Conversion of GIS Contour Maps into Surface Digital Elevation Models for Robotic Surveying

Henry (Heqing) Mei, Lonnie T. Parker, Ayanna M. Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
henry.mei@gatech.edu, lonnie@gatech.edu, ayanna@ece.gatech.edu

***Abstract*—With the advent of new technologies, robotic surveying systems are being developed to facilitate the collection of ground-based information to validate and complement data collected by traditional and satellite-based instruments. The development of such systems necessitates an accurate set of reference data. Given the limitations of current *in situ* measurement methods to aid remote sensing, this paper outlines a method for the creation of three-dimensional data from the most common public data source, 2D contour maps. Using image processing and interpolation techniques, this method was first tested against data collected by a robotic survey system and against methods that a human expert would use. Comparatively, our method yielded vertical RMSE in the range of (0.006066 - 0.39) [m] for different horizontal spatial resolutions. Twenty additional sample contour maps were identified to further vet our method against that of a human expert as a function of the 3D interpolation method selected. These tests provided errors in the centimeter range and also revealed that the linear triangular mesh interpolator is the best choice for this type of image input data.**

***Index Terms*—Digital elevation models (DEM), geodetic survey stations (GSS), geographic information systems (GIS)**

I. Introduction

Geological changes magnified by global warming have renewed scientific interest in effective surveying techniques. With the technological advancement in automation and distributed systems, robotic surveying systems are an increasingly attractive option for accurate, cost-effective data collection in the field. Currently, existing data from remote sensing measurements may be unavailable or too costly to obtain. Furthermore, methods for *in situ* measurements, such as geodetic survey stations (GSS), are both expensive to maintain and often dangerous to implement. Robust mobile robotic platforms, however, offer the flexibility in coverage while potentially lowering the cost associated with providing such validation equipment. In order to facilitate the development of these distributed surveying systems, accurate reference data must be made available to analyze the effectiveness of new robotic surveying techniques in their sampling schemes. As such, this paper presents a method for converting publicly available geographic information system (GIS) contour maps into three-dimensional surface digital elevation models (DEM) at multiple resolutions, regardless of the shape and arrangement of contours. The conversion process primarily involves contour registration and noise removal processes in computer vision.

II. Background & Theory

Non-deterministic features of terrains are often unobtainable through remote sensing at small spatial scales ($< 100\ m^2$) [1]. This area is usually representative of the pixel footprint demonstrating the spatial limitations of a given remote sensing instrument. These unique features are also costly to obtain through human land surveying and require a detailed amount of effort to plan and execute, especially within harsh weather conditions like the Arctic [2, 3]; however, multi-agent robotic systems may be able to scalably traverse terrain at the required precision and gather accurate survey measurements with high autonomy. The cost of distributed robotic surveying systems is also scalable, with the cost of equipment replacement significantly lower than that of existing *in situ* measurement techniques, such as automatic weather stations (AWS) [4] or geodetic survey stations (GSS). Considering such problems, multi-agent systems are ideal for locations where the loss of equipment or the endangerment of human life is a significant concern.

Unfortunately, the availability of accurate geographic measurements, such as those made by satellites using L-band synthetic aperture radar (SAR) or lidar, varies from site to site. If obtainable at all, these measurements are usually sparse approximations in the form of 2D contour maps. Thus, our work makes use of this 2D representation to produce more detail than was previously available to the public. By leveraging the simplicity of the information conveyed in the form of a two-toned representation along with advances in image processing, we are able to segment the relevant elevation-change data to assist in confirming proper navigation techniques for the mobile measurement system.

Past work in this area utilizes highly accurate satellite data to generate contour maps; however, very little work has been done to reverse this process. For example, because survey-quality satellite images taken by the U.S. government are considered proprietary information, national and local park services only provide converted two-dimensional topographic maps to the public.

III. Methodology

The conversion process from topographic map to digital elevation model (DEM) is essentially an edge detection problem. The topographic map contains contour lines that must be identified, assigned the correct height, and passed through an interpolation algorithm. The map images from this publicly available data often have poor resolution and thick contour lines, which require additional image processing and noise reduction (see Figure 1). We focus our efforts on data made available from a 2006 lidar survey conducted over the city of Atlanta. Our specific areas of interest are a section of a

Fig. 1: Sample contour image.

local park surveyed by the SECT-II robot from Bluebotics Inc. and 20 other surrounding areas identified with similar traits. These contour data can be found online (http://gis.atlantaga.gov). Techniques such as difference filtering, pixel intensity searching, and Roberts' Cross/Prewitt/Sobel/Canny filters were applied experimentally on our selected map segments of interest. More modern methods such as wavelet transforms were initially ruled out because of complexities in scaling for multiple image conditions.

