Power line detection using Hough transform and line tracing techniques

Lewis Baker, Steven Mills, Tobias Langlotz
University of Otago
Dunedin, New Zealand
Email: bakelew@gmail.com, steven@cs.otago.ac.nz, tobias.langlotz@otago.ac.nz

Carl Rathbone
LineSmarts Ltd.
Dunedin, New Zealand
Email: carl@linesmarts.com

***Abstract*—Fast power line detection in images is useful in photogrammetry applications such as measuring wire tension and sag. To make these measurements, images of entire power line spans must be used which may include large amounts of curvature. Previous work in power line detection has focused on aerial or close proximity images where no power line curvature is visible. This paper assesses feasibility of ground-based imaging utilising smartphone cameras together with fast and robust power line detection using two common Hough transform techniques, and a line tracing algorithm.**

I. Introduction

Overhead power lines are a common structure in urban and rural areas that require regular maintenance and inspection. Inspection procedures can be difficult and dangerous for linesmen for many reasons such as the large amount of electrical energy in the conductors, and the height of the poles above the ground. It would be useful if these inspection procedures could be completed remotely using cameras and photogrammetry techniques. LineSmarts is a mobile application that can perform these kinds of procedures [1]. It is able to make measurements of properties such as conductor tension, sag, and span length, while the user remains at a safe distance from the structures.

One of the challenges with this application is how to automatically detect power lines in the images. In particular, these images will be taken from ground level at medium-to-long distances, so the power lines will show large amounts of curvature in the image, such as the example in Fig. 1. There has been previous work in automatic power line detection, but these works generally assume an aerial perspective where the power lines form straight lines in the image.

However, straight line detection is not appropriate for measuring these properties of power lines. To measure sag, the curvature of the power line must be visible, and the whole span must be in the frame. So the problem is how to detect curves in images that approximately fit a parabolic shape. Another problem is that the power lines will be photographed from various angles, which results in perspective transformations on the parabolic curves.

Yan *et al.* use a Radon transform to detect power line segments which are linked at the endpoints using a Kalman filter [2]. Their solution is robust to noisy backgrounds, and can allow small amounts of curvature, but makes many assumptions about the properties of the lines in the image which cannot be made for mid-range terrestrial images.

Fig. 1. Power line fitting using 3 manually selected points on the curve (top) versus our automatic power line detection method.

Zhang *et al.* expand upon this, using a Hough transform to detect power lines in UAV imagery, and is able to achieve real-time performance for power line tracking [3]. The lines of interest are clustered together into one line, with the intention of using this line to guide the UAV aircraft.

Chen *et al.* use an improved Radon transform to detect power lines from high resolution satellite imagery [4]. They also present an algorithm for reliably distinguishing straight power lines from noise in the Radon space. However, they make similar assumptions to [2] such as the low curvature, and parallelism of the lines.

Aside from Hough and Radon transform techniques, there has also been work on power line detection using line tracing

978-1-5090-2748-4/16/$31.00 ©2016 IEEE

Fig. 2. Example of our method for power line detection. Clockwise from top-left: Input image, perspective transformation is applied, image is filtered, power lines are detected, reverse perspective transformation is applied to the wires, separate lines can be distinguished in the original image.

algorithms. Wan *et al.* use a line tracing algorithm which is able to extract curved power lines from aerial images. The starting points for the line tracer were calculated with prior knowledge about the 3-D co-ordinates of the power poles [5].

To our knowledge, there has been no previous work in detecting power lines in mid-range terrestrial images, such as the image in Fig. 1. However, there has been work in parabola detection in images. Some examples of this work include [6] where the authors employ a Hough transform to detect the boundaries of rib bones in X-ray images, and [7] where the authors apply a Hough transform to detect eyelashes and eyelids in normalised images.

The nature of the Hough transform allows it to be adapted to detect more complex shapes such as ellipses, or curves at the cost of more expensive computation [8]. This paper discusses the strengths and weaknesses of Hough transform approaches for automatically detecting power lines in images. In comparison, a line tracing algorithm is tested which requires a small amount of user input for faster results.

Our method for power line detection assumes that the images have been transformed to remove perspective distortion. Then we apply our parabola detection algorithms to identify the power lines in the transformed image.

