

HISTOGRAM-BASED TEMPLATE MATCHING OBJECT DETECTION
IN IMAGES WITH VARYING BRIGHTNESS AND CONTRAST

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

CHRISTINA DA-WANN SCHRIDER
B.S., Wright State University, 2005

2008
Wright State University

COPYRIGHT BY
CHRISTINA DA-WANN SCHRIDER
2008

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

August 21, 2008

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY Christina Da-Wann Schrider
ENTITLED Histogram-based template matching for object detection
in images with varying brightness and contrast BE ACCEPTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF Master of Science in Engineering.

Julie A. Skipper, Ph.D.
Thesis Director

S. Narayanan, Ph.D., P.E.
Department Chair

Committee on
Final Examination

Julie A. Skipper, Ph.D.

Thomas N. Hangartner, Ph.D.

Daniel W. Repperger, Ph.D.

Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

ABSTRACT

Schrider, Christina Da-Wann. M.S.E., Department of Biomedical, Industrial and Human Factors Engineering, Wright State University, 2008.

Histogram-based template matching for object detection in images with varying brightness and contrast.

Our challenge was to develop a semi-automatic target detection algorithm to aid human operators in locating potential targets within images. In contrast to currently available methods, our approach is relatively insensitive to image brightness, image contrast and object orientation. Working on overlapping image blocks, we used a sliding difference method of histogram matching. Incrementally sliding the histograms of the known object template and the image region of interest (ROI) together, the sum of absolute histogram differences was calculated. The minimum of the resultant array was stored in the corresponding spatial position of a response surface matrix. Local minima of the response surface suggest possible target locations. Because the template contrast will rarely perfectly match the contrast of the actual image contrast, which can be compromised by illumination conditions, background features, cloud cover, etc., we perform a random contrast manipulation, which we term ‘wobble’, on the template histogram. Our results have shown improved object detection with the combination of the sliding histogram difference and wobble.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. BACKGROUND	2
2.1 Object detection	2
2.1.1 Information overload	4
2.1.2 Second opinion	4
2.2 Standard terminology in computer-aided object detection	5
2.2.1 Histograms	5
2.2.2 Thresholding	11
2.2.3 Edge detection	11
2.2.4 Shape-based detection	14
2.2.5 Pattern matching	15
2.2.6 Template matching and correlation	16
3. METHOD	19
3.1 Tools used	19
3.2 Histogram differencing	22
3.2.1 Initial evaluation	22
3.2.2 Brightness consideration	27
3.2.3 Contrast consideration	31

3.2.4 Handling multiple minima	34
3.2.4.1 Contrast prediction	34
3.2.4.1.1 Building a contrast prediction library	35
3.2.4.1.2 Predict image contrast	36
3.2.5 Case studies	36
3.2.6 Algorithm evaluation	37
4. RESULTS	40
4.1 Response plane comparison	40
4.1.1 Case I: Optimum image	40
4.1.2 Cases II and III: Non-optimal and known image contrast	42
4.1.3 Cases IV and V: Non-optimal and unknown image contrast	45
4.2 Statistical analysis of method efficacy	49
4.2.1 t-Test individual terrain boards	52
5. DISCUSSION	56
5.1 Method evaluation	56
5.1.1 Case I	56
5.1.2 Cases II and III	57
5.1.3 Cases IV and V	57
5.1.4 Data analysis	58
5.1.5 Terrain board comparisons	59

5.1.6 ROC evaluation	59
5.2 Contrast prediction alternatives	60
5.3 Variable optimization	61
5.4 Method comparison	63
5.4.1 Fit with overall project goals	63
5.4.2 Combining methods	63
5.5 Future work	64
6. REFERENCES	66
APPEDIX A	A-1

LIST OF FIGURES

Figure 1. Grayscale image and histogram	6
Figure 2. Color component histograms	7
Figure 3. Identical histograms	9
Figure 4. Histogram equalization	10
Figure 5. Thresholding	12
Figure 6. Pattern matching of parallel lines	16
Figure 7. Histogram-based template matching algorithm flowchart	20
Figure 8. Terrain board diagram	21
Figure 9. Best-case terrain board and template image	22
Figure 10. Histogram smoothing	23
Figure 11. ROI elimination criterion	25
Figure 12. Assessment of histogram width for ROI elimination criterion	26
Figure 13. Single histogram differencing	27
Figure 14. Effect of changes in scene brightness on image histograms	29
Figure 15. Sliding histogram differencing for brightness consideration	30
Figure 16. A sample histogram differencing array	31

Figure 17. Effect of changes in scene contrast on image histograms	32
Figure 18. Multiple contrast-adjusted template histograms	33
Figure 19. Multiple minima occurrence in SAD arrays	34
Figure 20. Case I: Optimum response plane images	40
Figure 21. Case II and III: Response plane images	41
Figure 22. Threshold comparison: target and non-target with and without wobble	43
Figure 23. ROC analysis results for Cases II and III	44
Figure 24. Case II and III response plane comparison -70% contrast training image	45
Figure 25. Three different terrain board examples from the evaluation image set	47
Figure 26. Case IV and V histogram differencing response planes	48
Figure 27. ROC analysis for Case IV and V	49
Figure 28. ROC analysis results for +12% contrast adjusted evaluation image set	50

LIST OF TABLES

Table 1. Image contrast factors	3
Table 2. Edge detection algorithms	13
Table 3. Template matching by correlation	18
Table 4. Training and evaluation case studies	37
Table 5. Dataset statistical values used in t-test	51
Table 6. Data analysis of t-test images having +12% contrast adjustment	52
Table 7. Individual TB data analyses of evaluation images having unknown contrast	54
Table 8. Individual TB data analyses of +12% contrast adjustment t-test images	55

ACKNOWLEDGEMENT

I wish to convey my appreciation to my mother, sister, and brother for their unending love and encouragement.

For providing a platform of positive mentoring, advising and teaching, I express my gratitude to Dan Repperger, Ph.D., Julie Skipper, Ph.D. and Thomas Hangartner, Ph.D.

I shall always be grateful to my fellow team members Hrishikesh Karvir and Priya Ganapathy for their essential and influential contributions to this project.

For the financial support of this portion of the research project, I would like to thank the Dayton Area Graduate Studies Institute (DAGSI).

DEDICATION

This work is dedicated in loving memory of my dad, Ronald K. Smith, Sr.,
and grandmother, Hazel M. Smith.

1. INTRODUCTION

Speed and accuracy of locating objects within an image are two of the most important factors to consider when methods of object detection are evaluated. When a human observer is required to search for objects of interest in an image with a large field of view, the time needed to complete the task can be considerable. In this scenario, locating the objects becomes even more challenging as the luminance changes due to factors such as cloud cover or sun position. Much research has been performed to aid in object detection under conditions where brightness and contrast hinder the observer's detection of a given target within an image.

As one portion of a large-scale project in object detection research in conjunction with the Air Force Research Laboratories at Wright-Patterson AFB, an expanded template matching method for object detection was developed. Additional methods of object detection that were studied by others within the project included a weighted power spectrum edge detection method [1], an average histogram signature template matching method [2], and a shape-based detection method. Each method was evaluated using receiver operating characteristic (ROC) analysis to determine method effectiveness in locating objects of interest [3]. Information from the most effective methods obtained from the ROC analysis is fused into one single response plane providing the human observer the most likely areas for locating objects of interest.

2. BACKGROUND

2.1 Object detection

Detection of a known object of interest within a given image by a human observer may occur quickly and accurately when images are obtained under ideal conditions. However, the potential for the observer to overlook the object increases when the image is obtained under non-ideal conditions, such as when brightness and contrast changes visually blend the object into its surroundings [4]. Object edges that human observers use to define the boundary between an object and its surroundings become blurred or broken, which causes the observer to overlook the object. Some factors that contribute to brightness and contrast effects are included in Table 1.

Unfavorable consequences may arise if the human operator fails to locate an object of interest in the event that the object were a potential target, such as an enemy aircraft, tank, or weapon system in military images, or a malignant mass or vascular obstruction in medical images. In addition, incorrectly assigning positive detection of enemy aircraft when the aircraft is an ally, or detecting a malignant mass when none is present may result in ill-fated consequences, such as friendly fire or unnecessary surgeries. For these reasons, studies in object detection and recognition remain in the forefront of image processing research and development.

Table 1. A partial list of factors that can potentially affect image contrast. Many of the effects are similar for various conditions, but each effect can be problematic in many target detection algorithms that depend upon strong boundaries and little noise.

Factor	Source	Effect
Time of day	Sun or moon position	Brightness, shadow and edge changes
Artificial light source	Incandescent, fluorescent	Color, brightness, shadow and edge changes
Natural light source	Direct or reflected sunlight	Color, brightness, shadow and edge changes
Precipitation	Rain, sleet, snow, fog	Decreased shadow and edge definition
Cloud cover	Cloud, shadow	Decreased shadow and edge definition, object visibility
Season	Vegetation, color	Shadow changes and edge definition
Natural camouflage	Vegetation, earth, stone	Partial view, decreased shadow and edge definition
Artificial camouflage	Tarp, netting, structure	Partial view, decreased shadow and edge definition
Camera blur	Camera movement	Blurred boundaries, decreased edge definition
Camera to object distance	Arial or surface camera	Magnification, pose, perspective
Camera software	Automatic compression or correction	Background noise, blurred boundaries
Camera resolution	High, medium, low	Background noise, edge definition
Image saturation	Reflected or direct light source	Shadow and edge changes
Motion	Object or camera	Blurred boundaries, decreased edge definition
Rotation	Object or camera	Shadow and edge changes

With current technology, automatic target recognition (ATR) might provide the most advantageous solution for enemy target detection, and a great deal of research in the area of ATR continues. However, ATR methods do not offer 100% object detection accuracy, making it necessary to employ the human observer in the process. Rather than exclusively using autonomous decision-making algorithms, object detection algorithms are used to reduce information overload by offering fewer target candidates and to allow a second opinion for selected object verification by human observers. This can potentially reduce the number of bad choices and tragic results.

2.1.1 Information overload

Military information gatherers capture large numbers of images world-wide every second from sources such as Unmanned Aerial Vehicles (UAV) and satellite cameras [5]. In fact, terabytes of image data are acquired, stored, and assessed by the military daily [6]. Scanning terabytes of images for potential enemy targets is incredibly time consuming and may overwhelm military image analysts. Similarly, more than 300 million medical images are taken in the form of radiographs or Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) scans annually in hospitals and medical centers [7]. Radiologists spend a substantial amount of time visually scanning each image for abnormalities. Software that is able to assist in the detection of objects of interest in military and medical images potentially saves valuable time and lives.

2.1.2 Second opinion

Instead of solely relying on technology to detect a potential enemy target, placing the human in the final decision-making process provides a second opinion about the target. Objects having similar shapes, lines, or material surfaces may deceive object detection software. Allowing the human to interact and make final decisions based on the information provided by the software and prior knowledge about material identification, edge observation, and situation awareness decreases the likelihood of incorrectly identifying targets.

2.2 Standard terminology in computer-aided object detection

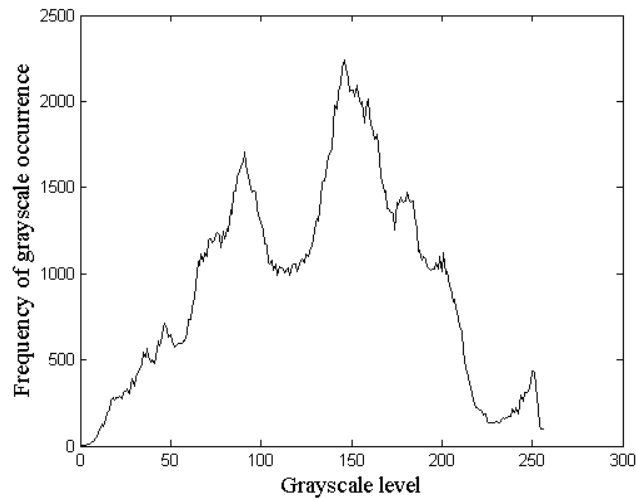
2.2.1 Histograms

The histogram of an image is a graphical representation of the frequency of pixel intensity occurrence in the image. In an 8-bit grayscale image, such as a scanned document, 256 grayscale values are possible. Another image acquisition method, for instance, a digital camera, might produce 16-bit images, which results in 65,536 possible grayscale values. A large peak in an image histogram identifies a group of common grayscale values, whereas a valley indicates that one or several adjacent grayscale values are less common in the image (Figure 1). In a color image, the individual red, green, and blue (RGB) color components can be similarly viewed as histograms (Figure 2). Though the histogram represents every pixel within an image, spatial information about each pixel is lost, i.e., the number of pixels at a specific grayscale value is represented in the histogram, but the histogram does not provide the location of those pixels within the

original image (Figure 3). Also, because the pixel location information is lost, an image cannot be regenerated from its histogram.



(a)

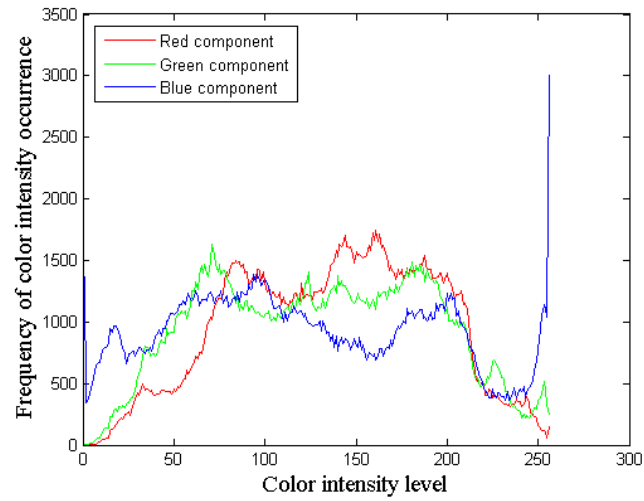


(b)

Figure 1. Grayscale image and histogram. (a) An 8-bit grayscale image of a morning glory and (b) its associated histogram. The vertical axis of the histogram represents the frequency of pixel occurrence; the horizontal axis represents the grayscale levels. In this histogram, black pixels are located at zero, and white pixels are located at 255. This image presents a large range of grayscale levels from near zero to 255. The greatest number of pixels in the image is in the mid-gray level, as evidenced by the largest peak showing more than 2000 pixels near a grayscale level of 150. In comparison, approximately 500 pixels neighboring the grayscale level of 250 represent the white regions in the image.



(a)



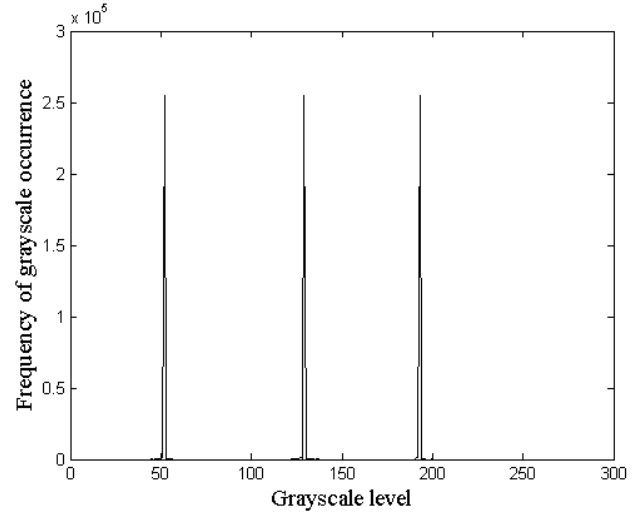
(b)

Figure 2. Color component histograms. (a) Color image of the morning glory and the (b) red, green, and blue (RGB) color component histograms. The vertical axis represents the frequency of color intensity level occurrence; the horizontal axis represents the color intensity levels. In this histogram, less saturated colors are located near zero, and more saturated colors are located near 255. The large peak, approximately 3000 pixels, in the blue histogram corresponds to the highly saturated blue of the morning glory.

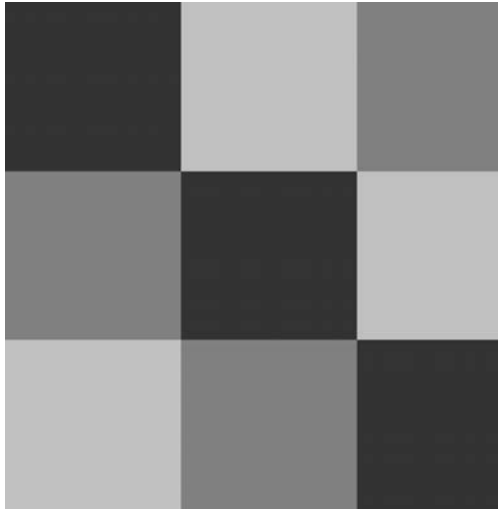
Even though the histogram does not provide pixel locations, knowing the shape of the histogram may be valuable. In some instances, manipulating the histogram can provide a more visually pleasing image. For example, in histogram equalization, a transfer function is applied to the histogram that fills in some of the valleys and spreads out some of the peaks to utilize more of the available grayscale values (Figure 4) [8, 9]. However, since histogram equalization assigns new grayscale levels to some pixels, quantitative analysis is no longer possible. Histogram equalization also fails in several situations; its effectiveness diminishes when variation in contrast occurs across an image, as might be the case with partial cloud cover [9]. Adaptive histogram equalization considers the issue of contrast changes by applying similar transfer functions to the histogram of each pixel's neighborhood. Even though this method outperforms traditional (global) histogram equalization for low contrast images, there are at least two drawbacks associated with adaptive histogram equalization: the process is computationally intensive, and the result is an artificial representation of the original image.



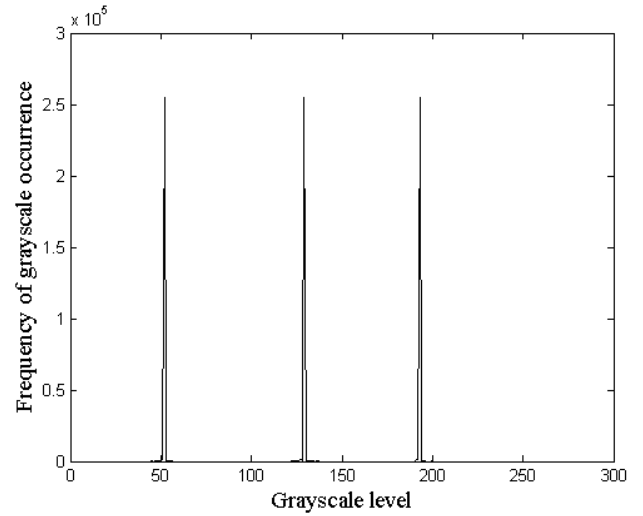
(a)



(b)



(c)

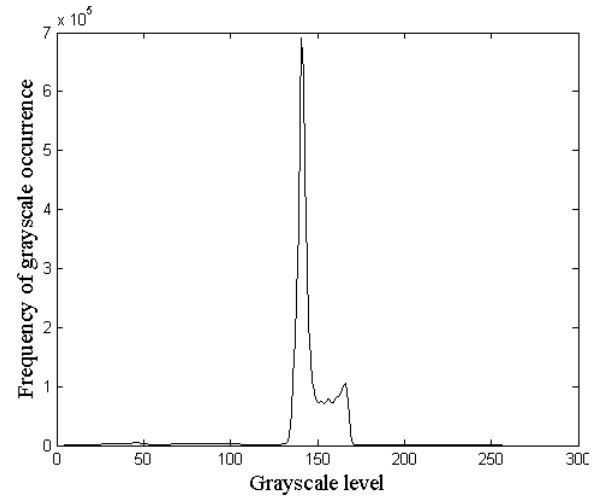


(d)

Figure 3. Identical histograms. Identical histograms can be generated from different images, as shown here. (a, b) The grayscale stripe image and its histogram present the same number pixels at each graylevel because the stripes are identical in size. (c, d) The grayscale block image generates an identical histogram to the striped histogram (b). Note that the pixel location within the image is not represented in the histogram.



