Disparity map estimation with deep learning in stereo vision

Mendoza Guzmán Víctor Manuel[1], Mejía Muñoz José Manuel[1], Moreno Márquez Nayeli Edith[1], Rodríguez Azar Paula Ivone[1], and Santiago Ramírez Everardo[1]

Universidad Autnoma de Ciudad Jurez @alumnos.uacj.mx
http://www.uacj.mx/Paginas/Default.aspx

Abstract. In this paper, we present a method for disparity map estimation from a rectified stereo image pair. We proposed, a new neural network architecture based on convolutional layers to predict the depth from the stereo vision images. The Middlebury datasets were used to train the network with a known disparity map in order to compare the error of the estimated map.

Keywords: Neural networks · stereo vision · disparity map.

1 Introduction

One of the techniques that have shown potential for obtaining three-dimensional (3D) information from two-dimensional (2D) images is the processing of stereo images [1]. Images are commonly considered as 2D representations of the real world (3D). Stereo vision by computer involves the acquisition of images with two or more cameras moved horizontally to each other. In this way, different views of a scene are recorded and can be processed for different applications such as vehicle tracking [4], aircraft estimation and positioning [5], and automatic adaptation systems that cover a wide range of applications [6], including 3D reconstruction or disparity map estimation.

Stereo vision tries to imitate the mechanisms that are made in the human visual system and the human brain. A scene depicted with two horizontally displaced cameras will get two slightly different projections of a scene. If these two images are compared, additional information can be reached, such as the depth of a scene. This process of extracting the three-dimensional structure of a scene from pairs of stereo images is called computational stereo, and the resul is generally a disparity map which is a map of the depth or distance at which the objects of a scene are located [2].

In recent years, numerous algorithms and applications for the estimation of disparity maps have been presented. In [13] a method based on satellite images is proposed to monitor trees and vegetation. The stereo matching algorithms

are calculated to measure the disparity map based on stereo satellite images. The estimation of the height of trees and vegetation near the base poles to the depth map is inversely proportional to the disparity map. In [14] another application for pedestrian detection is proposed based on the dense disparity map for smart vehicles. The dense disparity map is used to improve pedestrian detection performance. The method consists of several steps, detection of obstacle areas using information of characteristics of roads and detection of columns, detection of pedestrian areas using a segmentation based on dense disparity maps and detection of pedestrians using the optimum characteristic.

The work of [15] is a investigation for object tracking were a stereoscopic camera is used to detect objects, which makes it a low-cost solution for tracking objects. Its objective is to detect objects in a video sequence and track them throughout the video without prior knowledge about the objects. Calculate the disparity map using a pair of stereophonic images. Then, the disparity map is subjected to a depth-based segmentation to detect object blobs and the corresponding region in rectified stereo-image is the object of interest.

In [16] an efficient algorithm is presented to optimize the performance of a stereoscopic vision system and accurately relate the calculated disparity map with the real depth information.

With the new technologies and artificial intelligence in [17] a methodology for the detection of robust obstacles in outdoor scenes for autonomous driving applications is proposed using a multi-value stereo disparity approach. The disparity computation suffers a lot from reflections, lack of texture and repetitive patterns of objects. This can lead to incorrect estimates, which may introduce some bias in the obstacle detection approaches that make use of the disparity map. To overcome this problem, instead of a disparity estimate of a single value, a new research that uses a diversity of candidates for each point of the image is proposed. These are selected based on a statistical study characterized by the performance of different parameters: number of candidates and the distance between them compared to the real value of the disparity. It continues creating a location map from which the estimation of the obstacles is obtained.

In [18] explain different phases to perform the depth estimation: First they perform an extraction of characteristics, an initial estimation and a final refinement of the estimated depth.

Obtain the main characteristics of the two input images, stereo images left and right, and the final refinement is done in a main block of the network with two refinement sub-networks. The network contains a pair of convolution layers and a pair of deconvolution layers to perform the sampling at the output.

This subnetwork structure generates a depth map through an architecture that encodes and decodes information, inspired by the DispNetCorr1D network [19].

