The Reverse Projection Correlation Principle for Depth from Defocus

Scott McCloskey, Michael Langer, Kaleem Siddiqi
Centre for Intelligent Machines
McGill University
Montreal, Quebec H3A 2A7 Canada
{scott, langer, siddiqi}@cim.mcgill.ca

Abstract

In this paper, we address the problem of finding depth from defocus in a fundamentally new way. Most previous methods have used an approximate model in which blurring is shift invariant and pixel area is negligible. Our model avoids these assumptions. We consider the area in the scene whose radiance is recorded by a pixel on the sensor, and relate the size and shape of that area to the scene's position with respect to the plane of focus. This is the notion of reverse projection, which allows us to illustrate that, when out of focus, neighboring pixels will record light from overlapping regions in the scene. This overlap results in a measurable change in the correlation between the pixels' intensity values. We demonstrate that this relationship can be characterized in such a way as to recover depth from defocused images. Experimental results show the ability of this relationship to accurately predict depth from correlation measurements.

1. Introduction

The recovery of 3D scene coordinates from 2D images has long been an area of significant research within the computer vision community. Many of the methods that have been proposed are based on cues that humans use for depth perception. One such cue is blurring or defocus. Studies, such as [7], have shown that blurring is used by humans to find relative depth.

Blurring or defocus arises in natural images as a result of an optical system's limited depth of field. The locus of scene points that will be well-focused in an image is referred to as the *plane of focus*, and is parallel to the sensor plane. Other scene points will be blurred in proportion to their distance from this plane. Thus, one can estimate the depth to a point in the scene by measuring its defocus and reversing this relationship. This is the principle underlying methods of depth from defocus.

This paper introduces a new way to measure defocus from natural images. This method is based on a new *reverse projection correlation principle*, which states that the correlation between adjacent pixels increases as the parts of the scene that they sense overlap. Moreover, this overlap increases with increasing distance between the scene object and plane of focus. Based on this principle, we measure correlations between pixels to estimate defocus, and thus depth. We motivate this principle by presenting and validating a camera model that works on the principle of reverse projection. This model also accounts for the fact that pixels are not points on a sensor, but areas of substantial size, which is an important component of blurring that is ignored in most previous literature in depth from defocus. Experimental results from this model are shown that demonstrate the ability of measured correlations to give depth.

2. Basic Depth from Defocus

The use of focus and defocus to reconstruct the 3D structure of a scene is well established in the field of computer vision. One class of methods, referred to as depth from focus, includes techniques that search for lens settings which produce the sharpest image of each region of a scene. Using the thin lens equation

$$\frac{1}{f} = \frac{1}{d_s} + \frac{1}{d_o} \,, \tag{1}$$

it is possible to derive the depth to each region (d_o) from the focal length of the lens (f) and lens-sensor distance (d_s) that brought it into focus.

Depth from *de*focus is the name used to denote methods, including the one introduced in this paper, that use the amount of blur at a point to infer its depth. The fundamental relation underpinning these methods is illustrated in Fig. 1 and the following (adapted from [10])

$$d_o = \frac{f d_s}{d_s - f - F\sigma} \,, \tag{2}$$

where F is the lens aperture number and σ is the radius of the blur circle created by a point at distance d_o from the lens.

Essentially, if one can measure the blur radius at a point in the image then we can use Eq. 2 to determine the distance

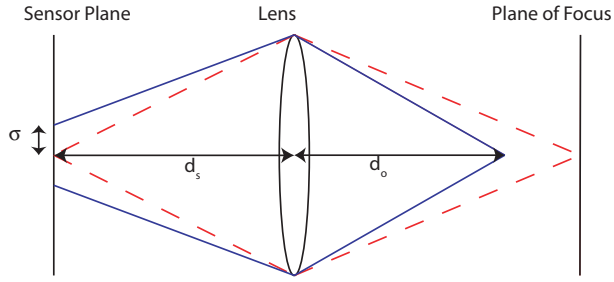

Figure 1: The basic relation for depth from defocus.

from the corresponding scene point to the lens. There is an inherent confusion due to the fact that points at the same distance from, but on opposite sides of, the plane of focus have the same blur radius. This is typically resolved by focusing on the nearest (or furthest) point in the scene.

One of the fundamental problems with depth from defocus (DFD) is that we can't tell whether or not something is blurred from a single image. An intensity gradient in an image, for example, could arise from a blurred step edge or a gradual transition within the plane of focus. Because of this, we take two pictures of the same scene with different aperture and measure the *change* in the blur, which gives depth by a relation derived from Eq. 2.

