

# FocusALL: Focal Stacking of Microscopic Images Using Modified Harris Corner Response Measure

Madhu S. Sigdel<sup>\*</sup>, Madhav Sigdel<sup>\*</sup>, Semih Dinç<sup>\*</sup>, İmren Dinç<sup>\*</sup>,  
Marc L. Pusey<sup>†</sup>, Ramazan S. Aygün<sup>\*</sup>

<sup>\*</sup>University of Alabama in Huntsville,  
Huntsville, Alabama 35899, United States

<sup>†</sup>iXpressGenes, Inc., 601 Genome Way, Huntsville, Alabama 35806, United States

<sup>\*</sup>{mss0025, ms0023, sd0016, id0002, aygunr}@uah.edu <sup>†</sup> marc.pusey@ixpressgenes.com

## Abstract—

Automated image analysis of microscopic images such as protein crystallization images and cellular images is one of the important research areas. If objects in a scene appear at different depths with respect to the camera's focal point, objects outside the depth of field usually appear blurred. Therefore, scientists capture a collection of images with different depths of field. Focal stacking is a technique of creating a single focused image from a stack of images collected with different depths of field. In this paper, we introduce a novel focal stacking technique, FocusALL, which is based on our modified Harris Corner Response Measure. We also propose enhanced FocusALL for application on images collected under high resolution and varying illumination. FocusALL resolves problems related to the assumption that in focus regions have high contrast and high intensity. Especially, FocusALL generates sharper boundaries around protein crystal regions and good in focus images for high resolution images in reasonable time. FocusALL outperforms other methods on protein crystallization images and performs comparably well on other datasets such as retinal epithelial images and simulated datasets.

**Index Terms**—Focal stacking, auto-focusing, Harris corner response measure, protein crystallization

## 1 INTRODUCTION

Imaging technology has become a critical module of scientific analysis systems in biochemistry, physics, and space sciences. Microscopy imaging enables researchers and experts to visualize and analyze microscopic world. Although there have been significant improvements on many aspects of imaging technology, focusing of objects is still a problem for many applications. Image acquisition systems are usually equipped with a camera that can only capture objects in focus if they lay in the depth of field of the camera. To capture other objects in focus, the microscope lens can be moved up or down to update the depth of field accordingly. Changing the depth of field does not solve the problem since there is no single in-focus image that covers all objects. As such, scientists are required to analyze a series of images since each image has only a section or region in focus.

Depending on the problem domain, focusing problems are dealt with 1) by adjusting the level or focal point of the camera to generate the best in focus image using a single depth of field, or 2) by fusing in-focus regions from multiple images that are captured with different depths of field. The first method is usually named as “auto-focusing” while the second one is usually termed as “focal stacking” in the literature. The microscopic images such as protein images may have 3D objects that can appear at different levels of a solution. If objects appear at different depths, auto-focusing usually fails. Focal stacking algorithms may also fail due to several assumptions made while fusing images:

(a) The contrast of a region will be higher when it is in focus with respect to when it is out of focus.

(b) The brightness of a region is higher when it is in focus compared to when it is out of focus.

There are also a few challenges of focal stacking:

- (a) There may be discontinuities in the final image, since pixel values are obtained from a set of images.
- (b) Since images are captured at different times, the lighting conditions may change.
- (c) The size of an object when it is in focus and out of focus might be different. Typically, perspective model as in pinhole camera model is observed when regular cameras capture images. However, the fused image follows orthographic projection model.

The target domain of our focusing problem is protein crystallization images. The major drawback of automatic protein crystal scoring systems is the possible miss of a crystal. Two good indicators of protein crystallization for fluorescence microscopy are high intensity and sharp edges around crystal regions. If the well illumination is poor or the captured image is out of focus, the automated techniques are likely to miss crystals due to deterioration of these indicators. It is important to process focused images not to miss protein crystals. Once a protein crystal is detected, an expert tries to optimize conditions to improve the quality of a protein crystal to be evaluated using X-ray diffraction [1].

In this paper, to address the limitations of assumptions and challenges of focal stacking algorithms, we propose a novel focal stacking method called *FocusALL* whose basics were introduced in [2]. Our FocusALL method benefits from Harris Corner Response Measure [3] to evaluate initial goodness of pixels. After ranking pixels based on this measure, FocusALL decides whether a pixel belongs to in-focus or out-of-focus

area using the neighborhood information. FocusALL can be adapted to images at any resolution after determining the base image resolution where HCRM performs effectively. FocusALL can also be adapted to images captured with varying illumination. The contributions of this paper are as follows:

- (a) FocusALL method can effectively identify low contrast and low intensity regions that are in focus,
- (b) Our enhanced FocusALL method can deal with high resolution images in reasonable time,
- (c) Our enhanced FocusALL technique generates sharper object regions for images collected under varying illumination, and
- (d) Our FocusALL method outperforms other compared work on our protein crystallization trial image set while also generating good results on other sample images.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. In Section 3, our FocusALL technique for focal stacking is discussed. Enhanced FocusALL technique for high resolution and varying illumination images are described in Section 4. In Section 5, the results and analysis of our experiments are provided. Finally, the last section concludes the paper with future work.

## 2 BACKGROUND AND RELATED WORK

This section provides a discussion of related work on auto-focusing and focal stacking.

### 2.1 Auto-focusing Techniques

Broadly, auto-focusing techniques can be categorized as active and passive. Both the active and passive auto-focusing techniques are limited by the range of depth of field since the output image depends on a single depth of field.

#### 2.1.1 Active Auto-focusing

In an image acquisition system, if the system allows selection of the object of interest and determines where the camera should be positioned with respect to its distance, it is called active auto-focusing. An active auto-focusing system is equipped with a special hardware that helps determine the correct position of the camera lens. Stauffer [4] describes an active auto-focusing system in which a beam of modulated energy is projected towards a subject. The special hardware provides a digital output indicating the position of the reflected energy, and thus the position of the object from the camera can be computed. The system finally captures the image using a single depth of field that is considered as the best depth of field. Bezzubik et al. [5] show how image contrast vary depending on the position of the stage relative to a microscope objective. The authors also propose an algorithm to find the best focusing position in a microscope. Active auto-focusing is generally expensive as it requires expensive hardware modification.

#### 2.1.2 Passive Auto-focusing

An alternate to active auto-focusing is passive auto-focusing where the best focused image is selected from a series of images captured at different depths of field. Let  $I$  represent an image set  $\{I_1, I_2, I_3, I_4, \dots, I_k\}$  and  $|I|$  represent the number of images in the set  $I$ . These images are captured with varying depths of field. All images in  $I$  have size  $W \times H$ . The pixel at  $(x, y)$  in  $i^{th}$  image  $I_i$  is represented as  $I_i(x, y)$ . In passive auto-focusing, an image is selected as the best focused image out of

all the images in the set. To define the best focused image, we use an objective function that provides a value for an image according to its clarity and details. Let  $F_m(I)$  be the function that measures the quality of image  $I$  using objective function  $m$ . Let  $I_f$  represent the best focused image in  $I$  and  $BF(I, F_m)$  represent the function for finding best focused image in  $I$  using objective measure  $F_m(I)$ . Then,  $BF(I, F_m) = I_f$  where  $F_m(I_f) = \max_{1 \leq i \leq |I|} F_m(I_i)$ ,  $I_f \in I$ , and  $1 \leq f \leq |I|$ .

In the literature, various quality measures have been proposed to evaluate image focus. Objective functions such as Laplacian, variance, Vollath-F4 [6], Vollath F5 [6], entropy, etc. are some basic examples of quality measures. Table 1 provides a list of some objective functions with their mathematical expression. Forero et al. [7] state that objective functions like Laplacian and variance do not benefit from clear and sharp parts that appear in all images.

It is hard to find a single algorithm that works best for all kinds of images. Some solutions have proven to be more efficient than others in particular domains. Comparative evaluation of these objective functions has been presented in several papers. For medical and biological images, many studies have shown that normalized variance and Vollath-F4 [6] are the best focusing measures ([8] [9] [10] [11] [12]). In [13], it has been shown that Vollath-F4 and Mid-frequency discrete cosine transform perform reasonably well in real-time autofocusing.

### 2.2 Focal Stacking

Focal stacking is a method of generating a focused image from images captured at varying depths of field by fusing in-focus areas. The objective is to generate a composite image with all regions in focus by selecting the in-focus pixels from the different image slices. Broadly, focal stacking algorithms can be classified into 3 classes: pixel based, neighbourhood based and transformation based methods ([14] [15] [16]).

#### 2.2.1 Pixel Based Focal Stacking (PBFS)

The most basic focal stacking method is the pixel based focal stacking (PBFS) where each pixel value at the corresponding position in all images is compared to determine the best in-focus pixel value. For an input stack image set  $I$  and pixel position  $(x, y)$ , the best representative pixel value is determined using an objective function and selection criteria. Laplacian is one of the commonly used objective functions. Using a certain kernel function, Laplacian ( $L$ ) value for every pixel position  $(x, y)$  is calculated. For each image  $I_i \in I$ , a Laplacian image  $L_i$  is created. The maximum selection criteria is then used to determine the best representative pixel for every position. At any position  $(x, y)$ ,  $I_f(x, y) = I_k(x, y)$  where  $L_k(x, y) = \max_{1 \leq i \leq |I|} L_i(x, y)$  and  $1 \leq k \leq |I|$ . This method can be used with different objective functions.

#### 2.2.2 Neighborhood Based Focal Stacking (NBFS)

This method is an improvement of pixel based focal stacking (PBFS). In PBFS, pixel values may be picked from different images. This may cause discontinuity around the objects in the final focused image. Neighborhood based focal stacking (NBFS) algorithms use neighborhood information to get appropriate value of a pixel to minimize the inconsistency [17] [18]. NBFS benefits from surrounding pixels rather than solely relying on pixels on the same projection. As in PBFS, an objective function is necessary to choose the best pixel value. Our FocusALL technique described in this paper is a NBFS method.

