

Laplacian-Gaussian Method of 3D Surface Extrapolation, from a Set of 2D, Shallow Depth of Field Images in MATLAB

Paul Killam
Moorpark College
Moorpark, CA
pskillam@umail.ucsb.edu

Mentors: Rohit Bhartia, Bill Abbey, and Greg Wanger
Jet Propulsion Laboratory - NASA
La Cañada Flintridge, CA

Abstract

The purpose of this paper is delve into the approach taken and challenges overcome to develop, in the vector-optimized MATLAB programming language, an algorithm which creates a three dimensional surface from a macrophotography-style stack of images - a set of images taken from the same point, but whose focal distance varies - to allow three dimensional surfaces to be extrapolated using only a single optic. This technique was developed for use in tandem with instruments, such as Deep UV Raman and Fluorescence instruments, and potential use on SHERLOC on Mars 2020. The program uses a Laplacian Mask as a method of focus-sensitive edge detection, as well as median noise filtering both before and after the Laplacian Mask is applied. A high-pass filter is then applied to that data, and either a Maximum or Gaussian fit is used to identify the distance to different points on the pictured surface. The gaps in the data are then interpolated using the natural neighbor method. It was found that a three dimensional surface can be almost perfectly estimated by this program to within one-half of the inter-image distance with the slower Gaussian method, and to within one full inter-image distance with the faster Maximum method. This program will aid the SHERLOC team in correlative microscopy, the data it provides being used to select target spaces, allow for minute adjustments in focus to be made, and to determine which areas of a sample are out of range or cannot be used due to their topography. Though 3D-extrapolation is not a new concept, this program is custom-fit to the needs of SHERLOC, and as opposed to proprietary methods out there, can be modified as needed to include new instruments, and the changing demands of SHERLOC in the future.

1. INTRODUCTION

A common challenge which comes up whenever optical data is collected, is how to correlate the two-dimensional data with the no doubt three-dimensional object it was extrapolated from. To do this, a three-dimensional model of the object, or at least the relevant three-dimensional surface, must be generated; but how to do this, and on a budget no less? The classic way of approaching this problem is the same method used both by astronomers to calculate the distance to the nearest stars, and the one we as humans use to judge distance in our everyday lives: parallax. This method relies on having either two optical systems, or at the very least two vantage points from which the object in question is observed. Using image comparison and some trigonometry, a three-dimensional surface can be extrapolated, but there is another way of approaching the problem.

That same three-dimensional surface can be derived with a single optic, from a single vantage point. This statement may arouse suspicion in those locked in the mindset of how to approach this with human eyes, but there is an aspect of mechanical lens systems that can be exploited: focus. A camera has to focus on an object to get a crisp picture, and that point of focus is always a set distance from the camera. If the depth of focus is shallow enough, meaning that to be in-focus,

an object must be placed within a very small margin, then this relationship between focus and distance can, once multiple images are taken, be used to estimate a surface. The development of a function in MATLAB to extrapolate a three-dimensional surface from a set of pictures, whose only variable is the point of focus, will be the topic of this paper.

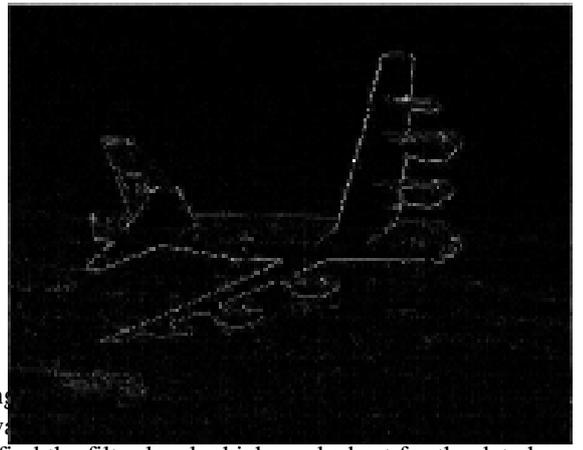
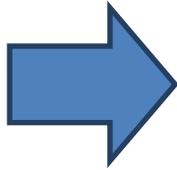
2. METHODS

I. THE ABSOLUTE LAPLACIAN

The approach to unraveling this relationship is two-fold. Firstly, a method must be derived to quantify how in-focus a pixel in a picture is, and secondly, the data from the first technique must be analyzed so the proper “peak of focus” can be found. Once that peak of focus is found for each pixel on the z-stack of images, a three-dimensional surface can be estimated based on where each z-pixel on the z-stack is in focus.

