RESEARCH ARTICLE OPEN ACCESS

Improvised Filter Approach for Estimating Depth from Monocular Images

Aditya Venkatraman[1], Sheetal Mahadik[2]
[1](Department of Electronics and Telecommunication, ST Francis Institute of Technology, Mumbai, India)
[2] (Department of Electronics and Telecommunication, ST Francis Institute of Technology, Mumbai, India

ABSTRACT
Depth estimation or extraction refers to the set of techniques and algorithm's aiming to obtain distance of each and every pixel from the camera view point. Depth Estimation poses various challenges and has wide range applications.Depth can be estimated using different monocular and binocular cues. In [1] depth is estimated from monocular images i.e. by using monocular cues. The filters used for texture gradient estimation in [1] detect false edges and are susceptible to noise. In this paper for depth estimation from monocular images, we have obtained a monocular cue named texture gradient using Canny edge detector which is more robust to noise and false edges as compared to six oriented edge detection filters (Nevatia Babu filters) used in [1]. We, have reduced the dimensions of feature vector as compared to [1] and hence feature optimization for texture gradient extraction is achieved.Even though reduced set of features are used our outputs gave better results as compared to [1].

Keywords **–** Canny**,** depth estimation, linear least squares problem, monocular cue, texture gradient

I. INTRODUCTION

Depth estimation is an important research area since it has its extensive use in robotic vision and machine vision because of which it will be possible to construct robots and cars that move themselves or automatically without human assistance. In related work, Saxena's algorithm [1] generates depth map from monocular images. Depth estimation has important applications in robotics, scene understanding and 3-D reconstruction.

By using depth estimation techniques, distance of each point seen in the scene can be obtained from the camera view point. Depth estimation has continuously become an effort-taking subject in visual computer sciences. Conventionally, depths on a monocular image are estimated by using a laser scanner or a binocular stereo vision system. However, using a binocular stereo vision system requires adjustment on the camera taking a scanner picture, and using a laser scanner takes huge capitals as well, so both these apparatuses bring significant complexities. Therefore, some algorithms have been developed to process monocular cues in the picture for depth estimation. In related work, Michel's, Saxena & Ng [2] used supervised learning to estimate

1-D distances to obstacles, for the application of autonomously driving a remote control car. Most work on depth estimation has focused on binocular vision (stereopsis) [3] and on other algorithms that require multiple images, such as shape from shading [4] and depth from focus [5]. Gini & Marchi [6] used single-camera vision to drive an in- door robot, but relied heavily on known ground colors and textures.

In this paper a more accurate depth map as compared to [1] is obtained by using an improvised filter for texture gradient extraction.

II. METHODOLOGY

In this paper monocular cues are used for depth estimation. There are different monocular cues such as texture variations, texture gradients, interposition, occlusion, known object sizes, light and shading, haze, defocus etc. which can be used for depth estimation .The monocular cues used in this paper are haze, texture gradient and texture energy as these cues are present in most images as in [1]. Many objects texture appear different depending on their distances from the camera viewpoints which help in indicating depth. Texture gradients, which capture the distribution of the direction of edges, also help to indicate depth. Haze is another cue resulting from atmospheric scattering light.

To obtain haze, averaging filters are used. To obtain texture energy, nine Laws filters are used and to obtain texture gradient, Canny edge detector is used as compared to six edge detectors (Nevatia Babu filters) used in [1].

