

IMPROVED EDGE DETECTION FOR SATELLITE IMAGES

Willard Mapurisa^{1,2,*}, George Sithole^{1,3}

¹ Geomatics Division, University Of Cape Town, Cape Town, South Africa - wmapurisa@luxcarta.com

² Luxcarta - wmapurisa@luxcarta.com

³ Cape Town, South Africa - Georgesithole76@gmail.com

Commission II, WG II/4

KEY WORDS: Edge Detection, Reconstruction, Object Detection, Building Detection, Satellite Images

ABSTRACT:

Edges are a key feature employed in various computer vision applications namely segmentation, object recognition, feature tracking and 3D reconstruction. Edges provide key information with regards to object presence, shape, form and detail which aid in many computer vision tasks. While there are various edge detection techniques in literature, challenges in edge detection remain. Varying image contrast due to non uniform scene illumination and imaging resolution affects the edge information obtained from any given image. The edge detection results are characterised by missing edges, edge fragmentation and some false positive edges. Gradient based edge detectors are the most commonly used detectors. These detectors all suffer from aforementioned challenges. In this, paper we present an edge detection framework that aims to recover long unfragmented edges from satellite images. This is achieved by using an edge accumulator that operates on the entire edge detection parameter space. Gradient based edge detectors rely on thresholding to retrieve salient edges. This usually results in missed or noisy edges. To counter this, the accumulator is run over a wide parameter space, growing edges at each accumulator level while maintaining edge position using a localisation filter. The results are longer unbroken edges that are detected for most objects, even in shadowy regions and low contrast areas. The results show improved edge detection that preserves the form and detail of objects when compared to current gradient based detectors.

1. INTRODUCTION

Edge features present contextual information and geometrical properties that have been utilised in various computer vision tasks. Application areas such as object detection and tracking, road detection and 3D reconstruction rely on accurate edge features. Edge features provide information with regards to object form and detail which is required in the various applications. Consequently, edge detection has remained an active area of research with the main goal being the ability to retrieve good edges from any given image (Ziou et al., 1998, Argialas and Mavrantza, 2004, Ofir et al., 2019).

Various algorithms have been proposed in literature that yield different results. Most edge detectors are gradient based and thus rely on significant intensity variations to detect edges. Recently, machine learning techniques have been explored in edge detection with the aim of improving edge detection edges under varying imaging conditions (Bertasius et al., 2015, Liu et al., 2017). Since most edge detectors use intensity variations in identifying edges, most of edge detection algorithms suffer from issues arising from low image contrast, shadows and image resolution. The effects of the aforementioned issues result in edge fragmentation, missing edges and incorrect edge localisation. Gradient thresholding is used to optimally select edge pixels in a given image while removing spurious pixels. However, variation of intensities across large image limit the global applicability of any given threshold. Thresholds vary from scene to scene and thus the performance on any given detector varies (Kaur et al., 2012). Canny edge detector has been found to perform best under different conditions. Canny edge detector has been widely used in various tasks that required salient, true edges with minimal false positives (Canny, 1986). To

improve detection, the canny edge detector utilises a threshold range with an upper and lower limit. This allows for the detection of edges with varying gradient magnitudes. Machine learning based techniques require training data for accurate detection. The major drawback for most machine learning based techniques is the lack of extensive training data sets that cater for the many types of topography, landscapes, cityscapes and imaging conditions.

This paper presents an edge detection framework that aims to leverage existing gradient based edge detectors in retrieving long unbroken edges which we will refer to as maximal boundaries. The framework also reduces edge fragmentation and missing edges while maintaining good localisation. This is achieved by using an accumulator space that is run on the entire parameter space. The main advantage of the proposed method is the ability to retrieve edges in varying imaging conditions, shadows and low contrast areas. In addition, any gradient based edge detector can be used, eliminating the need to find one optimal edge detector for a given scene. Finally, an improved line detection method is presented that preserves the true edge orientation in addition to edge position.

The next section presents a summary of previous work in edge detection. This is followed by a description of the methodology proposed in this paper in Section 2. After the methodology, results are presented in Section 3 together with a discussion of results. Finally, conclusions are drawn in the final section.

1.1 Previous Work

Edge detection has served as one of the many prerequisites for many computer vision tasks due to information contained such as length, orientation and object boundaries (Savant, 2014, Ok, 2008). Because of this, a plethora of algorithms exist for edge