The innovative aspect of the research in [20] is that for the first stage the network described includes modules to perform deconvolution in addition to the traditional which leads to estimate the disparity with the same size as the images that are being used for the input. In the second stage, which is refinement, they propose residual learning used in [21].

In [22] propose a new optimization method that uses strong smoothing restrictions obtained in a neural network. The goal for this is to soften the output disparity map in a robust manner. The first step in this research was to define the CNN architecture, called DD-CNN, to classify if the disparities are discontinuous. The training of this architecture was carried out with real data from Middlebury stereo data [23]. In the next step they define an energy function composed of a term of data obtained with the method of [24] and a term that penalizes disparity differences.

Finally in [25] a network is developed with a dual structure. Each of the structures takes an input image that passes through a finite set of layers followed by a normalization and a rectified linear unit. In their experiments, different filters were tested per layer and the parameters were shared between the two structures. For the training, small kernels of the images extracted from a set of pixels were randomly used. Providing a diverse set of examples and it was considered an efficient method in memory.

For this research it is proposed a new architecture based on convolutional networks, for the training of the network we will use the stereo images from the Middlebury database [3] and the disparity map results obtained in [7].

2 Theory

2.1 Convolutional Network

Convolutional networks are composed of a specific type of connections for data processing that have known properties. Examples can be highlighted from time series, which is considered a grid of one dimension with a regular time defined, and images, which is the same case as time series with more than one dimension ordered at specific points. CNN (Convolutional Neuronal Network) has demonstrate a high level of success in field applications. It is called the convolutional network because it performs the mathematical operation called convolution within neurons. Convolution is a specialized type of linear operation. Convolutional networks are simply neural networks that use convolution instead of the general multiplication of matrices in at least one of their layers.

In its most general form, convolution is an operation in two functions of an argument of real value.

In machine learning applications, the input is usually a multidimensional array of data, and the core is usually a multidimensional array of parameters that are adapted by the learning algorithm.

The convolution takes advantage of three important ideas that can help to improve a learning system: dispersed interactions, shared use of parameters and equivalent representations. In addition, convolution provides a means to work with entries of variable size. Traditional neural network layers use matrix multiplication through a parameter matrix with a separate parameter that describes the interaction between each input unit and each output unit. However, convolutional networks often have scattered interactions. This is achieved by making the kernel smaller than the input.

CNN are usually developed in the following stages: First, a defined number of convolutions are made at the same time to produce a group of linear activations. In the next stage, each activation is executed with an activation function that is not linear. This stage could be defined as the detector stage. The third stage uses a grouping function with the objective of modifying the output of each layer.

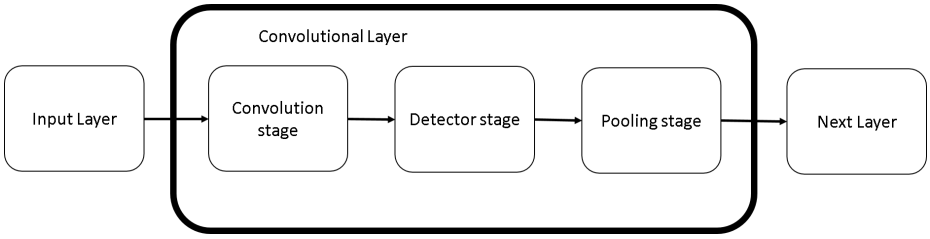

Fig. 1. Components of a typical convolutional network

A grouping function replaces the output of the network at a given location with a summary statistic of the nearby outputs.

In all cases, the grouping helps to make the representation almost invariant for small translations of the entry. The invariance to translation means that, if we translate the entry by a small amount, the values of most of the grouped outputs do not change [10].

3 Proposed architecture

The proposed convolutional network can be seen in Figure 2, the architecture of the network is as follows: The inputs, pair of stereo images, are individually processed through 3 convolutional layers, the first 2-D convolutional layer is 32 filters with a $3x3$ kernel returning the same size of the images. The output of the first layer continue with a MaxPooling with a size of 2, followed with another convolutional layer of 62 filters with a $3x3$ kernel and the output with a MaxPoling also with a size of 2 and finally a convolutional layer with a size of 92 filters and a $3x3$ kernel is applied to obtain an image in its original size of $96x96$. This process is applied independently to each image, the outputs are combined to apply another 3 convolutional layers.