TABLE 1: Objective functions

Name	Objective Function ( $F_m(I)$ )
Vollath-F4	$F_{vol4}(I) = \sum_{x=1}^{W-1} \sum_{y=1}^H I(x, y) \cdot I(x+1, y) - \sum_{x=1}^{W-2} \sum_{y=1}^H I(x, y) \cdot I(x+2, y)$
Vollath F5	$F_{vol5}(I) = \sum_{x=1}^{W-1} \sum_{y=1}^H I(x, y) \cdot I(x+1, y) - W \cdot H \cdot (\bar{I})^2$
Norm Variance	$F_{normvar}(I) = \frac{1}{WH(\bar{I})} \sum_{x=1}^W \sum_{y=1}^H [I(x, y) - \bar{I}]^2$
Laplacian	$F_{lap}(I) = \sum_{x=1}^W \sum_{y=1}^H [I(x-1, y) + I(x, y-1) + I(x+1, y) + I(x, y+1) - 4 \cdot I(x, y)]^2$

### 2.2.3 Transformation Based Focal Stacking (TBFS)

In this method, each input image in spatial domain is first transformed into another domain. The image quality and details are then compared in that domain using some objective functions and comparison methods. After determining appropriate output results, the image is re-transformed to the spatial domain by applying inverse transform. In the literature, image fusion using various transformation methods such as discrete wavelet transform, complex wavelet transform, and curvelet transform have been proposed ([19] [14] [20] [21]). Forster et al. [15] propose complex-valued wavelet based image fusion algorithm. This method utilizes real and complex wavelet transforms to identify in-focus regions. The paper provides experiments on simulated image stacks as well as biological images. The complex wavelet based method is shown to outperform focal stacking using real-valued wavelet. One important thing to note is that there is a trade-off between capability of obtaining spatial details and the sensitivity to noise in wavelet transform technique [22]. Image fusion algorithm by combining curvelet and wavelet transform is described in [23]. A comparative analysis of different multi-resolution transforms for image fusion has been presented in [22].

To the best of our knowledge, there has been no studies on auto-focusing for protein crystallization images. The state of the art techniques did not provide satisfactory results on our protein crystallization images. In this paper, we introduce a novel focal stacking technique called FocusALL, and evaluate the performance with comparative methods in the literature. The details of our technique is described in the next section.

## 3 OUR FOCUSALL TECHNIQUE

In this section, we introduce our neighborhood based focal stacking technique named as *FocusALL*. To describe our method, we consider a set of 6 images of size 320x240 shown in Fig. 1 with different depths of focus (distance from the lens). This image set has 2 regions of interest. Regions  $R_1$  and  $R_2$  are best focused in images  $I_2$  and  $I_6$ , respectively.

### 3.1 Depth Color Image

For every pixel position  $(x, y)$ , the final focused image  $I_f$  contains the pixel from an image  $I_i$  in the input stack  $I$ . Let  $C_i$  represent the color for image  $I_i$ . Depth color image can be represented as  $C_I(x, y) = C_i$ , if pixel  $(x, y)$  is chosen as  $I_i(x, y)$  where  $1 \leq i \leq |I|$ . The depth color image gives an insight of the depth view of the objects. Fig. 2(a) is the focused image using the lowest intensity pixels and Fig. 2(b) is the corresponding depth color image. Blue, green, red, cyan, yellow, and pink

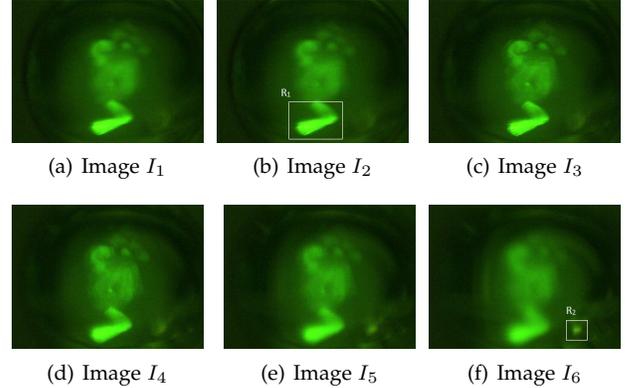


Fig. 1: Images of a protein crystallization sample captured with different depths of focus

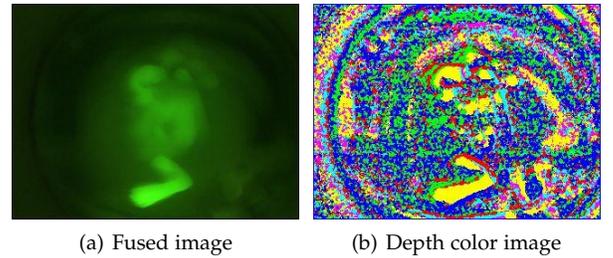


Fig. 2: Selecting lowest intensity pixels

colors represent pixels selected from images  $I_1, I_2, I_3, I_4, I_5$ , and  $I_6$ , respectively.

### 3.2 Harris Corner Response Measure (HCRM) and Our Modification

Harris et al. [3] introduced a measure for detecting corners in an image. Harris corner detector provides improvement to Moravec's corner detector [24]. 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  $A$  be represented as in Equation 1:

$$A = \begin{bmatrix} S_x S_x & S_x S_y \\ S_x S_y & S_y S_y \end{bmatrix} \quad (1)$$

where  $S_x S_x$ ,  $S_y S_y$ , and  $S_x S_y$  are obtained using product of first derivatives  $(S_x, S_y)$  using a smooth circular window  $w$

such as Gaussian as follows:

$$\begin{aligned} S_x &= \left(\frac{\partial I}{\partial x}\right) \otimes w & S_y &= \left(\frac{\partial I}{\partial y}\right) \otimes w \\ S_x S_x &= \left(\frac{\partial I}{\partial x}\right)^2 \otimes w & S_y S_y &= \left(\frac{\partial I}{\partial y}\right)^2 \otimes w \\ S_x S_y &= \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right) \otimes w \end{aligned}$$

Then, Harris corner response measure at a specific pixel  $(x,y)$  is computed as in Equation 2 where  $k$  is a constant.

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

The value of  $M(x,y)$  is high for the corner pixels. In out-of-focus image, pixels are smoothed by neighboring pixels. Hence, the difference in intensity between neighboring pixels of defocused image is less in comparison to the difference in intensity in the focused image. Harris Corner Response Measure (HCRM) depends on the difference in intensity. In a focused image, the variation from a pixel to its neighbor is expected to be higher than variation in defocused image. Therefore, it is reasonable to use this value as an objective function in focal stacking.

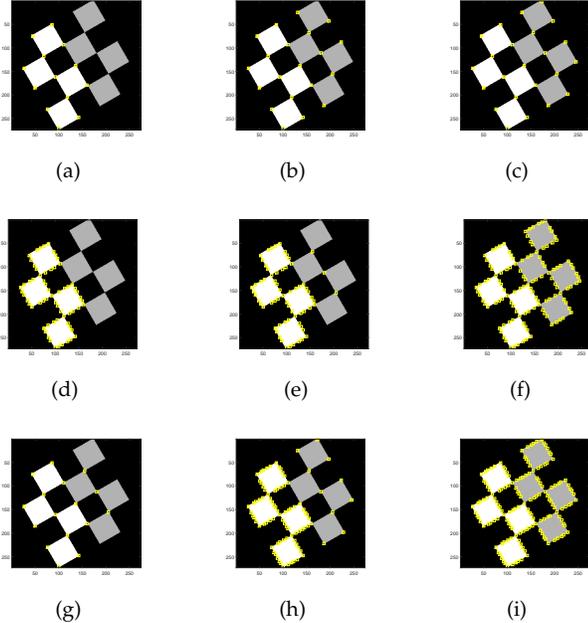


Fig. 4: Inclusion of pixels as threshold is reduced for a checkerboard image (a)-(c) Original Harris, (d)-(f) Trace, and (g)-(i) FocusALL

The  $M(x, y)$  is actually a function of eigenvalues ( $\alpha$  and  $\beta$ ) of the matrix  $A$  in Equation 1. These eigenvalues are correlated with the principal curvatures of the local auto-correlation function [3]. The determinant in  $M(x, y)$  can be computed as  $(\alpha * \beta)$  whereas the trace is equal to  $(\alpha + \beta)$ . The contours of  $M(x, y)$  with respect to  $\alpha$  and  $\beta$  are shown in Fig. 3(a). A good corner pixel is expected to have high positive  $M$  values. Negative values of  $M$  indicate edge pixels (if the values are above a threshold). Therefore, negative or low positive  $M$  values are eliminated for corner detection. Fig. 4(a-c) shows inclusion of pixels as the threshold for corner pixels is reduced for a checkerboard image. For positive values, only the number of corner pixels has increased in Fig. 4(b).

While HCRM can differentiate corners from edges, it gives little weight to edge pixels that has strong gradient in one direction. However, in focal stacking algorithm, both corners and edges are important. To address this, we modify the calculation of HCRM value. The key idea is to rank corner and edge pixels. If corner and edge pixels are given equal importance,  $M(x, y)$  can be represented with the trace of matrix  $A$  or the summation of eigenvalues. In such a case, the contours of  $M(x, y)$  would be as shown in Fig. 3(b). However, using the trace only may give more weight to edges. In Fig. 4(c-e), the darker corner pixels are added later than the other edge pixels as the threshold is lowered. To give more weight to corner pixels than edge pixels, the modified HCRM value is given in Equation 3, which is used as the objective function in our FocusALL method:

$$M(x, y) = \text{Det}(A(x, y)) + k(\text{Trace}(A(x, y)))^2 \quad (3)$$

The contours of this proposed measure are provided in Fig. 3(c). The curve of the contours is an indication of the emphasis on the corners pixels. A corner pixel with two low eigenvalues may be preferred to an edge with (one) high eigenvalue. Fig. 4(g-h) shows that using this measure adds first the corner pixels and then the edge pixels as the threshold is lowered. If the threshold is very low, the outputs of trace and the modified HCRM measure may yield the same outputs. It depends on the application domain whether to rank corner pixels higher than edge pixels or not. Since our algorithm gives precedence to pixels with high  $M$  values, it is reasonable to start around the corners and fill the regions around the corners accordingly while building the in-focus image.

### 3.3 FocusALL

In this section, we present our focal stacking technique, FocusALL, to generate the focused image from a stack of images collected under different depths of field. FocusALL is a neighborhood based focal stacking technique. NBFS uses neighborhood information to minimize discontinuity in the fused image. The modified HCRM in Equation 3 is used as the objective function in our technique. The two major steps in FocusAll are described next.

