An Efficient Depth Estimation Technique Using 3-Trait Luminance Profiling

Imran Usman
College of Computing and Informatics
Saudi Electronic University
Riyadh, Saudi Arabia
i.usman@seu.edu.sa

***Abstract*—This paper presents an efficient depth estimation technique for depth image-based rendering process in the 3-D television system. It uses three depth cues, namely linear perspective, motion information, and texture characteristics, to estimate the depth of an image. In addition, suitable weights are assigned to different components of the image based on their relative perspective position of either the foreground or the background in the scene. Experimental results on publicly available datasets validate the usefulness of the proposed technique for efficient estimation of depth maps.**

Keywords-depth estimation; 3D TV; DIBR; depth image; 3-D warping

I. INTRODUCTION

With the advancement of technology in recent decades and the reduced cost of hi-tech hardware, a wide range of new possibilities is realizable as the demand for enhanced viewing experience in the 3D field. In recent years, there has been rapid progress in the fields of image capturing, coding and display which brings the realm of 3D closer to reality than ever before [1]. Real, 3D world incorporates a third dimension (z-axis) that defines depth. Depth is perceived by human vision in the form of binocular disparity. As human eyes are located at slightly different positions, different views of the real world are perceived which are then used by the human brain to reconstruct the depth information of the scene. A 3D display takes advantage of this phenomenon, creating two slightly different images of every scene and then presenting them to the individual eyes. With an appropriate disparity and calibration of parameters, a correct 3D perception can be realized. In the field of 3D, one of the important steps is to generate the 3D content itself. For this purpose, special cameras have been designed to generate 3D models directly. For example, a stereoscopic dual-camera uses co-planar configuration of two monoscopic cameras. Depth information is computed using binocular disparity. Another example is the depth range camera. The examples mentioned above are used for direct generation of the 3D content. On the other hand, there is a tremendous amount of data in 2D which can be converted into 3D. Unfortunately, the conventional 2D camera does not provide any information about the depth in the image. Thus, developing a method for estimating an image's depth map close to the real depth map has become of prime interest [2, 3]. Once accurately estimated, the depth map can then be used to construct the 3D image using techniques such as depth image based rendering (DIBR).

Authors in [4] used the concept of tensor voting at two different levels for depth estimation in a semi-automatic sparse-to-dense structure aware depth estimation method. Authors in [5] proposed a collaborative deconvolutional neural network to concomitantly model semantic segmentation and single-view depth estimation for mutual benefits. Authors in [6] proposed a two-stage depth estimation technique based on Dense Feature Extractor and a Depth Map Generator. The Dense Feature Extractor extracts multi-scale information from input image while keeping the feature maps dense. These multi-scale features are then fused by the Depth Map Generator using a defined attention mechanism. Authors in [7] proposed a method for depth estimation corresponding to every viewpoint of a dense light field. Their proposed algorithm computes the disparity for every viewpoint by taking occlusions into consideration. It also preserves the continuity of the depth space and prior knowledge on the depth range is not required. Authors in [8] proposed a network for depth estimation combining encoder-decoder architecture with adversarial loss. Authors in [9] proposed a depth estimation framework based on deep convolutional neural network. It is based on depth prediction and depth enhancement sub-networks. All the mentioned works are computationally extensive and resource hungry. In this work, we propose a simple idea of depth profiling based on three depth cues, namely linear perspective, motion detection and texture characteristics, to estimate the depth of an image.

II. THE PROPOSED DEPTHMAP ESTIMATION TECHNIQUE

In order to generate the depth map, three cues are observed: motion information, linear perspective and texture characteristics. In motion information, the temporal difference is taken between the two consecutive frames and then thresholding is applied. In linear perspective, the vanishing line or the vanishing point [3] is used to find the points that are farthest in the image. Corresponding to these, gradient depth values are assigned. Texture characteristics are determined and analyzed. In addition, this work further demonstrates the use of

Corresponding author: Imran Usman

bilateral or Gaussian filter for smoothing of the depth map to achieve better results. Figure 1 presents the general architecture of the proposed technique in a flow diagram. The details of the proposed technique are presented in the following subsections.

