A moment-preserving approach for depth from defocus

D. M. Tsai and C. T. Lin
Machine Vision Lab.
Department of Industrial Engineering and Management
Yuan-Ze University, Chung-Li, Taiwan, R.O.C.
E-mail: jedmctsai@saturn.yzu.edu.tw

1. INTRODUCTION

Depth measurement is one of the most important tasks in computer vision for the applications of 3-D object recognition, scene interpretation and robotics. Various methods for depth measurement have been proposed [1]. Stereo vision [2, 3] is perhaps the most popular technique to obtain the depth image of a 3-D object. It generally uses two cameras to estimate stereo disparity and then recovers the 3-D structure of an object. The camera model of a stereo system involves a matching process between two images. This requires reliable extraction of features from the separate 2-D images and the matching of these features between images. Both of these tasks are non-trivial and can be computationally expensive.

In contrast to stereo vision, Pentland [4, 5] has proposed a depth-from-defocus (DFD) method to measure the depth information using a single camera so that the image-to-image correspondence process is not required. DFD methods are based on the fact that in the image formed by an optical system, objects at a particular distance from the lens will be focused, whereas objects at other distances will be blurred by varying degrees depending on their distances. As the distance between the imaged point and the surface of exact focus
increases, the imaged object becomes progressively more defocused. By measuring the amount of defocus (blur) of a point object in the observed image, the depth of the point object with respect to the lens can be recovered from the geometric optics.

The blur estimation algorithms generally determine the blur estimate from either the image’s power spectrum in the frequency domain, or from the image’s point spread function in the spatial domain [6]. Pentland [7] has proposed two methods to measure the amount of defocus. The first method requires only one image and is based on measuring the blur of edges which are step discontinuity in the focused image. The blurred edge is modeled as the result of convolving a focused image with a point spread function that is assumed to be a Gaussian distribution with spatial parameter \( \sigma \). The parameter \( \sigma \) is used as the measure of defocus, and has a one-to-one correspondence to the depth. The second method requires two images and is based on comparing the two images formed with different aperture diameter settings. A ratio of the Fourier powers between the two images is shown to be related to the amount of defocus.

Following Pentland’s second method, many blur estimation algorithms have been developed [6, 8, 9, 10, 11]. These algorithms generally require two or more images obtained by changing one of the three intrinsic camera parameters: 1) distance between the lens and the image detector plane, 2) focal length of the lens, and 3) diameter of the lens aperture (f-number). These involve relatively low mechanical movement of the camera and need specialized camera system whose parameter setting can be controlled precisely.
Lai et al. [12] have proposed a generalized algorithm that follows Pentland’s first method for estimating the spatial parameter $\sigma$ of a Gaussian point spread function. The spatial parameter $\sigma$ is decomposed into the horizontal and vertical components $\sigma_x$ and $\sigma_y$ so that the estimation of the edge orientation is not required. The horizontal and vertical intensities of an observed edge is assumed to be the convolution of the focused image and the Gaussians with spatial parameters $\sigma_x$ and $\sigma_y$, respectively. The blur estimation problem is then formulated as a nonlinear equation. The parameter $\sigma_x$ and $\sigma_y$ are evaluated using an iterative solution based upon Newton’s method in the vicinity of piecewise linear edges. Since no closed-form solution exists for their model, the nonlinear search procedure can be very time-consuming and the solution may get stuck in some local minimum.

In this paper, we use the moment-preserving principle, which gives closed-form solution and is computationally fast, to estimate the amount of defocus from a single image. The basic framework of our approach is as follows. The observed gray-level image is initially converted into a gradient image using the Sobel edge operator. For every edge point of interest in the gradient image, the proportion of the edge region $\mathbb{E}_p$ in a small neighborhood window centered at the edge point is then computed using the moment-preserving method. A focused edge will result in small value of $\mathbb{E}_p$, while a defocused edge will yield large value of $\mathbb{E}_p$. The proportion of blurred edge $\mathbb{E}_b$ is, therefore, used as the description of degradation of the point spread function for estimating the depth. In addition to the use of the depth formula derived from geometric optics for depth estimation, artificial neural networks (ANNs) are also proposed in this study to
compensate for the estimation error from the depth formula.