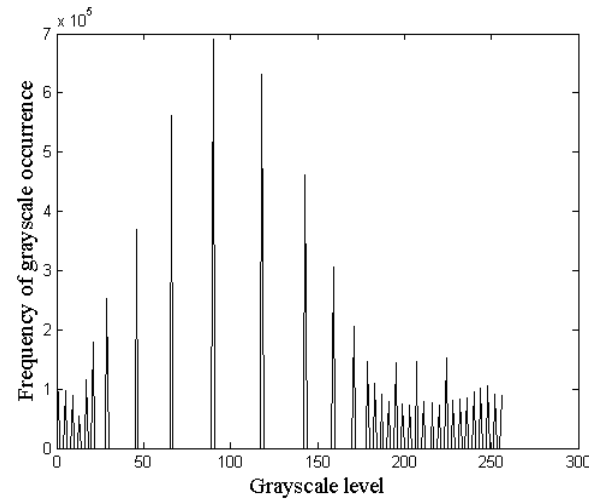
(a)



(b)



(c)



(d)

Figure 4. Histogram equalization. One commonly used method of separating clusters of pixels that congregate about a particular range of grayscale levels is histogram equalization. (a) Looking at the flag image, the sky and cloud appear to contain mostly the same grayscale value. However when histogram equalization is performed, (c) the subtle changes in grayscale value becomes easier for the human eye to identify in the equalized image. (b) The large and narrow peak in the original image's histogram is separated and spread across approximately 100 grayscale levels, as seen in the equalized histogram (d). Unfortunately, in the process of improving the identification of variances in the sky, details such as the stars, stripes, and rope are lost.

2.2.2 Thresholding

Thresholding an image generates a black and white, or *binary*, image by defining one range of grayscale values as the foreground and the remaining grayscale values as the background. Localizing a specific range of grayscale values based on object intensity sets the object as the foreground and facilitates object detection. However, setting a fixed threshold for use across multiple images is ineffective in locating features because changes in brightness levels result in fixed threshold method failure, as shown in Figure 5.

To circumvent the limitations and disadvantages of fixed threshold techniques, adaptive thresholding methods provide localized histogram equalization at each pixel that can accentuate subtle texture and detail in some cases. Adaptive thresholding techniques are generally suited to illumination normalization, textural enhancement, and anomaly detection [10]. However, time becomes the price paid for using pixel-by-pixel histogram equalization calculations of adaptive thresholding.

2.2.3 Edge detection

All object detection methods key on various important features of the target. Man-made objects tend to have straight edges, which make edge detection algorithms highly effective in many detection applications. An important feature of any edge detection algorithm is the preservation of the “useful structural information about object boundaries” [11]. John Canny, in 1986, defined three important criteria for an effective edge detection filter. The filter must return a low number of errors, maintain localized

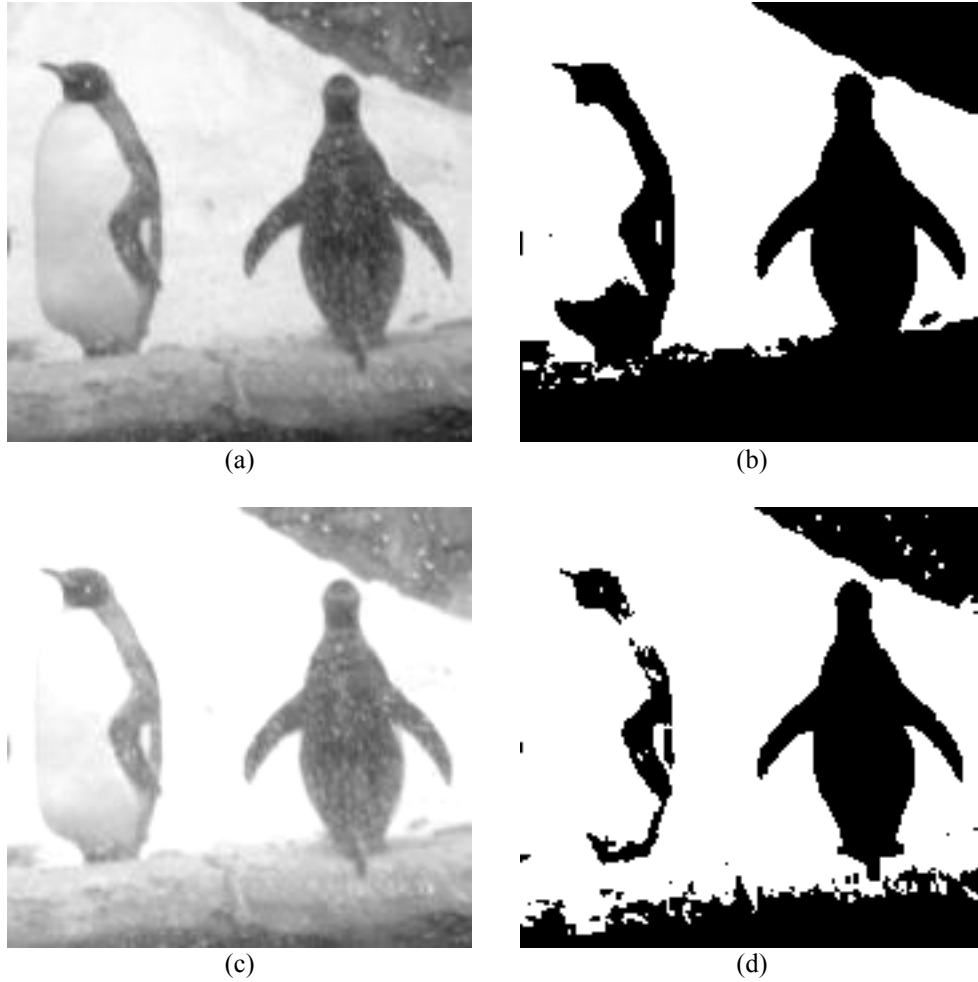


Figure 5. Thresholding. (a) To the penguin image a threshold level of 0.75 was used in MATLAB to generate (b) the binary image. (c) After applying contrast and brightness adjustments in Adobe Photoshop to the penguin image (a), an identical threshold level yielded (d) a different binary image. Notice that in (b) the shape and form of the penguins was retained, but in (d) the boundaries and shape information make the penguin on the left more difficult to discern.

edge points, and only provide one response to a local edge [11]. A summary of edge detection methods is presented in Table 2.

Table 2: Edge detection algorithms. A partial compilation of common edge detection methods is presented. In each method described, (x, y) refers to spatial coordinates within an image. Although other coordinate variables may be found in documented sources, the (x, y) coordinates were used here for unity and clarity of method comparison. In the edge detection operators, the variables I and f represent intensity value and function, respectively.

Method	Operator	Process	Disadvantages
Laplacian [8]	$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$	High-pass filter, Linear second order derivative brightness estimation	Incomplete boundary information, background noise
Difference of Gaussian [12]	$DoG(x, y) = I_{G_2(x, y)} - I_{G_1(x, y)}$	{a} Band-pass filter, subtraction of two Gaussian smoothed images	Incomplete boundary information, background noise
Laplacian of Gaussian [12]	$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$	{b} Band-pass filter, second derivative of smoothed image	Incomplete boundary information, background noise
Canny [11]	$SNR, Localization = \frac{\left \int_{-w}^w f(x) dx \right }{\sqrt{\int_{-w}^w f^2(x) dx}} \times \frac{ f'(0) }{\sqrt{\int_{-w}^w f'^2(x) dx}}$	Zero-crossings of Laplacian of Gaussian, single pixel edge	Discontinuous boundary
Roberts' cross operator [13]	$R(x, y) = \sqrt{(I_{x+1, y+1} - I_{x, y})^2 + (I_{x, y+1} - I_{x+1, y})^2}$	Magnitude of orthogonally adjacent pixels in two directions	Magnification of noise, sensitivity to edge orientation
Sobel [14]	$S(x, y) = magnitude = \sqrt{\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2}$	Magnitude of derivatives from orthogonally adjacent pixels in two directions	Computationally expensive, subtle edge noise
Kirsch [15]	$K(x, y) = \max_{i=0}^n \left[I_{\max} \left[5 \left((x, y)_{i+1} + (x, y)_{i+2} \right) - 3 \left((x, y)_{i+3} + (x, y)_{i+4} + \dots + (x, y)_{i+m} \right) \right] \right]$	Maximum of derivatives from orthogonally adjacent pixels in eight directions	Subtle edge noise
Frei & Chen [16]	<i>Variable operator</i>	Cosine of the square root of multiple summed, independent, orthogonal behavior kernels	Subtle edge noise
Fourier transform [8]	$F(x_2, y_2) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{-i2\pi \left(\frac{x_2}{N} + \frac{y_2}{N} \right)}$	{c} High frequency content from a large number of Fourier terms represents an edge	Sensitive to noise, blurred boundaries, number of edge pixels
Hough transform [17]	$\rho(\theta) = x_0 \cos \theta + y_0 \sin \theta$	{d} Maximum peaks in crossing polar sinusoids corresponding to points in real image space	Memory-intensive, time consuming
Prewitt [18]	$X - \text{kernel} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad Y - \text{kernel} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	Approximates magnitude and direction of edge using convolution of two 3 x 3 kernels' max response	Sensitive to edge direction

{a} G = Gaussian filtered image; {b} σ = standard deviation; {c} N = square image size; {d} in (ρ, θ) space, ρ = vector distance from origin and θ = vector angle

2.2.4 Shape-based detection

As an additional method for the larger-scale project, a shape-based object detection method is under study. The first step in this method is object segmentation using a band-pass edge detection filter that, in this instance, consists of the difference of two low-pass Gaussian filtered images with sequentially incremented standard deviations. Thresholding the difference image produces a binary image of edges and contours. Additional processing with morphological and logical operations eliminates specific unwanted features, for example regions that exceed predicted object size, and fills broken boundaries and holes. Next, objects of a specified shape are identified through a solidity measurement. Object solidity is determined as the ratio of the object ROI to the area of the smallest polygon required to enclose the object. This allows objects resembling a known polygon to be extracted from the image by implementing a solidity-based thresholding procedure.

Another relatively new and popular shape-based detection method is active shape modeling (ASM). ASM uses a deformable model, or *atlas*, which is an elastic-like template that can be stretched or shaped to match target boundaries in an image. To provide an automated system, the variability of feature shapes throughout a set of independently obtained images requires training models using images with identified landmark points to identify similar features [19]. ASM could potentially be an effective method of target detection for the overall scope of the project; however, because another shape-based method is being pursued by our lab as part of the larger project, we elected not to explore the ASM method at this time.

Image processing methods that have the ability to extract features from images also fall into the shape-based detection category. The top hat filter, a common feature detection method that is a Fourier transform-based kernel, looks for areas of bright and dark pixels in definite patterns. The kernel consists of a center region, or *crown*, and a larger outer region, or *brim*. If the difference in the brightest (or darkest) brim and crown pixels exceeds a chosen threshold, then the central pixel in the top hat is maintained, otherwise it is replaced by an average value of brightness from the brim. The elimination of undesirable pixels makes the top hat filter a noise suppression operator [8]. As with all thresholding operations, choosing the appropriate difference threshold to retain the desired pixels becomes challenging with different images.

2.2.5 Pattern matching

In pattern matching, algorithms search for known textures, homogeneous patches, and spatial patterns of intensity values. Determining the difference between brightest and darkest values in an area will determine if the texture in an area is uniform, a small difference, or rough, a large difference [8].

Pattern matching by means of frequency analysis is another detection method that is under investigation for our larger-scale project. The pattern is representative of the object's geometric signature; for example, the known geometric signature found in a scaled model of a Scud missile launcher is a set of three or four parallel lines as seen in Figure 6. These parallel lines are represented as peaks in the Fourier domain. Information such as the frequency of the pattern and the object orientation can thus be extracted using Fourier analysis. Analysis of images with a known source-to-object distance enables prediction of the edge width and spacing between edges so that a band-pass filter can be

used to extract the Fourier peaks that correspond to the target edges. Unwanted noise is reduced by peak amplitude thresholding [1].

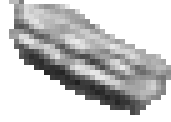


Figure 6. Pattern matching of parallel lines. The geometric signature of parallel lines found in the Scud missile launcher is located by means of frequency analysis. In the Fourier domain, these parallel lines are represented as peaks that can be used to determine object orientation and pattern frequency.

2.2.6 Template matching and correlation

Template matching uses a similarity criterion for locating an object, where one common method calculates a correlation coefficient using the following equation:





$$\rho = \frac{\sum_x \sum_y (A_{xy} - \bar{A})(B_{xy} - \bar{B})}{\sqrt{\left(\sum_x \sum_y (A_{xy} - \bar{A})^2 \right) \left(\sum_x \sum_y (B_{xy} - \bar{B})^2 \right)}}, \quad (1)$$

where A and B are image matrices, \bar{A} and \bar{B} are the 2-dimensional means of the respective image matrices, and (x, y) are the spatial coordinates within A and B [20]. This correlation coefficient closely resembles a traditional statistical correlation, with the difference being that the traditional method is calculated in one dimension instead of two dimensions. A high correlation coefficient in a pixel-by-pixel comparison between the template and the region of interest (ROI) indicates a good match. However, this template matching method quickly fails by influence of disturbance, such as with illumination changes and rotation of the object. For example, a circumstance where template matching

can fail is when real-world images acquired in outdoor settings are affected by the changes in sun location and cloud cover [4]. Template matching will also fail when the object in question is rotated within the image. This limitation is apparent when template matching is used to search text for a letter or word. When a letter is rotated, a low correlation coefficient is returned and the letter is not recognized as a match to the template letter. In some instances, the object orientation is not that of the template; for example, the letter 'A' shown in Table 3 was rotated 0, 45, 90, 135 and 180 degrees from the template orientation and processed to determine the correlation coefficient that represents how closely the template matches the object image. In images with 3-dimensional objects, the rotation of the object also results in illumination changes, since the light source will illuminate different parts of the object generating unique shadows and highlights. As a result, object-defining features may be lost with illumination changes. It is possible to iteratively rotate the object template incrementally and calculate the correlation coefficient at each position in order to achieve a better match, but this adds processing time.

Another potential template-matching method that is also being considered in the larger scope of this research project uses an average histogram signature template [2]. Our proposed method of template matching uses a differencing calculation of template and ROI pixel intensities to locate the object of interest, and it will be shown that our method is unaffected by the rotation and illumination changes of the object.

Table 3. Template matching by correlation. Letter templates created in Microsoft Power Point. These templates were compared to the ground truth, or 0° of rotation, to demonstrate the ineffectiveness of template matching under rotational conditions. The degrees of rotation from top to bottom are 0°, 45°, 90°, 135° and 180°. The corresponding correlation coefficients as calculated in MATLAB are shown; a value of 1 indicates a perfect match. Image rotation causes some interpolation artifacts that also lower the absolute value of the correlation coefficient.

Template	Object	Correlation coefficient, ρ
A	A	1.0000
A		0.2967
A		0.3128
A		0.3644
A		0.2175

3. METHOD

The following sections include descriptions of tools used, the techniques applied, cases studied, as well as how the algorithm was evaluated. An overview of the algorithm can be seen in the pictorial flowchart (Figure 7).

3.1 Tools used

The algorithm was developed and tested using MATLAB[®] (R2006, MathWorks, Natick, MA) with Image Processing and Signal Processing Toolboxes.

Test images were acquired from three unique 8' x 8' terrain boards containing artificial grass, rocks, trees, roads, hangars, tanks, aircraft, and Scud missile launchers. The images studied were taken under three light position angles, $\alpha = 10, 45,$ and 90 degrees, one board tilt angle, $\beta = 90$ degrees, and one camera zoom setting = 30 mm, (Figure 8), using an 6.1 megapixel, 12-bit, camera (Model EOS 10D, Canon, Inc., Lake Success, NY) at a distance, $d = 56.1$ inches, with a wide angle lens (17-40 mm, Canon, Inc.). The light sources for the terrain board included positional halogen and fixed fluorescent lighting. The target object replicas (aircraft, tanks, and Scud missile launchers) were presented as possible real-life situations on the terrain boards in different positions and orientations in each image. The object scale was known to be 1:700, with the length of one Russian Scud missile launcher being approximately 75 feet [21]. With

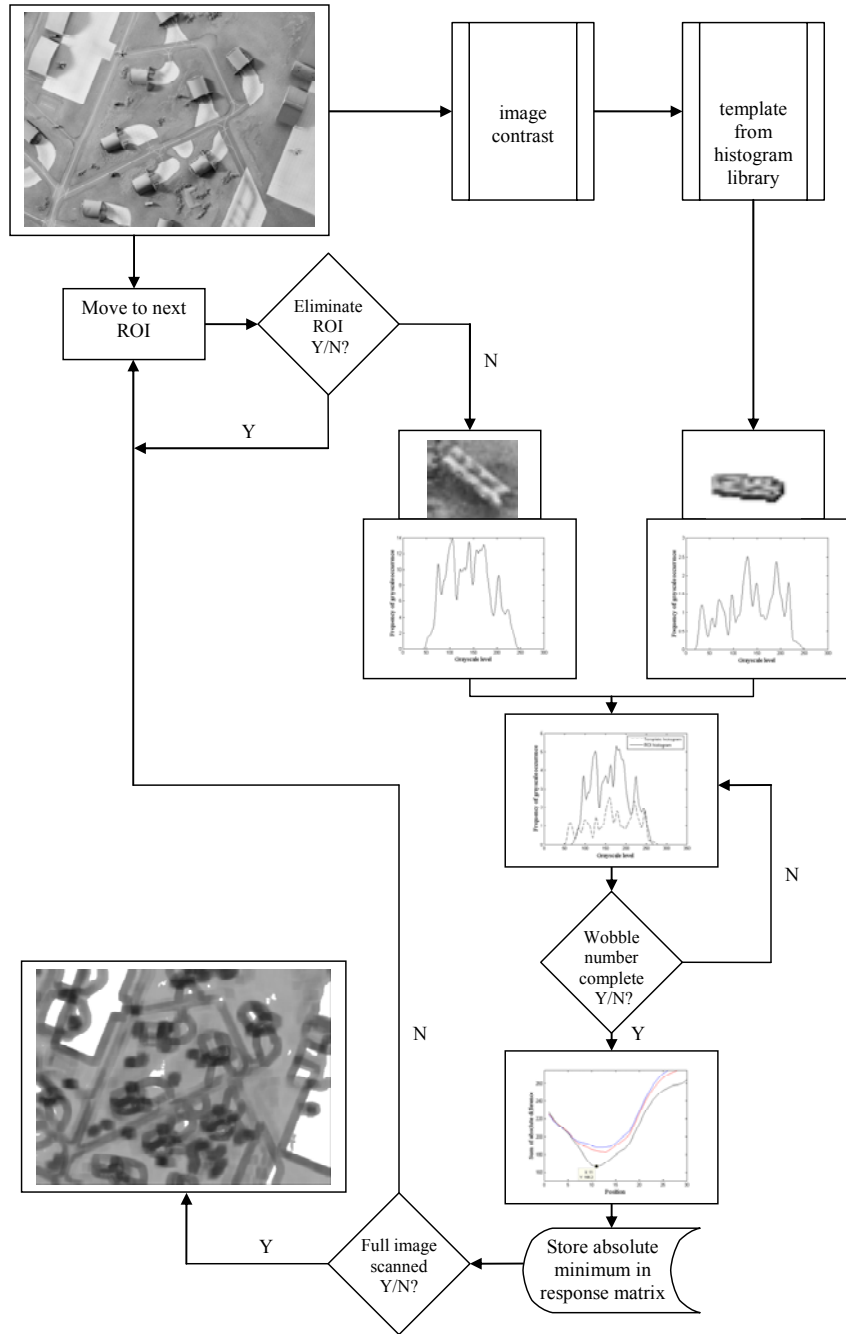


Figure 7. Histogram-based template matching algorithm flowchart. The first step in the histogram-based template matching method is the image contrast prediction. Once the image contrast is known, the optimum contrast-adjusted, or wobbled, template histograms can be chosen. At each region of interest (ROI) of similar size to the template, a sliding difference between the template and ROI histograms is calculated. This calculation is performed for each of the chosen wobbled templates. The absolute minimum of the wobbled difference calculations is stored in a response plane. The ROI evaluations are performed across the image until all have been evaluated. A response plane containing grayscale values from 0 to 256, where 0 represents a perfect match and 256 represents a definite non-match, is the output of the process.

our terrain boards, the distance from the board to the camera was known, and in a real-world situation the distance from the earth's surface to an Unmanned Aerial Vehicle (UAV) or other airborne camera would similarly be known.

