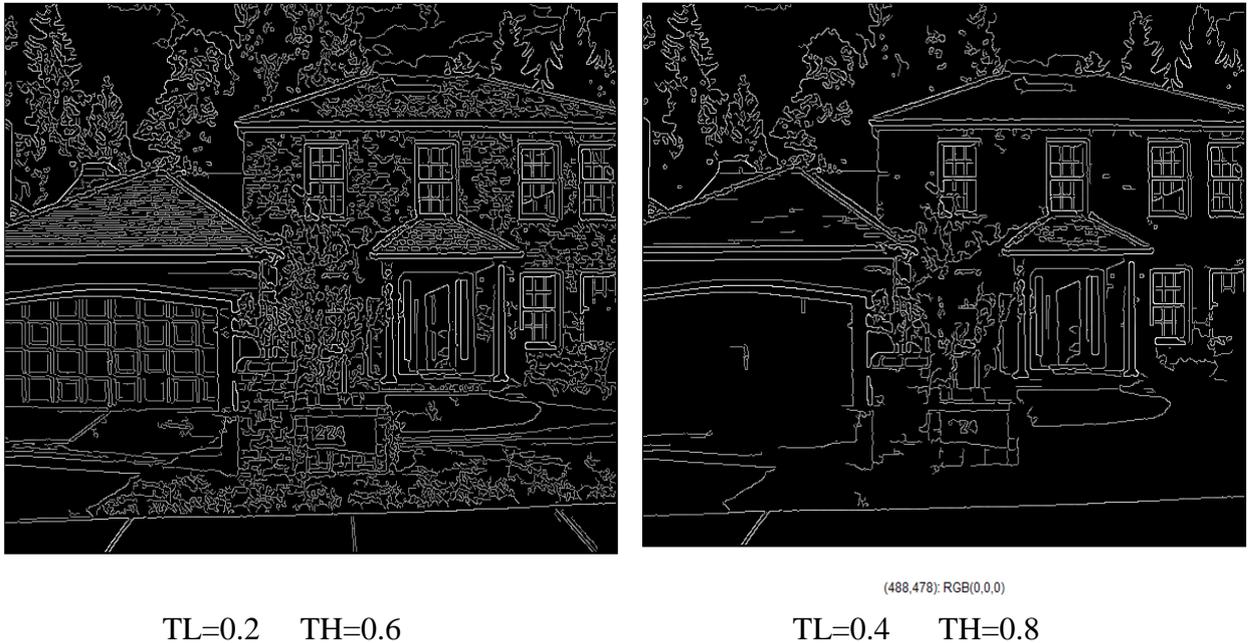


Figure 7 Applying hysteresis to Canny edge map
<http://www.cs.washington.edu/research/imagetdatabase/demo/edge>

Data and Method

Figure 8, illustrates the proposed algorithm for edge detection using curvelet transforms and canny operator with some pre-processing steps. The pre-processing steps involve data fusion between the multispectral bands with 2.00 m resolution with the panchromatic band with 0.5 m resolution. The resulting image will be a multi spectral image with 8-bands. The next step is to convert the RGB image to gray level image, Figure 9, using the standard perceptual weightings for the three-color components RGB using the following equation (<http://www.mathworks.com/help/toolbox/wavelet/gsf4-1013594.html>).

$$\text{Gray level} = 0.2990 * R + 0.5870 * G + 0.1140 * B \quad 15$$

The data is worldview-2 imagery, provided by Digital Globe Company for a part of San Francisco, USA. The imagery was captured on October, 2011 morning time. Forward second generation curvelet transform wrapping algorithm will be applied to the gray level image, Figure 9, resulting in generating of the curvelet coefficients in different scales and orientations. Number of scales is determined according to the size of the original image, following the equation:

$$n = \log_2(N) - 3 \quad 16$$

where N is the minimum number of the image size and n is the number of the scale levels, i.e. for $N=512$ pixels, the $n=6$ levels. The coarser level starts from scale number one and gets finer as we move to the second, third fourth scales.