This paper is organized as follows: Section 2 overviews the geometry of the depth formula. Section 3 describes the moment-preserving procedure for estimating the proportion of blurred edge region $\square_3$ in the neighborhood window. The ANNs used for compensating for the estimation error are discussed in Section 4. Section 5 presents the experimental results including the effect of varying sizes of the neighborhood window on estimation errors, and the depth accuracy of the geometric depth formula and the ANNs. The paper is concluded in Section 6.

2. THE DEPTH FORMULA

For a convex-lens camera with a lens of focal length $F$, the relation between the position of a point in the scene and the position of its focused image is given by the well-known lens law

$$\frac{\square}{\nu} + \frac{\square}{\square} = \frac{\square}{\square}$$

(1)

where $\square$ is the distance of the point object from the lens and $\nu$ is the distance of the focused image from the lens.

Let $\square$ be a point object on a visible surface in the scene, and $\square$ and $\square'$ be its corresponding points in the focused image and the image detector plane, respectively. If $\square$ is not in focus then it gives rise to a circular image called the blur circle on the image detector plane (see Figure 1). Let the diameter of the blur circle be denoted by $\square_3$. Pentland [7] has
shown that the relationship between the depth \( D \) of a point object and the diameter \( b \) of the blur circle is given by

\[
D = \frac{b}{v - n - n_0} \quad \text{for} \quad v > v \quad (2.a)
\]

\[
D = \frac{b}{v - n + n_0} \quad \text{for} \quad v < v \quad (2.b)
\]

where \( v \) is the distance between the lens and the image detector plane, and \( n \) is the f-number (aperture) of the lens system. As the sensor displacement increases (i.e., \( v - v \)), the defocusing diameter \( b \) increases. Note that defocusing is observed for both positive and negative sensor displacement. If the image detector is behind the focused image (i.e., \( v > v \)), the depth \( D \) is evaluated by eq.(2.a). If the image detector is in front of the focused image (i.e., \( v < v \)), the depth \( D \) is then evaluated by eq. (2.b). For a given lens system, the parameters \( n \), \( v \) and \( n_0 \) can be considered as constants. Therefore, eq.(2) shows that the defocus \( b \) is an unique indicator of depth \( D \). The depth formula of eq.(2) can be rewritten in a condensed form [12] as follows:

\[
D = \frac{b}{\pm n_0} \quad (3)
\]

where \( b = |v - n| \), \( n_0 = b - n \), and \( n_0 \) and \( n_0 \) are constants with respect to a given camera setting. The depth formulation of eq.(3) can be used to simplify the calibration procedure.

3. MEASURE OF DEFOCUS
The depth formula of eq.(3) shows that there is a one-to-one correspondence between the diameter of blur circle $b_d$ and the object depth $D$. The blur size $b_d$ is generally assumed to be proportional to the spatial parameter $\sigma$ of the point spread function, i.e.,

$$b_d = k \cdot \sigma$$

where $k$ is assumed to be a constant for a given lens system [7, 11, 12, 13]. Quantitative measurement of defocus is difficult and requires accurate modeling of the point spread function. Unlike the conventional blur estimation algorithms that assume the point spread function is a Gaussian distribution with spatial parameter $\sigma$ and solve for the value of $\sigma$ in a complex way, we use a more straightforward approach to find the amount of defocus by the moment-preserving technique. The observed image is initially converted into a gradient image using the Sobel edge operator so that edge pixels have large gradient magnitude, and non-edge pixels have approximately zero gradient magnitude. For each edge point of interest, the proportion of the edge region $e_p$ (i.e., the region with high gradient magnitude) with respect to the neighborhood window in the gradient image is computed using the moment-preserving principle. A focused edge will result in small $e_p$, whereas a defocused edge will yield large $e_p$. $e_p$ increases as the distance between the imaged point and the surface of exact focus increases. Therefore, $e_p$ is a measure for the amount of defocus. The estimation procedure for the proportion of edge region $e_p$ in a small window is described in detail as follows.

