ARTIFICIAL INTELLIGENCE ATLAS SESSION

AN END TO END LEARNING APPROACH FOR DISTANCE ESTIMATION TRAINED WITH ARTIFICIALLY GENERATED STEREO IMAGES

Nebojša Nešić*,
Mladen Vidović,
Ivan Radosavljević,
Aleksandra Mitrović,
Đorđe Obradović

Faculty of Informatics and Computing,
Singidunum University,
Belgrade, Serbia

Abstract:

This paper proposes a solution for distance estimation using stereo images. The solution is a convolutional neural network that takes two images as an input, and outputs the distance estimate, without the need for prior camera calibration or disparity map calculation. The dataset used for training consists of images generated from an artificially constructed 3D scene. The training algorithm used was stochastic gradient descent. Evaluation of the solution was conducted on a separate dataset. Mean absolute error after the evaluation was 1.59 m, while the median value of the absolute error was 1.2 m. These results show that the proposed solution is a valid proof of concept for the usage of convolutional neural networks for the distance estimation of objects in stereo images in a single step.

Keywords:

artificially generated data, convolutional neural networks, stereo vision.

INTRODUCTION

Distance measurement has applications in industrial environments and is an important component of obstacle avoidance systems used in autonomous vehicles. Some of the more usual approaches to distance measuring involve use of sonars, radars, LiDARs and cameras. Radars, sonars and LiDARs rely on time-of-flight to calculate distance of the object of interest [1]. Although radars and sonars can be used to accurately measure distance to single objects, they are not suited for high resolution measurements such as those required for construction of point clouds. Optical devices such as LiDARs do not suffer from this problem but their accuracy can be hindered by environmental conditions such as adverse weather [2]. Cameras, as another example of optical devices, can also be used to measure distance of single objects and to create point clouds. Accuracy of measurements acquired by cameras is largely dependent on environmental conditions and proper device calibration. However, an additional advantage of cameras, compared to the aforementioned devices, is that they can be used to collect more general information about the environment. In order to capitalize on the advantages of using

Correspondence:
Nebojša Nešić

e-mail:
nnesic@singidunum.ac.rs

DOI: 10.15308/Sinteza-2020-3-7

cameras, while trying to mitigate their disadvantages, in this paper we propose a solution for distance estimation using cameras without the need for prior calibration.

This paper is composed of four chapters. In chapter 1 we will present work related to our research. The following chapter describes the proposed method. In chapter 3 we will present the results obtained by applying the proposed method to test data. Finally, in chapter 4, the conclusion, we will review the results and propose further improvements.

1. RELATED WORK

There is a significant amount of research on the subject of absolute distance measuring. Most of the research in this field describes methods relying on time-of-flight distance measurement or optical triangulation. In [3], the authors describe a high precision distance measurement method using an ultrasonic sensor to measure the time of flight of an ultrasonic pulse reflected by the target. Paper [4] describes a laser range finder for industrial distance measurements achieving a 2 mm absolute error in the distance range of 0.5 - 34.5 m. A different approach to measure distances is to use images, as proposed in [5], in which the authors use inverse perspective mapping to transform a forward-facing image, taken from an automobile-mounted camera, to a top-down view, which is then used to estimate the distance to an object. Camera tilt, change of velocity of the vehicle the camera is mounted on, and undulations of the road were a negative impact on accuracy. The authors of [6] use stereo images to create a 3D point cloud and calculate the distance to points of interest, which are tightly grouped clusters in the point cloud. This method has a maximum detection range of about 90 m, with optimal range being between 10 – 60 m. The achieved measurement error is higher than that of solutions using radar systems, but was comparably low for a vision-based system, ranging from less than 10 cm at 10 m to about 2 m error at 95 m distance. In [7] stereovision is used to construct 3D world coordinates of objects, calculated by using camera parameters and stereoscopic image data, with experimental results showing that, in the optimal range of 4 – 50 m, under reasonable illumination conditions, an error of 5% of the estimated distance can be expected. A method using stereo vision to generate disparity maps and extract stixels, which are then clustered into individual objects, for which the distance is measured is proposed in [8]. The method was not reliable at distances less than 5 m. At distances above this threshold, the system achieved an accuracy of 92.51%. In [9, 10] the authors propose methods to generate disparity maps on image pairs using convolutional neural networks (CNN). The authors of [9] constructed two different models, one optimized for accuracy and the other for computation speed. The model optimized for accuracy achieved 3.89% pixel-wise error rate, while the fast model achieved an error rate of 4.62%. The model proposed in [10] achieved an error rate of 13% on the same dataset.

