International Journal of Engineering & Technology, 7(4.16) (2018) 234-237

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Stereo Matching Algorithm With Deep Learning Method Using Nvidia Platform

Masoud Samadi[1], Mohd Fauzi Othman[2]*, Akira Taguchi[3]

[1] Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 54100, Kuala Lumpur
[2] Tokyo City University, Tokyo 158-8557, Japan
**Corresponding author E-mail : othman@tcu.ac.jp*

Abstract

Autonomous vehicle has become a hot topic for researchers in recent years. One of the important sensors used in these vehicles is Stereo Cameras/Vision. Stereo vision systems are used to estimate the depth from the two cameras installed on the robots or vehicles. This method can deliver the 3D position of all objects captured in the scene at a lower cost and higher density compared to LIDAR. Recently, neural networks are vastly investigated and used in image processing problems and deep learning networks which has surpassed traditional computer vision methods specially in object recognition. In this paper, we propose to use a GPU with a new Siamese deep learning method to speed up the stereo matching algorithm. In this work, we use a high end Nvidia Platform DGX workstation to train and test our algorithm and compare the results with normal GPUs and CPUs. Based on numerical evaluation, the Nvidia DGX can train a neural network with higher input image resolution approximately 8 times faster than a normal GPU and 40 times faster than a Core i7 8 Cores CPU. Since it has the ability to train on a higher resolution the network can be trained in more iteration and results in higher accuracy.

Keywords: *LIDAR, Stereo Vision, Siamese Deep Neural Network, Nvidia Platform*

1. Introduction

For autonomous robots and vehicles to operate without the intervention of any human operators, they need to acquire the general knowledge about their surroundings. To obtain this information, they use several different sensors. Each sensor is responsible for providing some part of the required data.

In autonomous vehicle industries, cameras are responsible for object detection and recognition, while the depth estimation is done by LIDARs. LIDARs are responsible to detect the distance between the car and other objects in the environment. They are very accurate, but their result is not dense, and they are costly. To reduce the cost of autonomous robots and vehicles, researchers have tried to use cameras as the main depth estimation sensor [1]. Hence, stereo vision systems are invented. Stereo vision uses two cameras which are horizontally aligned for understanding the depth and distance (Figure 1).

This method has a number of advantages compared to other methods in obtaining 3D data from the environment. Conventional methods contain their own defects and difficulties [2]. Using a pair of cameras for image capturing brings the problem of correspondence between the images obtained.

Fig. 1: Stereo Vision Cameras

In general, matching the corresponding points captured by a pair of cameras is very difficult due to the similarities that exist in the images. By employing the stereo matching techniques, the robot can estimate the 3D position of any object in the visibility of its cameras by calculating the stereo disparity for that object. The stereo disparity is the difference between the locations of an object in the two images captured by the stereo cameras and is the result of the stereo matching function [3].

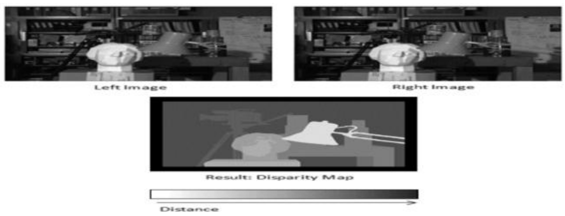

Fig. 2: Disparity Map Sample

To realize the 3D position of all the points in a pair of images from the stereo cameras, the dense disparity map must be computed. Figure 2, at the bottom, shows a disparity map that is calculated from the left and right images that are illustrated at the top of the figure. The lighter colors show closer locations to the cameras. It is necessary to mention that the results in Figure 2 constitute a manually made disparity map, which is regularly used by researchers for the purpose of the evaluation of their algorithms.

Stereo matching has been intensively investigated for several decades. Current stereo matching algorithms face problems in repetitive patterns, thin structures, reflective surfaces and textureless areas. Some stereo matching algorithms try to reduce these failures with gradient based regularization or pooling [4, 5]. However, this often requires a compromise between detecting detailed structures and smoothing surfaces.

On the other hand, deep learning models have been successful in learning powerful representations directly from the raw data in

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

object classification [6], detection [7] and semantic segmentation [8, 9].

Deep Learning networks are distinguished from the neural networks by their depth; that is, the number of node layers through which data passes in a multistep process of pattern recognition. Deep learning networks perform automatic feature extraction without human intervention, unlike most traditional machine-learning algorithms.

