



Published in final edited form as:

Proc IEEE Southeastcon. 2014 March ; 2014: . doi:10.1109/SECON.2014.6950754.

Autofocusing for Microscopic Images using Harris Corner Response Measure

Madhu S. Sigdel*, **Madhav Sigdel***, **Semih Dinç***, **mren Dinç***, **Marc. L. Pusey^{||}**, and **Ramazan S. Aygün***

Madhu S. Sigdel: mss0025@uah.edu; Madhav Sigdel: ms0023@uah.edu; Semih Dinç: sd0016@uah.edu; mren Dinç: id0002@uah.edu; Marc. L. Pusey: marc.pusey@ixpressgenes.com; Ramazan S. Aygün: aygunr@uah.edu

*Department of Computer Science, University of Alabama in Huntsville, Huntsville, Alabama 35899, United States

^{||}ixpressgenes, Inc., 601 Genome Way, Huntsville, Alabama 35806, United States

Abstract

One of the difficulties for proper imaging in microscopic image analysis is defocusing. Microscopic images such as cellular images, protein images, etc. need properly focused image for image analysis. A small difference in focal depth affects the details of an object significantly. In this paper, we introduce a novel auto-focusing approach based on Harris Corner Response Measure (HCRM) and compare the performance with some existing auto-focusing methods. We perform our experiments on protein images as well as a simulated image stack to evaluate the performance of our method. Our results show that our HCRM-based technique outperforms other techniques.

I. Introduction

Image or photo capture may generate images where some sections of the image are blurred or out of focus. This is due to varying depth of the scene from the camera. In microscopy, a plate may have 3D objects that may appear at different levels of a solution. Therefore, a single image may have in-focus and out-of-focus regions. This is related to the depth of field of the camera. Areas that lay in the depth of field are in focus and the rest of the areas are out of focus. Therefore, a single image may not be sufficient to capture all details of an object.

Fig. 1 shows how images can be captured by changing the depth of field. Assume that initially the depth of field is located at the top indicated by solid lines. The areas of the object that lay in this depth of field are generated properly and the rest of the areas usually appear as blurred. To see details at other levels, the microscope lens is moved up and down, so that the depth of field also moves up and down. If the depth of field corresponds to the area indicated by dashed lines, the corresponding regions are in focus and the rest are out of focus. In this way, a series of images are captured by moving the depth of field. This is a

typical scenario for protein crystal growth trials where protein crystals may lie at different depths in a solution.

To resolve this problem, a sequence of images is captured by changing the depth of field. Since the depth of field is different for each image, each image may contain different focused regions. Then, we may either select the image that has overall clarity or fuse these images to obtain the best focused image.

Selecting the best focused image is a simple approach since it is usually based on the comparison of a clarity measure for each image independently. The general approach is to apply an objective function such as Laplacian, variance, Vollath F4 [1], Vollath F5 [1], entropy, etc. to find the best focused image. The comparison of selecting the best focused image has been studied in [2], [3], and [4]. Forero et al. [5] have discussed about the drawbacks of using Laplacian and variance as objective functions. One major drawback of this approach is that it does not benefit from clear and sharp parts that appear in other images.

Another approach to solve the problem is focal (or focus) stacking. Focal stacking aims to produce a single image with extended depth of field, hopefully with all regions under focus. Thus, it would be good to generate a single image combining the focused regions. This has been also considered as image fusion in some literature [6]. This approach can be categorized into two: pixel-based focal stacking and neighborhood-based focal stacking. In pixel-based focal stacking, every pixel at position (x, y) of all input stack images is compared using an objective function. The best pixel value is picked using a predefined selection criteria to obtain the final focused image. In neighborhood-based focal stacking, after picking the best pixel value at coordinate (x, y) as in pixel-based approach, it looks neighborhood around that location to achieve spatial consistency.

Image transformation is an alternative method to find the in-focus areas in image. An input image is transformed into a new space in order to analyze the details of the image. This approach was implemented by [6] using real and complex wavelet transforms. Bercovici et al. [7] provide a list of software (both free and commercial) which can be used to solve the issue. However, the majority of them are designed for photographers using “macro” mode of camera.

This paper introduces a novel focal stacking technique based on Harris Corner Response Measure (HCRM) [8]. In our experiments, we have chosen existing auto-focusing algorithms and compared with our approach. We have also used simulated data for quantitative comparison.

This paper is divided into 6 sections. The following section briefly discusses the types of auto-focusing techniques and existing algorithms considered in the experiments. In Section 3, our auto-focusing technique using Harris corner measure response is described. Data sets used in the experiments are described in Section 4. In Section 5, results and analysis of our experiments are provided. Finally, the last section concludes our paper with future work.

II. Auto-focusing Techniques

Let \mathbf{I} represent an image set $\{I_1, I_2, I_3, \dots, I_k\}$ and $|\mathbf{I}|$ represent the number of images in the set. These images are captured with varying depth of field. All these images have the same size, i.e., image resolution is $W \times H$. The pixel at (x, y) in i^{th} image I_i is represented as $I_i(x, y)$.

Auto-focusing technique can be categorized into 4 distinct categories: best-focused image selection, transform-based focal stacking, pixel-based focal stacking, and neighborhood-based focal stacking.