In the previously mentioned papers utilizing images to calculate the distance to an object, prior calibration of the cameras is necessary, and the camera parameters must be known. However, authors of [11, 12, 13] propose methods for automatic extraction of camera parameters.

2. PROPOSED METHOD

Papers presented in the previous section showed that neural networks can be used to generate disparity maps and to extract camera parameters. However, the proposed solutions never combined those tasks in order to estimate the distance to the target. Given these circumstances we propose a solution, in the form of a neural network, that not only integrates both tasks, but is also capable to estimate distance to a target object.

Neural network architecture

The neural network utilized in our solution is a feed-forward network consisting of a convolutional part and a fully connected part. The convolutional part is used to extract features from image pairs that are then forwarded to the fully connected part of the network. The convolutional layers are divided in two identical branches, each handling a single input image. The outputs of the convolutional branches are passed through an adaptive max pooling layer and then concatenated. These concatenated outputs form a 100x100x128 tensor which is then flattened into a 1D tensor and is passed as the input into a fully connected part of the network. This part is composed of 6 layers with the PReLU activation function between the first five, and a ReLU activation after the final layer. The detailed preview of the neural network architecture is given in Fig. 1.

Data acquisition

The data used for training and validation of the neural network consists of 10000 image pairs. The images were generated from artificial 3D scenes. The 3D scenes were modelled using Blender, an open source 3D modelling software.

The benefits of generating images from artificial scenes are the ability to acquire a large dataset with exact distance measurements and camera and scene parameters under our control without the need for manual data labelling.

The scenes themselves consist of 20 objects of which 19 are cubes serving as noise and a target object, of varying shapes and dimensions, to which the distance is measured. A cuboid enveloping the aforementioned objects is used as background of the scene. The target distance, i.e. the label, is calculated as the Euclidean distance to the center the stereo camera baseline. The focal length of the camera lenses is 50 mm, clip start is 0.1 m and clip end is 100 m. The cameras are positioned into a parallel configuration and are always facing the center of the target object. The baseline is set to 6.5 cm. A setup of the scene pre-render is shown in Fig. 2.

The materials applied to the objects are Physically Based Rendering (PBR) materials, used in order to achieve a higher level of detail while maintaining a low poly count for objects. PBR materials, consisting of the following textures: albedo, metalness, roughness, ambient occlusion, bump and normal maps, were used in this scene to create specialized shaders which enabled the altering of specific material properties such as color, reflectivity and surface imperfections. The lighting of the scene was realized by using a high dynamic range imaging (HDRI) texture, which allows for a greater range of luminosity than can be achieved with the standard lighting model in Blender.

The Cycles render engine, which is a ray tracing engine, was used to generate the image pairs. The engine was configured to use 3 light ray bounces, which allowed for the creation of photorealistic images while significantly reducing render times. In order to further optimize rendering times, while maintaining image quality, a lower sample value was used along with an active denoising component.

Parameters of the scene are randomized before each image pair is rendered. First, the HDRI texture is randomly chosen from a pool of 12 textures containing examples of interior, exterior, day and night lighting conditions. Then, the camera is positioned in the scene by randomizing its y coordinate in the range of 4 m to the left of its origin to 4 m to the right of the origin. The camera's z coordinate is randomized in the range of 0.5 m to 5 m above the origin. The target object is placed along the x axis, its x coordinate ranging from 8 to 68 m from the origin, which is the initial x coordinate of the camera. The scale and rotation of the target object are randomized along all axes, the scale in the range of 0.5 m to 3 m and the rotation in the range of 0 to 2π. The shape of the target object is randomly chosen from a preset of 4 shapes: cube, sphere, cone, cylinder. The cubes used as noise are positioned from 5 to 10 m to the left and right of the x axis.

During the rendering process, a metadata file is formed. This file is composed of records describing each image pair. The records contain image names, the distance from the camera to the target object, and the coordinate of the point on the target object to which the distance is measured. The metadata file is used as the input into the training process.