* Corresponding author

detection. Various reviews of these detectors and their performance have been presented (Ziou et al., 1998, Koschan, 1995). Mostly, edges in images are detected using a gradient operation on an image, where areas of high gradient magnitude difference indicate the possible presence of an edge. Consequently, different first order gradient operators have been proposed for use in edge detection namely Prewitt (Prewitt, n.d.), Sobel (Sobel and Feldman, 3), Scharr (SCHARR, 2000), Kirsch and Roberts (Roberts, 1963). Other methods such as Laplacian zero crossing (second order derivatives) and Hough transform exist for detecting edges. The main variation with these algorithms are the mathematical complexity of derivatives employed. Higher order derivatives more computationally expensive (Baştürk and Günay, 2009, Argialas and Mavrantza, 2004) compared to first order derivatives. Multi-scale approaches aimed at improving the performance of detectors at different scales, luminosity and noise have also been proposed (Ren, 2008, Ofir et al., 2019). More recently, edge detectors based on deep learning have been proposed as an alternative (Waldner and Diakogiannis, 2020, Wang et al., 2020, Liu et al., 2017, Bertasius et al., 2015, Xie and Tu, 2015). With learning algorithms, the major drive is to improve edge detection using training sets for different edge contexts. However, having sufficient training data that caters for all types of edges remains a drawback. Even with deep learning based methods, thresholding and post processing is at times required to retrieve the final edges (Waldner and Diakogiannis, 2020). Other Various other edge detectors can be found in (Bhatti et al., 2021, Abraham and Wloka, 2021, Adamos and Faig, 1993, El-Sayed et al., 2013, Smith and Brady, 1997). The limitation in edge detection is that there is no universal edge detector that is invariant in different imaging contexts. This has led to the proposal of task specific edge detectors that operate optimally for a given context (Ziou et al., 1998). Canny edge detector has been the widely accepted and used due to its ability to detect edges using two thresholds. In addition, by using non-maxima suppression, canny edge detector returns pixel wide edges which has been another challenge in edge detection (Canny, 1986). Other detectors work well but require single thresholds to filter out weak, non significant edges and noise effects (Ofir et al., 2019). Due to the variance in imaging properties, different thresholds on the resulting gradient magnitude values are required.

Three main operations characterise the edge detection process. These operations are image smoothing, differentiation using first or second order derivatives, and finally labelling and chaining. Prerequisite to edge detection is smoothing. Image smoothing reduce the effects of noise from high frequencies as a result of high signal to noise ratio (Szeliski, 2010). The level of smoothing thus has a direct effect on the quality of edges that are detected. Various smoothing operations exist with the most popular ones being Gaussian smoothing and bilateral filter (Goyal et al., 2012). Image smoothing introduces other undesirable effects such as information loss, edge smoothing and slight feature displacement (Ziou et al., 1998). Some of the smoothing effects are lose edge blending, contrast reduction, object displacement. These effects can be reduced by selecting optimal smoothing parameters. The same applies to gradient thresholds (Kaur et al., 2012, Ofir et al., 2019). Good threshold ranges have been proposed for various edge detectors such as Canny (Lu et al., 2015, Biswas and Sil, 2012) that keep the most significant edges while filtering out weak and noisy edges.

A variable filter approach has been adopted in some cases to reduce the effects of noise on local small filters such as Sobel

filters, such that discontinuities are detected at varying scales (Ali and Clausi, 2001). While these effects might be minimal in small scenes typically acquired by ground based cameras, they become more pronounced in large images with reduced resolution such as satellite images. The inability to create continuous edge boundaries that describe an object fully from any edge detector has been the major drawback in the global applicability of edge detectors solely in object detection and boundary delineation.

As previously mentioned, the problems encountered in edge detection usually result in edge fragmentation, missed edges and some spurious edge pixels. Due to these difficulties, different edge detectors have been proposed each with advantages and disadvantages. For example, Lu et al (Lu et al., 2015), presents and adaptive canny edge detector where the gradient thresholds are automatically computed using the Helmholtz principle based on an input image. This improves the performance of the canny detector thereby reducing edge fragmentation. Similarly, Biswas et al (Biswas and Sil, 2012), uses Type-2 Fuzzy Sets to estimate the optimal thresholds for the canny edge detector in an effort to improve the edges detected. Akinlar et al (Akinlar and Topal, 2011), presents a parameter free edge detection algorithms that detects edges by using extreme parameter values for the Edge Drawing (ED) algorithm. Since running the algorithm using parameter extrema results in false positives, an "a cantario" approach based on the Helmholtz principle is utilised to remove outliers. Other efforts for automated edge detector selection have been explored (Bausys et al., 2020). Topal et al (Topal and Akinlar, 2012), presents a method that creates a list of unbroken edge chains which are better than the traditional canny edge detector. After smoothing the image with a low pass filter, gradient magnitude and edge direction maps are created. Anchors are extracted from the gradient maps based on gradient magnitude. The anchors are then connected based on direction and magnitude thereby creating an uninterrupted continuous edge. However, since thresholding is used to eliminate weak pixels, edges are still under-detected at certain thresholds. Even though the output is longer unbroken edges, a single edge usually connects different object boundaries and do not form closed boundaries. However, since single edge chains are at times constituted from multiple object edge boundaries, the edges lack semantic information about individual objects or their boundaries in the image. For extracting chained vector boundaries, this method performs well and will be incorporated in this paper.

The main parameters associated with most edge detectors are the smoothing parameter and the gradient magnitude thresholding parameter. These parameters are usually fixed for a given scene. However, this is not ideal, especially for large scenes with varying illuminating conditions and contrast. In this research, we propose the use of a multi-detector framework that uses an edge growing function while maintaining good edge localisation. This is achieved by growing the edges using a wide parameter space in an accumulator space, which results in maximal edge boundary retrieval.