DFD methods are also restricted to scenes that contain some texture, as it is the change in the appearance of the texture that we use to measure the change in blur radius. An untextured surface such as a white wall would look the same regardless of its position relative to the plane of focus, so it is not possible to derive its depth from defocus.

3. Previous Work

The method introduced in this paper is general purpose, in that it can measure depth at all textured regions in a scene. Broadly speaking, previous general DFD methods can be put into one of two categories: those based on a shift-invariant linear systems model, and those based on energy minimization methods. The theory and modeling of these different approaches is summarized below, along with a look at some of the more notable examples.

3.1. Shift-Invariant Linear Systems Methods

The most popular way to look at DFD has been to pose it in terms of a shift-invariant linear system. While not given as an input, the perfectly-focused image of a region within the scene is represented by the function i_0. Two images (i_1 and i_2) are taken of that region with different apertures, and are modeled as the result of convolving i_0 with blur filters h_1 and h_2. Thus $i_1 = i_0 * h_1$, and $i_2 = i_0 * h_2$.

In [10], Pentland uses the convolution theorem to relate the ratio of i_1 and i_2's Fourier representations to the change in σ which, in turn, gives depth. Nayar and Watanabe [8] change the focal position instead of aperture number, and measure the more stable *normalized ratio* of the two images' Fourier representations. Ens and Lawrence [4] search a set of filters for the one that best explains the difference between i_1 and i_2. That filter, h_3, minimizes the quantity $\|i_1 - i_2 * h_3\|$, and is uniquely related to depth. In [11] and [9], the authors use active illumination with structured light and measure the change in Fourier power at the frequency of the structured light.

While the shift-invariant linear systems interpretation is straightforward and results in elegant solutions, it is problematic for a number of reasons. The biggest problem is that it requires that scenes adhere to the *equifocal assumption*, meaning that the scene is comprised of planes parallel to the sensor.

Methods that relate depth to changes in Fourier power require large image regions to measure power accurately, which makes the equifocal assumption that much more tenuous. A more detailed analysis of related shortcomings is presented in [4].

3.2. Iterative Methods

In order to avoid the equifocal assumption, several iterative methods have been developed to recover *both* a surface (depth) and its radiance. The methods presented in [5, 6] define an energy functional which is jointly minimized with respect to both shape and radiance. In [5] the problem is posed as the minimization of an information divergence between blurred images and a solution is developed for the case of equifocal planes. In [6] a regularized solution is sought and developed using Hilbert space techniques and SVD. In [3], depth and radiance are both modeled as Markov Random Fields (MRF) and a maximum a postiori estimate is found. These approaches have the advantage that they do not require an equifocal imaging model, though regularization and MRF models implicitly assume that depth changes slowly.

4. Camera Model

Most of the methods reviewed above are illustrated with forward projection. That is, they attempt to map points in the real world to the area of their blurred image on the sensor, as in Figure 1. Our camera model is based on reverse projection, where we look for the area of the scene that is recorded by a certain point on the sensor.

The model begins by finding the point in the 3D scene that would be recorded by a sensor point through a pinhole aperture. Then we account for that point's defocus due to its displacement from the plane of focus and aperture setting, which gives the area of the scene whose reflected or emitted light reaches the sensor point. Finally, noting that pixels

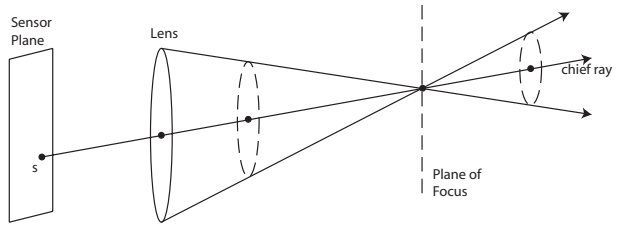

Figure 2: The double cone from which a point on the sensor records light.

have significant area, we aggregate the areas sensed by individual sensor points to determine the part of the scene from which the pixel records light. Given this, we can determine the pixel's value by taking the dot product of this aggregate with the scene's radiance. Each of these steps is described in greater detail in the following sections.

4.1. Reverse Projection

If we restrict ourselves to a camera with a pinhole aperture, we follow the *chief ray* that connects a point s on the sensor with the optical center of the lens. Ignoring transparency, light scattering and mirror reflections, the point at which the chief ray intersects the scene is what will be sensed at s. In the more general case of a non-pinhole aperture, the area recorded by a particular point on the sensor will be the intersection of the scene with a double cone[1]. This is also described, in the context of occlusion edges, by the reverse projection blur model of Asada et al [1]. The cone has its base at the aperture of the lens, and its vertex at the point where the chief ray intersects the plane of focus. Beyond the plane of focus, the cone expands without bounds. The point s will sense all light that is emitted or reflected from points within the cone as long as that ray remains within its bounds and it reaches the lens. An illustration of such a cone is shown in Fig. 2.

