

Defocus Magnification

Jean Forde

Charles Gorman

CS 534 Computational Photography – Spring 2011

University of Wisconsin – Madison

<https://mywebspace.wisc.edu/forde/CS534/index.html>

ABSTRACT

Shortening the depth of field of an image is often desirable for creating artistic images and portraiture. Due to the limitations of many compact cameras, this can not always be done using hardware. We compare two different methods of adding blur to an image to simulate shortening the depth of field of an image. One technique builds on method which uses Elder-Zucker blur estimation to accurately recreate a blur kernel. The other method was a novel approach using edge detection to create an initial blur map and additional analysis of the initial blur map to create an optimized map. Both methods require minimal user input, and neither method requires accurate estimation of absolute image-object depth.

1. INTRODUCTION & MOTIVATION

Creating photography with a shallow depth of field is desirable for many applications in both creative and artistic photography. The ability to create image defocus is, however, limited by what type of camera is available to the photographer. Point-and-shoot cameras, for example, have a lens and sensor which are too small to create large amounts of artistic background blurring. This means that users of compact cameras will be unable to create the sharp portraits possible with SLR cameras. The goal of our algorithm is to simulate the effects of a

shallow depth of field, and thereby give pictures a more desirable "artistic" blurring.

We compare two algorithms which increase defocus given a single input image. In both, the blur at scene edges is estimated to determine the scale of blurring. These estimates are then propagated out to every scene pixel, giving a complete scene-blur estimation map. Using these maps, defocus can be magnified in areas which contain large amounts of defocus, thus which approximates a shortening of the depth of field.

2. RELATED WORK

Much of the related work which solves similar problems is concerned with recovering three dimensional data from an image. This usually means taking several images, and using the varying amounts of blur to estimate the depth at different points of interest. Most of them also require additional information about the focal length and focal distance to calculate depth in a non-arbitrary way. Methods proposed by Pentland [Pen87] and Favaro and Soatto [FS05] both use multiple scene images, with different lens settings, to estimate scene depth. Eltoukhy and Kavus' technique [EK03] uses a similar algorithm and multiple scene images to extend the image's depth of focus. These methods are all limited by requiring multiple images and additional knowledge about camera settings

to create an accurate reconstruction of image-object depths.

Another class of related algorithms seek to accurately measure image blur to improve image sharpening techniques. In particular, an algorithm by Reeves and Mersereau [RM92] uses steerable Gaussian filters to generate a blur kernel, which is used to sharpen an image. While the goal of our algorithm is to increase image blur, the technique for estimating the amount of blur is very applicable.

A method published by Bae and Durand [BD07] is the one we seek to improve upon. This method attempted to increase defocus blur by constructing a blur map from a single image, and using that map to guide image blurring to produce the final result. This algorithm is limited both by its inconsistent results, and by its need for outside software to perform the image blurring.

3. PROBLEM STATEMENT

Evaluate and improve upon Bae and Durand's Defocus Magnification [BD07] and/or design a new method to enhance defocus blur using a single image with no user input.

4. OVERVIEW OF BAE AND DURAND'S METHOD

Bae and Durand's method attempts to magnify image defocus effects by accurately modeling the amount of scene blur, and increasing it. First, they find and detect the blurring across edges. They estimate the blur by assuming that blurring across edges is modeled by an intensity step function blurred by a Gaussian filter. They use a modified version of the edge-blur estimation technique proposed by Elder and Zucker [EZ98] to create their initial model. The blur

estimate is then propagated away from the edges by assuming that regions with similar color and intensity values will have similar amounts of blurring. Using the generated blur map, their next step was to use Adobe[®] Photoshop's[®] 'Lens Blur' feature to blur the image.

4.1. Our Reimplementation of Bae and Durand's Method

The original implementation of Bae and Durand's Method [BD07] was limited by its use of an image-intensity sensitive method of blur propagation, a large number of user supplied variables, and the lack of a MATLAB[®] inclusive method of applying the blur kernel to produce a blurred output. Our implementation fixes these limitations, and provides qualitatively improved results.

Given the input image, a modified version of the original Elder-Zucker algorithm is applied to estimate the amount of blurring at the edges [EZ98]. The edges are found using the Canny edge detector, and a steerable gradient function is applied to determine the direction and intensity of the edge. From this, the Laplacian of a Gaussian is estimated, and used to estimate the distance between extrema of the image's second derivative. Based on this distance, the amount of blurring across a given edge is estimated.