2. EDGE DETECTION FRAMEWORK

Edge detection has been extensively explored with many algorithms proposed. However, the major drawbacks presented by varying image contrast and resolution still hamper the successful retrieval of edges. What has not been fully explored is how existing edge detectors can be used in a framework

that results in maximal boundary extraction while maintaining good edge localisation and positional consistency. This paper presents a framework that allows for maximal boundary retrieval, while eliminating the need for fixed parameters for smoothing and gradient magnitude thresholds. To introduce genericity, the algorithms uses any gradient operator and traverses the entire parameter space building up the edges in an accumulator space.

Generally, most edge detectors begin by an image smoothing operation to avoid the effect of noise due to high frequencies as a result of high signal to noise ratio. This is then followed by applying a gradient operation on the smoothed image and then thresholding the gradient values to identify areas of high intensity changes, which is the property used in detecting edges. The identified pixels are deemed to be the edge pixel. Finally, a thinning operation is applied to reduce edge width since the effects of smoothing, low resolution and ramp edges. Ramp edges captured at low resolution results in the failure to distinguish the start and end of the ramp but results in connected double edges. The thinned single pixel width edges are then chained and vectorised.

The proposed approach in this paper follows a similar methodology rather than using a defined smoothing and gradient operator with fixed thresholds, a set of operators can be defined and applied to the image. In, addition, no thresholding or parameters inputs are required for any of the smoothing or gradient operators. The smoothing and gradient operators are applied in the entire parameter space. Since multiple parameters and threshold combinations are applied, the result of each threshold and smoothing parameter combination is an accumulator or edge layer. Each resulting layer of a given parameter combination is merged with the previous layer. To avoid edge positional differences when merging layers, a localisation filter is used. The localisation or positional filter ensures positional consistency of the edges across layers while growing edges in the accumulator. The filter also avoids edge duplication in the resulting edges. The result is maximal boundaries retrieved which capture form, detail and boundaries of objects in an image. Line retrieval is then applied to the final edge layer. The framework also allows any pre-processing operators before edge detection to be applied. The following sections detail each of the steps involved in the framework.

2.1 Image smoothing

Image smoothing has proven to yield better edge detection results by suppressing noise while preserving edges (Szeliski, 2010, Goyal et al., 2012). One of most common smoothing functions used are the Gaussian smoothing function, bilateral filter and Green's function. These smoothing functions require the selection of ideal smoothing parameter, for example σ for Gaussian smoothing and μ for Green's function. Green's function is defined as:

$$f_{gr}(x) = \frac{\mu}{2\sqrt{2}} e^{-\mu|x|} \left(\cos \frac{\mu|x|}{\sqrt{2}} + \sin \frac{\mu|x|}{\sqrt{2}} \right) \quad (1)$$

and the Gaussian function is given by:

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

where x, y = kernel coordinate offsets

Since there is loss of information when suppressing noise, the challenge comes to selecting optimal parameters for smoothing. To overcome this bottleneck, we define a range for the smoothing parameters. We can therefore define a set of smoothing functions f_s by changing the smoothing parameter to a function parameter. Using the Gaussian function as an example, the set of smoothing functions f_s can now be defined as

$$f_s = \{f_g(x, y, \rho) \mid \rho \in [a, b]\} \quad (3)$$

where ρ = smoothing parameter as a function variable

This implies any smoothing function in the set can be applied with varying smoothing parameters over a parameter space with range $[a, b]$. The range of a and b must be large enough to contain the optimal and extreme values of μ and σ , assuming we use Gaussian blurring or Green's function.

2.2 Gradient Operators

Similar to smoothing, we define the gradient based operation that establishes the gradient magnitude and direction as

$$G = g(x, \psi); \psi \in [m, n] \quad (4)$$

where ψ = threshold value for identifying edge pixels
 x = image

This also implies G exists in a parameter range $[m, n]$. The set G_s of gradient operators can now be defined as

$$G_s = \{G_g(x, \psi)\} \quad (5)$$

where x = gradient magnitude

2.3 Localisation Filter and Vectorisation Function

A localisation filter is a mask that ensures edges layers are grown while maintaining the correct edge position. Using different operators and thresholds often results in a shift in the edge position in the layers. For example, applying different smoothing parameters, object resolution is reduced. In addition to resolution reduction, object edges are slightly shifted. The localisation filter serves the purpose of merging the same edges albeit shifted as a result of smoothing. Directly merging two edge layers created using different smoothing operators results in double and spurious edges. This is illustrated in Figure 1. Edge e_a and e_b is the same edges but with a slight positional shift. Growing these two layers will result in the double edge effect.

However, merging the layers must result in the single edge without double or spurious edges as shown in Figure 2. This is achieved by using a localisation filter that assesses the same neighbourhood in both layers to ascertain if there are duplicate

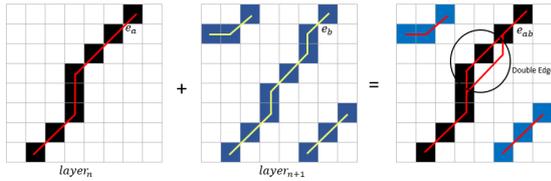


Figure 1. Effects of merging layers with edges e_a and e_b that have shifted edge positions. The resulting edge e_{ab} has an extra double edge created which results in noisy edges.