We assume that the small bundle of rays that leave a visible surface point and arrive at the lens has constant radiance, i.e. specularities are not allowed. We can then treat the problem geometrically, in that we consider those scene points that are visible at a particular point on the sensor plane. A visibility map can then be represented as a binary-valued function defined on scene coordinates. A scene point carries a value of 1 in the visibility map if it is within the double cone and is unoccluded, and 0 otherwise.

We define t to be the function that maps points on the sensor plane to a visibility map in the manner described above. A value of 1 in $t(s)$ indicates that a scene point is sensed at point s and a value of 0 indicates that it is not.

[1]A double cone consists of two cones that share the same vertex and whose circular cross-sections grow at the same rate.

4.2. Incorporating Pixel Size

One important camera parameter that is often overlooked in DFD literature is the physical size of a pixel on the image sensor. Previous methods have ignored this despite the fact that it can have a significant impact on blurring. The method presented in [9] accounts for the pixel size, but in a fundamentally different way, since the system uses active illumination projected through the camera's optical system.

Since a sensor pixel combines all the light that it senses from an area of the scene, the effect of a large pixel is a baseline blurring effect. This is particularly troublesome at the horizon or around occluding contours of smooth objects, where the scene is almost perpendicular to the image plane. In such places the scene area sensed by a pixel is quite large, and thus the baseline blurring is severe.

Given the area A_p on the sensor occupied by a pixel p, we can define that pixel's *transfer function* as

$$T(p) = \int_{s \in A_p} t(s) ds \ . \tag{3}$$

Conceptually, the transfer function is the sum of the visibility maps for each point within the pixel's area; it expresses the area of sensor points within a pixel that can see each point in the scene. While our implementation uses the summation, the continuous formulation presented here requires the integral. Note that the transfer function is no longer binary, as scene points can be visible to any fraction of the pixel's area. Finally, the transfer function must be normalized so that the sum of its values is equal to one. The transfer function is useful because it allows us to model the observed intensity I of a pixel as the inner product of T with the scene radiance R. That is,

$$I(p) = R \cdot T \ . \tag{4}$$

While they tend to be fairly small (e.g., 7.8μm square for the Nikon D100, our test camera), the area sensed by a pixel increases with the distance of a scene object from the lens. To get a basic understanding for the impact of a finite pixel area on our ability to measure DFD, consider a fronto-parallel plane with a checkerboard radiance pattern. If the checkerboard is near the lens, the transfer functions of most pixels will be small and integrate light reflected entirely from either the white or black checks. As the lens and checkerboard get further apart, the size of the transfer functions will increase. When the area of each pixel's transfer function equals 4 times the area of a check, the image will be a uniform gray *even if the checkerboard lies within the plane of focus*. At this point, the image will lack the texture necessary to measure depth from defocus and all known methods will fail.

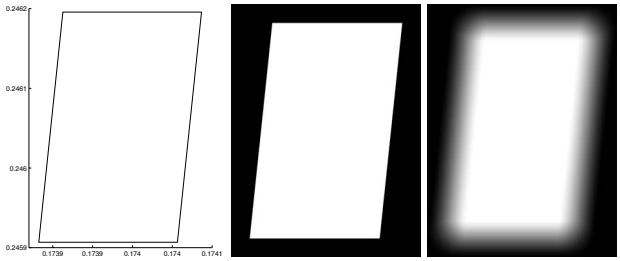

Figure 3: (Left) The surface area sensed by a pixel. Units are meters. (Middle) Its discretization. (Right) Its transfer function, including blur.

4.3. Implementing the Model to Render Scenes

Having defined the model as above, we describe an implementation made to perform validation (sec. 4.4) and experiments (sec. 7) with defocused images. Since this requires knowledge of the location of all scene points, we temporarily consider planes slanting away from the observer in the Y direction. Later, in section 8, we discard slanted planes and demonstrate the ability to recover depth in a more general scene using correlation.

By assuming that the scene is a slanted plane we reduce the scene coordinates to $\mathcal{R}^2$, expressed with respect to axes on this plane. For convenience we take the origin on the plane to be its intersection with the optical axis. Furthermore, we impose a discretization of the plane to further reduce the scene to $\mathcal{Z}^2$. This allows us to use an image to represent the scene's radiance. We also maintain a scale parameter that indicates the physical size of each pixel.