A. Best-focused image selection method

In this auto-focusing technique, an image is selected as the best focused image out of all the images in the set. It uses an objective function to determine image having the best clarity and details. Let $F_m(I)$ be an image function that measures the quality of image I with respect to method m . Let I_f represent the best focused image in \mathbf{I} . Let $BF(I, F_m)$ represent the function for finding best focused image in \mathbf{I} using objective measure F_m . $BF(I, F_m)$ is

computed as $BF(I, F_m) = I_f$ where $F_m(I_f) = \max_{1 \leq i \leq |\mathbf{I}|} F_m(I_i)$, $I_f \in \mathbf{I}$, and $1 \leq f \leq |\mathbf{I}|$.

Some of the objective functions with their mathematical expressions are listed below.

1) Vollath F4

$$F_{vol4}(I_i) = \sum_{x=1}^{W-1} \sum_{y=1}^H I_i(x, y) \cdot I_i(x+1, y) - \sum_{x=1}^{W-2} \sum_{y=1}^H I_i(x, y) \cdot I_i(x+2, y) \quad (1)$$

2) Vollath F5—

$$F_{vol5}(I_i) = \sum_{x=1}^{W-1} \sum_{y=1}^H I_i(x, y) \cdot I_i(x+1, y) - WH(\bar{I}_i)^2 \quad (2)$$

where \bar{I}_i is the average value of pixels in image I_i .

3) Normalized Variance

$$F_{normvar}(I_i) = \frac{1}{WH(\bar{I}_i)} \sum_{x=1}^W \sum_{y=1}^H [I_i(x, y) - \bar{I}_i]^2 \quad (3)$$

4) Laplacian

$$F_{lap}(I_i) = \sum_{x=1}^W \sum_{y=1}^H [I_i(x-1, y) + I_i(x, y-1) + I_i(x+1, y) + I_i(x, y+1) - 4 * I_i(x, y)]^2 \quad (4)$$

We consider Laplacian and VollaF4 in our evaluations since these give enough information about the performance of selecting the best-focused image.

B. Transform based focal stacking

In this method, an image transform is applied to the image set to determine regions that contain clear and sharp areas. For example, the complex wavelet transform (CWT) is applied to the input stack image set I to analyze image content [4].

C. Pixel-based Focal Stacking

In this approach, the best pixel at coordinate (x, y) is determined from input stack image set I . The best representative pixel value at that position is determined using an objective function and selection criteria. Laplacian is one of the objective functions used in this approach. Using a certain kernel function we find the Laplacian (L) value for every pixel position (x, y) . For each image in $I_i \in I$, a Laplacian image L_i is created. Then we use maximum selection criteria to determine best representative pixel at position (x, y) : $Pixel(x, y) = I_k(x, y)$ where $L_k(x, y) = \max_{1 \leq i \leq |I|} L_i(x, y)$ and $1 \leq i \leq |I|$.

D. Neighborhood-based focal stacking

This method is an improvement to pixel based focal stacking. While using selection criteria, it does not look only at a coordinate position (x, y) but also checks the neighborhood around the coordinate. After using the selection criteria, this method tries to satisfy spatial consistency. If pixels are taken from different images in a small neighborhood, this may cause discontinuity around the chosen pixel. The neighborhood-based method helps to eliminate or minimize discontinuity problem.

III. Auto-focusing Technique using Harris Corner Response Measure

In this section, we introduce our Harris Corner Response Measure(HCRM) based auto-focusing measure. Harris et al. introduced a measure for detecting corners in an image [8]. We use this measure as a basis of our auto-focusing technique. Before discussing our auto-focusing method, HCRM is briefly discussed.

Harris corner detector provides improvement to Moravec's corner detector. In Moravec's corner detector, the change in pixel intensities around a pixel is analyzed in four directions, and the minimum change is used as the interest measure for a pixel among changes in four directions. The basic idea is that the minimum change for a corner will still be high and this helps to distinguish a corner from an edge. The major limitation is the number of directions considered. To resolve this problem, Harris corner method uses the principal curvatures of a 2-dimensional local auto correlation matrix based on the first derivatives of an image. Let this matrix A is represented as

$$A = \begin{bmatrix} S_x S_x & S_x S_y \\ S_x S_y & S_y S_y \end{bmatrix} \quad (5)$$

where $S_x S_x$, $S_y S_y$, and $S_x S_y$ are obtained using multiplication of first derivatives (S_x , S_y) using a smooth circular window w such as Gaussian.

$$S_x S_x = \left(\frac{\partial I}{\partial x} \right)^2 \otimes w \quad (6)$$

$$S_y S_y = \left(\frac{\partial I}{\partial y} \right)^2 \otimes w \quad (7)$$

$$S_x S_y = \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) \otimes w \quad (8)$$

Then, Harris corner response measure at a specific pixel (x, y) is computed as

$$M(x, y) = \text{Det}(A(x, y)) - k (\text{Trace}(A(x, y)))^2 \quad (9)$$

where k is an empirically determined constant.

We can benefit from HCRM for auto-focusing problem. For example, consider an image having flat regions, a boundary, and a corner in Fig. 2. The right image shows these components when the image is out of focus. If an image is out of focus, pixels are smoothed by neighboring pixels. Hence, the difference in intensity between neighboring pixels (flat, corner, edge) of defocused image is less in comparison to focused image. Harris Corner Response Measure (HCRM) depends on the difference in intensity. Hence, in a focused image, the variation from a pixel to its neighbor will be higher than in defocused image. Using HCRM as an objective function we introduce our auto-focusing techniques: pixel-based autofocusing using HCRM and neighborhood-based autofocusing using HCRM.

