Comparison of Semi-Global Block Matching Algorithm and DispNet Neural Network on KITTI Data Set

Tin Kramberger[1], **Božidar Potočnik**[2]

[1]*Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, R Hrvaška*
[2]*Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor*
E-mail: tin@tvz.hr, bozidar.potocnik@um.si

Abstract

Disparity estimation is a challenging task with numerous real-world applications. There are two main approaches to this problem: the classic Semi-global block matching algorithm and a new approach using a trained convolutional neural network to estimate the disparity. This paper shows the advancement of disparity estimation in terms of accuracy. The accuracy of disparity estimation by using the Semi-global block matching algorithm (SGM) and trained DispNet convolutional neural network are assessed and compared in this paper. Results on the KITTI test data set show better performance of the DispNet neural network in terms of accuracy compared to SGM. It could be said that neural networks are taking the primacy by disparity estimation.

1 Introduction

Stereo reconstruction is one of the most significant research topics in the area of computer vision. To reconstruct a scene from a stereo camera, one first needs to create a disparity map. The disparity estimation is a technique which converts a pair or more images into a depth map to estimate the distance from a stereo camera. It is used in the 3D reconstruction, depth prediction, autonomous driving, scene understanding, etc. In the past, when hardware was somewhat lacking computational power, different mathematical approaches were used. They included three steps: feature extraction, matching cost aggregation, and disparity computation [1], [2]. The Semi-global block matching algorithm (SGM) [1] is often ambiguous and offers wrong matches due to reflection, noise, occlusions etc. Nowadays, as computer hardware is rapidly evolving, the hardware is becoming less and less an obstacle for researchers. Accordingly, the use of deep convolutional neural networks becomes much more convenient. The training of the neural network is computationally still the most expensive part of the process [3]. Deep neural networks are often used to generate disparity maps using additional components that do not belong to the domain of neural networks. Often, classical processing components are used to improve the performance of neural networks for a specific task in hand, e.g. GA-Net [4]. There are also deep convolutional neural networks that use non domain specific layers that were specifically created for the disparity map problem such as DispNet [5], [6].

This paper compares a classical Semi-global block matching algorithm (SGM) and the DispNet neural network by disparity map estimation. An accuracy of obtained disparity map by each method is assessed and compared. The KITTI dataset [6], [7], [8] was chosen for benchmarking SGM and DispNet due to the acknowledgement in research community and a decent development kit.

2 Semi-Global matching method

The Semi-Global matching method (SGM) is based on pixelwise matching of Mutual Information and approximation of a global, 2D smoothness constraint by combining many 1D constraints. Input images have to have a known epipolar geometry, but it is not required for them to be rectified [1]. SGM algorithm determines the disparity using a procedure that first computes match cost, then the aggregation cost, and, finally, the disparity.

Match cost is computed for base pixel p from its intensity I_{bp} and presumed correspondence I_{mq} with pixel $q = ebm(p, d)$ in matched images. The $ebm(p, d)$ function symbolizes the epipolar line, where p is a pixel and d is a line. The pixelwise cost can be computed in two different ways: using the Birdfitch and Tomasi method and the Mutual Information method (MI) [9], [10]. Birdfitch and Tomasi compute cost $CBT(p, d)$ as the absolute minimum of intensity difference between pixels p and $q = ebm(p, d)$ in a range of half pixel in each direction of an epipolar line.

MI cost is not susceptible to illumination changes and recording. This cost is defined as an entropy of two images and their mutual entropy (1). The Equation (1) works on full images and, theoretically, requires the disparity image [1].

$$MI_{I_1,I_2} = H_{I_1} + H_{I_2} - H \tag{1}$$

Entropy can be calculated from the probability distributions P of colour intensities i of the associated images which can be perceived from Equations (2) and (3). Intensity refers to the amount of light or the numerical value of a pixel.

$$H_I = -\int_0^1 P_I(i) \log P_I(i) di \tag{2}$$

$$H_{I_1,I_2} = -\int_0^1 \int_0^1 P_{I_1,I_2}(i_1,i_2) \log P_{I_1,I_2}(i_1,i_2) di_1 di_2 \quad (3)$$

For well registered images of the same scene, the joint entropy H_{I_1,I_2} in Equation (3) is low, because one image can be predicted by another image resulting in a low information that increases Mutual Information (MI). In the Equations (3) and (4), P_{I_1,I_2} is the joint probability distribution of corresponding intensities . In Equations (4) and (7), the operator T is defined in a way that if its argument is true it returns 1, and 0 otherwise.