Having described how we characterize the scene's reflectance, we turn now to the characterization of a pixel's transfer function. With respect to pixel areas, we model the sensor as a tightly packed array of square pixels of the appropriate size. As a result, the area A_p of a pixel is a square whose sides have length equal to the pixel pitch specified by the manufacturer. This is a realistic model for charged-couple device (CCD) sensors like the one in our test camera, or any sensor whose *fill factor* - the percentage of the area that records light - is at or near 100%.

We derive a pixel's transfer function by first projecting each of the real valued coordinates representing the pixel's corners to the real valued coordinates of their projections on the slanted plane. This is done by following the chief ray, as described above, and gives a trapezoid that represents the area that would be sensed by that pixel through a pinhole aperture. Fig. 3 (left) shows, for example, the area of a plane slanted at 45° sensed by a pixel away from the center of the sensor plane. It is elongated in the vertical direction as a result of the plane's slant away from the observer.

Next we discretize the projection on the lattice of scene coordinates, which gives the pixel's unblurred transfer function. Fig. 3 (middle) shows the discretized version of the projection shown in Fig. 3 (left).

In order to model the blurring, we use the depth to find the appropriate point spread function (PSF) *at each point* in the discretized projection. Note that, since each point is convolved with a *different* PSF, this respects the fact that the plane is not an equifocal surface. As a result, this process is linear and shift *variant*. Also, the PSF here is in the sense of reverse projection; it represents the spread in the scene of a point on the sensor. We add up the convolution of each point with its PSF to give the pixel's transfer function. Fig. 3 (right) shows the result for the previous example.

Finally, the response of the pixel is found by taking the dot product of the transfer function with the scene radiance image. Repeating this process for every pixel allows us to render what the scene would have looked like were it photographed with the given set of camera parameters.

4.4. Validation of the Camera Model

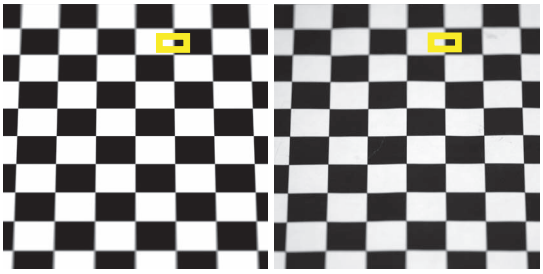

Figure 4: Rendered (left) and camera (right) 4 megapixel images of a checkerboard under near identical conditions.

In order to show the validity of this model and the integrity of the implementation, we compare a rendered image to a camera image taken with the same parameters. The scene was a checkerboard pattern on a plane sloped away from the camera at an angle of 45°. For the camera image, this angle was measured by markings on the tripod's axis. The lens had a focal length of 180mm and an aperture of f/2.8. The focal plane was at a distance of 1.8 meters, which intersected the scene in the center of the image. The accuracy of both the viewing angle and distance to the focal plane is probably quite low for the camera image, given the physical means used to measure them. The rendered and camera images are shown in Fig. 4. Note that the rendered image is tone mapped with a function (determined using the method of [2]) that approximates the camera's behavior.

Since the purpose of this model is to study defocus, we are particularly interested in the differences between the two images at blurred edges. Fig. 5 shows profiles of the blurred edge highlighted in Fig. 4. Pixel values are scaled so that the mean value of the white and black checks are 1 and 0, respectively. The root mean squared error for all of the edges

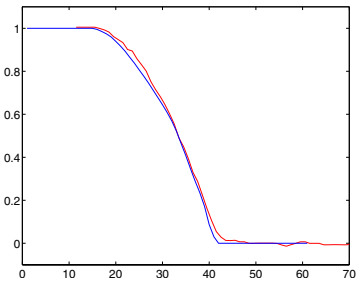

Figure 5: (Right) Profile of the edge highlighted in fig. 4: (red) camera data, (blue) rendered data.

was 3.8%. Given the presence of noise and lack of precision in some of the physical measurements this represents a validation of the camera model.

5. DFD with Reverse Projection

It is important to note that the definition of σ from Fig. 1 does not make sense with respect to reverse projection. The analogous quantity, which we will call $\hat{\sigma}$, is the radius of the cone at the point where the chief ray intersects the surface. The key difference between these two values is that σ is a blur radius within the camera whereas $\hat{\sigma}$ is a blur radius in the scene.

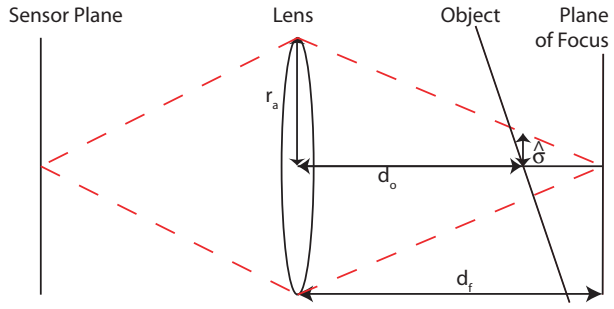

Figure 6: The value $\hat{\sigma}$ is the reverse projection analog of σ.