The method selected to refine the raw images into focus data was the Absolute Laplacian:

$$|L[f(x, y)]| = \left| \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right|$$



in edge-detection algorithms, as it is very sensitive to sudden changes. Since the Laplacian is related to the second derivative operator, its magnitude increases greatly as it passes over an edge. If that edge is blurred however, the Laplacian does not achieve the same intensity as before because the change caused by that edge is more gradual, and its onset is less sudden, meaning the second derivative is smaller. Therefore, if one were to apply an Absolute Laplacian to an edge as it goes into and out of focus, the resulting function would increase, then decrease. This increase and decrease happens to roughly resemble a Gaussian Curve as seen in figures two through four.

There is an issue however. As one can see in the examples above, the fidelity of the peak of focus for a point degrades as the maximum value of the Absolute Laplacian decreases for a particular point. (In case the pictures are too small, the scale for Fig.2 reads to 12, Fig.3 reads to 9, and Fig.4 reads to 4.5). Therefore, some method must be derived to either enhance or remove these low-fidelity points.

II. HIGH-PASS MAXIMUM FILTERING

The method chosen to deal with these low fidelity points was simply to create an adjustable high-pass filter, removing any data point whose maximum value for the Absolute Laplacian did not surpass the value set by the user. Ideally, this value would be set to a level so a data set such as in Fig.3 might barely make it through, and a data set such as in Fig.4 would be eliminated. These gaps would later be extrapolated over using the natural neighbor method, but this will be covered in more depth in section 2.5.

Allowing distinct advantage to the user to find the filter level which works best for the data he has to work with, and allows him to experiment with different values to find the best result. Secondly, this experimentation with different filter settings might allow the user to find new details in his data, which might be missed if only a single, preset filter setting was used. Finally, it allows the user to determine how much precision to trade off for detail, or vice versa.

This relationship is due to the fact that the more data points one allows through (lower filter setting), the less fidelity each point has on average. This causes some "spiking" to occur, making small imperfections in flatter surfaces become exaggerated and deformed in the end result, but it also greatly increases the detail of sharp edges and drops in the sample. The opposite is true when the filter value is set higher, the detail of flatter surfaces becomes more and more precise, while sharp edges become blurred and smudged as the data points that comprise them are extrapolated across.

This would mean that the user must run the program twice, once at a low filter setting, and once at a high filter setting, to be able to see all the proper detail of a very dynamic surface. However, a method of adaptive filtering will be added to the program as soon as possible, which will allow the program to adjust the filter lower for low-contrast areas and edges, while increasing it for higher-contrast areas and flat but detailed surfaces. This will allow the function to be run without needing to select a filter setting at all, and result in much better surfaces where neither sacrifice of detail nor precision is

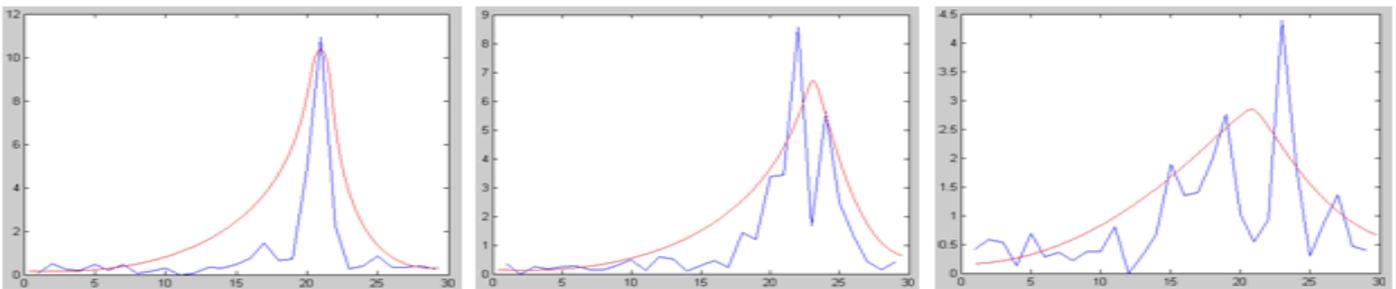


Fig. 2, 3, 4: Examples of the Gaussian nature of the result of the Absolute Laplacian with regards to focus.