Simple cascading of two one-dimensional difference filters and pixel intensity searching erroneously identified significant amounts of noise. Similarly, sum of square difference methods such as Roberts' Cross [5] also suffer from high sensitivity to noise. Gradient masking methods like the Prewitt/Sobbel method [6, 7] only analyze the vertical and horizontal directions and do not have perfect rotational symmetry; they were not used. In the end, portions of the Canny edge detection algorithm [8] were chosen, reducing the complexity of edge detection by introducing Gaussian noise reduction within the algorithm (1).

$$\frac{\partial^2}{\partial n^2}(\mathbf{g} * \mathbf{I}) = \frac{\partial}{\partial n}\left(\frac{\partial \mathbf{g}}{\partial n} * \mathbf{I}\right) \tag{1}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x + \mathbf{G}_y}$$
$$\mathbf{\Theta} = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Where $\mathbf{g}$ is the Gaussian noise-reduction filter, $\mathbf{I}$ is the image, $\mathbf{G}_x$ and $\mathbf{G}_y$ are x and y components of the edge gradient of the filtered image, respectively, and $\mathbf{\Theta}$ is the edge direction angle.

Because of its complexity, the conversion process is segmented into multiple stages. These include image processing, elevation (or contour height) assignment, and data interpolation (see Figure 2). Extraneous data are filtered from the raw contour map; the map then undergoes noise removal and registration.

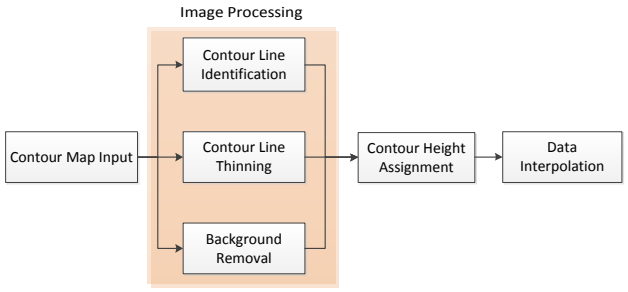

Fig. 2: Digital elevation model (DEM) conversion process.

A. *Pre-processing and Contour Line Identification*

The initial contour map segment was converted to grayscale to simplify the contour identification process. Like most previous work in the area of translating image data from one format to another, the source data contains noise. Fortunately, the noise in the given data could be easily suppressed using Gaussian smoothing. We sought to find an optimal method without *a priori* contour line knowledge, so the only usable information was pixel intensity. Therefore, edge detection algorithms were used for contour line identification. The subsequent contour line thinning method would vary by the edge detection method chosen; however, the Canny edge detection algorithm was selected because it also includes diagonal gradients, offering an advantage over previous methods. Contour line thinning was achieved through iterative subtraction of the identified edges. After each iteration, pixels were assigned intensities of 0 (black) or 255 (white), based on an experimental and adaptively determined threshold value, α, that set a cutoff point at $255 \cdot \alpha$. Remaining noise erroneously identified as contour pixels was removed by a blob filter through connectivity analysis. As general connectivity analysis fails for edge pixels, a one-pixel border on the image was removed. A brief summary of this process is outlined in pseudocode (see Algorithm 1).

Algorithm 1 Noise Reduction and Contour Identification

Require: input image X, number of iterations $N (N \geq 1)$, and threshold value α $(0 < \alpha \leq 1)$
  $X \Leftarrow$ grayscale(X)
  for $i = 1$ to N **do**
    $C \Leftarrow$ Canny(X)
    $X \Leftarrow X - C$
    if $X_{j,k} <$ Max$(X) \cdot \alpha$ **then**
      $X_{j,k} \Leftarrow 0$
    else
      $X_{j,k} \Leftarrow 255$
    end if
    $X \Leftarrow$ BlobRemove(X)
  end for
  $X_{labeled} \Leftarrow$ Logical(X)
  $X_{labeled} \Leftarrow$ ConnCompLabelAlg(X)
  return $\{X, X_{labeled}\}$

