

Deblurring Images

Everett R. Ramer, U.S. Department of Energy

If you are an optical microscopist, chances are you have sometimes wished for a way to increase the depth of focus of your images. In this article I describe a method that does this using a simple combination of functions built into most image processing software - so it will not cost you very much to try. The method, however, is not universal and some trial-and-error will be required to get it to work in different applications. Finally, this method is not limited to microscopy and will work with any imaging system.

The basic steps for creating an image with a large depth of focus are straight forward:

1. Take a series of images at different focal planes
2. Remove the blurred content from each image
3. Combine the deblurred images into a single image

These steps are illustrated in Fig. 1, which suggests two different ways of combining the deblurred images - top view, in which the images are overlaid on a single, 2D image; and solid model, in which the deblurred images are extruded vertically to fill the gaps between focal planes.

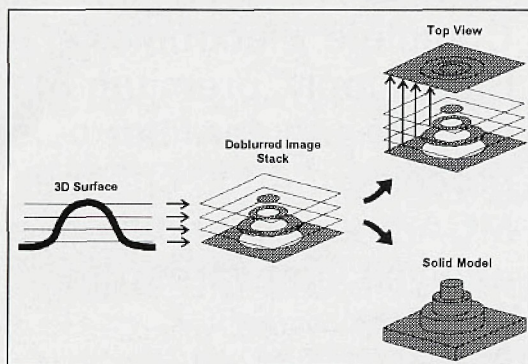


Fig. 1. Imaging of a 3D surface.

Since the first step is self-explanatory, we will skip it and go directly to the procedure for removing the blurred content of an image, which is described in the algorithm *DEBLUR_IMAGE*.

The basic logic for *DEBLUR_IMAGE* can be expressed as follows:

1. Identify blurred and focused regions in the image
2. Create a binary mask from the image

$$\text{mask pixel} = \begin{cases} 0, & \text{image pixel belongs to the blurred region} \\ 1, & \text{image pixel belongs to the focused region} \end{cases}$$

- 3 Logically "And" the mask with the image

Algorithm *DEBLUR_IMAGE*

```
//Copy original image and convert copy to gray scale
MASK ← ORIGINAL
If MASK is colored, convert MASK to gray scale
//Detect and place texture in foreground of binary image
MASK ← ( Laplace1 ( MASK )
FOREGROUND ← F1, 255 //F1 is a user-selected value
MASK ← GrayToBinary ( MASK, FOREGROUND )
//Consolidate scattered foreground pixels
MASK ← GrayDilate ( MASK ) //Enlarge foreground pixels
MASK ← Average ( MASK ) //Round blocky borders
... //Repeat Dilate/Average until foreground pixels merge
MASK ← GrayErode ( MASK ) //Shrink foreground blobs
MASK ← Average ( MASK ) //Round blocky borders
... //One less erode than dilate keeps fringes
FOREGROUND ← 127, 255 //Lower value not critical
MASK ← GrayToBinary ( MASK, FOREGROUND )
//Create DEBLURRED image
DEBLURRED ← And ( ORIGINAL, MASK )
```

Distinguishing between blurred and focused regions is done by noting that objects in focused regions have sharp edges and textured surfaces, compared to the softer, smoother look of objects in blurred regions. Mathematically, the fluctuations in pixel luminance are greater in focused regions than in blurred ones, so a high-pass filter should increase the contrast between these regions. The high-pass filter I use is the Laplacian, which can be approximated by either of the following kernels, with Laplace2 being more sensitive to texture than Laplace1:

$$\text{Laplace 1} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ or Laplace 2} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Thresholding the result of the Laplacian creates a binary image with patches of scattered white pixels on a black background. The patches correspond to the focused regions of the image and the scattered pixels in them must be consolidated before the binary image can be used as a mask. To merge the isolated white pixels I apply multiple dilations followed by multiple erosions. The Average filter is used after each dilation or erosion to round out the region boundaries, which become blocky because of the square dilation and erosion probes. Eroding once less than dilating makes the foreground regions of the mask a little bigger than the original patches and leaves a fringe of blurred material around each focused region when the mask is "Anded" with the original image. As a result, the deblurred images overlap slightly and gaps in the final top-view or solid-model images are minimized.

The method described in *DEBLUR_IMAGE* works well for colored images, but the mask should be built from a gray-scale copy of the image. When deblurring a series of images, experiment with the first one to determine whether to use Laplace1 or Laplace2, the threshold value F1 (usually just above the noise level of the image produced by the Laplacian), the size of the dilation probe (typically 7 x 7), and the number of dilation cycles. These choices will not change for the remaining images.

Moving next to the topic of combining the deblurred images, refer to the algorithm *MAKE_TOPVIEW*, which describes the procedure for creating the top view image. Because there are overlapping regions in the deblurred images, the deblurred images cannot be combined by simply adding them together. Instead, I use the Max operator, which merges two images by keeping the brighter pixel of each corresponding pair. The Max operator is a good choice for bright-field, reflected-light images because the focused pixels tend to be brighter than the blurred ones. If there are gaps (missing pixels) in the top view, these can be filled from one of the original, blurred images (from the top, middle, or bottom of the stack depending on whether the gap corresponds to a peak, slope, or valley of the object).

