AD

AD-E404 425

Technical Report ARMET-TR-21048

HIGH-PRECISION CENTER ESTIMATION OF POINT SOURCE INFRARED TARGETS

Steven Manole
Ryan Decker

December 2022

U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT
COMMAND ARMAMENTS CENTER

Munitions Engineering and Technology Center

Picatinny Arsenal, New Jersey

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
December 2022	Final	March 2020	April 2021

4. TITLE AND SUBTITLE
High-Precision Center Estimation of Point Source Infrared Targets

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
		622624
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
H1A00		

6. AUTHOR(S)
Steven Manole and Ryan Decker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army DEVCOM AC, METC
Armaments Engineering Analysis and Manufacturing Directorate (FCDD-ACM-AA)
Picatinny Arsenal, NJ 07806-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
N\A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
U.S. Army DEVCOM AC, ESIC Knowledge & Process Management Office (FCDD-ACE-K) Picatinny Arsenal, NJ 07806-5000		Technical Report ARMET-TR-21048

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Several different methods of estimating the subpixel center of a point source in infrared imagery are explored. These include optimized Gaussian fitting, analytic Gaussian and paraboloid global optimum solutions, and weighted-centroid approaches. Each of these methods is applied to a variety of randomly generated point source test images through Monte Carlo simulation. Several factors are incorporated, including Gaussian noise, image saturation, and orientation angle. Using the standard deviation of the errors from the Monte Carlo runs as a metric, each of the subpixel estimation algorithms is compared. Overall, the optimized Gaussian-fitting algorithm produces the best results on clean imagery, but the weighted-centroid method is most accurate for noisy, saturated images.

15. SUBJECT TERMS
Computer vision Image processing Infrared Point source Gaussian Subpixel accuracy Optimized Gaussian fitting

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	19

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER (Include area code)
Steven Manole	(973) 724-2744

Standard Form 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

CONTENTS

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the In-House Laboratory Independent Research (ILIR) panel at the U.S. Army Combat Capabilities Development Command (DEVCOM) Armaments Center (AC), Picatinny Arsenal, NJ, for their support in conducting this research.

INTRODUCTION

The purpose of this report is to explore multiple methods that might be used to estimate the center of an infrared (IR) target. The specific focus will be on targets that emit a signature in a very small region of space, approximating a point source. If only a simple solution is required, the center could be roughly estimated as the brightest pixel in the region of interest (ROI). While very easy to implement, this approach would only yield an estimate precise to a single pixel. Without any further refinement, this would establish a lower bound on accuracy. However, by considering the neighboring pixels, it is possible to refine the estimate to a fraction of a pixel. Thus, the selected algorithms must be capable of subpixel estimation in order to establish the necessary degree of precision for high-accuracy applications. Several of these applications might include tracking a projectile throughout a video or performing a camera calibration using IR targets.

The importance of limiting these algorithms to point sources is that it allows for very precise estimation of a target center. The ideal candidate would possess a single, well-defined peak over a small region of the image. The ability for an object to act as a point source varies based on the distances involved. A larger object may not be considered a point source except at very long distances, and a point source at short distances may fail to register at all at longer distances. Several objects could be considered applicable in this regard. Halogen light bulbs produce heat from a small filament in the center of the bulb. A projectile with a tracer will have a small, but bright, burning point at the base of the round. The head of a rivet cools quickly on the outer edges but slowly in the center, creating a hot spot at the dead center. All of these elements are good candidates for such an algorithm. A nearby vehicle, which may appear large (hundreds of pixels) with a shallow gradient in its IR signature, would not be amenable to such a tracking algorithm.

METHODS

First, it is necessary to establish a whole-pixel approach, which will act as a control and benchmark. This algorithm acts to find the pixel location with the greatest intensity. In the case of a tie, the solution will be the average of the peak locations. Therefore, it is possible that the whole-pixel estimate produces a decimal value if pixel values are saturated. Note that due to the nature of the image generation, the maximum value and therefore the whole-pixel estimate will occur at the center of the sub-image (or near the center in the case of a saturated image). This means that the results for the remaining algorithms will be valid for any size image as long as the image is cropped and centered to the ROI.

