Learning from scratch a confidence measure

Matteo Poggi
http://vision.disi.unibo.it/~mpoggi
Stefano Mattoccia
http://vision.disi.unibo.it/~smatt

University of Bologna
Department of Computer Science and Engineering (DISI)
Viale del Risorgimento 2
Bologna, Italy

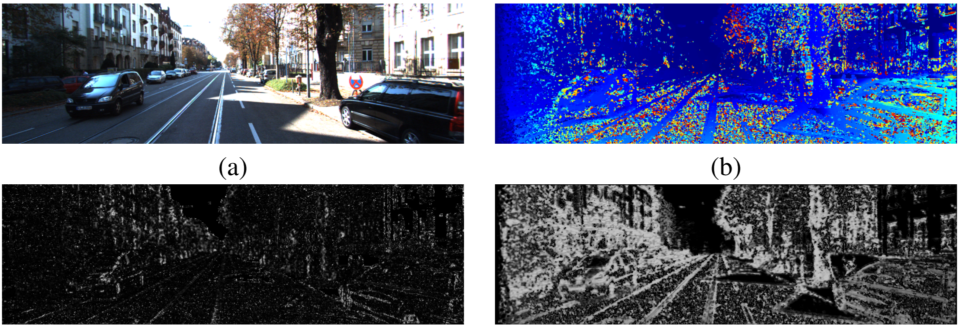
Abstract

Stereo vision is a popular technique to infer depth from two or more images. In this field, confidence measures, typically obtained from the analysis of the cost volume, aim at detecting uncertain disparity assignments. As recently proved, multiple confidence measures combined with hand-crafted features extracted from the cost volume can be used also for other purposes and in particular to improve the overall disparity accuracy leveraging on machine learning techniques. In this paper, starting from the observation that recurrent local patterns occurring in the disparity maps can tell a correct assignment from a wrong one, we follow a completely different methodology to infer a novel confidence measure from scratch. Specifically, leveraging on Convolutional Neural Networks, we pose the confidence formulation as a regression problem by analyzing the disparity map provided by a stereo vision system. Once trained on a subset of the KITTI 2012 dataset with the disparity maps provided by the simple block-matching algorithm, our confidence measure outperforms state-of-the-art with two datasets (KITTI 2015 and Middlebury 2014) as well as with two stereo algorithms. The experimental evaluation reported clearly highlights that our approach is capable to better generalize its behavior in different circumstances with respect to state-of-the-art. Finally, not being based on cost volume analysis, our proposal is also potentially suited for out-of-the-box depth generation devices which usually do not expose the cues required by top-performing approaches.

1 Introduction

Depth sensors are deployed in several computer vision applications and stereo (active or passive) is a popular method to infer depth from two or more images. Although several approaches have been proposed to tackle this problem and state-of-the art algorithms enable to obtain quite accurate results, intrinsic problems of this technique such as poorly textured areas, distinctiveness [19] and occlusions [3] as well as difficult environments characterized by specular surfaces or poor illumination conditions may lead to wrong disparity assignments. These facts have been further emphasized with the availability of realistic datasets such as KITTI [6, 22] and Middlebury [30]. Therefore, for practical applications it is mandatory to filter out wrong assignments by means of effective confidence measures (CMs) aimed at encoding the degree of uncertainty of each point.

Most approaches, recently reviewed and evaluated by Hu & Mordohai [14], analyze intermediate results provided by stereo algorithms (i.e., the cost volume (CV)) and/or the



(a) (b) (c) (d)

Figure 1: An image from KITTI 2015 dataset. (a) reference image, (b) disparity map (c) confidence map according to Left-Right Difference [14] CM (d) confidence map according to the proposed CCNN.

final disparity map(s) in order to encode the uncertainty according to the behavior of an ideal approach. Recently, some authors [9, 27, 31] proved that optimal results can be obtained by jointly processing a pool of CMs within a machine learning framework based on random forest (RF). In particular, state-of-the-art approach [27] proposed by Park & Yoon computes a very effective CM processing, by means of a RF, a feature vector made of existing CMs and hand-crafted features obtained from the analysis of CV and disparity map. In the same work, the novel CM was deployed to improve the overall disparity accuracy by CV modulation.