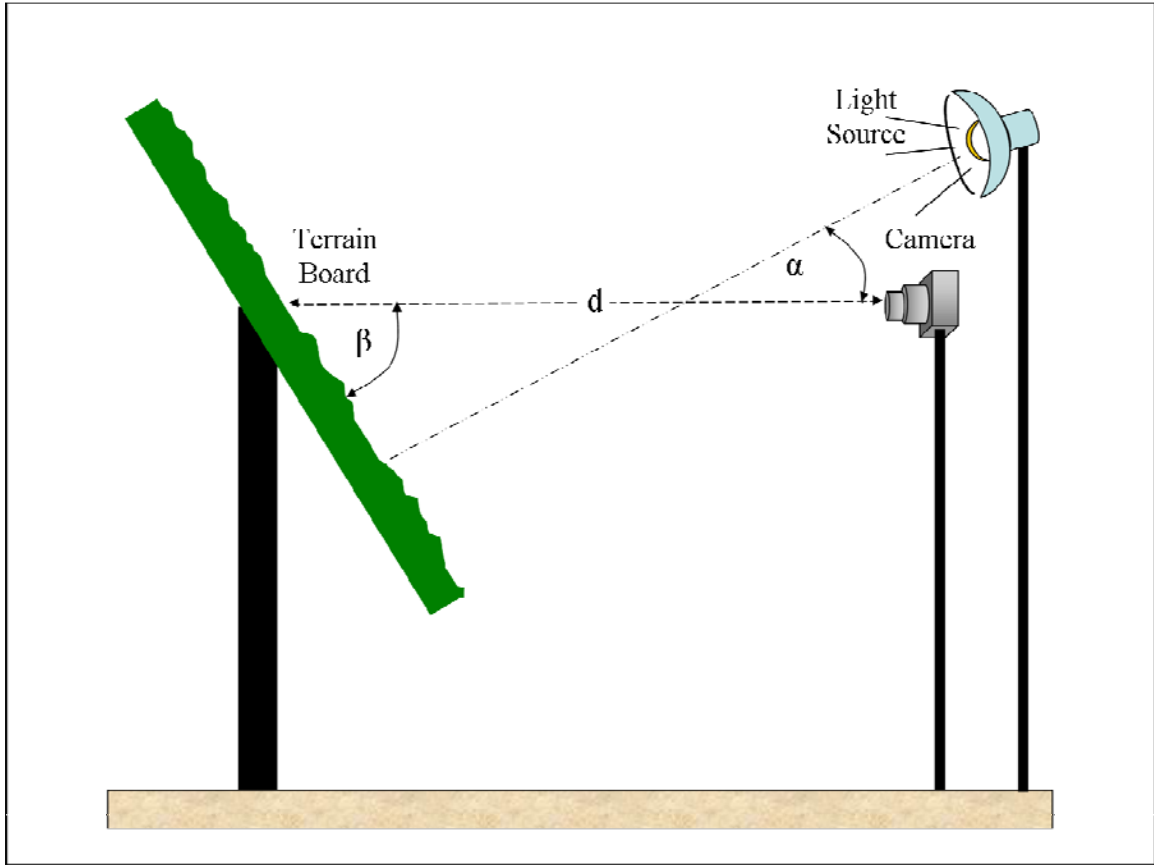


Figure 8. Terrain board diagram. The terrain board images studied were taken under three light position angles, α : 10, 45, and 90 degrees, one board tilt angle, β : 90 degrees, and one camera zoom setting: 30 mm, using a 6.1 megapixel, 12-bit, camera (Model EOS 10D, Canon, Inc., Lake Success, NY) at a distance, d , of 56.1 inches with a wide angle lens (17-40 mm, Canon, Inc.).

The template image that was used to locate potential targets in a test image was a closely cropped image of a Scud missile launcher on a white background. From a selected best-case terrain board image, we extracted the template image (Figure 9) and set all background pixels to white using Adobe Photoshop (Version 4.1, Adobe Systems Incorporated, San Jose, CA).

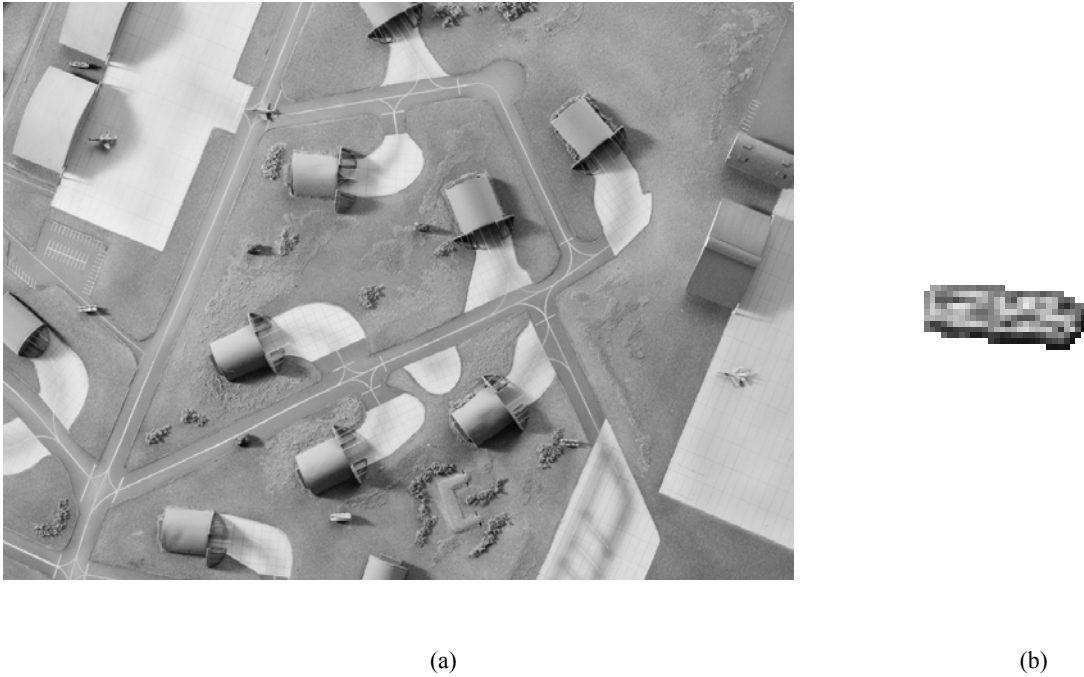


Figure 9. Best-case terrain board and template image. (a) Selected best-case terrain board image from which (b) the template image of a Scud missile launcher was extracted. The template background pixels were set to white in order to extract the signature histogram of the Scud missile launcher. The training image set was generated by applying contrast changes across this image.

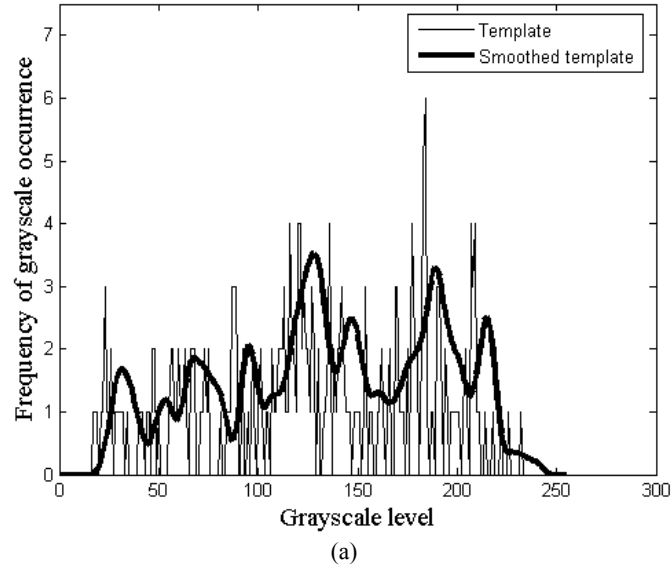
3.2 Histogram differencing

3.2.1 Initial evaluation

Selection of an image ROI occurred in a sliding block processing operation in increments of two pixels within the test image. The block processing step-size can be reduced or increased with a consequence of increased processing time or overlooked targets, respectively. Based on the object size, the ROI size was chosen to be 33 pixels by 33 pixels, which is adjustable as a known function of source-to-object distance. Because the histograms of the template and ROI were noisy, and the underlying histogram was most important, smoothing and scaling of both histograms were performed before applying the differencing algorithm (Figure 10). The smoothing process convolved a

Hamming window filter, a raised cosine shape digital signal smoothing filter, with the

Template



ROI

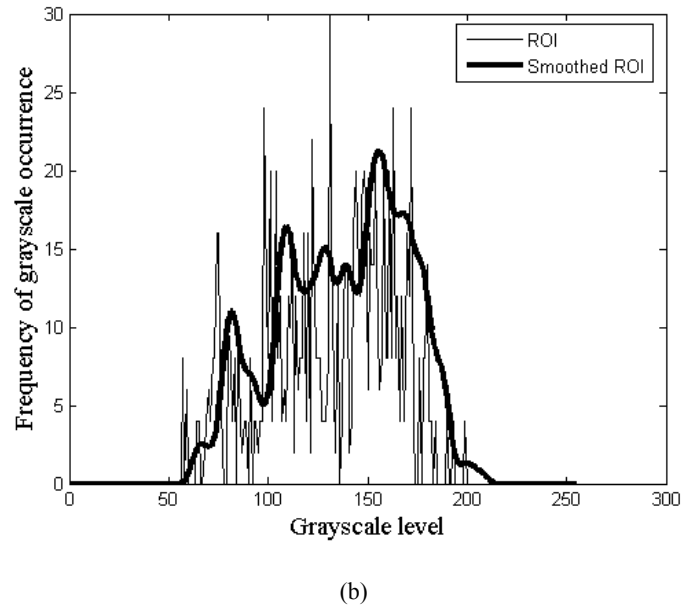


Figure 10. Histogram smoothing. (a) The histogram of our template image before and after smoothing. (b) A histogram, before and after smoothing, of an ROI containing shrubs and grass that should not match the template histogram. A Hamming window filter (15-point) was used to smooth the template and ROI histograms.

histograms of the template and ROI; scaling the smoothed template and ROI returned the area under the curves to approximately that of the unsmoothed histograms.

To reduce computation time, some homogeneous ROIs were excluded from processing (Figure 11). These excluded ROIs, such as areas of grass, cement, or asphalt that yield large and narrow ROI histogram peaks, present a poor histogram match. It was learned during testing that a homogeneous area could be detected if 90% of the area under the curve (AUC) had a width less than or equal to 7.8% of the total number of grayscale bins, which represents an approximate range of 20 grayscale levels in width (Figure 12). The width of each ROI histogram was determined using the following equation:

$$Histogram\ Width = \sum_{i=255}^{bin_{upper}} frequency(bin_{i,upper}) - \sum_{i=0}^{bin_{lower}} frequency(bin_{i,lower}), \quad (2)$$

where bin_{upper} decrements by one and bin_{lower} increments by one until the sums of both tails equal approximately 5% of the total AUC. In regions where the ROI was found to meet this homogeneity criterion, no additional processing occurred and a maximum difference possible that is equal to the total number of pixels in the ROI was recorded to indicate “no match” in the response plane.

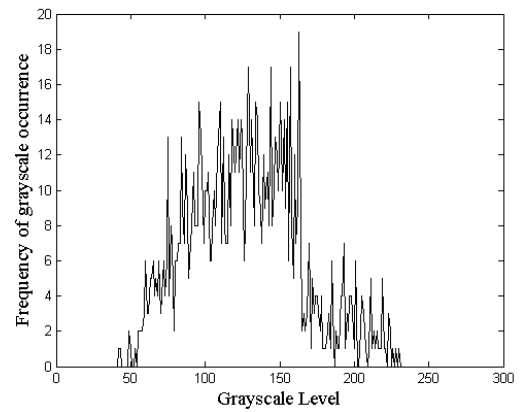
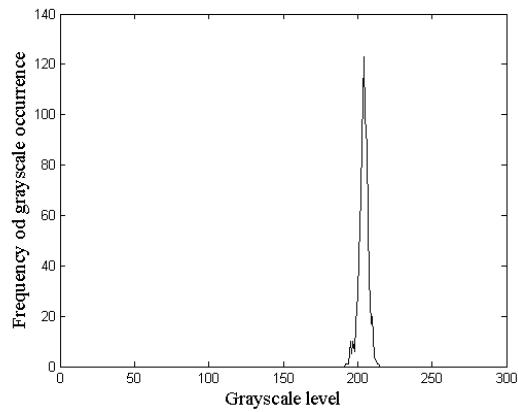
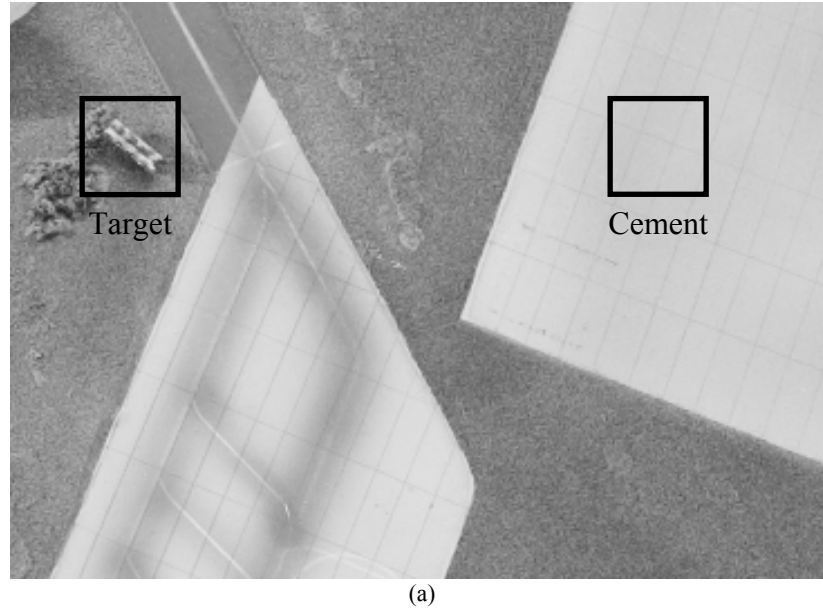


Figure 11. ROI elimination criterion. (b) A homogeneous ROI is identified by its narrow histogram. In image (a), the cement has fairly uniform grayscale values. This is an example of a region that should be excluded from histogram differencing process in order to reduce unnecessary processing. In comparison, the target ROI containing the Scud missile launcher in image (a) produces a broad histogram (c) that represents a region that should not be excluded from the histogram differencing process.

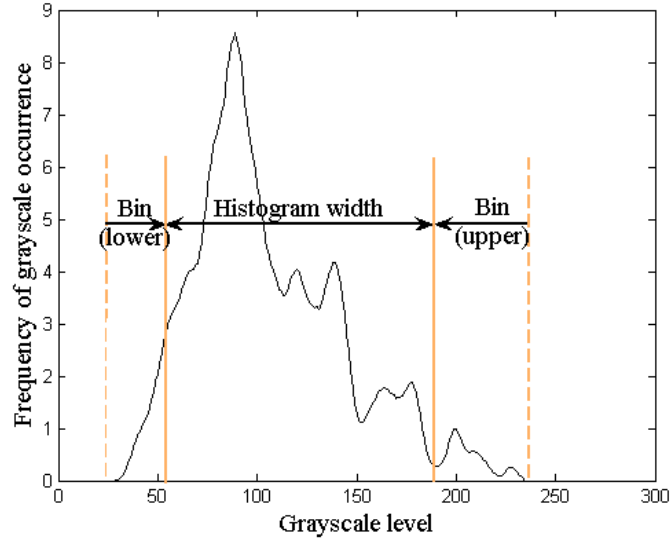


Figure 12. Assessment of histogram width for ROI elimination criterion. Homogeneous regions throughout the image, such as the cement region shown in Figure 11, are identified by a tall, narrow histogram peak; this allowed us to develop a method for eliminating these regions from the subsequent time-intensive computations, thus decreasing overall processing time. We defined a homogeneous area of our 8-bit images when 90% of the area under the curve had a width less than or equal to 7.8% of the total number of grayscale bins, which represents a range containing approximately 20 grayscale levels. The 90% area under the curve does not contain the tails of the histogram where the summations of lower and upper bins represent the 10% tails that are not evaluated. Thus, if at least 90% of an ROI histogram has a width less than or equal to 20 grayscale levels, the ROI is eliminated from additional processing and a maximum difference value is placed in the response plane to represent ‘no match’ for that region.

To compare the ROI and template histograms not excluded, our method utilized a sum of the absolute difference (SAD) concept that is similar to what is commonly employed in detecting and estimating motion in video imaging by subtracting two screen shots to detect subtle differences like movement [22]. In video SAD, a comparison between video cells calculates an absolute value of the difference between the pixel value of the original image area and the parallel comparison pixel in the evaluated frame. Instead, our method subtracts the ROI and template histograms, computes the absolute value of the difference, and stores the SAD in a corresponding output response plane matrix (Figure 13). Thus a low SAD represented a high target confidence. Thresholding

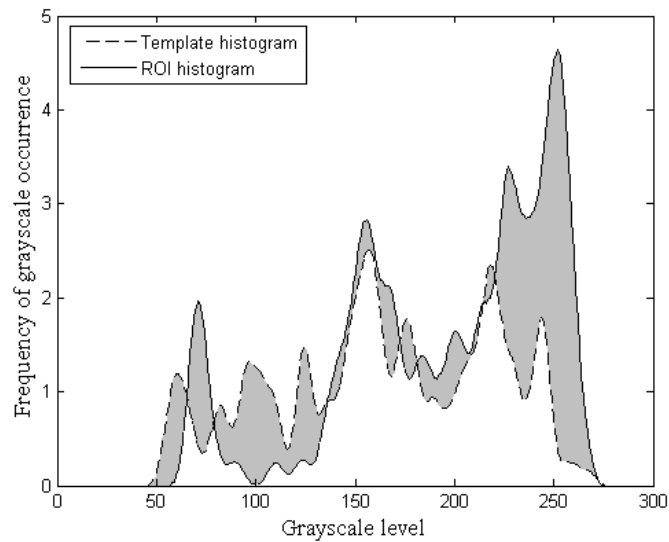


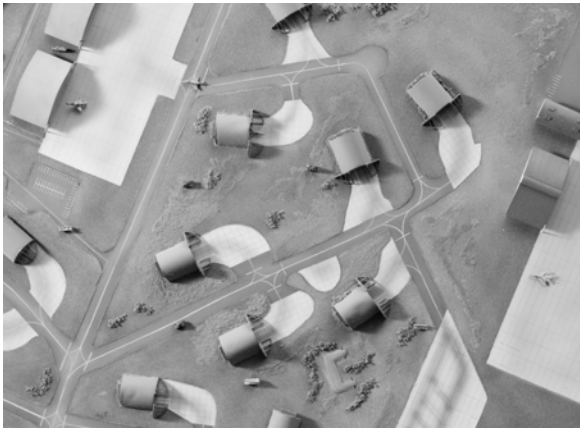
Figure 13. Single histogram differencing. The sum of the absolute difference (SAD) between the template and ROI histograms is calculated and stored in a response plane matrix. In this case, a large portion of the two histograms overlap and return a reasonably low difference (the shaded region). However, when brightness changes occur, a shift in the histogram to the left or right will return a higher SAD that might lead to a missed target.

the response planes revealed areas with histogram signatures that are similar to the template. A high occurrence of ROIs incorrectly identified as target regions was observed in the response plane. It was learned that brightness shifts and/or contrast reduction or expansion seemed responsible for the incorrect detection, and so the next step was to handle variations in image brightness.

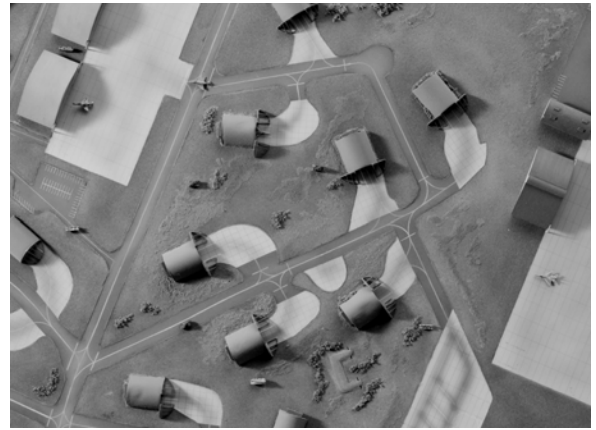
3.2.2 Brightness consideration

When the scene brightness was approximately equal to template brightness, simple histogram differencing located all of the objects of interest in our test images. However, when scene and template brightness diverged, brightness changes shifted the histogram left or right (Figure 14) and prevented a conclusive match when the histograms were subtracted. To circumvent this issue we used a sliding difference that rendered our

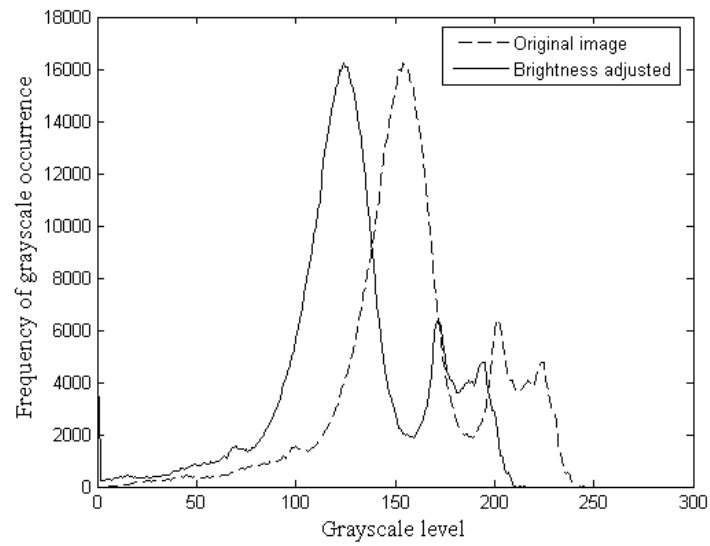
method insensitive to brightness changes. Similar to a convolution operation (but without flipping either histogram), the ROI histogram was positioned to the right of the template histogram and incrementally shifted left, gradually increasing the amount of overlap between the two signatures (Figure 15). At each step in the sliding process, the SAD was calculated and stored in a one-dimensional array (Figure 16); the minimum of that difference array was saved in a spatially corresponding response matrix. Minima in this matrix represented the best matches, such that thresholding the response matrix identified the spatial locations of possible targets. We accepted the identification of an incorrect target rather than miss an actual target at times when areas containing rocks or trees, for example, exhibited similar histograms to the template. The balance between false positive and false negative detections is a function of the selected response plane threshold. In practice, an appropriate threshold would be determined by the human operator who considers situational risks and the time criticality of the decision-making tasks.