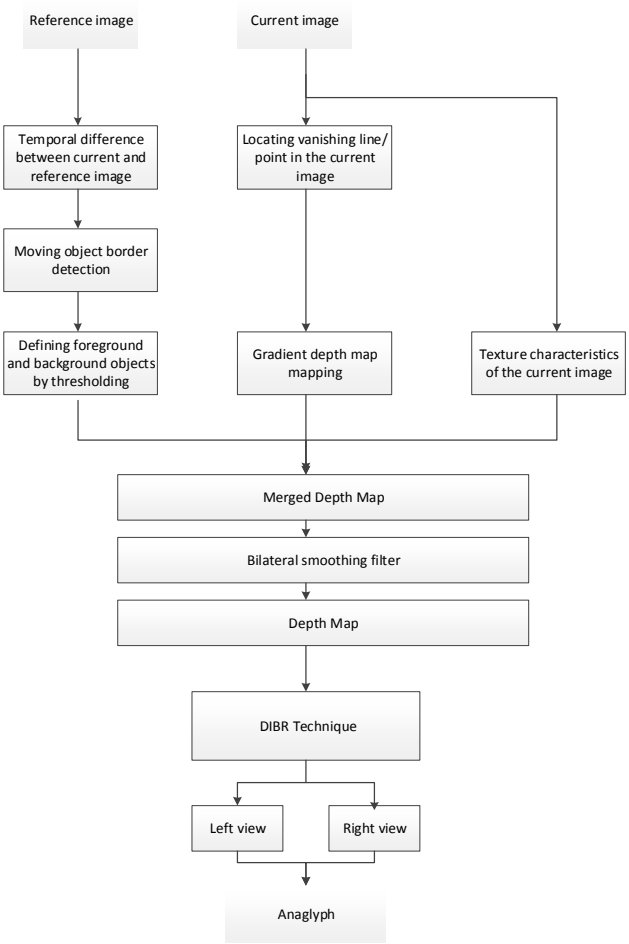

Fig. 1. General architecture of the proposed technique

A. Motion Information

The human brain pays more attention to the things that are in motion than to those which are static. So, information in the form of video (objects in motion) is more understandable and clearer than steady images. Objects that are far away, or are distant, seem to be static or moving very slowly when we look at them while we are in motion, whereas the objects that are nearer to us move fast when observed from a moving source. This effect is known as motion parallax. Things closer to us seem to move rapidly while the objects far away move slowly or are at rest. Keeping this fact into consideration, we use the approach of temporal difference to find out objects that are moving or are in a state of rest. By doing so, the objects are assigned to the foreground domain and the background domain in an image by assigning threshold values to the points in the image. The formula for finding the temporal difference is:

$$\Delta = \left| \rho_c(i,j) - \rho_{c-1}(i,j) \right| \quad (1)$$

where ρ_c is the current frame pixel at (i, j) and ρ_{c-1} is the reference frame pixel at point (i, j). After calculating the temporal difference between the frames, the values are compared with the threshold value. The threshold values are computed as:

$$\Delta = \begin{cases} 1, & if\ \Delta(i,j) > \partial_c \\ 0, & otherwise \end{cases} \quad (2)$$

where ∂_c is the standard deviation given as:

$$\partial_c = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(\rho_i - \overline{\rho})} \quad (3)$$

where m is the total number of pixels and $\overline{\rho}$ is the mean of total pixels in a frame. From the first two equations, moving objects are found. In depth map, the pixels value ranges from 0 to 255, whereby 0 represents the farthest point and 255 shows the nearest point in the image. In the algorithm, the values of temporal difference show which objects are static in the image and which are in motion. If the temporal difference between each pixel of the current frame and the reference frame is less than the standard deviation of the frame, then those pixels are assigned with the value 0. This denotes that these sets of points belong to the background (farthest region). Similarly, if the temporal difference generates a value greater than the threshold value, the pixels are assigned with higher gray level values, indicating that these points belong to the foreground (nearest region) of the image. In this cue the frame or image is separated between two parts: the foreground and the background.