#### 3.3.1 Calculate representative HCRM value

In this step, for all images in the input stack  $I$ , HCRM value for every pixel is calculated. Then, the best representative HCRM value is determined for every pixel position. Let  $M_i(x, y)$  be the HCRM value for the pixel position  $(x, y)$  of an image  $I_i$  calculated as in Equation 3. Once all  $M_i(x, y)$  values are calculated, maximum selection criteria is applied to determine the best representative  $M$  for every position  $(x, y)$ :  $M(x, y) = \max_{1 < i \leq |I|} M_i(x, y)$ . The pseudo-code for this algorithm is provided in Algorithm 1. The algorithm takes image stack  $I$  as the input and returns a list with the attributes: HCRM value, image index  $i$ , and pixel position  $(x, y)$  for the best representative HCRM values for all pixel positions.

#### 3.3.2 Generate focused image

We generate an image by selecting best pixels from the images in input image stack  $I$ . Firstly, the best representative  $M(x, y)$  values obtained from step 1 are sorted in descending order based on HCRM values. To obtain the final focused image, we start by filling the pixels having highest HCRM values in

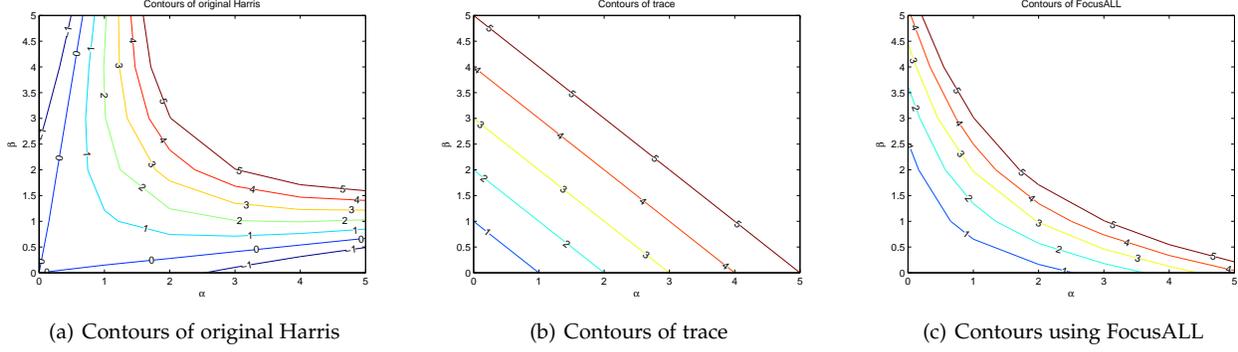


Fig. 3: Variation of contours with eigenvalues

---

**Algorithm 1** Find representative HCRM value for every position  $(x,y)$

---

```

1: Input:  $\mathbf{I}$  (Image stack)
2: Output:  $ObjList$  (Object array with attributes HCRM,
   imgIndx, x and y)
3:
4: procedure  $ObjList = \text{REP}HCRM(\mathbf{I})$ 
5:   //  $M_i(x,y)$  is HCRM at pixel  $(x,y)$  for image  $I_i$ 
6:    $i = 0$ 
7:   for  $x = 1; x \leq \mathbf{I}.Width; x++$  do
8:     for  $y = 1; y \leq \mathbf{I}.Height; y++$  do
9:        $M_{max} = 0$ 
10:       $maxIndx = 0$ 
11:      for  $k = 1$  to  $|\mathbf{I}|$  do
12:        if  $M_k(x,y) > M_{max}$  then
13:           $M_{max} = M_k(x,y)$ 
14:           $maxIndx = k$ 
15:        end if
16:      end for
17:       $ObjList[i].HCRM = M_{max}$ 
18:       $ObjList[i].imgIndx = maxIndx$ 
19:       $ObjList[i].x = x$ 
20:       $ObjList[i].y = y$ 
21:       $i++$ 
22:    end for
23:  end for
24: end procedure

```

---

the descending order. Let us consider the  $m^{th}$  highest HCRM value is for the position  $(x_m, y_m)$  and is obtained from image slice  $I_m$ . Also, consider the neighborhood window size is  $dx \times dy$ . To find the best pixels around  $(x_m, y_m)$ , we first determine the most frequently used image slice in the region  $(x_m - dx/2, y_m - dy/2)$  to  $(x_m + dx/2, y_m + dy/2)$  of the final focused image. In other words, we find the most repeatedly used image slice to fill the pixels around the neighborhood of  $(x_m, y_m)$ . We should note here that every pixel in the final focused image is obtained from one of the image in the stack. If none of the pixels in the region are filled already, the pixel values for this region is obtained from the image slice  $I_m$ . Otherwise, we fill the pixels values for all non-filled position in the region with the pixels from mostly used image. Suppose image  $I_f$  is the most frequently selected image slice in this region. Then, we fill all the non-filled pixels in the region  $(x_m - dx/2, y_m - dy/2)$  to  $(x_m + dx/2, y_m + dy/2)$  with the pixel values from  $I_f$ . This process is repeated with the next highest HCRM value until all the pixel positions are processed. At the end of the procedure, we obtain a focused image which we refer to as Full Harris Image (FHI). As more and more pixels are filled, the neighborhood information comes into use more often. Using the neighborhood information helps to maintain the spatial consistency. Besides, by filling the pixels with high HCRM

values first, we ensure we pick the best focused pixels in the final focused image.

The pseudocode for this algorithm is provided in Algorithm 2. The algorithm takes image stack ( $\mathbf{I}$ ), neighborhood size ( $dx, dy$ ) and the HCRM threshold, and returns the final focused image. The HCRM threshold is used to determine the pixels we would like to fill in the focused image. Only the pixels having HCRM values higher than the HCRM threshold are filled on the focused image. The focused image obtained using HCRM threshold 0 is called the Full Harris Image (FHI). Using 0 as the HCRM threshold ensures that representative pixels are determined for every pixel in the focused image. Fig. 5(a) shows the focused image for the protein crystallization trial image set shown in Fig. 1. The focused image has very few discontinuities, and all the objects are in focus. Fig. 5(b) shows the corresponding depth color image.

---

**Algorithm 2** Generate final focused image

---

```

1: Input:  $\mathbf{I}$  (Image stack),  $(dx, dy)$  (Neighborhood size) and
    $thres_{HCRM}$  (HCRM threshold)
2: Output:  $I_{harris}$  (FHI or PHI)
3: Note: If  $thres_{HCRM} = 0$ ,  $I_{harris}$  is FHI, else  $I_{harris}$  is PHI
4:
5: procedure  $I_{harris} = \text{GEN}IMGHARRIS(\mathbf{I}, dx, dy, thres_{HCRM})$ 
6:   //  $M_i(x,y)$  is HCRM at pixel  $(x,y)$  for image  $I_i$ 
7:    $ObjList = \text{rep}HCRM(\mathbf{I})$  //See Algorithm 1
8:   // Sort  $ObjList$  in descending order using HCRM value
9:    $\text{Sort}(ObjList, 'HCRM', 'Descending')$ 
10:  //Create a 2D array to keep track of selected image
   indices for each coordinate
11:   $track[ ][ ] = \text{NULL}$ 
12:  //Generate Harris image
13:  for  $i = 1$  to  $ObjList.size()$  do
14:     $x = ObjList[i].x$ 
15:     $y = ObjList[i].y$ 
16:    if (  $ObjList[i].HCRM \geq thres_{HCRM}$  ) then
17:      //Find index of most repeated image (Algorithm 3)
18:       $modeIndx = \text{findMode}(x, y, dx, dy, track, |\mathbf{I}|)$ 
19:      if (  $modeIndx$  is NULL ) then
20:         $modeIndx = ObjList[i].imgIndex$ 
21:      end if
22:      for  $p = -dx/2$  to  $+dx/2$  do
23:        for  $q = -dy/2$  to  $+dy/2$  do
24:          if (  $track(x+p, y+q)$  is NULL ) then
25:             $track(x+p, y+q) = modeIndx$ 
26:             $I_{harris}(x+p, y+q) = (M_{modeIndx}(x+p, y+q) >$ 
            $thres_{HCRM}) ? I_{modeIndx}(x+p, y+q) : \text{NULL}$ 
27:          end if
28:        end for
29:      end for
30:    end if
31:  end for
32: end procedure

```

---

---

**Algorithm 3** Find mode of representative images in the neighborhood
 

---

```

1: Input:  $(x, y)$  (Pixel Position),  $(dx, dy)$  (Neighborhood size),
    $track$  (2D array of image size that keeps track of selected
   image index for each coordinate,  $|I|$  (Number of images in image stack)
2: Output:  $modeIdx$  (Best representative image index at
    $(x, y)$ )
3:
4: procedure  $modeIdx = \text{FINDMODE}(x, y, dx, dy, track, |I|)$ 
5:   //Find frequency of image indices in the neighborhood
6:    $countList[] = \text{NULL}$ 
7:   for  $p = -dx/2$  to  $+dx/2$  do
8:     for  $q = -dy/2$  to  $+dy/2$  do
9:        $(countList[track(x + p, y + q)] + +)$ 
10:    end for
11:  end for
12:  //Find the mode image indexes in the neighborhood,
  return NULL if all count is 0
13:   $maxCnt = 0$ 
14:   $modeIdx = \text{NULL}$ 
15:  for  $k = 1$  to  $|I|$  do
16:    if  $(countList[k] > maxCnt)$  then
17:       $maxCnt = count[k]$ 
18:       $modeIdx = k$ 
19:    end if
20:  end for
21: end procedure

```

---

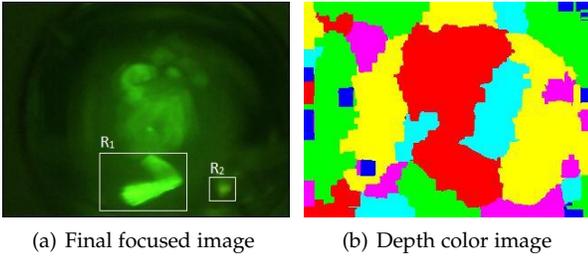


Fig. 5: Applying basic FocusALL

#### 4 ENHANCING FOCUSALL FOR HIGH RESOLUTION AND VARYING ILLUMINATION IMAGES

Automatic microscopic systems generally capture images in high resolution. The experts prefer to analyze the images in their original resolution, since some information or details may be lost after resizing or processing the images. As we mentioned in Section 2, focal stacking algorithms require processing every pixel in the image. Hence, applying focal stacking algorithms on high resolution images is time consuming and tedious. In addition, since the intensity difference between neighboring pixels is low in high resolution images, the objective function used for determining the clarity of the pixels may fail for these images. To resolve this, we propose enhanced FocusALL technique for high resolution images without significant sacrifice on the time efficiency of our algorithm.