In Deep Learning, a Convolutional Neural Network (CNN) is a type of feedforward neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. These networks use a special architecture which is particularly well-adapted to classify images. Using this architecture makes convolutional networks fast to train. This, in turn, help to train deep, many-layer networks, which are very good at classifying images. Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. Unlike a regular Neural Network, the layers of a convolutional network have neurons arranged in 3 dimensions, width, height, depth. A typical Convolutional Neural Network architecture is displayed in Figure 3.

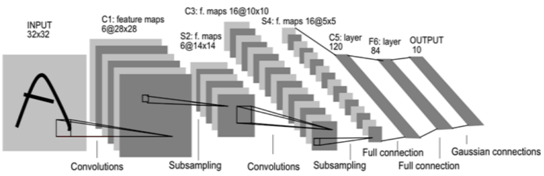

Fig. 3: Convolutional Neural Network

2. Related Works

In the recent years, CNN has been used to solve various problems in stereo matching. Zbontar and LeCun [10] trained a CNN to compute the matching cost between two 9x9 image patches, which is followed by several post-processing steps, including crossbased cost aggregation, semi-global matching, left-right constancy check, sub-pixel enhancement, median filtering and bilateral filtering. This architecture needs multiple forward passes to calculate matching cost at all possible disparities. Therefore, this method is computationally expensive.

A multi-scale embedding model from Chen et al. [11] also provided good local matching scores. Also noteworthy is the DeepStereo work of Flynn et al. [12], which learns a cost volume combined with a separate conditional color model to predict novel viewpoints in a multi-view stereo setting.

Park and Lee [13] introduced a pixelwise pyramid pooling scheme to enlarge the receptive field during the comparison of two input patches.

Mayer et al. created a large synthetic dataset to train a network for disparity estimation (as well as optical flow) [14], improving the state-of-the-art. As one variant of the network, a 1-D correlation was proposed along the disparity line which is a multiplicative approximation to the stereo cost volume. This is an encoder-decoder architecture for disparity regression. The matching cost calculation is seamlessly integrated to the encoder part. The disparity is directly regressed in a forward pass. Kendall et al. [15] used 3-D convolutions upon the matching costs to incorporate contextual information and introduced a differentiable "soft argmin" operation to regress the disparity.

The KITTI dataset [16, 17] is a large dataset from data collected from a moving vehicle with LIDAR ground truth. These datasets motivated the improvement hand-engineered techniques for all components of stereo, of which we mention a few notable examples.

All of the mentioned method employed CNN to generate a disparity map in a supervised manner. The KITTI dataset does not provide enough image to train a deep neural network, so the data augmentation technique is used to increase the number of training images for the neural network.

In our work, we propose Convolutional Neural Network with residual blocks and trained them using different platforms and tested the performance of the Nvidia DGX, Nvidia 1060 GTX and CPU and reported the findings.

3. Methodology

In this research, we use a Siamese neural network to estimate the depth. Throughout this paper, we assume that the image pairs are rectified, thus the horizontal image axis and epipolar lines are aligned. To estimate the depth, we use a Siamese architecture, where each branch processes the left or right image respectively.

Our goal is to minimize the discrepancy between the estimated disparity maps and the ground truth disparity maps, in this case LIDAR images from the KITTI dataset.

The first step in any image processing task is a step called feature extraction. Features are small, interesting, descriptive or informative patches in images. In traditional image processing method, this step is one of the most difficult and crucial part of the algorithm, where feature extraction is done manually to extract the best possible description in the image to be used in the next step of the algorithm. In deep learning method, the machine is told to learn what to look for with respect to each specific class of object. It works out for most descriptive and salient features for each object. In other words, neural networks are told to discover the underlying patterns in classes of images, by themselves. Therefore, with deep learning, there is no need to manually decide which traditional computer vision technique to use to describe your features. The machine works this all out itself. Additionally, this approach promises to reduce much of the engineering design complexity. The flowchart of the proposed deep learning algorithm is displayed in Figure 4.