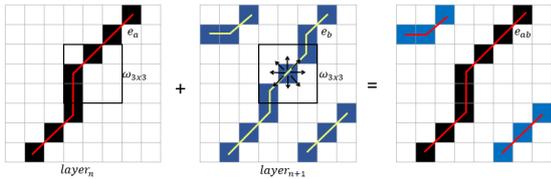


Figure 2. Correct merging is achieved using a localisation filter that ensures no duplicates or double edges. The result is a clean edge map as shown by resulting edge e_{ab} .

edge pixels that have been shifted. The result retains the main edge without introducing noisy or shifted edges.

The localisation kernel, $\omega_{3 \times 3}$, used in ensuring a clean edge map is defined as

$$\omega_{3 \times 3} = \begin{cases} 1 & g_x \geq \psi \\ 0 & g_x < \psi \end{cases} \quad (6)$$

The evaluation of the filter works by establishing the presence of any edge pixels in the filter neighbourhood. The filter returns true if any matching pixel exists in the neighbourhood.

The output from the filter is a binary image at each accumulator level. The binary image is vectorised. The edges obtained from the vectorisation are grown at every level. The growth function uses a voting strategy to identify exiting and new edges. In this paper, the vectorisation presented by (Topal and Akinlar, 2012) is used. However, any vectorisation function can be used.

We define the vectorised layer l_{v_t} at level t in the accumulator as.

$$l_{v_t} = \nu_t(g(x, \psi)) \quad (7)$$

where $\nu_t(g(x, \psi))$ is the vectorisation operation at accumulator level t . All the above mentioned functions are then implemented in the accumulator space in the edge growing process. This is detailed in the next section.

2.4 Accumulator Function

Since many detectors exist, users have the onerous task of selecting a fit for purpose detector. In addition, optimal parameters for image smoothing and gradient thresholding, which are all data dependent, are required. To eliminate these requirements, an accumulator based approach has been developed. The smoothing and gradient thresholding functions have been defined over the parameter space as opposed to using constants. The inputs to the accumulator function are smoothing and thresholding parameter ranges. Other optional inputs are

smoothing and gradient computation functions. These functions can also be fixed in the accumulator. Initially, the accumulator space is empty, then starting from the highest values in the parameter space for gradient threshold and lower values the smoothing functions, an edge map is produced for each set of parameters. This edge map, alternatively named edge layer, is termed the anchor layer since we know that higher values of ρ and low ψ values yield correct edges albeit with high edge omission levels.

Iterating through the parameter space creates a layer with every parameter set. When a new layer is created at every accumulator level, the previous or anchor layer is grown using new edges found in the current layer level while also voting for duplicate edges. The voting is an accumulation of the number of hits an existing edge pixels receives at each layer level. Edges are grown if, for an existing edge, partial votes are received from the new edge leaving candidates in the new edge that have no votes, which implies an extra edge section that has been identified. Layers are not stored at each level, but rather used to grow the anchor layer, based on the voting mechanism, which preserves memory and storage. At every region growing level, the localisation filter is applied to identify duplicate but misaligned edge pixels and avoid pixel wide parallel edges. In the process of creating layers at every level, new edges without partial votes are encountered. These edges are added as new edges after passing through the localisation filter. If no votes are received, this implies a new edge that is added to the edge layer. This eliminates the emergence of spurious edges down the accumulator levels. In this paper, the accumulator gradient and smoothing function sets are predefined in the accumulator space.

The region growing function R , which is applied to the vectorised edge layer obtained from Equation 7 and the vectorised anchor layer, is defined as

$$Rv_t = R(l_{v_t}, l_{v_{t_{anc}}}, \omega) \quad (8)$$

We make the localisation kernel ω , defined in section 2.3, a function parameter to allow for new improved merging filters to be to the accumulator function as a parameter.

where l_{v_t} = vectorised output edge layer at level t in the accumulator
 $l_{v_{t_{anc}}}$ = vectorised anchor layer that is continually grown

The algorithm of the framework described here is summarised in Algorithm 1. After the final layer is created, the result is a detailed edge map with both edges showing object form and detail. Since most of these edge are open curves, Edges that are close together are snapped to form closed boundaries where possible.