Fig. 1. Shows block diagram for estimating depth from monocular images

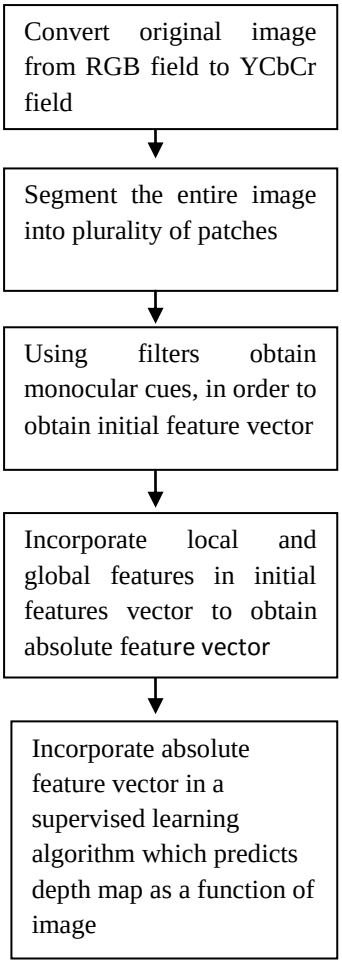

Figure 1: Block diagram for depth estimation from monocular images

After obtaining these monocular cues, using them, an initial feature vector is obtained. Although this initial feature vector gives some information about depth using local information such as variation in texture and color of a patch, these are insufficient to determine depth accurately and thus global properties have to be used. For example, just by looking at a blue patch it is difficult to tell whether this patch is of a sky or a part of a blue object. Due to these difficulties, one needs to look at both the local and global properties of an image to determine depth.

Thus the final feature vector which includes both local as well as global features is used in a supervised learning algorithm which predicts depth map as a function of image.

The detailed explanation of feature calculation and Learning is done in section 2.1, 2.2 and 2.3.

2.1 FEATURE VECTOR

The entire image is initially divided into small rectangular patches which are arranged in a uniform grid, and a single depth value for each patch is estimated. Here absolute depth features are used which are used to determine absolute depth of a patch. Three types of monocular cues are selected: texture variations, texture gradients and haze, as these are present in most of the indoor and outdoor images. Texture information is mostly contained within the image intensity channel so Laws' mask is applied to this channel, to compute the texture energy. Haze is reflected in the low frequency information in the color channels, and is captured by applying a local averaging filter to the color channels. To compute an estimate of texture gradient, Canny edge detector [7] is used which is more robust to noise and false edges as compared to six oriented edge filters (Nevatia Babu filters) used in [1]

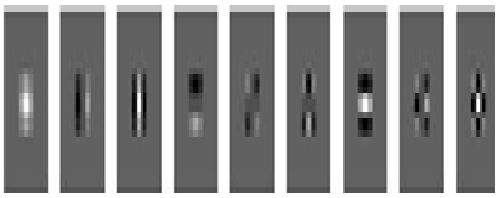

Figure 2: Filters used for depth estimation wherein, the nine filters indicate Law's 3X3 masks.

2.2 ABSOLUTE FEATURE VECTOR

For a patch i in an image I(x,y) the outputs of two local averaging filter, Canny edge detector and nine Law's mask filters which are in total of 12 outputs are used as:

$$E_i(n) = \sum_{(x,y)\ patch(i)} |I(x,y) * Fn(x,y)|^k \quad (1)$$

Where for $F_n(x, y)$ indicates the filter used. Here n indicates the number of filters used, where, n=1 to 12. Here k = {1, 2} gives the sum absolute energy and sum squared energy respectively. Thus an initial feature vector of dimension 12 is obtained as compared to dimension 17 in [1].

With these filters, local image features for a patch is obtained. But to obtain absolute depth of a patch, local image features centered on the patch are insufficient, and more global properties of the image have to be used. Image features extracted at multiple image resolutions are used for this very purpose. Objects at different depths exhibit very different behaviors at different resolutions, and using multi-scale features (scale 1, scale 3, and scale 9) allows us to capture these variations. Computing features at multiple spatial scales also helps to account for different relative sizes of objects. A closer object appears larger in the image, and hence will be captured in the larger scale features. The same object when far away will be small and hence be captured in the small scale features. To capture additional global features, the features used to predict the depth of a particular patch are computed from that patch as well as the four neighboring patches which is repeated at each of the three scales, so that the feature vector of a patch includes features of its immediate neighbors at large scale, features of far neighbors at a larger

spatial scale, and again features of very far neighbors at an even larger spatial scale. Along with local and global features, many structures found in outdoor scenes show vertical structure so, additional summary features of the column that the patch lies in, are added to the features of a patch.

