Patch Based Confidence Prediction for Dense Disparity Map

Akihito Seki[1,2]
akihito.seki@toshiba.co.jp

Marc Pollefeys[2]
marc.pollefeys@inf.ethz.ch

[1] Corporate R&D Center
Toshiba Corporation, Japan

[2] Department of Computer Science
ETH Zürich, Switzerland

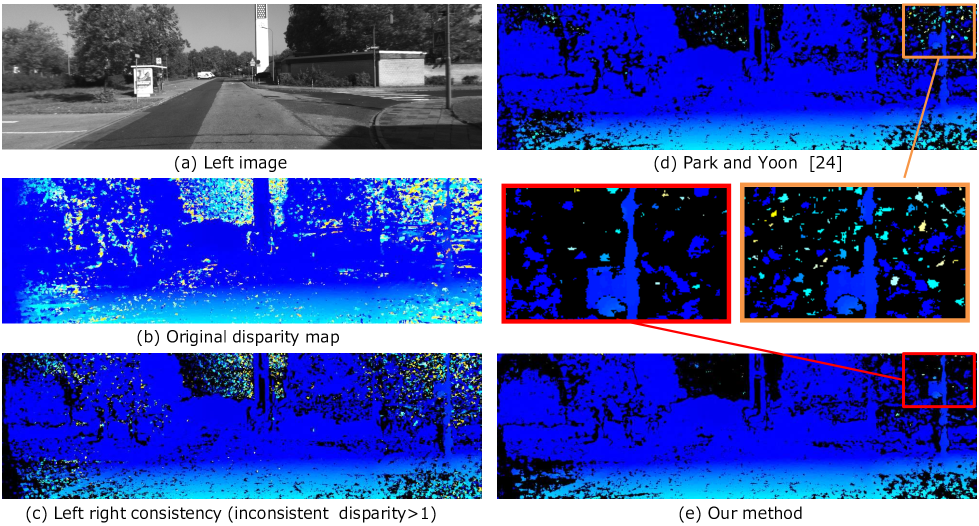
Abstract

In this paper, we propose a novel method to predict the correctness of stereo correspondences, which we call confidence, and a confidence fusion method for dense disparity estimation. The input of our method consists in a two channels local window (disparity patch) which is designed by taking into account ideas of conventional confidence features. 1st channel is coming from the idea that neighboring pixels which have consistent disparities are more likely to be correct matching. In 2nd channel, a disparity from another image is considered such that the matches from left to right image should be consistent with those from right to left. The disparity patches are used as inputs of Convolutional Neural Networks so that the features and classifiers are simultaneously trained unlike what is done by existing methods. Moreover, the confidence is incorporated into Semi-Global Matching(SGM) by adjusting its parameters directly. We show the prominent performance of both confidence prediction and dense disparity estimation on KITTI datasets which are real world scenery.

1 Introduction

Stereo disparity estimation is one of the most important problems in computer vision. Many correspondence methods and optimization methods have been proposed for many years [29]. The disparity map is widely used, for example in object detection [12], surveillance [27], and autonomous driving for cars and unmanned air vehicles [23].

Accurate stereo correspondences lead to better quality of the disparity maps. Recently, many accurate correspondence methods have been proposed[2, 18, 32, 34]. However, even the best matching methods come up with incorrect correspondences due to various reasons such as occlusion, saturation, pixel intensity noise, specularity, and calibration error. It is therefore necessary to estimate the confidence of correspondences in order to remove low reliable matches. Moreover these confidences can be used to interpolate correspondences over images [3, 8, 24, 25]. "Left right consistency check" is one of the most popular strategies [6]. It assumes the matches from left to right image should be consistent with those from right to left. Figure 1 shows an original disparity map and disparity maps which are purged from their low confidence matches for different confidence measures. Many incorrect correspondences appear around sky, leaves, and pavement especially close to image borders. "Left right consistency check" (Fig. 1(c)) removes some incorrect correspondences. To


(a) Left image

(d) Park and Yoon [24]

(b) Original disparity map

(c) Left right consistency (inconsistent disparity>1)

(e) Our method

Figure 1: Qualitative results on KITTI 2012 stereo pair. Black pixels in the disparity map are considered as wrong correspondence by each prediction method.

detect other wrong matches, many hand crafted features have been proposed [4, 14, 26]. Learning based confidence measures [11, 21, 24, 28] combine these features and are able to outperform their individual usage as shown in Fig. 1(d). These features are carefully designed, however beneficial information might be undescribed or their representation might be too redundant. As a consequence, the conventional learning based confidence measures might have limited accuracy.