B. Linear Perspective

Objects that are placed far away subtend a very small angle in our eye which the mind interprets as being the farthest in the scene. Parallel lines in the image, like the long railway tracks running in the image, provide linear perspective that helps to see objects in three-dimensional view. Linear perspective is also a depth cue related to texture gradient and relative size. In every image, when observing closely, we find either the vanishing line or point. The vanishing point is the farthest point in the image where all parallel lines converge. In terms of graphical perspective, it is a point in the image plane that is defined by a line in the space. The vanishing line in an image is defined as a set of vanishing points which are all located in one line. The vanishing line or point represent the farthest point in the image which ultimately gives the depth information. In this work, the vanishing line or point are located empirically using different tools. This algorithm, in general, discusses three possibilities for the occurrence of vanishing point/line. After finding the vanishing point/line, depth gradient values are assigned accordingly. These are discussed below in detail in three different cases.

1) CASE I: If Horizontal Line Exists

This case is applied when only a horizontal vanishing line appears. Pixels above the vanishing line are assigned '0' gray level and the pixels below this line are assigned lighter shades

of the gray level (1-255), since they move away from the vanishing line.

2) CASE II: If a Vanishing Point Appears on the Left Part of the Image

The approach used for assigning gradient value in this case is to first divide the image into two portions, left and right respectively. When the vanishing point appears on the left side of the frame, the upper left corner of the image is assigned '0' level of gray. Consequently, the upper right part of the frame is assigned ascending values of gray from 0 to 255. The lower left corner of the frame is assigned ascending values of gray level from darker to lighter shades of the gray scale. Finally, the lower right part of the frame is assigned ascending values of gray ranging from 0 to 255 diagonally. The following are the four regions in which the image is divided after locating a vanishing point for the assignment of gradient values.

- Upper left regions:

$$Image(1:i,1:j) = 0 \quad (4)$$

where i and j are the coordinates of the vanishing point in the image.

- Lower left Region:

$$Image(1:i,1:j) = 0 \quad (5)$$

where i and j are the vanishing point coordinates in the image and m is the total number of rows in the image.

- Upper Right Region:

$$Image(1:i, j+1:n) = \begin{cases} 0 \\ 255 \end{cases} \quad gradient\ values \quad (6)$$

where i and j are vanishing point co-ordinates in the image and n is the total number of rows in the image.

- Lower Right Region:

$$Image(i+1:m, j+1:n) = \begin{cases} 0 \\ 255 \end{cases} \quad gradient\ values \quad (7)$$

3) CASE III: If a Vanishing Point Appears on the Right Part of the Image

The upper right corner of the image is assigned '0' value of gray level, and the upper left part of the frame is assigned ascending values of gray from 0 to 255. Similarly, the lower right corner of the frame is assigned ascending values of gray level from darker to lighter shades of the gray scale. The left lower part of the frame is assigned ascending values of gray ranging from 0 to 255 diagonally. The details of the four regions are given below.

- Upper Right regions:

$$Image(1:i, j+1:n) = 0 \quad (8)$$

where, i and j are the co-ordinates of the vanishing point in the image and n is the total number of columns in the image.

- Lower Right Region:

$$Image(i+1:m, j:n) = \begin{cases} 0 \\ 255 \end{cases} \quad gradient\ values \quad (9)$$

where, i and j are vanishing point coordinates in the image, m is the total number of rows in the image, and n is the total number of columns.

- Upper Left Region:

$$Image(1:i, j-1:1) = \begin{cases} 0 \\ 255 \end{cases} \quad gradient\ values \quad (10)$$

where, i and j are vanishing point coordinates in the image .