III. GAUSSIAN FIT ALGORITHM

Two methods have been derived so far to make use of the data gained from applying the Absolute Laplacian mask to our data. The first one comes naturally. Once it was known that the data is Gaussian in nature, it is easy to conclude that one can fit a Gaussian curve to the data and calculate the center, denoting that as the point of focus. This detection of the natural function behind the set of data allows us to pick out far more precisely the estimated peak, and therefore, the exact index or distance from the camera, for every point which passes through the high-pass maximum filter. This allows for the sub-index peak detection alluded to in the abstract.

There is yet another challenge to overcome however. Using the Gaussian fit algorithm provided by MATLAB appeared to create its own extreme peaks and drops, either well above or well below what could ever be thought of as reasonable for the surfaces it was being tested on. In particular, this problem appeared most frequently when dealing with data points whose true center of focus lies close to or on, the maximum or minimum index. Once this was identified, it was easy to determine that the problem was originating when, instead of looking like a Gaussian curve, the data resembled either an exponential growth or decay, similar to the beginning or end of a Gaussian curve whose peak is far off. This was swiftly remedied by adding "control points" to the set of data passed to the Gaussian fitting function. Three of these control points were appended before the first point, and after the last point, each one being $1/2$, $1/4$, then $1/8$ of the data point it was appended to. This simulates the continuing of a Gaussian curve, and allows the Gaussian fit function to perform its task without error.

Though the Gaussian Fit method does an excellent job estimating the surface being examined, it does so very slowly. Even though a significant portion of the data points are being removed by the High-Pass filter, the Gaussian fit is still applied several hundred thousand times for the 2048x1536 images used, each surface extrapolation taking well over an hour to complete. Attempts were made to reduce the time taken to complete this task, including further point-filtering and pseudo-Gaussian algorithms, but these measures significantly reduced the effectiveness of the Gaussian fit algorithm, and thus were scrapped. The results of this method can be seen in figure 5 below.

IV. MAXIMUM FIT ALGORITHM

To conquer the issue of time-efficiency, it was decided to experiment with a method first used in the proof-of-concept phase of development, simply choosing the index of the highest "in-focusness" value for each z-pixel, after the Absolute Laplacian mask is applied. Though this method, in theory, has less potential precision than the Gaussian fit algorithm, its sheer computational efficiency makes it both easier to work on, and work with. As opposed to the around 90 minute wait for a surface to be extrapolated using the Gaussian

method, the same surface extrapolated using the Maximum Fit method only took 6 minutes, and later on, near 3.5 minutes after the algorithm was optimized. As such, more time was dedicated to making improvements on this Maximum Fit algorithm, both in terms of time efficiency and precision.

Of the many attempts at making the Maximum Fit algorithm more precise, the most notable was that of the selection of certain "points of interest", single points within a z-pixel whose intensity is high enough to be used to find the actual peak of the data, without fitting it to a Gaussian curve. Initially, these points were selected using yet another high-bypass filter, this time acting within the z-pixel. This however, did not take into account the relative intensities across several z-pixels, and was swiftly changed in favor of a filter set as a percentage of the maximum intensity. A method based on the standard deviation of the intensities of each point in a z-pixel was tried, but the computation of the standard deviation over several million z-pixels depreciated the main advantage of the Maximum Fit method, its time efficiency. Once these points were selected, among other methods, a weighted average was used to compute the real peak of the set of data, but after a good amount of exploration into the use of "points of interest", it never resulted in an algorithm with higher precision. If an algorithm that does result in higher precision for peak finding is found however, it may provide further improvement of the Maximum Fit algorithm as a whole.

In the end, the Maximum Fit algorithm has been selected as the default method for surface extrapolation, allowing the user to select the Gaussian method as an option. The single inter-image distance precision of the Maximum Fit algorithm was deemed acceptable for the surface mapping to be done in the lab. The result of an earlier version of this algorithm can be seen in figure 6 below.