To overcome the problem, we leverage Convolutional Neural Networks (CNNs). CNNs provide high performance from primitive level processing such as patch based matching[2, 18, 32, 34] to high level ones such as scene classification[1, 16] and object detection[9, 35]. Deep learning using a CNN offers a promising way to improve upon hand crafted features. To the best of our knowledge, we are the first to leverage CNN for stereo confidence measure and show its prominent accuracy.

Contributions of this paper are as follows. First, we design a two channels disparity patch which takes into account the ideas of conventional confidence features. The patch is used as an input for CNN so that the discriminative features and classifier are simultaneously trained. As a consequence, our method is able to filter incorrect correspondences more correctly as shown in Fig. 1(e). In order to handle trade-off between accuracy and computation time, we propose three types of network structures and their input patches. Second, in order to acquire dense disparity, we incorporate the confidence into Semi Global Matching (SGM) [13] with simpler operations than the existing method. Finally, our confidence measure outperforms state of the art method [24]. In addition, our confidence fusion was able to get the best accuracy on KITTI 2012 stereo benchmark [7] and the second best on KITTI 2015 [20] without the need for a strong foreground shape prior.

This paper is organized as follows: Section 2 describes our confidence prediction and dense disparity estimation combined with SGM and the confidence. Section 3 shows experimental results both the confidence and dense disparity map accuracy on challenging scene. Section 4 summarizes this paper.

2 Proposed method

What information can be considered discriminative to predict unreliable correspondences? In this section, we first describe valuable information which distinguishes reliability of stereo correspondence and then show how to design the classifier and its input.

2.1 Discriminative information

Many features have been proposed to predict the confidence of stereo correspondences. Hu and Mordohai[14] categorized them into five groups. The first group of features focuses on the matching cost: large matching costs are unlikely to present correct correspondences. In the second group, features capture local properties of the cost curve. The curvature around the minimum matching cost is used as a confidence measure, i.e. smaller values which correspond to flat curves such as texture-less area indicate higher ambiguity. The third group corresponds to features based on local minima of the cost curve, such as "Peak Ratio(PKR)". PKR is computed as the minimum matching cost divided by the second local minimum. The fourth group uses the entire cost curve in order to compute a probability mass function over disparity. Finally, the fifth group contains features which capture the consistency between the left and right disparity maps. The idea is that the matches from left to right image and those from right to left image should indicate consistent disparity when the matches are correct. Those features have been explored for a long time. However beneficial information might be undescribed or their representation might be too redundant. We consider a unified framework for feature extraction and classification. Deep learning is a promising way to realize it.

Our input candidates are a stereo image pair, a cost volume over disparity, and a disparity map provided by winner takes all method. In early experiments, we explored discriminative inputs from them by using a basic Convolutional Neural Network (CNN). The CNN structure consisted of two or three Convolutional layers, Non linear layers, Fully connected layers, and Softmax layer. The matching cost over disparity didn't perform well. A local window which was allocated in a stereo image (Image patch) could have a little prediction ability, however the accuracy was far from state of the art method[24]. A local window which was allocated in a disparity map (Disparity patch) with early experimental CNN structure could achieve roughly equivalent accuracy to the conventional learning based methods[11, 24].

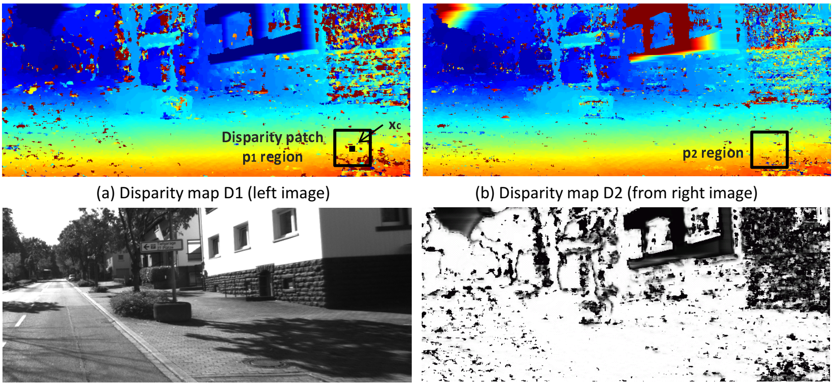
2.2 Confidence estimation with a CNN

Considering early experiments, we leverage the disparity patch and introduce the knowledge of the conventional features. We first take into account an idea from "Difference with Median Disparity(MED)"[28]. MED's idea is that neighboring pixels which have consistent value are more likely to be correct matching. Instead of subtracting median or mean value of the patch[6, 15], we simply subtract the disparity value at the central pixel $\mathbf{x}_c$ of the patch. Eq.(1) represents the disparity patch $\mathbf{p}_1$ converted from the disparity map D_1.

$$\mathbf{p}_1 = [D_1(\mathbf{x}) - D_1(\mathbf{x}_c)]_{\mathbf{x}\in W} \tag{1}$$

$\mathbf{x}$ indicates pixel position inside the local window W.