Fig. 6 shows how $\hat{\sigma}$ is related to the lens-object distance d_o, the lens-focal plane distance d_f and the aperture radius r_a. Writing this as an equation and replacing the aperture radius with the more familiar camera parameters,

$$\hat{\sigma} = \frac{f}{2F} \frac{\|d_o - d_f\|}{d_f} \ . \tag{5}$$

where, as before, f is focal length and F is the aperture f-number.

Now we express the relation between relative depth $d = \|d_o - d_f\|$ and the change in $\hat{\sigma}$ by taking two measurements and subtracting to give

$$\hat{\sigma_1} - \hat{\sigma_2} = \frac{df}{2d_f} \left(\frac{1}{F_1} - \frac{1}{F_2} \right) \ . \tag{6}$$

6. Correlation as a Measure of Defocus

Having established the model and a way to measure depth from the change in $\hat{\sigma}$, we look for a way to measure this defocus. To that end, consider the transfer functions of two adjacent pixels derived using our model. When the pixels are in focus, as in Fig. 7 (left), adjacent pixels sense light reflected from trapezoidal areas adjacent in the plane. As the plane moves farther out of focus the transfer functions of the pixels get blurrier, as in Fig. 7 (right). One important observation is that *the transfer functions begin to overlap*, which means that they both record light reflected from the same part of the scene. In terms of the model, the transfer functions of the two pixels will both have non-zero entries at a number of the same scene points. This will result in an increased correlation between the observed intensity values at adjacent pixels.

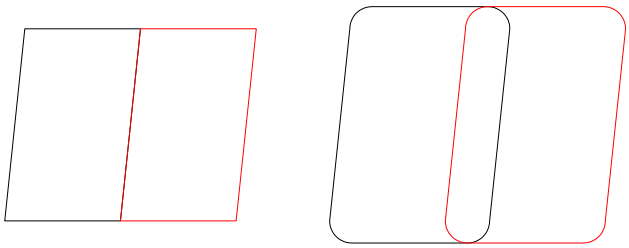

Figure 7: The area sensed by two adjacent pixels (black and red). (Left) Through pinhole aperture. (Right) Same pixels, through finite aperture.

Moreover, the width of the overlap between adjacent pixels is $2\hat{\sigma}$. This implies that *correlation between adjacent pixels increases as the object moves further from the plane of focus*. This is the **Reverse Projection Correlation Principle**, and is the basis of our new DFD method.

By similar reasoning, the correlation of pixels increases with $\hat{\sigma}$ even if those pixels are separated by some distance. This is the case for CCD pixels that are non-adjacent, as well as adjacent pixels on sensors where the fill factor is significantly less then 100%. The main difference will be that the correlation of well-focused pixels will decrease as the distance between the pixels increases.

6.1. Measuring Pixel Correlations

While the notion of correlation is generally well understood, there are many ways to measure it depending on the form of the expected relationship. Since we are interested in measuring the correlation between them, we wish to characterize the relationship between the intensities of a pixel (I) and one of its neighbors (I').

Intensities of nearby pixels in natural scenes tend to differ by only a small amount. In the sense of a scatter plot of (I, I') pairs, then, we would expect points to cluster about

the line $I = I'$. Moreover, as the image becomes blurred and the transfer functions of neighboring pixels begin to exhibit significant overlap, we would expect the differences between their intensities to decrease. In the scatter plot, we would expect points to cluster even more about the line $I = I'$.

In a more concrete sense, then, we wish to measure the degree to which the line $I = I'$ explains the relationship between the intensities of nearby pixels. Of existing correlation measures, we use the *sample correlation coefficient*. It is a standard measure that expresses the fraction of the variance of one variable that is explained by a linear fit between the two. Given N observations of I and I', the correlation coefficient (CC) is

$$CC = \frac{\sum_{j=1}^{N}(I_j - \overline{I})(I'_j - \overline{I'})}{(N-1)s_I s_{I'}} \,. \tag{7}$$

where s_I and $s_{I'}$ are the standard deviations of the right and left pixel intensities. The value of CC is strictly within the range [-1,1].

Readers may note that this is quantity is somewhat similar to the definition of the autocorrelation function with a shift of one. Given that the Wiener-Khintchine theorem relates an image's autocorrelation to its power spectrum, it is tempting to suggest that the value of the CC can be found within the power spectrum. The important difference is that the CC normalizes for contrast, whereas the autocorrelation does not. As an image becomes more blurred, the correlation between pixels increases (due to overlapping transfer functions) while the contrast decreases. The autocorrelation confounds these two effects, giving a decreased value for increasing blur. On the other hand, because it accounts for contrast explicitly, the CC increases with increasing blur, as our intuition would suggest.