Another challenge in focal stacking is that the lighting conditions may change while capturing the images. If we apply focal stacking on such image set, we may observe high discontinuity in the focused image due to pixels picked from images with different illumination. For such cases, basic FocusALL generates a focused image with discontinuities and artifacts. Therefore, we enhanced FocusALL to solve this problem.

##### 4.1 FocusALL for High Resolution (FocusALL-HR)

As the resolution of an image increases, the intensity difference between two neighboring pixels decreases. Since HCRM measures the change in intensity of neighbor pixels, edges and

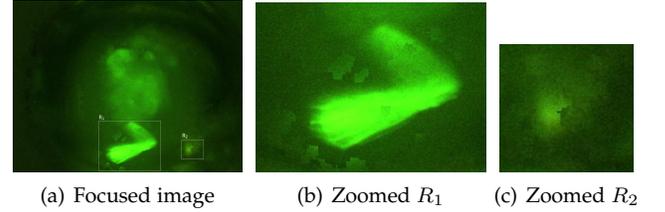


Fig. 6: Applying FocusALL to high resolution image

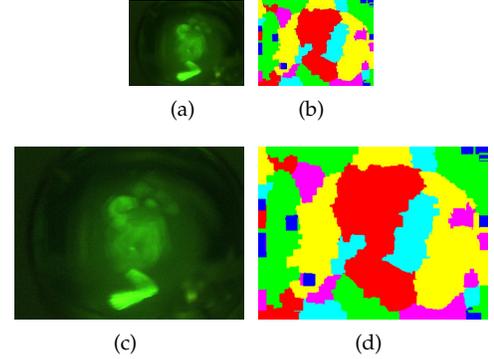


Fig. 7: Applying FocusALL-HR on a high resolution image. (a) Focused image at base resolution, (b) Depth color image at base resolution, (c) Enlarged depth color image, and (d) Focused image at high resolution

corner pixels may not be properly detected in high resolution images. Hence the basic FocusALL algorithm may not generate desired focused images for high resolution images. Fig. 6(a) shows the final focused image created by using basic FocusALL with 1280x960 resolution. Two regions are highlighted and the zoomed in versions are provided in Fig. 6(b-c), which shows discontinuities in the final focused image.

To solve this issue, we propose FocusALL for high resolution images (FocusALL-HR) by enhancing the basic FocusALL technique. We apply FocusALL on a base low resolution image as an initial step to obtain focused image in high resolution. The base resolution that FocusALL works properly with is determined empirically. Firstly, depth color map of the base resolution image is generated. Next, the depth color image is resized from base resolution to high resolution using interpolation. This step helps to generate appropriate depth color map for high resolution image. Then, using the enlarged depth color map and image slices in high resolution, final focused image is generated. Fig. 7(a) shows the focused image of the base resolution. Depth color map of the base resolution image is shown in Fig. 7(b). Fig. 7(c) shows the depth color image of high resolution image. Using the enlarged depth color map, the focused image is generated (Fig. 7(d)).

##### 4.2 FocusALL for Varying Illumination (FocusALL-VI)

Combining pixels from images with varying illumination to generate smooth focused image is quite challenging. Like any focal stacking algorithm, FocusALL also may not produce proper focus for these types of images. Because of illumination changes, the resulting focused image may consist of artifacts and discontinuities. Fig. 13(c) shows a set of 6 images collected

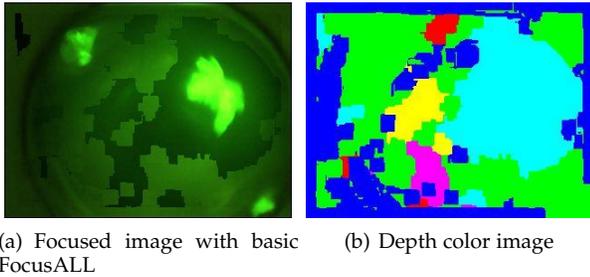


Fig. 8: Applying basic FocusALL on  $PC_3$

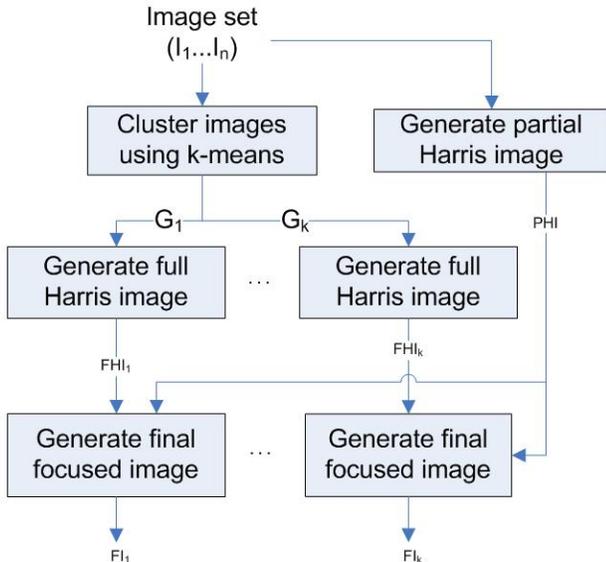


Fig. 9: Generating focused image for varying illumination

under different illumination. Here, the top two images in Fig. 13(c) have high illumination, while the bottom four images in Fig. 13(c) have comparatively lower illumination. Fig. 8 shows the final focused image after applying basic FocusALL. Here, objects are in focus in Fig. 8(a). There are several discontinuities and artifacts in the background. This is more obvious in the depth color image (Fig. 8(b)). Discontinuities are critical if they are observed inside object. We propose FocusALL for varying illumination images (FocusALL-VI) by enhancing the basic FocusALL technique for such cases.

Fig. 9 provides the basic flow of our proposed approach. Firstly, partial Harris image is obtained which separates the image pixels as object, background and holes. Next, images with similar illumination are grouped under each cluster and full Harris image is obtained from each cluster. To obtain the complete focused image, object pixels are obtained from the partial Harris image, holes are obtained using pixels from neighboring object pixel image and background is filled using the full Harris image. From  $k$  image clusters, we will have  $k$  resulting focused images. The expert can select one of these images as the best focused image.

#### 4.2.1 Generate Partial Harris Image (PHI)

The main objective of generating partial Harris image (PHI) is to get the best pixel values for the high contrast regions such as edges and corners from the image set  $|I|$ . Partial Harris Image (PHI) helps to distinguish high contrast object regions,

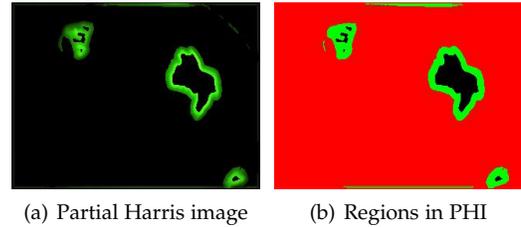


Fig. 10: Identify objects, holes and background in PHI

low contrast background regions, and low contrast internal object regions in an image. Normally, foreground regions are expected to have high HCRM values, while the background regions have low HCRM value. Using a threshold for HCRM value, we can generate an image by filling the pixels with high HCRM values only, which is called Partial Harris Image (PHI). Inside of a smooth in-focus object may have low HCRM value because of low intensity difference. Therefore, PHI may not fill up the pixels for an entire object completely, and thus holes can be present inside the objects. Fig. 10(a) shows the partial Harris image for the image stack shown in Fig. 13(c). Here, the holes are visible in all 3 regions in Fig. 10(a). PHI is generated using Algorithm 2. The image stack ( $I$ ), neighborhood size ( $dx, dy$ ) and HCRM threshold ( $thres_{HCRM}$ ) are provided as input. The procedure returns the PHI. Only the pixels having HCRM value greater than  $thres_{HCRM}$  are filled in PHI.

#### 4.2.2 Identify objects, holes and background in PHI

The pixels in PHI are divided into 3 regions: object, hole and background. All the object pixels in the PHI are clustered as object. All black regions inside the object regions are clustered as holes. Remaining region is clustered as background. Fig. 10(b) shows the resulting image after distinguishing these three regions. The object, hole and background regions are indicated by green, black and red colors, respectively.

#### 4.2.3 Cluster images using k-means

The images in image stack  $I$  are grouped using k-means clustering [25]. Firstly, intensity histogram is obtained for each image. The intensity histogram is input to the k-means algorithm as the features. The desired number of clusters should be provided as input. Accordingly, the images are grouped into each cluster. For the image set shown in Fig. 13(c), we applied k-means clustering with intensity histogram with 25 bins and 2 clusters. Using this procedure, the first two images in Fig. 13(c) fall under group  $G_1$ , and the rest of the images in Fig. 13(c) fall under group  $G_2$ . In the next section, we will explain how we generate full Harris image using these image groups.

#### 4.2.4 Generate Full Harris Image (FHI)

To generate the final focused image, we follow the steps of basic FocusALL with some modifications. We first calculate the representative HCRM values. Then, we generate Full Harris image (focused image) for each group using Algorithm 2. Here, the image stack  $G_1$  or  $G_2$  are the input to the FHI generating algorithm. Fig. 11(a) and Fig. 11(d) show the FHI generated from group  $G_1$  and group  $G_2$  separately. Let the FHI generated using group  $G_1$  be  $FHI_{G_1}$  and the FHI generated using group  $G_2$  be  $FHI_{G_2}$ . FHI images are used to fill regions of objects that have smooth low contrast inside the object.