The first convolutional layer has 62 filters with a $3x3$ kernel followed by a UpSampling with a size of $2x2$. At the exit, another convolutional layer with a size of 22 filters and a $3x3$ kernel is applied to the output and another UpSampling with a size of $2x2$, finally the last convolutional layer of 1 filter and a $2x2$ kernel is applied. All the activation functions used in the convolutional layers is the rectified linear unit (ReLU).

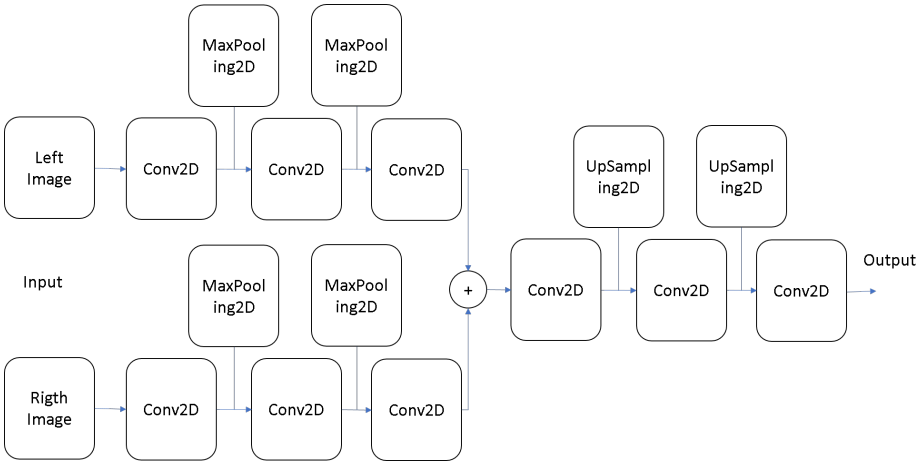

Fig. 2. Architecture of the convolutional network used in this research

3.1 Data preprocessing

In this research we used the images of the database of Middlebury [3]. The images of the database are in color, and consist of a pair of stereo images as show in Figure 3. The original database images were pre-processed, a change was made in the size of the images in such a way that all the images had a size of 92x92,

the main idea of leaving the square images is to speed up the computations and to lower memory requirements.

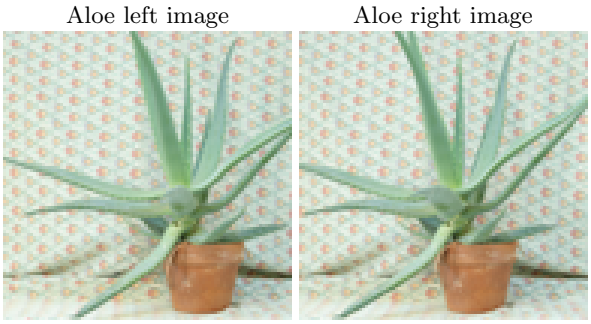

Fig. 3. Pair of images, left and right, from the database

Since the neural networks need a large amounts of data to work effectively, data augmentation was used to increase the number of images in the data set. For data augmentation, operations of translation, rotation, and scaling were used to increase the database to 500 images.

3.2 Metrics

The metrics that will be used for the evaluation of this research will be the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). These metrics have been used as a reference point for the comparison of input images and output images in the evaluation of image quality. The PSNR uses the Mean Square Error (MSE), the MSE is calculated between the average of the original intensity and the intensity of the output image and is given by:

$$MSE = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m,n)^2 \tag{1}$$

Where $e(m,n)$ is the difference of the error between the original image and the output image PSNR is the mathematical measure of image quality based on the pixel difference of two images. And it is defined by:

$$PSNR = 10log\frac{s^2}{MSE} \tag{2}$$

Where $s = 255$ for an 8-bit image [11].