2.5 Line Retrieval

The edges produced from the accumulator framework are curves or linestrings that have no descriptors or parameters attached. These curves usually have a stair stepped appearance from the edge thinning and vectorisation process. Image object boundaries are known to be composed of connected line and elliptical curve segments that are connected to form closed boundaries. The final step is to retrieve true lines and curve segments while removing the stepped effects. For this a new line

Algorithm 1 Accumulator Framework Algorithm

```

Require:  $l_{anc} = Null$ 
Require:  $t = 0$ 
Require:  $\rho_{step} = 1$ 
Require:  $\psi_{step} = 4$ 
Require:  $x$  is not  $Null$  ▷ input image
for  $\rho = b$  to  $a$  do
  for  $\psi = m$  to  $n$  do ▷ Start with highest threshold
    for  $g$  in set  $g_s$  do
       $f_t \leftarrow f_g(x, \rho)$  ▷ Smoothing filter set
       $g_t \leftarrow g(f_t, \psi)$ 
      if  $l_{vtanc}$  is  $NULL$  then
         $l_{vtanc} \leftarrow v_t(g_t)$ 
      else
         $l_{vtanc} \leftarrow R(l_{vt}, l_{vtanc}, \omega)$  ▷  $\omega$  a variable
      end if
       $t \leftarrow t + 1$ 
    end for
     $\psi \leftarrow \psi - \psi_{step}$ 
  end for
   $\rho \leftarrow \rho - \rho_{step}$ 
end for
 $l_{lines} \leftarrow L(l_{anc}, l_{min})$ 

```

fitting algorithm is proposed. Most line fitting methods minimise the point to line distance d for any given point on the line using Equation 9.

$$f(t, \theta) = \arg \min_{\theta} \sum_{i=1}^n d(t_i, \theta) \quad (9)$$

where θ = line parameters
 t_i = line point i

The line is then obtained by using the parameters θ to retrieve the line l_a .

$$l_a = L(t, \theta) \quad (10)$$

where L = the line retrieval function
 t_n = point set of size n used in computing θ

The Lines detected with Equation 9 using image edge points have a slight error in the direction of the line. This is as a result of object corners. Points at object corners at times introduce a slight rotation in the resulting line. This is because most line fitting methods distribute the errors equally using a fixed distance function. For example, suppose a line is fit with a maximum error threshold of ϵ , the line l_a is obtained since the corner edge pixels fall within this threshold instead of l_b as illustrated in Figure 3.



Figure 3. Rotational line fitting error introduced by corner edge pixels that fall within error threshold ϵ defined for the line l_a . The desired line is l_b .

To correct for this error, the pixel errors at the end of the line need to be excluded in the fitting. This is achieved by performing another line fit from the midpoint of the line going outwards towards the line endpoints. This allows less rotational freedom of the line in the final result. Similarly in Figure 4. The desired line is either l_b or l_c and not l_a which has a directional error compared to the true direction.

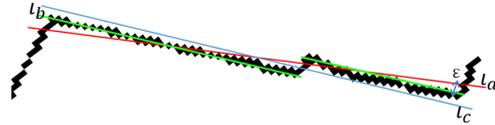


Figure 4. Rotational line fitting error introduced by corner edge pixels that fall within error threshold ϵ defined for the line l_a . The desired line is l_b or l_c . As shown with l_b and l_c , line direction is maintained.

A new fitting procedure is proposed in this research. Initially a line is fit using Equation 9. This gives us the line which might have the rotational error as a result of corner points. After obtaining this initial line, we correct for the rotational error if it exists. This is achieved by reducing the line points on either ends by a small percentage g and then computing $f(t, \theta_g)$. The results are then used to compute δr , the difference in direction between the directions computed by $f(t, \theta)$ and $f(t, \theta_g)$. If this exceeds a set threshold ϵ , then parameters for $f(t, \theta_g)$ are adopted otherwise $f(t, \theta)$ is kept as the correct parameters for the line. This computation can be enhanced by changing the value of g iteratively as long as δr remains less than ϵ with the first iteration as the reference direction. The initial iteration must contain at least half the line points used to obtain $f(t, \theta)$ centered around the line midpoint. Resultantly, parameters computed from the last iteration with δr below ϵ are adapted as the correct parameters. This process is depicted in the Figure 5 below where the last iteration before the directional error threshold is exceeded is chosen. ϵ value is usually set to 2° .

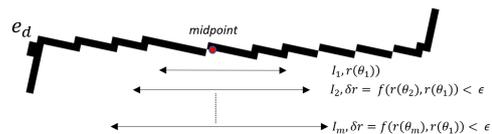


Figure 5. Line fitting with directional error control. Fitting starts from the midsection of the line going outwards maintaining a line direction variation δr that is less than a threshold ϵ at each iteration l_m .

This step is used to refine line direction for all lines obtained by least squares and the improved line fitting results are shown in the next section.

3. RESULTS

The framework was tested on 50cm satellite imagery. The focus was on specific cases to test performance of the framework under difficult conditions where most edge detectors would omit valid edges. The range for ρ , for smoothing is automatically computed from the window size ranging from 3 to 12 pixels using the Gaussian and Bilateral Filter. The gradient threshold ψ ranges from 36 down to 2 for our experiments but may cover a

large range. Three gradient operators are set in the accumulator namely Sobel, prewitt and Scharr.

The first result was a test on line detection on a tennis court with limited contrast. The results were compared with three most popular line detectors namely Line Segment Detector (LSD) (Grompone von Gioi et al., 2012), Edge Drawing Lines (EDLines) (Akinlar and Topal, 2011) and Hough Lines. Hough Lines makes use of the canny edge detector for detecting the lines. Figure 6 shows the results obtained from all algorithms including the method proposed in this paper. Top left are results from LSD using default parameters supplied by the authors. Top right are the results from EDLines followed by Hough Lines on the bottom left. Finally the algorithm presented here is shown on the bottom right. The algorithm detects more boundaries compared to the other methods. In addition, the benefit of the line fitting method is evidenced by the linear and perpendicular nature of neighbouring lines.