For each patch, after including features from itself and its four neighbors at 3 scales, and summary features for its four column patches, absolute feature vector of dimension 24*19=456 is obtained as compared to dimension of 34*19=646 in [1].

2.3 SUPERVISED LEARNING

A change in depth along the row of any image as compared to same along the columns is very less. This is clearly evident in outdoor images since depth along the column is till infinity as the outdoor scene is unbounded due to the presence of sky. Since depth is estimated for each patch in an image, feature is calculated for each patch whereas learning done for each row as changes along the row is very less. Linear least squares method is used for learning whose equation is given by:

$$\Theta_r = \min \left(\sum_{i=1 \text{ to } N} (d_i - x_i^T \Theta_r)^2\right) \quad (2)$$

In equation (2), N represents the total number of patches in the image. Here d_i is the ground truth depth map for patch i, x_i is the absolute feature vector for patch i. Here Θ_r, where r is the row of an image, is estimated using linear least squares problem.

III. EXPERIMENTS

3.1 DATA

The data set is available online (http://make3d.cs.cornell.edu/data.html) which consists of 400 images and their corresponding ground truth depth maps which includes real world set of images of forests, campus (roadside) areas and indoor images

3.2 RESULTS

Depth is estimated from real-world test-set images of forests (containing trees, bushes, etc.), campus areas (buildings, trees and roads). The algorithm was trained on a training set comprising images from all of these environments. Here 300 images are used for training and the rest 100 images are used for testing.

TABLE 1 shows the comparison of depth map using Canny edge detector and depth map obtained by [1] i.e. by using Nevatia Babu filters based on RMS (root mean square) errors in various environments such as campus, forest, indoor and areas which include both campus and forest. The result on the test set shows that our output less RMS error as compared to [1].

TABLE 2 shows the comparison of depth map obtained using Canny edge detector and depth map obtained by [1] i.e by using Nevatia Babu filters based on computation time ,set of features used and average RMS error .It can be seen that we have reduced set of features as only 456 as compared to 646 features used by [1]. Since we have used less set of features , our total computation time is only as compared to 24sec total computation of 44sec [1].Even though we have used less set of features our average RMS error is less than [1]

Table 1: Comparison of RMS errors resulting from depth map using Canny edge detector and depth map obtained using [1]

Test Environments	Depth map using Nevatia Babu filters	Depth map using Canny edge detector
Forest	1.1035	1.0365
Campus	0.6367	0.5968
Forest &Campus	0.6254	0.4386
Indoor	1.234	1.220

Table 2: Comparison of depth map using Canny edge detector and depth map using Nevatia Babu filters based on different parameters

Parameters	Depth map using Nevatia Babu filters	Depth map using Canny edge detector
Average RMS error	0.899	0.8229
Set of features	646	456
Computation time(sec)	44	24

Figure 3: Original image (Forest)

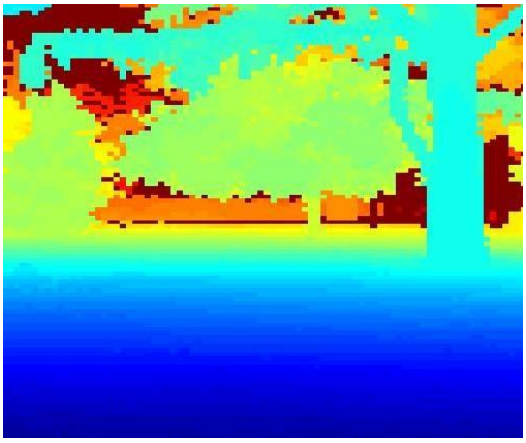

Figure 3.1: Ground truth depth map (Forest)