Starting from the observation that correct and wrong measurements are typically characterized by recurrent patterns and considering the effectiveness of Convolutional Neural Networks (CNNs) applied to computer vision problems we decided to investigate the opportunity to obtain a confidence measure from scratch. This means that our proposal follows a completely different strategy w.r.t. previous work in this field because it does not rely on any existing CM nor it extracts hand-crafted features from the CV or the disparity map. Moreover, our proposal taking as input only the disparity map is also suited for out-of-the-box depth sensors (e.g., Intel Realsense [15], Zed camera [32], or FPGA-based stereo cameras [12, 20]) that in most cases do not provide the CV due to intellectual property issues, limited bandwidth, etc. Exhaustive experimental results and a cross-validation with different datasets and algorithms confirm that our proposal outperforms state-of-the-art. Figure 1 shows for frame 000000 of the KITTI 2015 dataset the reference image (a), the disparity map computed by a stereo algorithm (b), the outcome of a CM known in literature (c) and the result yielded by the CM proposed in this paper (d), referred to as CCNN.

2 Related work

Several confidence measures aimed at detecting well-known issues in stereo matching have been proposed and evaluated in the literature [3, 4, 14]. Hu & Mordohai [14] categorized CMs into six main groups according to the methodology used to assess match reliability: analysis of matching costs, local properties of the cost curve, analysis of local minima within the cost curve, analysis of the matching curve, consistency between left and right disparity maps and distinctiveness-based measures. Most of these cues are extracted from the CV. They also defined a metric and performed an extensive evaluation of the accuracy when dealing with detection of correct matches, discontinuities, occlusions and disparity selection on controlled [29] and outdoor [33] data.

CMs can be exploited to detect unreliable disparity assignments [28], occlusions [11, 24], improving accuracy near depth discontinuities [5] and sensors fusion [23]. Moreover, they can be used to improve disparity map accuracy by modulating the raw matching costs according to the supposed uncertainty. In this latter context [25] and [7] proposed CMs aimed at enabling more distinctive matching costs. More effective approaches, leveraging on machine learning techniques and the deployment of multiple CMs, significantly improved previous methods. In [17], stereo matches are classified into three categories based on their correctness by means of a MRF framework, while more recent approaches [9, 27, 31] used random decision forests to learn the reliability of disparity assignments, taking as input feature vector containing multiple state-of-the-art confidence measures. In particular, [9] and [31] achieved an improved sparsification performance on Middlebury [29] and KITTI [6] with respect to the original stand-alone measures. Park & Yoon [27] obtained even better results, training a RF on KITTI and Middlebury datasets, with a first phase dealing with the selection, from a large pool of CMs and features, of the most influent variables for the purpose according to [8] criteria and a second one training on such selected features. These works also proved that CMs can be very effective to improve the accuracy of popular stereo algorithms based on MRF [16], by selecting highly confident disparity assignments as ground control points [31], or cost volume filtering approach based on the *guided filter* [2, 10, 13] and Semi Global Matching (SGM) [11], modulating the raw matching costs according to the outcome of the learning process as proposed in [27].

Deep learning techniques have been successfully applied to computer vision but seldom for stereo matching and related problems so far. In [34], Zagoruyko & Komodakis reported a complete study about how to learn directly from image data a general similarity function by exploiting CNN architectures. Specifically, they used 2-channels, *siamese* and *pseudo-siamese* models, reporting results related to stereo matching as a particular case of image matching. More recently, Zbontar & LeCunn, proposed in [35, 36] an effective methodology for matching cost computation relying on a CNN. Their strategy turned out to be very effective and enabled this method to rank, respectively, third [35] on 2012 KITTI dataset [6] and first [36] on both 2012 and 2015 KITTI datasets [6, 22]. In [36], an accurate architecture and a faster/simplified one were proposed. The latter showed a remarkable speed-up with respect to the accurate CNN architecture (0.8 sec vs 67 sec) with an increase of the error rate smaller than 1% on both KITTI datasets. Other recent works that addressed stereo by means of CNNs are [1, 18]. Finally, deep architectures usually require huge amount of data for training and popular stereo datasets [6, 22, 30] might be not large enough for this purpose. Therefore, some authors deal with this issue by proposing a data-augmentation process leveraging on multiple view points and *contradictions* between multiple depth maps [26], or by producing synthetic datasets [21] large enough to run an end-to-end training of deep architectures.