II. Method for power line detection

The four stages of our power line detection method are shown in Fig. 2. First, a perspective transformation is applied which distorts the curves into a catenary shape in the image. Second, the transformed image is filtered using edge detection techniques. Third, the power lines are detected in the filtered image using one of three algorithms: Hough transform, randomized Hough transform, or the line tracer. Finally, the detected power lines are transformed back to the original perspective.

III. Image preparation

There are two main steps required to prepare the images for the parabola detection algorithms.

A. Perspective transformation

This step transforms the image such that the power lines are aligned with the image plane of the camera. This perspective transformation is necessary in order to perform parabola detection on images taken from various angles such as the steep angle shown in Fig. 1. The quadratic equation

$$y = c(x-a)^2 + b \tag{1}$$

was used as an approximation of the natural catenary equation of the power lines.

The perspective transformation step requires some previously acquired information: the distance to the poles, the pixel locations of the poles in the images, the cameras intrinsic data, and the gravity vector from the accelerometer sensor in the smartphone. The pole locations are selected by the user, and the distances are acquired using a laser range finder.

The following steps describe the perspective transformation: Firstly, the pole locations are projected into 3-D using the pole distances and the cameras intrinsic data. Secondly, a scaled gravity vector is added to these 3-D points, and the new 3-D points are projected back to 2-D. Lastly, a perspective warp is applied to the image to transform the four 2-D points from a distorted quad into a rectangular shape.

Fig. 2 (b) demonstrates the effects of this transformation, which we achieve through features of LineSmarts.

B. Image filtering

Some image filtering is required before the Hough transform or line tracing algorithms can be used to detect power lines. All three of the algorithms tested require a binary input image where edge pixels are black, and non-edge pixels are white.

The image filtering used consisted of the following steps:

1) Gaussian blur
2) Gradient filter
3) Denoising based on non-local means [9]
4) Binary threshold
5) Edge thinning

This method is similar to that used in [3]. The main differences are in the Gaussian blur, denoising, and edge thinning.

The Gaussian blur is used with a 5×3 kernel, with a smaller σ value in the Y-dimension. The result is a stronger blur across vertical edges, while maintaining sharp gradients above and below the power lines.

The denoising step using non-local means removes more noise before the final thresholding step. This step is especially useful in reducing noise from objects such as trees which are a common obstruction in our power line images.

Edge thinning is employed to improve the efficiency of the Hough transform approaches, and simplify the peak selection stage. Power lines with a thickness greater than 1 pixel often produce multiple similar outputs which are difficult to distinguish and extract reliably.

Fig. 2 (c) shows an example of the filtering output.

IV. Hough transform for curve detection

The Hough transform [10] is a commonly used method for detecting curves in images. The first step for applying a Hough transform to an image is usually a type of edge detection filtering that produces a binary image. The Hough transform algorithm for detecting parabolas defined by Equation (1) is shown in Algorithm 1.

```
procedure HT(image matrix I)
    allocate a 3-D accumulator with i × j × k cells
    to represent a, b, and c parameter values
    for edge pixel P in I do
        compute all parabolas that pass through P
        for each candidate parabola C do
            increment accumulator cell for C
        end for
    end for
end procedure
```

Algorithm 1: Algorithm for computing a Hough accumulator for detecting parabolic curves.

The result is a 3-dimensional accumulator with $i \times j \times k$ cells, where cells with many votes represent the equations of the dominant parabolas in the input image. Searching this accumulator for local maxima is a simple but expensive way of finding the parabolas in the image.

A. Randomized Hough transform

The randomized Hough transform [11] is a common variant of the Hough transform that takes a randomized rather than a brute-force approach for sampling the edge pixels. This allows it to have more control over the computation time, and makes it more robust to noise in the image.

The randomized Hough transform algorithm also works to solve the problem of searching for curves in the accumulator. The image is sampled randomly for n iterations, and votes for the best matching parabolas are accumulated based on a similarity threshold t. Next, the parabola with the most votes is saved and its associated pixels are cleared from the image, then the process repeats.