(a)



(b)



(c)

Figure 14. Effect of changes in scene brightness on image histograms. (a) The original image, which was considered to have optimal brightness and contrast. (b) Using Adobe Photoshop, a brightness level reduction was applied to the original image. (c) A histogram comparison plot of the original and brightness-adjusted histograms that demonstrates a left shift with a brightness reduction and saturation at the grayscale value of zero.

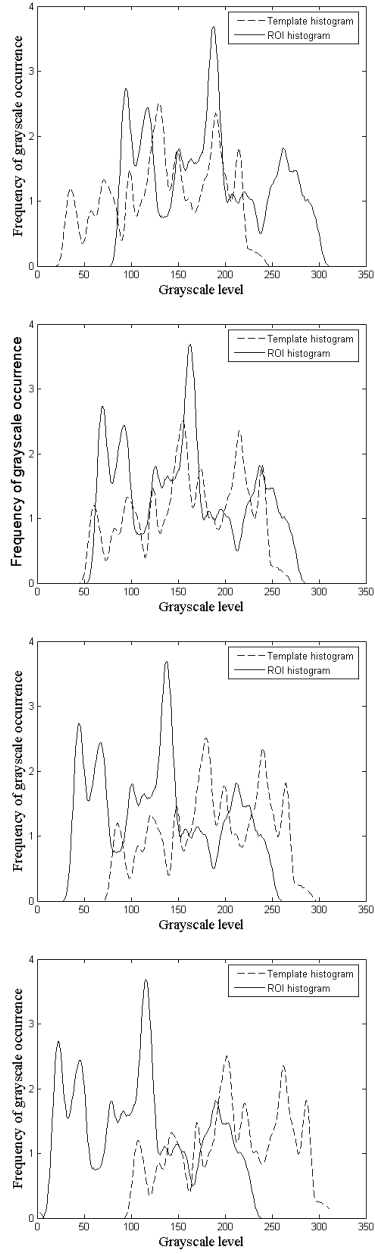


Figure 15. Sliding histogram differencing for brightness consideration. A sliding sum of the absolute difference is performed that facilitates brightness insensitivity in the target detection algorithm. Similar to a convolution without flipping either histogram, each of the histograms is incrementally circularly shifted by one bin and the absolute sum of differences at each position is calculated and stored in an array (Figure 16).

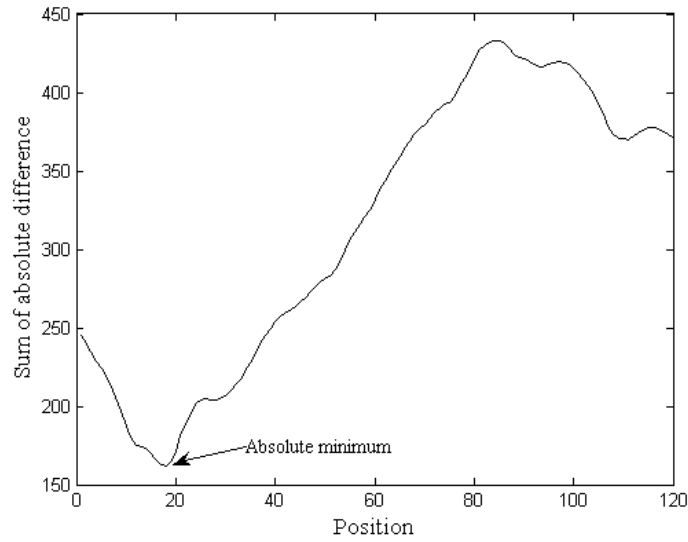
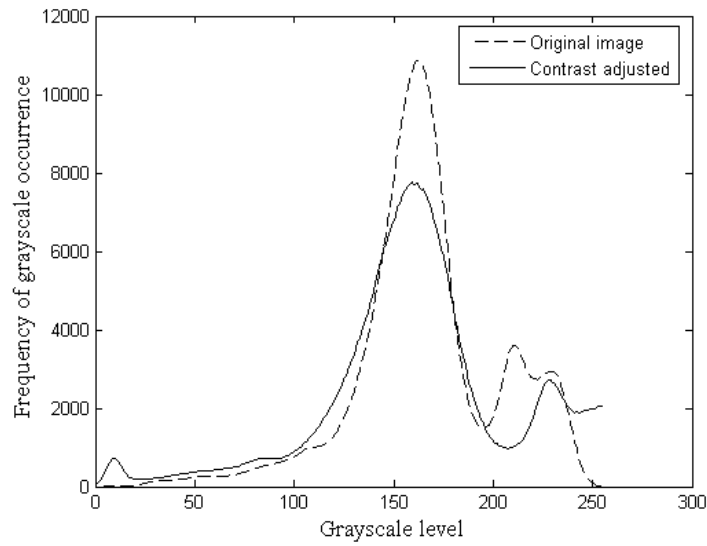
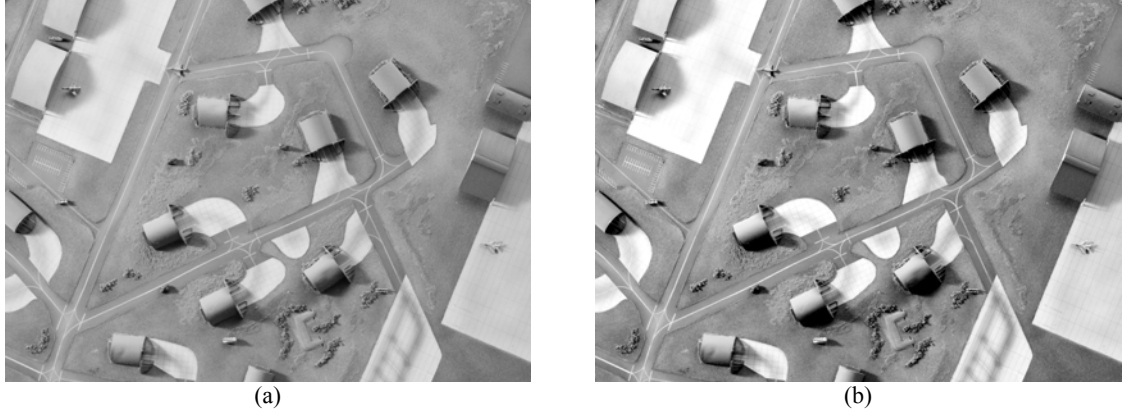


Figure 16. A sample histogram differencing array. The result of each SAD calculation from the circular shift sliding operation is stored in an array. Next, the minimum of the SAD array is determined. This difference value is placed in the response plane at a location corresponding to the ROIs center pixel. Once all ROIs have been processed, thresholding the response matrix identifies the spatial locations of possible targets.

3.2.3 Contrast consideration

In the differencing method described above, we assumed that the contrast of the template was the same as the contrast of the ROI; however, realistically the contrast will change as a function of many factors (Table 1). Image contrast changes are represented in the histogram by a width change and possible clipping at the lowest and highest grayscale values (Figure 17). Because contrast is highly variable, processing hundreds of possibilities at each ROI is unrealistic due to the required computation time. Instead, a novel technique, which we term wobble, was incorporated into our algorithm.

The wobble operation is a stretch or compression of the histogram that mimics true changes in histograms when image contrast varies. By wobbling the width of the template histogram, it is possible to create a database (library) of contrast-adjusted



(c)

Figure 17. Effect of changes in scene contrast on image histograms. (a) The original image, which was considered to have optimal brightness and contrast. (b) Using Adobe Photoshop, the original image contrast was increased. (c) A histogram comparison plot of the original and contrast-adjusted histograms demonstrate expansion of the histogram due to increased image contrast. The right tail of the histogram that represents the number of white and near-white pixels increased as a consequence of contrast stretching.

histograms. We postulated that it is better to wobble at each ROI position and find matches some of the time versus using a template with fixed contrast that will likely differ from the image all of the time. At each ROI, prior to the sliding differencing operation, we randomly selected one wobbled histogram within a user defined range from the library for the comparison process. Even though any given wobbled histogram may not be an exact contrast match to the object, the dissimilarity might be negligible.

Additionally, since the object spans a number of computations, more than one opportunity exists to obtain an improved target match. Next, the resultant minimum value of the SAD array was stored in our response matrix as before.

To increase the probability of matching the contrast at any given ROI, we randomly selected multiple wobbled histograms, which resulted in multiple minima from the differencing arrays (Figure 18). Specific to our study, three wobbles were used in

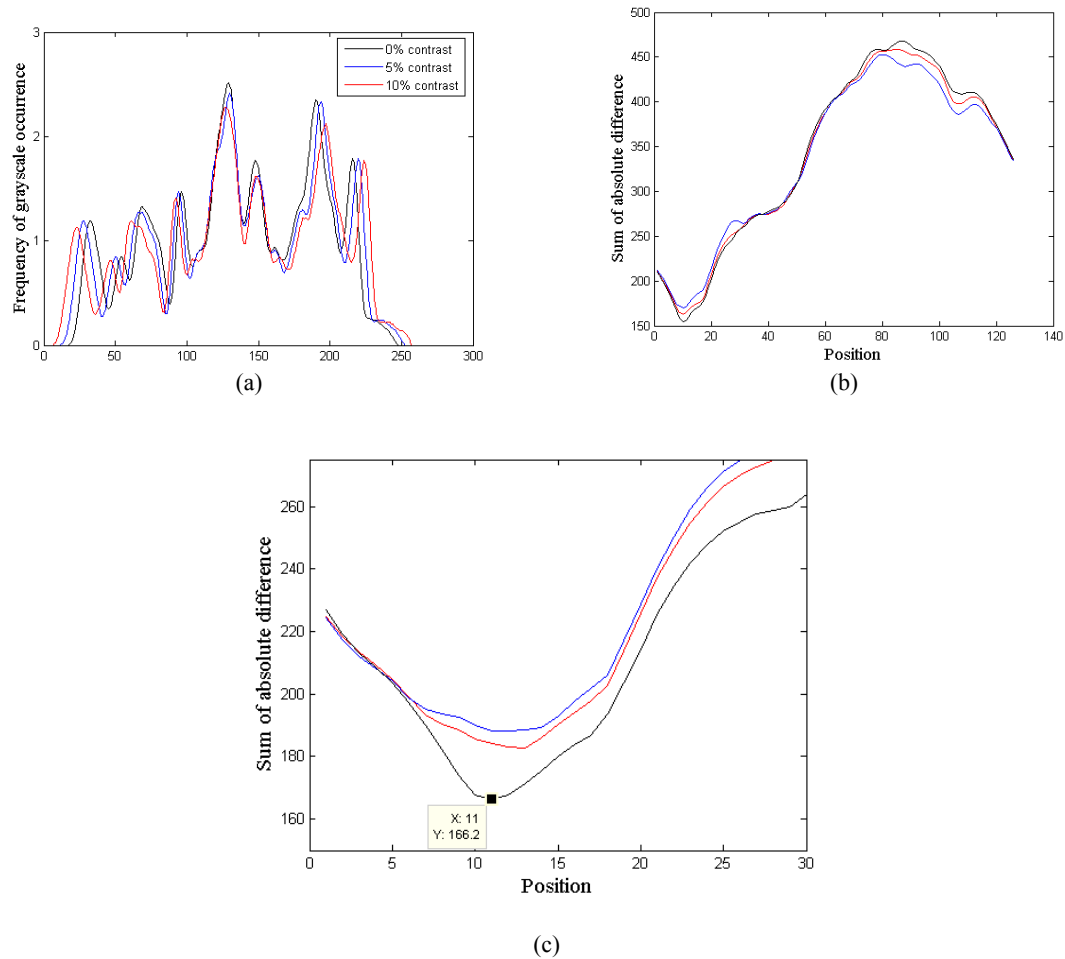


Figure 18. Multiple contrast-adjusted template histograms provide better opportunity for a target match. Wobbled, contrast-adjusted histograms from the template histogram library give us a set of SAD arrays from the differencing calculations. (a) A sample of three histograms at 0% (no wobble), 5% wobble and 10% wobble. (b) Multiple differencing arrays from calculations using multiple wobbled histograms with the minima indicated within the rectangle. (c) The absolute minimum of 166 at position 11 represents the best match of the contrast-adjusted ROI.

testing that resulted in three SAD arrays. However, the number of wobbled histograms is largely determined by the available processing time. Selecting the absolute minimum value of the multiple differencing arrays to place in the response plane offered the best match of the wobbled template histograms to the ROI histogram.

3.2.4 Handling multiple minima

Instances of multiple local minima within a particular SAD array necessitated identification of a global minimum (Figure 19). The absolute minimum of the zero-crossings taken from the derivative of each differencing array provided a best-case minimum result and was placed in the response plane.

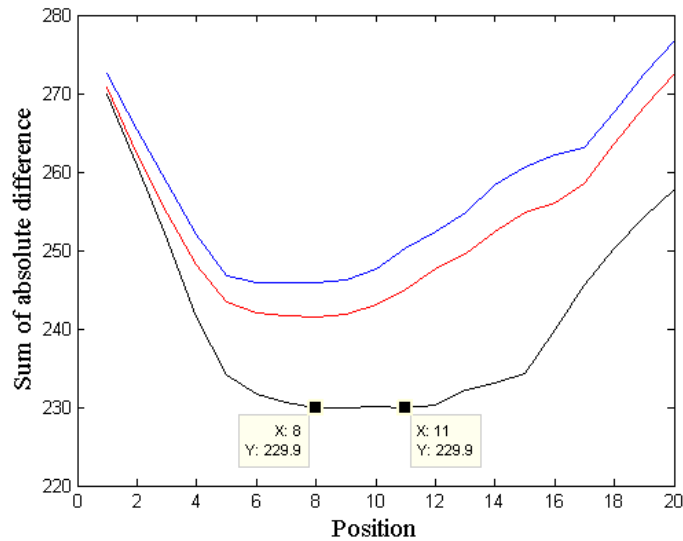


Figure 19. Multiple minima occurrence in SAD arrays. In one or more of the SAD arrays, multiple minima can occur. Duplicate SAD arrays result when the random selected wobbles are equal in two or more trials. The multiple minima shown here occur when the same array offers duplicate minima. For consistency, in these instances, the local minimum at the lowest numbered position (left-most position) was chosen each time duplicate minima occurred for method consistency.

3.2.4.1 Contrast Prediction

Initial testing and development of the wobbled sliding difference algorithm began with a training set of known contrast (C_{known}) images that were adjusted using the same method employed to generate the template library. When processing the training images, a small wobble range about the known contrast ($C_{\text{known}} \pm 5\%$) was selected for building the wobble library. Subsequently, an evaluation set of images having unknown contrast was used to verify the efficacy of the differencing algorithm in realistic scenarios.

3.2.4.1.1 Building a contrast prediction library

In order to select the narrow range of wobble desired, the image contrast was predicted by comparing a coefficient of variation-based (CV) contrast measure between the evaluation and optimum images with the following equation:

$$CV = \frac{\sigma}{\mu}, \quad (3)$$

where σ and μ are the ROI standard deviation and mean. A library of CV values, calculated from the training image set, became the library used for comparison. The CV library was generated by first finding the global image mean μ_{image} and the global image standard deviation σ_{image} to establish a set of boundaries for choosing the initial estimates of the best image regions for evaluation of image contrast where $\mu_{\text{range}} = \mu_{\text{image}} \pm 15\% \sigma_{\text{image}}$. Trial and error was used to determine that a 15% bound on this range was most appropriate. To soften edges and reduce the potential influence of speckle noise within the image, a 9 x 9 median filter was applied. The image was divided into adjacent 33 x 33 pixel blocks, or ROIs, for block processing. The block size was chosen based on the template size, such that the full target could fit completely within the block. Within each

ROI, the mean was computed, μ_{smooth} . Next, the CV at the ROIs where μ_{smooth} falls within μ_{range} was calculated and saved. An average CV , μ_{cv} , and CV standard deviation, σ_{cv} , of those selected ROIs was determined and used to set a new range CV_{range} for narrowing the candidate ROIs within $CV_{\text{range}} = \mu_{cv} \pm 15\% \sigma_{cv}$. From the new candidate ROIs, a new average CV_{contrast} was calculated and becomes the predicted contrast for that image.

Using a selection of images, incremented in steps of 5% contrast adjustment (e.g. -20%, -15%, -10%), from our known contrast training image set, the procedure for finding the predicted image contrast was repeated. The CV_{contrast} values and associated wobble range (percent of image contrast $\pm 5\%$) of each image was stored within a library to be used for predicting contrast of a new image having unknown contrast.

3.2.4.1.2 Predict image contrast

Once the CV library was established, the range of wobble for an image having unknown contrast could be predicted by repeating the procedure for finding CV_{contrast} . The CV obtained was matched to those stored in the CV library where the associated wobble range was collected as the optimum for processing the image.

3.2.5 Case Studies

Thirty-six training images and thirty-six evaluation images were used to develop and test our algorithm (Table 4). The training set of images having known contrast was generated from a single test image wherein the wobble algorithm was applied to the histogram of the image in order to simulate contrast changes. From the contrast-adjusted images, a sample set of thirty-six images was selected. In addition to the training set, a

different evaluation set of images was selected with three varying light angles and three unique board scenes.

To evaluate the effectiveness of the wobble method, Cases I and II applied the differencing algorithm with and without the wobble, respectively, to the training images having known contrast. In Case III, a new template was extracted from each of the evaluation images and processed with no wobble to replicate an optimum situation. Theoretically, the template would generate a perfect match; however, the background pixels in each ROI prevent an exact match. Cases IV and V applied the differencing algorithm with and without wobble, respectively, to the evaluation set of images having unknown contrast.

Table 4. Training and evaluation case studies. Five image sets were constructed to determine the effectiveness of our histogram differencing method. The first three cases represent our training data set. The conditions for Case I include an optimal image and template contrast with no wobble applied to the template histograms. In Case I, a new template was extracted from each image as a best-case scenario. In Cases II and III, the images were subjected to a non-optimal but known image contrast that was generated using our contrast adjusting algorithm, or wobble. Case II had an optimal template contrast with no wobble applied. Case III considered an adjusted template contrast with wobble. Cases IV and V comprised our evaluation data set, wherein the contrast was not known a priori and was non-optimal (similar to what would be presented in a real-world scenario). With Case IV, the same template histogram of optimal contrast was used with no wobble. Finally, with Case V, our method of contrast prediction enabled the range of wobble at each ROI to be selected from the template library.

Case	Image Contrast	Template Contrast	Wobble
<i>Training Set</i>			
I	Optimal	Optimal	No
II	Not Optimal; Known	Optimal	No
III	Not Optimal; Known	Adjusted	Yes
<i>Evaluation Set</i>			
IV	Not Optimal; Unknown	Optimal	No
V	Not Optimal; Unknown	Predicted	Yes

3.2.6 Algorithm evaluation

To test the effectiveness of the histogram differencing method of target detection, our laboratory developed an evaluation technique based on receiver operating characteristic (ROC) analysis. The ROC analysis compared the response plane obtained from the histogram differencing method to the ground truth. In addition, we developed a masking graphical user interface (GUI) tool that was used for selecting actual targets, or ground truth, within a test image. The masked areas become the true positive (TP) template. The test image was divided into a grid of uniformly shaped squares to which the TP template was superimposed. The estimated true negative template was found by excluding the squares where the TP template overlapped the background and non-target areas. Next, the TP template was applied to the response plane, and the number of targets actually located was computed to arrive at a measure of the method's sensitivity. The located targets were removed, and the remaining response plane was superimposed with the TN template. Squares that contained incorrectly classified non-targets were calculated and method specificity was determined. These steps were repeated at each grayscale (decision) threshold. The ROC curve was plotted as the sensitivity versus 1-specificity across the grayscale thresholds. Using the trapezoidal rule of integration, the area under the ROC curve (AUC) was calculated as the effectiveness metric of our detection algorithm [3].