3 Confidence measure inferred by a CNN

Our proposal starts from the observation that recurrent patterns characterize wrong and correct disparity assignments. In fact, as highlighted in Figure 2, local regions in the disparity map often contain recurrent patterns that enable to clearly assess the reliability of the disparity assignments. Motivated by recent work in this field [21, 26, 35, 36], we train, on a large

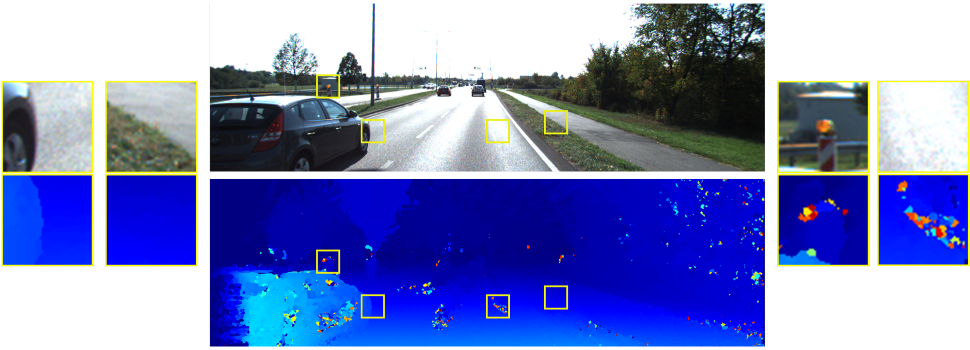


Figure 2: Reference image and disparity map computed by the SGM [11] algorithm with highlighted four regions. On the left, two regions including correct disparity assignments and, on the right, two regions including wrong disparity assignments.

dataset with ground-truth, a deep architecture to encode the degree of uncertainty from the disparity map. For each pixel, we extract a square patch centered on the disparity map and forward it to a CNN, trained to distinguish between patterns corresponding to correct and erroneous disparity assignments and, thus, to infer a confidence value. To this aim, we deploy a deep architecture, made of a relatively low number of layers with respect to state-of-the-art CNNs designed for higher level tasks, capable to learn such property and hence to infer an effective CM.

3.1 Proposed architecture

The architecture of our CNN is made of a single channel network that takes as input $N \times N$ patches, each one containing disparity values normalized between zero and one, represented by a $1 \times N \times N$ tensor. Although the size of the patches is relatively small compared to the disparity map, it should provide to the CNN enough cues to infer the degree of uncertainty for each point. In our experiments we found that $N = 9$ enables to obtain quite effective results as reported in the experimental evaluation. The first part of our network is made of $\frac{N-1}{2}$ convolutional layers, each one followed by a Rectifier Linear Unit (ReLU).

$$ReLU(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases} \tag{1}$$

Each convolutional layer contains F filters of size 3×3. No padding or stride is applied, making the final output of the convolutional layers, a $F \times 1 \times 1$ tensor (each layer reduces the initial size N by 2 pixels), directly forwarded to the fully-connected part of the network deploying two layers, made of L neurons each, followed by ReLUs (1). The final layer collapses into a single neuron in charge of the regression.

According to a common methodology usually deployed when dealing with deep architectures, the fully-connected layers are replaced by convolutional layers made of L kernels 1×1. This allows us to train the network on image patches (and, then, to easily handle samples generation and mini-batch dimension) as well as to compute a dense confidence map with a single forward of the full resolution image with a *0-padding* of $\frac{N-1}{2}$ around it, keeping for the output the same $w \times h$ size of the input disparity map due to the absence

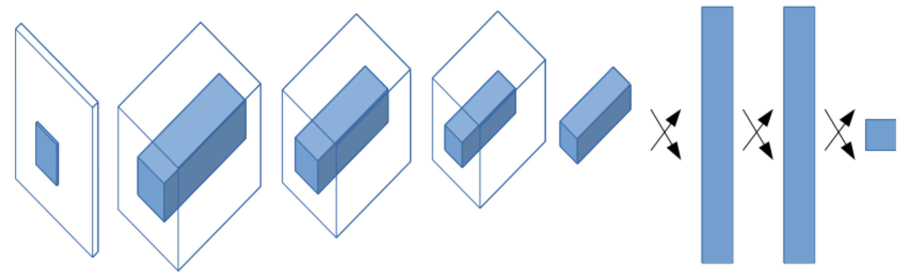


Figure 3: Architecture of the proposed CNN to infer the CM from from the raw disparity map. It is a single channel network, designed for 9×9 image patches. Four convolutional layers apply 64 overlapping kernels (stride equal to 1) of size 3×3. Two fully-connected layers made of 100 neurons each (i.e., 100 1×1 convolution kernels) lead to the final regression node.

of pooling operations or stride factors inside the convolutional layers. Forwarding a single $w \times h$ image, which allows to reuse many intermediate results, rather than forwarding $w \times h$ patches of size $N \times N$ enables to significantly reduce the execution time [35, 36]. For instance, by running our approach on a standard Intel i7 6600K processor the time required to obtain a full confidence map (on a typical KITTI disparity map and $N = 9$) is about 5 minutes by forwarding single patches through the fully-connected network and only 630 ms with the outlined fully-convolutional architecture. Moreover, with a Titan X GPU, the same fully-convolutional network takes only 116 ms.