Let $f_{xy}$ be the gray-level of a pixel at $(x, y)$ in the observed image. The gradient of $f_{xy}$ is given by

$$\nabla f_{xy} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$
where
\[
\begin{align*}
\mathcal{I}_x &= \sum_i \sum_j \mathcal{I}_{xj} + \mathcal{I}_{xj+} + \mathcal{I}_{xj-} + \mathcal{I}_{xj}\cdot \\
\mathcal{I}_y &= \sum_i \sum_j \mathcal{I}_{yj} + \mathcal{I}_{yj+} + \mathcal{I}_{yj-} + \mathcal{I}_{yj}\cdot 
\end{align*}
\]

The horizontal and vertical Sobel edge operators $\mathcal{I}_x$ and $\mathcal{I}_y$ are given in Figure 2. The magnitude of the gradient is defined by
\[
\mathcal{G}_{xy} = \mathcal{G} = \sqrt{\mathcal{G}_x^2 + \mathcal{G}_y^2}
\]

\[
\mathcal{G}_{xy} \text{ forms the gradient image of the observed image } \mathcal{F}_{xy} \text{ Figure 3(a) demonstrates the observed gray-level image of a multi-step block. The camera is focused on the lower steps of the block (lower-right in the image), and the upper steps are close to the lens and result in defocused image (upper-left in the image). Figure 3(b) presents the resulting gradient image of the observed image. It shows that the focused steps result in thin and sharp edges, and the defocused steps yield thick and scattering edges. The width of edges increases from lower-right to upper-left in the gradient image as the multi-step block is defocused progressively from lower steps to upper steps. The width of edges in the gradient image can be a description for the diameter of blur circle $\mathcal{B}_b$.}
\]

As observed in Figure 3(b), the gradient image can be divided into two regions, the bright region that represents the edges with high gradient magnitudes, and the dark region that represents the interior portions of objects or the background with low gradient magnitudes. Given a local neighborhood window centered at the edge point of interest, the gradient image defined in the window can be converted into a binary image that contains only white region (i.e., high gradient magnitude for edges) and black region (i.e., low gradient magnitude for backgrounds) using the moment preserving method. The proportion
of the white region with respect to the entire window region represents the width of the
imaged edge in the gradient image and, therefore, indicates the diameter of blur circle $d$. 

Let the gradient image defined in a local neighborhood window be the
real-world version of an ideal gradient image that consists of only two homogeneous regions,
the bright region with a uniform gradient magnitude $e$, and the dark region with a uniform
gradient magnitude $b$. Denote $e$ and $b$ by the proportions of the bright region and
the dark region, respectively, in the ideal gradient image. Note that $e > b$, 
$0 \leq e \leq 1$, 
and $e + b = 1$. For a given edge point at the first three moments of
are given by

$$\sum_{n=1}^{N_{xy}} \left[ \frac{\partial^j g_{xy}}{\partial x^j \partial y^j} \right]^n_{xy} = e^j b^j,$$

where $N_{xy}$ is the neighborhood window that consists of neighboring points around
and $n$ is the total number of pixels in the window.

By preserving the first three moments in both real-world gradient image and the ideal gradient image, we can obtain four equations as follows:

$$e^1 b^1 + e^2 b^2 = 1$$

and

$$e^0 b^0 = 1$$

There exists a closed-form solution for the four unknown variables $e$, $b$, $e$, and $b$. 

8
which are given by [14]

\[ q_b = \left[ - q_b - q_b^0 - q_b^0 \right] \]

\[ q_b = \left[ - q_b + q_b^0 - q_b^0 \right] \]

\[ q_b = q_e - q_b \]

where \[ q_b = \begin{bmatrix} - q_b^0 & q_b^0 & q_b^0 \\ - q_b^0 & q_0 \end{bmatrix} \]
\[ q_b = \begin{bmatrix} q_b^0 & q_b^0 & q_b^0 \\ q_0 & q_b^0 \end{bmatrix} \]

The value of \( q_b \), \( 0 \leq q_b \leq 1 \), gives the proportion of edge region in the neighborhood window. The larger value of \( q_b \), the larger amount of defocus. In this study, \( q_b \) is assumed to be proportional to the diameter of blur circle \( q_b \), i.e., \( q_b = q \cdot q_b \), where \( q \) is a constant. Therefore, the depth formula derived in eq.(3) can be rewritten as

\[ \Delta = \frac{\Delta}{\Delta' + \Delta_b} \]  \hspace{1cm} (4)

where \( \Delta = \Delta \cdot \delta \cdot \Delta, \Delta' = \Delta \cdot \delta \cdot \Delta \), and \( \Delta' \) and \( \Delta \) are constants for a given camera setting.

