

# A Feature-based Classifier for Dragonflies and Damselflies

Kyle Ryan<sup>1</sup> and Carl Salvaggio<sup>2</sup>

<sup>1,2</sup>Rochester Institute of Technology  
54 Lomb Memorial Drive  
Chester F. Carlson Center for Imaging Science  
<sup>1</sup>kjr6491@rit.edu  
<sup>2</sup>salvaggio@cis.rit.edu

May 24, 2012

## Abstract

Unique patterns present in the wings of dragonflies and damselflies can be used to determine their family, genus, and species. A method for classifying dragonflies and damselflies using a particular pattern known as the triangle was developed using scanned images of the wings. Digital image processing techniques, such as image segmentation and feature detection, are used to determine properties of the triangle useful for classification. These properties are then compared against a triangle property database of known dragonflies and damselflies. A prototype implementing this method has been shown to demonstrate a high degree of accuracy.

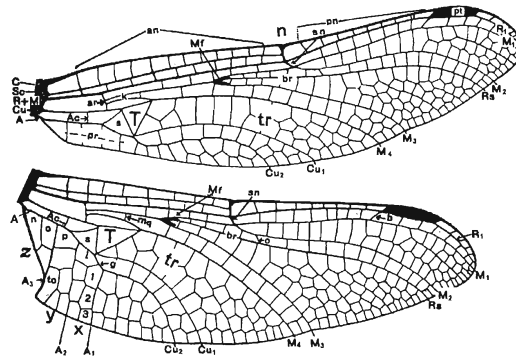


Figure 1: Diagram from Needham et al. [2] of dragonfly wing venation on both a front (top) and rear wing (bottom). The primary feature of interest, the triangle, is labeled with a T.

## 1 Introduction

Researchers in the field require a reliable and efficient method of classifying captured dragonflies and damselflies. It is known that the vascular patterns between different genera and species of dragonflies and damselflies are unique and of particular use in their identification; however, it can be difficult for even expert observers to compare these patterns. Therefore, an automated system using images of the wings is desired. Presently no

such system for the identification of dragonflies is known to the authors. There have, however, been similar systems developed for the identification of plants, such as LeafSnap, which uses leaf shape to identify trees [1].

Careful examination of the wings revealed that the triangle (T in Figure 1) is of particular use in classification [3]. This feature is easily identifiable and its properties vary between different species, but not individuals. Furthermore,

the description of only a few of its properties is enough to allow for reliable classification, namely area, aspect ratio, orientation, and number of enclosed cells.

The system was developed using a set of wing images obtained from previously collected and classified dragonflies. Due to the diverse nature of the sample set and dragonfly and damselfly wing structure, a robust image processing system is required. In order to ensure this requirement is met and to determine the triangle properties with a minimum level of user interaction, a multilevel image processing approach is taken using a variety of conventional techniques. These techniques include thresholding, segmentation, feature detection, and morphological operations.

Classification and feature reduction methods developed for remote sensing were used to perform the classification and analyze the feature set. While the aforementioned features are all good indicators of species or genera, it was not known whether some features may be redundant prior to system design, so further analysis using feature reduction techniques was performed.

## 2 Method

### 2.1 Scan Segmentation

Before analyses of the triangle can begin, it must be segmented from the image. In images used for the development of this system, all four wings are removed from the dragonfly and placed on a flatbed scanner along with a ruler for calibration, as in Figure 2(a). In order to extract each portion of the image and identify it as a wing or ruler, the image must first undergo segmentation. The image is binary thresholded using Otsu's method [4] and the complement is used to create the image shown in Figure 2(b). This way, the vascular pattern that is of interest is considered the foreground ("on" pixels) and the rest of the wing the background ("off" pixels). Next, the convex hull of each wing and the ruler is computed, resulting in Figure 2(c) [5]. Note that this completely fills each wing and most of