3.2 Training procedure

In our evaluation, we trained the proposed CNN architecture on the first 50 frames of the the KITTI 2012 dataset [6] extracting samples only centered on pixels with available ground-truth values (approximatively $\frac{1}{3}$ of the overall disparity values). This strategy provides more than 6.5 million samples to the CNN. Experiments with larger training datasets did not improve significantly the effectiveness of CCNN. Disparity maps for the training procedure are computed by the Block Matching algorithm (BM) aggregating costs on 5×5 patches. The pointwise matching costs are obtained according to the Hamming distance on census transformed images computed on 5×5 patches. The disparity map is obtained from the CV by means of the Winner-Takes-All strategy (WTA). We label with '1' all the confident disparity assignments (i.e., those values that differ by one or less from the ground-truth) and with '0' otherwise. According to this strategy, the average error rate of the BM algorithm is approximatively 50%. This fact provides a balanced distribution of samples for training the CNN.

In our evaluation, we found out that 9×9 patches enable a quite effective learning for our method. Therefore, our architecture is composed of 4 convolutional layers, each one made of $F = 64$ kernels as depicted in Figure 3. We deploy random connection tables, which improve learning and runtime speed and lead to superior matching prediction during the

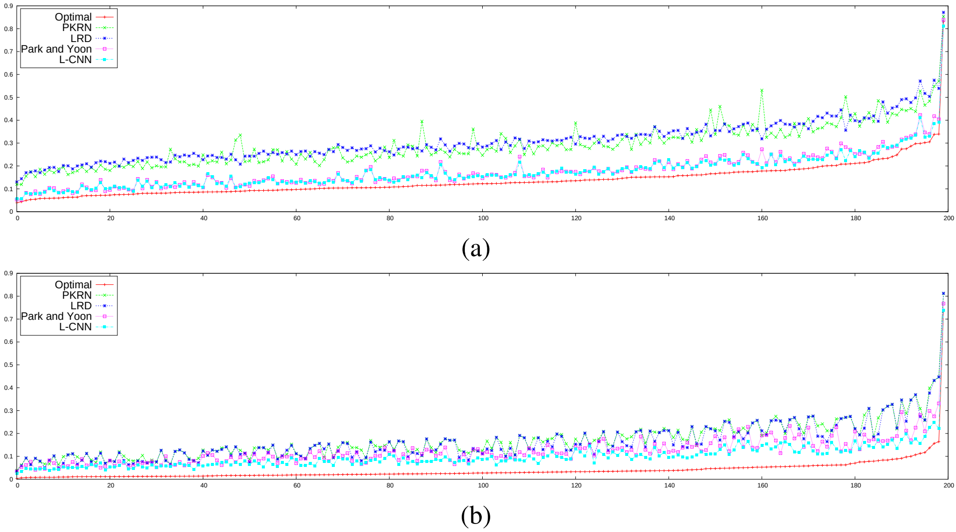


Figure 4: AUC values computed for the four CMs (PKRN, LRD, Park & Yoon and proposed CCNN) on KITTI 2015 dataset sorted in non-descending order according to optimal values (in red). The lower, the better. (a) BM algorithm. (b) SGM algorithm.

validation and cross-validation procedures. In particular, we obtained the best results with convolutional layers having a fan-in of 1 (i.e. , each kernel randomly takes as input one of the maps obtained from the previous layer), higher fan-in values did not lead to improvements. The two fully-connected layers are made of $L = 100$ neurons each (i.e., they are deployed as two 1×1 convolutional layers with 100 kernels each). During the training phase, we follow the Stochastic Gradient Descent (SGD) of the Binary Cross Entropy (BCE) between output o of the network and label t on each sample i of the mini-batch (2) by applying a sigmoid function S(x) (3) on the output of the network.

$$BCE(o,t) = -\frac{1}{n}\sum_i (t[i]\log(o[i])) + (1-t[i])(\log(1-o[i])) \quad (2)$$

$$S(x) = \frac{1}{1+e^{-x}} \quad (3)$$

We carried out 14 training *epochs*, with an initial learning rate of 0.003, increased by a factor 10 after the 10^{th} epoch, and a *momentum* of 0.9, inspired by [36] and confirmed by our experiments. To compare the confidence provided by our CNN with state-of-the-art, we also trained a RF as described in [27], adopting the full feature vector f_{22} described in the paper in order to obtain the best results. For a fair comparison with our proposal, we trained [27] on the same 50 images of the KITTI 2012 dataset.

