Three Dimensional Shape Recovery from Blurred Images

Kyeongwan Roh, Choongwon Kim
Dept. of Computer Engineering, Chosun University
#375 Seoseok-dong, Dong-gu, Kwangju 501-759, Republic of KOREA
Tel / Fax: +82-62-230-7381, E-mail: brianroh2@hotmail.com

Gueesang Lee, Soohyung Kim
Dept. of Computer Science, Chonnam National University
#300 Yongbong-dong, Buk-gu, Kwangju 500-757, Republic of KOREA

Abstract: There are many methods that extract the depth information based on the blurring ratio for object point in DFD(Depth from Defocus). However, it is often difficult to measure the depth of the object in two-dimensional images that was affected by various elements such as edges, textures, and etc. To solve the problem, new DFD method employing the texture classification with a neural network is proposed. This method extracts the feature of texture from an evaluation window in an image and classifies the texture class. Finally, It allocates the correspondent value for the blurring ratio. The experimental result shows that the method gives more accurate than the previous methods.

1. Introduction

The scene image for three-dimensional environment consists of geometric and photometric information. The distance and shape of objects in a scene constitute the geometric information, while color and image irradiance of objects constitute the photometric information. There are two ways to measure the depth of an object using image focusing. The first method, called DFF(Depth From Focus), is to detect the most optimally focused object point using 10 or more images and then measures the distance from the camera to the focused object point. The second method, called DFD(Depth From Defocus), finds depth of the object using only two images with different optical settings. The relative defocus in the two images can, in principle, be used to determine three-dimensional structure. The distance from lens to object and the texture of the image have an effect on focus values and blur ratios respectively in DFF and DFD.

Many efforts have been made to estimate depth from the degree of blurring of the image[1,2,3]. In this paper, we use a Subbarao's mathematical theory[3] that the image is obtained by changing one or two camera parameters by small values. Most of previous works use a zoom lens. But the zoom lens gives rise to the changes of the distance between an object and a lens. And it is often difficult to measure the depth of the object in two-dimensional images that was affected by various elements such as edges, textures, and etc.

To solve the problem, new DFD method is proposed. That uses the mechanical devices and employs the texture classification with a neural network. The devices fix the distance and adjust a micrometer at the bellows that changed the only distance between from image plane to lens. The texture classification method extracts the feature of texture from an evaluation window in an image and classifies the texture class. And it allocates the correspondent value for the blurring ratio. Finally the depth of object is obtained. The experimental result shows that the method gives more accurate than the previous methods.

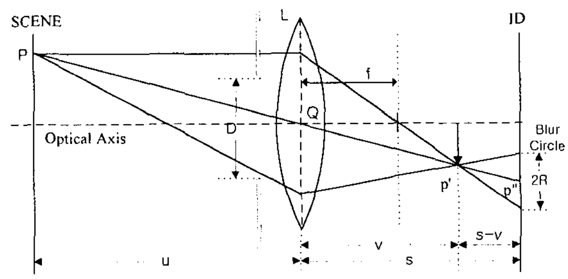
This paper presents a new method for combining DFD and neural network to classify the texture to improve the accuracy of depth estimation. Section 2 and 3 review theoretical background. Section 4 shows implementation and experiments. Finally Section 5 presents conclusions.

2. Camera Model and PSF

The relation between the distances u and v of the point P and p', respectively, from the lens plane is given by the lens formula. So, a focused image is obtained under the condition that s equals v. The resulting image of point P is a circular blurred image p" on the detector plane and p' is the focused image of point P.

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{1}$$

The blur of an image is the response of the camera to a single point source. This response is called the point spread function (PSF). The PSF represents the precise structure of the blur circle.



(Q: Optical Center, ID: image detector plane,
p': Focused image, p": Blurred image,
R: Blur circle radius, P: Object point,
f: Focal length, D: Aperture diameter,
L: Lens, v: Distance from focused point to lens,
s: Distance from ID to lens, u: Object distance)

Fig. 1. Camera geometry in a convex lens.