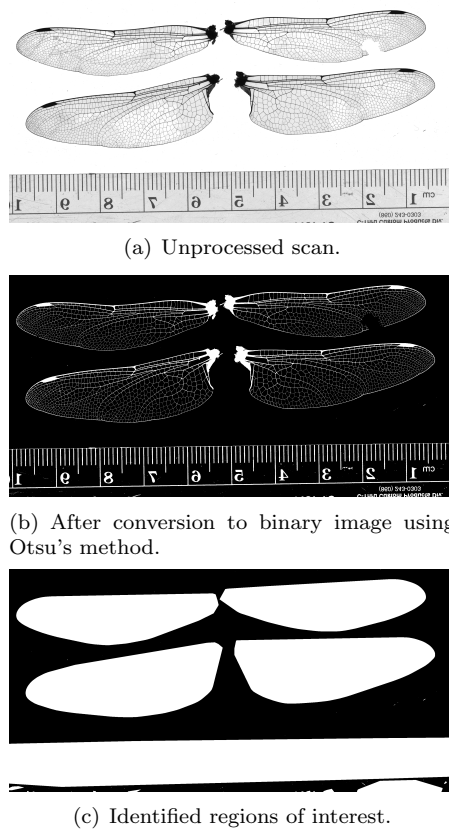


Figure 2: Scan at various stages of segmentation.

the ruler, ensuring that the area calculation for each region is accurate and unaffected by any missing portions of the wings.

Each region is then sorted by area, with the ruler assumed to be the largest, the two rear wings the second largest, and the two front wings the third largest components. The position of each region is used to identify each wing as a right or left-side wing. Each region is then used to crop the original scan into five separate images for display and analysis. The right side wings are flipped across the y-axis for consistency of display; however, because the wings are symmetrical this has no effect on the classification method employed.

Each segmented wing, before thresholding, is

presented to the user for manual identification of the triangle in each wing. This is performed using a polygonal lasso tool, allowing the user to completely enclose the triangle in a polygonal selection region using the mouse. At least one front triangle and one rear triangle must be selected, as features from both the front and rear wings are used. The ruler is used to calibrate each image, measuring the number of pixels present over a given unit distance. This ensures a consistent area calculation despite any changes in image resolution or magnification between different scans.

Calibration is performed automatically through application of the Hough transform [6]. Using the transform, the longest vertical lines in the ruler are identified. The shortest distance between the two closest lines is taken to be 1 cm. (This, of course, requires that only metric rulers be used, as it is not possible to determine the units of the ruler from the image alone.)

## 2.2 Registration

Each wing is rotated such that the vein across the top of the wing is parallel to the  $x$ -axis. This is done to ensure that triangle orientation is correctly calculated and consistently oriented for all samples, regardless of how the wings were placed on the scanner.

To determine the rotation angle, a matched filter is used to locate the thickest vein in the image, as this generally corresponds to the cross vein. The filter was designed such that its response was greatest when it was centered over the thickest portion of the wing, as shown in Figure 3(a). Areas in the binary image where the response of the filter was less than half of the maximum were then set to zero, as shown in Figure 3(b). A line was fit through the remaining pixels using least squares regression, as shown in Figure 4. The arctangent of the slope of this line gives the angle that the wing must be rotated by to make it parallel with the  $x$ -axis.

## 2.3 Feature Extraction

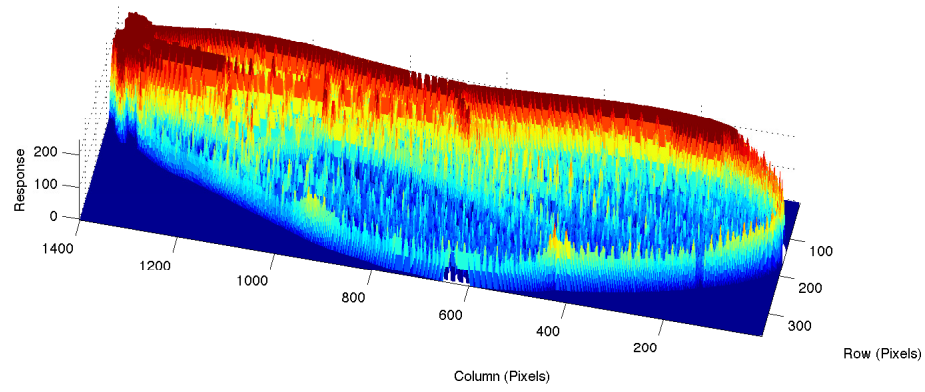
After each triangle is selected in Section 2.1, a binary mask in the shape of the selected region is generated, multiplied with the original wing scan, and cropped. This leaves only the triangle in Figure 5(a). The grayscale image is then converted to the binary image in Figure 5(b) (also using Otsu’s method). Any pixels that may have become disconnected from this process are then bridged. Bridging is performed by extending “on” pixels until they become connected to other “on” pixels. The border is then cleared to delete any regions outside of the triangle that could be falsely counted as cells. To clear the border, any “on” pixels that are connected to the border of the image, and in turn connected to other “on” pixels, are turned “off”, resulting in Figure 5(c). Note that this has no effect on the triangle cells as they are separated from the edge by a line of “off” pixels.