6.2. Relating Correlation to Depth

As previously mentioned, it is impossible to know if a blurred edge in an image is the result of a blurred step edge or a well-focused gradient. Because we don't know the properties of the scene in advance, it would be ill-advised to try and derive depth from pixel correlations in a single image. Instead, we measure the change in CC between two images taken with different apertures. As noted in section 6, the change in correlation (measured by CC) is related to the change in the scene blur radius $\hat{\sigma}$. If that relation is accurately characterized we can find the change in $\hat{\sigma}$ and use Eq. 6 to find depth.

It remains to be shown how to get a large enough sample size of adjacent pixel pairs at the same depth from which to compute the CC. In the case of the images rendered by our model, where the scene is a plane slanted away from the camera in the Y direction, this can be done by looking at pixels in the same row. We treat the values of the first/second pixels in the row as one observation of I/I' pixel values. The values of the second and third pixels constitute a second observation, etc. Since pixels in the same row are at the same depth, each of these pairs will have the same amount of overlap. We get a value of the CC for each row of an image with a small aperture, as well as for each row of an image with a large one. The difference between the CC observed on the same row in the two images is the value that we will relate to the change in $\hat{\sigma}$ and, in turn, depth.

Fig. 8 (right) shows an example of the change in $\hat{\sigma}$ as a function of the observed change in CC between adjacent pixels. These are derived from the small and large (f/4.5 - Fig. 8 left) aperture renderings made by the camera model. Given a characterization of the relationship shown in Fig. 8 (right), we can relate change in CC to the change in scene blur radius and, finally, to depth.

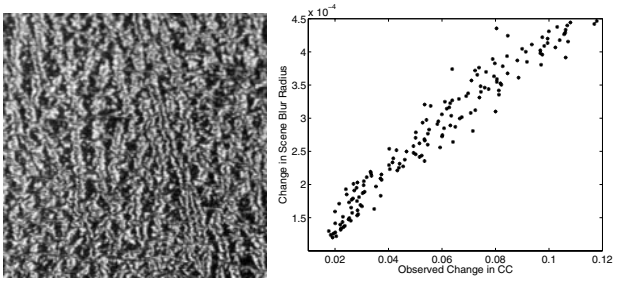

Figure 8: (Left) Large aperture rendering of a pattern on a slanted plane. (Right) Change in scene blur radius as a function of change in CC.

7. Experiments on Rendered Images

Plots like those shown in Fig. 8 (right) and others like it suggest that there is a stable relationship between the change in CC and the change in $\hat{\sigma}$. In this section we demonstrate that this relationship can be characterized in the form of a lookup table, which can be used to give estimates for the change in $\hat{\sigma}$ from new images. Such an estimate can then be converted to relative depth by Eq. 6. This demonstrates the utility of the change in CC as a measure of depth, and constitutes a new method of deriving depth from defocus that is not subject to the equifocal assumption.

7.1. Characterization of $\Delta\hat{\sigma}$ vs ΔCC

In order to characterize the relationship between the change in CC and the change in $\hat{\sigma}$, we use our implementation of the camera model to render images of a textured plane under a number of different conditions. For the purpose of the experiments in this section, our training texture is the wood image from the Brodatz set. We use this as the scene radiance and render its appearance at a number of different

depths and viewing angles. For each combination of depth and angle, we render a small (f/29) and large (f/4.5) aperture image. We measure the CC of pixels in each line of both images in the manner described in section 6.2, and find the change in correlation between the two images. Since the model also computes the value of $\hat{\sigma}$ at all points, we can compute its change between apertures at each row. So, for each row in the images rendered with each combination of depth and angle we get an observation of the change in the CC and the change in $\hat{\sigma}$.

We render 25 pairs of images, each with a different combination of depth and viewing angle. This gives 4516 observations, from which we build our lookup table. We quantize the change in CC into bins of width 0.005. Within each bin we use the median observed change in $\hat{\sigma}$, and we store the standard deviation of the changes in $\hat{\sigma}$. The standard deviation will be used as an estimate of the quality of the fit for weighting purposes.

This process is repeated four times. In the first iteration, CCs are measured between adjacent pixels. In the second, CCs are measured between pixels that are separated by a distance of one pixel, etc. In the end, the lookup table consists of four sets of bins for the change in CC. Each bin has an associated value of the change in $\hat{\sigma}$, as well as the standard deviation of observations within the bin.