In order to show the effectiveness of the pre-processing and

line thinning, a test segment was used to confirm the proper operation of the algorithm before comparing its results with the *in situ* data (see Figure 3). In running the sample data, once the image was converted to grayscale, edge detection was applied. After smoothing the image with a Gaussian filter, the Canny edge detection algorithm (1) caculated zero crossings of the derivative of that Gaussian, and then identified the edge gradient magnitudes and angles (rounded to nearest 45 degree multiple) at those crossings. The area being converted had relatively low pixel intensity difference, high granularity, and square pixelation (see Figure 3a). Nonetheless, the initial Gaussian filter of the Canny algorithm smoothed most of the granularity and pixelation; the ensuing computation identified the edges, subtracted the edge pixels from the original image, and applied a cutoff filter. The results of a single iteration show that this process seems to be effective in noise reduction; thinning of the contour lines is less apparent (Figure 3b), yet iterative refinement did yield the desired result in few passes (Figure 3c).

After the image passed through the first stage, the intensity array representing the image was converted to a logical array, with ones representing the contour pixels and zeros marking background pixels. The subsequent application of the connected component labeling algorithm (CCLA) produced the correct identification of the contour lines shown without any local noise. Figure 3d is a smaller subsection of the contour line in Figure 3a.

B. Elevation Assignment

Few viable methods were discovered for elevation assignments to contour lines. Initially, a recursive floodfill algorithm was adapted to only traverse individual contour lines and assign the corresponding heights. However, large map sections proved this approach too computationally cumbersome for practical use; the algorithm yields a high overhead of recursion on large pixel grids ($\sim$1500 by $\sim$1500).

Ultimately, elevation assignment was divided into three stages. First, the map section was converted into a logical array (with 1's marking contour line pixels) after thinning. Then, the connected component labeling algorithm was used to label each set of contour line pixels with an integer value–a contour identification number; four-connectivity was used to exclude diagonal fringe pixels and any residual noise pixels. Last, elevation values were assigned by replacing each set of contour line pixels, identified by their contour identification number, with their corresponding elevations. This operation was accomplished through replacement by linear traversal of the pixel array representing the map section.

C. Data Interpolation

Multiple data interpolation algorithms were available [9], including both regular and irregular methods: inverse distance weighted (IDW), Kriging, and Delaunay triangulation; however, only deterministic algorithms were utilized to limit the scope of our preliminary work. Inverse distance weighted interpolation is simple to implement, but it is not ideal for dealing with discontinuities. On the opposite end, Kriging interpolation utilizes a stochastic method and is the most rigorous interpolation method. However, this method is sensitive to model specifications, requiring a unique variogram to represent the spatial correlation among available data points. Meanwhile, triangulation methods are easier to implement and work well when control points are critical points. Used with steepest descent search patterns, triangulation methods potentially proved to be the most accurate. Within such methods, the linear Delaunay triangular mesh interpolator produced the best error (relative and absolute) when compared to the others considered, such as linear/cubic triangular mesh and natural/nearest neighbor [10]. Non-contour boundary elevation values were also included to contribute in generating a complete data set. Additionally, Delaunay triangulation methods handle discontinuities well at the expense of angular irregularities in the interpolated data. Preference was given to the interpolation method that could be applied universally and automatically with limited data, especially for this particular terrain type where slowly varying undulation is observed. In the end, triangulation methods were chosen over the others.

Linear, cubic spline, and nearest neighbor methods were implemented for comparison. In addition, MATLAB's v4 method was included to compare the results of local polynomial interpolation methods to global Greens' function/radial basis techniques. During analysis, it is important to note that radial basis function methods generally consume large amounts of memory and, therefore, may not be well-suited for mobile, real-time applications. The v4 method is no exception. Because of the widespread use and relative simplicity of the methods compared, the mathematics of each particular method is not described here.

D. RMSE

Based on photogrammetry and standards for mapping [9], the root mean squared error, $\hat{\theta}$, as expressed in (2), is used as the metric to evaluate the performance of our work.

$$RMSE = \hat{\theta} = \sqrt{E[(Z_0 - \hat{Z}_0)^2]} \tag{2}$$

Z_0 and $\hat{Z}_0$ represent our ground truth data and estimated data produced from the various interpolation schemes selected to regenerate our terrains. While other acceptable alternatives, such as mean absolute error, average standardized error, and mean bias error exist, RMSE is the most suitable for quantifying the size of error attained for non-clustered data [9].