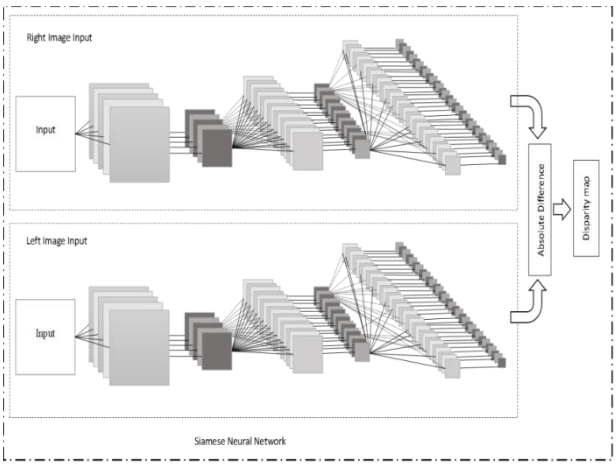

Fig. 4: Flowchart of the Proposed Siamese Deep Learning Algorithm

In the proposed method, the feature extraction is done by the aforementioned Siamese network. To elaborate in detail, each branch takes an image as input, and passes it through a set of layers, each consisting of a spatial convolution with a small filter-size, followed by the activation function. The first convolutional layer uses a 7x7 kernel followed by a 3x3 kernel. In this research, we use a rectified linear unit (ReLU) as the activation function. Each even layer is followed by maxpooling with stride 2, to reduce the image size, or as the researchers call it, downsizing. After 7 layers of 3x3 convolutions, the downsized feature goes through a series of deconvolution layers with stride 2 to upsample again, while using a residual connection, they are concatenated with the respective conventional layer to increase the feature extraction details.

One of the main problems of a Deep Neural Network is to propagate the error all the way to the first layer. For a deep network, the gradients keep getting smaller until it has no effect on the network weights. Residual connection was designed to overcome such problem (Figure 5), by defining a block with identity path. During back propagation, the gradients have a path that does not affect its magnitude. The network needs to learn residual mapping.

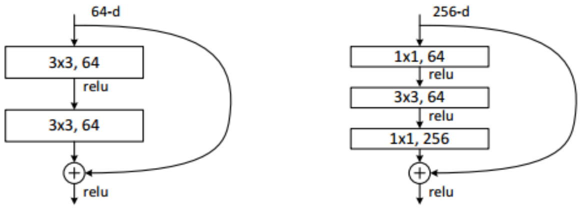

Fig. 5: Residual connection Sample

Table 1: The detailed architecture of each Siamese branch

name	Kernel	Stride	input layer
Conv1	9x9x16	1	Input Image
Conv2	3x3x16	1	Conv1
Maxpool1	3x3	2	Conv2
Conv3	3x3x64	1	Maxpool1
Conv4	3x3x64	1	Conv3
Maxpool2	3x3	2	Conv4
Conv5	9x9x128	1	Maxpool2
Conv6	3x3x128	1	Conv5
Maxpool3	3x3	2	Conv6
Conv7	3x3x256	1	Maxpool3
Conv8	3x3x256	1	Conv7
Deconv1	5x5x128	2	Cov8
Deconv2	5x5x64	2	Deconv1+ Conv6
Deconv3	5x5x16	2	Deconv2+ Conv4
Deconv4	3x3x1	2	Deconv3+ Conv2

The result of the two networks are the high descriptive features from the left and right images. These features are treated like normal images to calculate the disparity map. Inspired by a traditional SAD (Sum of Absolute Differences) method, the two outputs are subtracted and the result used as the generated disparity map [18].
In the loss function, we would like to minimize the discrepancy between the ground truth and the generated disparity map. It can be done by forming a loss, by simply computing the L1 distance between the images and their gradients;

$$L(d_l, d'_l) = \frac{1}{N}\sum \lambda_1 |d_l - d'_l| + \lambda_2 |\nabla \mathrm{d}_l - \nabla \mathrm{d}'_l|$$

where N is the total number of pixels, d_l is the ground truth disparity, d'_l is the predicted disparity map and ∇ is image gradient. λ_1 *and* λ_2 are used as coefficient to balance the absoluter difference and the gradient difference.

4. Results and Discussion

In this section the results of the evaluation of the proposed model on the KITTI dataset is reported and the speed comparison between the different dataset is tabulated in Table 2. The network is developed using TensorFlow library [20] and it trained for 20000 epochs of size 50. The network could achieve the error rate of 2.41 in 5 pixel error rate evaluation method which is slightly better than the previous models with error rate of 2.53. The result and LIDAR ground truth are shown in Figures 6 and 7.
Please note that the network was trained with a larger size on Nvidia DGX workstation, because the other platform did not support the bigger image size.