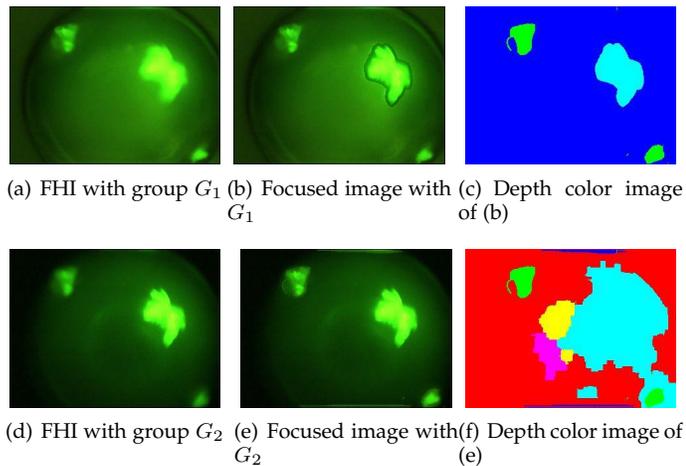


Fig. 11: Results of FocusALL-VI

#### 4.2.5 Generate final focused image

In this step, the 3 clustered regions, objects, background and holes in the PHI are filled separately to generate final focused image. Firstly, in the final focused image, the object regions excluding the holes are obtained from the PHI. Next, the hole regions, which are part of the object, are filled to get the complete object regions. Getting hole regions from random image may create discontinuity inside the object. Therefore, for each pixel position  $(x,y)$  in hole region, we first find the nearest object region by finding the image  $I_i$  from which the object region pixel was obtained. Then, we fill the hole at pixel position  $(x,y)$  using the pixel values at  $(x,y)$  from image  $I_i$ . Finally, background regions should provide good contrast with the complete object regions. Moreover, it is desired that the background region should have spatial consistency. Hence, these regions are obtained from either of the FHI (FHI $_{G_1}$  or FHI $_{G_2}$ ) to generate final focused image. Fig. 11(b) shows the final focused images using FHI $_{G_1}$  as the background region and Fig. 11(c) shows the corresponding depth color image representation. Similarly, Fig. 11(e) shows the final focused image using FHI $_{G_2}$  as the background region and Fig. 11(f) shows the corresponding depth color image representation.

## 5 EXPERIMENTS

In order to validate our FocusALL algorithm, we performed experiments on 3 protein crystallization image test cases:  $PC_1$ ,  $PC_2$  and  $PC_3$ . The images for protein crystallization trial sets were captured using the acquisition system described in [1]. The images are collected at a resolution 2560x1920. Each dataset consists of 6 images collected with different depths of field. Our protein crystallization datasets contain random scattered noise pixels. Thus, we apply median filtering with window size 3x3 prior to applying focusing algorithms. Fig. 13(a), Fig. 13(b) and Fig. 13(c) provide the images after median filter for the test cases  $PC_1$ ,  $PC_2$  and  $PC_3$ , respectively. We also evaluate the performance of our method on retinal pigment epithelial (RPE) images.<sup>1</sup> There are 4 images in the RPE image set provided in Fig. 13(d).

1. Images obtained from [http://bigwww.epfl.ch/demo/edf/demo\\_5.html](http://bigwww.epfl.ch/demo/edf/demo_5.html) (Courtesy of Peter Lundh von Leithner and Heba Ahmad, Institute of Ophthalmology, London).

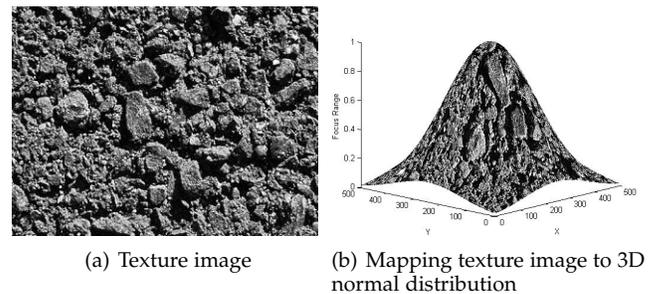


Fig. 12: Simulation of different focal depth on a texture image

For quantitative analysis of our algorithm, we created simulated data of a texture image shown in Fig. 12(a).<sup>2</sup> 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 first mapped the image to 3D normal distribution model to create different focus level for a 2D texture image (Fig. 12(b)). Then, using the height of each pixel as a smoothing parameter, we applied smoothing partially for different parts of the image. Fig. 13(e) shows the set of 6 images with simulated different focal depths. The resolution of the images is 320x240. The advantage with a simulated dataset is that we have a ground truth image to compare the results with.

We compare the performance of our FocusAll technique with other focusing algorithms. Vollath-F4 [6] has usually performed well in diverse domains. Therefore, we choose Vollath-F4 as the objective function for best-focused image selection method. As a transformation based method, we choose the complex wavelet transform (EDF-CWT) method since it provided good results in fluorescence microscopy [15]. To evaluate this, we use the extended depth of field (EDF) plugin for ImageJ application [15]. In addition to the EDF-CWT method in the EDF program, we evaluate the results using Sobel based method (EDF-Sobel), variance based method with window size 5 (EDF-Var5) and real valued wavelet transform (EDF-RW). For the real wavelet method, we select the medium quality option since it provided better result compared to the real wavelet medium high quality option. For the FocusALL algorithm, the default neighborhood size is 15x15 pixels. HCRM threshold value is determined empirically and chosen as 20.

### 5.1 Low Resolution Image

For low resolution, we downsample the images in Fig. 13 to 320x240 and apply the focusing algorithms. The *RPE* images are of size 321x256. Fig. 14 provides the focusing results using different techniques on 4 image sets ( $PC_1$ ,  $PC_2$ , *RPE*, *SIM*). The  $PC_1$  image set (Fig. 13(a)) has mainly 2 regions of interest highlighted as region  $R_1$  in the 2<sup>nd</sup> and 3<sup>rd</sup> images, and region  $R_2$  in the 6<sup>th</sup> image. In other words,  $R_1$  is best focused in the 2<sup>nd</sup> or 3<sup>rd</sup> image, and  $R_2$  is best focused in the 6<sup>th</sup> image of the set. The Vollath-F4 method selects the third image in the input set as the best focused image. The selected image has only one region in focus and the other region is barely noticeable. The focused images using EDF-Sobel and EDF-Var5 methods introduce significant noise in the final images. Moreover, the region  $R_2$  is not clear. The focused images using EDF-RW and EDF-CWT have both the regions in focus.

2. [http://www.textureking.com/content/img/stock/big/DSC\\_3518.JPG](http://www.textureking.com/content/img/stock/big/DSC_3518.JPG)

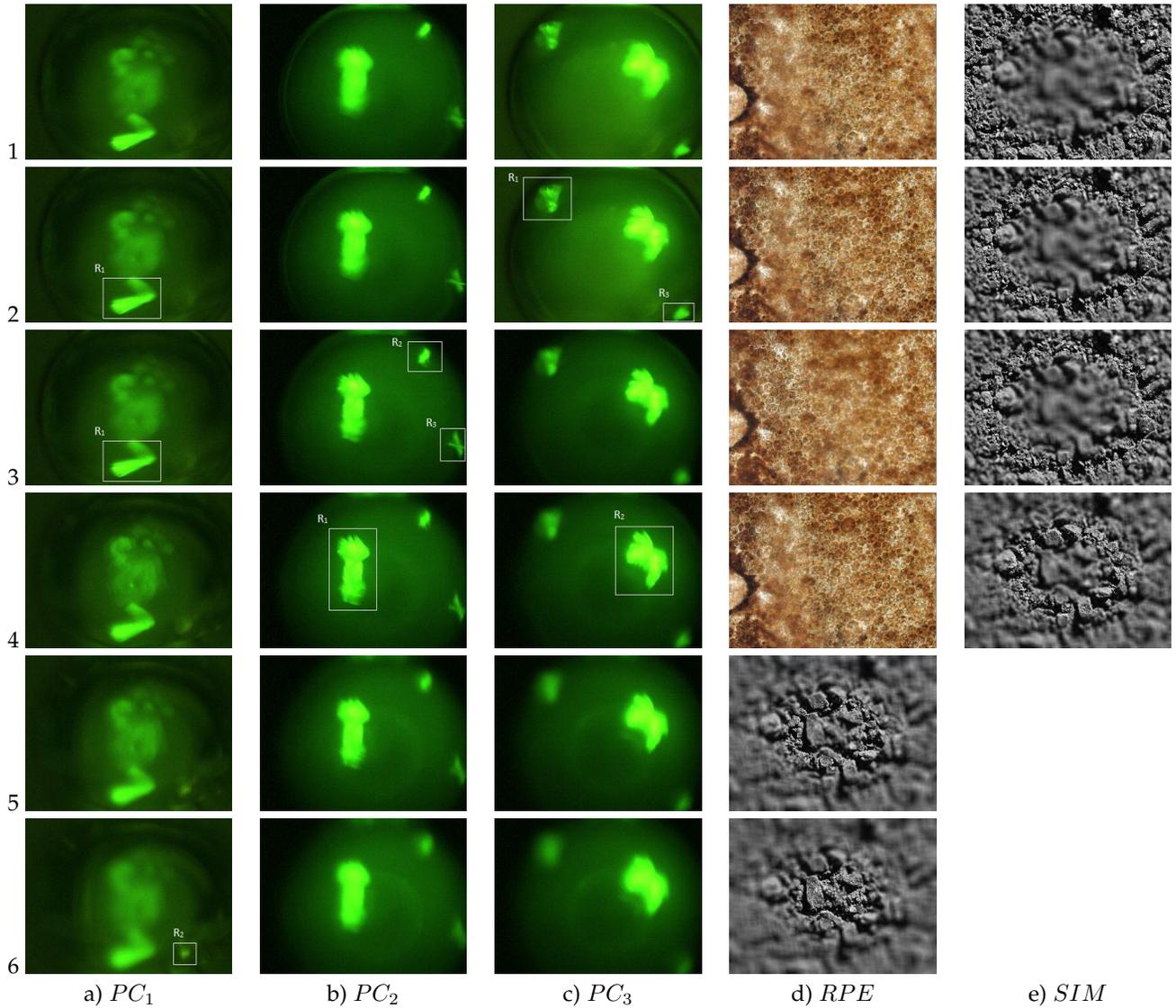


Fig. 13: Experimental dataset (images captured with different depths of field a) Protein images 1 ( $PC_1$ ), b) Protein images 2 ( $PC_2$ ), c) Protein images 3 ( $PC_3$ ), d) Retinal pigment epithelial (RPE) images, and (e) Simulated texture images

However, around the borders of region  $R_1$ , there are noise pixels and artifacts. The focused image using FocusALL has the regions of interest in good focus and has a good contrast with the background. Fig. 15 provides a zoomed in view of region  $R_1$  from the focused images using EDF-RW, EDF-CWT and FocusALL methods. The result from EDF-RW method shows artifacts around the region. The EDF-CWT method performed comparatively better than the EDF-RW method. However, there is random noise around the object. The  $R_1$  region using FocusALL has smooth boundary of the object and the discontinuity is minimized.