The constants \( \Delta' \) and \( \Delta \) in eq.(4) can be determined initially once and for all by a suitable camera calibration. We may manually collect \( n \) data points of the measured
depths $D_i$, at different distances from the camera, and use the moment-preserving method to calculate their corresponding proportions of edge region $e_i$ in the local window. Let $D_n^T = (D_1^T, D_2^T, ..., D_n^T)$ and $e_n^T = (e_1^T, e_2^T, ..., e_n^T)$. $(D, e)$ gives a set of $n$ known data pairs. Then, the best estimates of $P'$ and $Q'$, in the least-squares sense, are given by

$$
\begin{bmatrix}
P' \\
Q'
\end{bmatrix} = \left(\begin{bmatrix} P \end{bmatrix} - \begin{bmatrix} Q \end{bmatrix}\right) \left(\begin{bmatrix} P \\
Q
\end{bmatrix} - \begin{bmatrix} Q \end{bmatrix}\right)^{-1} \cdot \begin{bmatrix} P \end{bmatrix}.
$$

where $v = (D_1, D_2, D_3, ..., D_n, e_1, e_2, ..., e_n)$. Once $P'$ and $Q'$ are fixed for a given camera setting, the numerical relationship between the depth $D$ and $e$ is uniquely determined by eq.(4).

4. ANN APPROACH FOR ERROR COMPENSATION

Since the depth formula of eq.(3) arises from the geometric optics of lens imaging, the diameter of blur cycle $d_0$ only represents the geometric blur. However, the actual blur is not due to geometric defocus alone [15]. The geometric depth formula may yield nonlinear errors in calculating the depth $D$ owing to optical aberrations, vignetting, etc. To overcome this problem, we use artificial neural networks (ANNs) to compensate for the errors resulted from the depth formula. The advantages of an ANN in estimation applications are that it provides a model-free approach to reducing the estimation error, and it generates nonlinear interpolation for input data which are previously unseen in training.

An ANN is specified by the topology of the network, the characteristics of the
nodes and the processing algorithm. The neural networks used in this work are multilayer feedforward neural networks composed of an input layer, a single hidden layer, and an output layer. Each layer is fully connected to the succeeding layer. The outputs of nodes in one layer are transmitted to nodes in another layer through links. The link between nodes indicates flow of information during recall. During learning, information is also propagated back through the network and used to update connection weights between nodes.

Let \( y_j \) be the output of the previous layer and \( w_{ij} \) the connection weight between the \( i \)th node in one layer and \( j \)th node in the previous layer. The total input to the \( i \)th node of a layer is

\[
I_i = \sum_j w_{ij} o_j
\]

A hyperbolic tangent activation function is used here to determine the output of the node \( i \), which is given by

\[
o_i = \frac{e^{I_i} - e^{-I_i}}{e^{I_i} + e^{-I_i}}
\]

In the learning phase for such a network, we present the training pattern \( i = \{ i_p \} \), where \( i_p \) is the \( p \)th node in the input layer, and ask the network to adjust the weights in all the connecting links such that the desired outputs \( \{ d_j \} \) are obtained at the output nodes. Let \( \{ o_j \} \) be the evaluated outputs of the network in its current state. For a training pattern the squared error of the system can be written as

\[
E = \frac{1}{2} \sum_p (o_p - d_p)^2
\]

The generalized delta-rule learning algorithm [16] is applied to adjust the weights such that
the error $\Box$ is a minimum. A detailed derivation of the learning procedure can be found in [17].

Two neural networks are developed in this study. The first neural network, denoted by $\text{ANN}_1$, is a three-layer back-propagation network with two nodes in the input layer, seven nodes in the hidden layer, and one single node in the output layer. The topology of the network $\text{ANN}_1$ is illustrated in Figure 4. The input vector $\Box_1 = (\Box_1, \Box_2)$ of the network $\text{ANN}_1$ includes two components, which are

$\Box_1 =$ the proportion of edge region in the neighborhood window obtained from the moment-preserving method.

$\Box_2 =$ the depth of an edge point derived from the depth formula of eq.4.