A true point source as recorded by a camera can be represented mathematically as an Airy disk. Since this function is rather complex, a slightly less-accurate estimate can be made using the well-known Gaussian distribution. This is a very common simplification, and while the tails of both functions do not match, the center, the most important location for estimation, matches very nicely (ref. 1). The candidate point sources, given the appropriate distance, would be expected to exhibit a signature similar to an Airy disk or Gaussian distribution. Therefore, the most logical method of finding the subpixel center would be to fit a two-dimensional (2D) Gaussian function to the image region, obtaining the coordinates of its center. In practice, fitting a 2D Gaussian distribution using MATLAB requires running an optimization, which can be rather slow. Therefore, although this method can be very precise, it may be desirable to have a solution that can run much more quickly. The method of fitting a 2D Gaussian distribution to the image region is known as optimized Gaussian fitting.

For a less computationally expensive solution, a deterministic and analytic approach is sought. To start, a novel method for fitting a parabolic estimator is described in reference 2. While this fulfills both requirements of a deterministic and analytic form, it is only derived for the one-dimensional (1D) case. In order to be useful for estimating the subpixel center of a point source, it

must be derived in two dimensions. This estimator might be expanded into 2D space according to equations 1 through 9.

In order to derive the 2D Gaussian form of the subpixel peak detection, begin by using the fact that taking the natural logarithm of the Gaussian function will not change the peak location, while also transforming it into a parabolic function

$$\hat{f}(x,y) = A\, e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)} \tag{1}$$

$$\ln\left(\hat{f}(x,y)\right) = f(x,y) = lnA - \frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2} \tag{2}$$

The peak of a paraboloid will be located at the position where

$$\text{grad}(f(x+\delta x, y+\delta y)) = 0 \tag{3}$$

Substituting a Taylor series expansion along both axes (ref. 3) arrives at

$$grad\left(f + f_x\delta x + f_y\delta y + \frac{1}{2}\left[f_{xx}\delta x^2 + 2f_{xy}\delta x\delta y + f_{yy}\delta y^2\right] + O(\delta x^3 + \delta y^3)\right) = 0 \tag{4}$$

and

$$0 = f_x + f_{xx}\delta_x + f_{xy}\delta_y + O(\delta_x^2) \tag{5}$$

$$0 = f_y + f_{yy}\delta_y + f_{xy}\delta_x + O\left(\delta_y^2\right) \tag{6}$$

If ignoring the higher-order terms, this can be rewritten in matrix form as

$$\begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} -f_x \\ -f_y \end{bmatrix} \tag{7}$$

and rearranged as

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}^{-1}\begin{bmatrix} -f_x \\ -f_y \end{bmatrix} \tag{8}$$

Note that in order to use this form to determine the subpixel peak of an image, the natural logarithm of the pixel values must first be computed such that the function f represents the logarithmic transform of the outputs $\hat{f}$. The subpixel peak would then be (δ_x, δ_y). Note that the equation can also be condensed as

$$\boldsymbol{\delta} = -\boldsymbol{H}^{-1}\nabla\boldsymbol{f} \tag{9}$$

An identical derivation can be used to construct the 2D paraboloid fit, omitting the need to take the natural logarithm, as the form would already be parabolic. Both versions will be used to make comparisons. These will be notated as analytic Gaussian and analytic paraboloid.

The 2D Gaussian and paraboloid estimators derived are approximations of the Airy disk. Figure 1 shows a cross section of these three functions as they appear in two dimensions. Those would be the 1D Gaussian, the parabola, and the Airy pattern. The Gaussian estimator clearly

follows the Airy disk more closely, especially at the ends. The simpler paraboloid estimator is sufficient near the center but deviates much more strongly from the Airy disk at the ends.