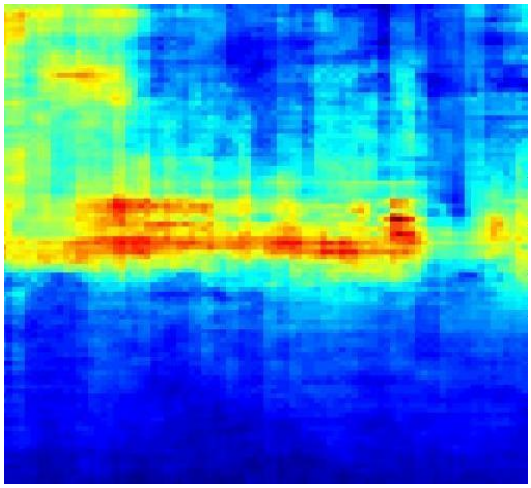

Figure 3.2: Depth map using Nevatia Babu filters (Forest)

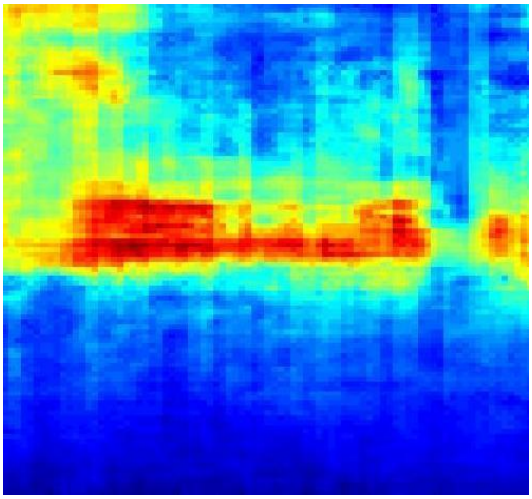

Figure 3.3: Depth map using Canny edge detector (Forest)

Figure 4: Original image (Campus-roadside (combined)

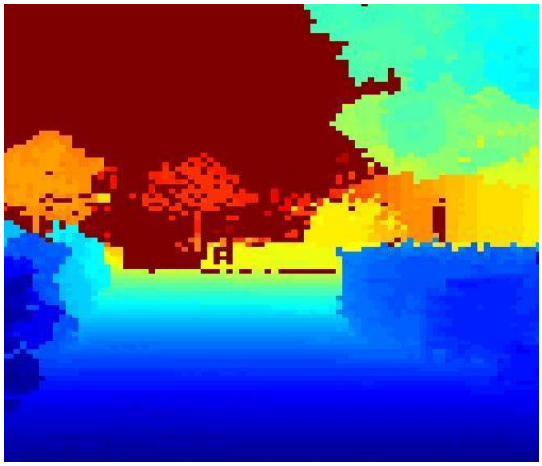

Figure 4.1: Ground truth depth map (Campus-roadside (combined))

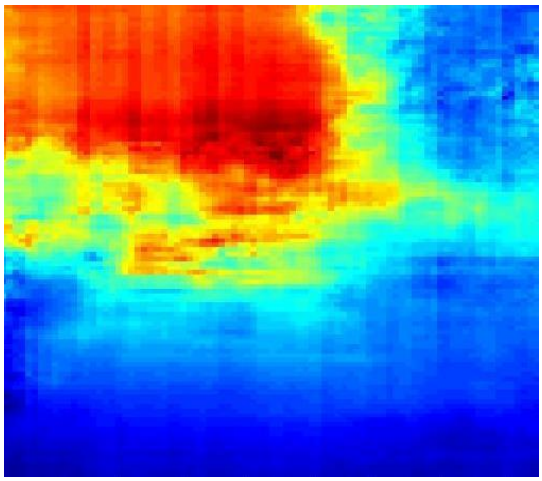

Figure 4.2: Depth map using Nevatia Babu filters (campus-roadside (combined)).