Conventional methods suggest to use a disparity from another image. Hence we employ the idea of the fifth group in previous section. We get the disparity map D_2 by converting



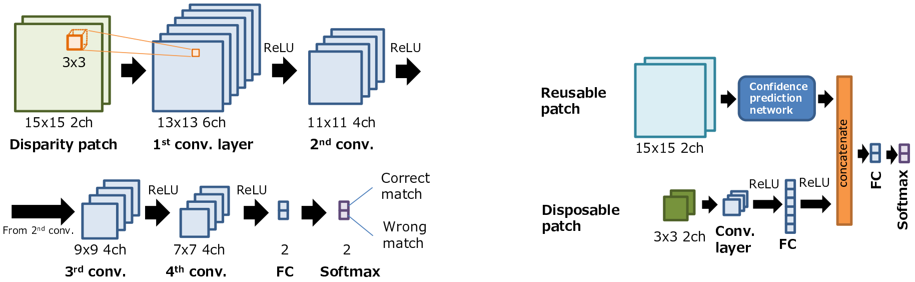
(a) Disparity map D1 (left image)

(b) Disparity map D2 (from right image)

(c) Left image

(d) Estimated confidence map

Figure 2: (a) Disparity map derived from the left image D_1 and (b) disparity map D_2 converted from right to left image. The color of D_1 and D_2 encodes disparity. (c) Left image. (d) Confidence map represents reliable pixels in white.



(a) Confidence prediction network

(b) Hybrid confidence prediction network

Figure 3: (a) Confidence prediction network consists of 4 layers of convolution and ReLU, Fully connected layer, and Softmax layer. (b) Hybrid network consists of normal network with “reusable patch” and small sub network with “disposable patch”.

the disparity map in the right image to the left image coordinate[28]. Figure 2 shows the disparity map D_1 and D_2. It appears that corresponding points have the same disparity, and that texture-less or saturated regions at walls and sky have different disparities. The second disparity patch $\mathbf{p}_2$ is designed as follows

$$\mathbf{p}_2 = [D_2(\mathbf{x}) - D_1(\mathbf{x}_c)]_{\mathbf{x}\in W}\,. \tag{2}$$

We use the two channels disparity patch $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2)$ as an input of CNN. The CNN is trained in a classifier manner. It means that a label is annotated to every input patch, indicating whether the correspondence at the center position of the patch is correct or not. As shown in Fig. 3(a), a disparity patch $\mathbf{p}$ of size 15×15 is input to the first layer which consists of 6 different 3×3 kernels and ReLU. Then, at the second, third, and fourth layers, 4 different 3×3 kernels and ReLU are applied respectively. A fully connected layer which has two outputs is connected next to the fourth layer. Finally, Softmax layer outputs a correspondence confidence at the center pixel of the patch.

We employ a basic and small CNN for the sake of reducing potential computation cost of the network. Note that Eq. (1) and (2) are computed for each patch due to subtraction of its central value $D_1(\mathbf{x}_c)$. It will be shown in Sec. 3.1 that the patch improves prediction accuracy drastically. Meanwhile, it makes slow computation because the output of the network for each pixel has to be computed from scratch, so we call the patch "disposable patch". If the patch is independent of the value at the central position of the patch, the confidence can be computed for all pixels in a single forward pass of the network by propagating entire image. It leads faster computation because many intermediate results can be reused [33]. We propose "reusable patch" $\mathbf{p}' = (\mathbf{p}_1{}', \mathbf{p}_2{}')$ as follows

$$\mathbf{p}_1{}' = [D_1(\mathbf{x})]_{\mathbf{x}\in W}\,, \quad \mathbf{p}_2{}' = [D_2(\mathbf{x}) - D_1(\mathbf{x})]_{\mathbf{x}\in W}\,. \tag{3}$$

In order to compensate for the computation time and accuracy, we also propose a hybrid network, which combines the "reusable patch" with a miniaturized "disposable patch" as shown in Fig. 3(b). The miniaturized 3 × 3 patch is applied to 3 different 3 × 3 kernels and ReLU, and then follows 6 outputs of fully connected layer. The outputs from the "reusable patch" and the "disposable patch" are concatenated and then applied to a fully connected layer which has two outputs. Softmax layer predicts the class of the patch. The hybrid network predicts better than the normal network with the "reusable patch". See results in Sec.3.1.