Algorithm *MAKE_TOPVIEW*

```
//Copy bottom image in deblurred stack
TOPVIEW ← ( DEBLURRED_1
//Combine remaining images with copy
TOPVIEW ← Max ( TOPVIEW, DEBLURRED_2 )
...
TOPVIEW ← Max ( TOPVIEW, DEBLURRED_N )
//Optional steps to fill gaps in TOPVIEW
//Copy TOPVIEW and convert copy to gray scale
MASK ← TOPVIEW
If MASK is colored, convert MASK to gray scale
//Create binary mask with gaps in foreground
FOREGROUND ← 0, 0 //Gaps are black
MASK ← GrayToBinary ( MASK, FOREGROUND )
//Get filler from an original image (user selected)
FILLER ← And ( ORIGINAL_I, MASK )
//Fill gaps
TOPVIEW ← Add ( TOPVIEW, FILLER )
```

Using the stack of deblurred images to create a solid model requires processing steps and specialized software that are not included in most image processing packages. However, the rewards for doing this are great because the object's topological information is preserved, and the solid model obtained can be rotated and sliced to provide a variety of additional perspectives.

To create the solid model, the stack of deblurred images must be converted into a 3D array of scalar data, which can be displayed by numerous technical plotting packages. The data file prepared for the plotting package will contain four columns: X, Y, Z, and *Pixel Value*. For each pixel, X and Y are its row and column indices in the 2D image and the Z indicates the location of its image in the stack. For example, if the images are equally spaced and the microscope stage was dropped between images, all the pixels of *DEBLURRED_1* are assigned $Z = 0$, those from *DEBLURRED_2* are assigned $Z = 1$, etc. The plotting package displays the pixel by placing a rectangular prism at the coordinates X, Y, and Z; using the color of the prism to represent *Pixel Value*. Since the plotting is done using pixel coordinates, the user must adjust the scales of the X, Y, and Z axes to preserve the aspect ratios of the object. Actually, some technical plotting packages (e.g., T3D by Fortner Software, LLC) can read a series of images

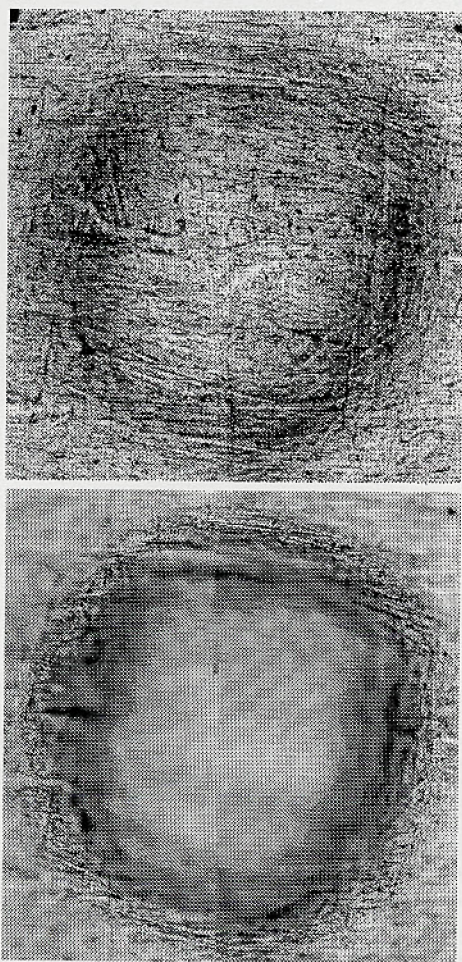


Fig. 2. Images of dot on back of U.S. penny. Top: Top view, Bottom: Original (fourth from bottom in vertical stack of 14)

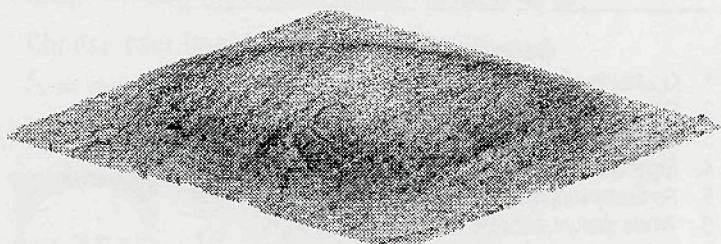


Fig. 3. Solid model of dot on back of U.S. penny.

(TIFF, BMP, etc.) at the click of a mouse button and perform this transformation automatically.

Examples of images created using the procedures described in this article are shown in Figs. 2 and 3. The object in this case is the raised dot between the *E* and the *P* in the inscription *E•PLURIBUS•UNUM* found on the back of a U.S. penny. The top-view and solid-model images were constructed from a stack of 14 two-dimensional images taken with a 50X objective using bright-field, reflected-light illumination. The pixel dimension in the image plane was $0.28 \mu\text{m}$ and the vertical spacing between images was $2 \mu\text{m}$. One of the 2D images from the stack (fourth from the bottom) is shown on the left in Fig. 2, dramatically illustrating the increased depth of focus obtained in the top-view and solid-model images. If you look closely at the original image, you will note that it is a montage of four images, a step required to obtain a field of view large enough to encompass the dot. All the X, Y, and Z movements of the stage were done manually, demonstrating that computer-automated staging is not required to achieve these results. ■

NORAN Instruments Inc. an internationally known manufacturer of x-ray microanalysis and confocal imaging instrumentation is seeking skilled and motivated individuals to join our Microanalysis Development Team

APPLICATIONS SCIENTIST

The ideal applicant will research and define scientific algorithms; participate in the creation of system requirements and specifications; design, implement and maintain software components; and present scientific papers at key conferences. A minimum M.S. (Ph.D preferred) in one of the Physical or Material Sciences or Computer Science, experience programming in C or