The algorithm then proceeds out from the edges, calculating the blur farther and farther from the original edge estimates using fixed width neighborhoods. The amount of blurring was assumed to be constant in regions with similar color and intensity values, meaning the blur estimate was calculated using a weighted average of already calculated blur estimates weighted by pixel similarity.

At each neighborhood, the estimate is blurred slightly to even out any errors generated by the local scope of the image analysis, resulting in a smoothly varying blur estimate at every pixel. The estimate is then normalized and remapped to fit the MATLAB[®] unsigned 8-bit integer scale range, and separated out into different divisions based on intensity, and converted into logical masks for use with MATLAB[®] region of interest filtering. The original image is then blurred, beginning with regions with the least initial blurring, and moving out toward regions of the most blurring. This results in areas of the most blur being blurred more than regions which were initially sharp. This gives the desired artistic effect of altering the focal length of the camera without the need for multiple images or additional camera information.

5. THEORY

Our theory is based upon Bae and Durand's method and the guiding theory that the subject of the image has more detail than the remaining parts of the image.

Based on this premise, we find the least detailed areas or the blurred areas and apply additional blur leaving the subject or sharp edges untouched.

6. METHOD

We pre-process the image to eliminate potential items that could skew the blur map, create the blur map to identify areas that should be in focus and should be blurred. Finally, we blur the image using a disk filter and output the result. This process does not require any user input except for the initial image.

6.1. Pre-Processing

It is important to note that the image should not be resized prior to processing as resizing will dramatically reduce the blur map quality.

We converted the image to grayscale and then processed it to remove any lighting related shadows or highlights as these may appear as detail or edges in the following steps. To flatten the lighting, we used the Frankle-McCann Retinex [FM83] method and used the implementation that was created by Funt, Ciurea, and McCann [FCM00]. As the goal of this step is not to make the image aesthetically pleasing, but to flatten any lighting, we found that 60 iterations worked for the majority of images including those that had very strong lighting issues and those that were lighted fairly evenly.

We then ran a light Gaussian filter over the image to reduce the extreme high frequency areas.

6.2. Create the Blur Map

To create the blur map, we used MATLAB[®]'s 'Canny' edge detector. This edge detector was based off John Canny's approach [Can86] which uses three criteria to ensure the most accurate edge map is created, while only displaying the detection of each edge one time.

Once the edges have been located, we blur the image using a strong Gaussian filter and adjust the levels of the image so that they span from 0 to 255 in intensity.

With this initial blur map, we run an averaging filter over the image a number of times using different neighborhood sizes to determine when the overall minimum and maximum averages occur. We then

combine the minimum and maximum maps, with the minimum weighted more heavily. This map provides a good general overview of the most important edges.

We then optimized this map by finding the areas that contained the most detail and increased the edges in these areas. Using a number of test images, we determined thresholds that work decently for the majority of our test images. This resulted in a map that allowed for the image subject to have relatively flat areas and still remain in focus.

6.3. Apply the Blur

We enlarged the initial image by duplicating the outside edges and the blur map by indicating that the area should be blurred to avoid any framing issues.

We then applied a number of Gaussian filters for each color channel at successive levels in the blur map.

The final step cropped the image back to its original size.

7. EXPERIMENTAL RESULTS

7.1. Implementation Trials

In addition to what is included in our program, we tried a number of techniques to determine the subject area that failed.

We attempted to use an image normalized into 3 grayscale ranges to assist in reducing the amount of blur for that region. We did not find the results useful as it was fairly unreliable. However, future work in this area may be able to utilize this data.

We also attempted to increase the blur in regions that were relatively flat that contained small edges. This resulted in a

loss of “spiked” features. Our tests for areas used a square region. Using other shapes, such as rectangles may be useful; however, the running time would drastically increase to accommodate these tests.

We also considered making the edges in the middle of the image more important and lessening the importance of the outside edges. However, if the subject wasn’t centered the results were very poor.