The AUC of our ROC curve quantitatively represents how close to perfect (all targets and non-targets correctly categorized) the evaluated method of target detection performed. In testing the efficacy of histogram differencing, the AUC is the probability

that a randomly selected ROI from a target region will result in a better match to the template than a randomly selected ROI from a non-target region [23]. “The operating points on the ROC curve are generated under fixed decision thresholds for all detection algorithms,” offering consistent comparisons of all response planes from the different target detection methods under investigation within the larger scope of the project. [3]. The final decision threshold must be determined by the user, based on the calculated sensitivity and specificity combination at each threshold and the perceived or known situational risks. If the cost of false positives and false negatives are quantified a priori, then cost analysis is a feasible and impartial approach for selection of to selection of the final decision threshold.

4. RESULTS

4.1 Response plane comparison

The following figures allow a visual assessment and comparison of the histogram differencing target detection response planes obtained with and without wobble, according to the cases described in Table 4.

4.1.1 Case I: Optimum image

In Case I (Figure 20) image (a) has optimal contrast; the response plane (b) is obtained using templates extracted from the optimum image (optimal template) with no wobble applied. Regions of red identify the highest difference within the response plane,

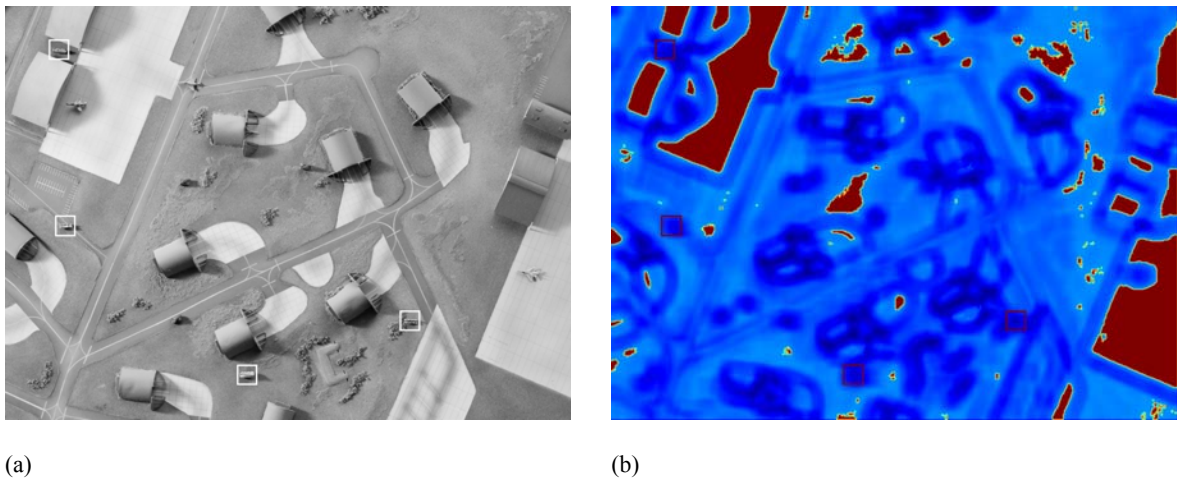
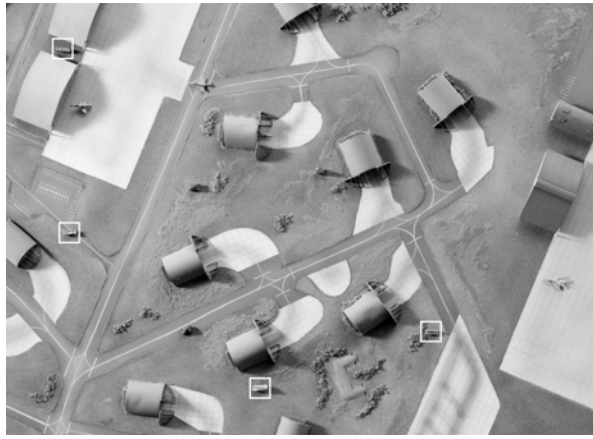
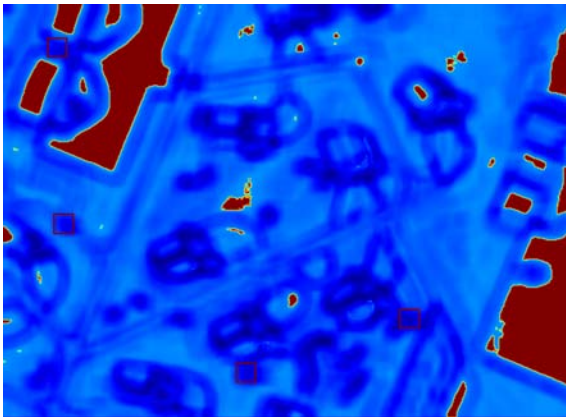


Figure 20. Case I: Optimum terrain board and histogram differencing response plane images. (a) Our selected best-case image contains identified targets that have spatially corresponding markers in (b) the response plane image generated using the optimum template, without wobble. With the template extracted from (a), the best-case scenario is anticipated and achieved, as demonstrated by the dark regions within the target markers. Even in this optimum case, many additional areas that do not contain true targets indicate potential targets and are deemed false positive regions. The area under the ROC curve for Case I was evaluated as 0.74.

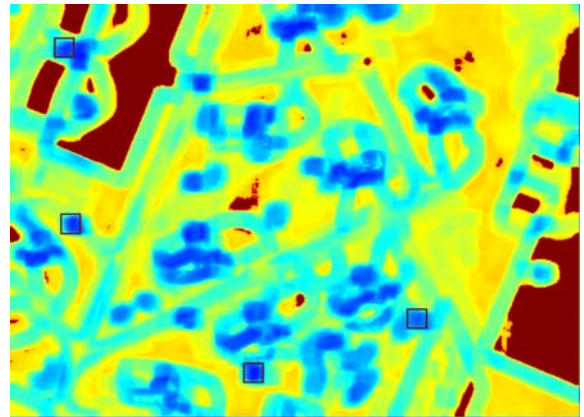
whereas dark blue areas suggest potential targets. Existing targets were flagged, but there is a substantial amount of FP regions. The edges of hangars and grayscale gradient areas from grass to cement are falsely identified as potential targets, because the histograms of these ROIs contain similar proportions of white and light-to-medium gray values as in the Scud template image. As evidenced by homogenous patches of dark red, regions of ‘no match’ were identified in this image.



(a)



(b)



(c)

Figure 21. Case II and III: Increased contrast terrain board and histogram differencing response plane images. (a) A terrain board image with an artificially applied 10% contrast increase and its response planes obtained from histogram differencing under (b) Case II, no wobble and (c) Case III, wobble conditions demonstrate the successful identification of the four targets. The areas under the ROC curves for (b) and (c) were determined to be 0.721 and 0.918, respectively. Simply viewing the response planes suggests that in (b), a large number of regions are falsely identified as potential targets as signified by the large areas of blue, where blue represents the smallest difference. The large number of false positives reduces the AUC. An improvement in the area under the ROC curve was seen when wobble was used; the large areas of blue from (b) were reduced, representing a reduction in the number of false positive regions.

4.1.2 Cases II and III: Non-optimal and known image contrast

The optimum image, or best-case image, underwent contrast adjustment based on a non-optimal but known image contrast using our wobble algorithm in order to generate a training image set for Cases II and III. For Case II, an optimal template contrast was used, and no wobble was applied. In comparison, Case III made use of an adjusted template contrast and applied wobble. If our hypothesis is correct, we expect Case III (with wobble) to outperform Case II (no wobble) when the contrast of acquired imagery is suboptimal. In Figure 21, large regions of the Case II response plane are blue in color, indicating low specificity; however, a reduction in the number of blue regions is seen with the Case III response plane. Unlike its non-wobbled analog, fewer regions appear dark blue in color indicating that wobble reduced the number of falsely identified potential targets. A closer look at an area within the terrain board image containing a single target reveals an interesting result after thresholding the response planes. Figure 22 presents the effects of three near-sequential threshold steps of corresponding regions selected from the Case II and Case III response planes. Although the target was identified at each threshold in both cases, in the non-wobbled instances, a large number of falsely detected regions lead to a lower ROC analysis AUC as presented in Figure 23. With the single exception of an extreme contrast compression (-70%), shown in Figure 24, the wobble method outperforms the non-wobbled method for all evaluated images.

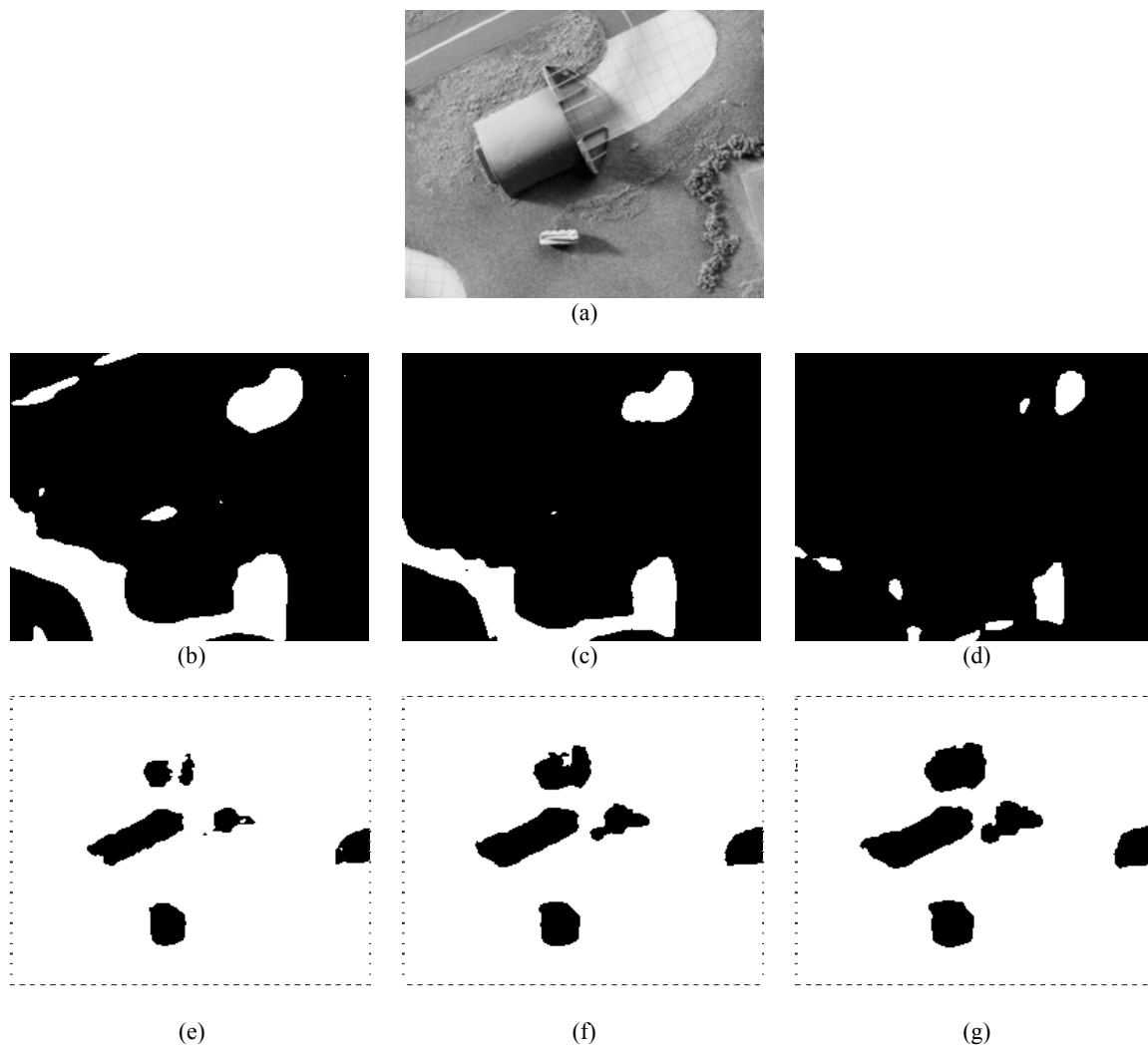
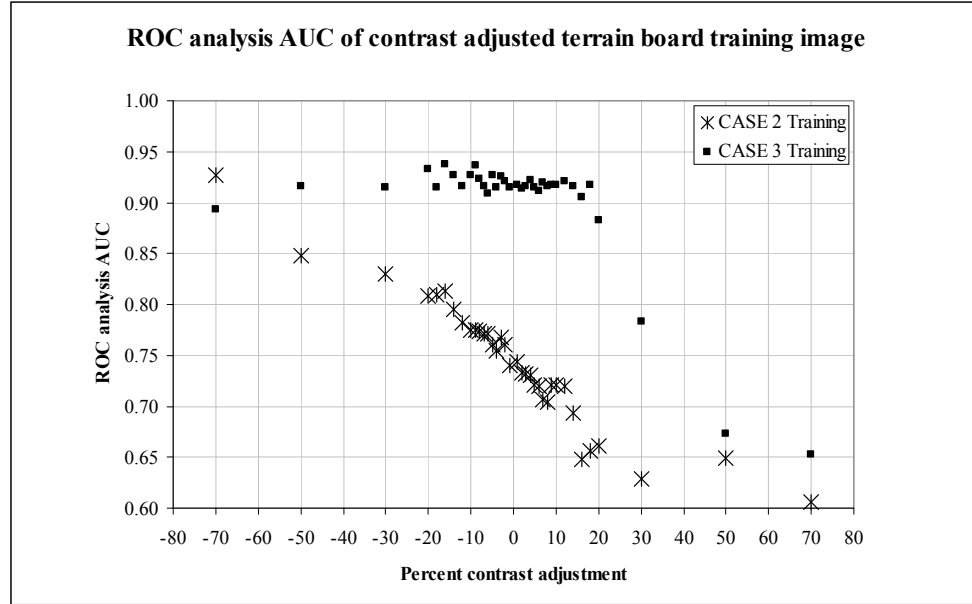
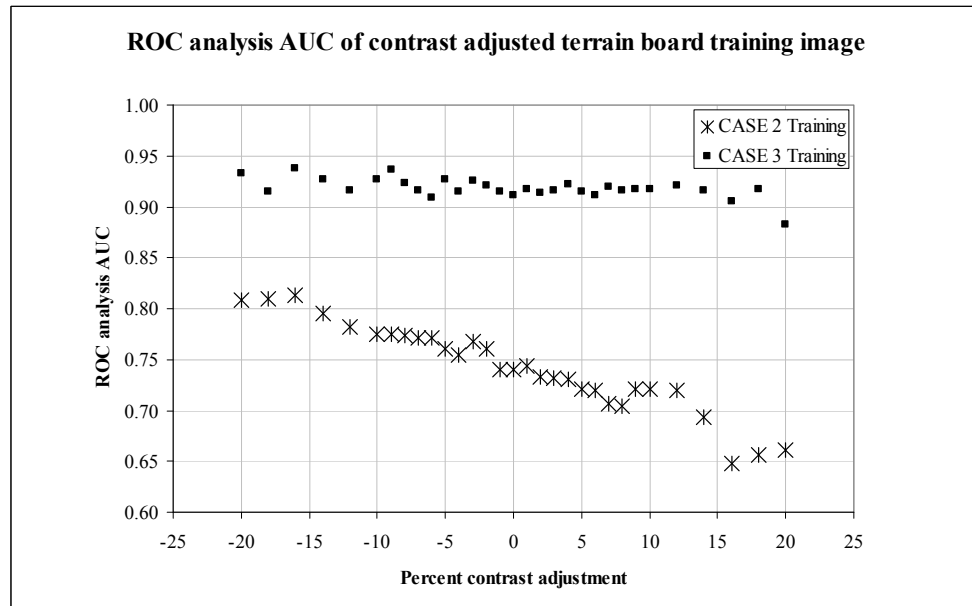


Figure 22. Threshold comparison of target and non-target regions with and without wobble. (a) A region selected from a training image of known contrast contains one target and background. Identical regions collected from the response planes generated from Case II and Case III were threshold at grayscale levels of 56 (b,e), 60 (c,f), and 64 (d, g). Here, a black output in the binary images corresponds to suprathreshold pixels. The targets were detected and identified in all 6 threshold response planes; however, the majority of the background is falsely identified as a potential target in the Case II response planes. In contrast, the Case III response planes segregate the target from the background and display fewer regions of falsely detected potential targets. In each of the six threshold response planes, an increase in the number of target-identified pixels occurs as the grayscale threshold level increases.



(a)



(b)

Figure 23. ROC analysis results for Cases II and III. (a) The ROC analysis AUC values for the 36 artificially contrast-adjusted training images from -70% to 70% contrast. In all but one instance (-70% contrast), the wobble method outperformed the no wobble method. In the failed case, the low contrast blurred the targets into the background such that they were difficult to discern. (b) A closer view of the AUC values in (a) within the range of -20% to 20% contrast adjustment, where we anticipate most real world images to exist. As supported by statistical analysis, the wobble method clearly outperforms no wobble.

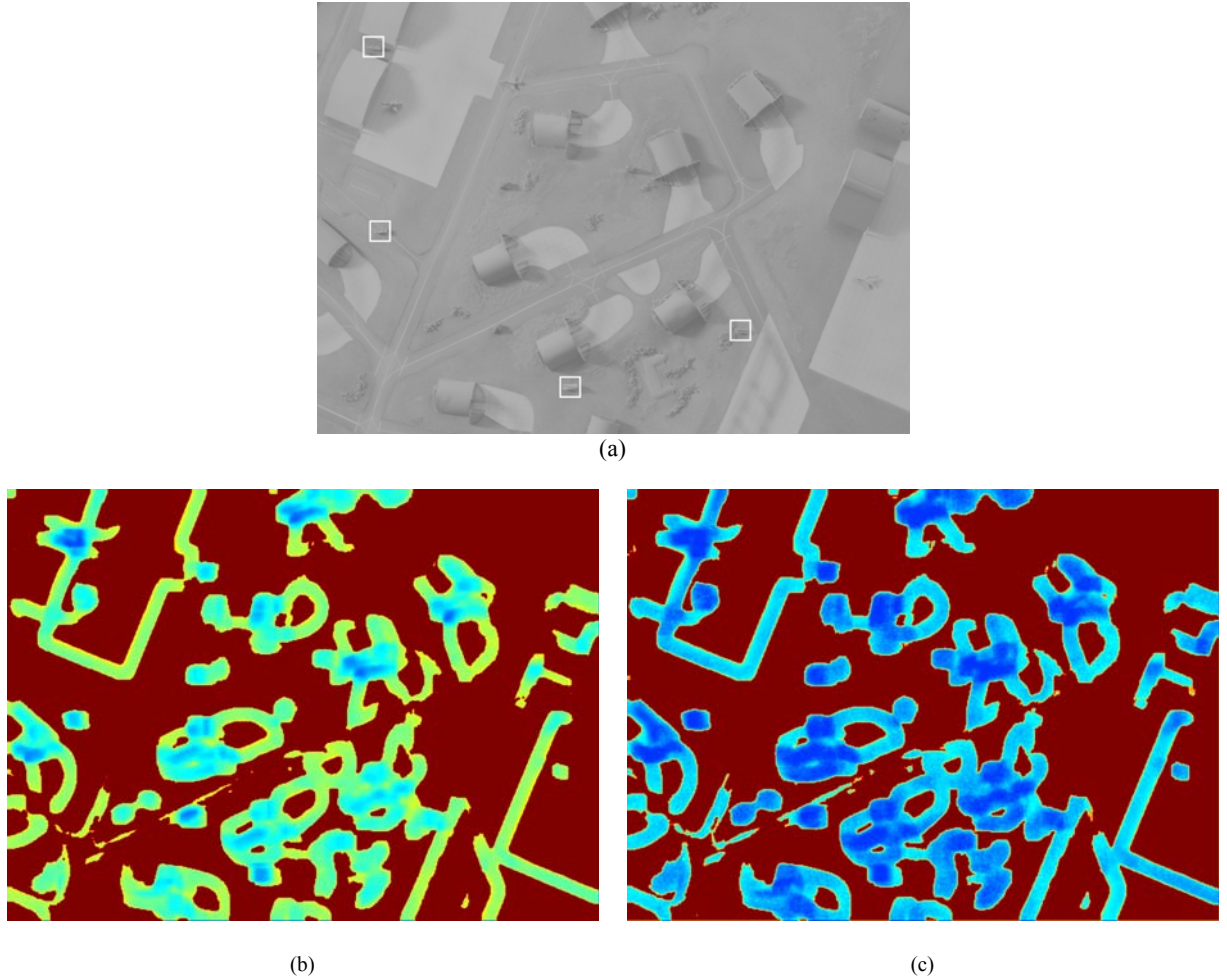


Figure 24. Case II and III response plane comparison of training image having -70% contrast. (a) For the artificially -70% contrast adjusted training image, we obtained the response planes for (b) no wobble, Case II, and (c) with wobble, Case III. The ROC analysis AUC for Case II and Case III were 0.927 and 0.893, respectively. In this instance, no wobble preformed slightly better than wobble. The low contrast in this case reduced image gradients, making the targets difficult to discern from the background. Visually, it is easy to see that there are more falsely indicated targets in (c), as indicated by the larger blue regions, than are represented in (b). The higher number of false positives in (c) leads to a lower ROC analysis AUC. However, a better match for each of the four targets was made in Case III as indicated by the darker blue pixels.