4 Experimental results

Once trained[1] our CCNN approach on the 50 images of the KITTI 2012 dataset with the BM algorithm, in this section we assess its performance w.r.t. state-of-the-art with two datasets

[1]Source code and trained network publicly available at: http://vision.disi.unibo.it/~mpoggi

Dataset/Alg.	Opt.	PKRN	LRD	Park&Yoon	CCNN	CCNN vs Park&Yoon
KITTI/BM	0.137	0.294	0.308	0.179	**0.175**	-2.2% (119/200)
KITTI/SGM	0.038	0.171	0.162	0.124	**0.099**	-20.2% (183/200)
Middl./BM	0.093	0.165	0.170	0.114	**0.107**	-6.1% (13/15)
Middl./SGM	0.042	0.095	0.098	0.093	**0.074**	-20.4% (13/15)

Table 1: Average AUC on the validation datasets KITTI 2015 and Middlebury 2014 with BM and SGM algorithms. Average values closer to optimum are in bold. The last column shows, for our proposal, the average AUC improvements, in percentage, with respect to Park&Yoon [27] and the number of cases it performs better out the number of images in the dataset.

(KITTI 2015 and Middlebury 2014) and with two stereo algorithms, BM and SGM [11]. The top performing CMs considered in our evaluation are: Park and Yoon [27], trained on the same dataset and algorithm, and two conventional, yet effective, CMs described in [14] referred to as Left Right Difference (LRD) and Peak Ratio Naive (PKRN).

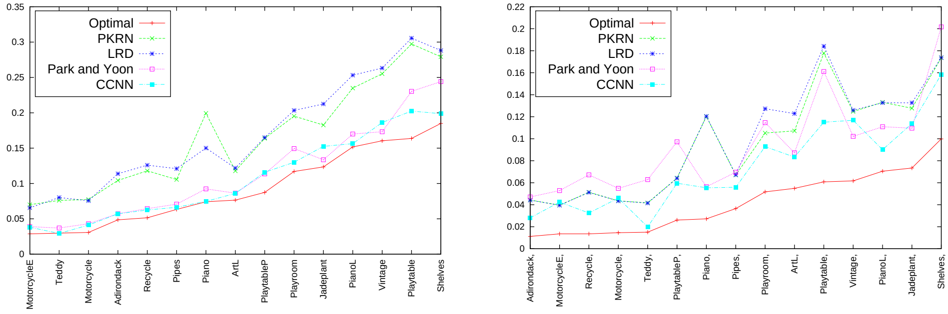
4.1 Evaluation methodology

In order to assess the performance of the CM inferred by our method with respect to state-of-the-art we rely on ROC curve analysis, as proposed in [14], which is a commonly adopted protocol when dealing with CMs. ROC curves are depicted, for each image, by sorting pixels according to decreasing confidence values. A subset of them equal to 5% of the total is sampled and the error rate is plotted, then the subset is increased to 10% of the total and so on until 100%. Ties are handled by taking into the subset all the points with the same confidence value. The Area Under the Curve (AUC) is then used to evaluate the capability of the confidence measure to distinguish correct disparity assignments from erroneous ones with respect to the optimal solution. Given the percentage of erroneous points ε, according to [14], the optimal AUC can be obtained as $\varepsilon + (1-\varepsilon)ln(1-\varepsilon)$. AUC closer to the optimal value reflects a better confidence prediction.

In the remainder we compare the same four CMs on KITTI 2015 training data, a dataset with a content similar to the one adopted for training, and a second cross-validation experiment on Middlebury 2014 training dataset (quarter resolution) [30] containing quite different scenes with respect to the other two datasets. In particular, this latter evaluation enables to further emphasize the ability of machine learning approaches, CCNN and Park & Yoon, to adapt not only to different algorithms but also to quite different scene content. The outcome of this evaluation is crucial to determine if these methods, once trained, can be used as out-of-the-box CMs.