7.2. Final Method Results

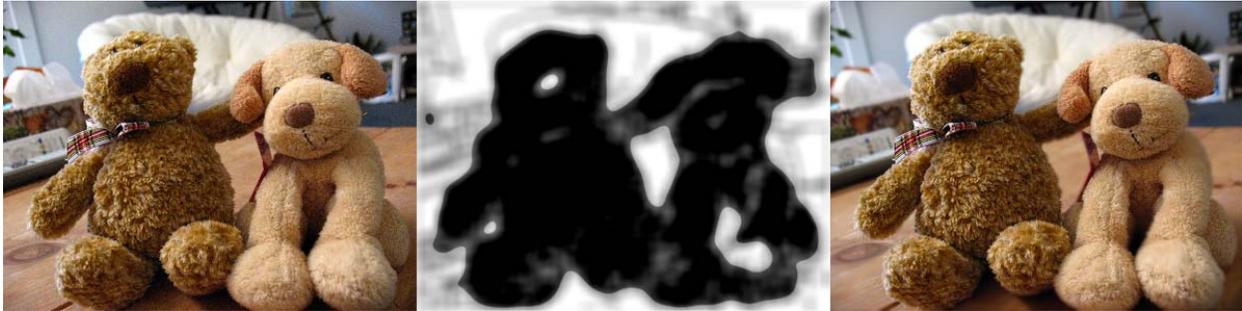
While the thresholds could have been adjusted to optimize each image map and blur amount, the goal was to create a fully automated procedure. As such, all images were processed with the same settings, even though different settings for some images worked much better.

Note: The area in white on the blurred map is the area that gets blur.

7.2.1. Our Results Compared to Bae and Durand's Results

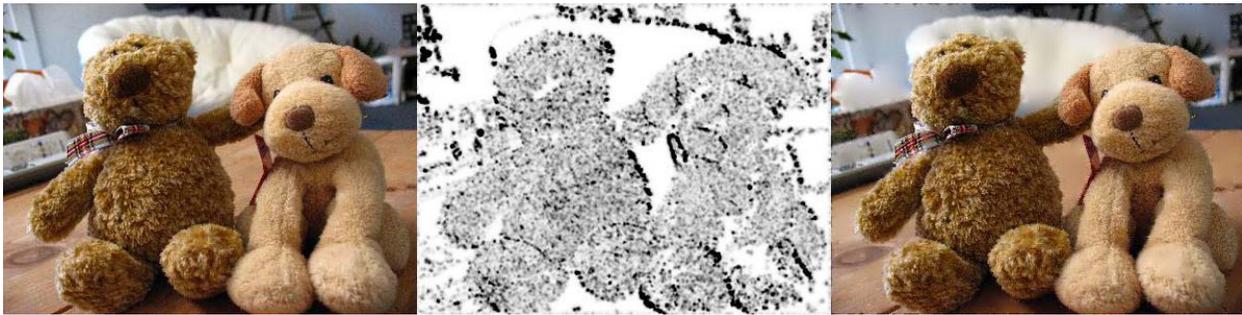
<u>Original Image</u>	<u>Blur Map</u>	<u>Resulting Image</u>
-----------------------	-----------------	------------------------

Our Results



Original image taken from Bae and Durand's website [BD11].

Our Results of Our Implementation of Bae and Durand's Method



Original image from Bae and Durand's website [BD11].

Bae and Durand's Results



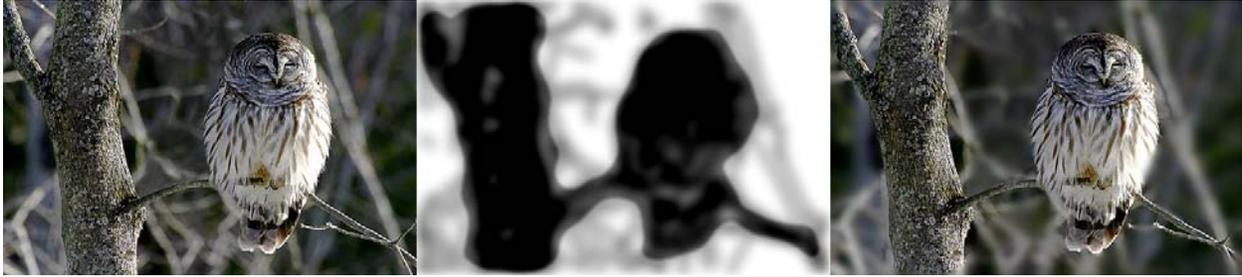
Original image, blur map, and results taken from Bae and Durand's website [BD11].

Original Image

Blur Map

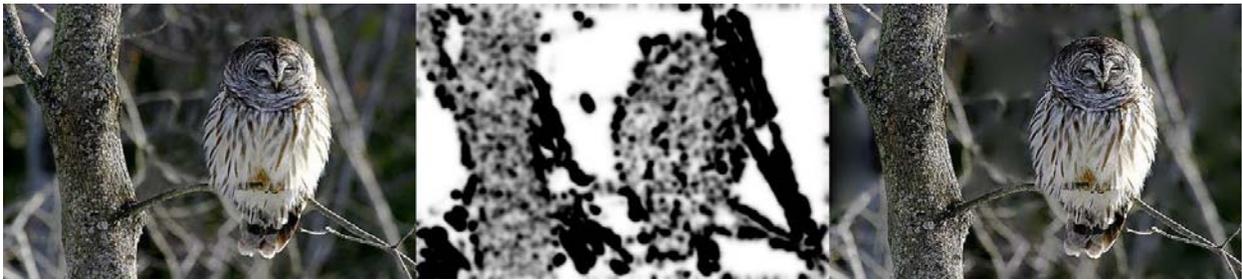
Resulting Image

Our Results



Original image taken from Mike Moraski's website [Mor11].