A. Pixel-based focal stacking using HCRM

In this method, HCRM $M_i(x, y)$ is calculated for every pixel position (x, y) in each image in I . Then maximum selection criteria is used to determine the best representative pixel for

every position (x, y) : $\text{Pixel}(x, y) = I_k(x, y)$ where $M_k(x, y) = \max_{1 \leq i \leq |I|} M_i(x, y)$ and $1 \leq i \leq |I|$.

The pseudo-code for this algorithm is provided next in procedure *PixelBFS – HCRM* which returns the focused image. Input parameters to the algorithm are the input image stack (I), image width (W) and image height (H).

```

1: procedure PixelBFS-HCRM( $I$ ,  $W$ ,  $H$ )
2: for
 $x = 1$  to  $W$ 
do
3: for
 $y = 1$  to  $H$ 
```

```

do
4:  $M_{max} = 0$ 
5:  $img_{Mmax} = 0$ 
6: for
 $img\_i = 1$  to  $|I|$  do
7:  $M = HCRM$  at pixel  $(x,y)$  of image  $img\_i$ 
8: if
 $M > M_{max}$ 
then
9:  $M_{max} = M$ 
10:  $img_{Mmax} = img\_i$ 
11: end if
12: end for
13: final_img(x,y) = get_pixel(x,y) of  $img_{Mmax}$ 
14: end for
15: end for
16: return final_img
17: end procedure

```

B. Neighborhood-based focal stacking using HCRM

This method also uses HCRM as an objective function. This approach uses neighborhood information for determining the best representative pixel. Mode statistical tool is used to find the image which is mostly picked around the neighborhood as selection criteria. After using the selection criteria, this method tries to satisfy spatial consistency.

The pseudo-code for this algorithm is provided next in procedure *NeighborBFS – HCRM* which returns the focused image. Input parameters to the algorithm are the input image stack (I), image width (W), image height (H), neighborhood size ($xwsize \times ywsize$), and track table array (track). Track table array is used to track which image is used as the final pixel for every (x, y) position.

```

1: procedure NeighborBFS-HCRM( $I$ ,  $W$ ,  $H$ ,  $xwsize$ ,  $ywsize$ , track)
2:
3: //Find HCRM values for every pixel  $(x,y)$  for images
 $I$ 
4:  $M[img][x,y] = HCRM$  at pixel  $(x,y)$  of image  $img$ 
5:
6: //Generating final focused image
7: for  $n = 1$  to  $M.size()$  do
8:  $M_{cur} = n^{th}$  highest  $M$  value from array  $M[img][x,y]$ 
9:  $(x,y) = (x,y)$  value corresponding to  $M_{cur}$ 
10:  $img_{mod} = \text{Mode of } img \text{ value in track table in the neighborhood}$ 
11: if (  $img_{mod}$  is NULL ) then
12:  $img_{mod} = img$  value corresponding to  $M_{cur}$ 
13: end if
14: left_xwsize =  $-(xwsize-1)/2$ 
15: right_xwsize =  $(xwsize-1)/2$ 
16: above_ywsize =  $-(ywsize-1)/2$ 
17: below_ywsize =  $(ywsize-1)/2$ 
18: for  $dx = left\_xwsize$  to  $right\_xwsize$  do
19: for  $dy = above\_ywsize$  to  $below\_ywsize$  do
20: if ( track( $x+dx,y+dy$ ) is NULL ) then
21: final_img( $x+dx,y+dy$ ) = get_pixel( $x+dx, y+dy$ ) of  $img_{mod}$ 
22: track( $x+dx,y+dy$ ) =  $img_{mod}$ 
23: end if
24: end for
25: end for
26: end for

```

```

27: return final_img
28: end procedure

```

IV. Datasets

In this study, we applied auto-focusing algorithms to a protein crystal image set. 6 images of the same protein well with different depths of field were captured using the protein image acquisition system described in [9]. The size of all images used in the experiment is 320×240 . Fig. 3 shows the 6 images of a single protein-well used in this experiment. As can be seen in Fig. 3(d), Fig. 3(e) and Fig. 3(f), there is a region in the bottom-right that is visible only in these images. The details of this region are missing in other images. However, Fig. 3(b) and Fig. 3(c) provide better quality for the rest of the regions.

For quantitative analysis of our research we created simulated synthetic data of a texture image. Fig. 4 shows the texture images used in the paper. To simulate the images with different focal depth of a microscope from a single image, we applied Gaussian smoothing for varying depth of field. We mapped the image to 2D normal distribution model to create different focus level for a 2D texture (see Fig. 5). Then, using the height of each pixel as a smoothing parameter, we apply the smoothing partially for different parts of the image. Fig. 6 shows the set of 6 images with different focal depth for the texture image shown in Fig. 4(a).

V. Experiments and Analysis

Auto-focusing technique discussed in this paper is applied to a protein image set and two simulated image sets.