In constructing this lookup table, we note that the quality of the fit, as reflected in the standard deviation of observations in the same bin, reduces significantly as the depth - either absolute or relative - increases. This is because, as the camera model predicts, the scene area sensed by a pixel increases with respect to the size of the scene's texture elements. After a point, the CC of well-focused pixels is close enough to 1 that additional correlation can not be observed.

7.2. Recovering Depth from Test Images

We test the quality of the characterization by taking new textures and rendering them with different depths and viewing angles. For each combination of texture, depth, and viewing angle we render images with the same apertures as used in the characterization phase. Then we measure the CC of adjacent pixels for each line, as well as the CC of those separated by one, two, and three pixels.

For each line, then, we have four measures of the change in CC corresponding to different distances between pixels. For each measured change, we find the appropriate bin in the lookup table and note the estimate of the change in $\hat{\sigma}$, s_i, as well as the standard deviation τ_i corresponding to that bin. We combine these different estimates to get our final estimate of $\hat{\sigma_1} - \hat{\sigma_2}$ by weighting each estimate by the inverse of its associated standard deviation.

Fig. 9 shows the estimated and actual values of $\hat{\sigma_1} - \hat{\sigma_2}$ at each row of the images for different combinations of texture, depth, and viewing angle. The textures are taken from Brodatz and are, from left to right, leather (shown in Fig. 8) and weave. The change in $\hat{\sigma}$ is converted to depth using Eq. 6, and is compared to ground truth. The root mean squared errors (RMSE) in depth for the three examples are 0.014 and 0.009 meters. These correspond to RMSE of 0.8% and 0.4% in absolute depth. In terms of relative depth, as defined in section 5, the RMSE are 8.0% and 7.9%. As we can see, our characterization of the relationship between the change in $\hat{\sigma}$ and the change in CC was accurate, though noisy.

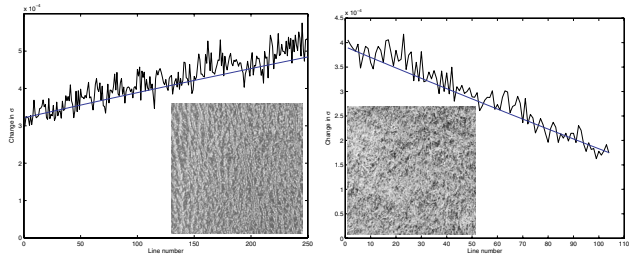

Figure 9: $\Delta\hat{\sigma}$: (blue) ground truth, and (black) using change in CC. Inset pictures show the test textures which are: leather (left) and grass (right).

8. Recovering Depth from Real Images

While the recovery of depth by observing changes in correlation is best done along equifocal contours in the image, it is also possible to find the relative depth of scenes without a priori knowledge of equifocal contours. Within an $n \times n$ window, one can observe $2n^2 - 2n$ pairs of adjacent pixel intensities; $n^2 - n$ are horizontally adjacent and $n^2 - n$ are vertically adjacent. For a separation of k pixels, the same window provides $2n^2 - 2(k+1)n$ observations. We can use equation 7 to measure the correlation coefficient from this set of observations. Given two images taken with different apertures, we can measure the change in the CC of pixels within a window centered at each pixel. This is analogous to classical DFD methods based on square windows.

In Fig. 10, we show the results of this method on a real scene from the windowed measurement scheme just described. The scene consists of a cylindrical can textured with a white noise pattern lying on a (roughly frontoparallel) carpeted floor. Two images were taken with a focal length of 50mm; the apertures were f/9 and f/16. The large aperture input image is shown in Fig. 10 (top left). Fig. 10 (top right) shows a grayscale map indicating the change in the CC of adjacent pixels. As we have shown, this quantity is linearly related to depth. The input image was 1.5 megapixels, and the can was approximately 50cm from the lens. The window size for this example was 51×51 pixels. Fig. 10 (bottom left) shows the normalized ΔCC along a row of the image, along with the mean over all rows.