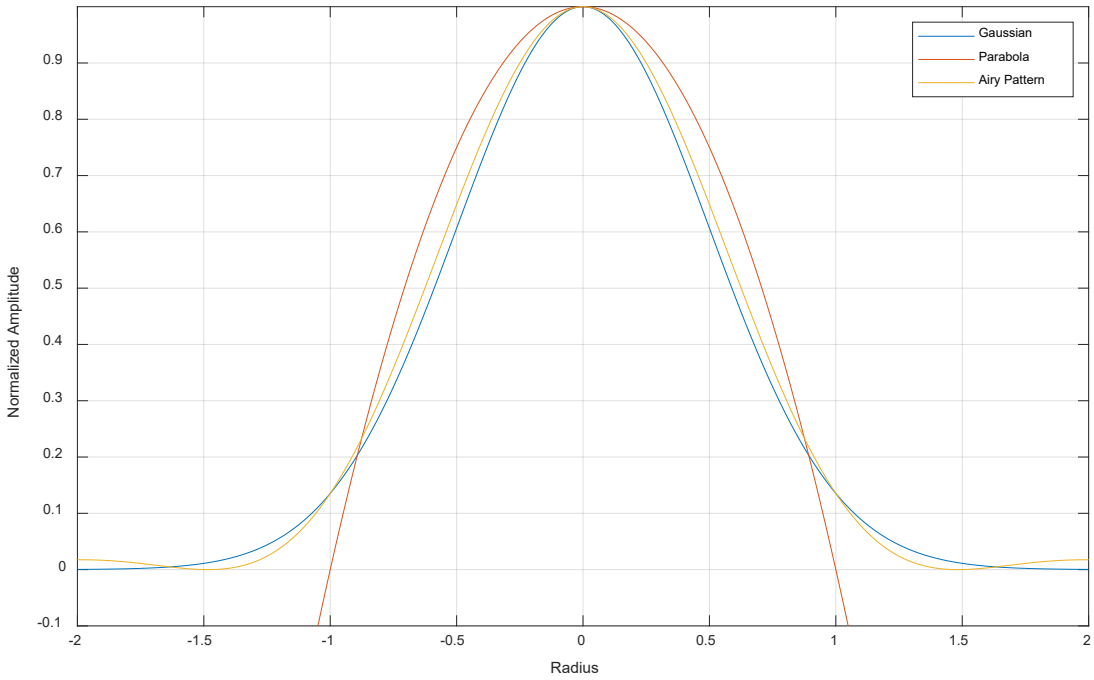

Figure 1
Cross-sectional view of various estimators

In order to solve directly for the subpixel peak, it is necessary to define matrix operators for the derivatives. The concept of a second derivative of discrete pixel values in 2D is not well established. While the Sobel filter (ref. 4) estimates a single derivative, there is no standard way of describing a second partial derivative. Therefore, several potential solutions were investigated. To construct a 3 x 3 operator, the Sobel kernel can be used to compute the derivatives. However, in the case of a very bright or saturated object, it may be necessary to enlarge the size of the fitting window. Therefore, a higher order operator must be devised. The method used herein is to generate the coefficients of a higher degree, first-derivative central difference in 1D. These coefficients will form the basis of a larger derivative operator. In order to do so, the methods used by Kreider (ref. 5) can be generalized to any number of central difference coefficients. In this method, the even derivatives of the Taylor expansion are set to zero and the odd derivatives are alternating. The objective is to sum all derivatives besides the one in question to zero. Thus, it is a matter of solving a system of equations where the odd derivative coefficients sum to zero.

$$\begin{bmatrix} 1^3 & 2^3 & \cdots & n^3 \\ 1^5 & 2^5 & \cdots & n^5 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{2m+1} & 2^{2m+1} & \cdots & n^{2m+1} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{10}$$

The vector of coefficients $\boldsymbol{A}$ will alternate in sign in a known pattern, so it is only necessary to solve for the positive coefficients. Furthermore, the previous equation is of dimension $m \times n$ such that $m = n - 1$. Since there are infinitely many solutions, for the sake of simplicity, let $A_m = -1$ and reduce the column dimension by one before solving. For a 3rd degree solution, the result is:

$$\boldsymbol{A}^{m=3}_{\text{positive}} = [45 \quad 9 \quad 1] \text{ and } \boldsymbol{A}^{m=3} = [-1 \quad 9 \quad -45 \quad 0 \quad 45 \quad -9 \quad 1] \tag{11}$$

