SHAPE FROM FOCUS USING OPTIMIZATION TECHNIQUE

Muhammad Bilal Ahmad and Tae-Sun Choi, SeniorMember IEEE

Gwangju Institute of Science and Technology, Gwangju, Korea. Email: bilal@gist.ac.kr

ABSTRACT

Three dimensional (3D) shape recovery using shape from focus (SFF) is presented as an optimization problem, i.e., maximization of the focus measure in the 3D image volume. The whole image volume (sequence) is divided into a number of sub image volumes. The search of 3D shape is made in the sub image volumes using dynamic programming optimization technique. The final depth map is obtained by collecting the depth map of the sub volumes. The new algorithm has considerably decreased the computational complexity by searching the 3D shape in sub image volumes and has shown better results. New 3D focus measure operators are also introduced for higher accuracy at the cost of some computational costs.

1. INTRODUCTION

Three dimensional (3D) shape recoveries from images is very important area of research in machine vision. Shape from focus (SFF) [1] can be used to reconstruct 3D shape of objects from 2D images. SFF is a problem of reconstructing the depth of a scene changing actively the optics of the camera until the point of interest is in focus. The basic idea of image focus is that the objects at different distances from a lens are focused at different distances. The change in the optics is obtained either by changing the lens position or the object position relative to the camera.

In this paper, the recovery of 3D shape from image sequence is presented as an optimization problem, i.e., maximization of the focus measure in the 3D image volume. The whole image sequence (3D image volume) is divided into a number of sub image sequences (3D sub image volumes). Dynamic programming optimization technique is used to search the 3D shape in the sub image volumes. A direct application of dynamic programming on a 3D data is impractical, because of exponential computational complexity. Therefore a fast heuristic model based on dynamic programming [2] is used. The new algorithm has considerably decreased the computational complexity by searching the 3D shape in sub image volumes and shows better 3D shape recovery results.

In the literature, SFF applies focus measures in small 2D windows [3] on image sequences. In this paper, we have also shown that the focused measures can be applied in small 3D windows for better results.

2. SFF USING OPTIMIZATION TECHNIQUE (SFFOPT)

Focus measurement is not a point operation [4]. It must be calculated over a small patch implicitly assuming that the depth of the scene is constant (or moderately changing) within the patch. The state of focus is detected by comparison of focus ("sharpness") measurements in the same patch over several focus settings. To have correct depth estimation, the focus measure in the patch should be largest in the focused state. In traditional SFF method (SFFTR) [1], 2D patches (or windows) were used for maximizing the focus measure, while in SFFFIS [5], 3D patches were used. The SFFFIS showed better results by using 3D patches, but at the cost of considerable computations. The proposed algorithm SFFOPT also uses 3D patches, and maximizes the focus measure using fast heuristic approach similar to dynamic programming optimization technique.

In SFFOPT, a sequence of images or an image volume $V_{i,x,y}$, is recorded by moving the image detector for $i = 1,2,..., I$; $x = 1,2,..., X$; and $y = 1,2,..., Y$. Where *I* is the number of image frames, and *X, Y* are the number of rows and columns in each image frame respectively. The problem is to search the set of pixels which lie on the focused image surface (FIS) of the object in the image volume. FIS of an object is defined as the surface formed from the focused points. The search of FIS is presented as an optimization problem i.e. maximizes the focus measure in the 3D image volume.

A focus measure operator is applied on each image frame in the input image sequence $V_{i,x,y}$, and focus measure image volume $O_{i,x,y}$ is obtained. $O_{i,x,y}$ is divided into sub volumes (sub-sequences). Each image frame from $O_{i,x,y}$ is divided into sub-images. The size of a sub-image is selected as 15 x 15. The rough depth map is determined only at the center pixel of each sub-image, by using one of the traditional SFF methods. The rough depth map tells the approximate image number I_o where the center pixel of the sub-image is focused. A small number of images, say 11, around I_o are selected and a sub image volume $S^{sub}_{p,m,n}$ centered at I_o is made as shown in Fig. 1, for $p = 1,2,...,11$; $m = 1,2,...,15$; and $n = 1,2,...,15$. The subscripts *p, m,* and *n* are indices for images, rows and columns respectively.