This algorithm eliminates the requirement of a peak selection step, and as long as n is large enough, the image will be sampled fairly and the correct parabolas will be output. Algorithm 2 shows the process for detecting parabolas using a randomized Hough transform.

```
procedure RHT(image matrix I)
    for e epochs do
        for n iterations do
            select 3 random pixels from I and fit a parabola
            if accumulator contains a similar parabola then
                increment the vote count for that cell
                set the cell to the average parabola
            else
                create a new cell with the new equation
                set its vote count to 1
            end if
        end for
        output the accumulator cell with the best score
        zero pixels in I associated with the best parabola
        reset the accumulator
    end for
end procedure
```

Algorithm 2: Finding parabolic curves with the randomized Hough Transform.

B. Limitations of Hough transform techniques

When using a Hough transform to detect curves, the final stage involves searching the accumulator for peaks or clusters that represent the dominant curves in the image. An important drawback of Hough transform techniques is the grouping of similar curves. Choosing the parameters for the standard Hough transform involves fairly descretizing the parameter space into $i \times j \times k$ dimensions. If i, j and k are too small, two similar curves in an image may be grouped together, and stored as one large peak in the accumulator.

Another challenge with Hough transform is the false detection of curves. Fig. 3 demonstrates an example of this. The input image contains two parabolas, which will produce two large peaks in the Hough space. However, a third peak will also be present which corresponds to the parabola drawn in red. This third peak can overshadow any smaller parabolas that may be present in the image, and cause them to go undetected.

Another challenge lies in selecting appropriate accumulator dimensions and parameter ranges to achieve the required precision. For our standard Hough transform experiments, these values were selected empirically based on the observed usual curvature of power lines.

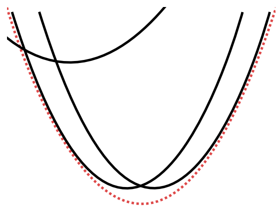

Fig. 3. An example of the types of curves that can be falsely identified by the standard Hough transform. The real curve at the top-left will produce a smaller peak than the spurious curve shown in red.

Slow computation time is another drawback of Hough transform techniques. Since our curve equation has three parameters a, b and c, the process of computing all possible parabolas through a point is much more computationally expensive than the equivalent algorithm for straight lines. Furthermore, finding local maxima in a 3-dimensional accumulator is also costly, especially if our accumulator dimensions i, j and k are large. For these reasons we only consider the randomized Hough transform in our evaluation.

The randomized Hough transform can also suffer from these drawbacks, based mainly on the values set for t and n. Large values for n will improve sampling fairness at the cost of more computation, and large values of t can cause more peaks to group together but will keep the accumulator small.

V. Line tracing algorithm

For our application, a high performance solution is required. For this reason, the standard Hough transform algorithm was not appropriate. Even with relatively small values of i, j and k, execution can take minutes to generate an accumulator. Even then, the results were not accurate with our peak selection algorithm.

We also propose an alternative method that requires an extra amount of user input but provides much more accurate, and faster results than the standard Hough transform. Based on an initial region indicated by the user, we use a line tracing algorithm similar to [5], but with some modifications. Wan *et al.* produce lines one pixel at a time, using three directional gradient values to determine the line direction. Our method builds a parabolic equation at each iteration, and projects short line segments of a predetermined width, W, to build a curve.

The difficulty of both of these line tracing algorithms is determining an initial point for the line tracer to start from. Wan *et al.* use prior knowledge about the 3-D coordinates of the poles and the focal length of the camera to determine these. For our application, we receive these starting points as input from the user.

The line tracing algorithm requires that the user swipes over a group of power lines. Since this algorithm is designed to be used on smartphones, we decided that a swiping gesture could provide the required starting pixels with only a small amount of extra effort from the user. This swipe gesture is combined with the filtered image to provide a number of starting pixels, from which the power lines are traced outwards toward the poles in short segments. Each time a segment is added, the power line equation is recomputed to improve the accuracy of the following segment predictions. Algorithm 3 describes the line tracing algorithm in detail, and Fig. 4 demonstrates one iteration of the line tracing algorithm.

```
1:  procedure LINETRACE(image matrix I, Starting pixel P)
2:      let CL = CR = P
3:      define a default parabolic curve C
4:      use C to generate points L and R,
5:      −W and W pixels away from P in x-axis
6:      while CL and CR are within the image do
7:          for h in range −H to H do
8:              generate line from CL to L with y-offset h
9:              generate line from CR to R with y-offset h
10:             compute scores for these segments based on
11:     number of pixels that match I
12:         end for
13:         break if no score is good enough
14:         save the segments with the best score
15:         set CL and CR to the ends of the best segments
16:         approximate new parabola C through CL, P, CR
17:         update L and R as in line 4
18:     end while
19: end procedure
```

Algorithm 3: Tracing parabolic curves in a binary image from a starting pixel. Parameters W and H are predefined, and represent the line segment width and search height respectively.