4.2 Validation on KITTI 2015

We perform, on the same four CMs, a first validation phase on the KITTI 2015 dataset [22] containing 200 stereo pairs with ground-truth data. Figure 4 depicts, for BM (a) and SGM (b), AUC values for each stereo pairs belonging to the KITTI 2015 training set, sorted in non-descending order with respect to their optimal values. First of all, the figure shows that, with both stereo algorithms, approaches based on machine learning techniques have significantly better performance. Observing the top of the figure, concerned with BM, we can notice that the proposed CCNN approach obtains slightly better results, as summarized in the first row of Table 1, with respect to Park & Yoon outperforming it in 119 out of 200 cases. Moreover,



(a) (b)

Figure 5: AUC values computed for four CMs (PKRN, LRD, Park & Yoon and proposed CCNN) on Middlebury 2014 dataset sorted in non-descending order according to optimal values (in red). The lower, the better. (a) BM algorithm. (b) SGM algorithm.

when dealing with disparity maps characterized by higher error rates, CCNN frequently provides results very close to optimality. Observing the bottom of the figure and the second row of the table, concerned with SGM, we can notice that the CCNN better generalizes to different input data with respect to state-of-the-art outperforming Park & Yoon in 183 out of 200 cases with an average improvement greater than 20%. This indicates that our proposal is more independent of the matching algorithm, not being based on CV whose content is strictly related to the stereo algorithm adopted for training. Finally, LRD and PKRN behave similarly to the previous BM case. However, with SGM, Park & Yoon is outperformed by LRD or PKRN in 27 out of 200 cases while this never happens for CCNN.

We also tested architectures with a lower number of convolutional kernels (i.e., 32 and 48 for each convolutional layers) obtaining higher AUC values w.r.t. the proposed architecture. In particular, processing BM disparity maps, the network with 32 kernels achieves an average AUC of 0.423, with 48 kernels 0.227 and with the final network with 64 kernels 0.175. On the disparity maps provided by SGM, we report an average AUC of 0.234 with 32 kernels, 0.110 with 48 kernels and 0.099 with the proposed network.

4.3 Cross-validation on Middlebury 2014

In order to further stress the ability to generalize the performance of the considered CMs to more challenging conditions, we carried out a cross-validation on the Middlebury 2014 dataset [30] containing 15 stereo pairs with ground-truth. This dataset depicts indoor environments, completely different w.r.t. those of the training dataset (KITTI 2012) and of the previous testing dataset (KITTI 2015) both concerned with outdoor environments. As for previous evaluation we tested the four CMs with BM and SGM. Table 1, rows 3 and 4, summarizes the results reported in detail in Figure 5. With both stereo algorithms our method outperforms Park & Yoon in 13 out of 15 cases leading to an average improvement for BM and SGM, respectively, of 6.1% and 20.4%. Concerning BM, LRD and PKRN always provide worse results compared to approaches based on machine learning. On the other hand, although these latter approaches have similar performance CCNN performs better and in 4 cases out of 15 (Teddy, Pipes, Piano and PianoL) achieves results very close to optimality. With SGM, on average, LRD and PKRN provide worse results w.r.t. CCNN and Park & Yoon. However, Park & Yoon is significantly outperformed in 8 out of 15 cases by LRD or