We have used the following filter as Laplacian pixel-based focal stacking:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

A window size of 15×15 pixels is used for neighborhood-based focal stacking. Accord.NET Framework (<https://code.google.com/p/accord/>) was used for calculation of Harris Corner Response Measure. Extended depth of field plugin of ImageJ application was used to test transform-based focal stacking using complex wavelet transform (CWT) [6]. We used fast speed and high quality parameters of this plugin in our experiments. The results of autofusing experiments for the protein image and simulated data sets are provided in Fig. 7 and Fig. 8, respectively. We provide the subjective and quantitative analyses of experiments below. We should note that to observe the results of experiments, the readers are recommend to zoom in these images on the digital file of this paper.

A. Subjective Analysis

1) Best-focused image selection method—While the Laplacian and VollathF4 methods have chosen different images for the protein image set (Fig. 7(a–b)), they have

chosen the same image for the simulated data set (Fig. 8(a–b)). For the protein image set, the Laplacian method picked the image having the bottom-right in focus. The VollathF4 method has chosen the image having the bottom-middle region in focus. It can be seen that each image has different areas in focus and out of focus. If objects appear at different depths, a single image may not capture all objects with clarity. The similar problem occurs even for a large 3D object that cannot be captured with a single depth of field. This method is only useful if objects are almost flat and lay at the same distance from the lens. In microscopic images, objects may appear at different depths and when crystals grow they may have a large 3D shapes. Therefore, these techniques are not beneficial for protein image sets.

2) Transform-based focal stacking—The results of applying autofocus using CWT are provided in Fig. 7(c–d) and Fig. 8(c–d). The fast speed and high quality versions both introduce noise for the final images. The enlargement of problematic regions for protein crystal images are shown in Fig. 9. The fast speed version (Fig. 9(a)) introduces another layer around the high intensity region. The smaller region had a gap or a black border in the middle. The high quality version performed little bit better than the fast speed version. In the high quality version, the bottom region has still some noise and does not have the expected smooth boundary (Fig. 9(b)).

3) Pixel-based focal stacking using Laplacian—Since pixel-based method using Laplacian picks the best pixel for every pixel location, this technique introduces significant discontinuity. This also gives an impression like noise is added to the final image. The problem can be observed in Fig. 7(e) and Fig. 8(e). This type of fusing images is not suitable for further image analysis.

4) Our pixel-based focal stacking using HCRM—This method also selects pixels from different images to get the final image. The noise problem that appeared in Laplacian version has been significantly reduced by the pixel-based HCRM (Fig. 7(f) and Fig. 8(f)). However, this method could not eliminate the discontinuity problem. We highlight this problem for protein crystal image in Fig. 10(a). Abrupt change in image intensity around the bottom region is clearly visible.

5) Our neighborhood-based focal stacking using HCRM—The result on protein image set is provided in Fig. 7(g). It maintains all the regions at proper clarity. It further minimizes the discontinuity in the final focused image with respect to its pixel-based version. The reduction in discontinuity can be visualized in Fig. 10(b). The discontinuity problem is minimized by picking the neighborhood pixels that minimize the noise when a certain pixel is chosen. This method worked well with both simulated and protein image data set. We should also note that the neighborhood size may affect the output quality. If two closed objects lie within the neighborhood size and are at different depths, then we may miss some details in the final focused image.

B. Quantitative Analysis—For quantitative analysis, we have chosen PSNR (Peak Signal to Noise Ratio) as the objective function. High PSNR values indicate good focused image generation. PSNR is calculated as follows:

$$MSE = \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H [X(i, j) - Y(i, j)]^2 \quad (10)$$

$$PSNR = 20 * \log_{10}(P_{max} / \sqrt{MSE}) \quad (11)$$

where X is the actual true focused image (ground-truth), Y is an output focused image, and P_{max} is the maximum pixel value of image X .

Since we need the ground-truth, PSNR values are computed on simulated data sets. PSNR values for both data sets using the techniques covered in this paper are provided in Table I. The table shows that the PSNR values (around 36.5) of best-focused image selection methods for Laplacian and VollathF4 are same because images selected as best focused for both data sets are same. Only pixel-based Laplacian performs worse than these best-focused image selection methods. Its PSNR values are in lower 30s. Transform-based focal stacking using CWT made a significant improvement by generating PSNR values above 40. The PSNR values of the fast speed version (around 45) is a little bit better than the PSNR values of the high quality version. This suggests that CWT with fast speed performs better for our data set.

Our pixel-based HCRM performed significantly better than its Laplacian version. It had around 16db to 21db on PSNR values with respect to the pixel-based Laplacian. In addition, our pixel-based HCRM also performed better than the fast speed version of the complex wavelet transform. Our neighborhood-based method performed as good as the pixel-based HCRM. However, our observations indicate that neighborhood-based HCRM introduces less discontinuity.

VI. Conclusion

In this paper, we introduced new auto-focusing technique using HCRM and compared with some existing auto-focusing techniques. Our autofocusing based on HCRM performed better than the others. Our neighborhood-based HCRM introduced lesser discontinuity than its pixel-based version despite their PSNR values are close. Our early results show some of the problems in the existing algorithms in the literature. Our HCRM-based autofocusing techniques may resolve those issues. We plan to apply our techniques in other domains and further protein image data sets.

Acknowledgments

This research was supported by NIH grant GM090453.

References

1. Vollath D. The influence of the scene parameters and of noise on the behaviour of automatic focusing algorithms. *Journal of Microscopy*. Aug; 1988 151(Issue 2):133–146.