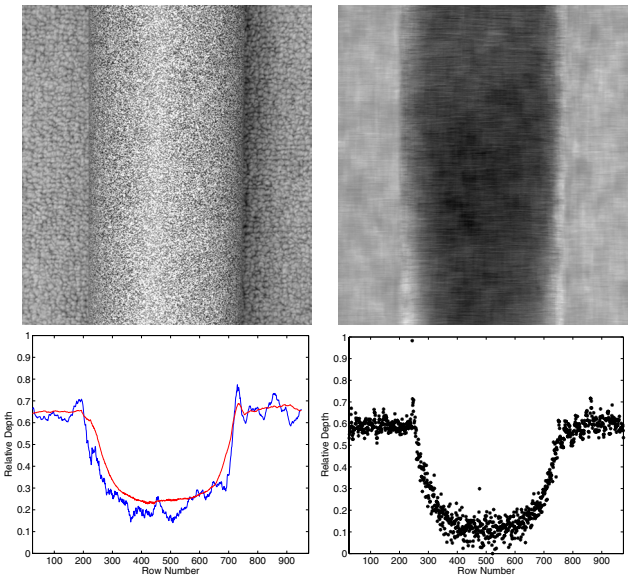

Figure 10: (Top Left) Input image of test scene. (Top Right) Grayscale map of ΔCC. (Bottom Left) Plot of cross-section of relative depth (blue) and mean cross-section (red). (Bottom Right) Depth from equifocal contours.

Given this initial estimate of the relative depth, one can compute the direction of the depth gradient and, in turn, find equifocal contours in the image. This will allow for computing the CC along such contours, which gives numerical depth that preserves discontinuities. An automatic method for the computation of the initial depth estimate and refinement to numerical depth is the subject of ongoing research. For the purpose of illustration only, Fig. 10 (bottom right) shows the change in CC observed along equifocal contours (columns of the image, in this case). While the measure is noisy, it better preserves the depth discontinuity at the occluding boundary between the object and background.

9. Concluding Remarks

As these experiments have shown, the change in correlation between nearby pixels can be used to measure of depth from defocus. This relationship is a consequence of a new reverse projection correlation principle, which we have motivated with a new camera model. This represents an advancement over existing DFD methods in the linear systems framework, which assume that the scene is comprised of equifocal planes. One of the important aspects of this model - accounting for the area of a pixel on the sensor - has generally been ignored in DFD literature despite its impact on the ability to measure defocus. Our implementation of the model was validated by comparing its output to a real camera image, and was used to characterize the relationship between the change in correlation coefficient and defocus.

The main point of this paper has been the introduction of the reverse projection correlation principle, the reverse projection camera model, and the relationship between the change in correlation coefficient and depth. We have shown that this relationship can provide depth information for scenes of unknown geometry. Future research will use an initial estimate of depth - obtained by measurements from square windows - to determine equifocal contours along which the depth can be measured more accurately. As we have shown, this method has the advantage of preserving depth discontinuities.

References

[1] N. Asada, H. Fujiwara, and T. Matsuyama. *Seeing Behind the Scene: Analysis of Photometric Properties of Occluding Edges by the Reversed Projection Blurring Model*. IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 20, pp. 155-167, 1998.

[2] P. Debevec and J. Malik. *Recovering High Dynamic Range Radiance Maps from Photographs*. Proc. SIGGRAPH 1997, pp. 369-378

[3] S. Chaudhuri and A. Rajagopalan. *Depth from Defocus: A Real Aperture Imaging Approach*. Springer Verlag, 1999.

[4] J. Ens and P. Lawrence. *Investigation of Methods for Determining Depth from Focus*. IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 15, No. 2, pp. 97-108, 1993.

[5] P. Favaro, A. Mennucci and S. Soatto. *Observing Shape from Defocused Images*. Intl. J. of Comp. Vision, 52(1), pp. 25-43, 2003.

[6] P. Favaro, and S. Soatto. *A Geometric Approach to Shape from Defocus*. IEEE Trans. on Patt. Anal. and Mach. Intell., 27(3), pp. 406-417, 2005.

[7] J. Marshall, C. Burbeck, D. Ariely, J. Rolland, and K. Martin. *Occlusion Edge Blur: A Cue to Relative Visual Depth*. J. of the Optical Soc. Am., 13(4), pp. 681-688, 1996.

[8] S.K. Nayar and M. Watanabe. *Minimal Operator Set for Passive Depth from Defocus*. Proc. CVPR 1996, pp. 431-438, June 1996.

[9] S.K. Nayar, M. Watanabe, M. and Noguchi. *Real-Time Focus Range Sensor*. IEEE Trans. on Patt. Anal. and Mach. Intell., vol. 18, no. 12, pp. 1186-1198, Dec. 1996.

[10] A. Pentland *A New Sense for Depth of Field*. IEEE Trans. on Patt. Anal. and Mach. Intell., vol. 9, no. 4, pp. 523-531, July 1987.

[11] A. Pentland, S. Scherock, T. Darrell, and B. Girod *Simple Range Cameras Based on Focal Error*. J. of the Optical Soc. Am., vol. 11, no. 11, pp. 2925-2935, Nov. 1994.

[12] M. Subbarao and G. Surya. *Depth from Defocus: A Spatial Domain Approach*. Intl. J. of Comp. Vision, vol. 13, pp. 271-294, 1994.