IV. Results

A. Benchmarking and Error Analysis

To quantify the success of the conversion process over a larger space, our reference map was used as input. Comparable 3D maps were generated from the 2D contour map in Figure 5 through data collected by both 1) a human expert (who identified contours by hand) and 2) a robotic survey conducted in April 2010 (see Figures 4 and 5). The surveyor robot executed a traditional parallel transect (or "lawnmower") pattern while

(a) Smaller test contour map segment (b) Result of single pass thinning at 0.8 threshold (c) Result of 4-pass thinning at 0.8 threshold (d) Labeled contour after applying connected component labeling algorithm

Fig. 3: Canny edge detection applied to test segment

gathering pitch and roll information. The collected data were then translated into a 3D map for this analysis. We evaluated the error between our algorithm and each of the two alternative map-generating options as presented in Tables I – III at three different horizontal scales of resolution (see Figure 6).

Fig. 4: Test area of local park used for field trials.

Fig. 5: Contour map of test area.

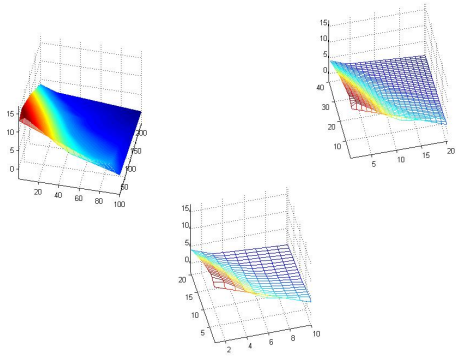

Fig. 6: 3D-generated maps at various resolutions

Fig. 7: SECT II (from Bluebotics, Inc.), rugged-terrain platform used for field survey.

In general, error increases with decreasing sample resolutions as detail is lost in the downscaling of the data; however, this demonstrates the importance of gaining access to data sets at multiple resolutions and the possibility of having greater accuracy with higher spatial resolution. Oddly, when comparing the human expert to our algorithm, error decreases when the sample resolution decreases, a trend noticed in later results.

The error data obtained by comparing the SECT-II data, human expert, and the models created by our method provide a strong case for the accuracy of the contour identification and thinning system. The human expert accurately traced the contour lines by hand in each test case, while our method automated this process. Both sets of pre-processed data were then interpolated using the same interpolation method. The discrepancies between the methods are mostly attributed to the lack of line thinning by the human expert, causing smoother interpolation data to be generated ($i.e.$ data without detail), and also the lack of human vision resolution at the pixel level. Obviously, the re-processed image of the human expert does not contain noise, and quantitative comparison of our pre-processed image with that of the human expert showed only differences in contour thickness.

Analyzing the error data of the SECT-II data against the algorithm-generated data, the vertical accuracy of results at higher horizontal resolutions ($i.e.$ 0.2 [m] and 1 [m]) is in accordance with accuracy standards for Class 1, Class 2, and Class 3 maps; the error produced by the algorithm satisfies all levels of accuracy standards used ([11]). Assuming the input data has minimal error, the data suggest that the algorithm can be implemented in existing systems without significant loss of accuracy or precision.

TABLE I: Error between Human expert-generated map vs. SECT-II Map.

Resolution	Relative (%)	Absolute ([m])	RMSE ([m])
0.2 [m]	20.175	0.184	0.234
1 [m]	23.983	0.210	0.280
2 [m]	33.237	0.311	0.390

TABLE II: Error between Algorithm-generated map vs. SECT-II Map.

Resolution	Relative (%)	Absolute ([m])	RMSE ([m])
0.2 [m]	20.256	0.185	0.234
1 [m]	24.102	0.210	0.279
2 [m]	33.234	0.310	0.388

Further testing was performed with 20 terrain samples extracted from the online reference site, mentioned earlier in Section III, to compare the effectiveness of the selected interpolation methods (see Table IV) and to examine the peculiarity of direct relationship between error and sample resolution when comparing the human expert to the algorithm. In running the tests, we note that despite noise removal, our process is still sensitive to large, high-intensity blobs.This noise is easily misinterpreted as small contour segments. Fortunately, the calculated error between the human expert and our process is exceedingly low, with the linear interpolator and MATLAB's v4 method generating minimal error when compared with the cubic and nearest neighbor interpolators. Generally, among the chosen interpolation methods, cubic spline produces lower error; however, because most terrain samples are smooth and have relatively low gradients, the linear method and the Greens' function/radial basis function method used by the v4 algorithm better approximate the trajectories of subsequent contour pixels and produce lower error. The terrain is generally Lipschitz continuous in two dimensions [12], contributing to the low error because of the boundedness of the terrain gradient. If the variability of the terrain were significantly higher ($i.e.$ resembling greater undulating features), the cubic spline method would most likely outperform the linear and v4 methods.