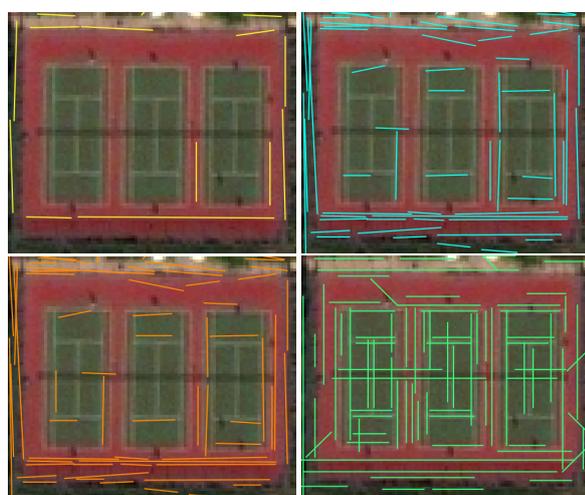


Figure 6. Edge detection around a tennis court with varying edge strength. Top left are edges detected using LSD, top right are edges from EDLines. Bottom Left are edges obtained from Hough lines and bottom right is the edge detection framework.

To illustrate the performance of the framework under varying contrast and shadowy areas, a building shown in Figure 7 was selected. Improved line detection is shown with the method presented in this paper in the bottom right image. As seen from the image, most boundaries that are missed by the other methods are detected by the method proposed in this paper. A line is found for most of the building edges. This is an added advantage in boundary retrieval algorithms. However, some over-detected edges can be seen on the topmost structure.

To, illustrate the line fitting results, an overlay of all 4 methods was done on a building and this is shown in Figure 8. As shown in the circled area, the corner edge pixel effect affects most of the methods in retrieving correct line orientations. The method detailed in section 2.5 improves line orientation. Finally, as shown in the parking lot image in figure 9, small feature detection with limited contrast was also achieved while long unbroken lines are also retrieved.

3.1 Quantitative Evaluation

To evaluate the level of detection, the detected edges were compared to ground truth edges. The ground truth edges of the tennis court in 6 were manually digitised. The comparison was based on assessing the true length of real edges against the

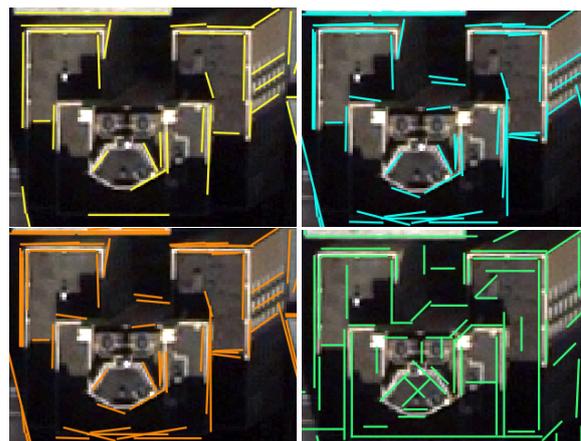


Figure 7. Edge detection in low contrast and shadowy areas. Top left are edges detected using LSD, top right are edges from EDLines. Bottom Left are edges obtained from Hough lines and bottom right is the edge detection framework



Figure 8. Overlaid results of lines detected by LSD (Yellow), EDLines(light blue), Hough Lines (Orange) and edge accumulator (green). The effects of edge pixels can be seen around the red circles on the slightly rotated lines.

detected edges. Firstly, detected and ground truth edges were paired using a proximity threshold of 1.5 pixels. Pairs were established by selecting the closest near parallel edges. A one to one mapping was enforced to avoid one to many line pairings. From these matches, the number of correct edges and overall pixel coverage was computed and the results are presented below.

All ground truth and detected edges were converted to pixels and overlaid. Firstly the total edge pixels from the ground truth was computed and used as the baseline. At each accumulator level acc_L , the number of edges detected that has a match in the ground truth edge list ED was recorded. The percentage of matched edge pixel EP against the baseline was also computed. Next, the pixel coverage PC which is the percentage of detected edge pixels that where matched with the baseline and finally the amount of extra pixels RG from each region growing process at a given accumulator level. Only the accumulator levels that returned valid edges where recorded out of 243 levels. From the table, the accumulator contributed approximately 37% more edge pixels from the initially detected 52% from a single parameter set. The blue edges in Figure 10 are outliers that where not matched to any ground truth edges.