2.3 Confidence fusion for dense disparity map

So far, we have described pixel-wise confidence prediction. In this section, we incorporate the predicted confidence into Semi-Global Matching(SGM) [13]. SGM is widely used for dense disparity estimation due to its high accuracy while keeping low computation cost. An energy function E is defined as

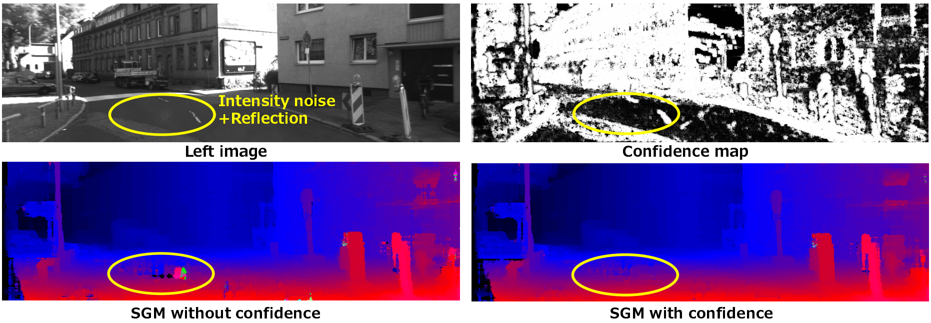
$$E(D) = \sum_{\mathbf{x}} \left(C(\mathbf{x}, D_{\mathbf{x}}) + \sum_{\mathbf{y}\in\mathbf{N_x}} P_1 T[|D_{\mathbf{x}} - D_{\mathbf{y}}| = 1] + \sum_{\mathbf{y}\in\mathbf{N_x}} P_2 T[|D_{\mathbf{x}} - D_{\mathbf{y}}| > 1] \right). \tag{4}$$

$C(\mathbf{x}, D_{\mathbf{x}})$ represents a matching cost at pixel $\mathbf{x}$ of disparity $D_{\mathbf{x}}$, so the first term is the sum of all pixel matching costs for the disparities of D. The second term represents slanted surface penalty P_1 for all pixels $\mathbf{y}$ in the neighborhood $\mathbf{N_x}$ of $\mathbf{x}$. $T[\cdot]$ represents Kronecker delta function. The third term indicates penalty P_2 for discontinuity disparity. P_2 should be set small according to the magnitude of the image gradient, for example $P_2 = P_2'/|I(\mathbf{x}) - I(\mathbf{y})|$ so that the discontinuities are easily selected[13].

Park and Yoon[24] proposed to modulate the matching cost volume with the confidence for SGM. They assume the low confidence pixels have unreliable matching costs over the disparity, therefore they enforce to lessen the fluctuation of the matching cost over the disparity. As a consequence, the discontinuities are unlikely to be selected at the low confidence pixels.

Recently, Zbontar and LeCun[34] have shown adequate improvement with Cross-based cost aggregation(CBCA) [36]. CBCA is an aggregation method of the matching cost volume, which consists in merging the costs of pixels that are located closely and have similar intensity values. A combination of the modulation[24] and CBCA requires considerable computation cost. We propose a simpler fusion method.

We assume the discontinuities are likely to have the large magnitude of the image gradient using the same assumption as the original SGM, but not all large gradient pixels correspond to them. We consider the pixels with high confidence should be trusted and are able



Left image

Confidence map

SGM without confidence

SGM with confidence

Figure 4: SGM with and without confidence estimated by our method. The parameters are the same as in Sec. 3.2.

to be the discontinuities easily. Hence, penalties at the high confidence pixel are designed to be decreased. We modify P_1 and P_2 as follows

$$P_{1,2}(\mathbf{x}) = f_{1,2}(I(\mathbf{x})) + P'_{1,2}\lambda \max(-\xi(\mathbf{x}) + m, 0). \tag{5}$$

$\xi(\mathbf{x})$ represents the confidence value at pixel $\mathbf{x}$, and is normalized from 0 to 1 so that larger value means higher probability of correct correspondence. m and λ are the parameters of a margin with possible range between 0 to 1 and blend ratio respectively. The function $f_{1,2}(I)$ is defined according to [33]. $P'_{1,2}$ is each maximum value of the penalty, i.e. `sgm_P1` and `sgm_P2` [33]. Eq. (5) adds an extra penalty to the pixels with confidence lower than m by a proportion of negative confidence $(-\xi(\mathbf{x}) + m)$ and λ. Penalties are directly changed so that the matching cost volume need not to be modulated.