As mentioned earlier, the error decreases with decreased horizontal resolution for three of the four methods tested contrary to the SECT-II comparisons. Because the human expert and the original sample differ by the lack of noise and line thickness, a decrease in resolution led to a corresponding increase in similarity between the two downscaled samples. Thus, the resulting interpolation error decreases overall.

Comparing the change in error calculations from 1 [m] resolution to 10 [m] resolution, the average overall error decreased by 751 [mm] (see Table IV), while the nearest neighbor method is the only interpolator where an inconclusive change in error occured. We attribute this behavior to the lack of "good" neighbours as the method relies heavily on reference

TABLE III: Error between Human expert-generated map vs. Algorithm-generated map.

Resolution	Relative (%)	Absolute ([m])	RMSE ([m])
0.2 [m]	0.3802	4.023e-3	6.066e-3
1 [m]	0.3786	4.054e-3	5.761e-3
2 [m]	0.3824	4.084e-3	5.974e-3

TABLE IV: Average RMS vertical accuracy of 20 map samples at three horizontal resolutions of accuracy.

	RMSE [m]		
METHOD	1[m] Res.	5[m] Res.	10[m] Res.
Cubic	1.234e-2	1.108e-2	9.789e-3
Linear	5.670e-3	5.370e-3	4.835e-3
Nearest	4.401e-2	3.828e-2	4.046e-2
v4	4.780e-3	4.922e-3	5.116e-3

points close by, an assumption that cannot be met by our application. In reality, contour maps often contain statistically significant levels of error; therefore, higher resolutions do not necessarily indicate an increase in accuracy.

Looking at the boxplots (see Figures 8 – 10), v4 produces the lowest error, while linear is a close second. Both interpolation methods produce both accurate and precise interpolated data with few higher error outliers. We note that the cubic spline method also produces acceptable but worse results while the nearest neighbor method produces poor results when fewer data points are given, as seen by the high error outliers and wider interquartile range (IQR) on the plot.

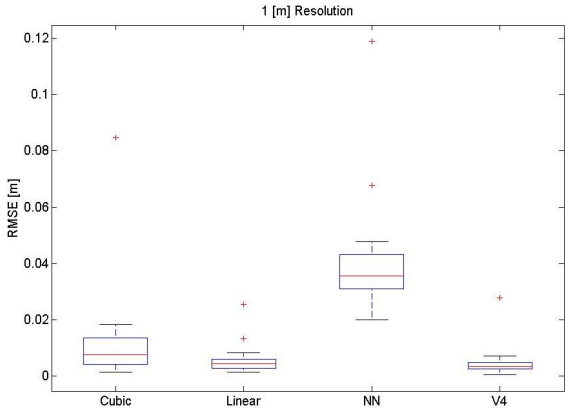

Fig. 8: Range of error at 1 [m] resolution.

As mentioned before, all of the interpolants scale directly with resolution. The nearest neighbor interpolator has the worst scaling as it interpolates using the nearest known data points. As resolution decreases, known data points decrease, causing an increase in the variability of error and the highest error outliers out of all the methods compared; however, the average RMSE error for all methods other than nearest neighbour decrease with a downscaling in spatial resolution.

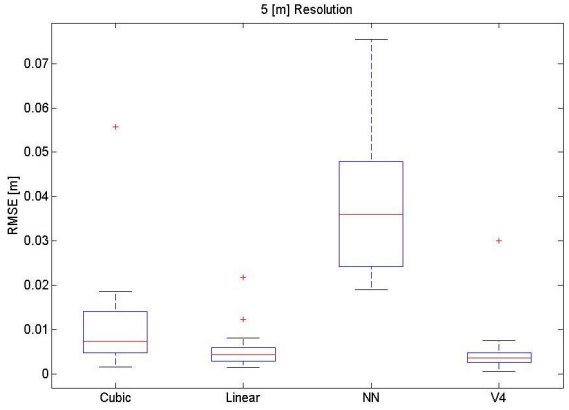

Fig. 9: Range of error at 5 [m] resolution.