### 2.3.1 Cell Count

The binary image now consists of the individual closed regions in Figure 5(c), which can be identified using connected components [5]. Any “on” pixel connected to another “on” pixel in any direction will be identified as part of an individual cell. Because of this, and to prevent multiple closely spaced cells from being identified as one, the image is morphologically eroded [5] with a small disk shaped structuring element prior to further analyses. This removes pixels from around the cells, increasing their separation and removing any false connections left by the thresholding process.

### 2.3.2 Area

To calculate the area, the cleared binary image is morphologically closed using a large structuring element. The closing operation performs the opposite task of the erosion described in Section 2.3.1, and merges all of the cells into one while preserving the shape of the triangle (Figure 5(d)). The area is then equal to the number



(a) Matched Filter Response.



(b) Image after Thresholding

Figure 3: Matched filter response and image after thresholding. The remaining “on” pixels comprise the thickest lines in the wing.

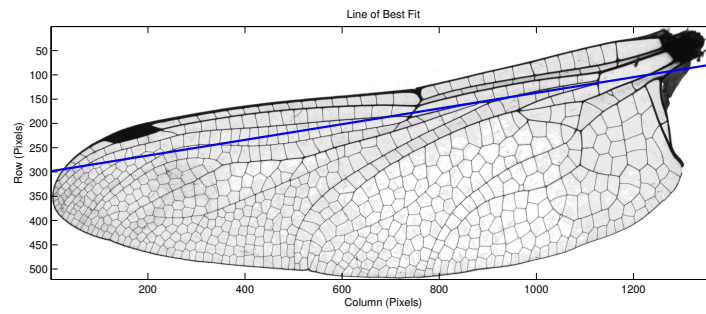


Figure 4: Line of best fit calculated using Figure 3(b) plotted over the original image.

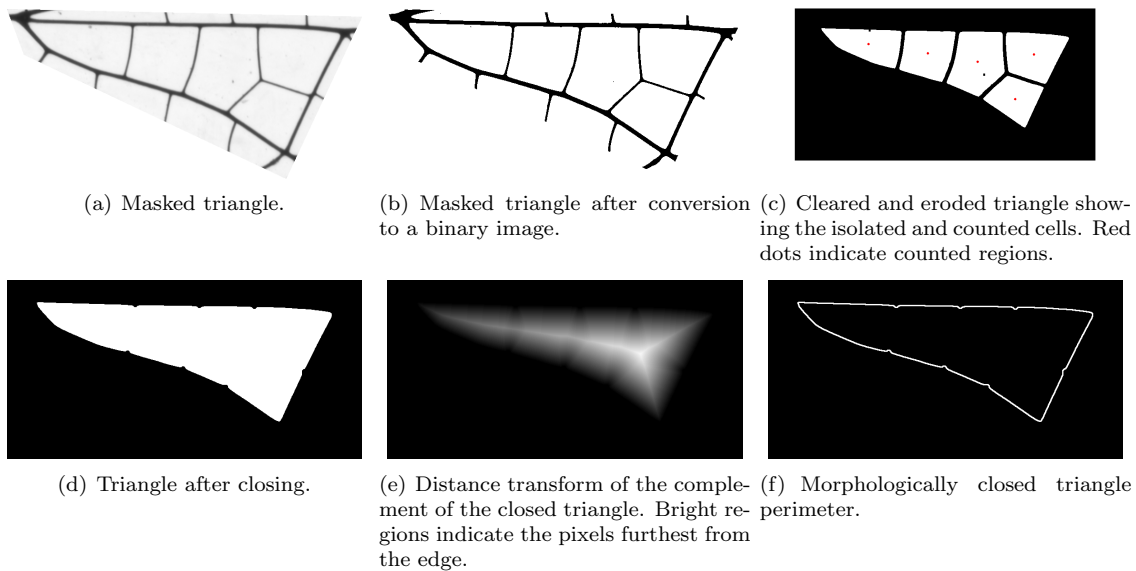


Figure 5: The extracted triangle at various stages of processing.

of “on” pixels divided by the number of pixels per unit squared area, as obtained by the calibration from Section 2.1.