For the SSIM, Wang[12], proposed the Structural Similarity Index as an improvement of the Universal Image Quality Index (UIQI). The SSIM is calculated as follows.

The Input and output images are divided into blocks then the blocks are converted into vectors, two means, two standard derivations and one covariance value are computed from the images.

Then the luminance, contrast, and structure comparisons based on statistical values are computed, the structural similarity index measure is given by:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x + \mu_y + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

Where $\mu_x\mu_y$ denotes the mean values of original and distorted images. And $\sigma_x\sigma y$ denotes the standard deviation of original and distorted images, and σ_{xy} is the covariance of both images, c_1 and c_2 are constants. [11].

The image quality MSSIM is obtained by calculating the mean of SSIM values given by:

$$MSSIM = \frac{1}{P}\sum_{j=1}^{P} SSIM_j \tag{4}$$

Where p is the number of sliding windows [11].

4 Results

Experiments were made using a computer with microprocessor Intel (R) Core (TM) i5-2410M of 2.30 GHz, 8GB of RAM GPU NVIDIA CUDA GeForce 315M with a CNN training of 8 to 16 hours.

For the evaluation of our convolutional neuronal network, the following images were used: a) Bowling fig 4, b) Midd fig 5, c) Lamp fig 6, d) Monopoly fig 7 and e) Baby fig 8. The left image and the right image were used as input for each estimation of the disparity map.