These coefficients that makeup $\boldsymbol{A}$ are multiplied by a 1D Gaussian-smoothing filter of the same order to create an *n* x *n* derivative operator. To compute the second derivative operator, the (*n* - 1) x (*n* - 1) derivative operator is convolved with the Sobel kernel. These filters are normalized such that the absolute values of all elements sum to one. The set of 3 x 3 operators constructed in this manner would be:

$$\frac{d}{dx} = \begin{bmatrix} -.0533 & 0 & .0533 \\ -.3935 & 0 & .3935 \\ -.0533 & 0 & .0533 \end{bmatrix} \frac{d}{dy} = \begin{bmatrix} -.0533 & -.3935 & -.0533 \\ 0 & 0 & 0 \\ .0533 & .3935 & .0533 \end{bmatrix}$$

$$\frac{d}{dx^2} = \begin{bmatrix} .0625 & -.1250 & .0625 \\ .1250 & -.2500 & .1250 \\ .0625 & -.1250 & .0625 \end{bmatrix} \frac{d}{dy^2} = \begin{bmatrix} .0625 & .1250 & .0625 \\ -.1250 & -.2500 & -.1250 \\ .0625 & .1250 & .0625 \end{bmatrix} \tag{12}$$

$$\frac{d}{dxdy} = \begin{bmatrix} .2500 & 0 & -.2500 \\ 0 & 0 & 0 \\ -.2500 & 0 & .2500 \end{bmatrix}$$

Notice that the first derivative operators differ from those of the Sobel kernel. To remain truer to the Sobel kernel, a weighted-average smoothing instead of a Gaussian smoothing might have been applied. This was not done here because the weights would add additional parameters that would need to be tuned. Several other methods might be used to compute these higher order derivatives and result in a different set of operators. Those presented here are one of many ways to do so and there is no clear consensus on which produce the most accurate result.

Finally, an estimate of the subpixel center is constructed using the centroid of the image region. Each coordinate is given a weight based on the pixel value, and then the average is taken in each direction. This algorithm will be notated by weighted centroid.

In order to measure the accuracy of each method, a MATLAB (ref. 6) script was written to apply each algorithm to a set of computer-generated images. These images were obtained by discretizing the 2D Gaussian equation at each pixel location. The size of the image represents a cropped region of a larger image, or ROI. For each image, the standard deviations in both directions, as well as the rotation of the principal axes, were randomized. These values were used to compute the sigma matrix. The subpixel peak location was also randomized within one pixel of the center. Using the MATLAB built-in mvnpdf() function, a normalized 2D Gaussian was generated using the sigma matrix, and the subpixel peak passed as the mean. The outputs were multiplied by 65,535 and cast as 16-bit integers to represent a 16-bit grayscale image.

There were several different cases applied for each image. First was the control, where the subpixel peak location was forced to the center of the image. This is denoted as centered. Second, the off-center image was generated using the randomized, subpixel peak location. The remaining images each applied additional image processing to the off-center image. The saturated case multiplied each pixel value by a factor of four in order to saturate one or more pixels to a value of 65,535 and simulate a longer integration time. The noisy case applied Gaussian white noise to the image. The mean of the noise was zero, while the variance was a random value within a specified

range. Finally, the saturated, noisy image applied noise with the same mean and variance to the saturated case.

All of these images were run through a Monte Carlo simulation with the following parameters. The filter order controlled both the size of the filters and the size of the image. In practice, the ROI could be cropped to a size that represented the filter order. The standard deviations of the Gaussian image were restricted to between one and a maximum value. The rotation angle of the image was allowed to vary between 0 and 45 deg on a uniform distribution. The noise variance was picked randomly from a normal distribution with a fixed mean and standard deviation. Histogram equalization and Gaussian filtering could also be toggled. However, these two processing steps did not appear to make much of a difference and were turned off after a few initial runs.

RESULTS

It took several iterations to tune the range of the Monte Carlo parameters. First, the histogram equalization and Gaussian filtering appeared to make little difference. These were turned off to simplify the problem. Since the images were idealized generalizations, it is possible that these filters may have an effect in practice and should not be discarded simply due to this result. The next observation was that matching the filter order to the 2D Gaussian standard deviation seemed to give the best results. To simplify the number of parameters further, the filter order was fixed to three (resulting in a 7 x 7 pixel image) and the standard deviation was allowed to vary between one and five. The camera resolution, size of the infrared target, and distance from the target are factors that could affect this parameter. Finally, the noise level was tuned by eye based on the values that resulted in a noticeable, but not overwhelming, level of noise. This is entirely subjective, but the parameters that were settled on were a mean of 0.003 and a standard deviation of 0.001 from which to select the noise variance.