From the geometry of the optics relation and the lens formula, the size of the blur circle is proportional to the size of the aperture. Then, the diameter of a blur circle is given by

$$R=\frac{D}{2}\left(\frac{s-v}{v}\right)=s\frac{D}{2}\left[\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right] \tag{2}$$

Taking diffraction, the variation of the wavelength of light in the image and other non-idealities of the lens into account, the blur circle is better described by the Gaussian function, h(x, y).

$$h(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}},\quad H(w,v)=e^{-\frac{x^2+y^2}{2}\sigma^2} \tag{3}$$

where σ is the spread parameter such as

$$\sigma=kR=k\frac{D}{2}s\left(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right) \tag{4}$$

$$d\sigma=k\frac{D}{2}\left(\frac{1}{f}-\frac{1}{u}\right)ds \tag{5}$$

Let g(x, y) be the blurred image of an object on the CCD plane, and f(x, y) be the corresponding focused image. For a planar object perpendicular to the optical axis, the camera acts as a linear shift invariant system. Therefore g(x, y) will be equal to the convolution of focused image f(x, y) with the PSF h(x, y). Convolution in the spatial domain corresponds to multiplication in the Fourier domain. Let G(w, v) and F(w, v) be the corresponding Fourier transforms.

$$G(w, v) = H(w, v)\ F(w, v) \tag{6}$$

And, the power spectral density for a Gaussian point spread function is

$$P(w,v)=e^{-(w^2+v^2)\sigma^2}\,FF^* \tag{7}$$

This shows that the blurring caused by defocusing is equivalent to an exponential decay function in frequency domain. Thus, the degree of defocusing is estimated by the difference of the focused image and the blurred image in the frequency domain.

3. DFD and Texture

Most methods for DFD carry out the depth perception based on the geometric and photometric information, which is usually obtained through time-frequency representations such as Fourier transforms. They assume the image is stationary. But two-dimensional images are non-stationary with edges, textures, and deterministic objects at different location. Hence, we present a new DFD method employing texture classification by a neural network.

3.1 DFD using band pass filtering

The calculation of Fourier transforms at each image point is too expensive in practice. In order to use equation (12), we need only the Fourier power. We can then make use of Parseval's theorem, which states that the integral of the squared values over the spatial domain is equal to the integral of the squared Fourier components over the frequency domain[2]. By doing convolution with a band pass filter, we get a signal restricted to a limited range of frequencies. However, the approximation by band pass filtering described above doesn't give exact values of C. Because the Laplacian filter has a wide band in frequency as a band pass filter, the value of w^2+v^2 is a little varied according to the frequency distribution of the image[4]. Thus, the value of C is dependent of the texture.

$$f_1'=f_1*G,\qquad f_2'=f_2*G \tag{8}$$

$$f_1''=f_1'*L,\qquad f_2''=f_2'*L \tag{9}$$

Two images f_1 and f_2 are acquired when the distance between the image detector plane and the center of lens are s and $s+\sigma$ respectively. In order to remove noise, Gaussian convolution operation is employed on both the images f_1', f_2'. After the Gaussian smoothing, a band-pass filtering by a Laplacian kernel is applied to both the images, and then the square of the pixel values after band-pass filtering are summed up to the Laplacian power (P_1, P_2).

$$P_1=\sum_x\sum_y(f_1'')^2,\quad P_2=\sum_x\sum_y(f_2'')^2 \tag{10}$$

dP/P is calculated as

$$\frac{dP}{P}=\frac{2(P_2-P_1)}{P_1+P_2} \tag{11}$$

We can therefore calculate the circle of confusion, $\sigma d\sigma$. Let $\sigma d\sigma$ is C,

$$C=\sigma\, d\sigma=-\frac{1}{2}\frac{1}{w^2+v^2}\frac{dP}{P} \tag{12}$$

$$C=k^2\frac{D^2}{4}s\left(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right)\left(\frac{1}{f}-\frac{1}{u}\right)ds \tag{13}$$

if $T=dP/P$,

$$C=-\frac{1}{2}\frac{1}{w^2+v^2}\frac{dP}{P}=-\frac{1}{2}\frac{1}{w^2+v^2}T \tag{14}$$