Fig. 14(b) provides the focusing results using different techniques for  $PC_2$  (Fig. 13(b)). This image set has mainly 3 regions of interest represented as  $R_1$ ,  $R_2$  and  $R_3$ . The region  $R_1$  is best focused in the 4<sup>th</sup> image of the set. Similarly, regions  $R_2$  and  $R_3$  are best focused in the 3<sup>rd</sup> image of the set. The Vollath-F4 method selects the 3<sup>rd</sup> image from the set as the best focused image. This image looks satisfactory although the edges in region  $R_1$  are not very sharp. The focused images from EDF-Sobel and EDF-Var5 have additional layers in  $R_1$  region.

There are lots of noise pixels around the regions of interest and the objects are distorted. The EDF-RW and EDF-CWT methods perform reasonably well on this image set. However, if we look closer on  $R_1$  region, we can see additional layers around the borders of the object. In the focused image from FocusALL method, all the regions of interest are clear. The edges of the objects are more noticeable compared to other results.

To analyze the performance of our method on image set from different domain, we selected the retinal epithelial images (Fig. 13(d)). On this image set, it is difficult to select the regions of interest. However, we can see the blurred regions in the input images. By combining the in-focus pixels from different images, we want to have sharp focus throughout the image. Fig. 14(c) provides the focusing results using different techniques. Here, instead of showing the regions of interest, the major problematic regions in the result images are highlighted. The focusing result using Vollath-F4 has the most blurred regions. The EDF-Var5 method has the best result. Other methods, EDF-Sobel, EDF-RW, EDF-CWT and FocusALL have relatively small blurred regions. All these methods result

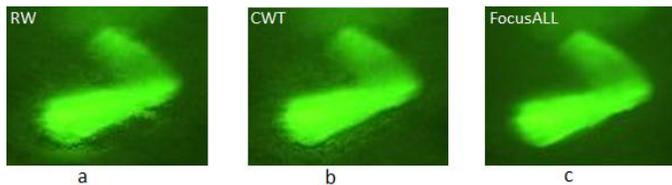


Fig. 15: Comparison of region  $R_1$  in focused images on  $PC_1$  (a) EDF-RW, (b) EDF-CWT, and (c) FocusALL

in a good focused image compared to any single image in the input set.

On the simulated dataset (Fig. 13(e)), each image has different regions blurred. It is difficult to show the regions of interest in this set. By applying the focusing algorithms, we expect to obtain the original image shown in Fig. 12(a). We can evaluate the focusing methods by comparing the resulting focused images with the original texture image. Similarly, we can also analyze the clarity of details and overall image sharpness. Fig. 14(d) provides the focusing results with different techniques. For each image, the problematic regions are shown in rectangular box. The outcome using best image selection method with Vollath-F4 is the most problematic. A large portion of the resulting image is out of focus. Similarly, the results with EDF-RW and EDF-CWT methods have large regions that are out of focus. The focused images with EDF-Sobel and our FocusALL (neighborhood size 3x3) have small blurred regions in different parts of the images. Nevertheless, these are satisfactory results and does not affect the details in the images very much. The focusing outcome with EDF-Var5 has the least image portion that is out of focus. Therefore, variance method provides the best outcome, and the results from EDF-Sobel and FocusALL methods are of acceptable quality.

## 5.2 High Resolution Image

To evaluate the performance of our method on high resolution images, we applied FocusALL-HR on  $PC_1$  and  $PC_2$  image sets at 1280x960 resolution. Fig. 16 provides the focusing results on  $PC_1$  and  $PC_2$  for different techniques. To highlight the problems, only the region  $R_1$  is provided for both the image sets. Since the best image selection method does not benefit from focused regions in different image slices, the result from best image selection method is not provided. Likewise, the EDF-CWT method performed better compared to the EDF-RW method. Therefore, the result from EDF-RW is not shown. The EDF-Sobel and EDF-Var5 methods introduce significant noise around the objects. This can be observed in Fig. 16(a)-(b) and Fig. 16(e)-(f). It is difficult to distinguish the object boundary because of several artifacts around the object. This is true for both the image sets. The result from EDF-CWT method and FocusALL-HR provide good contrast between the foreground and background. For  $PC_1$ , the result from EDF-CWT and FocusALL are similar. On  $PC_2$ , the EDF-CWT has some noise on the border of the object (Fig. 16(g)). FocusALL performed better on this data as the edges are clear, and the noise around the object is less. The outputs of EDF-RW, EDF-CWT, and FocusALL on low-resolution images look to be like the lower resolution of outputs generated from high resolution images. When EDF-Var5 and EDF-Sobel are applied on a high resolution image, it was observed that the outputs had more noise than the low resolution outputs.

In terms of the computation time, the Vollath-F4 best image selection (Vollath-F4), Sobel-based (EDF-Sobel), variance-based (EDF-Var5) and FocusALL methods complete in similar times. On a Windows 7 Intel Core i7 CPU @2.8 GHz system with 4 GB memory, the processing time for all these methods for 1280x960 image resolution was less than 10 seconds. The EDF-RW method took around 20 seconds to process the same resolution while the EDF-CWT method took around 40 seconds. As the image resolution goes higher, the computation time for the RW and CWT methods increases significantly. For image resolution 2560x1920, the CWT technique takes at least 10 minutes to generate the focused image. The complexity of the FocusALL algorithm does not increase with the increase in image resolution. This is because the main processing is done in base resolution. The depth color image obtained for base resolution is enlarged to determine the pixel selection on the desired high resolution.

## 5.3 Varying Illumination Images

We considered the protein crystallization image set  $PC_3$  shown in Fig. 13(c) for evaluating algorithms with respect to varying illumination. This test case has 3 regions of interest. The image resolution is 320 x 240. As explained in Section 4, the images were grouped into 2 clusters by applying K-means clustering on intensity histogram. Accordingly, first two images were grouped into cluster 1 and the remaining images into cluster 2. The results of FocusALL-VI and comparison with other techniques are provided in Fig. 17.

Using the best-focused image selection method using Vollath-F4, the second image in the set (Fig. 17(a)) is selected as the best focused image. Here, the regions  $R_1$  and  $R_3$  are in good focus but region  $R_2$  could be improved if it were picked from the 4<sup>th</sup> image in the set. Using the Sobel technique, the resulting image shown in Fig. 17(b) introduces significant noise throughout the image. The focused images using the variance method (EDF-Var5) (Fig. 17(c)), real wavelet (EDF-RW) (Fig. 17(d)) and complex wavelet method (EDF-CWT) (Fig. 17(e)) all have dark regions around regions  $R_1$  and  $R_3$ . The problematic regions are marked by red rectangle. For the FocusALL-VI, we had 2 clusters. Therefore, there are 2 focused images shown in Fig. 17(f) and Fig. 17(g). Using this method, all 3 regions are in good focus. The image in Fig. 17(g) looks better than the image in Fig. 17(f) since it does not have an artificial boundary around the large object region  $R_2$ . The expert can make selection among the two images for further analysis. We also performed experiments on varying illumination for high resolution images and get results similar to Fig. 17(a-g). The region  $R_2$  for varying illumination on high resolution images are shown for EDF-Var5, EDF-RW, EDF-CWT, and our FocusALL (from  $G_2$  cluster) techniques in Fig. 17(h-k). FocusALL generates sharper object regions than EDF-RW and EDF-CWT, and it does not have the noisy regions in the background as in EDF-Var5. FocusALL may generate artificial boundaries in the final focused image. Therefore, if the accuracy of the complete image is more critical than individual in-focus regions, EDF-RW may be preferred to FocusALL.

## 5.4 Quantitative analysis

For quantitative analysis of FocusALL method, we use PSNR (Peak Signal to Noise Ratio). A high PSNR value is expected for a proper focused image. PSNR is calculated as in 4:

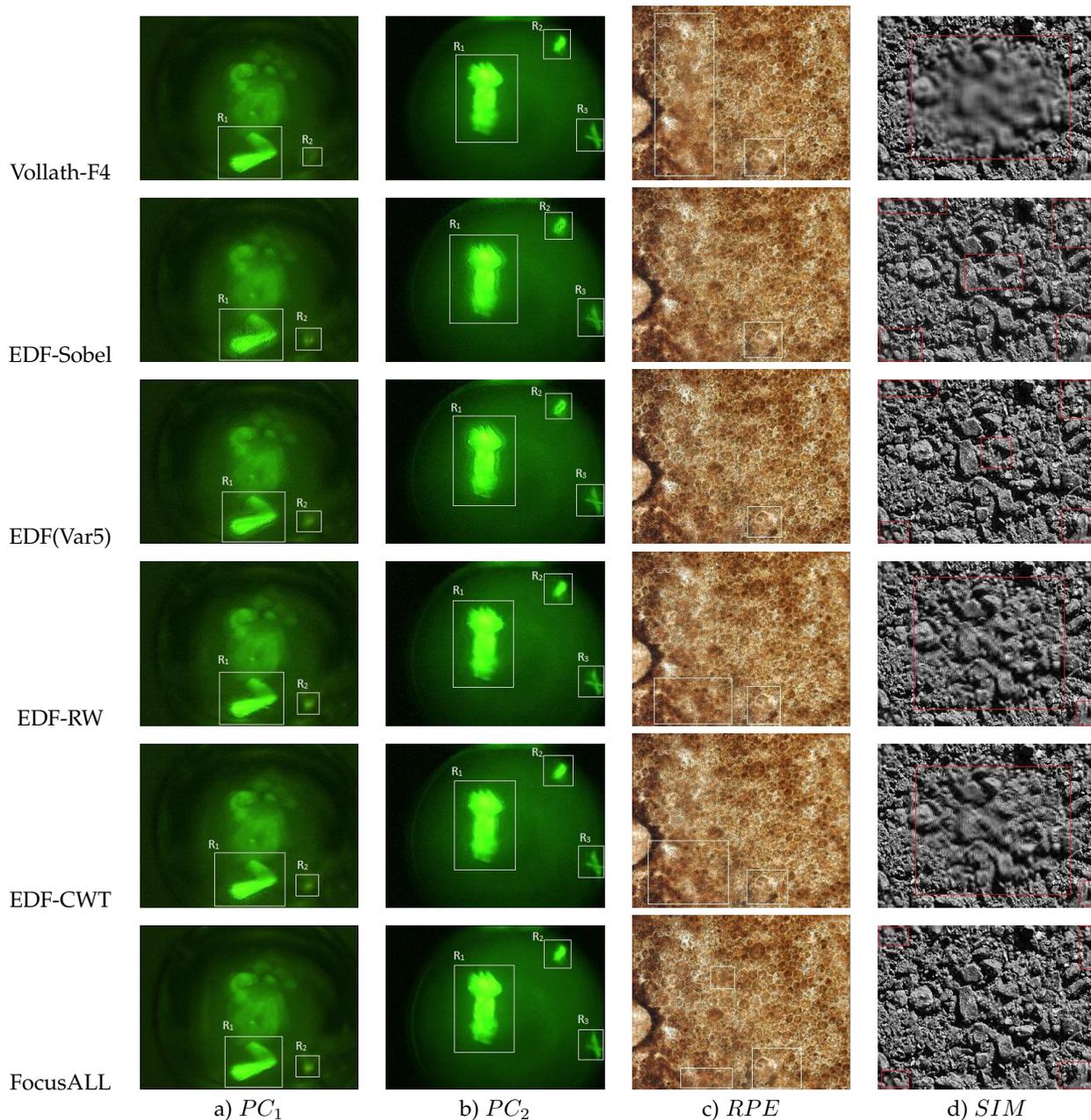


Fig. 14: Focusing results using different techniques (a) Protein crystallization images 1 ( $PC_1$ ), (b) Protein crystallization images 2 ( $PC_2$ ), (c) Retinal pigment epithelial (RPE) images, and (d) Simulated texture images

$$PSNR = 20 * \log_{10}(P_{max}/\sqrt{MSE}) \quad (4)$$

where

$$MSE = \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H [X(i,j) - Y(i,j)]^2 \quad (5)$$

In 5,  $X$  is the original image (expected image),  $Y$  is the focused image by using any algorithm, and  $P_{max}$  is the maximum pixel value of image  $X$ .

Table 2 gives the PSNR results for 3 simulated data sets using different techniques. The table shows that best-focused image selection method performs the worst as its PSNR value is the lowest. The EDF-RW and EDF-CWT have similar performances. The PSNR value is the highest for EDF-Var5 among all methods. Our FocusALL performs the second best after EDF-

Var5. This shows that our technique performs reasonably well on the simulated datasets as well.

TABLE 2: PSNR values in dB for different focusing techniques

Method	Texture 1	Texture 2	Texture 3
Vollath-F4	63.9	59.8	61.3
EDF-Sobel	66.9	63.9	65.3
EDF-Var5	<b>69.2</b>	<b>65.1</b>	<b>67.3</b>
EDF-RW	66.0	63.4	65.3
EDF-CWT	65.8	63.2	65.2
FocusALL	68.0	63.8	65.9

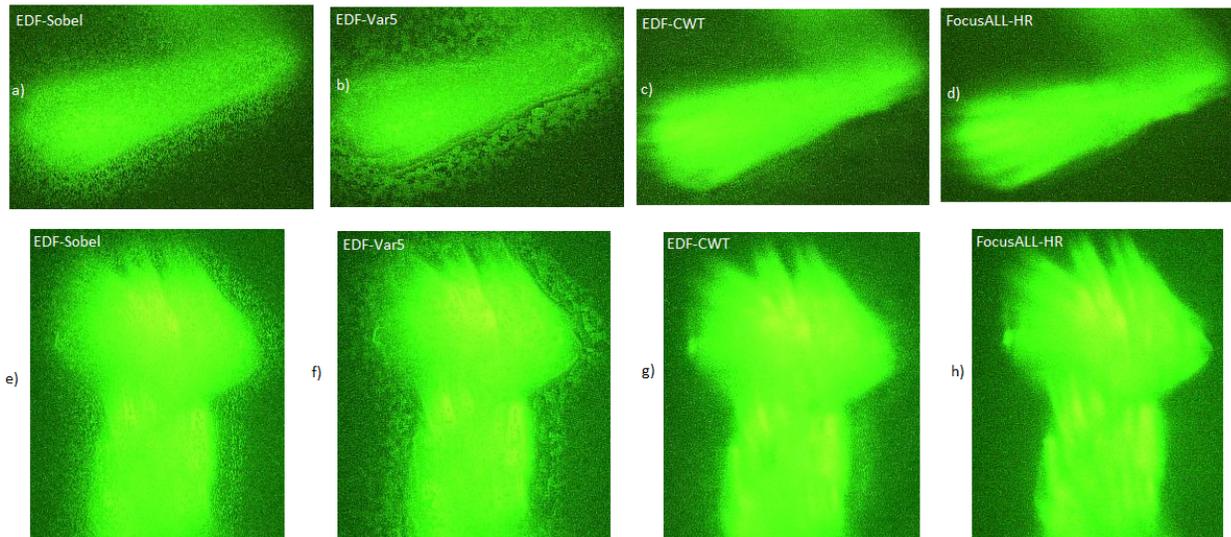


Fig. 16: Comparison of focusing results on high resolution (a)-(d) Results on region  $R_1$  of  $PC_1$  dataset, and (e)-(f) Results on region  $R_1$  of  $PC_2$  dataset

### 5.5 Comparison of different methods

It is difficult to have a focal stacking algorithm that works well for images in diverse domains. Manually selecting the methods and tuning the parameters for each image can be tedious and time consuming. Therefore, for automated systems, it is desirable to select a method that works well for most of the images for the particular problem. Our FocusALL method provided the best results on the protein crystallization images, and also generated reasonably good results on other datasets such as the retinal epithelial dataset and the simulated datasets.

In the experiments provided earlier, Vollath-F4 method picks up the overall best image from a given image set. The main problem for other methods is to pick up the best pixel for each pixel position. While CWT and RWT use wavelet coefficients, Sobel and variance use intensity change within neighborhood. Our method utilizes corner information to select the best pixel. For the discontinuity problem, CWT method checks consistency in sub-bands and spatial context (3x3 neighborhood). FocusALL method uses a window to fill the regions around a corner. In addition, the window size in FocusALL is used to deal with blurriness caused by high-intensity regions. These choices are the major differences between the techniques. If a method does not perform well for a specific data set, the pixel selection strategy and/or dealing with the discontinuity problems by that method does not work well for that data set.

The best image selection method using Vollath-F4 [6] method performed well based on visual inspection on  $PC_2$  but poorly on the other image sets. In  $PC_2$ , there are 3 regions all of which are clearly focused in the 4<sup>th</sup> image in the set. In this method, the best image in the set is selected. Since the pixels come from the same image, there is no discontinuity problem. Vollath-F4 has been proven to be an effective objective function for sharpness in many domains; however it requires at least one image to contain all objects in focus.

Focal stacking algorithms benefit by combining the infocus pixels in different images to get a clear composite image. However, the focal stacking algorithms have added complexity and chances for discontinuities in the final focused image compared to the best image selection method. We evaluated the

Sobel, variance, real wavelet transform, and complex wavelet transform based focal stacking available in Extended Depth of Field (EDF) [15] in our experiments. The Sobel technique worked well on the retinal epithelial image set and simulated datasets. However, the results were poor on protein images. Likewise, on the protein crystallization images, the complex wavelet transform method performed good for some images while several discontinuities and artifacts were produced in other images. This method performed reasonably well on the simulated data and the retinal epithelial data. On simple images all methods perform well. However, if images have artifacts that affect the neighboring pixel values in an image, the basic methods such as Sobel and variance start to perform poorly. EDF-RWT, EDF-CWT, and FocusALL can handle image datasets with complexities due to blurring of pixels better than Sobel and variance methods. However, EDF-RWT and EDF-CWT cause an additional layer or border around the high intensity regions.

The window size in FocusALL helps to deal with an artifact similar to diffused reflection that causes intensity and contrast increase in low intensity and low contrast regions next to high-intensity regions. This phenomenon occurs especially when the corresponding high-intensity region is out of focus. In our protein crystallization experiments, the default window size is 15x15 and FocusALL copes better than other techniques. If such an artifact is not part of the image set, the window size can be reduced to even 1x1.

FocusALL is more versatile than the other techniques as it can provide either the best or as good as the best technique for all experiments in this paper. It did not fail for any experiment while others did not provide good results for some of our protein crystallization experiments. FocusALL also produced reasonably good results on the retinal epithelial image set and the simulated datasets. Our proposed FocusALL-HR produces focal stacking results on high resolution images in a reasonable time. Similarly, our FocusALL-VI can deal with images of varying illumination. The modifications on the basic method are generic and hence can be applied on other focal stacking algorithms to improve the accuracy and computational time.