After these parameters were established, a series of Monte Carlo simulations was performed with increasing numbers of runs. This was done to identify convergence in the solution. For each run, five images were generated, and each of the five algorithms was applied to each image. The differences in the x and y directions between the true peak and the estimated subpixel peak were recorded for each case. At the completion of the simulation, the standard deviation of these differences was computed for each combination. It was assumed that the x and y directions would have the same standard deviation since the directions in this experiment were arbitrary, so they were lumped as a single set. Note that they were used to compute a radial distance, as that would preclude the computation of a standard deviation due to the distribution not being normal. A single run of the Monte Carlo simulation can be seen in figure 2.

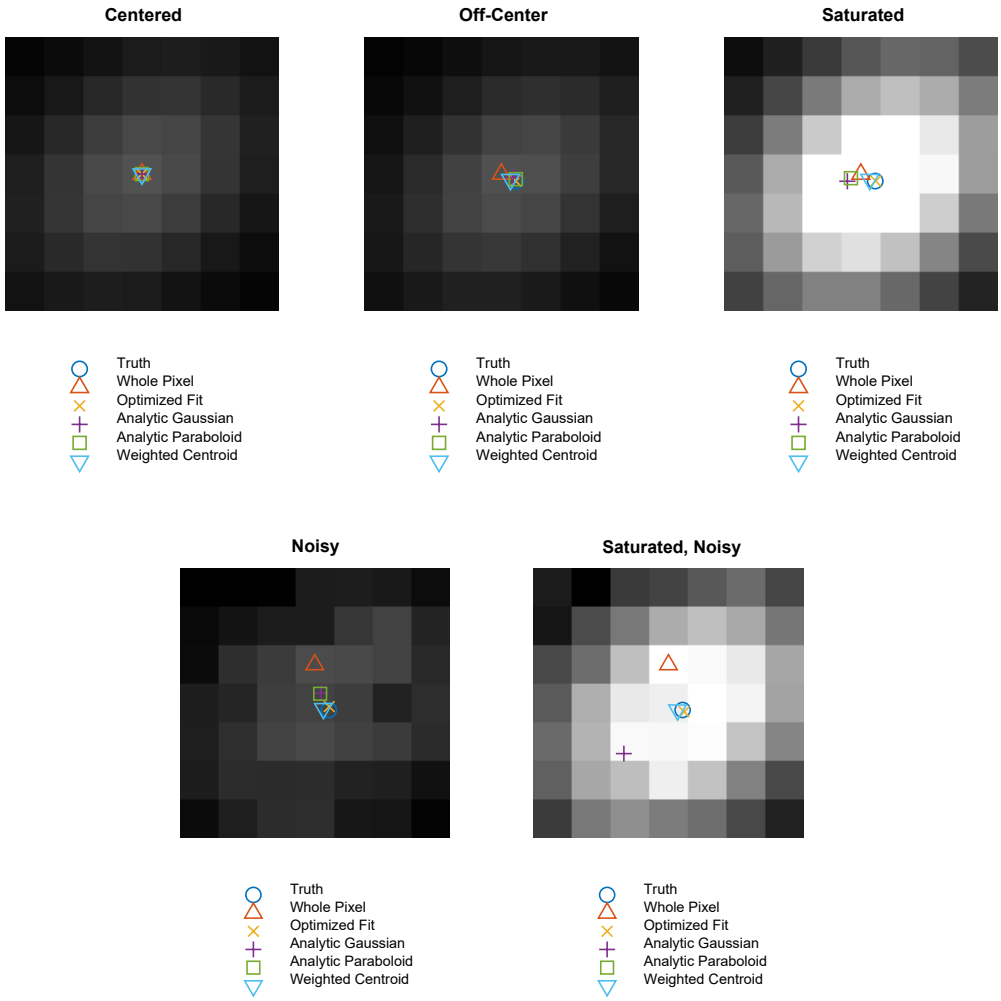

Figure 2
Sample result for a single run of the Monte Carlo simulation

To understand when to stop the series of simulations, the convergence was monitored. The criterion for convergence was a difference of 1% from one run to the next. Since there were 25 standard deviations total, only the maximum standard deviation for each algorithm was monitored. However, it was noted that all followed the same convergence trends within an algorithm group. Figure 3 shows a plot of the convergence as the number of runs increases. Note that only the weighted-centroid and whole-pixel