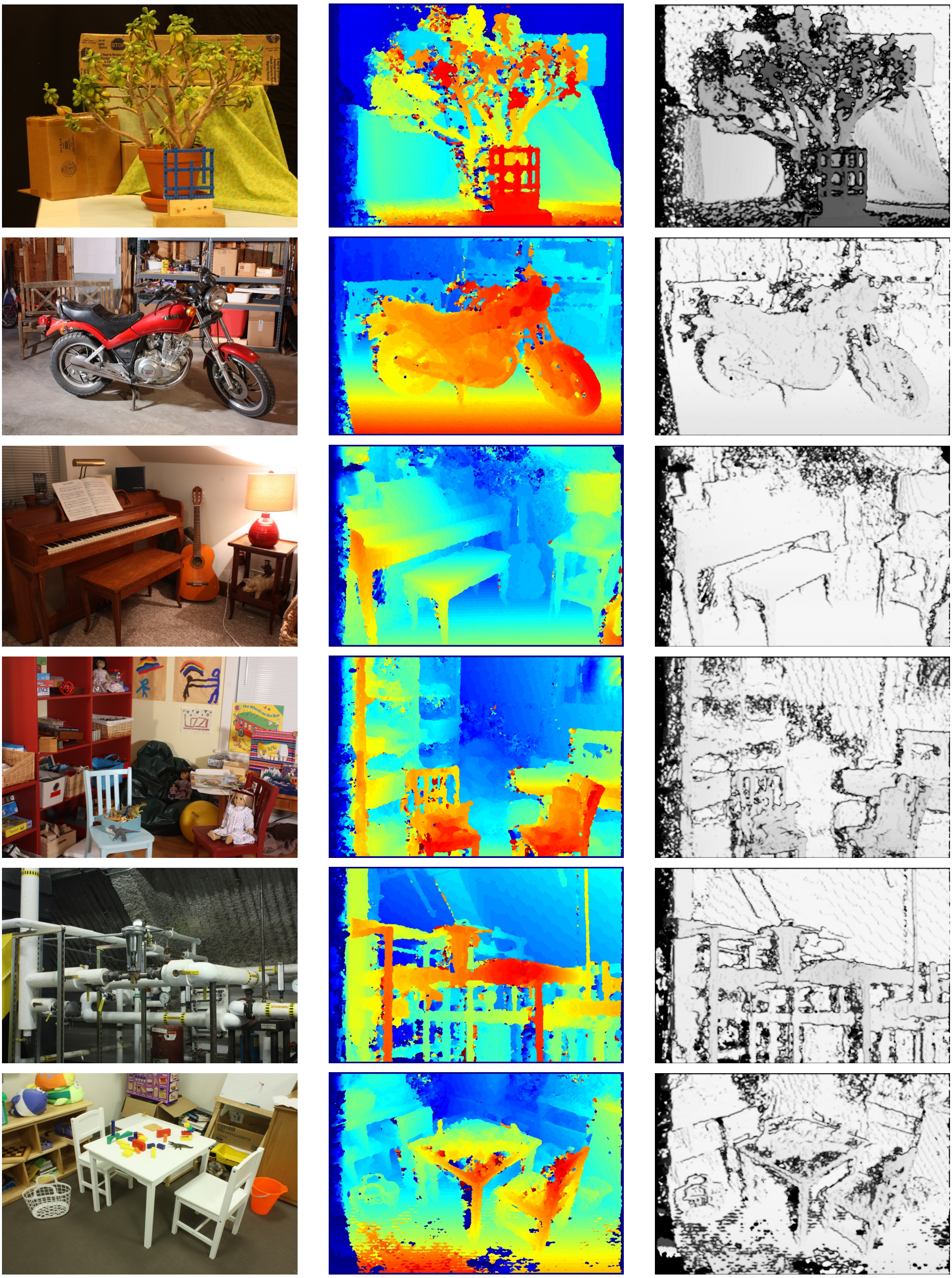


Figure 6: Output of CCNN cross-validation on three frames from Middlebury 2014 dataset, respectively *Jadeplant*, *Motorcycle* and *Playtable*. On left column: reference image, on central column: disparity map obtained from SGM algorithm (warmer color for closer points, colder for farther ones), on right column: confidence map yielded by CCNN.

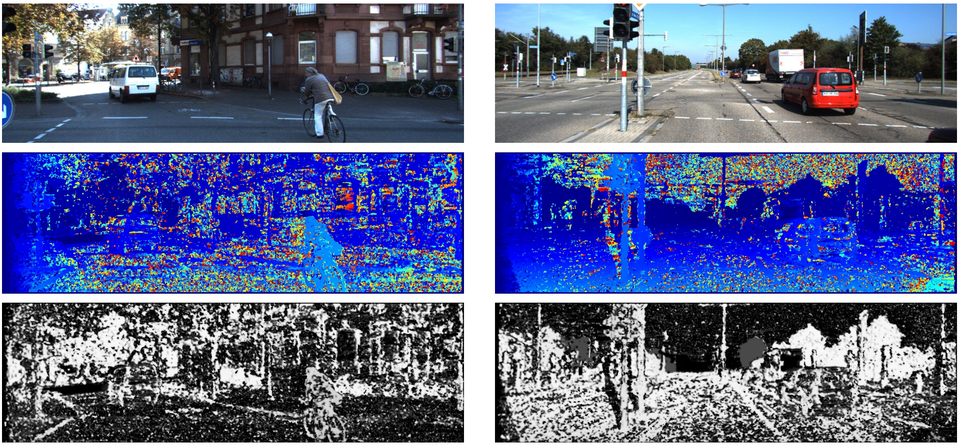


Figure 7: Output of CCNN validation on two frames from KITTI 2015 dataset, 000002 on left and 000051 on right. On top row: reference image, on central row: disparity map obtained from BM algorithm (warmer color for closer points, colder for farther ones), on bottom row: confidence map yielded by CCNN.

PKRN while slightly better results (MotorcycleE and Motorcycle) are obtained by these CMs w.r.t. CCNN in only 2 cases. The evaluation on Middlebury 2014 confirms that, compared to Park & Yoon, CCNN better generalizes to a different algorithm for the reason reported in the previous section.