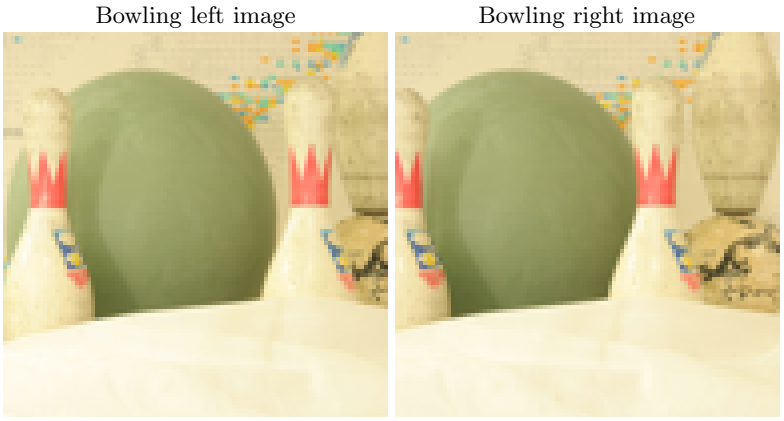

Fig. 4. Images to test the CNN Bowling from the database

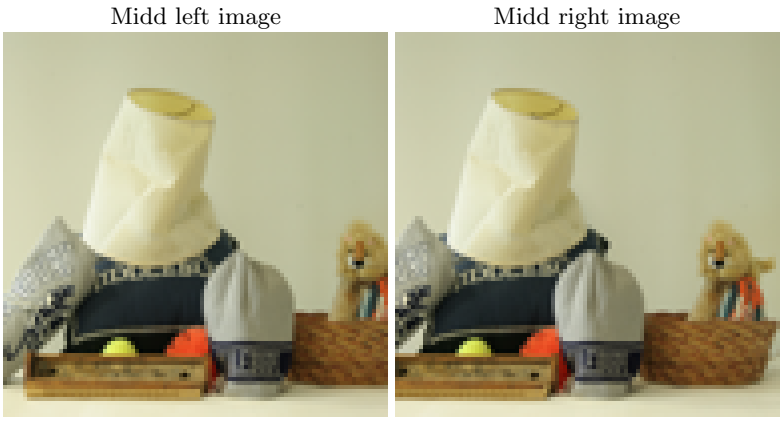

Fig. 5. Images to test the CNN Midd from the database

Lamp left image Lamp right image

Fig. 6. Images to test the CNN Lamp from the database

Monopoly left image Monopoly right image

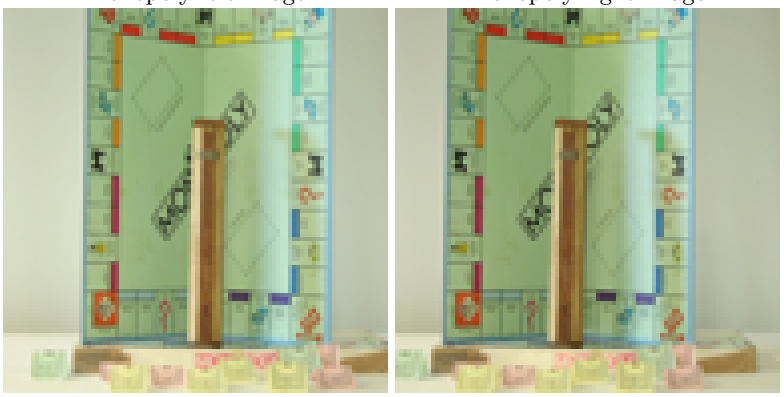

Fig. 7. Images to test the CNN Monopoly from the database

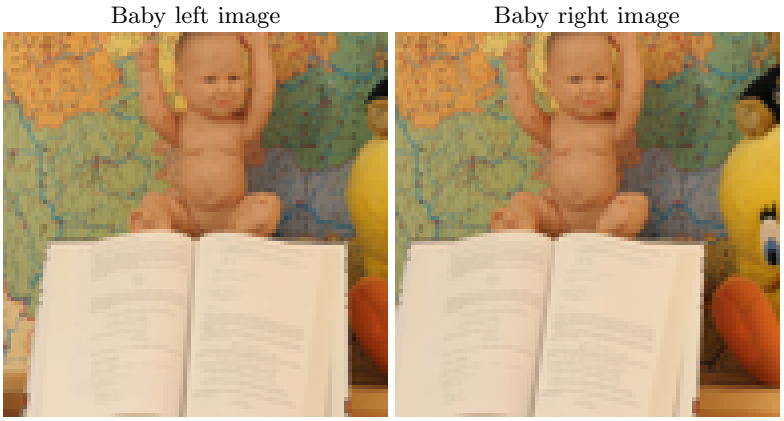

Fig. 8. Images to test the CNN Baby from the database

In the table 1 the PSNR and SSIM results are shown between the original disparity map and the estimated disparity Map with our convolutional neuronal network. With these values it can be seen that PSNR values are not the most optimal but with the SSIM it shows optimal similarity values between the images.

Table 1. Results for the test images

	PSNR	SSIM
Bowling	57.3966977	0.93
Midd	57.5238763	0.83
Lamp	57.7759308	0.91
Monopoly	58.7132492	0.82
Baby	60.440848	0.92

In the images 9, 10, 11, 12 and 13 it can be clearly seen how the convolutional neural network performed in the estimation of the disparity map. The output images of our network show low definition of the edges with respect to the original disparity map, however the accuracy of the estimate has an acceptable level.