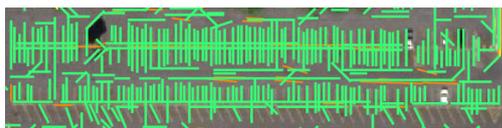


Figure 9. Parking lot with small lines also detected with the bulk of the parking edges detected

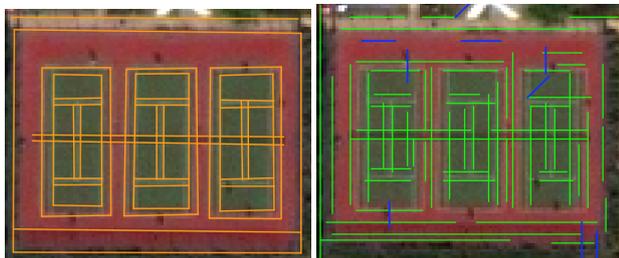


Figure 10. Ground truth image and detected Edges of tennis court used in evaluation

3.2 Discussion

The framework presented in this paper has the ability to detect more edges under difficult imaging conditions. This is achieved by using the entire parameter space to discover edges while minimising spurious edges. In this paper, two parameters are traversed without the need for fixed threshold values. These parameters are gradient magnitude and smoothing parameter. The smoothing and gradient functions can also be added in as accumulator parameters. In this paper, two smoothing and three gradient operators are used. This allows for both strong and weak edges to be retrieved while leveraging the advantages of different gradient functions. There are no scaling requirements of the input image. Since the resulting dense edge map has more edges, any subsequent line detection and fitting benefits from the edge map as illustrated in the different scenarios in the result section.

The main advantage of the framework is the ability to produce dense edge maps compared to other detectors. However, the disadvantage is that run-time of the algorithm increases with the number of smoothing and gradient operators. This can potentially be reduced by running accumulator levels in a parallel manner and then fusing more than two layers. The framework improves detected edge lengths using edge growing, which in turn improves detected edges of both object form and detail. While the research presented here is limited to satellite images, the framework can potentially be applied to other types of large area images such as aerial and drone images and this will be the subject of future research.

The line fitting with skewness control increases the line orientation accuracy as shown in Figure 8. This is in addition to the increase in the average length of the lines retrieved from the framework. However, the robustness of the line fitting requires further investigation. The main disadvantage is that there are false positives found as with the other detectors. Current work is on minimising false positives and detecting curves in the output. A method for filtering out false positives is required. Current research is on the use local neighbourhood gradients magnitudes of salient edges to identify very weak edges. In addition, the use of line arrangements and local edge patterns based on gestalt principles to identify edges that do not follow local neighbourhood patterns is being explored.

Edge Detection Evaluation				
<i>accL</i>	<i>ED</i>	<i>EP</i> (%)	<i>PC</i> (%)	<i>RG</i> (%)
0	28	52.2219	52.2219	0
1	3	4.63325	56.8552	4.63325
2	1	0.92665	57.7818	5.5599
4	1	2.34388	60.1257	7.90378
5	1	0.872141	60.9978	8.77592
6	1	0.599597	61.5974	9.37551
7	1	0.817632	62.4151	10.1931
9	1	0.654106	63.0692	10.8473
12	1	0.872141	63.9413	11.7194
13	1	1.09018	65.0315	12.8096
18	1	1.19919	66.2307	14.0088
19	1	0.981158	67.2118	14.9899
20	1	0.709952	67.9218	15.6999
27	3	4.03468	71.9565	19.7346
28	2	1.68388	73.6403	21.4184
30	1	0.708614	74.349	22.1271
32	1	4.52454	78.8735	26.6516
33	1	2.23486	81.1084	28.8864
36	2	1.79845	82.9068	30.6849
47	2	2.77995	85.6868	33.4648
50	2	1.47174	87.1585	34.9366
206	1	0.922637	88.0811	35.8592
224	1	0.708614	88.7897	36.5678
232	1	1.19919	89.9889	37.767

Table 1. Accumulator rate of detection and edge growth

4. CONCLUSION

This paper presents an edge detection framework that is neither case specific nor fixed to any gradient operator. Instead, the framework operates using generic operators that are executed in the entire parameter space. Good edge localisation is maintained and maximal boundaries are retrieved even in dark shadowy areas. The need for selecting optimal parameters is eliminated by defining a parameter range where edge detection is carried out. In addition, a line retrieval algorithms that reduces line orientation errors is presented. The accumulator framework aims to introduce genericity in edge detection, where new detectors and smoothing operators can be added to the set and used to detect edges in a region growing manner. The result is an improved detection rate even in areas where objects are not visible to the human eye.

The result is a dense set of edges that can be further processed to retrieve lines and curves with one major advantage being boundary completeness. While there might be many levels, to reduce the memory footprint, only consecutive accumulator level are kept and passed to the region growing function and then retaining only the result. Since most edge points at each level are duplicates they are not re-processed but rather passed as votes to existing edges. The framework produces a dense edge map which benefits the subsequent line detection process as show in the results. The new line fitting approach produces lines with less directional error. The skewness encountered with the other methods is reduced.

Current work is on minimising framework runtime and detecting and filtering out false positives. Future work will look at improving curve retrieval from the dense edge network and recovering closed boundaries from the dense edge map. In addition, benchmarking the performance of the accumulator space during region merge is required in order to optimise the framework performance. Investigation on the impact if any on the use of different functions and order in the accumulator will also benefit this research.