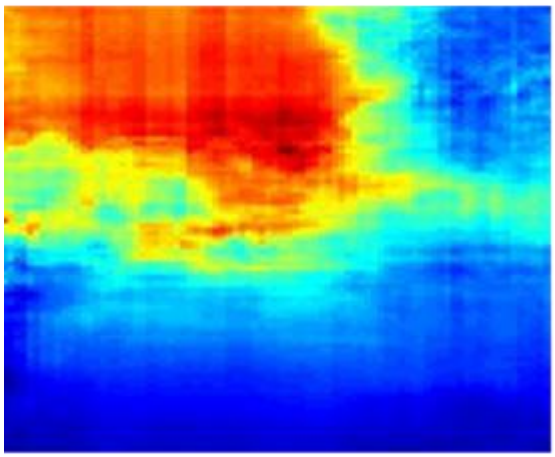

Figure 4.3: Depth map using Canny edge detector (Campus-roadside (combined))

Figure 5: Original image (campus-roadside and forest (combined))

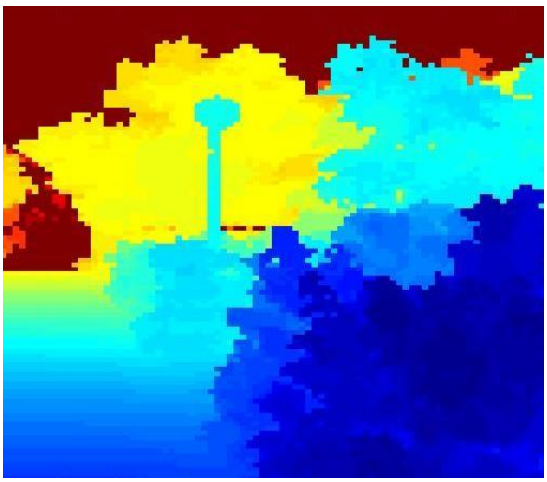

Figure 5.1: Ground truth depth map (campus-roadside and forest (combined))

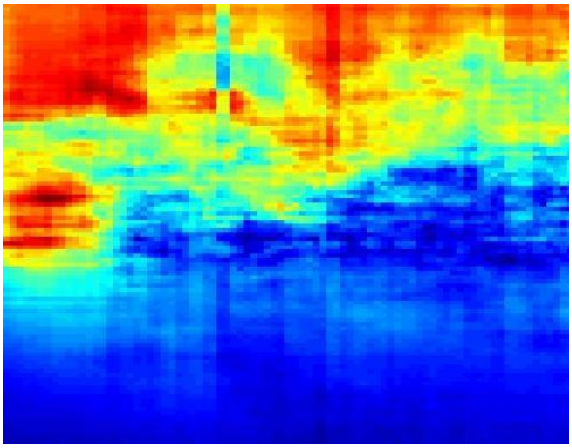

Figure 5.2: Depth map using Nevatia Babu filters (campus-roadside (combined)).

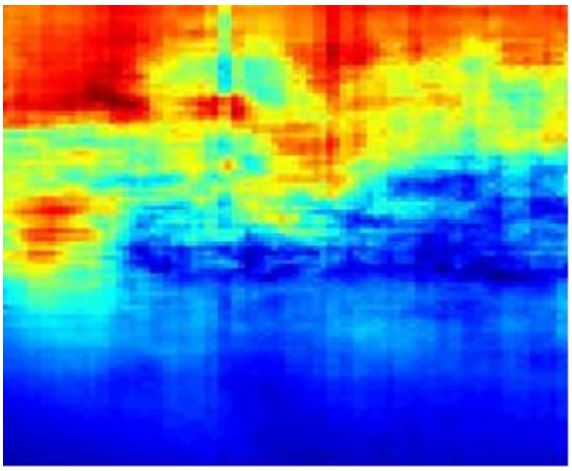

Figure 5.3: Depth map using Canny edge detector (campus-roadside (combined)).