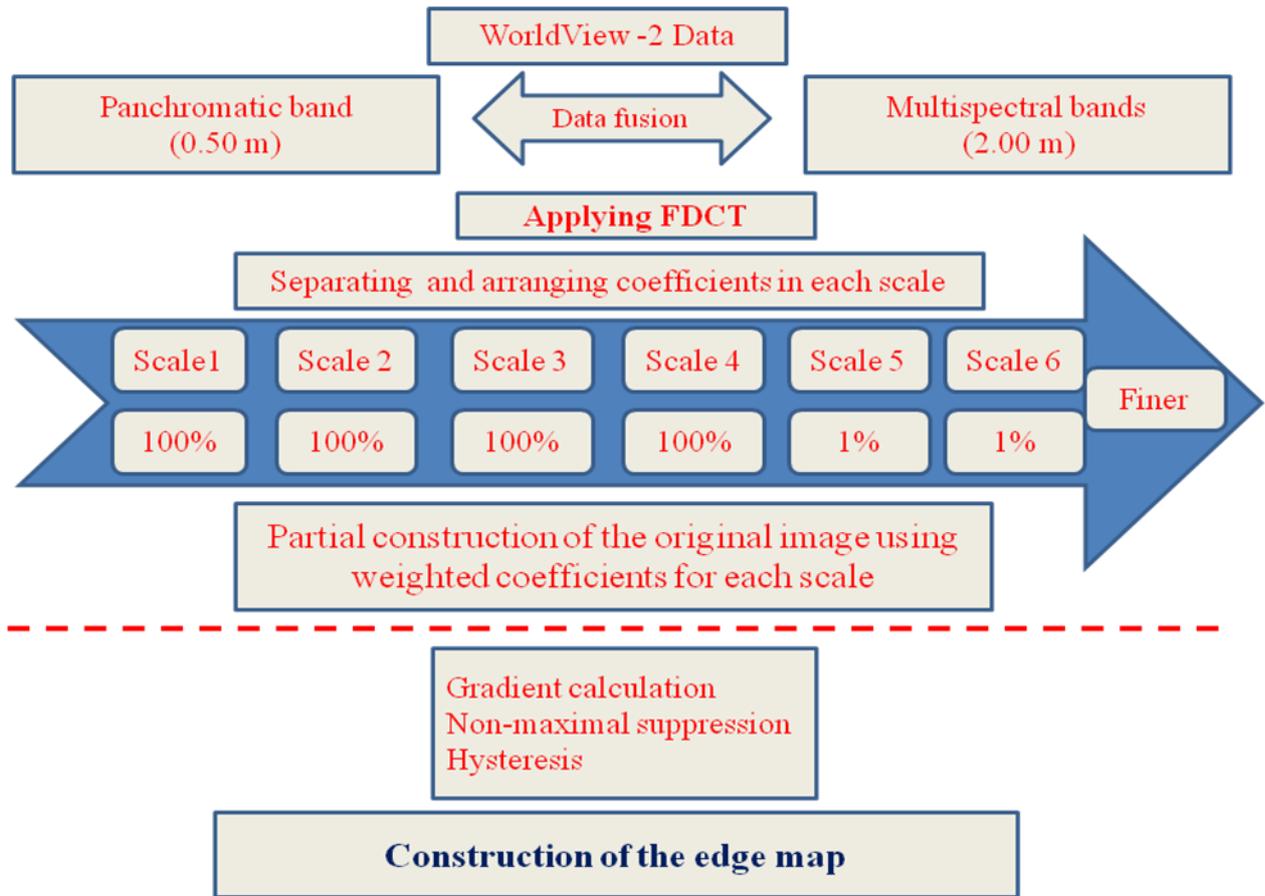


Figure 8 The proposed edge detection technique using curvelet-canny



Figure 9 The grey level image for the area of the study

Reconstructing the images from each scale individually shows that coarser scales, first four scales, contain the most important information about the image, while finer scales, fifth and sixth scales mostly contain noise or very fine details about the image, as in Figure 10. Therefore, the main purpose of this step is to suppress the coefficients coming from the finer scales, as a result the reconstructed image will be partially free from insignificant information, noise and very fine details, to help the next step for better detecting the strong edges. Consequently, by arranging the coefficients of each level and taking the most significant part of them, this will enhance the edge information that represents the image part of interest. Then, the coefficients are reconstructed to get a new image called the edge map ([Elsharkawy et al., 2011](#)), as shown in Figure 11, where the edge parts are enhanced.