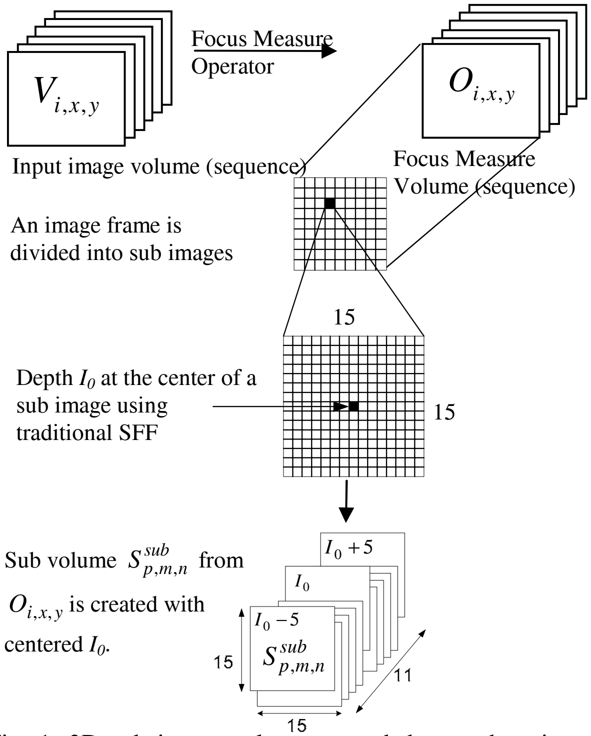


Fig. 1. 3D sub image volume around the rough estimate I_o at the center of sub-image.

The problem is now reduced to search the shape of FIS in the sub volume $S^{sub}_{p,m,n}$ that maximizes the focus measure. The search of the FIS shape in the sub volume is done using fast heuristic model based on dynamic programming optimization technique. The depth maps obtained from the sub volumes are then combined together to get the final 3D depth map.

The 3D sub volume $S^{sub}_{p,m,n}$ is divided into 2D matrices. The 2D matrices are constructed by slicing $S^{sub}_{p,m,n}$ along *x* and *y* axes. First, row i.e., *m* is kept constant, and y-slice matrices $A^m_{p,n}$ are made. The y-slice matrices are made from one row of each image frame of $S^{sub}_{p,m,n}$, where the row number is determined by the value of m. For example, $A^1_{p,n}$ is made from the first row of each image frame of $S^{sub}_{p,m,n}$. Second, column i.e., *n* is kept constant, and x-slice matrices $B^m_{p,n}$ are made. The x-slice matrices are made from one column of each image frame of $S^{sub}_{p,m,n}$, where the column number is determined by the value of *n*.

After constructing 2D matrices from 3D image volume, new matrices "Right_Sum" and "Left_Sum" are defined for each of the 2D matrices using the recurrence formulation. The recurrence formulation for one matrix is explained as follows. The same formulation can be easily applied for all matrices. Consider a y-slice matrix $A^1_{p,n}$. The Right_Sum $R^{sum}_{p,n}$ for the matrix $A^1_{p,n}$ is defined as:

$$R^{sum}_{p,n} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n-1} & r_{1,n} & \cdots & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n-1} & r_{2,n} & \cdots & \cdots & r_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & r_{p-1,n-1} & \vdots & \cdots & \vdots & \vdots \\ r_{p,1} & r_{p,2} & \cdots & r_{p,n-1} & r_{p,n} & \cdots & \cdots & r_{p,N} \\ \vdots & \vdots & \cdots & r_{p+1,n-1} & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ r_{P-1,1} & r_{P-1,2} & \cdots & r_{P-1,n-1} & r_{P-1,n} & \cdots & \cdots & r_{P-1,N} \\ r_{P,1} & r_{P,2} & \cdots & r_{P,n-1} & r_{P,n} & \cdots & \cdots & r_{P,N} \end{bmatrix} \quad (1)$$

where the recursive formulae involved in the calculation of the Right_Sum $R^{sum}_{p,n}$ are given as:

For first column of $R^{sum}_{p,n}$; $r_{p,1} = a_{p,1}$

Generally, $r_{p,n} = a_{P,n} + \max\{r_{p-1,n-1}, r_{p,n-1}, r_{p+1,n-1}\}$

For first row of $R^{sum}_{p,n}$; $r_{1,n} = a_{1,n} + \max\{r_{1,n-1}, r_{2,n-1}\}$ and

For last row of $R^{sum}_{p,n}$; $r_{P,n} = a_{P,n} + \max\{r_{P-1,n-1}, r_{P,n-1}\}$

In general, the *(p,n)th* element of matrix Right_Sum $R^{sum}_{p,n}$ is defined as the sum of Focus Measure value at *(p,n)th* element from the matrix $A^1_{p,n}$, and the maximum of the three previous Right_Sum $R^{sum}_{p,n}$ elements at the left column.

Similarly the Left_Sum $L^{sum}_{p,n}$ for the matrix $A^1_{p,n}$ is defined. It should be noted that the matrix Left_Sum $L^{sum}_{p,n}$ is filled from the right side, i.e., the last column of $L^{sum}_{p,n}$ is filled first and the first column is filled at the end. Again, the *(p,n)th* element of matrix Left_Sum $L^{sum}_{p,n}$ is defined as the sum of Focus Measure value at *(p,n)th* element from the matrix $A^1_{p,n}$, and the maximum of the three previous Left_Sum $L^{sum}_{p,n}$ elements at the right column.

A new matrix "Y_Total_Sum" $Y^{Tsum}_{p,n}$ for the y-slice matrix $A^{1}_{p,n}$ is defined as:

$$Y^{Tsum}_{p,n} = R^{sum}_{p,n} + L^{sum}_{p,n} - A^{1}_{p,n} \,, \qquad (2)$$

The Y_Total_Sum $Y^{Tsum}_{p,n}$ matrices are calculated for all rows (m = 1,2,…,M, where M = 15) in $S^{sub}_{p,m,n}$ using (2) and 3D volume $Y^{Tsum}_{p,m,n}$ is made.

Similarly, the "X_Total_Sum" $X^{Tsum}_{p,m,n}$ 3D volume for x-slice matrices are calculated using the same procedure as done for y-slice matrices. The X_Total_Sum $X^{Tsum}_{p,m,n}$ for x-slice matrices can be simply obtained by changing the matrices $A^{m}_{p,n}$ with matrices $B^{n}_{p,m}$ and switch *n* with *m* in the above mentioned procedure for y-slice matrices. The only difference is that now *n* (columns) is kept constant instead of *m* (rows).

The "Total_Sum" $S^{Tsum}_{p,m,n}$ for the sub volume $S^{sub}_{p,m,n}$ is determined as:

$$S^{Tsum}_{p,m,n} = Y^{Tsum}_{p,m,n} + X^{Tsum}_{p,m,n} \qquad (3)$$

The focus map $F^{sub}_{m,n}$ of the sub volume $S^{sub}_{p,m,n}$ is the image frame among the sub volume (sub image sequence) that gives maximum value of Total_Sum $S^{Tsum}_{p,m,n}$ along the *p* (image) axis and is expressed as:

$$F^{sub}_{m,n} = \arg \max_{p=1,\dots,P} S^{Tsum}_{p,m,n} \qquad (4)$$

Equation (4) returns for each pixel the index position p = 1,2,…,P=11 of the image with the largest Total_Sum value at that pixel position. The sub focus map $F^{sub}_{m,n}$ contains the image number for the best focused points of the sub volume $S^{sub}_{p,m,n}$. For getting the absolute image number corresponding to the best focused points, the initial rough estimate image number at the center of the sub image volume is added to the focus map $F^{sub}_{m,n}$.