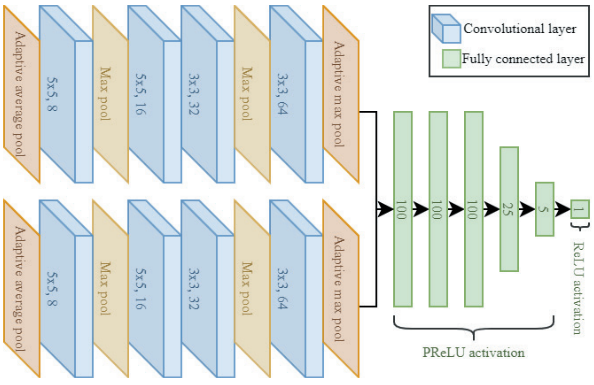

Fig. 1. Neural network architecture.

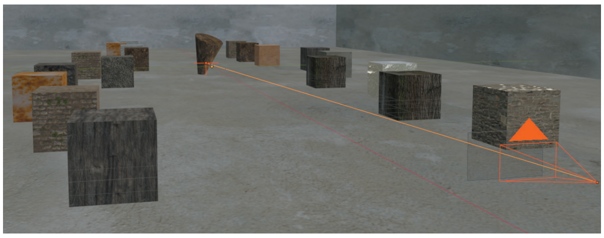

Fig. 2. Scene pre-render.

Training process

The neural network was trained using the stochastic gradient descent (SGD) algorithm. The learning rate of the algorithm was set to 1×10^{-5}, whilst the momentum was set to 0.9. Each training sample was composed of an image pair and the distance measured from the stereo camera to the target object. The training was done in batches, each batch contained 5 training samples. The batch size was chosen empirically with the goal of maximizing utilization of the graphics card and minimizing IO operations.

The dataset used to train the neural network was the previously acquired simple scene dataset. This dataset contained 10000 image pairs, divided into a training subset, containing 80% of the images and a validation subset containing 20% of the image pairs.

During training both the loss on the training dataset, and the average and median error on the validation dataset were monitored. The loss measure used during the training was the mean square error (MSE). The error measure used on the validation dataset is expressed as an absolute difference of expected and estimated value. Once the error measured on the validation dataset stopped changing significantly the training process was terminated. This happened after 12 epochs. The change of loss and validation error with respect to the number of epochs are shown in Fig. 3.

3. RESULTS

The neural network was tested on a separate dataset consisting of 2000 image pairs generated in the same manner as is described in II. These image pairs were not part of the training or validation subsets. Prior to testing, the neural network weights from the 12th checkpoint were loaded for the purpose of testing, as these scored the lowest error on the validation dataset. The mean absolute error on the test dataset was 1.59 m, while the median absolute error was 1.2 m. In order to facilitate further error analysis, the test results of every individual image were grouped into bins based on the value of the label, i.e. the real distance, in 10 m intervals. The mean and median absolute error values of each bin, as well as a boxplot of the errors per bin are shown in Table 1 and Fig. 4, respectively.

Detailed analysis of reported errors has shown that images generated from scenes using HDRIs with lower light levels have the largest difference between actual distance to object and the estimate. Similarly, a majority of objects sharing the same material as the background have shown to have a negative impact on results.

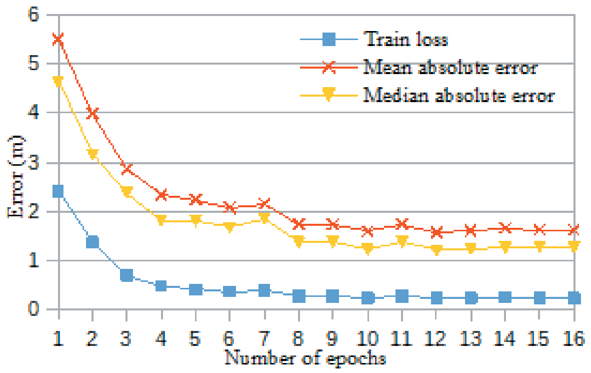

Fig. 3. Change of loss and validation error with respect to the number of epochs.

Table 1. MEAN AND MEDIAN ERRORS WITH RESPECT TO BINS

Bins	Errors (m)	
	Mean error	*Median error*
1	1.76	1.25
2	1.55	1.17
3	1.51	1.19
4	1.58	1.23

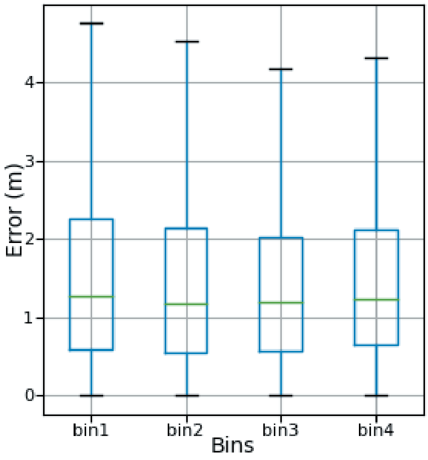

Fig. 4. Error distribution across bins.