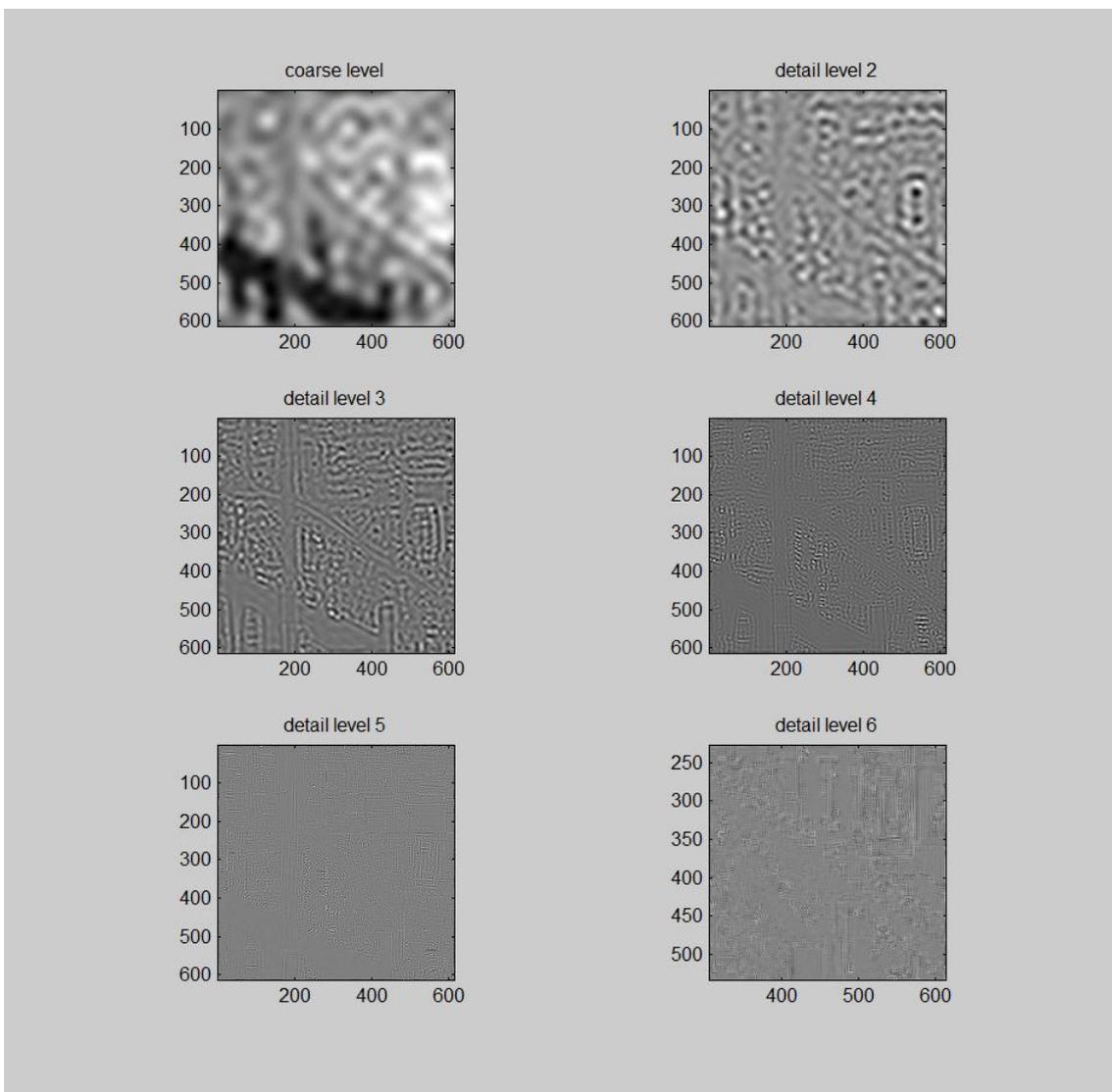


Figure 10 The reconstructed images using different scales levels



Figure 11 The reconstructed edge map

Table 1 summarizes the total number of coefficient in each scale and the weight of each scale level used to reconstruct the edge map.

Table 1 The percentage used in reconstructing the edge map image

Scale	No. of total coefficients	Percentage used	No. of used coefficients	Weight of each scale level
1	625	100%	625	1
2	8320	100%	8320	1
3	33488	100%	33488	1
4	129792	100%	129792	1
5	510848	1%	5108	0.9