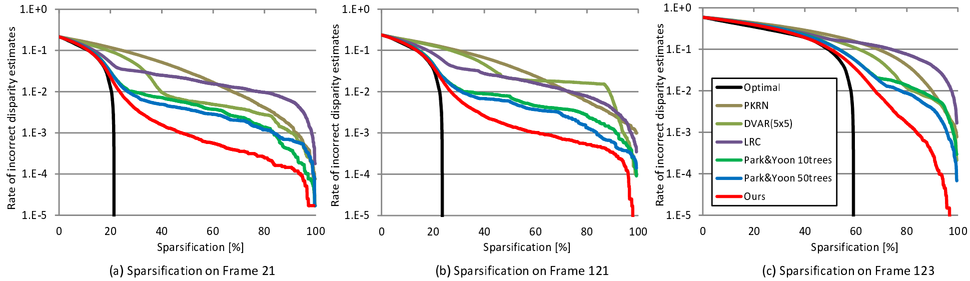
Figure 4 shows dense disparity maps given by SGM with and without confidence. These images aren't post processed. Bad estimates on pavement caused by image noise and reflection of texture on a dashboard are successfully removed. Quantitative results are enclosed in Sec.3.2.

3 Experimental Result

We describe two experiments. In the first one, we compare the accuracy of our patch based confidence prediction to other conventional methods (Sec.3.1). In the next we evaluate dense disparity accuracy on public benchmarks (Sec.3.2).

3.1 Confidence accuracy

Following recent publications for evaluation of the confidence measure, we use KITTI 2012 dataset[7]. The dataset was captured by vehicle mounted stereo cameras and LIDAR was used for acquiring ground truth of disparity maps. Ground truth is not provided for test images, so the stereo pairs from training dataset are used for both training and evaluation. As conventional methods[11, 24, 28] used eight stereo pairs 43, 71, 82, 87, 94, 120, 122, and 180th for training and the other 186 stereo pairs for evaluation, we followed the setting. The eight pairs have relatively much incorrect correspondences than the rest pairs. We extracted 0.66 million of positive and negative samples from the pairs. In the later of this section, we will show the effects of the size of the training set. The architecture is trained with Caffe[15]. We employed stochastic gradient descent to minimize the cost of the network



(a) Sparsification on Frame 21 (b) Sparsification on Frame 121 (c) Sparsification on Frame 123

Figure 5: Comparison of sparsification plots on frame 21, 121, and 123 of KITTI training stereo pairs. Naive peak ratio (PKRN)[14], variances of the disparity values in a local 5 × 5 window (DVAR)[24], and left right consistency (LRC)[14] are drawn as conventional features [14]. Park&Yoon[24] by 22 dimensional features and 10 or 50 trees of forests are plotted as state of the art method.

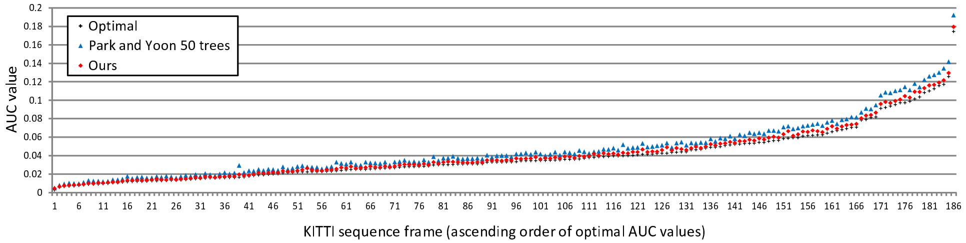


Figure 6: AUC value according to the ascending order of optimal AUC values. We plotted our method and Park&Yoon with 22 dimensions of 50 trees as state of the art method.

with a softmax loss function. Learning rate, momentum, and batch-size are set to 0.001, 0.9, and 64 respectively. After ten million iterations, the trained parameters were used.

For quantitative comparison, we employ the sparsification curve and its area under curve (AUC) value[11, 24, 28]. Better confidence prediction methods have AUC values that are closer to the optimal curve: It means the method removes incorrect correspondence pixels while keeping the correct ones. Fig. 5 shows the sparsification curve which represents the evaluation of incorrect disparity estimation rate. As state of the art method, we chose [24] with 22 dimensional features since this setting outperformed other learning based methods [11, 28]. All methods are trained on the same training set. Our method provides superior accuracy to any other methods. Fig. 6 represents AUC values which are sorted with respect to ascending order of optimal AUC values over evaluation frames. Smaller AUC value of optimal indicates better quality of the disparity map. As one can see, our method has superior accuracy on almost all evaluation frames in spite of the difference of optimal AUC values.