Our Results of Our Implementation of Bae and Durand's Method



Original image taken from Mike Moraski's website [Mor11].

Our Results



Our Results of Our Implementation of Bae and Durand's Method



Original Image

Blur Map

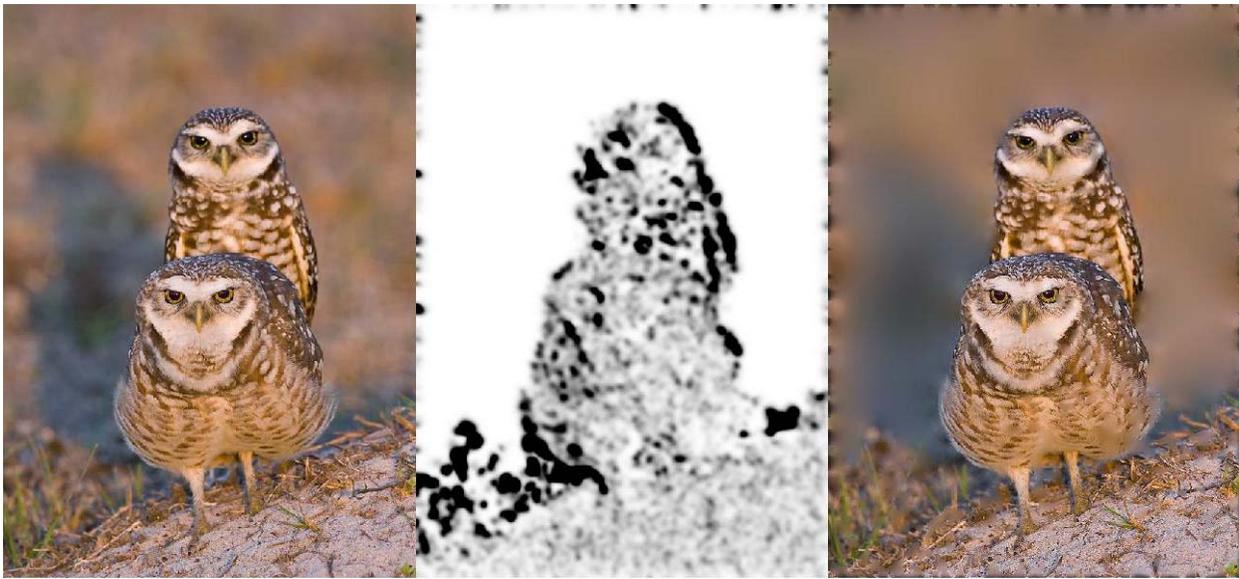
Resulting Image

Our Results



Original image from Birds as Art.com [Mor11]

Our Results of Our Implementation of Bae and Durand's Method



Original image from Birds as Art.com [Mor11]

7.2.2. Failures

<u>Original Image</u>	<u>Blur Map</u>	<u>Resulting Image</u>
-----------------------	-----------------	------------------------