Figure 6: Original image (indoor)

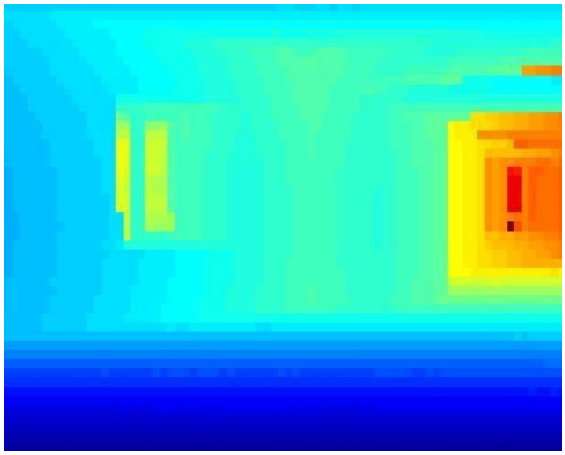

Figure 6.1: Ground truth depth map (indoor)

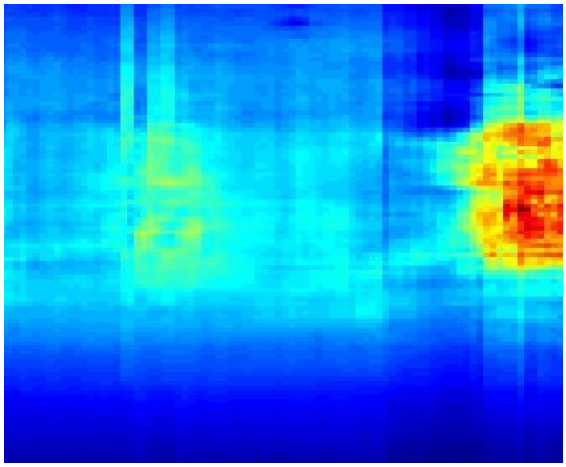

Figure 6.2: Depth map using Nevatia Babu filters (indoor).

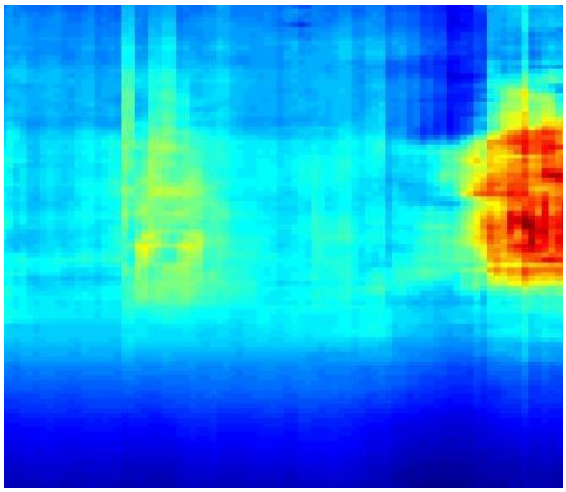

Figure 6.3: Depth map using Canny edge detector (indoor).

IV. CONCLUSION

A detailed comparison of our method and the method used by [1] is done in terms of average RMS errors, computation time and set of features used. This detailed comparison is done in set of various indoor and outdoor environments. It is found that by using Canny edge detector for estimating a monocular cue named texture gradient not only set of features are reduced but also depth map is predicted more accurately as compared to [1].

REFERENCES

[1] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. "Learning depth from single monocular images", In *NIPS 18*, 2006.

[2] Jeff Michels, Ashutosh Saxena and A.Y. Ng. "High speed obstacle avoidance using monocular vision", In *Proceedings of the Twenty First International Conference on Machine Learning* (ICML), 2005

[3] D.Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int'l Journal of Computer Vision*, 47:7–42, 2002M.

[4] M. Shao, T. Simchony, and R. Chellappa. New algorithms from reconstruction of a 3-d depth map from one or more images. In *Proc IEEE CVPR*, 1988

[5] S.Das and N. Ahuja. Performance analysis of stereo, vergence, and focus as depth cues for active vision. *IEEE Trans Pattern Analysis& Machine Intelligence*, 17:1213–1219, 1995.

[6] G. Gini and A. Marchi. Indoor robot navigation with single camera vision. In *PRIS*, 2002.

[7] J. Canny, "A computational approach to edge detection," *IEEE Trans.Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986