Table 1 shows overall AUC value of 186 evaluation frames. We evaluated two kinds of similarity measure, census transform (“Census”) [31] and CNN based matcher (“MC-CNN”) which gives much more correct correspondences [34]. In our methods, “fast” and “hybrid” indicate disparity patch given by Eq.(3) and combination of both types of disparity patches, respectively. In both similarity measures, ours outperforms state of the art method.

Figure 7 shows AUC values and computation time with respect to patch size. 9 × 9 patch gives the fastest while the worst AUC value. Considering the accuracy and computation time, 15 × 15 seems reasonable. We evaluated the effects of the size of the training set as

Method		AUC [Census]	AUC [MC-CNN]	Runtime[sec.]
Optimal		0.03953	0.02144	–
Ours		**0.04198(6.2%)**	**0.02287(6.7%)**	28.5(0.5*)
	fast	0.04497(13.8%)	0.02545(18.7%)	**0.3**
	hybrid	0.04483(13.4%)	0.02525(17.8%)	0.5
Park&Yoon [24](22 dim.)	50 trees	0.04702(18.9%)	0.02635(22.9%)	2.2
	10 trees	0.05152(30.3%)	0.02739(27.8%)	0.4

Table 1: Comparison of overall AUC value over 186 frames of KITTI training data with different similarity measures. A bracket at AUC means difference between optimal and estimated result. Runtime of our method is measured on single thread with 1st generation of Intel(R) Core(TM) i7 2.8GHz with 12GB memory. "*" indicates computation time on NVIDIA(R) Titan X.

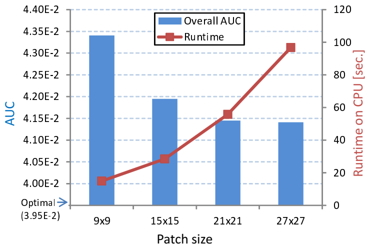


Figure 7: AUC values and computation time with respect to patch size.

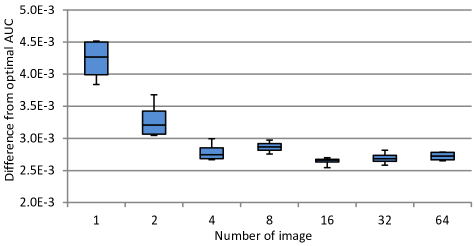


Figure 8: The number of training images and difference from optimal AUC value.

shown in Fig. 8. The normal networks were trained by using randomly selected images from 0 to 99th frame of KITTI 2012 on each of the training set size. After ten million iterations, AUC values were computed over images from 100 to 193th frame. We tested 4 times on each of the size and drew the boxplots. The performances of the networks seem saturated over 4 images.

3.2 Dense disparity accuracy

For dense disparity estimation, we employed MC-CNN[33, 34] and its post processing. We modified one of the bilateral filter at the post processing step in order to preserve object borders more strongly, and parameters are optimized. Table 2 shows estimated errors with these modifications. We use the default error criterion: The percentage of erroneous pixels on non-occluded areas with an error threshold of 3 pixels.

Figure 9 and Table 3 show the accuracy on KITTI 2012 testing dataset [1]. We get the best accuracy when MC-CNN-acrt[33, 34] is employed as a similarity measure. We found that the combination of our approach with MC-CNN-fast which runs in less than 2 seconds also provides a significant improvement with respect to its baseline.

Figure 10 and Table 4 show disparity map error on KITTI 2015 testing dataset [20]. Our method lost top on this dataset because annotated density of foreground (vehicle) is much higher than that of background. Additionally, the foregrounds which have transparent region

[1] You can see more results at http://www.cvlibs.net/datasets/kitti/eval_stereo.php

MC-CNN-acrt original	Ours without confidence	Ours
2.61% (3.19%)	2.55% (3.14%)	2.50% (3.07%)

Table 2: Out-Noc error on KITTI 2012 training dataset. Pure SGM errors are inside bracket. “Ours without confidence” corresponds to leverage only the optimized paramters and the modified bilateral filter.