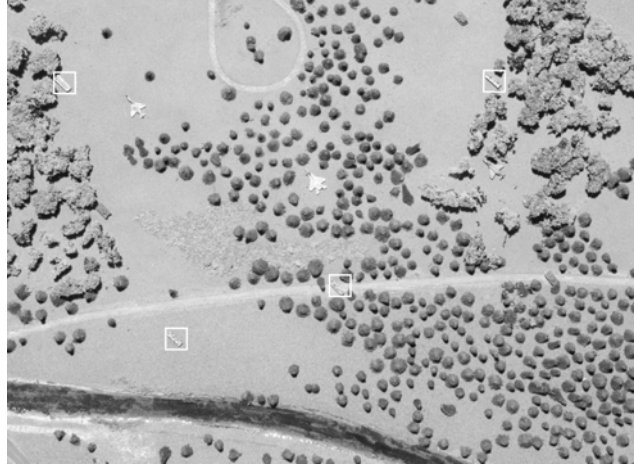
4.1.3 Cases IV and V: Non-optimal and unknown image contrast

Three different terrain boards (Figure 25) containing varying amounts of deciduous trees, paved and unpaved roads, ground cover, building structures, and military vehicles comprised our evaluation image set having non-optimal and unknown image contrast. For Case IV, an optimal template contrast was used, and no wobble was applied. In comparison, Case V made use of an adjusted template contrast and applied wobble,

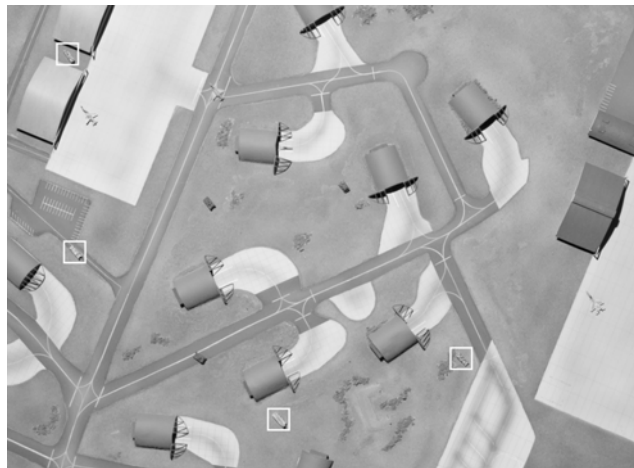
based on the predicted image contrast. The results of Case IV versus Case V testing were analogous to those of Case II and Case III. Case IV contained larger regions of falsely identified potential targets than did Case V (Figure 26). When the majority of the terrain board contained vegetation (Figure 26 e, f), the resulting response planes using predicted contrast and a wobbled template showed fewer regions of dark blue than those containing a proportionate amount of vegetation and man-made structures because these large homogenous regions were eliminated by the histogram width criterion.

The plot in Figure 27 presents the ROC analysis AUC for each evaluation image response plane, categorized by terrain board. Terrain Board (TB) 1, as represented in Figure 24, contains an even distribution of deciduous trees, even-toned ground cover, unpaved roads, and military vehicles. TB2 models a more urban scene with its building structures, paved roads, and fewer background trees. A large portion of TB3 contains even-toned ground cover, unpaved roads, and only a few areas containing trees. The results presented in Figure 27 allow a glimpse into the effectiveness of the histogram differencing algorithm under differing terrain conditions.

TB1



TB2



TB3

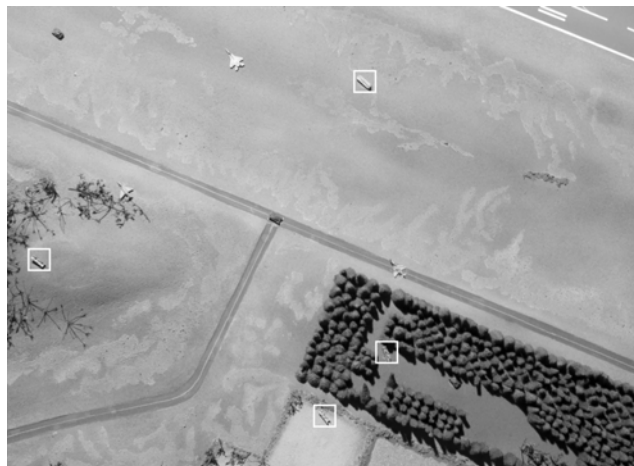


Figure 25. Three different terrain board examples from the evaluation image set. Each of these terrain board images from the evaluation image set demonstrates unique traits that affect the outcomes of the ROC analyses, including deciduous trees, pavement, ground vegetation, and military structures and vehicles.

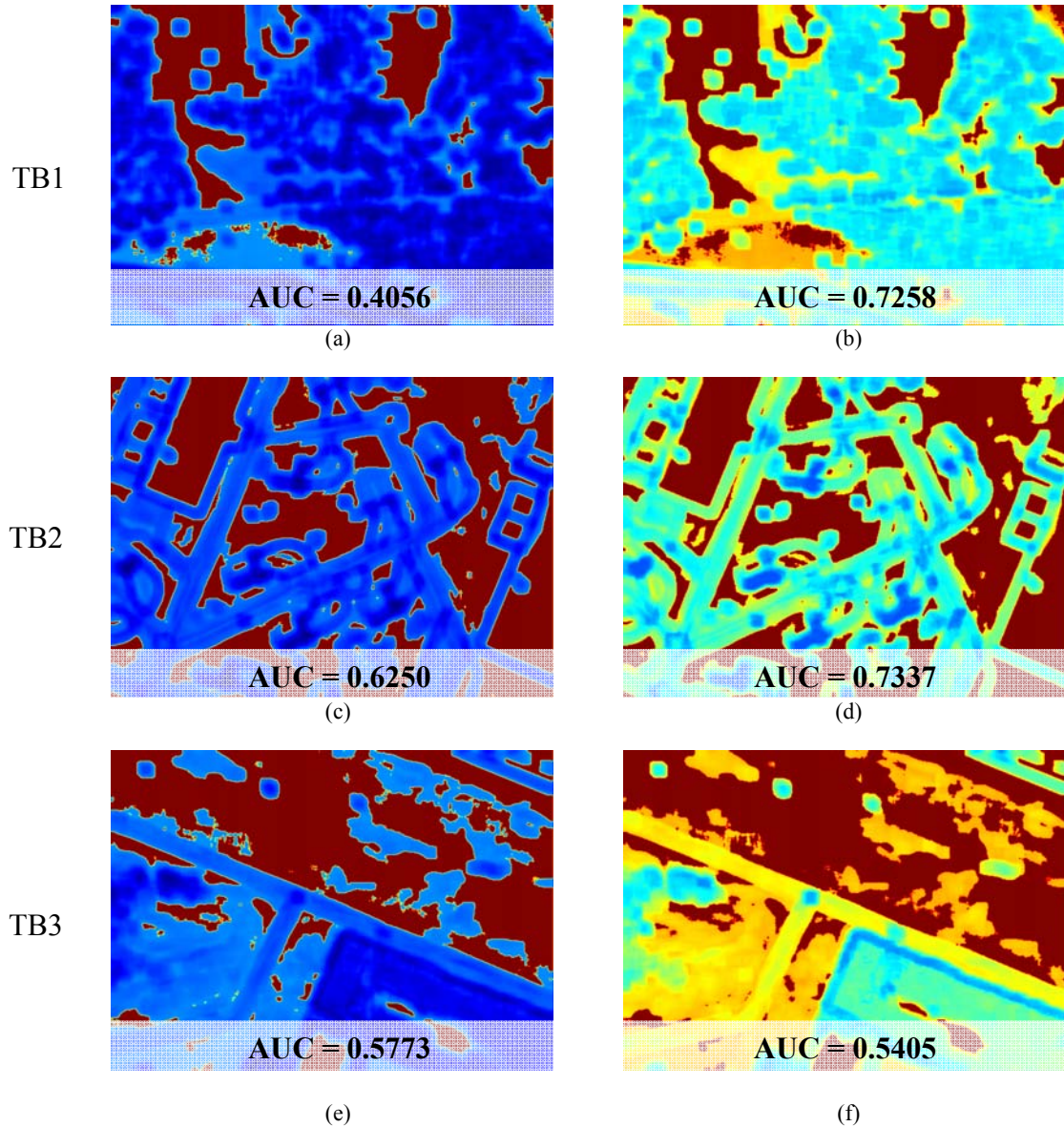


Figure 26. Case IV and V histogram differencing response planes from three unknown contrast terrain board images. The terrain board images from Figure 25, having unknown contrast, were processed with the histogram differencing algorithm under (a, c, e) Case IV, no wobble and (b, d, f) Case V, predicted contrast with wobble. The four targets were successfully identified. The areas under the ROC curves for TB1 response planes (a) and (b) were determined to be 0.4056 and 0.7258, respectively. Similar results for TB2 response planes (c) and (d) resulted in an area under the ROC curve of 0.6250 when wobble was not used, and an improved with wobble with the area under the ROC curve as 0.7337. Less than significant improvement was seen in the areas under the ROC curves for TB3 histogram differencing response planes, as in the examples (e) and (f), where the no-wobble and with wobble results were 0.5773 and 0.5405, respectively. As in Figure 21(b), we again see in (a, c) that a large number of regions are falsely identified as targets, which reduces the AUC. The AUC increased when wobble was used but, as evidenced in (b, d) by lighter shades of blue, the targets are identified with less confidence and the AUC values are reduced compared to the results as found in Figure 21(c) (wobble results on non-optimal, but known, contrast-adjusted image).

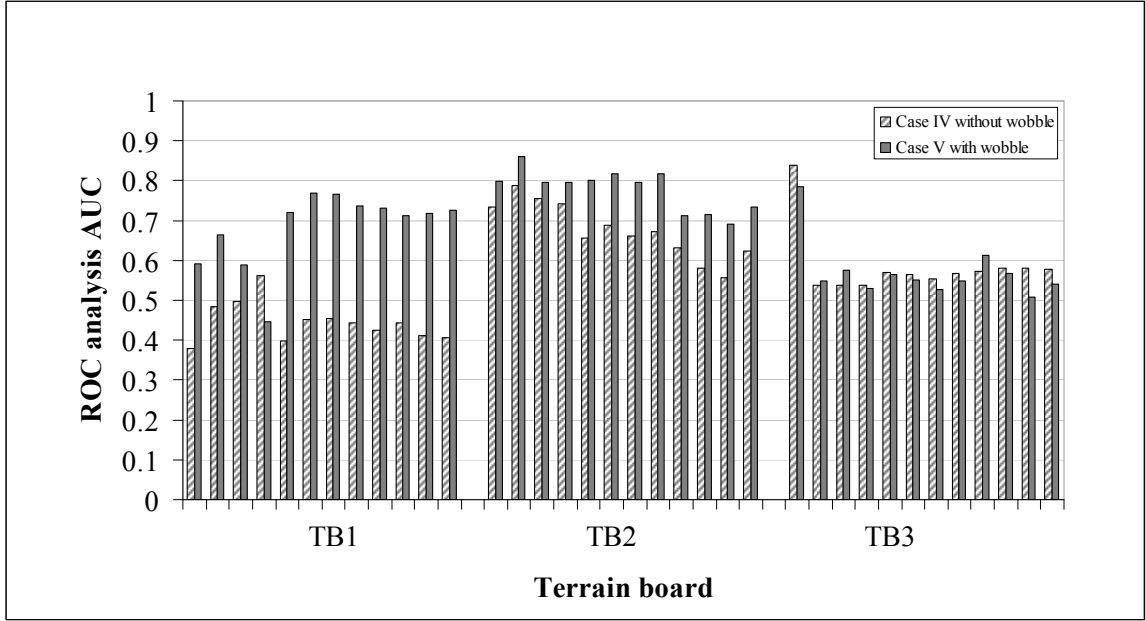


Figure 27. ROC analysis for Case IV and V. In the cases where the image contrast was unknown, our method performed best on two of the terrain board evaluation image sets, TB1 and TB2. Since the algorithm was optimized using a best-case image found on TB2, a successful result in TB2 images was anticipated. TB1, containing large regions of deciduous trees, was also analyzed successfully. However, when the results from processing TB3 images, containing roads, large flat and grassy areas, and a small number of deciduous trees, were analyzed, the individual image results where no wobble was used (Case IV) outperformed the individual image results where wobble was used (Case V) in 9 out of 12 of the images, although the improvement was modest in each case. To evaluate the significance of each terrain board, a t-test was ultimately performed. A t-test for each terrain board revealed that the difference with wobble for TB1 and TB2 were highly significant ($p < 0.05$) having a p-value of 1.457×10^{-07} and 4.792×10^{-4} , respectively, and the difference in using wobble for TB3 was not significant having a p-value of 0.6718. TB3 contains roads, large flat and grassy areas, and a small number of deciduous trees.

4.2 Statistical analysis of method efficacy

Was there a true difference in the effectiveness of our target detection method by using or not using wobble? This question lead to the following null hypothesis (H_0): there was no statistical difference between ROC analyses AUCs of the differencing algorithm with and without wobble. With this hypothesis in mind, a new set of evaluation images was generated by applying a +12% contrast adjustment to the 36 evaluation images. The results of histogram differencing with and without wobble, categorized by TB, are presented in Figure 28. The trend for TB1 and TB2 (Figure 28) demonstrated that wobble outperformed no wobble. In the case of TB3, the results appear to be similar for both methods with the average areas under the ROC curves for wobble and no wobble of 0.68

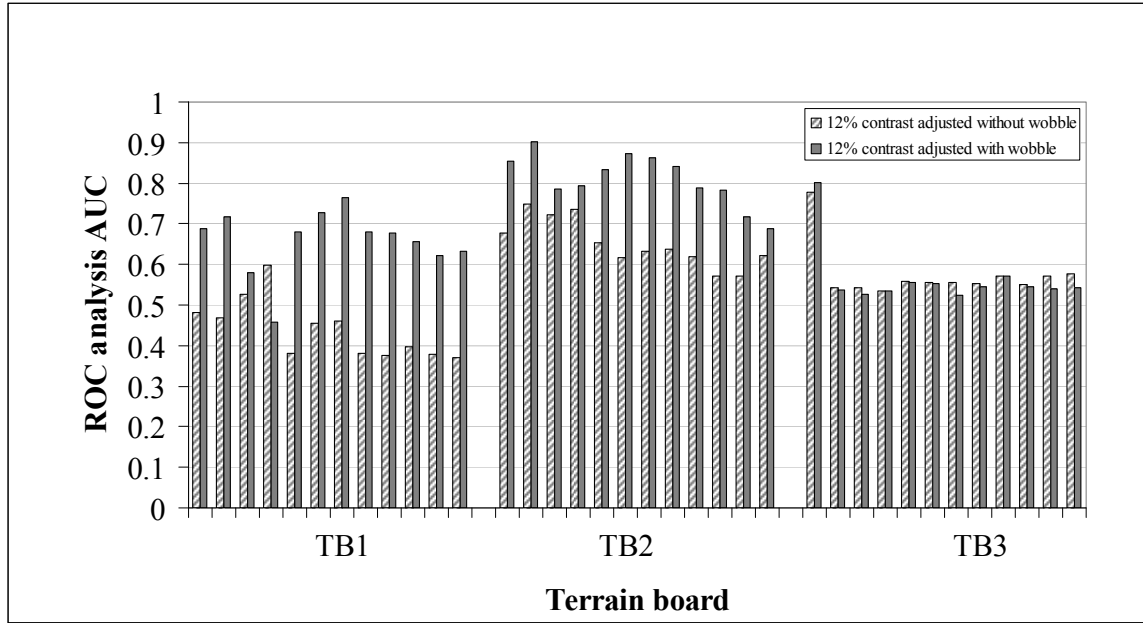


Figure 28. ROC analysis results for +12% contrast adjusted evaluation image set. Similar results occurred with the images having +12% contrast adjustment applied as with Cases IV and V (Figure 27). The wobble method outperformed the no-wobble method in all but one of the TB1 and TB2 response planes, but the results for TB3 were essentially equivalent. As with Cases IV and V with regard to TB3 response planes, the data produced by the two methods were ultimately subjected to individual TB t-tests to test for significant difference.

and 0.57, respectively. The ROC data from each TB individually, as well as the combined results were subjected to a t-test as a final test of method effectiveness. For the combined TB analysis, we have equal sample sizes and an unknown population variance. The small sample sizes (less than 40 images per set) required the use of a t-test. The same image set was used for both cases, resulting in two sample data sets, wobble and no wobble. The sum, mean, variance, and standard deviation were calculated for each data set (Table 5),

Table 5. Dataset statistical values used in t-test. Dataset statistical values used in t-test to assess significance of difference between the wobble method and the no wobble cases for the combined +12% contrast adjusted image RPs. The sum, mean, variance, and standard deviations were calculated from the original datasets, and these values were used to determine the pooled standard deviation, degrees of freedom, t-statistic, and the t-critical points for our two-tail, two sample, t-test that assumed equal variance

Dataset statistical calculations		
	<i>No wobble</i>	<i>Wobble</i>
Sum	19.96	24.38
Mean	0.55	0.68
Variance, s^2	0.01	0.02
Standard deviation, s	0.11	0.12

and a pooled standard deviation was then calculated using the following equation:

$$\sigma_p = \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{n_1 + n_2 - k}}, \quad (4)$$

where n_1 and n_2 were the number of observations in each data set, σ_1 and σ_2 were the standard deviation of each data set, and k represented the number of sample data sets, in this case $k = 2$. The pooled standard deviation was used because the pair of ROC analysis AUC data sets was considered to be generated from the same model. This allowed the pooling, or combining, of the standard deviations as an estimate [24]. Using the information from Table 5, the resulting σ_p was 0.1173. The degrees of freedom df for estimating variability, calculated with the following equation:

$$df = n_1 + n_2 - k, \quad (5)$$

was found to be 70. To test the equality of sample variance s , the following chi-squared distribution equation was used:

$$x_0^2 = \frac{df \times s_{wobble}^2}{s_{no\ wobble}^2}. \quad (6)$$

The result from Equation (6) was 45.88, which falls within the chi-squared test statistic of ± 53.16 . It was concluded that the equality of variance could not be proven; thus, unequal variance for the t-test was assumed. Next, the test statistic, t_{stat} was calculated from the following equation:

$$t_{stat} = \frac{\mu_1 - \mu_2}{\sigma_p \sqrt{\frac{2}{n}}}, \quad (7)$$

where μ_1 and μ_2 represented the mean of each data set as listed in Table 5. The final t_{stat} resulting from Equation (7) was -4.44. An estimated percentage point of the two-tailed t-distribution from tables [23] set our $t_{critical}$ to ± 1.99 . Because $t_{stat} \ll t_{critical}$, we reject H_0 and accept the alternative hypothesis (H_1) that the results using wobble are significantly different from those using no wobble. Table 6 provides the Microsoft[®] Excel (Version 2003, service pack 3, Microsoft Corporation, Redmond, WA) results of this statistical analysis as verification of the calculations performed without statistical analysis software.

Table 6. Data analysis of t-test images having +12% contrast adjustment. As evidenced by a two-tailed t-test, the ROC analysis AUCs of the +12% contrast adjusted images' response planes, processed using wobble versus no wobble, were significantly different, with a p-value of 0.000034. This p-value is considered highly significant ($p < 0.05$).

t-Test: Two-Sample Assuming Unequal Variances		
	<i>CASE II t-test</i>	<i>CASE III t-test</i>
Mean	0.55	0.68
Variance	0.01	0.02
Observations	36	36
Hypothesized Mean Difference	0	
df	70	
t Stat	-4.44	
P(T<=t), two-tail	3.36E-05	
t Critical, two-tail	1.99	

4.2.1 t-Test individual terrain boards

While evaluating the area under the curve from the ROC analysis of the evaluation image response planes and those of the +12% contrast adjustment image

response planes, some discussion of individual terrain board results arose. The ROC analysis of the TB1 and TB2 response planes appeared to show significant results when wobble was applied, but it was not clear if wobble or no wobble was a better choice when evaluating TB3. To statistically evaluate the results, a t-test for each terrain board ROC analysis result was performed using Microsoft® Excel.