The focus maps $F^{sub}_{m,n}$ are calculated for all sub volumes created by the sub-images. The focus maps from the sub volumes are then integrated to get the final focus map or 3D shape of the object. If we take the gray-level value of the pixel at (x,y) of the image number obtained from the final focus map, we can construct the image which is focused at all pixels. For the known values of camera setting, we can also find the distance of each pixel from the camera using the lens formula.

2.1. 3D Focus Measure

In the literature, SFF computes focus measures at each pixel in the image frame using a small 2D window around the pixel [3]. At each pixel, the image frame among the image sequence that gives a maximum sharpness measure is determined. This corresponds to a piecewise constant approximation of the object shape in the window. Because of this approximation, the focused image reconstructed from the image sequence is an approximation of the actual focused image. The focused image reconstruction can be improved if 3D focus measure operators are used. This will correspond to planar surface approximation, one higher degree than the piecewise approximation. As we have sequence of images with small step distance among the frames, 3D focus measure operators can be easily defined. The three axes of the 3D focus operator are: rows, columns and image frames. 3D focus measure version of 2D Laplacian can be defined as:

$$\nabla^2 V_{i,x,y} = \frac{\partial^2 V}{\partial i^2} + \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} \qquad (5)$$

A discrete version of 3D Laplacian (3x3x3) is defined as:

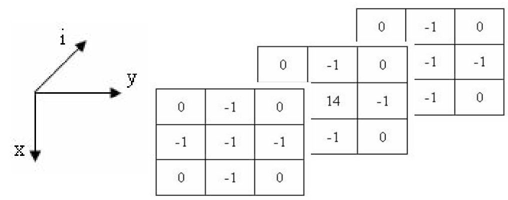


Fig. 2. 3D Laplacian Focus Measure Operator

Similarly, we can define 3D focus measure for others 2D focus measures in the literature by adding the third axis, i.e., the image axis.

3. SIMULATION RESULTS

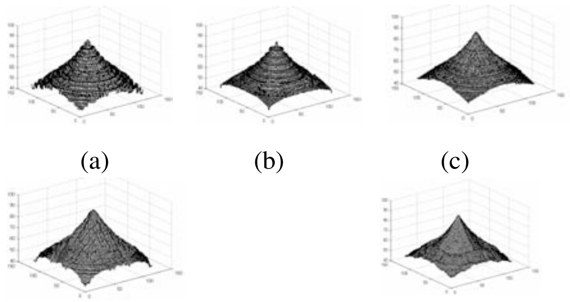
Figure 3 shows one image from each test sequence where only one part of the image is focused, whereas the other parts are blurred to varying degrees. The 3D shapes or depth maps recovered by SFF methods are shown in Fig. 4-6. If we take the full image volume as a small volume in the proposed algorithm, we get global SFFOPT [2]. The local SFFOPT can be obtained by dividing the whole image sequence into a number of small volumes as we have explained in the proposed algorithm.

(a) (b) (c)

Fig. 3. Images from the test sequences (a) Simulated cone (b) Real cone (c) Microscopic object.

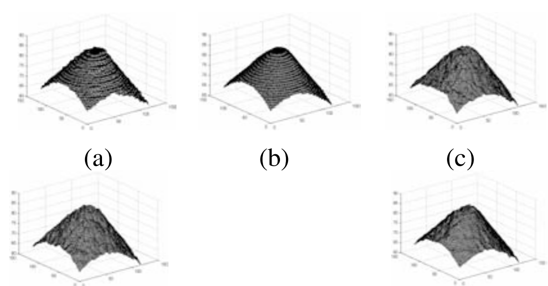
The ideal depth map for the simulated cone should be very smooth and the tip should be very sharp. From Figs. 4 and 5, we can see that the depth maps obtained from

SFFTR on the simulated and real cones are not smooth. The depth maps seem to change in large jumps instead of varying gradually, and the tips of the cones are not very sharp. The SFFFIS shows better depth maps. The depth map obtained from global SFFOPT on the simulated cone is very smooth and the tip is very sharp. The local SFFOPT also shows comparable results with those of SFFFIS and global SFFOPT. Local SFFOPT with 3D focus measure operator shows better results comparing smoothness of the surface and sharpness of tips of the cones.