From equation (12) and (13), replacing $K=k^2D^2ds\,s(w^2+v^2)$. Since $f<u<\infty$ in normal experimental settings, the solution becomes unique. We can finally obtain the depth of the object is

$$u=\frac{1}{\frac{1}{f}-\frac{K+\sqrt{K^2+8KTs^2}}{2Ks}} \tag{15}$$

3.2 Texture Classification

The degree of defocus is dependent of the texture. Therefore, we classified the texture for depth extraction and used a texture feature extraction scheme based on a Gabor decomposition. The Gabor features provide excellent pattern retrieval performance.

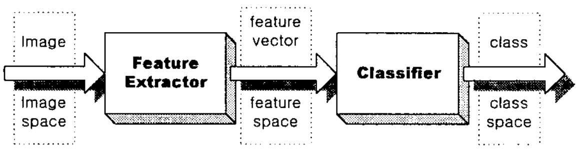


Fig. 2. Feature extraction and classification for texture

The prototype of Gabor filter for the texture feature extraction is shown below[5].

$$G(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right) \tag{16}$$

A bank of Gabor filters can be generated by dilating and rotating the above function:

$$G_{i,j}(x,y)=a^{-m}G(x',y'),\qquad a>1,\quad m,n=\text{integer} \tag{17}$$

$$x'=a^{-m}(x\cos\theta+y\sin\theta),\quad y'=a^{-m}(-x\cos\theta+y\sin\theta)$$

where $\theta=j\pi/K$ and K is the total number of orientations. The scale factor a^{-m} is meant to ensure the equal energy among different filters. These Gabor filters can be considered as orientation and scale tunable edge and line detectors. The statistics of the detected features can be used to characterize the underlying texture information. When an Image I(x,y) is given, a Gabor decomposition can be obtained by

$$W_{i,j}(x,y) = \iint I(x,y)\, G^{*}_{i,j}(x-x_1, y-y_1)\, dx_1 dy_1 \tag{18}$$

where * means the complex conjugate. Because $W_{i,j}(x,y)$ is a complex number, it can be further represented as $W_{i,j}(x,y) = m_{i,j}(x,y)\exp[\phi_{i,j}(x,y)]$. A simple texture feature representation (f_t) can be constructed using the mean and standard deviation of the amplitude information:

$$u_{ij} = \iint W_{i,j}(x,y)\, dxdy, \quad \sigma_{ij} = \sqrt{\iint (m_{i,j}(x,y) - u_{ij})^2 dxdy} \tag{19}$$

$$f_t = [u_{00}, \sigma_{00}, u_{01}, \cdots, u_{(S-1)(K-1)}, \sigma_{(S-1)(K-1)}] \tag{20}$$

In this experiment, we used four different scales, S=4, and six orientations, K=6. This results in a feature vector of 48 dimensions. These feature vector make 48 input nodes of the input layer of a neural network.

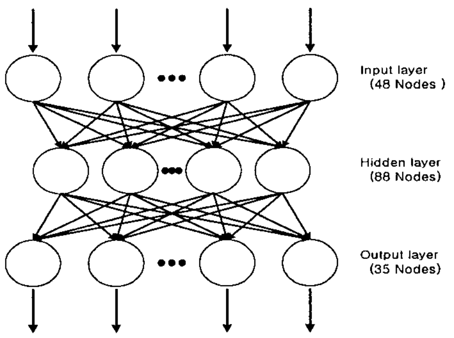


Fig. 3. Neural network for Classification

A Multi-layer perceptron can be used to approximate the texture classification. We use a three-layered neural network model in Figure 3. The input layer has 48 feature vectors, the hidden layer has 88 nodes, and output layer has 35 nodes. The learning data has 600 feature sets that consist of 30 textures with 20 varied focus distances.

In order to accelerate the learning process, we introduce a moment method. The values of moment constant and learning rate parameter are varied throughout the experiments, typical values being 0.1 and 0.5, respectively. As a test method for the classification, we use a 'leave-one-out' cross-validation that each time leaves one data out for testing and the rest for training [6]. Here the error rate was 0.7%.