4. CONCLUSION

Distance estimation from stereo images is a commonly used and well-known technique. While related works mainly focus on either disparity map generation or automatic camera calibration, the solution proposed in this paper unifies both into a single process in the form of a convolutional neural network. Stereo images were used as an input in order to estimate the distance to a target object in the images. The solution was trained, validated and tested using a dataset containing images generated from a 3D scene constructed using Blender. The training dataset consisted of 8000 image pairs, while the validation and testing datasets contained 2000 image pairs each. The neural network was trained for 12 epochs using SGD. Finally, the network was evaluated on the test dataset and achieved a mean absolute error of 1.59 m. Upon further examination of the results gathered from the evaluation, it was established that the network achieved similar performances across all distance ranges.

While the solution presented in this paper was shown to be a valid proof of concept, further research can be undertaken. One promising direction would be the construction of a more complex 3D scene for the generation of training data, as well as a larger dataset which would contain depth maps. Such a dataset could be used to train a depth map generating model or to improve the results of the solution presented in this paper. The viability of the solution for real-life application could be evaluated using manually labelled stereo images acquired with calibrated cameras. Transfer learning with the utilization of such images could prove as a promising direction for future research.

ACKNOWLEDGMENT

The authors acknowledge funding provided by the Science Fund of the Republic of Serbia #GRANT No. 6524105, AI – ATLAS.

REFERENCES

[1] T. Bosch, “Laser ranging: a critical review of usual techniques for distance measurement,” Opt. Eng., vol. 40, no. 1, p. 10, Jan. 2001, doi: 10.1117/1.1330700.

[2] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, “Weather Influence and Classification with Automotive Lidar Sensors,” 2019 IEEE Intell. Veh. Symp. IV, pp. 1527–1534, Jun. 2019, doi: 10.1109/IVS.2019.8814205.

[3] A. Carullo and M. Parvis, “An ultrasonic sensor for distance measurement in automotive applications,” IEEE Sens. J., vol. 1, no. 2, p. 143, 2001, doi: 10.1109/JSEN.2001.936931.

[4] A. Kilpelä, R. Pennala, and J. Kostamovaara, “Precise pulsed time-of-flight laser range finder for industrial distance measurements,” Rev. Sci. Instrum., vol. 72, no. 4, pp. 2197–2202, Apr. 2001, doi: 10.1063/1.1355268.

[5] S. Tuohy, D. O’Cualain, E. Jones, and M. Glavin, “Distance determination for an automobile environment using inverse perspective mapping in OpenCV,” pp. 100–105, Jan. 2010, doi: 10.1049/cp.2010.0495.

[6] S. Nedevschi et al., “High accuracy stereo vision system for far distance obstacle detection,” in IEEE Intelligent Vehicles Symposium, 2004, Jun. 2004, pp. 292–297, doi: 10.1109/IVS.2004.1336397.

[7] Y. Huang, S. Fu, and C. Thompson, “Stereovision-Based Object Segmentation for Automotive Applications,” EURASIP J. Adv. Signal Process., vol. 2005, no. 14, p. 910950, Dec. 2005, doi: 10.1155/ASP.2005.2322.

[8] B. Lim, T. Woo, and H. Kim, “Integration of Vehicle Detection and Distance Estimation using Stereo Vision for Real-Time AEB System:,” in Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems, Porto, Portugal, 2017, pp. 211–216, doi: 10.5220/0006296702110216.

[9] J. Zˇbontar and Y. LeCun, “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches,” p. 32.

[10] T. S. Jordan and S. Shridhar, “Usings CNNs to Estimate Depth from Stereo Imagery,” p. 6.

[11] M. Mendonça, “CAMERA CALIBRATION USING NEURAL NETWORKS,” p. 4.

[12] Yongtae Do, “Application of neural networks for stereo-camera calibration,” in IJCNN’99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), Washington, DC, USA, 1999, vol. 4, pp. 2719–2722, doi: 10.1109/IJCNN.1999.833509.

[13] M. T. Ahmed, E. E. Hemayed, and A. A. Farag, “Neurocalibration: a neural network that can tell camera calibration parameters,” in Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 463–468 vol.1, doi: 10.1109/ICCV.1999.791257.