The results of the evaluation image sets from TB1 and TB2 from Case IV and Case V (Table 7) demonstrated the improvement of true target detection when wobble was applied to the differencing algorithm, with highly significant ($p < 0.05$) p-values of 1.5×10^{-7} and 4.8×10^{-4} , respectively. There was no difference between the results with and without wobble for TB3 (p-value = 0.67). Similarly, the resulting terrain boards of the image sets having +12% artificially altered contrast from TB1 and TB2 demonstrated the improvement of true target detection when wobble was applied to the differencing algorithm with highly significant ($p < 0.05$) p-values of 4.9×10^{-7} and 2.1×10^{-6} , respectively. In contrast, analysis of the ROC results for TB3 showed no statistically significant

difference when wobble was used (p-value = 0.75) (Table 8). Although the ROC results for TB3 with wobble were not found to be statistically significant from those without wobble, using wobble provides visual information and a better opportunity to identify targets. Comparison of Figures 26 e and f shows that wobble provides more segregation of target from non-target regions than in the RP without wobble where the targets fall within large FP regions. These segregated areas are identified with a lower confidence level as indicated by the lighter shade of blue, but they are more readily observed than those found in the no wobble RP.

Table 7. Individual TB data analyses of evaluation images having unknown contrast. Using a two-tailed t-test, the ROC analysis AUCs of the response planes resulting from images having unknown contrast processed using wobble versus no wobble resulted in p-values of 0.00000015 from TB1, 0.00048 from TB2, and 0.67 from TB3. We reject the null hypotheses for TB1 and TB2 and concluded that wobble improves our outcome. The t-test results for TB3 were not significant, as expected from the evaluation of Figure 27; we concluded that there is no difference between using and not using wobble when analyzing TB3 images.

TB1 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE IV Evaluation</i>	<i>CASE V Evaluation</i>
Mean	0.45	0.68
Variance	0.00	0.01
Observations	12	12
Pooled Variance	0.01	
Hypothesized Mean Difference	0	
df	22	
t Stat	-7.57	
P(T<=t), two-tail	1.46E-07	Highly significant
t Critical, two-tail	2.07	

TB2 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE IV Evaluation</i>	<i>CASE V Evaluation</i>
Mean	0.67	0.78
Variance	0.00	0.00
Observations	12	12
Pooled Variance	0.00	
Hypothesized Mean Difference	0	
df	22	
t Stat	-4.09	
P(T<=t), two-tail	4.79E-04	Highly significant
t Critical, two-tail	2.07	

TB3 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE IV Evaluation</i>	<i>CASE V Evaluation</i>
Mean	0.59	0.57
Variance	0.01	0.01
Observations	12	12
Pooled Variance	0.01	
Hypothesized Mean Difference	0	
df	22	
t Stat	0.43	
P(T<=t), two-tail	6.72E-01	Not significant
t Critical, two-tail	2.07	

Table 8. Individual TB data analyses of t-test images having +12% contrast adjustment. With a two-tailed t-test, the ROC analysis AUCs of the response planes resulting from images having +12% contrast adjustment processed using wobble versus no wobble resulted in p-values of 0.00000049 from TB1, 0.0000021 from TB2, and 0.75 from TB3. We rejected the null hypotheses for TB1 and TB2 and concluded that the wobble results were significantly different from the no wobble results. The t-test results for TB3 were not significant, as expected from the evaluation of Figure 28, and we accepted the null hypothesis with the conclusion that there was no difference between wobble and no wobble for TB3 images.

TB1 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE II t-test</i>	<i>CASE III t-test</i>
Mean	0.44	0.66
Variance	0.01	0.01
Observations	12	12
Pooled Variance	0.01	
Hypothesized Mean Difference	0	
df	22	
t Stat	-7.01	
P(T<=t), two-tail	4.90E-07	Highly significant
t Critical, two-tail	2.07	

TB2 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE II t-test</i>	<i>CASE III t-test</i>
Mean	0.65	0.81
Variance	0.00	0.00
Observations	12	12
Pooled Variance	0.00	
Hypothesized Mean Difference	0	
df	22	
t Stat	-6.37	
P(T<=t), two-tail	2.09E-06	Highly significant
t Critical, two-tail	2.07	

TB3 t-Test: Two-Sample Assuming Unequal Variances

	<i>CASE II t-test</i>	<i>CASE III t-test</i>
Mean	0.57	0.56
Variance	0.00	0.01
Observations	12	12
Pooled Variance	0.01	
Hypothesized Mean Difference	0	
df	22	
t Stat	0.33	
P(T<=t), two-tail	7.47E-01	Not significant
t Critical, two-tail	2.07	

5. DISCUSSION

5.1 Method evaluation

In this study, we demonstrated the efficacy of a histogram-based template matching method to locate potential targets in digital images acquired under non-ideal conditions. The proposed method predicts the presence or absence of a known target object by comparing the histograms of image ROIs to those from a contrast-adjusted target template library.

5.1.1 Case I

For Case I, a best-case image having optimal contrast and a template extracted directly from the best-case image resulted in an ROC analysis AUC value of 74.0%. This case demonstrated the best match attainable with our histogram differencing method but without wobble. To achieve a 100% AUC, all targets must be marked as positives (a goal that was met) and all non-target regions must be marked as negatives (a goal that was not met), under all decision thresholds. In our best-case scenario, image ROIs containing deciduous trees or road edges had a histogram signature that was similar to that of the Scud missile launcher, so these areas were marked as potential target regions. Since imaging conditions are rarely ideal, particularly in military and medical scenarios, it was important to develop a brightness- and contrast-sensitive approach and to validate that

method by comparing the detection results on non-ideal images to the best-case image result.

5.1.2 Cases II and III

To determine if the wobble algorithm improved the ROC analysis outcome, we generated a training set of images by artificially altering the contrast of our best-case image in known amounts. The training image response planes generated using wobble illustrated marked improvement in the average ROC analysis AUC over those generated without wobble having values of 90.0%, and 74.3%, respectively. All targets were identified in both cases, but the wobble method considerably reduced the number of FP regions.

5.1.3 Cases IV and V

For the evaluation set of unknown contrast images, development of a contrast prediction algorithm provided a contrast estimate and offered the opportunity to narrow the range of wobbled histograms to use at each region of interest. Without contrast prediction, the random wobble range used to evaluate images having unknown contrast would span -90% to +90% wobble, which lowers the chances of correctly identifying targets. A test of prediction accuracy was performed wherein eight images having random, but known, contrast (-88%, -77%, -15%, -4%, +7%, +23%, +49% and +72% contrast) were evaluated and the appropriate wobble range predicted by our algorithm. In all but one image, the applied contrast adjustment was categorized within the correct wobble range, with the single exception of -77% that fell into its neighboring wobble range.

Results for the evaluation image set follow those of Cases II and III where wobble performed better than no wobble on two of the terrain board image sets. A significant difference between wobble and no wobble was not seen in the third terrain board image set. In the TB3 histogram differencing with wobble, the tree-line ROIs returned a smaller difference value when compared to those of the target ROIs. This resulted in a lower confidence level of the identified targets in Case V. Because the light blue-identified targets at high thresholds in the ROC analysis no longer contributed to the final outcome, the AUC was reduced.

5.1.4 Data analysis

In addition to comparing average performance, it was important to validate our wobble and contrast prediction results statistically. To each of the evaluation images of unknown contrast and varying landscapes, a +12% contrast adjustment was applied using our wobble algorithm; this became our image set for statistical evaluation. A +12% contrast adjustment was chosen based on the results from Case III where wobble within the range of $\pm 20\%$ returned consistent results. Algorithm testing with and without wobble (Figure 25) demonstrated an improvement when wobble was used on images of TB1 and TB2, but wobble outperformed no wobble on only 25% of TB3 images. As with Cases IV and V results for TB3, the better histogram match between the tree-line ROI and template histograms lowered the confidence level of the identified targets. Similarly, the final AUC was reduced because the targets did not contribute throughout the high thresholds of ROC analysis. However, the t-test of the ROC analysis of the three sets of response planes (from the combined set of TB1, TB2 and TB3 evaluation images) verified that the wobble results were significantly different from the no wobble results (Table 5). An

important observation was that all targets were identified in all response planes when wobble was applied. In those response planes where no wobble outperformed wobble, the histogram SAD calculated and displayed in those response planes was represented as a light blue that showed the target with less confidence. Visually, the targets were more readily identified within the wobble response planes.

5.1.5 Terrain board comparisons

Image regions of deciduous trees replicate the high contrast of regions containing man-made structures, such as aircraft hangars, because of the dark trees on the light grassy background; many of the tree-occupied regions return a false positive response. Under these conditions, the tree areas are not eliminated from histogram differencing because their histograms are sufficiently wide to meet the heterogeneity criterion. Increasing this width threshold could possibly eliminate these regions from consideration, but would potentially lead to missed targets. Instead, the goal was to capture all targets regardless of the number of false positive regions.

5.1.6 ROC evaluation

Our method to test the efficacy of target detection is ROC analysis. The ROC analysis algorithm was developed within our lab as a method of comparing the results of each target detection algorithm developed as part of our larger-scale project. Whereas using ROC analysis to validate each method maintains fairness, other methods exist that might offer more insight to the usefulness of the wobble algorithm. For instance, a visual identification test by a set of human observers could be used in conjunction with the results found in ROC analysis as a determinant of method efficacy. Other mathematical

testing methods used to evaluate detection methods include perturbation theory [25] and optimal matching [26].

5.2 Contrast prediction alternatives

In addition to the chosen method of contrast prediction used in processing the evaluation image set, a review of several contrast prediction methods offered a glimpse into different options that have been used previously by others [27]. Peli reviewed definitions of contrast, including those based on luminance of simple patterns, such as the Weber fraction, a periodic pattern like the Michelson formula, a ratio of luminance change to background in the Westheimer equation [28, 29], and a comparison of contrast between two different images using a root-mean-square (*rms*) contrast. In the *rms* method of contrast prediction, two images are identified as having the same contrast if the *rms* values are equal. Because the *rms* method is a comparison between two images, it presented itself as a viable option for predicting contrast in the histogram-based template matching algorithm, and this method was evaluated. Using the *rms* equation,

$$rms = \left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{1/2}$$

(4)

$x_i = a \text{ normalized graylevel between } 0 \text{ and } 1,$
 $\bar{x} = \text{the mean normalized graylevel}$
 $n = \text{number of values within ROI}$

the process required manual selection of an ROI in each new image that served as a comparison region to a stored library of *rms* values based on known contrast images. Selection of the ROI proved to vary the predicted value outside the best wobble range when tested using images having known contrast. Further, automation of contrast prediction was preferable for providing unbiased results. To automate the process, the

rms method was replaced with the coefficient of variation-based contrast measure (3) as described in Chapter 3.

5.3 Variable optimization

Algorithm processing times vary with computer hardware capabilities; presently, on an eMachine™ notebook (Mobile AMD Athlon™ 64 processor 3200+, 2.0 GHz and 512 MB RAM) with Microsoft Windows XP®, Home edition (Version 2002, service pack 2, Microsoft Corporation, Redmond, WA), the algorithm processes a 1076 pixel x 786 pixel image in approximately 2 hours without wobble and 5.5 hours with three wobble instances. To reduce computation time, with the ultimate goal of obtaining a near real-time response, several simultaneous strategies could be employed: excluding additional ROI's, increasing the sliding step size in the histogram differencing process, using parallel processing, and/or implementing hardware acceleration. At present, the code eliminates regions based on a histogram peak ratio of width to area under the curve. Choosing a wider histogram width for the elimination criterion could potentially increase the number of regions excluded from processing; however, the risk of target region elimination is also increased. In the algorithm, a sliding difference calculation is performed to make our target detection method insensitive to image brightness changes, but the sliding steps add to the overall processing time. Thus, increasing the step-size leads to a potential decrease in computation time. There exists a possibility that a decrease in accuracy and sensitivity would result from excluding certain ROI's from analysis or increasing the sliding step size; therefore, deciding what level of accuracy and sensitivity can be sacrificed in order to process at higher speeds becomes an important issue. Another option to obtain faster results is parallel processing. By dividing the image

into n smaller non-overlapping subimages and processing each of these images in parallel using n central processing units (CPUs), the computation times should be decreased by a factor of n . By further capitalizing on the redundancy of many of our processing commands, graphics processing unit (GPU)-based acceleration would provide appreciable efficiency increases.

When a new set of images and/or targets is to be evaluated using our target detection method, variables such as template size, bit-depth, template library size, and wobble iterations can be easily adjusted as needed. For example, the code is presently optimized for a 33 pixel by 33 pixel template based on the size of the Scud missile launcher found in our training images. However, if the target is an aircraft, the template size would increase according to the area occupied by the aircraft in the image. Though the image set used to test the algorithm initially contains 16-bit images, the bit-depth is converted to 8-bit within the algorithm since the 16-bit images provided an expansive and sparse histogram that offered a poor signature of our template Scud missile launcher. Other objects might offer better histogram signatures at alternative bit-depths. The size of the template library is determined entirely by project needs. In our case, the library contained 181 wobbled template histograms, which provided histograms to test images having contrast adjustments from $\pm 85\%$ with a $\pm 1\%$ wobble step size. Also, the library is incremented by 1% in our case, but could be reduced or increased as required by a particular situation. Finally, an increase in the number of wobbled histograms to randomly apply at each ROI (we used three instances) might improve the results by increasing the opportunity to obtain the best match, with the understanding that additional

computation time would be required. Each of these variables can be modified according to project needs and target specifications to improve the RP output.

5.4 Method comparison

5.4.1 Fit with overall project goals

When compared to the additional methods of target detection developed in our lab for the larger-scale project, the wobble algorithm offers lower a lower ROC AUC, in part due to large regions of falsely identified targets. This algorithm also requires a longer processing time than other methods. However, wobble performs well even when images of low contrast are presented. In comparison, when edge information is compromised in the low contrast images, the shape-based and Fourier transform-based methods would likely fail to identify some or all targets. Whereas these methods are edge-based, the histogram-based method captures a unique attribute of the target object – its grayscale distribution – to provide diversity within the suite of detection algorithms.

5.4.2 Combining methods

Every unique target detection method has its advantages and disadvantages or successes and failures depending on the content and condition of the images under evaluation. One would be hard pressed to find a target detection method that successfully identifies all targets all of the time in all images. This is why our lab has decided that a combination of methods with a well designed ranking scheme offers a more appropriate and useful solution to target detection.

Rather than being a stand-alone product, combining our histogram-based method with other detection methods, such as shape analysis and a Fourier transform-based

parallel edge finder, improves detection specificity. In a separately-developed information fusion scheme, the response planes from each detection algorithm yield essential information to the observer. Rather than relying on a single target feature, by combining multiple object signatures, such as grayscale intensity, shape, size, edges, homogeneity and pattern, all which may be intentionally or unintentionally obscured, we maximize the probability of finding the target of interest.

Higher accuracy in the decision making process results from presentation of the fused response plane information. Using the ROC results to assess the usefulness of the information, such as when the ROC probability resides below some pre-defined level, we could exclude one or more response planes from the fusion process due to lack of information or incorrect detections. The flexibility of this framework allows us to optimize computation time for a given imaging scenario and also supports this method for use in alternate imaging tasks such as computer-aided diagnoses in medicine.

Because this method of object detection makes use of digital images, a smooth transition to computer-aided detection (CAD) in medical imaging is possible. Similar to this method, object detection using template matching methods is presently being studied in mammography by using a library of known ground truth ROIs to detect breast masses [30].

5.5 Future work

Given additional resources and time, potential new avenues to explore would include testing real-world images, experimenting with image preprocessing, and varying the target objects. The images used in this study have been staged and artificially altered. Testing the algorithm on real-world images would confirm its effectiveness. Additionally,

it would be interesting to see the results from histogram differencing without wobble on an image that is preprocessed with a readily accepted image enhancement filter as compared to the results of differencing with wobble on the unprocessed image. Since the template and target, a Scud missile launcher, remained constant throughout this study, another interesting avenue to pursue would be the evaluation of different target objects. For example, learning if the wobble method responds best to hard edges of man-made objects, or does it respond equally as well to those edges found in nature could prove useful especially when considering medical images. Given the promising results of this study, continued testing is warranted.

6. REFERENCES

- [1] Karvir HV and Skipper JA (2007) Power spectrum weighted edge analysis for straight edge detection in images, SPIE Defense and Security Symposium: Visual Information Processing XVI, 6575.
- [2] Repperger DW, Pinkus AR, Skipper JA and Schrider CD (2007) Object recognition via information theoretic measures/metrics, IS&T/SPIE 19th Annual Symposium Electronic Imaging Science and Technology: Computational Imaging V, 6498.
- [3] Ganapathy P and Skipper JA (2007) A novel approach for performance evaluation of target detection algorithms, SPIE Defense and Security Symposium: Visual Information Processing XVI, 6566.
- [4] Kunimitsu S, Asama H, Kawabata K, and Mishima T (2005) Detection of object under outdoor environment by matching with partial and whole templates, Electrical Engineering in Japan, 154-C, 55-66.
- [5] Predator RQ-1 / MQ-1 / MQ-9 reaper-unmanned aerial vehicle (UAV), USA, Internet: United States Air Force, <http://www.airforce-technology.com/projects/predator/>, accessed December 12, 2007.
- [6] Roberts K, Imagery for the battlefield, Internet: Military Geospatial Technology, Online Edition, http://www.military-geospatial-technology.com/print_article.cfm?DocID=860, accessed August 8, 2007.
- [7] Medical imaging revolutionizes modern medicine, Internet: American Society of Radiologic Technologists, News release: October 1, 2007, <https://www.asrt.org/Content/News/PressRoom/PR2007/medicalima071001.aspx>, accessed February 1, 2008.
- [8] Russ JC (2002) *The Image Processing Handbook*, 4th ed., Boca Raton, Florida: CRC Press LLC.
- [9] Stark JA and Fitzgerald WJ (1996) An alternative algorithm for adaptive histogram equalization, Graphical Models and Image Processing 58 (2), 180-185.
- [10] Tom VT and Wolfe GJ (1982) Adaptive histogram equalization and its applications, SPIE Applications of Digital Image Processing IV 359, 204-209.
- [11] Canny J (1986) A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6), 679-698.

- [12] Marr D and Hildreth E (1980) Theory of edge detection, Proceedings of the Royal Society of London. Series B, Biological Sciences 207 (1167), 187-217.
- [13] Roberts LG (1963) Machine perception of three-dimensional solids, Thesis, Massachusetts Institution of Technology.
- [14] Sobel I (1970) Camera models and machine perception, Thesis, Stanford University.
- [15] Kirsch RA (1971) Computer determination of the constituent structure of biological images, Computers and Biomedical Research 4, 315-328.
- [16] Frei W (1977) Fast boundary detection: a generalization and a new algorithm, IEEE Transactions on Computers, C-26 (10), 988-998.
- [17] Hough PVC (1962) U.S. Patent 3 069 654, Method and means for recognizing complex patterns.
- [18] Prewitt JMS and Mendelsohn ML (1965) The analysis of cell images, Annals of the New York Academy of Sciences 128 (3), 1035-1053.
- [19] Cootes TF and Taylor CJ (2001) Statistical models of appearance for medical image analysis and computer vision, Medical Imaging: Image Processing, Proceedings SPIE 4322, 236-248.
- [20] Garboczi EJ, Bentz DP and Martys NS (1999) Digital images and computer modelling, Experimental Methods for Porous Media, 35, 1-41.
- [21] Weapons of mass destruction, Internet: WMD Around the World, <http://www.fas.org/nuke/guide/russia/icbm/rt-2pmu.htm>, accessed January 7, 2008.
- [22] Vassiliadis S, Hakkennes EA, Wong JSSM, and Pechanek G (1998) The sum-absolute-difference motion estimation accelerator, 24th EUROMICRO conference, 2, 559-556.
- [23] Green DM and Swets JA, (1966) *Signal Detection Theory and Psychophysics*, New York: John Wiley & Sons, Inc..
- [24] Montgomery DC and Runger GC (2002) *Applied Statistics and Probability for Engineers*, 3rd ed., New York: John Wiley & Sons, Inc..
- [25] Chalidabhongse TH, Kim K, Harwood D and Davis L (2003) A perturbation method for evaluating background subtraction algorithms, Performance Metrics for Intelligent Systems Workshop 2003, NIST.