6	2022976	1%	20229	0.9
Total	2706049	7.3 %	197562	

The reconstructed edge map, Figure 11, will be the input to the next step, which consist of gradient calculation, non-maximal suppression and hysteresis. The result from this stage is illustrated in Figure 12.

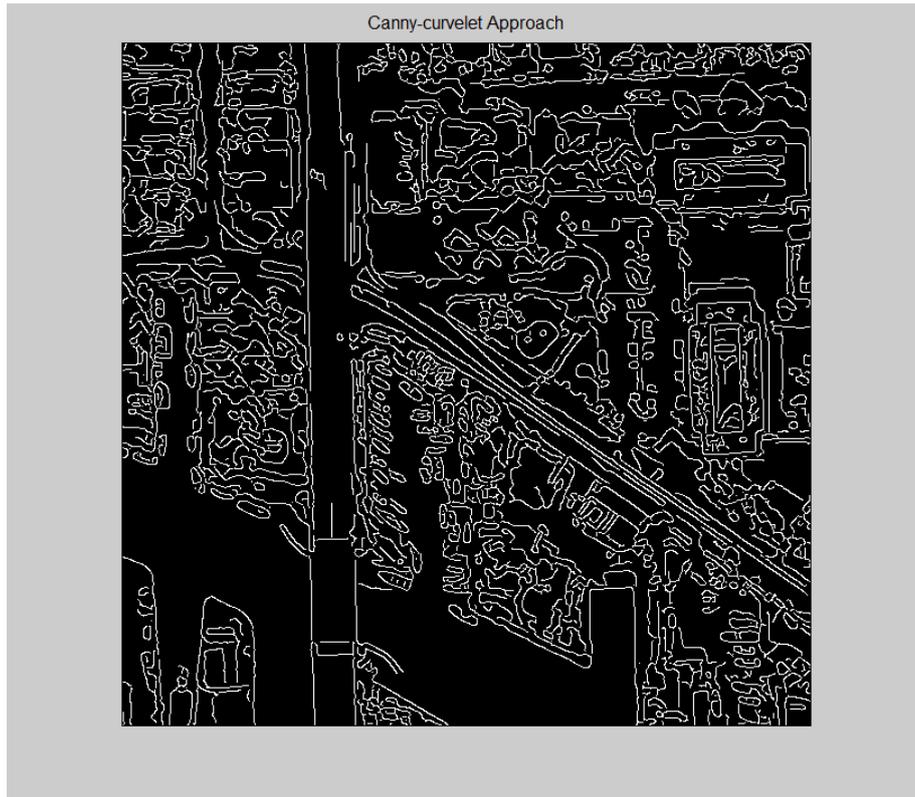


Figure 12 Edge detection result using the proposed algorithm

To illustrate the quality of this algorithm the result was compared with the traditional Canny, Figure 13. Details of the comparison are given in the next section.