$$P_{I_1,I_2}(i,\ k) = \frac{1}{n} \sum_p T[(i,\ k)\ =\ (I_{1p},\ I_{2p})] \quad (4)$$

Kim et al. [11] used the Taylor expansion to transform the calculation of the joint entropy H_{I_1,I_2} into sum over pixels (see Equation (5)). The entropy calculation is implemented using convolution in Equation (6), where h_{I_1,I_2} is calculated from the joint probability distribution using Equation (4). The number of corresponding pixels is n, and a convolution using 2D Gaussian is denoted by $\otimes g(i,k)$.

$$H_{I_1,I_2} = \sum_p h_{I_1,I_2}(I_{1p},\ I_{2p}) \quad (5)$$

$$h_{I_1,I_2}(i,\ k)\ = -\frac{1}{n} log(P_{I_1,I_2}(i,k) \otimes\ g(i,k)) \otimes\ g(i,k) \quad (6)$$

After computing match cost, the aggregation cost must be computed. The SGM algorithm in contrast to the classic block matching (BM) algorithm, in addition to the pixel-wise cost calculation also adds a smoothness constraint which penalizes changes at neighbour's disparities. Pixel-wise cost and the smoothness constraint are described in the Equation (7). The first sum in this equation is the sum of all matching pixels, the second sum adds a constant penalty P_1 for all pixels q in the neighbourhood N_p of p for which the change of disparities is low. The third sum adds a larger constant penalty P_2 for larger disparity changes [1].

$$E(D) = \sum_p (C(p,D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \quad (7)$$

The disparity map is determined by using local stereo methods in a way that for each pixel p the disparity d, which is a minimum cost, must be calculated. The process can be seen on Figure 1.

3 DispNet neural network

The DispNet was created by Mayer et al. and is a classic convolutional neural network without any additional filters in its architecture [6]. For this research, a pretrained DispNet neural network was used. It was trained and fine-tuned on 400 stereo pairs with ground truth disparities from the KITTI 2012 and 2015 training data sets [7], [12]. The network was trained using the Adam optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$ with a learning rate $\lambda = 1e-4$ which was divided by 2 every 200 000 iterations starting from iteration 400 000 [13]. Mayer et al. claim that a loss weight schedule is beneficial and started training with a loss weight of 1 assigned to the lowest resolution (loss6) and weight 0 for other losses. During training the weights of losses with higher resolutions were increased, and the weights of lower resolutions were deactivated. Spatial transformations and chromatic transformations were performed for data augmentation on the flow data set, and for the disparity data set only chromatic transformations were performed due to possible breakage of the epipolar constraint [6].

Table 1: Specification of DispNet architecture

Name	Kernel	Str.	Input
conv1	7x7	2	Images
conv2	5x5	2	conv1
conv3a	5x5	2	conv2
conv3b	3x3	1	conv3a
conv4a	3x3	2	conv3b
conv4b	3x3	1	conv4a
conv5a	3x3	2	conv4b
conv5b	3x3	1	conv5a
conv6a	3x3	2	conv5b
conv6b	3x3	1	conv6a
pr6+loss6	3x3	1	conv6b
upconv5	4x4	2	conv6b
iconv5	3x3	1	upconv5+pr6+conv5b
pr5+loss5	3x3	1	iconv5
upconv4	4x4	2	iconv5
iconv4	3x3	1	upconv4+pr5+conv4b
pr4+loss4	3x3	1	iconv4
upconv3	4x4	2	iconv4
iconv3	3x3	1	upconv3+pr4+conv3b
pr3+loss3	3x3	1	iconv3
upconv2	4x4	2	iconv3
iconv2	3x3	1	upconv2+pr3+conv2
pr2+loss2	3x3	1	iconv2
upconv1	4x4	2	iconv2
iconv1	3x3	1	upconv1+pr2+conv1
pr1+loss1	3x3	1	iconv1

The DispNet convolutional neural network was chosen for this work because it does not have additional layers that were specifically developed for stereo vision problems and could add a better performance to the network [4]. Table 1 presents the DispNet architecture.