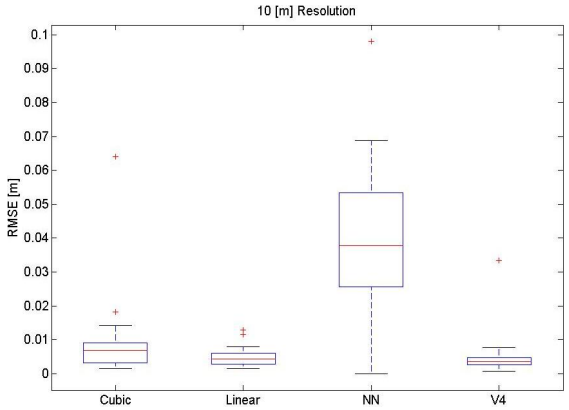

Fig. 10: Range of error at 10 [m] resolution.

V. Conclusion & Future Work

The results are promising; the linear triangular interpolator was chosen as the optimal method over the v4 Greens' function/radial basis interpolator because of "intrapolation" issues (primarily caused by large holes across the data, a likely possibility for geographic systems). Despite the optimal choice of interpolator, the conversion process requires continued iterations for improvement. Other algorithms, such as the Canny-Deriche algorithm, which utilizes an IIR filter and is computationally less complex for applications that require larger amounts of smoothing, will be considered for implementation. Also, more modern techniques using wavelets or the Minimal Path for KeyPoint Detection (MWKPD) algorithm may be investigated for usability in the future. More accurate results can be obtained when obstructive elements such as elevation labels placed directly on the map image do not conflict with integer-labeled contour numbers as these defects must be removed by hand. Furthermore, the process has not been adapted to multi-typed data structures such as cell arrays to prevent duplicate height assignments. Eventually, an automated contour identification process will be incorporated. Also, meshing large interpolation arrays is a significant task for consumer desktops, and an efficient way to select interpolation points has not been determined. In the end, the conversion process itself generally holds for all cases with moderate noise and without height-contour number overlap. Pursuant to the purpose of this work, feature recognition will be investigated to automate the differentiation of surveyed landmarks.

Acknowledgments

The authors wish to thank the City of Atlanta Department of Watershed as well as the Science, Mathematics and Research for Transformation Program (SMART) and the National Aeronautics and Space Administration (NASA) under the Earth Science and Technology Office, Applied Information Systems Technology Program for funding of this work.

References

[1] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf, "The shuttle radar topography mission," *American Geophysical Union*, 2007.

[2] S. Williams, L. T. Parker, and A. M. Howard, "Calibration and validation of earth-observing sensors using deployable surface-based sensor networks," in *IEEE Journal of Selected Topics in Earth Observations and Remote Sensing*, 2009.

[3] V. B. Spikes and G. S. Hamilton, "Glas calibration-validation sites established on the west antarctic ice sheet," in *International Symposium on Remote Sensing of Environment*, Honolulu, Hawaii, 2003.

[4] J. Stroeve, J. E. Box, F. Gao, S. Liang, A. Nolin, and C. Schaaf, "Accuracy assessment of the modis 16-day albedo product for snow: comparisons with greenland in situ measurements," *Remote Sensing of Environment*, vol. 94, pp. 46 – 60, 2005.

[5] L. G. Roberts, "Machine perception of three-dimensional solids," in *Lincoln Laboratory Technical Report 513*, 1963.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[7] W. K. Pratt, *Digital Image Processing*. Wiley, 1978.

[8] J. Canny, "A computational approach to edge detection," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, 1986, pp. 679–714.

[9] J. Li and A. D. Heap, *A Review of Spatial Interpolation Methods for Environmental Scientists*, Australian Government, 2008.

[10] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.

[11] Q. A. Abdullah, "Mapping matters: The laymans's perspective on tehcnical theory and practical applications of mapping and gis," *Journal of Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 6, pp. 683 – 685, June 2008.

[12] K. Hormann, S. Spinello, and P. Schrder, "C1-continuous terrain reconstruction from sparse contours." in *VMV'03*, 2003, pp. 289–297.