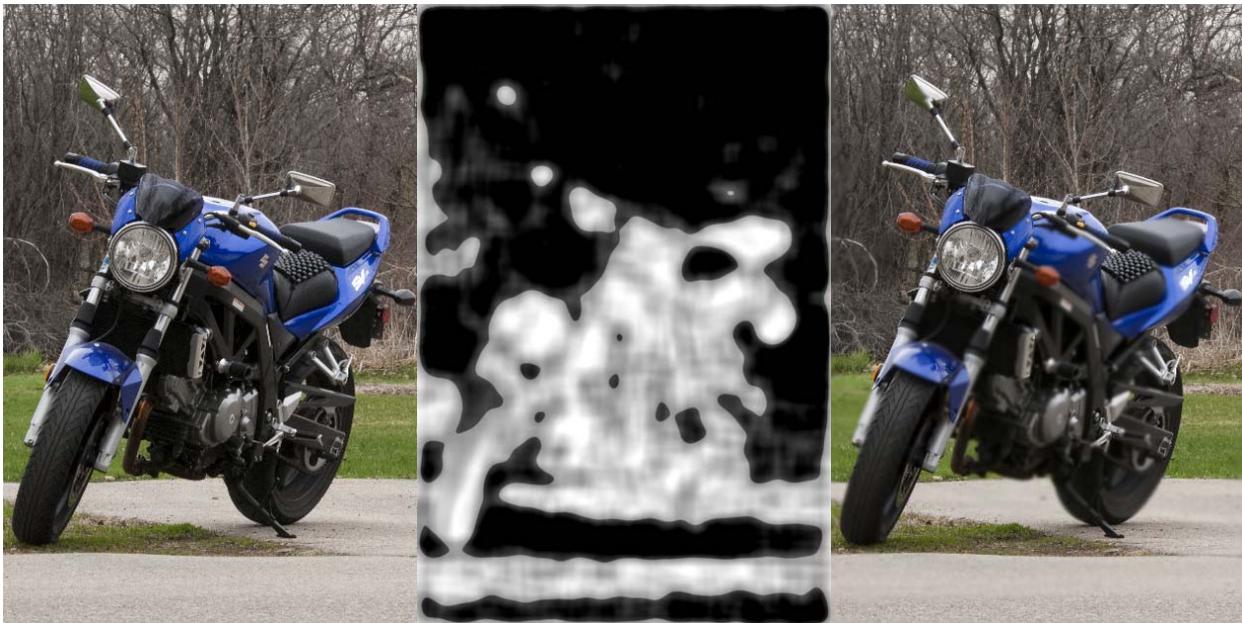
Our Results

The man's face and the parrot were both blurred.



Original image from Making Pictures Blog [Kat11]

This result was inverted, the bike was blurred while the background remained unchanged.



8. CONCLUDING REMARKS

The goal of this project was to detect the subject and apply blur without any user input. To do this, we chose thresholds that were not optimal for all images. However, while testing this program, we discovered that there were general “rules of thumb” that would output better results for different images. We believe that many of these “rules of thumb” can be converted into statistical guides and incorporated into the program, providing enhanced results.

In addition, implementing a blur similar to Adobe[®] Photoshop's[®] 'Lens Blur' would also enhance the output.

9. REFERENCES:

[BD07] BAE S., DURAND F.: Defocus Magnification. *Eurographics*, vol. 26, number 3, (2007)

[BD11] BAE S., DURAND F.: Defocus Magnification. <http://people.csail.mit.edu/soonmin/dof/>. (Accessed May 2011).

[Can86] CANNY J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, No. 6 (1986), 679-698.

[EK03] ELTOUKHY H. A., KAVUSI S.: A computation- ally efficient algorithm for multi-focus image reconstruction. In *SPIE Electronic imaging* (June 2003), vol. 3813.

[EZ98] ELDER J. H., ZUCKER S. W.: Local scale control for edge detection and blur estimation. *IEEE Transactions on PAMI* 20, 7 (1998), 699–716.

[FCM00] FUNT B., CIUREA F., MCCANN

J.: Retinex in MATLAB[®]. *Proceedings of the IS&T/SID Eighth Color Imaging Conference: Color Science, Systems and Applications*, (2000), 112-121.

[FM83] FRANKLE J., MCCANN: Method and apparatus for lightness imaging. *U.S. Patent 4,348,336* (1983).

[FS05] FAVARO P., SOATTO S.: A geometric approach to shape from defocus. *IEEE Transactions on PAMI* 27, 3 (2005), 406–417.

[Har11] HARVEY J.: John Harvey Photo. <http://www.johnharveyphoto.com/>. (Accessed May 2011).

[Kat11] Kathe.: Making Pictures. <http://makingpics.wordpress.com/2011/03/21/depth-of-field/>. (Accessed May 2011)

[Mor11] MORASKI, M.: 8 Steps to Successful Nature Photography. Vivid Light Photography. <http://www.vividlight.com/articles/3403.htm> (Accessed May 2011).

[Des11] DESKTOPIA.NET.: <http://desktopia.net/miscellaneous/stone-bird-desktop-wallpaper#>

[Mor11] MORRIS A.: BirdsAsArt.com <http://www.birdsasart.com/2009/03/17/birds-as-art-bulletin-286/>. (Accessed May 2011).

[Pen87] PENTLAND A. P.: A new sense for depth of field. *IEEE Transactions on PAMI* 9, 4 (1987), 523–531.

[RM92] REEVES S. J., MERSEREAU R. M.: Blur identification by the method of generalized cross-validation. *IEEE Transactions on IP* 1, 3 (1992), 301–311.