2. Santos A, Ortiz De Solorzano C, Vaquero JJ, Pena JM, Malpica N, Del Pozo F. Evaluation of autofocus functions in molecular cytogenetic analysis. *Journal of Microscopy*. 1997; 188(3): 264272.
3. Osibote OA, Dendere R, Krishnan S, Douglas TS. Automated focusing in bright-field microscopy for tuberculosis detection. *Journal of Microscopy*. 2010; 240(2):155163.
4. Kimura, A.; Costa, MGF.; Filho, CFFC.; Fujimoto, LBM.; Salem, J. Evaluation of autofocus functions of conventional sputum smear microscopy for tuberculosis. *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*; 2010. p. 3041-3044.
5. Forero MG, Sroubek F, Cristbal G. Identification of tuberculosis bacteria based on shape and color. *Real-Time Imaging*. Aug; 2004 10(4):251–262.
6. Forster B, Van De Ville D, Berent J, Sage D, Unser M. Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images. *Microsc Res Tech*. 2004; 65(1–2):33–42. [PubMed: 15570586]
7. Bercovici A, Hadley A, Amadoz UV. Improving Depth of Field Resolution for Palynological Photomicrography. *Palaeontology Electronica*. Aug.2009 12(2):Article no 12.2.5T.
8. Harris, C.; Stephens, M. Plessey Research Roke Manor. United Kingdom: 1988. A Combined Corner and Edge Detector.
9. Sigdel M, Pusey ML, Aygun RS. Real-Time Protein Crystallization Image Acquisition and Classification System. *Crystal Growth & Design*. 2013; 13(7):2728–2736. [PubMed: 24532991]

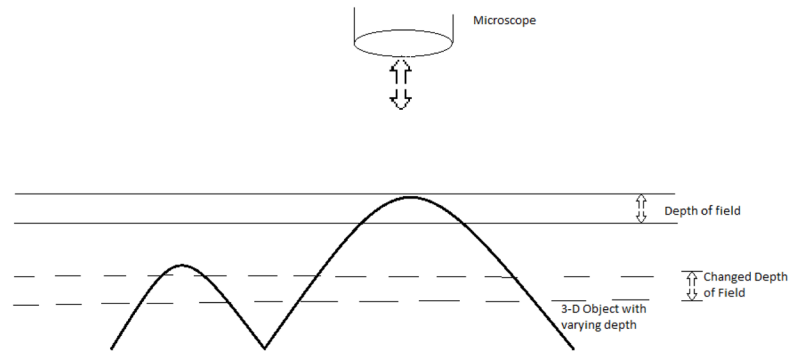


Fig. 1.
Depth of field illustration for a microscope.

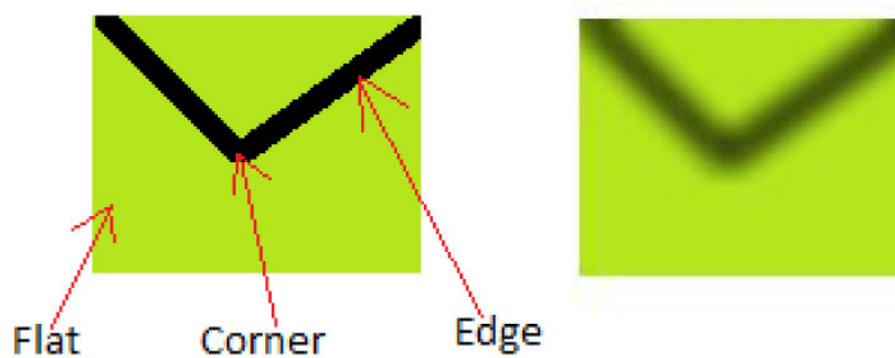


Fig. 2.
Focused and defocused image with flat, edge, and corner components

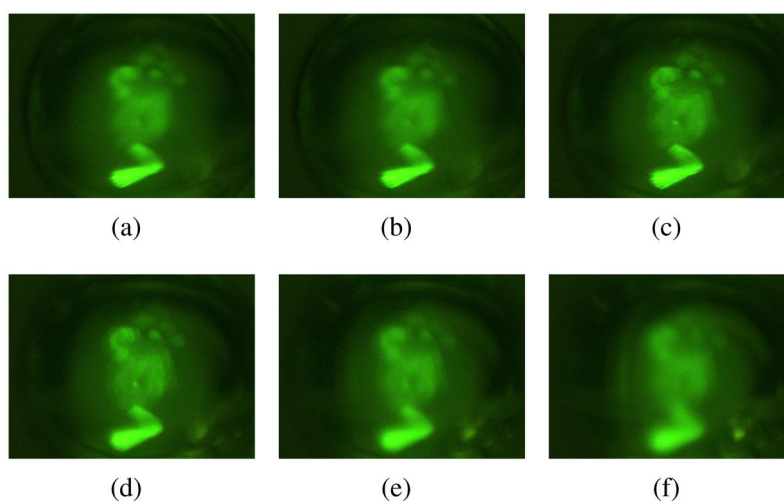


Fig. 3.
Protein crystal image (a–f) captured with 6 different focal depths

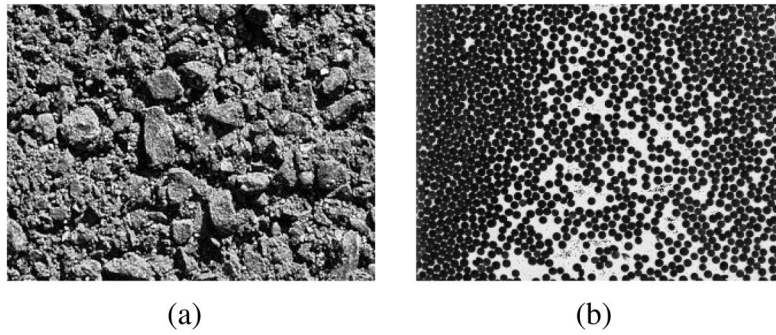


Fig. 4. Original texture image which was subjected for Gaussian blur to create 6 different focused simulated data stack.