REFERENCES

- Abraham, J., Wloka, C., 2021. Edge detection for satellite images without deep networks. *arXiv preprint arXiv:2105.12633*.
- Adamos, C., Faig, W., 1993. Hough transform in digital photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 29, 250–250.
- Akinlar, C., Topal, C., 2011. Edlines: Real-time line segment detection by edge drawing (ed). *2011 18th IEEE International Conference on Image Processing*, 2837–2840.
- Ali, M., Claudi, D., 2001. Using the canny edge detector for feature extraction and enhancement of remote sensing images. *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, 5, IEEE, 2298–2300.
- Argialas, D., Mavrantza, O., 2004. Comparison of edge detection and Hough transform techniques for the extraction of geologic features. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part XXX).
- Baştürk, A., Günay, E., 2009. Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. *Expert Systems with applications*, 36(2), 2645–2650.
- Bausys, R., Kazakeviciute-Januskeviciene, G., Cavallaro, F., Usovaite, A., 2020. Algorithm selection for edge detection in satellite images by neutrosophic WASPAS method. *Sustainability*, 12(2), 548.
- Bertasius, G., Shi, J., Torresani, L., 2015. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4380–4389.
- Bhatti, U. A., Ming-Quan, Z., Qing-Song, H., Ali, S., Hussain, A., Yuhuan, Y., Yu, Z., Yuan, L., Nawaz, S. A., 2021. Advanced Color Edge Detection Using Clifford Algebra in Satellite Images. *IEEE Photonics Journal*, 13(2), 1–20.
- Biswas, R., Sil, J., 2012. An improved canny edge detection algorithm based on type-2 fuzzy sets. *Procedia Technology*, 4, 820–824.
- Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 679–698.
- El-Sayed, M. A., Estaitia, Y. A., Khafagy, M. A., 2013. Automated edge detection using convolutional neural network. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 4(10).
- Goyal, A., Bijalwan, A., Chowdhury, M. K., 2012. A comprehensive review of image smoothing techniques. *International Journal of Advanced Research in Computer Engineering and Technology*, 1(4), 315–319.
- Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., Randall, G., 2012. LSD: a Line Segment Detector. *Image Processing On Line*, 2, 35–55. <https://doi.org/10.5201/ipl.2012.gjmr-lsd>.
- Kaur, J., Agrawal, S., Vig, R., 2012. A comparative analysis of thresholding and edge detection segmentation techniques. *International Journal of computer applications*, 39(15), 29–34.
- Koschan, A., 1995. A comparative study on color edge detection. *Proceedings of the 2nd Asian Conference on Computer Vision*, 3, ACCV Singapore, Singapore, 574–578.
- Liu, Y., Cheng, M.-M., Hu, X., Wang, K., Bai, X., 2017. Richer convolutional features for edge detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3000–3009.
- Lu, X., Yao, J., Li, K., Li, L., 2015. Cannylines: A parameter-free line segment detector. *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 507–511.
- Ofir, N., Galun, M., Alpert, S., Brandt, A., Nadler, B., Basri, R., 2019. On detection of faint edges in noisy images. *IEEE transactions on pattern analysis and machine intelligence*, 42(4), 894–908.
- Ok, A. Ö., 2008. Robust detection of buildings from a single color aerial image. *Proceedings of GEOBIA*, 6.
- Prewitt, J., n.d. Object enhancement and extraction picture processing and psychopictorics. 1970.
- Ren, X., 2008. Multi-scale improves boundary detection in natural images. *European conference on computer vision*, Springer, 533–545.
- Roberts, L., 1963. *Machine Perception of Three-dimensional Solids*. Its Technical report, M.I.T. Lincoln Laboratory.
- Savant, S., 2014. A review on edge detection techniques for image segmentation. *International Journal of Computer Science and Information Technologies*, 5(4), 5898–5900.
- SCHARR, H., 2000. Optimal operators in digital image processing dissertation.
- Smith, S. M., Brady, J. M., 1997. SUSAN: a new approach to low level image processing. *International journal of computer vision*, 23(1), 45–78.
- Sobel, I., Feldman, G., 3. A 3x3 isotropic gradient operator for image processing, presented at a talk at the Stanford Artificial Project. *Pattern Classification and Scene Analysis*, 271–272.
- Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Topal, C., Akinlar, C., 2012. Edge drawing: a combined real-time edge and segment detector. *Journal of Visual Communication and Image Representation*, 23(6), 862–872.
- Waldner, F., Diakogiannis, F. I., 2020. Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sensing of Environment*, 245, 111741.
- Wang, D., Li, C., Song, H., Xiong, H., Liu, C., He, D., 2020. Deep Learning Approach for Apple Edge Detection to Remotely Monitor Apple Growth in Orchards. *IEEE Access*, 8, 26911–26925.
- Xie, S., Tu, Z., 2015. Holistically-nested edge detection. *Proceedings of the IEEE international conference on computer vision*, 1395–1403.
- Ziou, D., Tabbone, S. et al., 1998. Edge detection techniques—an overview. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, 8, 537–559.