V. NATURAL NEIGHBOR INTERPOLATION

One result of the use of the High-Pass filter's removal of lower quality data points, mentioned in section 2.2, is that once all of the maximum points are computed, there are still holes in the data that must be filled to allow the three dimensional surface to be displayed. Initially, a simple Linear Interpolation method was used for just that reason, to allow the end-result to be displayed so different iterations of the program can be compared to one another. This linear interpolation served the project well as it was going through the proof of concept phase of development, however it soon became clear that this method needed to be replaced to improve the quality and accuracy of the end result.

Thankfully, there is another interpolation function built into MATLAB, one that uses the natural neighbor method of surface interpolation. The natural neighbor method of interpolation assigns weights to each of the points around the point being inserted, resulting in a much smoother surface as opposed to the linear interpolation method, at the tradeoff of

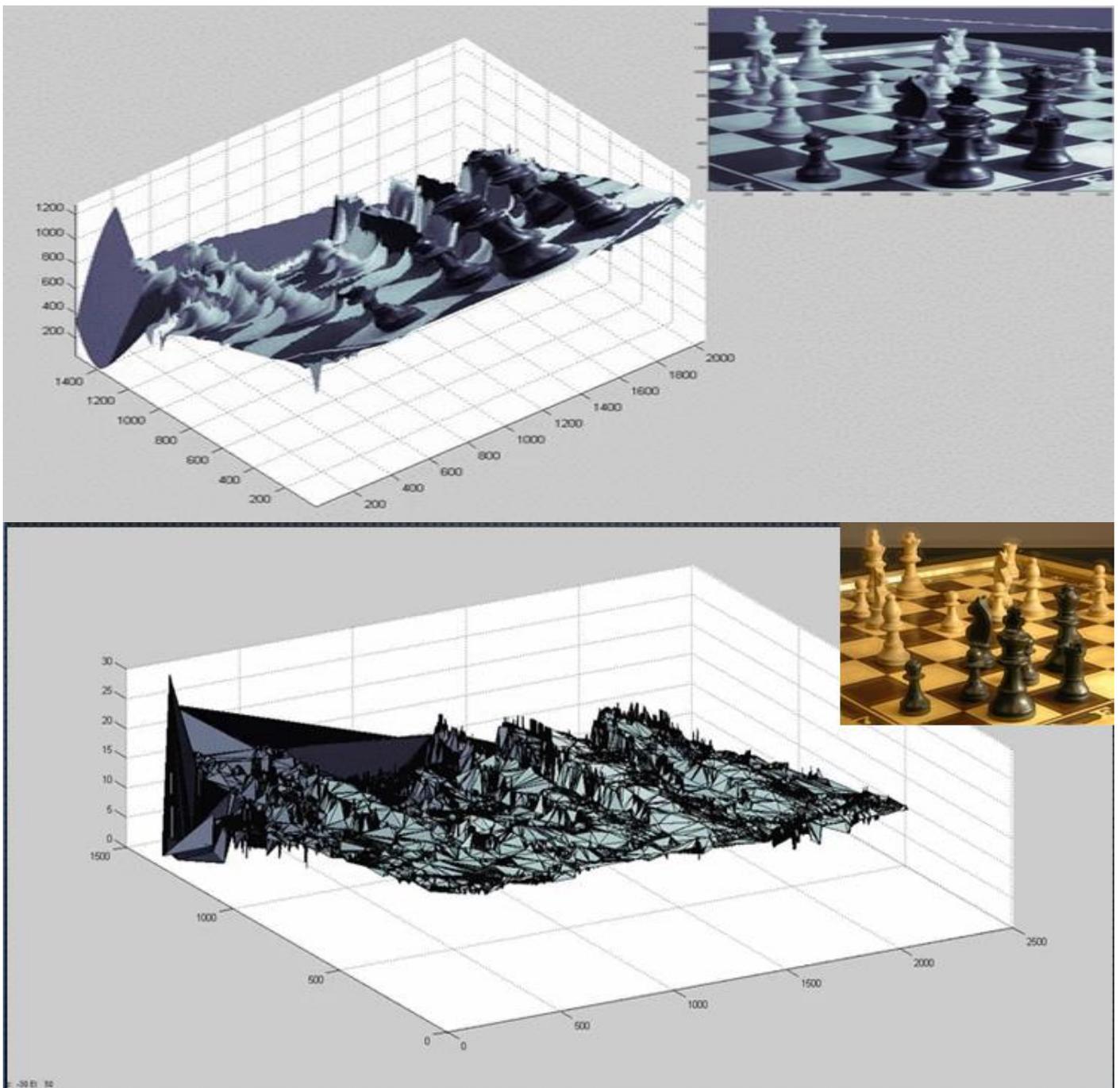


Fig. 5 (top) The result of a high-detail Gaussian peak match at each data point

Fig. 6 (bottom) The result of an early version of the high-speed Maximum peak match at each data point