4 KITTI Stereo Evaluation 2015 data set

This data set consists of 200 training scenes and 200 test scenes with 4 colour images per scene saved in a lossless PNG format. The ground truth has been established with a semi-automatic process. The evaluation framework written in MATLAB comes with a development kit

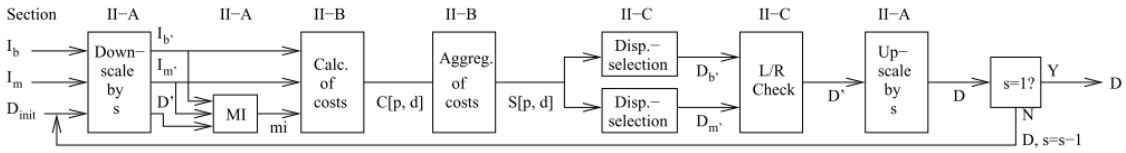

Figure 1: Flowchart of SGM [1].

that computes the percentage of faulty pixels averaged over ground truth pixels. A pixel is correctly estimated if the disparity of flow end-point error is below 3 pixels or 5% of its true value [7]. Figure 2 depicts sample stereo images with ground truth disparity from this data set.

Figure 2: KITTI data set stereo images and lidar created disparity map.

5 Testing methodology

The performance of algorithm can be assessed with respect to the execution time and accuracy in general. However, both algorithms were compared just with respect to the disparity map estimation accuracy in this paper. The performance was evaluated using a custom created script in MATLAB which used the KITTI evaluation framework as its base. SGM was implemented using the OpenCV Stereo SGBM method. The method was implemented with the possibility to change the block size. It should be stressed that with block size of 1 this method performs equally as the classic SGM algorithm described in Section 2 [14]. Other parameters of SGM algorithm are gathered in Table 2.

Table 2: SGM algorithm parameters.

Parameter	Value
pre_filter_cap	63
sad_window_size	3
p1	3*3*4
p2	3*3*32
min_disparity	0
num_disparities	128
uniqueness_ratio	10
speckle_window_size	100
speckle_range	32
disp_max_diff	1
full_dp	1

The DispNet convolutional neural network was implemented using fine-tuned weights that are available on the researchers GitHub channel [15][16]. Testing was conducted on the first 100 images of the KITTI data set. Errors by disparity estimation were calculated for each test image, whereat the ground truth and the protocol from KITTI data set were employed. Finally, the obtained errors were averaged over entire data set.

6 Results

The results are presented in the form of a cumulative error for each image and as the average error over entire KITTI testing data set. Figure 3 depicts qualitative results for the sample images, namely the estimated disparity, the ground truth disparity, and the calculated error for SGM method (left column) and DispNet (right column). It could be observed that the ground truth images are in different colours. The reason is differences in the maximum disparity that can be computed by specific disparity estimation method. Results are presented in Figure 4 in the form of cumulative errors graphs for each test image from KITTI data set and for both disparity estimation method. Values are in percentages. For example, the value of 35% means that the disparity was wrongly determined (i.e., calculated disparity for a pixel differs from the ground truth for more than 3 pixels) for 35% of the test image pixels. It can be easily seen the better performance of DispNet compared to the SGM algorithm. Obtained results are within expected results and as reported on benchmark pages of the KITTI data set [17]. The average error for the SGM algorithm on the complete KITTI test data set is 21.5% (σ = 7.7) and 6.1% (σ = 3.4) for the DispNet.

7 Conclusion

Within this paper, a simple experiment comparing SGM and DispNet methods demonstrated that neural networks possess exceptional strength for disparity map estimation even without additional image processing and transformations. The obtained research results show that neural networks have progressed over the last few years and gained an advantage over classic methods for the disparity estimation. The KITTI data set was an excellent choice for benchmarking, because it consists of real images with the addition of ground truth. Results on the KITTI data set pointed out that the DispNet neural network estimate disparity accurately for 15% more image pixels on average than the classic SGM method. In addition, the standard deviation is lower as well which means that the disparity estimation using DispNet is more consistent.