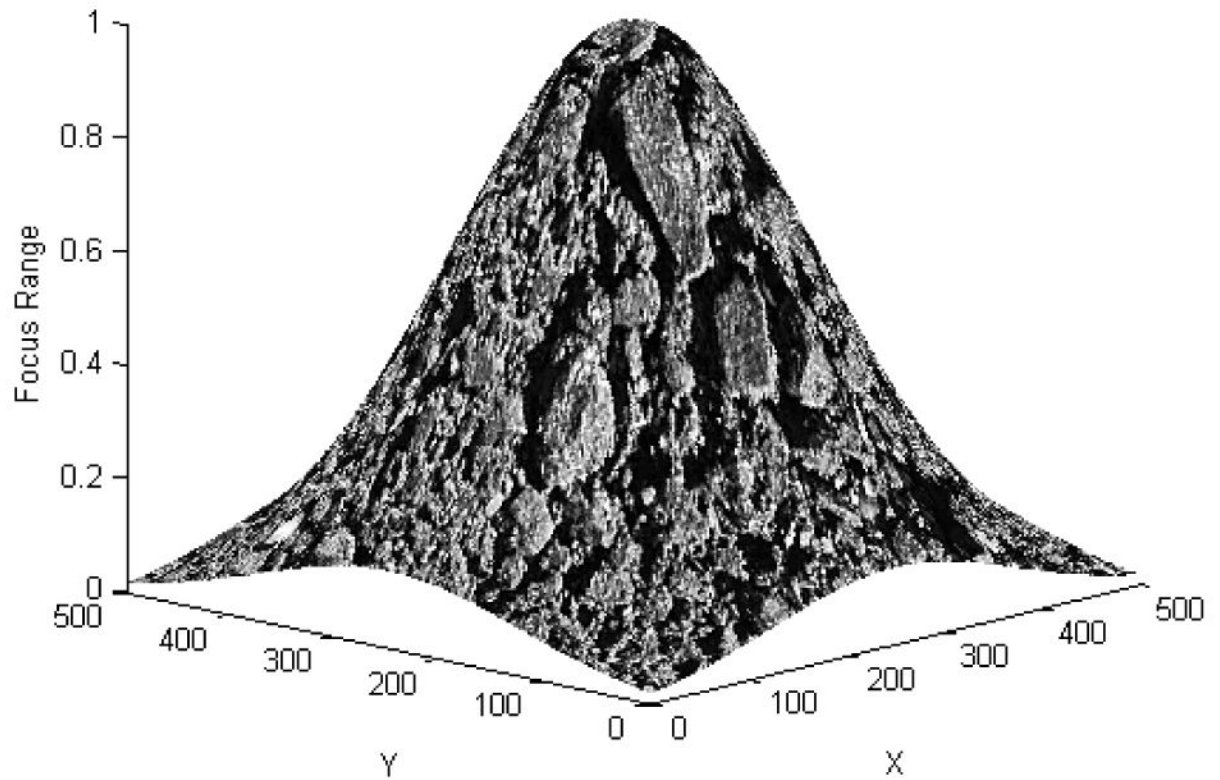


Fig. 5.
Application of 2D Normal Distribution on texture image shown in Fig. 4(a)

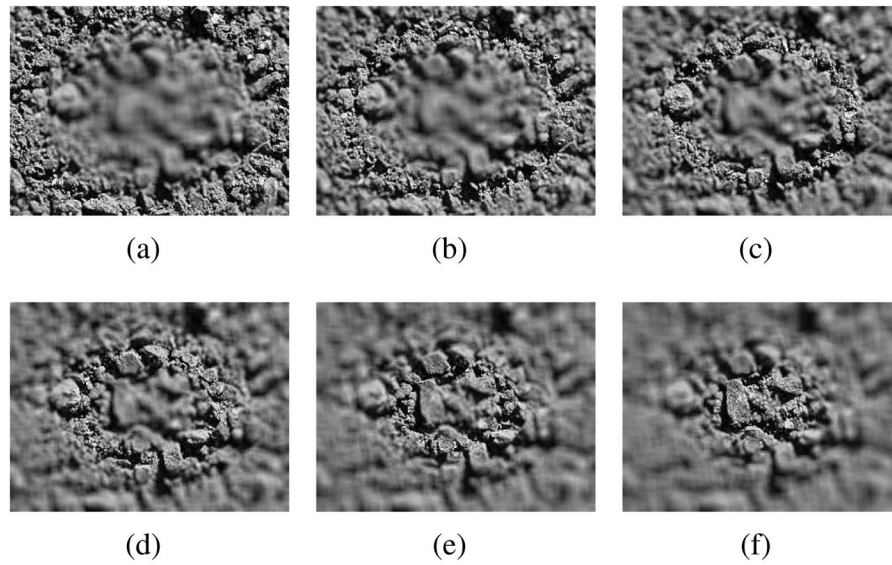


Fig. 6.
Simulated synthetic data for texture image in Fig. 4(a) created using Gaussian blur