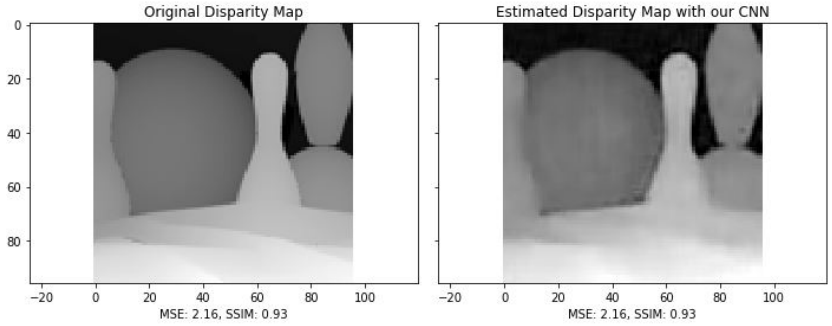

Fig. 9. Bowling disparity map original and estimated

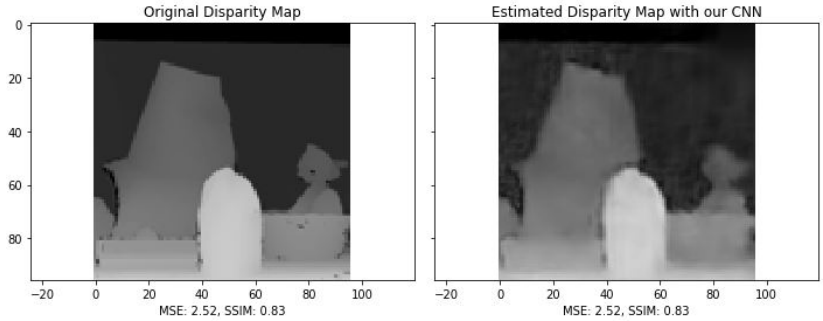

Fig. 10. Midd disparity map original and estimated

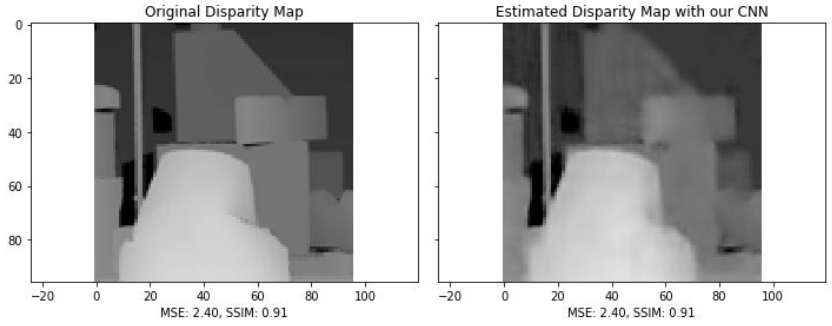

Fig. 11. Lamp disparity map original and estimated

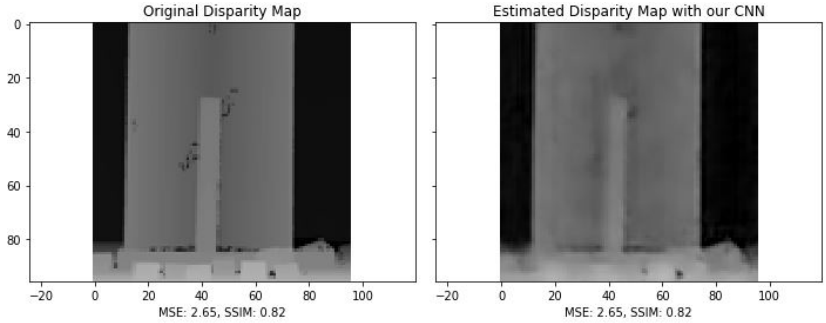

Fig. 12. Monopoly disparity map original and estimated

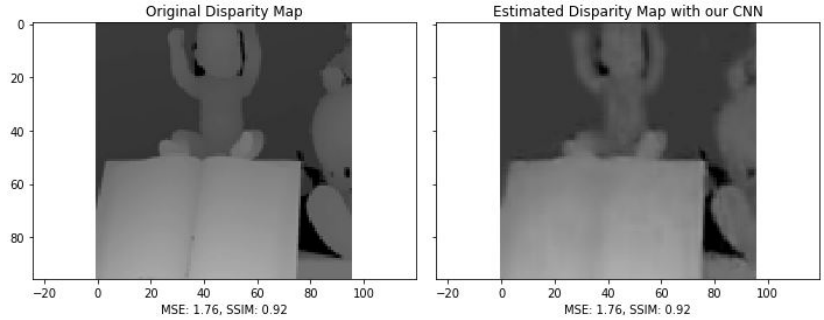

Fig. 13. Baby disparity map original and estimated

5 Conclusions

In this research, it was demonstrated how a new architecture of a convolutional network can estimate the disparity map between stereo images. With the obtained results it can be observed how a post processing of the output images could help the definition of the edges in the images, which seems to be the main problem to be solved as a next step in the investigation in order to obtain results more precise.

The limitation of hardware was another problem for this research, the training times of the convolutional neuronal network was from 8 to 16 hours. In addition, it can be concluded how applications for stereo vision systems can be solved by convolutional neural networks, as future work we plan to apply this neural network to stereo vision in real time video for obstacles detection to continue searching applications in stereo vision systems.