- [26] Liu G and Haralick RM (2002) Optimal matching problem in detection and recognition performance evaluation, *The Journal of the Pattern Recognition Society* 35, 2125-2139.
- [27] Peli E (1990) Contrast in complex images, *Journal of the Optical Society of America*, 7 (10), 2032-2040.
- [28] Westheimer G (1965) Spatial interaction in the human retina during scotopic vision, *Journal of Physiology* 181, 881-894.
- [29] Westheimer G (1985) The oscilloscopic view: Retinal illuminance and contrast of point and line targets, *Vision Research* 25 (8), 1097-1103.
- [30] Tourassi GD, Vargas-Voracek R, Catarious DM, Jr and Floyd CE, Jr, (2003) Computer assisted detection of mammographic masses: A template matching scheme based on mutual information, *Medical Physics* 30 (8), 2123-2130.

APPENDIX A

search.m

```
% search.m
% Code developed by Biomedical Medical Imaging Laboratory of Wright State
% University 2006-2008

% By: CD Schrider, HV Karvir, and JA Skipper

% search.m This is the main program for using sliding histogram differences
% to locate targets in an image. The search.m program calls the following
% functions: contrastprediction.m, target_histogram.m,
% histogram_differencing.m, and local_min.m. In addition, the search.m
% program loads contrast prediction library.mat and Template Scud
% library -90 to 90.mat that contain necessary arrays to perform contrast
% prediction and selection of wobble range.

clear all;
close all;
clc;

% READ TARGET IMAGE-----
imagedir = dir('*.tif');
for selectedimage = 1:length(imagedir)
    filename = imagedir(selectedimage).name

    % [FileName,PathName]=uigetfile({'*.tif'},'File Selector');
    % IMFN and IMEXT variables are used to name the output response plane
    [IMFN, IMEXT] = strtok(filename, '.');
    % imagepath = [PathName FileName];
    I = imread(filename);
    I = imcrop(I, [260 120 1075 785]); % image border is excluded, new size (786 x 1076)

% INITIALIZING PARAMETERS-----
starttime = cputime; % Initialize timer
blk = 33; % Block size for processing image

%%%%%%%%%(Algorithm optimized for blk size 30-35) For a different blk size,
% a different library range and histogram scaling factor in the function
% target_histogram will provide optimal results. %%%%%%%%%%

inc = 2; % Pixel increment size (value = 1 or 2)
n = [5 5]; % Averaging filter size to smooth output
sigma = 1; % Averaging filter std to smooth output
rlim = [8 8]; % Histogram width limits for selective processing based on percentage
```

```

% using rlim = [8 8] selects a histogram having width of 20 gray levels to
% eliminate from difference processing

N = 3; % Number of comparisons in the sliding difference operation

%% WOBBLE ENABLE/DISABLE WINDOW
wobble = 'y';

% IMAGE BIT CONVERSION-----
% Preparing for bit depth other than 8-bit
BD = imfinfo(filename);
bits = 2.^BD.BitDepth;
imbits = 8;
cbits = 8;
if bits > 256
    I=double(I);
    I = round(I/bits*255);
    I=uint8(I);
end

% IMAGE HISTOGRAM-----
[counts bins] = imhist(I); % Histogram of original image

% Setting range limit as percentage of total bins for selective processing
rlim = [round(rlim(1)*length(counts)/100) round(length(counts)-rlim(2)*length(counts)/100)];

% LOADING WOBBLE LIBRARY-----
% Loading library created using Ltransform function
load('Template Scud library -90 to 90.mat');
library = library';
s = size(library);
library = padarray(library,[0 55],'post'); % Form padded array all of wobbled library

% SELECT RANGE FROM LIBRARY-----
if wobble=='y' || wobble=='Y' % Select wobble range
    % MASK SIZE
    [wobblerange] = contrastprediction(I); % Find upper/lower limits using RMS of selected ar
    size1 = wobblerange(1,1);
    size2 = wobblerange(1,2);
    l1 = size1+(ceil(s(1)/2)); % Compute lower limit in library
    l2 = size2+(ceil(s(1)/2)); % Compute upper limit in library

    if l1 == l2
        wobble = 'n';
    end

elseif wobble=='n' || wobble=='N' % With no wobble, choose center bin
    l1 = ceil((s(1)-1)/2+1);
    l2 = ceil((s(1)-1)/2+1);
    size1 = 0;
    size2 = 0;
else
    disp('Input error please enter y or n');
end
library = library(l1:l2,:); % Selecting the wobble range from library

% GET IMAGE SIZE-----
Im = size(I,1);
In = size(I,2);

% CREATE PADDED IMAGE FOR ANALYSIS-----
Ipad=padarray(I,[ceil(blk/2) ceil(blk/2)],'symmetric','both');

```

```

% [x y] = size(Ipad);

% INITIALIZE MAP OF ZEROS FOR RESPONSE PLANE -----
map = ones(size(Ipad,1),size(Ipad,2));

% MAIN-----
h = waitbar(0,'PROCESSING IMAGE... Please wait...');

% ROI BASED PROCESSING-----

for ii = ceil(blk/2):inc:size(Ipad,1)-floor(blk/2)
    for jj = ceil(blk/2):inc:size(Ipad,2)-floor(blk/2)
        ROI=Ipad(ii-ceil(blk/2)+1:ii+floor(blk/2),jj-ceil(blk/2)+1:jj+floor(blk/2));

        smooth = 'n'; % Tag(y/n) to enable histogram smoothing
        [R_freq2, R_bins2] = target_histogram(ROI,smooth,imbits,cbits);
        R_freq2 = R_freq2;

% FIND HISTOGRAM WIDTH FOR SELECTIVE PROCESSING-----
r = [0 0];
k = 1;
gt = [0 0];
while gt(1)<0.05*sum(R_freq2) % Lower marker at 5 % of area level
    gt(1) = sum(R_freq2(1:k)); % Sum of counts
    r(1) = k; % Moving into the histogram from the lowest bin
    k = k+1; % Counter
end
k = 1; % Reset counter
while gt(2)<0.05*sum(R_freq2) % Upper marker at 5 % of area level
    gt(2) = sum(R_freq2(length(R_bins2)-k:length(R_bins2))); % Sum of counts
    r(2) = length(R_bins2)-k; % Moving into the histogram from the highest bin
    k = k+1; % Counter
end
range = r(2) - r(1); % Histogram width

% CONDITIONAL PROCESSING BASED ON HISTOGRAM WIDTH-----
if range < rlim
    m = blk * blk; % this would be the maximum difference possible
else
    smooth = 'y'; % Tag(y/n) to enable histogram smoothing
    [R_freq, R_bins] = target_histogram(ROI,smooth,imbits,cbits);

    % HISTOGRAM DIFFERENCING ('N' COMPARISONS FOR EACH ROI)
    [c1] = histogram_differencing(R_freq,R_bins,library,wobble,N);

    % LOCAL MINIMUM LOCATIONS FOR 'N' DIFFERENCE ARRAYS
    [lm_c1] = local_min(c1);

    if size(lm_c1,1) ~= 1
        m = lm_c1(1,1);
    else
        m = lm_c1;
    end
end

% CREATING A MAP-----
% Need to add provision for block step size greater than 2, filling in
% all values around ROI center with m.
if inc == 1
    map(ii-1,jj-1) = m;
elseif inc > 1
    map(ii-(inc-1):ii,jj-(inc-1):jj)=m;

```

```

        elseif inc > 5
            disp('Error: Inc value should be 1, 2, 3, or 4');
            return;
        end
    end
    waitbar((ii-floor(blk/2))/(size(Ipad,1)-blk));
end
close (h);

% SMOOTHING OUTPUT MAP USING GAUSSIAN FILTER-----
h = fspecial('gaussian',n,sigma);
map = conv2(map,h);
map_crop = imcrop(map,[(ceil(blk/2) + ceil(n(1)/2)) (ceil(blk/2) + ceil(n(1)/2)) In-1 Im-1]);

% SCALE THE RESPONSE PLANE
map_min=min(min(map_crop));
map_max=max(max(map_crop));
denom = map_max-map_min;

if denom == 0
    denom = 1; % avoid denominator divide by zero patch
end

map_scaled = (map_crop-map_min)*256/denom;
map_final = uint8(map_scaled);

% % SAVE MAP-----
lowerwobble = num2str(size1);
upperwobble = num2str(size2);
RPfilename = ['CDS RP CASE5',IMFN,' ',lowerwobble,' to ',upperwobble,' wobble.tif'];
imwrite(map_final,RPfilename);

% Clear variables before next image processes
clear I;
clear counts;
clear bins;
clear Im;
clear In;
clear map;
clear h;
clear ii;
clear jj;
clear R_freq2;
clear R_bins2;
clear ROI;
clear range;
clear m;
clear c1;
clear lm_c1;
clear map_crop;
clear map_min;
clear map_max;
clear denom;
clear map_scaled;
clear map_final;

end % End of the image directory loop

disp('Code has completed');

endtime = (cputime - starttime)/60 % Compute total processing time

```

contrastprediction.m

```
function [wobblerange] = contrastprediction(I)

% contrastprediction.m
% Code developed by Biomedical Medical Imaging Laboratory of Wright State
% University 2006-2008

% By: CD Schrider and JA Skipper

% contrastprediction calculates a predicted image contrast to provide an
% effective wobble range for use in the histogram differencing algorithm.
% This function is called by search.m, and it loads
% contrast prediction library.mat to select the best range of wobble.

% INITIALIZING VARIABLES-----
RC = 9; % median filter kernel row size
B = I; % Image for smoothing
block = 33; % Block size for processing variance
RD = B; % Red color layer for displaying CV candidate regions
GR = B; % Green color layer for displaying CV candidate regions
BL = B; % Blue color layer for displaying CV candidate regions

% MEAN AND STANDARD DEVIATION GRAYSCALE VALUES FROM ORIGINAL IMAGE-----
image_mean = round(mean2(I));
image_std = round(std2(I));
lowbound = round(image_mean-(0.15*image_std)); % Lower bound of global range
hibound = round(image_mean+(0.15*image_std)); % Upper bound of global range

% PAD THE IMAGE FOR BLOCK PROCESSING-----
padC = abs(block-round(block*(size(B,1)/block - floor(size(B,1)/block + 1))));
padR = abs(block-round(block*(size(B,2)/block - floor(size(B,2)/block + 1))));
B = padarray(B, [padC padR], 'symmetric', 'post');

% MEDIAN FILTERED IMAGE-----
B = medfilt2(B,[RC RC]); % median filter image with RC x RC kernel
ii = 1; jj = 1; % Initialize x & y coordinates to store

% MEAN AND VARIANCE COMPARISON TO ORIGINAL IMAGE-----
% Store/save the locations of the regions matching the mean conditions
% of lowbound and hibound
g = 1; % initialize loop variable
k = 1; % initialize loop variable
for ii = 1:block:size(B,1)-block-1
    for jj = 1:block:size(B,2)-block-1
        ROI=B(ii:ii-1+block, jj:jj-1+block);
% LOCATING REGIONS HAVING MEAN AT IMAGE_MEAN +/- IMAGE_STD-----
        Bmean = mean2(ROI); % calculate mean of the block
        Bvar = var(ROI(:)); % Calculate the variance of the block
        Lib(k,1) = ii + ceil(block/2)-1; % Store center block row coordinate
        Lib(k,2) = jj + ceil(block/2)-1; % Store center block column coordinate

        if lowbound < Bmean && Bmean < hibound
            % Coefficient of variation for ROIs that fit into mean
            % graylevel range
            Lib(k,3) = sqrt(Bvar)./Bmean;
            cvsample(g,1) = sqrt(Bvar)./Bmean; % Array of selected CV values
            g = g+1;
        else
            Lib(k,3) = 0;
        end
    end
end
```

```

        k = k+1;
    end
end

% USE SHAPE OF CVSAMPLE DATA TO CONTROL RANGE OF CV USED IN FINAL SELECTION
% CRITERION
[N,X] = hist(cvsample); % histogram of CV array
cvmu = mean(cvsample); % mean of CV array
cvsigma = std(cvsample); % Standar deviation of CV array
cvlow = (cvmu-cvsigma); % set lower bound of optimum CV range using CV array data
cvhi = (cvmu+cvsigma); % set upper bound of optimum CV range using CV array data

% if calculated CVLOW falls below zero, set its value to zero
if cvlow < 0
    cvlow = 0;
% if calculated CVHI exceeds maximum CVSAMPLE, set its value to the maximum
% CVSAMPLE
elseif cvhi > max(cvsample)
    cvhi = max(cvsample);
end
CVrange = [cvlow cvhi]; % Optimum CV range

% LOOKING AT THE CV OF ROIS WITHIN THE IMAGE -----
Lib_size=k-1;
p = 1; % initialize loop variable
for k=1:Lib_size
    if Lib(k,3)>CVrange(1) && Lib(k,3)<=CVrange(2) % turn center pixels orange
        RD(Lib(k,1)-2:Lib(k,1)+2, Lib(k,2)-2:Lib(k,2)+2) = 255; % Red color layer
        GR(Lib(k,1)-2:Lib(k,1)+2, Lib(k,2)-2:Lib(k,2)+2) = 128; % Green color layer
        BL(Lib(k,1)-2:Lib(k,1)+2, Lib(k,2)-2:Lib(k,2)+2) = 0; % Blue color layer
        cvset(p,1)= Lib(k,3); % Save the CV value within desired range
        p = p+1;
    end
end

% DISPLAY IMAGE WITH SELECTED ROIs DOTTED WITH ORANGE 5X5 BLOCKS -----
Out(:,1)=RD; % Red color layer
Out(:,2)=GR; % Green color layer
Out(:,3)=BL; % Blue color layer
figure(3); imshow(Out), title('Selected ROIs at orange dots');

% LOAD LIBRARY OF RMS VALUES -----
[LibFileName,LibPathName]=uigetfile({'*.mat'},'File Selector');
librarypath = [LibPathName LibFileName];
load (librarypath); % Loading RMS library
avgCV = mean(cvset); % Build library with CV in column 1

% LOCATE BEST MATCH OF THE IMAGE CV WITHIN THE CV LIBRARY -----
scv = size(CVlib,1); % length of CV array
if avgCV >= CVlib(scv,1) % in case avgCV exceeds CV library values
    minCV = CVlib(scv,1);
    maxCV = minCV;
elseif avgCV <= CVlib(1,1) % in case avg CV less than CV library values
    minCV = CVlib(1,1);
    maxCV = minCV;
else
    minCV = max(find(CVlib(:,1)<=avgCV));
    maxCV = min(find(CVlib(:,1)>=avgCV));
end

% CALCULATE THE APPROPRIATE WOBBLE RANGE -----
CVdiff = scv - minCV;

```

```

if minCV == maxCV
    if CVdiff >= 4
        size1 = minCV - 10 + CVdiff;
        size2 = maxCV + CVdiff;
    elseif CVdiff <= scv - 4
        size1 = minCV - CVdiff;
        size2 = maxCV + 10 - CVdiff;
    else
        size1 = CVlib(minCV,2)-5;
        size2 = CVlib(maxCV,2)+5;
    end
else
    size1 = CVlib(minCV,2)-5;
    size2 = CVlib(maxCV,2)+5;
end
wobblerange = [size1 size2]; % Range of wobble to be used
return;

```

target_histogram.m

```
function [R_freq1, R_bins] = target_histogram(A,tag,imbits,cbits% target_histogram. is called by search.m and uses a
Hamming window for
% smoothing the histogram signal.

% COMPUTE HISTOGRAM-----
[R_freq,R_bins]=imhist(A);

% ELLIMINATE BACKGROUND PIXELS (VALUE = 255 OR VALUE = 0)-----
lastbin = 2^imbits-1;
R_freq(find(R_bins==lastbin))==0;

R_freq = R_freq(1:2^cbits); % Select histogram range based on image bits
R_bins = R_bins(1:2^cbits); % Select histogram range based on image bits

% SAVE ORIGINAL HISTOGRAM-----
R_freq_unsmoothed = R_freq;

% SMOOTHING WINDOW-----
% w = hamming(19);
w = [0.08 0.1077 0.1876 0.31 0.4601 0.6198 0.77 0.8923 0.9722...
     1 0.9722 0.8923 0.77 0.6198 0.4601 0.31 0.1876 0.1077 0.08];

w = w./sum(w); % Normalize filter to ensure sum(w) = 1

% SMOOTHING HISTOGRAM (CONDITIONAL PROCESSING)-----

scalefactor = 0.7;

if tag(1) == 'y' || tag(1) == 'Y'
    R_freq1 = conv(R_freq,w)*scalefactor; % Convolve to smooth
    % % % % Histogram scaling factor (0.7) is optimized for blk size of 33.
    % For other blk sizes change factor by comparing target and template
    % histograms. % % % %
else
    R_freq1 = R_freq;
end

R_freq1 = R_freq1(1:2^cbits);

return;
```


histogram_differencing.m

```
function [c1] = histogram_differencing(R_freq,R_bins,library,wobble,N)

% histogram_differencing
% Code developed by Biomedical Medical Imaging Laboratory of Wright State
% University 2006-2008

% By: CD Schrider, HV Karvir, and JA Skipper

% histogram_differencing.m overlaps the template and target ROI histograms
% and uses a circular shift to slide the histograms together. At each step
% a difference is calculated. histogram_differencing.m is called by
% search.m.

s1 = size(library);

% FORM PADDED ARRAY-----
R_freq1 = padarray(R_freq,[55 0],'pre'); % ROI histogram frequency
R_freq2 = R_freq1;

% SLIDING DIFFERENCE-----

if wobble(1) == 'y' || wobble(1) == 'Y'
    % GENERATE RANDOM NUMBER FOR PICKING WOBBLE-----
    d = round((rand(1,3)*10)*((s1(1)-2)/10))+2;

    for b = 1:N
        T_freq1 = library(d(b),:); % Pick histogram corresponding to the random number generated
        bins = 1:1:length(T_freq1); % Number of grayscale levels

        for p = 1:126 % Number of slides
            c1(b,p) = sum(abs(T_freq1-R_freq1)); % Compute sum of absolute differences
            R_freq1 = circshift(R_freq1,[-1 0]); % Shift target histogram to right
            T_freq1 = circshift(T_freq1,[1 0]); % Shift template histogram to left
        end
        R_freq1 = R_freq2; % Reset to original histogram for next comparison
    end

else % For no wobble, compare with the center bin (optimal histogram)
    T_freq1 = library(1,:); % Pick center histogram from library
    bins = 1:1:length(T_freq1); % Number of grayscale levels
    for p = 1:120 % Number of slides
        c1(p) = sum(abs(T_freq1-R_freq1)); % Compute sum of absolute differences
        R_freq1 = circshift(R_freq1,[-1 0]); % Shift target histogram to right
        T_freq1 = circshift(T_freq1,[1 0]); % Shift template histogram to left
    end
end

return;
```

local_min.m

```
function [lm_c1] = local_min(c1)

% local_min.m
% Code developed by Biomedical Medical Imaging Laboratory of Wright State
% University 2006-2008

% By: CD Schrider and JA Skipper

% local_min.m calculates the derivative of the difference arrays obtained
% from the histogram_differencing function. local_min.m is called by the
% main program, search.m.

% row_num = 0;
c1_A = c1(:,1:size(c1,2)-1);
c1_B = c1(:,2:size(c1,2));

for row_num = 1:size(c1,1) % have possibility of multiple wobbles

    % Finding the derivative of difference array(s) [c1]
    der_c1(row_num,:) = c1_A(row_num,:)-c1_B(row_num,:);

    % Look for zero-crossings and find the local minima
    der_c1_A = der_c1(:,1:size(c1_A,2)-1);
    der_c1_B = der_c1(:,2:size(c1_A,2));

    mult_der(row_num,:) = der_c1_A(row_num,:).*der_c1_B(row_num,:);

    if any(mult_der(row_num,:) <= 0)
        [ROW,COL] = find(mult_der <= 0);
    else
        disp('No negative in mult_der...');
        ROW = 1; COL=1;
    end
end
% finds absolute minimum from the rows and columns of local minima
if COL == 1
    lm_c1 = c1(ROW,2);
else
    lm_c1 = min(min(c1(ROW,COL+1)));
    val_c1 = min(min(c1));
end

% need to consider the endpoints in comparison to the local min
R = min(min(c1(:,1))); % minimum right endpoint
L = min(min(c1(:, size(c1,2))))); % minimum of left endpoint
endpt_diff = R-L; % if (-) then right endpt is less than left
if endpt_diff > 0
    compare_L = lm_c1 - L;
    if compare_L > 200
        lm_c1 = L; % Use left endpoint instead of local minimum
    end
elseif endpt_diff <= 0
    compare_R = lm_c1 - R;

    if compare_R > 200
        lm_c1 = R;
    end
end

return;
```