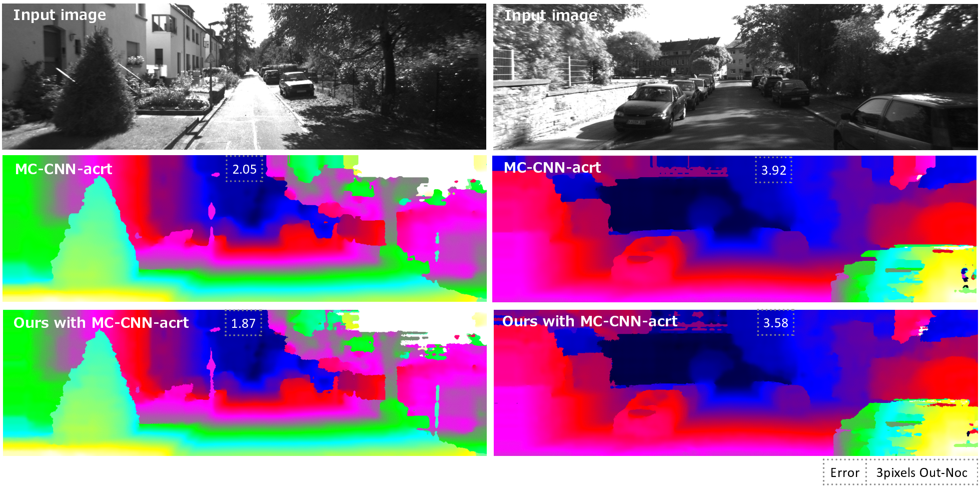


Figure 9: Example results of original MC-CNN-acrt and our fusion method with MC-CNN-acrt in KITTI 2012.

such as windshield and reflected region are hard to be predicted to fit vehicle’s shape. It makes an advantage for the method which uses object knowledge such as Displets [10]. However, our method could get the best rank on background region which was equivalent to KITTI 2012 evaluation criterion. Compared to MC-CNN-acrt, our method provides more accurate disparity not only on background but also on foreground.

4 Conclusion

In this paper, we proposed a method for predicting correspondence confidence. Moreover, we proposed a fusion method of the confidence for dense disparity estimation. We exploited a two channels disparity patch which was designed by taking into account ideas of conventional confidence features. Neural networks which predict correspondence confidence were trained with the patches. Then, the confidence was incorporated into SGM by adjusting parameters without modulating matching cost volume. Our method was able to reduce confidence prediction error up to 1/3 against state of the art method. Moreover, accuracy of dense disparity achieved the best and the second best rank on KITTI 2012 and 2015 benchmark respectively.

Acknowledgment The author would like to thank Ian Cherabier working at Pollefeys Lab. for helping to improve the paper.

Rank	Method	Setting	Error	Runtime[sec.]
1	Ours with MC-CNN-acrt		**2.36%**	68*
2	Displets v2[10]		2.37%	265
3	VDS(anonymous)		2.42%	68*
4	MC-CNN-acrt[33, 34]		2.43%	67*
5	cfusion[22]	MV	2.46%	70*
6	Ours with MC-CNN-fast		2.68%	1.8*
7	PRSM[30]	F,MV	2.78%	300
8	MC-CNN-fast[33]		2.82%	**0.8***

Table 3: Out-Noc error on KITTI 2012 testing dataset by May 1st 2016. Rank is based on "Error". F and MV at Setting represent the method uses two and more than two temporally adjacent images, respectively. "*" at Runtime means GPU computation. The parameters of our fusion with MC-CNN-acrt are (`sgm_P1`,`sgm_P2`,`sgm_Q1`,`sgm_Q2`,`sgm_V`,`sgm_D`)= $(1.2, 24, 2, 4, 1.5, 0.06)$[33] and $(m, \lambda) = (0.6, 0.7)$ in Eq.5.

Rank	Method	D1-bg	D1-fg	D1-all	Runtime[sec.]
1	Displets v2[10]	3.00%	5.56%	**3.43%**	265
2	Ours with MC-CNN-acrt	**2.58%**	8.74%	3.61%	68*
3	MC-CNN-acrt[33, 34]	2.89%	8.88%	3.89%	67*
4	CNN-SPS[17]	3.30%	7.92%	4.07%	80*
6	DispNetC[19]	4.32%	**4.41%**	4.34%	0.06*

Table 4: Out-Noc error on KITTI 2015 testing dataset by May 1st 2016. Rank is based on D1-all error. The parameters of our fusion with MC-CNN-acrt are (`sgm_P1`,`sgm_P2`,`sgm_Q1`,`sgm_Q2`,`sgm_V`,`sgm_D`)= $(1.8, 27, 2, 4, 1.5, 0.08)$ and $(m, \lambda) = (0.6, 2.0)$.