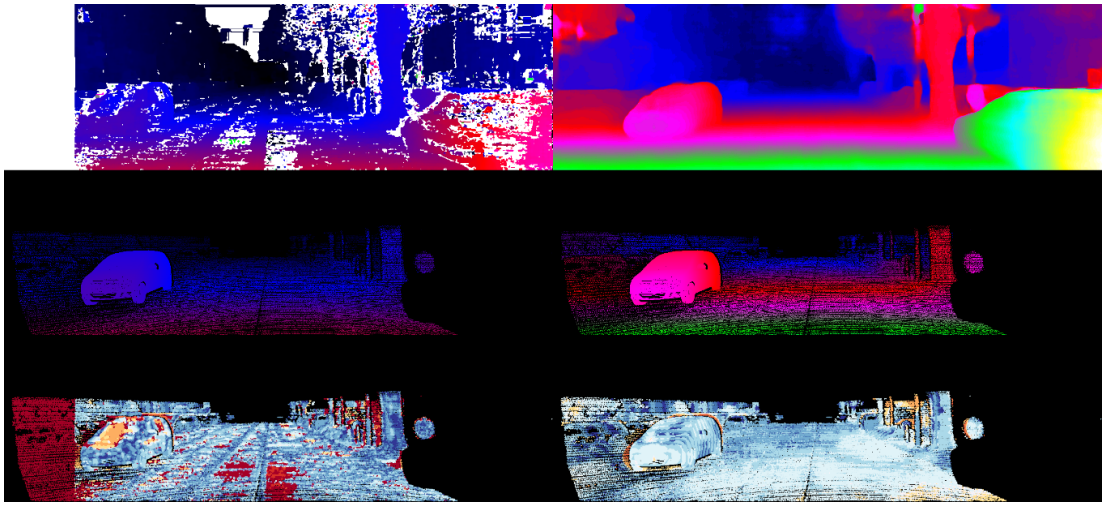

Figure 3: Qualitative results for the sample testing stereo image. Results for the SGM are in the left column, while the DispNet are in the right column. The topmost row depicts estimated disparity maps, in the middle row is a ground truth, while the bottommost row presents the computed error between estimation and ground truth.

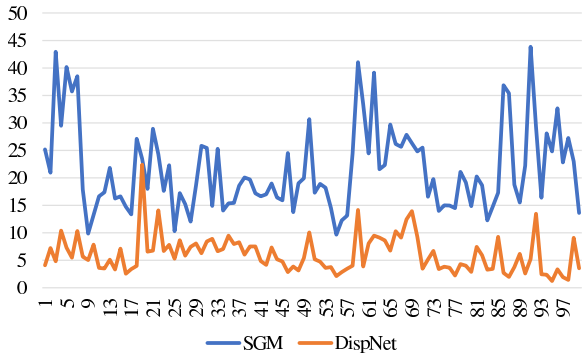

Figure 4: Cumulative errors (in percent) for each testing image from KITTI data set for both disparity estimation methods (SGM and DispNet).

References

[1] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 2, pp. 328–341, Feb. 2008.

[2] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001), pp. 131–140.

[3] C. C. Aggarwal, Neural Networks and Deep Learning. Springer, 2018.

[4] F. Zhang, V. Prisacariu, R. Yang, and P. H. S. Torr, "GA-Net: Guided Aggregation Net for End-to-end Stereo Matching," CoRR, vol. 1904.06587, 2019.

[5] J. Žbontar and Y. Le Cun, "Computing the stereo matching cost with a convolutional neural network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June, pp. 1592–1599.

[6] N. Mayer et al., "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, pp. 4040–4048.

[7] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in Proceedings of the CVPR, 2015, vol. 07-12-June, pp. 3061–3070.

[8] M. Menze, C. Heipke, and A. Geiger, "Object Scene Flow," ISPRS, vol. 140, pp. 60–76, 2018.

[9] P. Viola and W. M. Wells, "Alignment by Maximization of Mutual Information," Int. J. Comput. Vis., vol. 24, no. 2, pp. 137–154, 1997.

[10] C. Tomasi and S. Birchfield, "Depth discontinuities by pixel-to-pixel stereo," Int. J. Comput. Vis., vol. 35, no. 3, pp. 269–293, 1998.

[11] J. Kim, V. Kolmogorov, and R. Zabih, "Visual Correspondence Using Energy Minimization and Mutual Information," Proc. Ninth IEEE Int. Conf. Comput. Vis., vol. 2, pp. 1033–, 2003.

[12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Proc. 3rd Int. Conf. Learn. Represent., pp. 1–15, Dec. 2014.

[14] "OpenCV documentation: StereoSGBM Class Reference." [Online]. Available: https://docs.opencv.org/3.4.1/d2/d85/classcv_1_1StereoSGBM.html. [Accessed: 11-Jun-2019].

[15] A. Tonioni, "Real-time self-adaptive deep stereo GitHub." [Online]. Available: https://github.com/CVLAB-Unibo/Real-time-self-adaptive-deep-stereo.

[16] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. Di Stefano, "Real-time self-adaptive deep stereo," CoRR, vol. 1810.05424, Oct. 2018.

[17] "KITTI Stereo Evaluation 2015." [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo. [Accessed: 13-Jun-2019].