Fig. 6: Up: The left camera image, the right camera image The LIDAR (ground Truth) Predicted Disparity map by the proposed method

Table 2 Speed Comparison between different platform

Platform	Image Dimensions	Time (ms)
Nvidia DGX	1280x384	0.16
Nvidia 1060 GTX	960x192	1.30
Intel Core i7 4870	960x192	6.40

Fig. 7: Up: The left camera image, The right camera image The LIDAR (ground Truth) Predicted Disparity map by the proposed method

5. Conclusion

In this study, we used a Siamese convolutional neural network and trained it using different platforms. Based on the observation, the Nvidia DGX can train a neural network with higher input image resolution approximately 8 times faster than Nvidia 1060 GTX GPU and 40 times faster than a Core i7 8 Cores CPU.

Acknowledgement

We would like to thank the Center for Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia (UTM) and Tokyo City University (TCU) for providing the research facilities.

References

[1] Ben-Tzvi, P. and Xu, X. An embedded feature-based stereo vision system for autonomous mobile robots. Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on. 2010. 1 –6. DOI : http://dx.doi.org/10.1109/ROSE.2010. 5675303.
[2] Wang, L., Liao, M., Gong, M., Yang, R. and Nister, D. High-Quality Real-Time Stereo Using Adaptive Cost Aggregation and Dynamic Programming. Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission. Washington, DC, USA: IEEE Computer Society. DOI : https://doi.org/10.1109/3DPVT.2006.75
[3] Samadi M., Othman M.F. (2013) A New Fast and Robust Stereo Matching Algorithm for Robotic Systems. Advances in Intelligent Systems and Computing, vol 209. Springer, Berlin, Heidelberg DOI : https://doi.org/10.1007/978-3-642-37371-8_31
[4] Geiger A., Roser M., Urtasun R. (2011) Efficient Large-Scale Stereo Matching. In: Kimmel R., Klette R., Sugimoto A. (eds) Computer Vision – ACCV 2010. ACCV 2010. Lecture Notes in Computer Science, vol 6492. Springer, Berlin, Heidelberg. DOI : https://doi.org/10.1007/978-3-642-19315-6_3

[5] H. Hirschmuller. Accurate and efficient stereo processing by semiglobal matching and mutual information. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 807–814. DOI : http://dx.doi.org/10.1109/CVPR.2005.56

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. DOI : http://dl.acm.org/citation.cfm?id=2999134.2999257

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014. DOI : https://dblp.org/rec/bib/journals/corr/GirshickDDM13

[8] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In Proceedings of the IEEE International Conference on Computer Vision, pages 972–980, 2015. DOI : http://dx.doi.org/10.1109/ICCV.2015.117

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. DOI : https://doi.org/10.1109/TPAMI.2016.2644615

[10] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research, 17. DOI : https://dblp.org/rec/bib/journals/corr/ZbontarL15

[11] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In Proceedings of the IEEE International Conference on Computer Vision, pages 972–980, 2016. DOI : http://dx.doi.org/10.1109/ICCV.2015.117

[12] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. DeepStereo: Learning to Predict New Views from the World's Imagery. DOI : https://dblp.org/rec/bib/journals/corr/FlynnNPS15

[13] H. Park and K. M. Lee. Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters, PP(99):11, 2017. DOI : http://dx.doi.org/10.1109/LSP.2016.2637355

[14] N. Mayer, E. Ilg, P. H¨ausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. CoRR, abs/1510.0(2002), 2015. DOI : http://dx.doi.org/10.1109/CVPR.2016.438

[15] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In IEEE Conference on Computer Vision and Pattern Recognition. DOI : http://arxiv.org/abs/1703.04309

[16] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361. DOI : http://dx.doi.org/10.11113/jt.v79.9987

[17] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In Conference on Computer Vision and Pattern Recognition (CVPR). DOI : http://dx.doi.org/10.1109/CVPR.2015.7298925

[18] Samadi M., Othman M.F., Talib M.F. 2016. Fast and Robust Stereo Matching Algorithm For Obstacle Detection In Robotic Vision Systems. 6-13, Jurnal Teknologi DOI: https://doi.org/10.11113/jt.v78.92

[19] Mahammed M. A., Melhum A. I., Kochery F.A. 2013. Object Distance Measurement by Stereo VISION. International Journal of Science and Applied Information Technology (IJSAIT), Vol.2, No.2, Pages: 05-08

[20] https://www.tensorflow.org/, TensorFlow opensource deep learning library, Retrieved 15 October 2018.