- Lower Right Region:

$$Image(i+1:m, j-1:1) = \begin{cases} 0 \\ 255 \end{cases} \quad gradient\ values \quad (11)$$

C. Texture Characteristics

Texture characteristics of an image also provide very important clues about the depth of the image. It is a common observation that when any object is placed closer, it gives more details about its color, shape, brightness etc. The objects that are located at some distance, or are located very far away are hard to be determined in terms of their brightness and contrast. Their details might get lost, get blended in the background and may even seem to be a part of the background itself. This common observation adds up value to the importance of texture characteristics in finding the nearer and the farthest objects in an image providing us with depth information. The texture characteristics of an image include brightness, luminance, hue and chrominance. It has been discovered that human perception gives priority to brightness, hue, and chrominance [10]. Brightness is the characteristic of the visual perception which helps in recognizing whether the object is reflecting or radiating light or not. It is also termed as an attribute that tells us about the luminance of an object. Objects present in the foreground usually have greater value of brightness as compared to the objects present in the background. So, greater values of brightness are assigned to the objects in the foreground and lesser values to the objects present in the background of an image. The quantity of light that is absorbed or reflected from the object is described as luminance, which is a property defining color. Hue shows the wavelength of the energy band of the light at which it has maximum value. Chrominance is the remaining component of the color arrangement when luminance is removed from it. Converting images into gray scale gives us the values of pixels demonstrating texture characteristics [11]. Converting images into gray scale means removing the RGB contents from the image and assigning each pixel values between 0 to 255 according to its characteristics.

D. Merging the Three Cues to Generate the Depth Map

The three depth cues (motion detection, linear perspective and texture characteristics) are used to estimate the depth of the

image. In order to convert a 2D image into 3D, the image is divided into two parts: background and foreground. While merging the cues for estimating the values in depth map, more values are assigned to the objects present in the foreground of the image, whereas the objects present in the background of the image are assigned smaller weights. This is done because objects present in the foreground of the image receive more attention by human brain than objects in the background. Cue merging to generate depth map is divided into the following two cases.

1) CASE I: Foreground

For the foreground region, the depth is calculated as:

$$depth_f = \omega_{mo} \times depth_{mo} + \omega_{lin} \times depth_{lin} \quad (12)$$

where, $depth_{mo}$ and $depth_{lin}$ are the depth values extracted in motion information and linear perspective cues respectively, ω_{mo} is the weight assigned to the motion information cue, and ω_{lin} is the weight assigned to the linear perspective cue. As discussed earlier, moving objects get more attention from the human visual system, hence, higher weights are assigned to the depth values calculated in motion information cue i.e. $\omega_{mo} = 0.7$. The value assigned to linear perspective cue is $\omega_{lin} = 0.25$. These values are determined and assigned empirically after subjective tests of the human visual system.

2) CASE II: Background

For the background region, the depth is calculated as:

$$depth_b = \omega_{lin} \times depth_{lin} + \omega_{tex} \times depth_{tex} \quad (13)$$

where, $depth_{tex}$ and $depth_{lin}$ are the depth values calculated in texture characteristics and linear perspective respectively. Linear perspective dominates gradient depth values for generating depth in static objects and background, therefore, a higher weight value is assigned to it and a lower weight value is assigned to texture characteristics. Hence, $\omega_{lin} = 0.7$ and $\omega_{tex} = 0.25$. The final depth map is generated by combining the results of these two cases.

$$depth = depth_f + depth_b \quad (14)$$

E. Smoothing of the Depth Map

In order to reduce noise in the generated depth map while preserving its edges, and to achieve a smooth depth map, bilateral filters [12] are used (other filters, e.g. Gaussian, can also be used). Another strategy to effectively remove noise is to use a bilateral filter along with directional Gaussian kernels which are edge-dependent for the non-hole regions. In addition, considering the similarity of the depth pixels, a trilateral filter can be selectively used in combination according to the type of pixels.