References

1. "Encyclopedia of Science and Technology", McGraw-Hill, 2009, pp. 594-596
2. M. Z. Brown, D. Burschka, G. D. Hager, "Advances in Computational Stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, 2003, pp. 993-1008
3. D. Scharstein, R. Szeliski, Middlebury Stereo Vision Page, available: http://vision.middlebury.edu/stereo
4. R. Canals, A. Roussel, J. L. Famechon and S. Treuillet, A biprocessor-oriented visionbased target tracking system," in IEEE Transactions on Industrial Electronics, vol. 49, no. 2, pp. 500-506, Apr 2002.
5. J. Zhang, Y. Wu, W. Liu and X. Chen, Approach to Position and Orientation Estimation in Vision-Based UAV Navigation," in IEEE Transactions on Aerospace and Electronic Systems, vol. 46, no. 2, pp. 687-700, April 2010.
6. S. E. Shih and W. H. Tsai, "A Two-Omni-Camera Stereo Vision System With an Automatic Adaptation Capability to Any System Setup for 3-D Vision Applications," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 7, pp. 1156-1169, July 2013.
7. D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.
8. D. Scharstein and C. Pal. Learning conditional random fields for stereo. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, June 2007.
9. H. Hirschmller and D. Scharstein. Evaluation of cost functions for stereo matching. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, June 2007
10. I.Goodfellow, Y. Bengio and A. Courville. Deep Learning MIT Press http://www.deeplearningbook.org, 2016
11. A. Yusra, Y. Al-Najjar and D. C. Soong. Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI. International Journal of Scientific and Engineering Research, August 2012
12. B. Zhou Wang, A Universal Image Quality Index, IEEE Signal Processing Letters, vol. 9, pp. 81-84, 2002.
13. A.Qayyum, A. Malik , M. Naufal B. Muhammad Saad, F. Abdullah and M. Iqbal, Disparity Map Estimation Based on Optimization Algorithms using Satellite Stereo Imagery, IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 127-132, 2015.
14. L. Chung-Hee and K. Dongyoung, Dense Disparity Map-based Pedestrian Detection for Intelligent Vehicle, IEEE International Conference on Intelligent Transportation Engineering, pp. 1015-1018, 2017.
15. A. K. Wasim, R. P Dibakar, A. Bhisma and M. Rasana, 3D Object Tracking Using DIsparity Map, International Conference on Computing Communication and Automation (ICCA2017), pp. 108-111, 2016.
16. J. Du and J. Okae, Optimization of Stereo Vision Depth Estimation using Edge-Based Disparity Map, 10th International Conference on Electrical and Electronics Engineering, pp. 1171-1175, 2017.
17. Q. Ge and E. Lobaton, Obstacle Detection in Outdoor Scenes based on Multi-Valued Stereo Disparity Maps, IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2017.

18. Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, J. Zhang, Learning for Disparity Estimation through Feature Constancy, CVPR 2018.
19. N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In IEEE Conference on Computer Vision and Pattern Recognition, pp. 4040-4048, 2016.
20. J. Pang, W. Sun, J. SJ. Ren, C. Yang, Q. Yan, Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching, ICCVW 2017.
21. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770778, 2016.
22. G. Song, H. Zheng, Q. Wang, and Z. Su. 2017. Training a Convolutional Neural Network for Disparity Optimization in Stereo Matching. In Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics (ICCBB 2017). ACM, New York, NY, USA, pp. 48-52.
23. Scharstein, D., Hirschmller, H., Kitajima, Y., Krathwohl, G., Nei, N., Wang, X., and Westling, P. 2014. High resolution stereo datasets with subpixel-accurate ground truth. Proc. German Conf. Pattern Recognit. (GCPR). 31-42, (Jan. 2014).
24. Zbontar, J. and LeCun, Y. 2015. Stereo matching by training a convolutional neural network to compare image patches. arXiv: 1510.05970.
25. W. Luo, A. G. Schwing and R. Urtasun, Efficient Deep Learning for Stereo Matching, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 5695-5703.