(a) (b) (c) (d) (e)

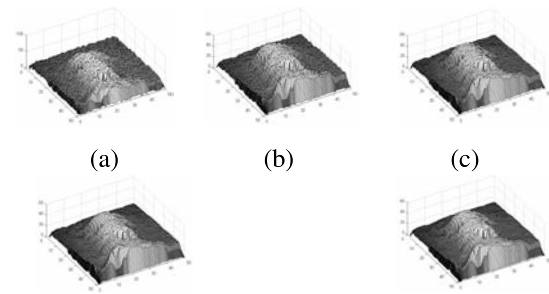
Fig. 4. The reconstructed 3D depth map for the Simulated cone object by (a) SFFTR (b) SFFFIS (c) Global SFFOPT (d) Local SFFOPT and (e) Local SFFOPT with 3D focus measure.



(a) (b) (c) (d) (e)

Fig. 5. The reconstructed 3D depth map for the Real cone object by (a) SFFTR (b) SFFFIS (c) Global SFFOPT (d) Local SFFOPT and (e) Local SFFOPT with 3D focus measure.

The computer simulation was carried out on 2.8 GHz P-IV PC using Visual C++. The depth estimation time of different algorithms are shown in Table 1 for a sequence of 97 images, and the size of each image frame being 256 x 256 pixels. The number of arithmetic operations reduces considerably in local SFFOPT as compared to global SFFOPT because local SFFOPT uses small cubic volumes for the search of FIS shape. The global SFFOPT algorithm is much faster than the SFFFIS, but slower than the SFFTR. However, the local SFFOPT is so fast that it executes almost in the same time as SFFTR. The local SFFOPT with 3D focus measure requires higher computations than local SFFOPT because of 3D focus measure.



(a) (b) (c) (d) (e)

Fig. 6. The reconstructed 3D depth map for the microscopic object by (a) SFFTR (b) SFFFIS (c) Global SFFOPT (d) Local SFFOPT and (e) Local SFFOPT with 3D focus measure

Table 1 Comparison of depth estimation time

	SFFTR	SFFFIS	Global SFFOPT	Local SFFOPT	Local SFFOPT with 3D FM
Depth estimation time	7 sec	4 min	18 sec	5 sec	7 sec

4. CONCLUSION

The search of 3D shape from image sequences is presented as an optimization problem, i.e., maximizes the focus measure in the input image sequence. The dynamic programming optimization technique is used to recover the 3D shape of the objects. The new proposed algorithms are more accurate and faster than previous algorithms in the literature.

5. REFERENCES

[1] Shree K. Nayar and Yasuo Nakagawa, "Shape from focus," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 8, pp. 824-831, Aug. 1994.

[2] Muhammad Bilal Ahmad and Tae-Sun Choi, "A heuristic approach for finding best focused shape," IEEE Transactions on Circuit System for Video Technologies, vol. 15, no. 4, pp. 566-574, Apr. 2005.

[3] E. Krotkov, "Focusing," International Journal of Computer Vision, vol. 1 pp. 223-237, Oct. 1987.

[4] Y.Y. Schechner, and N. Kiryati, "Depth from Defocus vs. Stereo: How Different Really are They?," International Journal of Computer Vision, vol. 39, no. 2, pp. 141-162, Sep. 2000.

[5] Murali Subbarao, and Tae-Sun Choi, "Accurate recovery of three dimensional shape from image focus," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 3, pp. 2824-2836, Mar. 1995.