III. RESULTS AND DISCUSSION

The proposed technique is implemented in Matlab using an Intel Core i7 3.4GHz processor with 8GB RAM. For experimentation, we used the Middlebury 2005 dataset [13] which is comprised of a number of different images having a variety of depth perspectives. These images also contain the corresponding stereo images and the created anaglyph images. The anaglyph images are created by either using the stereo images, or by using the center image and the associated depth map. Figure 2 presents some of the images from the Middlebury dataset and their associated depth maps. These random images show daily life objects with different depth perspectives. The images are rectified and radial distortion has been removed. The depth maps are created using a focal length of 3740 pixels, and a baseline of 160mm, while the intensity and disparity values are kept at 60. More details on other parameter settings can be found in [13].

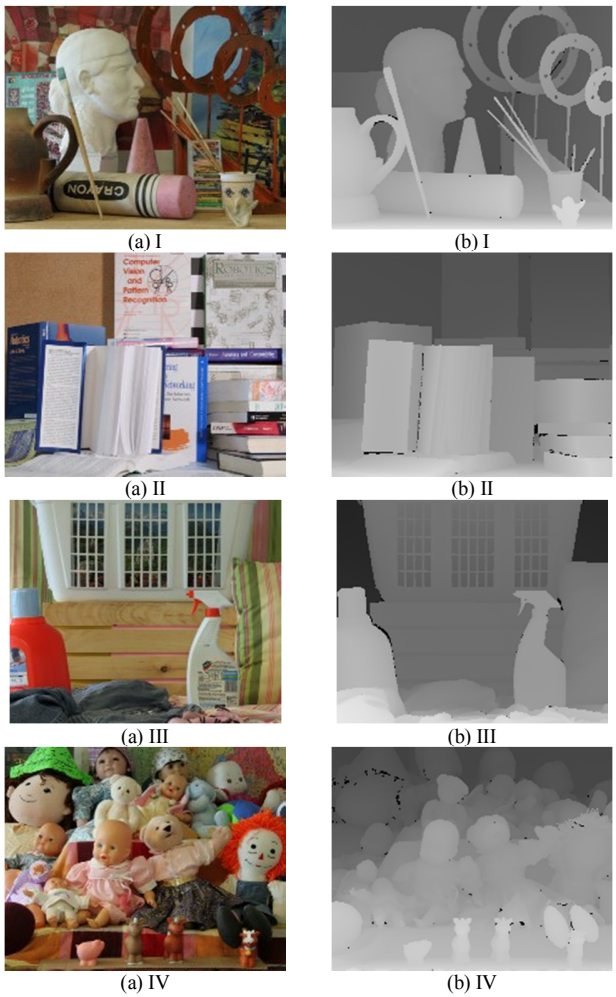

Fig. 2. Sample images from the Middlebury dataset [13]. (a) I-IV are the original images, and (b) I-IV are the corresponding depth maps.

Table I presents the performance comparison of the proposed technique and the depth maps from the Middlebury data set for Art, Books, Computer, Dolls, and Drumsticks images. For comparison, we assume the depth maps from the Middlebury data set as the standard depth maps corresponding

to the scene and compare them with the luminance profiled depth maps generated through the proposed technique. We used peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) as comparison metrics between the estimated depth map and the original depth map. It can be observed from Table I that the proposed technique yields an acceptable level of similarity compared with the standard depth maps for all of the selected images.

TABLE I. PERFORMANCE COMPARISON OF THE PROPOSED TECHNIQUE AND THE DEPTH MAPS FROM MIDDLEBURY DATASET

Depth map image name	PSNR	SSIM
Art	28.970	0.689
Books	28.516	0.651
Computer	29.387	0.713
Dolls	29.146	0.704
Drumsticks	28.632	0.701

Figure 3 shows the average performance comparison of the proposed technique and the standard depth maps using a number of images from the Middlebury data set. It is to be noted that for the purposes of comparison and display, the values of PSNR are scaled down using a dividing factor of 30 in order to bring them close to the values of SSIM. Once again, it can be observed that the proposed technique produces depth maps that are in acceptable range of similarity and luminance profiling as compared to the original depth maps.