### 2.3.3 Aspect Ratio

The same morphologically closed image in Figure 5(d), that is used to calculate the area, is also used to determine the triangle aspect ratio. The distance transform is used to locate the centroid of the triangle, which corresponds to the peak in Figure 5(e) [7]. Next, the furthest pixel along the perimeter of the triangle is located (Figure 5(f)). The furthest pixel from this point is then found, forming a line. Lastly, the furthest point along the perimeter from this line is found using the point-line distance formula in Equation 1 [8], where  $x_0$  and  $y_0$  are the coordinates of the point, and  $x_1$ ,  $x_2$ ,  $y_1$ , and  $y_2$  are coordinates of points on the line.

$$d = \frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad (1)$$

The resulting set of points in Figure 6 define all 3 corners of the triangle. The triangle’s width is the distance between the second and third point, and its height the distance from the line formed by the second and third point and the first point. Dividing the height by the base is the aspect ratio of the triangle.

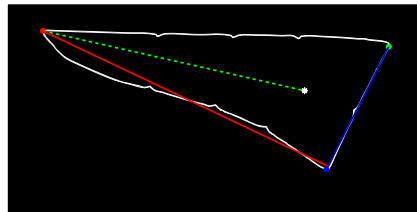


Figure 6: Image showing the identified corners of the triangle along with the lines used for distance calculations. Note that the centroid is also shown along with the line used to determine orientation (dashed green).

### 2.3.4 Orientation

Using the triangle centroid and furthest point from the centroid, the triangle's orientation is easily determined by calculating the arctangent of the slope of the line between these two points. This line is shown in Figure 6 (dashed green).

## 2.4 Classification

The system was trained using a set of previously identified dragonfly and damselfly wings. These wings were subjected to the same processing routine as outlined above, and the properties of all extracted features were stored in a database as 8 element vectors. The first 4 elements are the number of cells, area, aspect ratio, and orientation of the front triangle, the last 4 elements the number of cells, area, aspect ratio, and orientation of the rear triangle. Multiple individuals of the same species are used to create an average vector representing that particular species.

To classify an unknown wing, the angle between its 8 element feature vector and each average vector is calculated using equation 2, adapted from the Spectral Angle Mapper [9]. In this equation,  $\mathbf{a}$  is the unknown vector and  $\mu_i$  a known vector. The known vector,  $\mu_i$ , with the smallest angle,  $\theta_i$ , between it and the unknown vector is the best match ( $i \in K$  known vectors for each species).

$$\theta_i = \arccos \left( \frac{\mathbf{a} \cdot \mu_i}{|\mathbf{a}| |\mu_i|} \right) \quad (2)$$

Figure 7 shows a simplified two dimensional version of the SAM. The vector undergoing classification,  $\mathbf{a}$ , is shown in red and compared to two mean vectors,  $\mu_1$  and  $\mu_2$ .

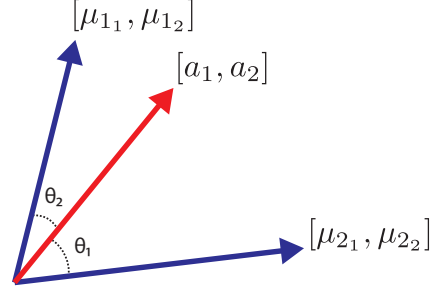


Figure 7: Example angle calculation between an unknown vector and two mean vectors in two dimensional space.

redundant, backward feature selection was performed [10]. The algorithm begins with all features, performs classification, tabulates the number of misclassifications, and then eliminates a feature before beginning again. This is an exhaustive search, trying all possible combinations of feature vectors. The process is then repeated until the fewest number of features with the lowest error rate is found.