methods meet the convergence criteria established earlier. The optimized Gaussian fitting (optimized fit) algorithm appears to be converging as well and may have come close at 1e5 runs. However, that algorithm runs too slowly to complete more than 1e4 runs in a reasonable period of time. Therefore, it will have to be accepted that the result for the optimized fit may not be as accurate. The analytic Gaussian and analytic paraboloid do not converge. The results for 1e5 runs (except for optimized fit, for 1e4 runs) are shown in table 1.

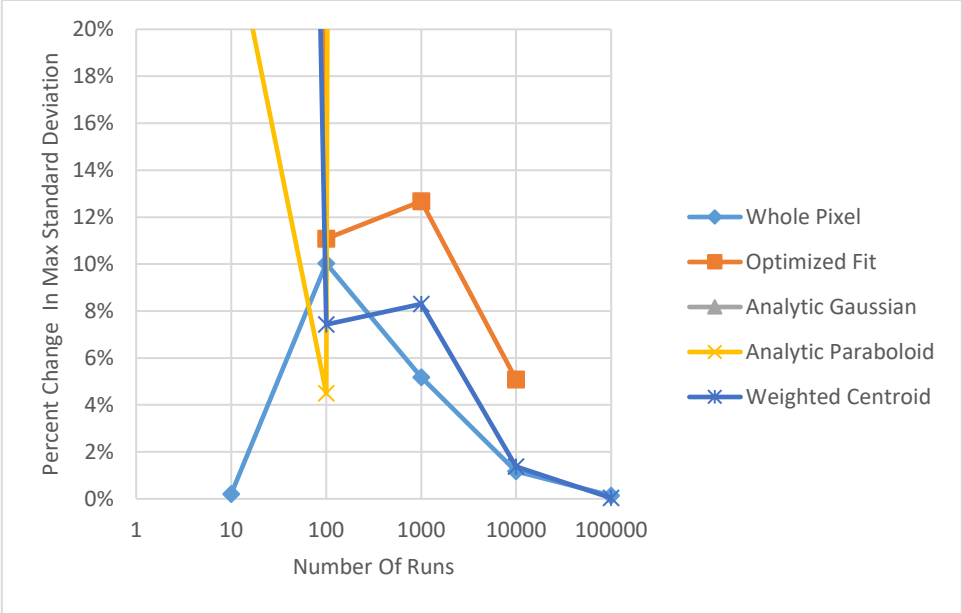

Figure 3
Convergence of point-source estimation methods with increasing number of runs

Table 1
Standard deviation of center-estimation error in pixels

	Whole pixel	Optimized fit	Analytic Gaussian	Analytic paraboloid	Weighted centroid
Centered	0	3.1E-4	4.08E-12	1.52E-14	0
Off-center	0.3123	5.609E-3	17.2100	25.9103	0.1737
Saturated	0.3150	0.0255	16.0922	33.7500	0.1753
Noisy	1.2810	1.4398	1,480.485	85.1235	0.2315
Saturated, noisy	0.7879	0.3529	37.9967	11,050.1	0.1803

DISCUSSION

For the images where the peak was at the whole-pixel center, most of the algorithms found the center exactly. The only algorithm that did not always match the center exactly was the optimized Gaussian-fitting algorithm. This is likely because the use of optimization meant that it was the only nondeterministic algorithm, and certain values of random seeds may have caused some slight misses. Regardless, the standard deviation of the error was still very low.