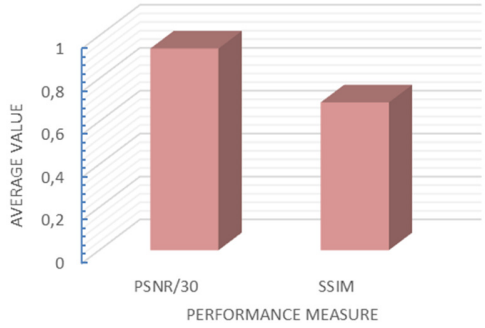

Fig. 3. Average performance comparison of the proposed and standard technique on depth maps

IV. CONCLUSION

This work presents a simple depth estimation method which is computationally fast and resource efficient. The proposed technique utilizes linear perspective, motion detection and texture characteristics to estimate the luminance profiling of an image scene. In motion information, the temporal difference is taken between two consecutive frames and then thresholding is applied. In linear perspective, the vanishing line or the vanishing point are used to find the farthest points in the image. Corresponding to these, gradient depth values are assigned. Texture characteristics are determined and analyzed in order to estimate the depth map. Finally, bilateral filters are used to smooth the depth map. The experimental results show that the depth maps generated through the proposed technique are of acceptable quality and can be used in real world applications.

ACKNOWLEDGMENT

The author acknowledges Saudi Electronic University's help and financial support.

REFERENCES

[1] J. Son, B. Javidi, S. Yano, K. Choi, "Recent Developments in 3-D Imaging Technologies", Journal of Display Technology, Vol. 6, No. 10, pp. 394-403, 2010

[2] L. Zhang, W. J. Tam, "Stereoscopic image generation based on depth images for 3D TV", IEEE Transactions on Broadcasting, Vol. 51, No. 2, pp. 191-199, 2005

[3] A. Almansa, A. Desolneux, S. Vamech, "Vanishing point detection without any a priori information", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 4, pp. 502-507, 2003

[4] B. Wang, J. Zou, Y. Li, K. Ju, H. Xiong, Y. F. Zheng, "Sparse-to-Dense Depth Estimation in Videos via High-Dimensional Tensor Voting", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 29, No. 1, pp. 68-79, 2019

[5] J. Liu, Y. Wang, Y. Li, J. Fu, J. Li, H. Lu, "Collaborative Deconvolutional Neural Networks for Joint Depth Estimation and Semantic Segmentation", IEEE Transactions on Neural Networks and Learning Systems, Vol. 29, No. 11, pp. 5655-5666, 2018

[6] Z. Hao, Y. Li, S. You, F. Lu, "Detail Preserving Depth Estimation from a Single Image Using Attention Guided Networks", 2018 International Conference on 3D Vision (3DV), Verona, Italy, September 5-8, 2018

[7] X. Jiang, M. L. Pendu, C. Guillemot, "Depth Estimation with Occlusion Handling from a Sparse Set of Light Field Views", 25th IEEE International Conference on Image Processing, Athens, Greece, October 7-10, 2018

[8] M. Carvalho, B. Le Saux, P. Trouve-Peloux, A. Almansa, F. Champagnat, "On Regression Losses for Deep Depth Estimation", 25th IEEE International Conference on Image Processing, Athens, Greece, October 7-10, 2018

[9] X. Duan, X. Ye, Y. Li, H. Li, "High Quality Depth Estimation from Monocular Images Based on Depth Prediction and Enhancement Sub-Networks", IEEE International Conference on Multimedia and Expo, San Diego, USA, July 23-27, 2018

[10] K. Ghosh, S. K. Pal, "Some Insights Into Brightness Perception of Images in the Light of a New Computational Model of Figure–Ground Segregation", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol. 40, No. 4, pp. 758-766, 2010

[11] M. Song, D. Tao, C. Chen, X. Li, C. W. Chen, "Color to Gray: Visual Cue Preservation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, pp. 1537-1552, 2010

[12] A. V. Le, S. W. Jung, C. S. Won, "Directional Joint Bilateral Filter for Depth Images", Sensors, Vol. 14, No. 7, pp. 11362-11378, 2014

[13] http://vision.middlebury.edu/stereo/data/scenes2005/[Accessed: 21-Apr-2019]