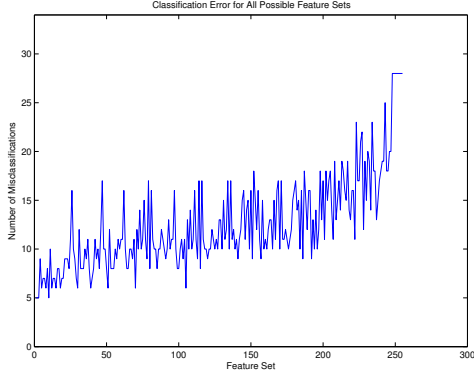
The results of this feature selection process are shown in Figure 8. Figure 8(a) shows the error rate (number of misclassifications for 34 samples) for every possible feature set combination, with all 8 of the original features as set 0 and a single feature as set 255. While the error rate of 5 misclassifications was achieved using all 8 features, only 6 wings were misclassified when using only front wing triangle areas, front wing triangle aspect ratios, rear wing cell counts, and rear wing triangle areas. (This is shown as feature set number 105 in Figure 8(a).)

An attempt to improve the system by normalizing triangle area as a fraction of wing area increased the error rate significantly, as shown in Figure 8(b). This is because several of the wings had similar area ratios. The smallest feature set supports this conclusion, as both the front and rear triangle areas have been eliminated from the feature vector. Although the error rate is double that of using unnormalized data, this may be an acceptable tradeoff when using a mobile device,

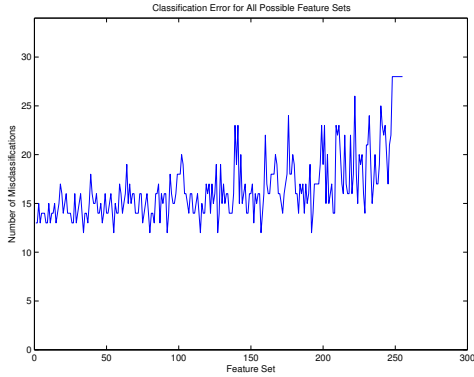
## 3 Results

### 3.1 Feature Selection

Not all of the features extracted from the triangle may be required for proper classification of the wings. To determine which features were



(a) Unnormalized Data



(b) Normalized Data

Figure 8: Classification error rate for every possible combination of features. Data has been normalized by dividing the triangle area by the total wing area.

such as a cell phone, to capture images of the wings. This way, the user would not be required to include a ruler in the image.

### 3.2 Classification Accuracy

The system was trained using a dataset of 3 families, 6 species, and 34 individuals in total. Each individual was then compared against each representative member in the database. Tables 1 and 2 show the resulting confusion matrices us-

ing both reduced unnormalized and normalized data, respectively. Using these matrices, the average accuracy of the system can be calculated using equation 3 [11]. In this equation,  $c_{ii}$  is the  $i$ -th diagonal entry of the confusion matrix, and  $n_i$  the number of samples per species. For unnormalized data, the average accuracy was 82.2%, and for normalized data 63.9%. Excluding the comparison vector from the training data resulted in a marginal decrease in overall accuracy to 79.4% for unnormalized data, and no change in accuracy for normalized data.

$$\hat{A} = \frac{\sum_{i=1}^k n_i c_{ii}}{\sum_{i=1}^k n_i} \quad (3)$$

Kappa coefficients,  $\hat{\kappa}$ , were also computed for both datasets using equation 4 and the confusion matrix, where  $c_{ti}$  is the sum of each column,  $c_{it}$  is the sum of each row, and  $N$  the total number of samples [11]. Unlike average accuracy,  $\hat{\kappa}$  accounts for the probability of a correct classification due to random chance, and is therefore a more reliable indicator of system performance. For unnormalized data, the kappa coefficient was 0.7882 (with variance 0.0061) and for normalized data 0.5741 (with variance 0.0096). This implies that the system using unnormalized data was 78.8% more accurate than if the classification was performed by randomly assigning vectors to a particular class.

$$\hat{\kappa} = \frac{p_o - p_c}{1 - p_c} \quad (4)$$

$$p_o = \frac{1}{N} \left( \sum_{i=1}^k c_{ii} \right)$$

$$p_c = \frac{1}{N^2} \sum_{i=1}^k c_{it} c_{ti}$$

$$c_{it} = \sum_{j=1}^k c_{ij}$$

$$c_{ti} = \sum_{j=1}^k c_{ji}$$

Table 1: Confusion matrix for reduced, unnormalized data.