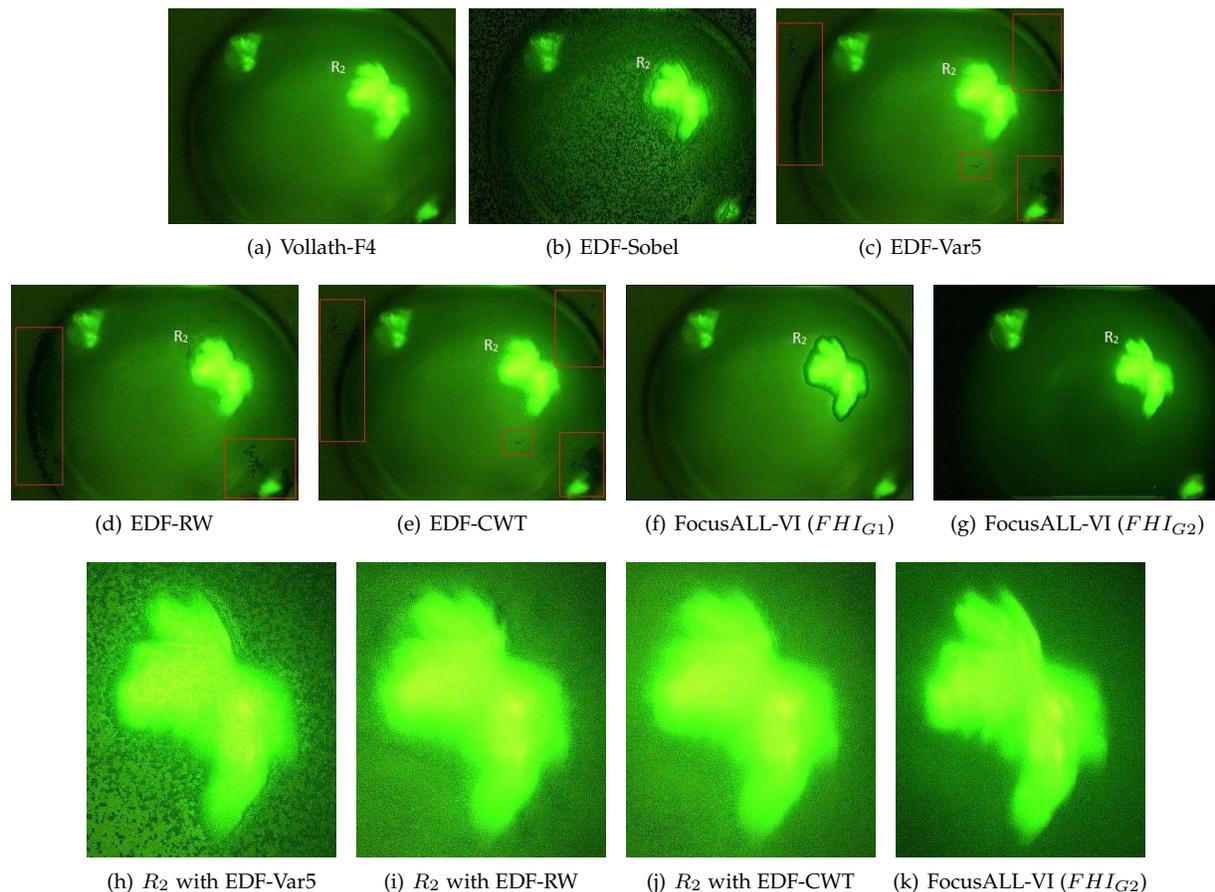


Fig. 17: Varying illumination results on  $PC_3$  (Fig. 13c), (a)-(g) Results on low resolution (320x240), and (h)-(k) Region  $R_2$  in high resolution (1280x960)

## 6 CONCLUSION

In this paper, we introduced a new focal stacking technique called FocusALL. FocusALL tackles two important assumptions that are not always true: a) high contrast regions belong to in focus regions and b) high intensity regions belong to in focus regions. Focal stacking techniques may yield discontinuities in the final image. FocusALL method minimizes the discontinuities by using neighborhood information.

We enhanced our FocusALL to overcome the limitations of focal stacking algorithms caused by high resolution and varying illumination images. Our FocusALL method together with the enhanced FocusALL methods performed the best results on the protein crystallization images. Besides, it produced reasonably good results on the retinal epithelial dataset and the simulated datasets. Moreover, FocusALL can generate good in-focus images to deal with focal stacking on high resolution images in a reasonable time ( $< 10$  seconds) while some methods generate results in minutes. Our proposed FocusALL-VI can also deal with images of varying illumination. For varying illumination images, although FocusALL generates sharp object regions, it may introduce artificial boundaries in the output image. In future work, we plan to identify these boundaries in the image and propose solutions for removing these artificial boundaries. Moreover, we plan to evaluate our FocusALL technique on other domains such as other microscopic images (e.g., cellular images) and macro-photography images.

## ACKNOWLEDGEMENT

This research was supported by National Institutes of Health (GM-090453) grant.

## REFERENCES

- [1] M. Sigdel, M.L. Pusey, and R.S. Aygün. Real-time protein crystallization image acquisition and classification system. *Crystal growth & design*, 13(7):2728–2736, 2013.
- [2] M.S. Sigdel, M. Sigdel, S. Dinç, İ. Dinç, M.L. Pusey, and R.S. Aygün. Autofocusing of microscopic images using harris corner response measure. In *IEEE southEastConf*, pages 1–6. IEEE, 2014.
- [3] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [4] N.L. Stauffer. Active auto focus system improvement, January 4 1983. US Patent 4,367,027.
- [5] V.V. Bezzubik, S.N. Ustinov, and N.R. Belashenkov. Optimization of algorithms for autofocusing a digital microscope. *Journal of Optical Technology*, 76(10):603–608, 2009.
- [6] D. Vollath. The influence of the scene parameters and of noise on the behaviour of automatic focusing algorithms. *Journal of microscopy*, 151(2):133–146, 1988.
- [7] M.G. Forero, F. Sroubek, and G. Cristóbal. Identification of tuberculosis bacteria based on shape and color. *Real-time imaging*, 10(4):251–262, 2004.
- [8] A. Junior, M. Costa, Cicero F.F. Costa F., L. Fujimoto, and J. Salem. Evaluation of autofocus functions of conventional sputum smear microscopy for tuberculosis [c]. In *IEEE International Conference on Engineering in Medicine and Biology Society (EMBS)*, pages 3041–3044, 2010.
- [9] R. Redondo et al. Autofocus evaluation for brightfield microscopy pathology. *Journal of biomedical optics*, 17(3):0360081–0360088, 2012.

- [10] Y. Sun, S. Duthaler, and B. Nelson. Autofocusing in computer microscopy: selecting the optimal focus algorithm. *Microscopy research and technique*, 65(3):139–149, 2004.
- [11] X.Y. Liu, W.H. Wang, and Y. Sun. Dynamic evaluation of autofocusing for automated microscopic analysis of blood smear and pap smear. *Journal of microscopy*, 227(1):15–23, 2007.
- [12] O.A. Osibote, R. Dendere, S. Krishnan, and T.S. Douglas. Automated focusing in bright-field microscopy for tuberculosis detection. *Journal of microscopy*, 240(2):155–163, 2010.
- [13] J. Mateos-Pérez et al. Comparative evaluation of autofocus algorithms for a real-time system for automatic detection of mycobacterium tuberculosis. *Cytometry Part A*, 81(3):213–221, 2012.
- [14] A. Valdecasas, D. Marshall, J. Becerra, and J. Terrero. On the extended depth of focus algorithms for bright field microscopy. *Micron*, 32(6):559–569, 2001.
- [15] B. Forster, D. Van De Ville, J. Berent, D. Sage, and M. Unser. Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images. *Microscopy Research and technique*, 65(1-2):33–42, 2004.
- [16] R. Dendere, O. Osibote, S. Krishnan, and T. Douglas. Image fusion for autofocusing in fluorescence microscopy for tuberculosis screening. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1383–1386. IEEE, 2011.
- [17] A.S. Sugimoto and Y. Ichioka. Digital composition of images with increased depth of focus considering depth information. *Applied optics*, 24(14):2076–2080, 1985.
- [18] N.T. Goldsmith. Deep focus; a digital image processing technique to produce improved focal depth in light microscopy. *Image Anal Stereol*, 19:163–167, 2000.
- [19] P. Hill, C. Canagarajah, and D. Bull. Image fusion using complex wavelets. In *BMVC*, pages 1–10. Citeseer, 2002.
- [20] J. Lewis, R. O’Callaghan, S. Nikolov, D. Bull, and N. Canagarajah. Pixel-and region-based image fusion with complex wavelets. *Information fusion*, 8(2):119–130, 2007.
- [21] W. Shi, C. Zhu, Y. Tian, and J. Nichol. Wavelet-based image fusion and quality assessment. *International Journal of Applied Earth Observation and Geoinformation*, 6(3):241–251, 2005.
- [22] S. Li, B. Yang, and J. Hu. Performance comparison of different multi-resolution transforms for image fusion. *Information Fusion*, 12(2):74–84, 2011.
- [23] S. Li and B. Yang. Multifocus image fusion by combining curvelet and wavelet transform. *Pattern Recognition Letters*, 29(9):1295–1301, 2008.
- [24] H.P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.
- [25] J.A. Hartigan and M.A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.



**Madhu S. Sigdel** received his Bachelor’s Degree in Computer Engineering from Institute of Engineering, Kathmandu, Nepal in 2011, and a M.S. degree in Computer Science from the University of Alabama in Huntsville in 2014. His M.S. thesis is related to focal stacking and autofocusing of microscopic images. His research interests include image processing, computer vision and data

mining.



**Madhav Sigdel** received the Bachelor’s Degree in Computer Engineering from Pulchowk Campus, Kathmandu, Nepal in 2008, and the M.S. and Ph.D. degree in Computer Science from the University of Alabama in Huntsville in 2012 and 2015, respectively. His Ph.D. thesis is related to real-time protein crystallization image analysis. His research interests include data mining, computer vision, multimedia systems, and information visualization.



**Semih Dinc** is a Ph.D. student at Computer Science Department in University of Alabama in Huntsville since 2012. He received his B.S. degree in Computer Science at Dokuz Eylul University in 2004, and his

M.S. degree in Control Engineering at Yildiz Technical University in 2012. His research areas are computer vision, image processing, and pattern recognition. He is recently studying vision based trajectory tracking system for mobile robots.



mining, image processing, and pattern recognition.

**Imren Dinc** is a Ph.D. student at Computer Science Department, University of Alabama in Huntsville, and working as a GRA since 2013. She received her B.S. degree in computer engineering from Dokuz Eylul University, Turkey in 2012 with first rank in the department and an honor degree. She is currently studying experimental design for protein crystallization experiments and visualization. Her research areas are data



of his research being improved methods for protein crystal screening, crystal detection, and crystallization condition identification using visible fluorescence methods.

**Marc L. Pusey** is a research scientist working at iXpressGenes, Inc., Huntsville Alabama. Dr. Pusey received his Ph.D. in Biochemistry from the University of Miami, Fl., then did post-doctoral research at the University of Minnesota. He moved to Huntsville in 1985, and worked at NASA/MSFC for the next 23 years in the new field of protein crystal growth. After retirement from NASA, he obtained his current position at iXpressGenes with the focus



interests include protein crystallization image analysis, data mining, image and video processing, spatio-temporal indexing and querying, multimedia databases, semantic computing, multimedia networking, and multimedia synchronization.

**Ramazan S Aygün**: received the B.S. degree in computer engineering from Bilkent University, Ankara, Turkey in 1996, the M.S. degree from Middle East Technical University, Ankara in 1998, and the Ph.D. degree in computer science and engineering from State University of New York at Buffalo in 2003. He is currently an Associate Professor in Computer Science Department, University of Alabama in Huntsville. His research