$(\Box_1, \Box_2)$ correspond to the two nodes in the input layer in sequence. In the learning phase of the network, the desired value of the node in the output layer is the actual depth $\Box^*$ known a priori. A pair of (Input,Output) = $((\Box_1, \Box_2), \Box^*)$ forms the training sample for the network. In the recall phase of the network, the measured depth is simply given by the value of the node in the output layer.

It has been found [13] that the edge orientation is crucial to the estimation of the amount of defocus. A good strategy for improving the estimation accuracy of depth is to calibrate the constants $\Box'$ and $\Box'$ in eq.(4) using known data points in separate orientations, and then present the information of edge orientations to the network. The gradient $\nabla = f_{xy} g_{xy}$ used for computing the gradient magnitude as described in section 3 provides the additional information of edge orientation. The orientation of an edge point with gradient $\Box_1, \Box_2$ is given by
The value of $\Theta$ along with the signs of $\Box_1$ and $\Box_2$ can uniquely define the edge orientation between $\Box^\dagger$ and $\Box^\dagger_0$.

The proposed second neural network, denoted by ANN$_2$, therefore takes the edge orientation, and constants $\Box^\dagger$ and $\Box^\dagger_0$ calibrated in individual orientations as the additional input. The topology of the network ANN$_2$ is the same as that of the ANN$_1$, except that ANN$_2$ has five nodes in the input layer. The topology of the network ANN$_2$ is shown in Figure 5. The input vector $\Box_1 = \Box_1 \Box_2 \Theta \Box^\dagger_0 \Box^\dagger_0$ of the network ANN$_2$ consists of five components, which are

- $\Box_1$ = the same as those defined previously for the network ANN$_1$
- $\Theta$ = the edge orientation given by eq.(5)
- $\Box^\dagger_0$ = the constants in eq.(4) calibrated in the orientation of $\Theta$

In the training phase of the network ANN$_2$, pairs of $\Box_1 \Box^\dagger_0$ form the training samples with finite number of edge orientations. In the recall phase of the network, the edge orientation evaluated by eq.(5) is converted to the nearest orientation $\Theta$ used in training, and the corresponding $\Box^\dagger_0$ and $\Box^\dagger_0$ are selected from a look-up table. The value of the node in the output layer of the network gives the depth of the edge point.

5. EXPERIMENTAL RESULTS
In this section we present experimental results for evaluating the performance of the proposed depth estimators. In our implementations, all algorithms are programmed in the C language and executed on a personal computer with a Pentium 66 MHz processor. The image size is $512 \times 480$ pixels with 256 gray levels. The camera is set up so that the camera is 415 mm from the tabletop, and the optical axis of the camera is perpendicular to the table surface. All experiments are performed with the point of sharpest focus approximately set at the top of the table. A three-step block as shown in Figure 6 is used as the benchmark in the experiments to evaluate the performance of the proposed depth estimators. The first step (the one closest to the table), the second step and the third step (the one closest to the camera) are 21 mm, 40 mm and 40 mm in deep, respectively.

The first series of experiments use the three-step block to evaluate the effect of varying sizes of the neighborhood window on estimation errors of depth. The neighborhood window selected in this work is of circular shape. Figure 7(a) depicts the $p_e$ value versus the depth of each step of the block for the neighborhood windows of radii 45, 35, 25 and 19 pixels. It can be seen from the figure that the value of $p_e$ increased as the depth decreases, i.e., the amount of defocus increases as the object gets closer to the camera. The root-mean-squares (RMS) depth errors obtained by the depth formula for individual radii of the neighborhood windows are presented in Figure 7(b). It shows that too small the size of the window may not include sufficient data to estimate $p_e$ reliably, whereas too large the size of the window may include superfluous data and increases the computational requirement. Based on the experimental results, the neighborhood window of radius 35
pixels is valid for accurate estimation of $\quad_{p}$, and is used in the subsequent experiments.