		Predicted					
		<i>A. canandensis</i>	<i>A. verticalis</i>	<i>G. lividus</i>	<i>G. spicatus</i>	<i>S. obtrusum</i>	<i>S. rubicundulum</i>
Actual	Species						
	<i>Aeshna canandensis</i>	6	0	0	0	0	0
	<i>Aeshna verticalis</i>	0	6	0	0	0	0
	<i>Gomphus lividus</i>	0	0	4	2	0	0
	<i>Gomphus spicatus</i>	0	0	1	4	0	0
	<i>Sympetrum obtrusum</i>	0	0	0	0	4	1
	<i>Sympetrum rubicundulum</i>	0	0	1	0	1	4

Table 2: Confusion matrix for reduced, normalized data.

		Predicted					
		<i>A. canandensis</i>	<i>A. verticalis</i>	<i>G. lividus</i>	<i>G. spicatus</i>	<i>S. obtrusum</i>	<i>S. rubicundulum</i>
Actual	Species						
	<i>Aeshna canandensis</i>	4	2	0	0	0	0
	<i>Aeshna verticalis</i>	2	4	0	0	0	0
	<i>Gomphus lividus</i>	0	0	6	0	0	0
	<i>Gomphus spicatus</i>	0	0	2	3	0	0
	<i>Sympetrum obtrusum</i>	0	0	0	0	2	3
	<i>Sympetrum rubicundulum</i>	0	0	2	0	1	3

## 4 Conclusions

A prototype of the classification system has demonstrated reliability and accuracy with a limited dataset of 3 families, 6 species, and 34 individuals in total. Using unnormalized data, the average species classification accuracy was 82.2%, with a kappa coefficient of 0.7882. All but one individual was classified into the correct genera when using unnormalized data, and all but 2 when using normalized data.

Furthermore, it was determined that not all

features are required for classification of the wings, allowing for the use of a considerably smaller database and processing capability.

## 5 Future Work

Presently, the system requires user interaction to locate the triangle. Ultimately, the goal is to develop a fully automated system that will automatically locate the triangle using computer vision. This would pave the way for the development of an application for mobile devices.



Additional samples will need to be collected to expand the dataset and to better determine the reliability of SAM as a classifier, once the true statistical nature of the computed features is known. For example, additional data is required to determine if the data is normally distributed. This would allow for other classification methods, such as Gaussian maximum likelihood, to be used [10]. Furthermore, other methods, such as decision trees, may be more suited to this type of problem. Lastly, there may be additional wing features not considered in this implementation that might aid in classification.

## 6 Acknowledgements

The authors would like to thank Jeff Pelz of the Chester F. Carlson Center for Imaging Science, and Steve Diehl of the College of Imaging Arts and Sciences at RIT, for their expertise and feedback while working on this project.

## References

- [1] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang, “Searching the world’s herbaria: A system for visual identification of plant species,” in *Computer Vision—ECCV 2008* (D. Forsyth, P. Torr, and A. Zisserman, eds.), vol. 5305 of *Lecture Notes in Computer Science*, pp. 116–129, Springer Berlin/Heidelberg, 2008.
- [2] J. G. Needham, M. J. Westfall, and M. L. May, *Dragonflies of North America*. Scientific Publishers, 2000.
- [3] S. Diehl, “Mihran notes.” Personal Communication, January 2011.
- [4] N. Otsu, “A threshold selection method from gray-level histograms,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, pp. 62–66, January 1979.
- [5] R. C. González and R. E. Woods, *Digital Image Processing*, ch. 9, pp. 523–540. Upper Saddle River, NJ, USA: Prentice Hall, 2nd ed., 2002.
- [6] P. V. C. Hough, “Machine analysis of bubble chamber pictures,” in *Int. Conf. High Energy Accelerators and Instrumentation*, 1959.
- [7] J. Maurer, C.R., R. Qi, and V. Raghavan, “A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, pp. 265 – 270, Feb 2003.
- [8] E. W. Weisstein, “Point-line distance—2-dimensional.” <http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html>.
- [9] O. A. de Carvalho Jr. and P. R. Menezes, “Spectral correlation mapper (scm): An improvement on the spectral angle mapper (sam),” *AVIRIS Workshop Proceedings*, 2000.
- [10] P. M. Mather, *Computer Processing of Remotely Sensed Images: An Introduction*, ch. 8, pp. 221–224 and 242–245. John Wiley and Sons, 3rd ed., 2004.
- [11] J. R. Schott, *Remote Sensing: The Image Chain Approach*, ch. 9. Cary, NC, USA: Oxford University Press, 2nd ed., April 2007.