using more computation time. However, this added time to the end program was insignificant compared to the millions of times the peak-finding algorithms ran, even the Maximum Fit algorithm.

This natural neighbor interpolation algorithm allowed the resulting surfaces to more closely resemble those of the actual object being imaged, fleshing out those areas of lower fidelity that have been left out by the high-pass filter.

VI. ADDING MACROPHOTOGRAPHY

To finish off the project and make the resulting surfaces easier to analyze by the naked eye, the concept of macrophotography was taken out of the realm of just inspiration, and actually introduced as a feature. The initial pictures read into the program were kept in memory until the surface was extrapolated, then pixel by pixel, portions of each image were laid onto the surface according to their index, rounding up or down to the nearest integer according to the “altitude” of the surface at that point.



Fig. 7a (left) A full-focus image reconstructed by a typical macrophotography method.
 Fig. 7b (right) A full-focus image reconstructed using height mapping to the extrapolated surface.

This method of macrophotography turned out to be very successful in the end, resulting in even a better image than the end product from the macrophotography website in which the test images of the chess set were found. Using the actual interpolated surface also avoided the “halo” effect which plagues many different macrophotography methods; this can be seen in figure 7

3. RESULTS AND DISCUSSION

Two methods of creating a three-dimensional surface from a series of images have been developed from this project. The Gaussian method allows for sub-inter-image-distance precision with a run time of about 90 minutes, and the Maximum-Fit method allows for a precision to around a single inter-image-distance with a run time of only 3.5 minutes. A method to exactly quantize the error of these surfaces has not been developed however, so this error has been approximated using knowledge of the surface and rudimentary analysis by-eye. The runtimes on the other hand are based off a mid-range laptop, and would certainly be a lot shorter if run on a computer better designed to handle a large amount of computation; the ratio between the two times however, remains relevant for comparing the two algorithms.

More optimization is always possible, especially with the Gaussian method, as most of the optimization done so far has been to improve the Maximum-Fit method. This is due to the Gaussian method's already long runtime, making any attempt of iterative optimization take longer than is practical to get a good amount of work done.

One drawback in particular which will be confronted is the high-pass maximum filter used to remove data points. As explained earlier, this method of a constant filter does not adapt to a dynamic surface very well, leaving some areas smoothed over and others far too detailed and sharp. A solution for this,

which has not yet been implemented, is an adaptive high-pass filter. Utilizing some method of analyzing the "quality" of each area of the image, an adaptive filter would be able to allow more or fewer data points through to either provide more data where it may be smoothed out, or less where the average quality goes down and causes noise. This adaptive filter would remove the need to make multiple passes over a single sample, allowing for much smoother operation.

ACKNOWLEDGEMENTS

I would like to sincerely thank Professors Erik Reese and James Somers from Moorpark College for connecting me with this opportunity, as well as Professor Paul McCudden from Los Angeles Community College for creating it and managing it so well. Most of all I would like to thank Dr. Rohit Bhartia, Dr. Greg Wanger, and Dr. Bill Abbey from JPL, the mentors that taught me so much, and created such an amazing collaborative environment to make this internship at JPL even more incredible than I could imagine. I would also like to thank the National Science Foundation for making this possible with its generous contributions to the Consortium for Undergraduate Research Experience, giving many other community college students the opportunity to have such great and inspiring summers at JPL.

REFERENCES

- [1] http://cis.cs.technion.ac.il/Done_Projects/Projects_done/VisionClasses/Vision/Edge_Detection/nod_e5.html
- [2] <http://www.cs.cornell.edu/~erenrich/dof>

This work was supported by National Science Foundation grant #AST-1156756 to Los Angeles City College.