More learning data is necessary for this system to get effective depth information. But, it can't learn the all the texture. Therefore, it needs the remarkable texture as many as possible. Then, this system extracts the depth information using the feature of the texture that is very similar to the object.

4. Implementation and Experiments

In order to experiment on DFD algorithm by using texture classification, we used an inclined planar object with the following texture shown in Figure 4. Figure 5 and 6 show the relation between the depth and the blur ratio according to the five textures.

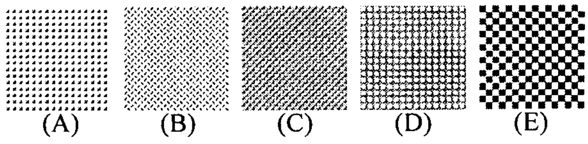


Fig. 4. The texture used for experiments

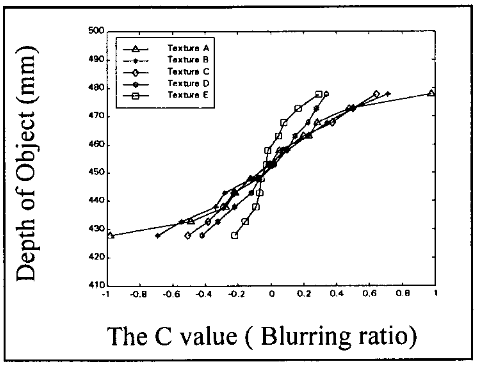


Fig. 5. The relations between the C value and the depth

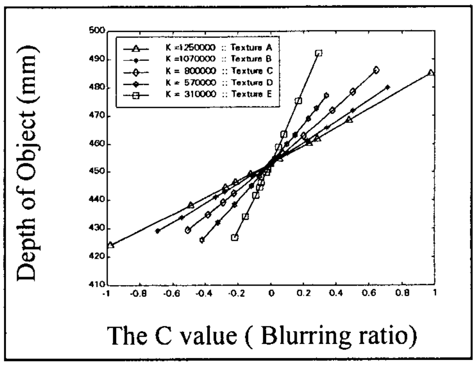


Fig. 6. The fitted K value according to texture

The image size is 448*448 pixels and the evaluation windows size is 64*64 pixels. In a texture leaning stage, an image is acquired at every distance and then the texture feature is extracted from each evaluation window. Using the equation (11) and (15), insert the K value into the reference table. In this case, the K is fitted as a constant using a least square fitting because K is varied from depth and texture (see figure 5, 6). Figure 5 and 6 described how the depth map about the texture is uniform and how well the result is related to the real distance of the object. In a test stage, after the acquiring an image, an image is divided with 49 windowed images. Then Gabor filter extracts the feature of the evaluating window and the extracted features were sent to the neural network that classifies the texture and allocates the K value according to the texture class. Given K value from reference table, dP/P from equation (11), and equation (15), we can compute the depth of object.

In Figure 8 and 9, X and Y axis show the evaluated window for an image, and Z axis shows the range of depth information. Figure 7 shows the experimental condition used for the experiment. In this condition, the inclined planar object has two different textures. Figure 8 shows a depth map generated without the texture classification. As shown in Figure 9, it shows a depth map generated using the texture classification with a neural network.

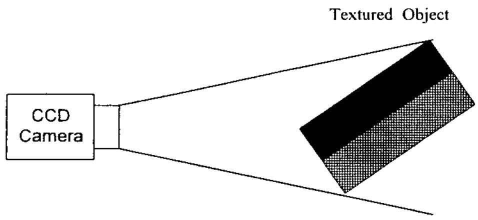


Fig. 7. The geometric settings of the experiment

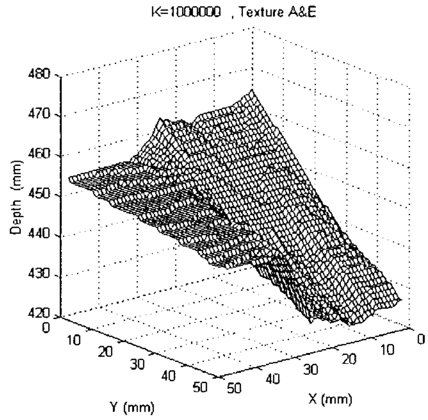


Fig. 8. 3D reconstruction of inclined object with two textures (K=1000000)