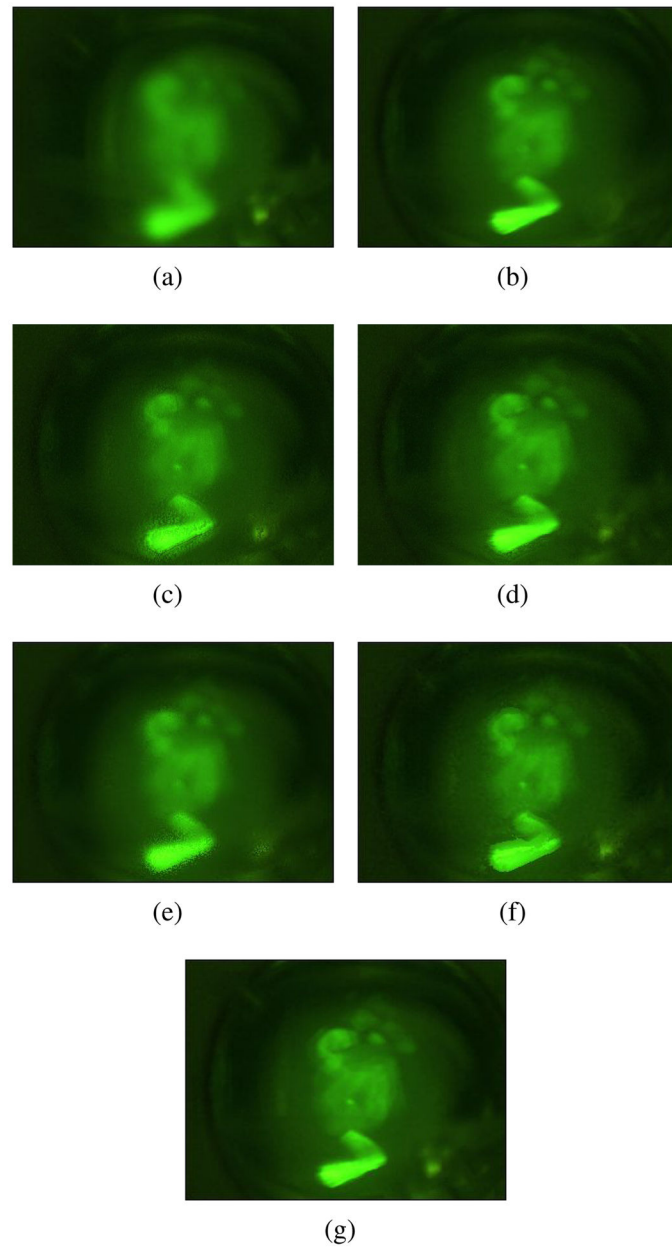


Fig. 7. Results of autofocusing experiments on protein image set (a) best-focused image using Laplacian (b) best-focused image using VollathF4 (c) CWT (fast speed) (d) CWT (high quality) (e) pixel-based Laplacian (f) pixel-based HCRM (g) neighborhood-based HCRM.