Figure 13 Results with Canny Operator



Figure 14 Comparing certain areas in curvelet-canny and canny

Results and Discussion

Comparing the two figures, it was clear that Canny wasn't able to sieve or block many features leaving a lot of open boundaries, and also canny alone, was not able to detect many elongated structures, while curvelet-canny approach gave better results regarding more detection of elongated structures and more closed boundaries. Figure 14, enlarge some snapshots to illustrate these findings.

Conclusions

A curvelet-canny combined approach for edge detection was proposed and applied on high-resolution satellite imagery data, and repeated using the traditional canny operator, the results were promising. The curvelet-canny approach give close or even improved delineation to edges compared to Canny. The proposed method was able to detect longer edges and generate more closed objects. This method can be used as an alternative for the segmentation process, as it gives a closed boundary for almost all main features in the input image, in an object-based classification approach.

References

- [1] Candes, E., Demanet, L., Donoho, D. and Ying, L.X., 2006. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3): 861-899.
- [2] Candes, E.J. and Donoho, D.L., 2005. Continuous Curvelet Transform - I. Resolution of the wavefront set. *Applied and Computational Harmonic Analysis*, 19(2): 162-197.
- [3] Canny, J., 1986. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6)*: 679-698.
- [4] Donoho, D.L. and Duncan, M.R., 2000. Digital curvelet transform: Strategy, implementation and experiments. In: H.H. Szu, M. Vetterli, W.J. Campbell and J.R. Buss (Editors), *Wavelet Applications VII. Proceedings of the Society of Photo-Optical Instrumentation Engineers (Spie)*, pp. 12-30.
- [5] Elsharkawy, A., Elhabiby, M. and El-Sheimy, N., 2011. Second Generation Curvelet Transform for Building Extraction from High Resolution Satellite Imagery. *GEOMATICA*, 65(4): 387-399.
- [6] Elsharkawy, A., Elhabiby, M. and El-Sheimy, N., 2012. New Combined Pixel/Object-Based Technique for Efficient Urban Classification Using Worldview-2 Data, XXII International Society for Photogrammetry & Remote Sensing Congress, VII/4
- [7] Geback, T. and Koumoutsakos, P., 2009. Edge detection in microscopy images using curvelets. *BMC Bioinformatics*, 10(1): 75.
- [8] Guha, T. and Wu, Q.M.J., 2010. Curvelet Based Feature Extraction. In: M. Oravec (Editor), *Face Recognition. InTech*, Available from:
- [9] <http://www.intechopen.com/articles/show/title/curvelet-based-feature-extraction>, pp. 35-46.
- [10] Hui-qian, D., Fei, G. and Wen-bo, M., 2007. Edge detection for millimeter-wave images based on Curvelet transform. In: L. Zhou (Editor), *International Symposium on Photoelectronic Detection and Imaging. SPIE*.
- [11] Ma, J. and Plonka, G., 2009. Computing with Curvelets: From Image Processing to Turbulent Flows. *Computing in Science and Engg.*, 11(2): 72-80.
- [12] Ouma, Y.O., Tateishi, R. and Sri-Sumantyo, J.T., 2010. Urban features recognition and extraction from very-high resolution multi-spectral satellite imagery: a micro-macro texture determination and integration framework. *IET Image Processing*, 4(4): 235.

- [13] Xiao, M., Jia, Y., He, Z. and Chen, Y., 2008. Edge Detection of Riverway in Remote Sensing Images Based on Curvelet Transform and GVF Snake, Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Shanghai, P. R. China, , pp. pp. 344-351.
- [14] Zhenghai, L. and Jianxiong, Q., 2009. Study of Technique of Edge Detection Based on Curvelet Transform, Second International Symposium on Computational Intelligence and Design. IEEE computer society, pp. 543-545.