When the 2D Gaussian was moved off center, the performance of the algorithms began to diverge. The whole-pixel-estimate standard deviation was very close to the expected value. That is,

one standard deviation would be expected to capture 68% of the distribution. Since the center was moved 0.5 pixels in any direction, the expected standard deviation of the whole-pixel estimate would be 0.34 pixels. The actual reported value was closer to 0.312 pixels. It is possible that some other small effects such as the quantization of the pixel intensity values played a role in this difference. Against this benchmark, only the optimized Gaussian-fitting algorithm and weighted-centroid algorithm had better performance. As seen in the other cases from here out, the analytical Gaussian and analytical paraboloid fits had astoundingly high standard deviations. In fact, the error only grew greater with more runs, signaling that the solution itself was divergent. This is interpreted to mean that these algorithms were extraordinarily sensitive to the locations of the pixel boundaries. The optimized Gaussian-fitting algorithm performed best, followed by the weighted centroid.

The failure of the analytic solutions to provide consistent results is worth further discussion. Clearly, based on the centered results, the formulation is working as intended to some degree. Once the image is no longer symmetric, the mathematics appear to fall apart. Upon closer inspection, the majority of individual results are actually comparable to those of the other algorithms. However, it appears that the analytic Gaussian and paraboloid algorithms, giving equal weight to the whole image, can be thrown off by the values on the outside of the image. This causes outliers that can be tens if not hundreds of pixels off, significantly affecting the standard deviation. It may be possible to resolve this issue through a different set of operators, but no resolution was discovered within the scope of this experiment.

When adding some saturation to the uncentered Gaussian, the whole-pixel benchmark standard deviation remained the same, as expected. The level of saturation was not enough to fully saturate more than one pixel and throw off the estimate. However, the optimized Gaussian-fitting algorithm's performance dropped, though the performance was still much better than the weighted centroid. The weighted centroid was unaffected by the small increase in saturation.

The fourth set of images was generated by adding noise to the off-center Gaussians. This dramatically reduced accuracy across the board. It should be noted that the amount of noise was subjective, so the results would be expected to vary if the mean of the noise were greater or lesser. For comparative purposes of moderate to high levels of noise, this set of images should suffice. Most interestingly, the whole-pixel estimate was off by more than a pixel due to the noise. This is likely because the noise caused pixels close to the center to have values greater than those at the center did. This also had a major effect on the optimized Gaussian-fitting algorithm, which uses the whole-pixel estimate as its initial guess. In fact, the optimized Gaussian-fitting algorithm performed worse than the benchmark with noise introduced. The weighted-centroid algorithm suffered a reduction of accuracy, though minor relative to the accuracy lost by the other algorithms upon the introduction of noise; this is the only algorithm with a standard deviation below one pixel in this case. The results for the weighted centroid should be verified, as they are reliant on being centered by the whole-pixel estimate. A follow-on investigation may be necessary to understand how much accuracy is lost if the weighted centroid is not centered within one pixel. Likely, this loss will be minimal, as the area around peak will retain the greatest weight.

The final set of images added noise to the saturated Gaussian. The expectation was that this would be the worst case, though surprisingly, it performed better than the noise on the unsaturated Gaussian. The whole-pixel estimate was still degraded, but the optimized Gaussian-fitting algorithm was able to improve the accuracy above that of the initial guess. The weighted-centroid accuracy was only very slightly reduced from the nominal case, beating out the optimized Gaussian-fitting algorithm. It is theorized that amplifying the values in the image, while keeping the level of noise the same, resulted in a higher signal-to-noise (SNR) ratio. This effect was enough to overcome the small detriment due to saturation. Since only a single level of noise and saturation was tested here, it is possible that these results could vary. However, it is interesting to note that allowing for some saturation could in fact improve accuracy.

CONCLUSIONS

Based on the results of this experiment, it is possible to quantify the performance of various subpixel peak detection algorithms. The results were kept general enough to apply to any object viewed in infrared (IR) that approximates a point source. Assuming the object can be identified and the image cropped to an appropriately sized region of interest, the results are agnostic of image resolution, distance to the object, and type of object. One can surmise that in the case of a perfect image, the optimized Gaussian-fitting algorithm is the most accurate algorithm by far. Once noise is introduced, the weighted-centroid method can be more accurate, depending on the level of noise. Saturation plays a role in accuracy reduction; however, trading some saturated pixels for a higher signal-to-noise ratio (SNR) looks to be well worth it for all but the lowest levels of noise. A higher SNR would allow for detections of smaller objects or at longer ranges, while taking only a small loss in accuracy. Therefore, if one desired the most accurate subpixel peak location in practice, one might consider adjusting the integration time to the edge of saturation. Since the optimized fit is so much slower than the analytic methods, its use should be restricted to the cleanest of imagery or as a redundant method when speed is not important. The analytic Gaussian and analytic paraboloid should not be considered in their current form, but perhaps the derivation could be used to build less-sensitive operators. Overall, the weighted-centroid algorithm performed acceptably in all cases detecting subpixel peaks. Further research should focus on the effect of a greater offset on the weighted-centroid algorithm as well as additional cases of deleterious effects on the image, such as dead pixels and median filter masking.