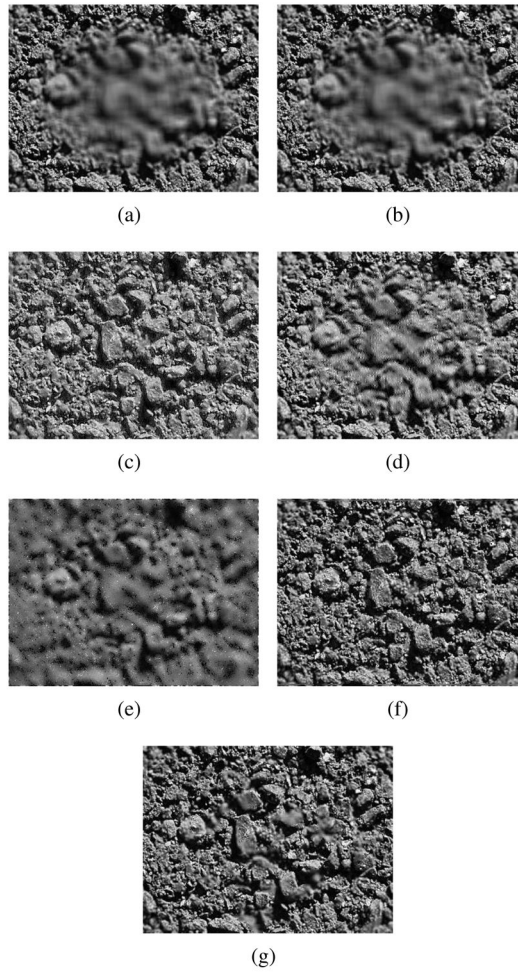
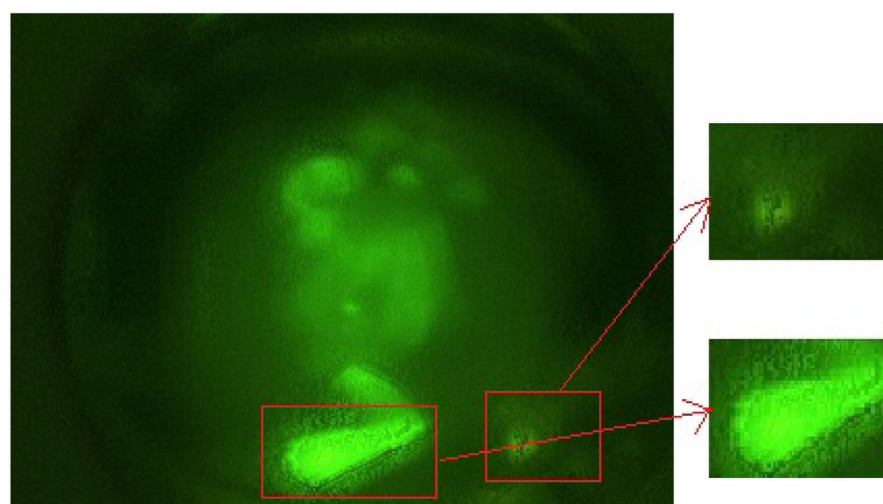
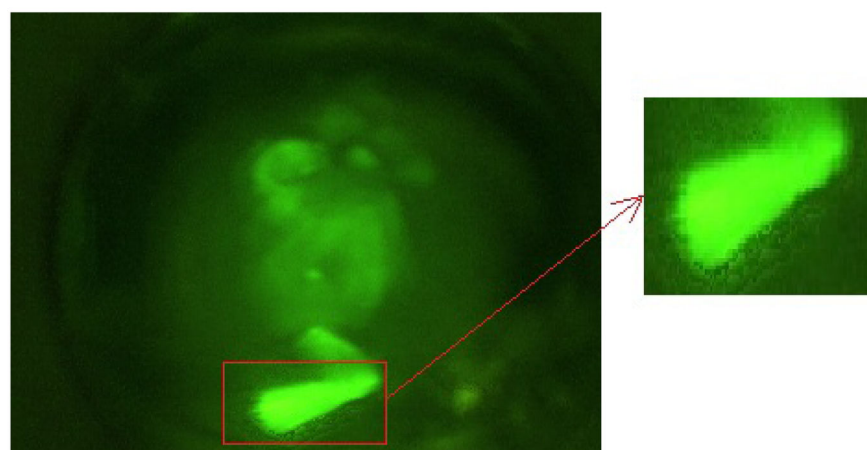


Fig. 8.

Results of autofocusing experiments on simulated data set shown in Fig. 6 (a) best-focused image using Laplacian (b) best-focused image using VollaF4 (c) CWT (fast speed) (d) CWT (high quality) (e) pixel-based Laplacian (f) pixel-based HCRM (g) neighborhood-based HCRM.



(a)



(b)

Fig. 9.
Problem with CWT based method (a) fast speed (b) high quality

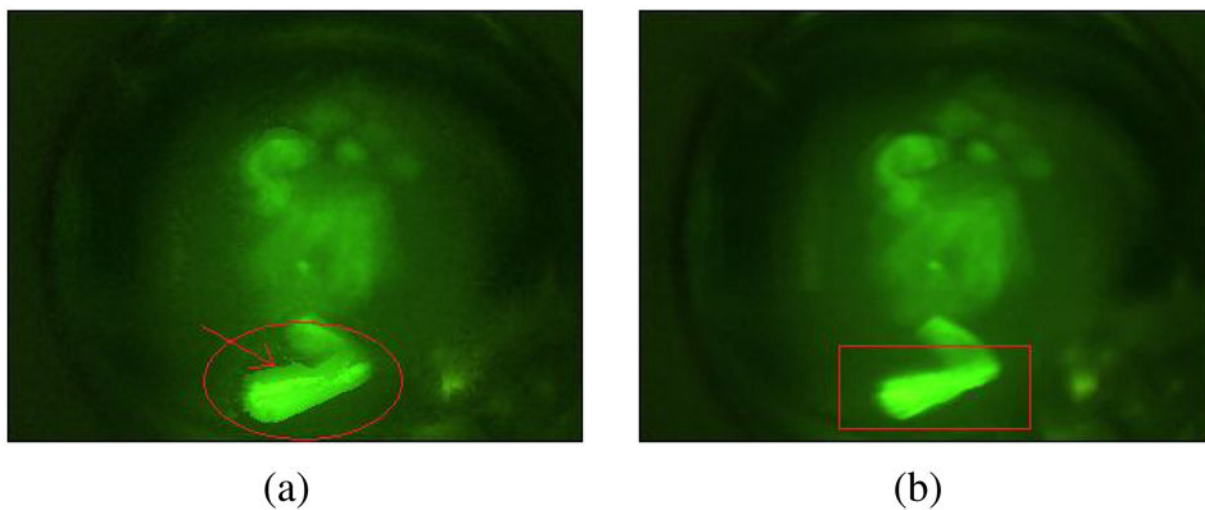


Fig. 10.
Analysis of focused image created by (a) Pixel-based HCRM (b) Neighborhood-based HCRM

Table I

PSNR values in db for different autofocusing

Auto-focusing technique	Dataset I	Dataset II
Best-focused image using Laplacian	36.5931	36.4619
Best-focused image using VollathF4	36.5931	36.4619
CWT (fast speed)	44.1436	45.0108
CWT (high quality)	41.1891	43.2657
Pixel-based Laplacian	31.0175	32.0342
Pixel-based HCRM	47.7262	53.8972
Neighborhood-based HCRM	47.0687	53.4681