The second series of experiments are to use the three-step block to evaluate the performances of the geometric depth formula and the neural networks $\text{ANN}_1$ and $\text{ANN}_2$. In order to analyze the effect of heights and orientations of objects with respect to a fixed camera, we have experimented the block placed at seven heights with respect to the tabletop varying from 0 mm to 60 mm in 10 mm increments. The block at each of the seven heights is rotated through eight orientations in approximately $\frac{\pi}{4}$ increments. For each image of the block at a given height and orientation, we select two edge points from each step of the block as the test samples. Figure 8 shows the images of the three step block at seven different heights. Of the seven heights, data sampled from the heights 0 mm, 20 mm and 50 mm are used for both calibrating the constants $\quad_{p}$ and $\quad_{q}$ in eq.(4), and training the neural networks $\text{ANN}_1$ and $\text{ANN}_2$. Data sampled from the heights 10 mm, 30 mm, 40 mm and 60 mm are used for testing the estimation accuracy of the depth formula of eq.(4) and the compensation capability of $\text{ANN}_1$ and $\text{ANN}_2$. Therefore, a total of 336 ( 3 steps $\times$ 2 edge points per step $\times$ 7 heights $\times$ 8 orientations ) samples is generated. Of the 336 samples, 144 are used as the training patterns, and the remaining 192 untrained samples are used as the test set.

Furthermore, in order to evaluate the effect of gray-level contrasts on the estimation accuracy of depth, we have also experimented the placement of the three-step block on two backgrounds with distinct gray-levels. The average gray-level of the block in the image is 100, and the average gray-levels of the two backgrounds used in the experiments are 202
and 145. The block on the background with gray-level 202 is referred to as a high contrast image, whereas the block on the background with gray-level 145 is referred to as a low contrast image. Each contrast category contains 336 samples generated as described above. These two contrast categories generate following four combinations of experiments: 1) Both training samples and test samples are collected from high contrast images, denoted by $E_{HH}$, 2) Training samples are generated from low contrast images, but test samples are collected from high contrast images, denoted by $E_{HL}$, 3) Both training samples and test samples are generated from low contrast images, denoted by $E_{LL}$, and 4) Training samples are generated from high contrast images, but test samples are collected from low contrast images, denoted by $E_{HL}$.

Now we evaluate the performance of the proposed depth estimators under two conditions: 1) calibrating and training the system without using the information of edge orientations, and 2) calibrating and training the system with the information of edge orientations.

Let the constants $P'$ and $Q'$ in eq.(4) be calibrated, and the network ANN$_1$ be trained by the 144 known data samples without considering the information of edge orientations. Table 1 summarizes the experimental results of the root-mean-squares (RMS) depth errors in percentage for the geometric depth formula and the network ANN$_1$. It can be seen from Table 1 that the experiment of $E_{HH}$ gives the best performance with the RMS error of 1.77% from the depth formula. The proposed methods also work well when the training environment does not coincide with the testing environment. The
The neural network approach with the network ANN$_1$ generally yields better depth estimation, especially for the experiments $E_{HL}$, $E_{LL}$, and $E_{LH}$, compared with the geometric depth formula. In general, the RMS error from the depth formula is within 5%, and the RMS error from the network ANN$_1$ is within 3% for the camera at 145 mm distance. These results compare competitively with the measured errors reported in references [10, 12, 18].

Now let the constants $P'$ and $Q'$ in eq.(4) be separately calibrated using the known data samples in each edge orientation. Table 2 presents the experimental results of the RMS depth errors in percentage from the geometric depth formula and the network ANN$_2$ that uses the additional information of edge orientations as the input. The trend resulting from the experiments in Table 2 are consistent with that in Table 1. The experiment $E_{HH}$ yields the best performance with the RMS error of 0.64% from the network ANN$_2$. The experiment $E_{HL}$ yields twofold improvement over the
experiment when the training environment does not coincide with the scene environment.

The network ANN\(_2\) works extremely well even for low-contrast images and non-coincident environments in training and testing. The improvement of the network ANN\(_2\) versus the depth formula is about twofold. Given that the depth formula is used for estimating the depth in the experiments, the use of additional information of edge orientations for training individual \(P'\) and \(Q'\) does not generate significant improvement in the measured depth errors. However, if the neural network approach is used for measuring the depth in the experiments, the network ANN\(_2\) that uses edge orientations to the input layer yields significant improvement in the measured errors, compared with the network ANN\(_1\) that does not use the information of edge orientations as the input. In general, the RMS error from the geometric depth formula is still within 5% even with the information of edge orientations, and the RMS error from the network ANN\(_2\) is within 2% as seen in Table 2. Based on the experimental results described above, the proposed moment-preserving method for estimating the proportion of edge region \(p_e\) and the proposed neural network approach have demonstrated their efficiency and effectiveness for edge-based depth estimation.