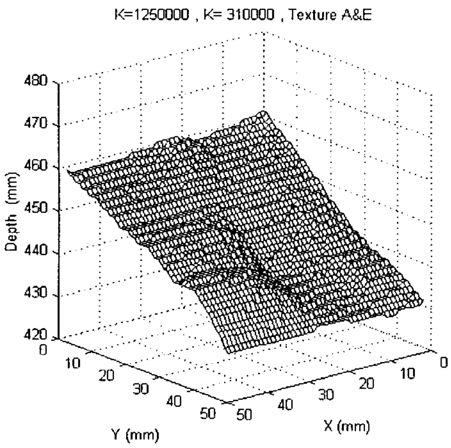


Fig. 9. 3D reconstruction of inclined object with two textures (K=1250000, K=310000)

The upper edge of squared box means the depth= 460mm and the lower edge means the depth= 430mm. The depth map obtained by the proposed method is more uniform and smoother than that of existing method. This means that since the existing method's response of the blur ratio varies with texture frequency, a single broadband Laplacian filter that produces an aggregate estimate of defocus for an unknown texture and is inevitably sensitive to the frequency spectra of local scene textures. The result of root mean square (RMS) error in this experiment shows 2.61% that is lower than that of existing method (4.57%)[7]. Therefore, we could obtain a satisfactory result.

In our method, there are two constraints. First, the window selected for blurring ratio analysis must contain information at a single depth. If the window has multiple peaks, the analysis of the blurring ratio fails. Second, the range of depth is dependent on the length of bellows. Short bellows are useful to general scene while long bellows generates a microscope image.

This work was implemented using Matrox Meteor frame grabber, Pulnix TMS-7AS CCD Camera, Nikon 50mm lens, Bellows with micrometers, Z-stage, and LED ring illumination.

5. Conclusions

In this paper, texture classifier with neural network for acquiring a high accuracy is presented in DFD. This method measures the more accurate depth information in an image affected by various textures. The texture classifier with a neural network extracts the feature of texture from an evaluation window in an image and classifies the texture class. And it allocates the correspondent value for the blurring ratio. When an unregistered texture is used, this method substitutes it by the much similar texture. Finally the depth of object is obtained. The experimental results have 2.61% RMS error that indicates the accuracy of this method is satisfactory. Therefore, the DFD with the texture classifier is a powerful method for depth estimation at image focusing researches in computer vision.

Acknowledgements

This work was supported by the Brain Korea 21 project.

References

[1] A. P. Pentland, "A New Sense for Depth of Fiend," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, pp. 523-531, July 1987.

[2] A. P. Pentland, T. Darrel, M. Turk, and W. Huang, "A Simple Real-Time Range Camera", Proc. the IEEE CVPR, pp. 256-261. 1989.

[3] M. Subbarao, "Parallel Depth Recovery by Changing Camera Parameters," Proc. IEEE Sec. Intl. Conf. Computer Vision, pp. 58-65, Dec. 1987.

[4] A. Horii, "Depth from Defocusing", ISRN KTH/NA/P-92/16-SE, June 1992.

[5] B. S. Manjunath and W. Y. Ma, "Texture Feature for Browsing and Retrieval of Image Data", IEEE Trans. on PAMI, Vol. 18, No. 8, pp. 837-842, Aug. 1996.

[6] K. Fukunaga and D. M. Hummels, "Leave-one-Out Procedures for Nonparametric Error Estimates", IEEE PAMI, Vol. 11, No. 4, pp. 421-423, April, 1989.

[7] M. Subbarao, T. C. Wei, and G. Surya, "Focused Image Recovery from Two Defocused Images Recorded with Different Camera Settings", IEEE Trans. on Image Processing, Vol. 4, No. 12, pp. 1613-1628, Dec. 1995.