REFERENCES

1. Anthony, S. M. and Granick, S., "Image Analysis with Rapid and Accurate Two-Dimensional Gaussian Fitting," Langmuir, Vol. 25, Issue 14, pp. 8152-8160, 2009.

2. Fisher, R. B. and Naidu, D. K., "A Comparison of Algorithms for Subpixel Peak Detection," Advances in Image Processing, Multimedia and Machine Vision, Springer-Verlag, Heidelberg, 2001.

3. Feldman, J., "Taylor Expansions in 2d," The University of British Columbia Department of Mathematics, <http://www.math.ubc.ca/~feldman/m200/taylor2dSlides.pdf>, Accessed March 2021.

4. Sobel, I. and Feldman, G., "An Isotropic 3x3 Image Gradient Operator," Presentation at the Stanford A.I. Project in 1968, <https://www.researchgate.net/publication/239398674_An_Isotropic_3x3_Image_Gradient_Operator>, February 2014.

5. Kreider, K. L., "Differentiation Formula CD4," The University of Akron Department of Mathematics, <http://www.math.uakron.edu/~kreider/num2/CD4.pdf>, Accessed March 2021.

6. “MATLAB,” Release 2018b, MathWorks, 2018.

DISTRIBUTION LIST

U.S. Army DEVCOM AC
ATTN: FCDD-ACE-K
FCDD-ACM-AA, S. Manole
R. Decker
Picatinny Arsenal, NJ 07806-5000

Defense Technical Information Center (DTIC)
ATTN: Accessions Division
8725 John J. Kingman Road, Ste 0944
Fort Belvoir, VA 22060-6218

GIDEP Operations Center
P.O. Box 8000
Corona, CA 91718-8000
gidep@gidep.org

REVIEW AND APPROVAL OF ARDEC REPORTS

THIS IS A:

- [x] **TECHNICAL REPORT**
- [ ] **SPECIAL REPORT**
- [ ] **MEMORANDUM REPORT**
- [ ] **ARMAMENT GRADUATE SCHOOL REPORT**

FUNDING SOURCE ILIR
[e.g., TEX3; 6.1 (ILIR, FTAS); 6.2; 6.3; PM funded EMD; PM funded Production/ESIP; Other (please identify)]

High Precision Center Estimation of Point Source Infrared Targets
Title

ILIR
Project

Steven Manole, Ryan Decker
Author/Project Engineer

Report number/Date received (to be completed by LCSD)

X2744
Extension

94
Building

FCDD-ACM-AA
Author's Office Symbol

PART 1. **Must be signed before the report can be edited.**

a. The draft copy of this report has been reviewed for technical accuracy and is approved for editing.

b. Use Distribution Statement A X, B____, C____, D____, E____, or F____ for the reason checked on the continuation of this form. Reason: _____

1. If Statement A is selected, the report will be released to the National Technical Information Service (NTIS) for sale to the general public. Only unclassified reports whose distribution is not limited or controlled in any way are released to NTIS.

2. If Statement B, C, D, E, or F is selected, the report will be released to the Defense Technical Information Center (DTIC) which will limit distribution according to the conditions indicated in the statement.

c. The distribution list for this report has been reviewed for accuracy and completeness.

Edward Bauer
Division Chief (Date)
7/21/21

PART 2. To be signed either when draft report is submitted or after review of reproduction copy.

This report is approved for publication.

Christopher Stout
Division Chief (Date)

RDAR-CIS (Date)

LCSD 49 (1 Sept 16)
Supersedes SMCAR Form 49, 20 Dec 06