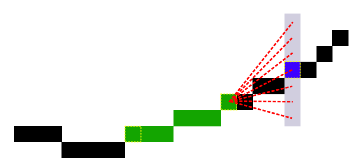

Fig. 4. One iteration of the line tracing algorithm. The green pixels represent pixels traced in previous iterations. The red lines indicate the candidate segments. The blue pixel is the predicted end-point.

VI. Experimental results

Our implementations of three parabola detection algorithms were tested on two image sets, with 17 images total. The first set contains synthetic images of between 4 and 5 complete parabolas with no noise. The second set contains real images of power lines taken from various angles, with between 3 and 8 visible conductors, and mostly clear sky backgrounds behind the conductors. Each visible conductor was hand labelled to establish the ground-truth pixels in the image. This ground-truth data was used as a basis for measuring the detection accuracy of our algorithms.

A. Possibility of power line detection

The first experiment was to test the possibility to detect power lines using Hough transform, randomized Hough transform, and line tracing. Large parameters were set for both

TABLE I
SYNTHETIC AND REAL IMAGES

Synthetic images

Algorithm	Detection rate (%)	Detection accuracy (px)	Time (s)
HT	9.30	4.12	1033
RHT	88.37	3.88	416.5
LT	90.70	0.96	1.11

Real images

Algorithm	Detection rate (%)	Detection accuracy (px)	Time (s)
HT	17.14	9.86	1554
RHT	94.29	2.07	652.6
LT	85.71	3.11	5.55

TABLE II
SYNTHETIC AND REAL IMAGES, FEWER ITERATIONS

Synthetic images

Algorithm	Detection rate (%)	Detection accuracy (px)	Time (s)
RHT	88.37	2.87	10.73
LT	90.70	0.96	1.11

Real images

Algorithm	Detection rate (%)	Detection accuracy (px)	Time (s)
RHT	88.57	2.91	13.08
LT	85.71	3.11	5.55

Hough transform algorithms to observe the upper limits of detection accuracy. The three algorithms were run with each of our test images, and the detection results were compared to the ground-truth data. The input images were taken from a smartphone camera with a resolution of 5312×2988. This resolution defines the scale of our accuracy metric.

In our results, we define detection accuracy to be the average pixel deviation of the detected line from the best matching ground-truth line. Detection rate is the percentage of power lines that were detected with an accuracy of less than 10 pixels. Lines with larger pixel deviations were discarded, and the line is treated as unidentified. Execution time is the average time taken per image, excluding user input time for the line tracer. The results of our first experiment are shown in Table I.

The results show that the standard Hough transform has a low detection rate, poor detection accuracy, and a slow execution time. The randomized Hough transform algorithm performed much better with almost all power lines detected, higher accuracy, but still with a slow execution time. The line tracer performed much faster than both Hough transform approaches, with good detection rate and accuracy. Fig. 5 (b) is an example of a power line detected with high accuracy using the line tracer.

B. Constraints on execution time

Our main aim is to automatically detect power lines in images in order to make measurements of conductor properties from smartphone images. For our application, the detection algorithm must perform faster than a user manually identifying the wires. Our line tracer takes approximately 1 second to execute per line in the image, which we consider reasonably fast for a convenient user experience.

For the next experiment, we reduced the iteration count for the randomized Hough transform such that its execution time is comparable to the line tracer, while maintaining consistent results over multiple runs. The results are shown in Table II.

The results from this experiment suggest that the randomized Hough transform with a lower iteration count can detect power lines with a similar rate and accuracy as the line tracer, but at approximately four times the execution time.

C. Discussion

Fig. 5 outlines some interesting cases where we can compare the line tracer to the randomized Hough transform.

The first row shows a case where both algorithms detected 7 out of the 8 visible power lines with good accuracy. However, the undetected power line is different in both images. The randomized Hough transform failed to detect a very thin wire. This wire was so thin that only approximately half of it was visible after the image was filtered, so the additional noise in the image prevented this from being detected. The line tracer was able to detect this since most of the visible pixels were traced with good precision and an accurate curve was fit to fill in the missing pixels.