6. CONCLUSION

In this paper, the geometric depth formula is described by \(D = P'Q' + D_i\), where \(P'\) and \(Q'\) are constants for a given camera setting, and \(D_i\) is the proportion of edge
region in a small neighborhood window. To compute the value of $\mathbb{D}_b$, the original gray-level image is converted into a gradient image using the Sobel edge operator. For each edge point of interest in the gradient image, the proportion $\mathbb{D}_b$ is then evaluated using the moment-preserving principle. The moment-preserving method provides a closed-form solution to obtain the value of $\mathbb{D}_b$, and is computationally fast. The resulting value of $\mathbb{D}_b$ is between 0 and 1, and increases as the amount of defocus increases. In addition to estimating the depth by using the geometric depth formula, two artificial neural networks ANN$_1$ and ANN$_2$ are also proposed in this study to compensate for the estimation error of the depth formula.

The best depth accuracy is obtained for objects in high-contrast images where the training environment coincides with the scene environment. The proposed methods also work well for objects that their training images and scene images have different gray-level contrasts. Experimental results have shown that the RMS error from the geometric depth formula is within 5%, and the RMS errors from the networks ANN$_1$ and ANN$_2$ are within 3% and 2%, respectively.

The interior edge that distinguishes between two homogeneous surfaces of an object generally has very low gradient magnitude in the gradient image. Since the proposed moment-preserving approach is based on the measurement of the proportion of edge region $\mathbb{D}_b$ in a local window in the gradient image, this restricts the proposed method in its current form to be only applicable to the edges between objects and the background.
REFERENCES


Figure 1. Image formation and defocus in a convex lens

\[ D = \frac{F \cdot v_0}{v_0 - F - \sigma \cdot f} \]

Figure 2. The horizontal and vertical Sobel edge operators.
Figure 3. Images of a multi-step block. (a) The original gray-level image. (b) The corresponding gradient image. The camera is focused on the top of the table where the block is located.
Figure 4. The system architecture of the network ANN₁.

Figure 5. The system architecture of the network ANN₂.
(Only partial connections are presented.)
Figure 6. A three-step block used for experiments.

Figure 7. (a) The plots of the proportion of edge region \( p e \) against the depth \( D \) for varying sizes of windows. (b) The measured errors of depth for varying sizes of windows.

<table>
<thead>
<tr>
<th>Radius of window (pixels)</th>
<th>45</th>
<th>35</th>
<th>25</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS depth error (%)</td>
<td>2.55</td>
<td>2.41</td>
<td>4.02</td>
<td>5.45</td>
</tr>
</tbody>
</table>
Figure 8. The images of the three-step block at seven different heights. $H$ represents the distance from the base of the block to the top of the table.
Table 1. Comparison of RMS depth errors from the depth formula and the network ANN\textsubscript{1} under different gray-level contrasts for training and testing. (The information of edge orientations is not applied.)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>RMS depth error (%)</th>
<th>Depth formula</th>
<th>Network ANN\textsubscript{1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>E H H ( , )</td>
<td>1.77</td>
<td>1.97</td>
<td></td>
</tr>
<tr>
<td>E L H ( , )</td>
<td>3.25</td>
<td>1.97</td>
<td></td>
</tr>
<tr>
<td>E L L ( , )</td>
<td>4.16</td>
<td>2.75</td>
<td></td>
</tr>
<tr>
<td>E H L ( , )</td>
<td>4.27</td>
<td>2.75</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Comparison of RMS depth errors from the depth formula and the network ANN\textsubscript{2} under different gray-level contrasts for training and testing. (The information of edge orientations is utilized.)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>RMS depth error (%)</th>
<th>Depth formula</th>
<th>Network ANN\textsubscript{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>E H H ( , )</td>
<td>1.22</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>E L H ( , )</td>
<td>2.88</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>E L L ( , )</td>
<td>4.06</td>
<td>1.52</td>
<td></td>
</tr>
<tr>
<td>E H L ( , )</td>
<td>4.16</td>
<td>2.00</td>
<td></td>
</tr>
</tbody>
</table>