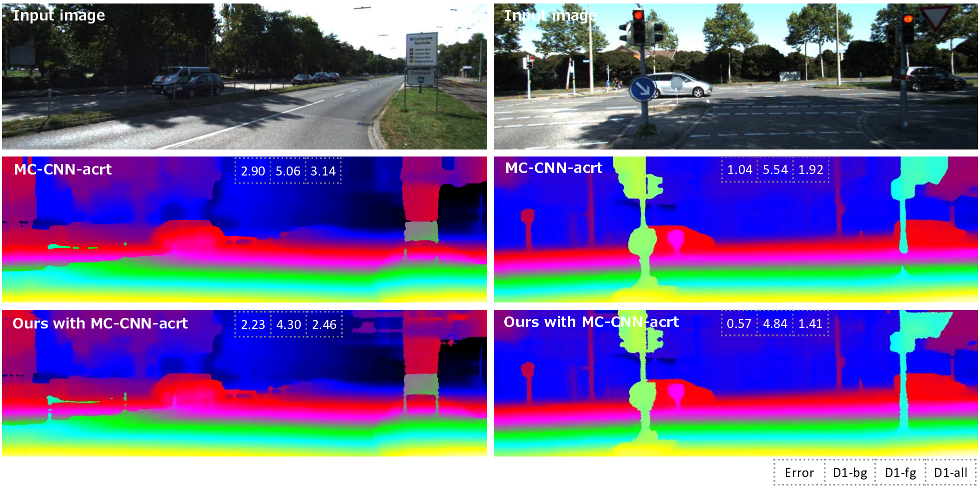


Figure 10: Example results of original MC-CNN-acrt and our fusion method with MC-CNN-acrt in KITTI 2015.

References

[1] Xinlei Chen and C. Lawrence Zitnick. Mind's Eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] Zhuoyuan Chen, Xun Sun, and Liang Wang. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[3] Stefan Gehrig David Pfeiffer and Nicolai Schneider. Exploiting the power of stereo confidences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 297–304, 2013.

[4] Geoffrey Egnal, Max Mintza, and Richard P. Wildesb. A stereo confidence metric using single view imagery with comparison to five alternative approaches. In *Image and Vision Computing*, volume 11, pages 943–957, 2004.

[5] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[6] Pascal Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49, 1993.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] Riccardo Gherardi. Confidence-based cost modulation for stereo matching. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2008.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] Fatma Guney and Andreas Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] Ralf Haeusler, Rahul Nair, and Daniel Kondermann. Ensemble learning for confidence measures in stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 305–312, 2013.

[12] Scott Helmer and David Lowe. Using stereo for object recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3121–3127, 2010.

[13] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, February 2008.

[14] Xiaoyan Hu and Philippos Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11):2121–2133, 2012.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[17] Lei Fan Zexian Wang Long Chen, Jianda Chen and Guodong Xie. *A Convolutional Neural Networks based Full Density Stereo Matching Framework*, 2015. Computer Vision and Image Understanding.

[18] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alex Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, and Horst Bischof. Using self-contradiction to learn confidence measures in stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Valsamis Ntouskos and Fiora Pirri. Confidence driven TGV fusion. In *arXiv:1603.09302*, 2016.

[23] Helen Oleynikova, Dominik Honegger, and Marc Pollefeys. Reactive avoidance using embedded stereo vision for MAV flight. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 50–56, 2015.

[24] Min-Gyu Park and Kuk-Jin Yoon. Leveraging stereo matching with learning-based confidence measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 101–109, 2015.

[25] Gorkem Saygili, Laurens van der Maaten, and Emile A. Hendriks. Stereo similarity metric fusion using stereo confidence. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2014.

[26] Daniel Scharstein and Richard Szeliski. Stereo matching with non-linear diffusion. *International Journal of Computer Vision*, 28:155–174, 1998.

[27] Akihito Seki, Oliver J. Woodford, Satoshi Ito, Björn Stenger, Makoto Hatakeyama, and Junichi Shimamura. Reconstructing fukushima: A case study. In *Proceedings of the International Conference on 3D Vision*, pages 681–688, 2014.

[28] Aristotle Spyropoulos, Nikos Komodakis, and Philippos Mordohai. Learning to detect ground control points for improving the accuracy of stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1621–1628, 2014.

[29] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., 1st edition, 2010.

[30] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, pages 1–28, 2015.

[31] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–158. Springer-Verlag New York, Inc., 1994.

[32] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. In *arXiv:1510.05970*, October 2015.

[34] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[35] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2014.

[36] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):1073–1079, 2009.

Product names mentioned in this paper are trademarks of their respective companies.