The second row demonstrates a case with large amounts of perspective distortion on the power lines, caused by a steep photographed angle. Both algorithms managed to detect the lines with good precision, but the line tracer fails on one of the wires as the predicted segments jumped to an adjacent wire, producing an incorrect result. The randomized Hough transform is much less likely to output non-existent curves such as this one.

The third row is an example where the randomized Hough transform beats the line tracer significantly. The noise from the tree in the background distracted the line tracer and as a result, only 1 of 5 wires were detected accurately. However, the randomized Hough transform manages to detect 4 out of 5 wires accurately.

VII. CONCLUSION

In this paper, three different algorithms were tested on their ability to detect power lines in mid-range terrestrial photographs. This type of power line detection is useful for safe inspection and measurement of power lines. Most of the previous work in power line detection focused on straight line detection techniques which cannot be applied to images of power lines with large amounts of curvature.

The results show that Hough transform and line tracing are both good solutions for detecting power lines in transformed images, but with different strengths and weaknesses.

The standard Hough transform approach produces many challenges such as the detection of spurious curves, slow execution time, and difficulty of searching for peaks in the Hough space. Randomized Hough transform improves upon

Fig. 5. Comparing the line tracer to the randomized Hough transform in three cases. From left to right: input image, line tracer output, randomized Hough transform output.

this, and eliminates the peak searching step altogether, improving performance and accuracy drastically.

Though a large improvement, the randomized Hough transform can still perform slowly. Line tracing algorithms can provide a high performance solution to power line detection, but require extra user input to select the starting points.

Our solutions have shown that detecting power lines in ground-based images is feasible. The line tracer provides fast detection, whereas the randomized Hough transform provides accurate and automatic, but slower detection. Both solutions can be useful depending on the user's requirements. If responsiveness is a priority, the line tracer should be implemented. If high accuracy is required, the randomized Hough transform could be implemented as a background process.

There are plenty of paths for future work in this area. For example, a better filtering algorithm could improve results on noisy images. Also, more work could be done optimizing the line tracing algorithm, which often performs poorly on images where power lines cross, or parts of the line are not visible.

Acknowledgment

The authors would like to thank the University of Otago for supporting this project through the Priming Partnerships fund.

References

[1] "Home | LineSmarts." LineSmarts Ltd. 2016 [Online]. Available: http://www.linesmarts.com/. [Accessed: 12-Sep-2016].

[2] G. Yan, C. Li, G. Zhou, W. Zhang, and X. Li, "Automatic Extraction of Power Lines From Aerial Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, pp. 387–391, July 2007.

[3] J. Zhang, L. Liu, B. Wang, X. Chen, Q. Wang, and T. Zheng, "High Speed Automatic Power Line Detection and Tracking for a UAV-Based Inspection," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, pp. 266–269, Aug. 2012.

[4] Y. Chen, Y. Li, H. Zhang, L. Tong, Y. Cao, and Z. Xue, "Automatic power line extraction from high resolution remote sensing imagery based on an improved Radon transform," *Pattern Recognition*, vol. 49, pp. 174–186, Jan. 2016.

[5] X. Wan, X. Qu, L. Wang, B. Wu, J. Zhang, and S. Zheng, "Photogrammetric techniques for power line ranging," *Pattern Recognition Letters*, 2010.

[6] Z. Yue, A. Goshtasby, and L. V. Ackerman, "Automatic detection of rib borders in chest radiographs," *IEEE Transactions on Medical Imaging*, vol. 14, pp. 525–536, Sept. 1995.

[7] T.-H. Min and R.-H. Park, "Eyelid and eyelash detection method in the normalized iris image using the parabolic Hough model and Otsus thresholding method," *Pattern Recognition Letters*, vol. 30, pp. 1138–1143, Sept. 2009.

[8] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun. ACM*, vol. 15, pp. 11–15, Jan. 1972.

[9] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65 vol. 2, June 2005.

[10] P. V. C. Hough, "Method and means for recognizing complex patterns." US Patent Number 3069654, Dec. 1962.

[11] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, pp. 331–338, May 1990.