As for the KITTI 2015 dataset, we provide experimental results with a lower number of convolutional kernels (i.e., 32 and 48 for each convolutional layers). Processing BM disparity maps, the network with 32 kernels achieves an average AUC of 0.367, with 48 kernels 0.159 and 0.107 with the final network with 64 kernels. On the disparity maps provided by SGM, we report an average AUC of 0.233 with 32 kernels, 0.117 with 48 kernels and 0.079 with the proposed network. These results confirm the trend previously reported on the KITTI 2015 dataset; a deeper analysis of this behaviour is left to future research.

Finally, Figure 6 and 7 depicts some examples of confidence maps generated by the proposed CCNN with 64 kernels outlined in Figure 3, respectively, on the Middlebury dataset with the SGM algorithm and on KITTI 2015 dataset with the BM algorithm.

5 Conclusions

In this paper, arguing that disparity assignments can be classified according to recurrent patterns detectable in the disparity map, we have proposed a novel confidence measure CCNN based on a deep architecture. Experimental results, including a cross validation on different datasets, clearly confirm that our proposal outperforms state-of-art. On a GPU, CCNN delivers confidence maps at almost 9 fps. Moreover, not being based on cost volume analysis, it is more independent of the particular stereo algorithm deployed for training and also suited for out-of-the-box stereo vision systems. To the best of our knowledge, this is the first method that allows to infer from scratch, using as input cue only the disparity map, an effective confidence measure exploiting a CNN.

References

[1] Zhuoyuan Chen, Xun Sun, Liang Wang, Yinan Yu, and Chang 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.

[2] Leonardo De-Maeztu, Stefano Mattoccia, Arantxa Villanueva, and Rafael Cabeza. Linear stereo matching. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 1708–1715, 2011.

[3] Geoffrey Egnal and Richard P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 24(8):1127–1133, 2002.

[4] Geoffrey Egnal, Max Mintz, and Richard P. Wildes. A stereo confidence metric using single view imagery. In *PROC. VISION INTERFACE*, pages 162–170, 2002.

[5] Frederic Garcia, Bruno Mirbach, Björn E. Ottersten, Frederic Grandidier, and Ángel Cuesta-Contreras. Pixel weighted average strategy for depth sensor data fusion. In *ICIP*, pages 2805–2808. IEEE, 2010.

[6] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Rob. Res.*, 32(11):1231–1237, sep 2013.

[7] R. Gherardi. Confidence-based cost modulation for stereo matching. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008.

[8] Ulrike Grömping. Variable importance assessment in regression: Linear regression versus random forest. *The American Statistician*, 63(4):308–319, 2009.

[9] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *CVPR. Proceedings*, pages 305–312, 2013. 1.

[10] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag.

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

[12] Dominik Honegger, Helen Oleynikova, and Marc Pollefeys. Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 4930–4935, 2014.

[13] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(2):504 – 511, 2013.

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

[15] Intel. Realsense camera. URL http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html.

[16] Nikos Komodakis, Georgios Tziritas, and Nikos Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[17] D. Kong and H. Tao. A method for learning matching errors in stereo computation. In *British Machine Vision Conference (BMVC)*, 2004 2004.

[18] W. Luo, A. G. Schwing, and R. Urtasun. Efficient Deep Learning for Stereo Matching. In *Proc. CVPR*, 2016.

[19] R. Manduchi and C. Tomasi. Distinctiveness maps for image matching. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 26–31. IEEE, 1999.

[20] Stefano Mattoccia and Matteo Poggi. A passive rgbd sensor for accurate and real-time depth sensing self-contained into an fpga. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, pages 146–151. ACM, 2015.

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

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

[23] Paul Merrell, Amir Akbarzadeh, Liang Wang, Jan michael Frahm, and Ruigang Yang David Nistér. Real-time visibility-based fusion of depth maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[24] Dong Bo Min and Kwanghoon Sohn. An asymmetric post-processing for correspondence problem. *Sig. Proc.: Image Comm.*, 25(2):130–142, 2010.

[25] Philippos Mordohai. The self-aware matching measure for stereo. In *The International Conference on Computer Vision (ICCV)*, pages 1841–1848. IEEE, 2009.

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

[27] Min-Gyu Park and Kuk-Jin Yoon. Leveraging stereo matching with learning-based confidence measures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[28] Neus Sabater, Andrés Almansa, and Jean-Michel Morel. Meaningful Matches in Stereovision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34 (5):930–42, dec 2011.

[29] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, apr 2002.

[30] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'03, pages 195–202, Washington, DC, USA, 2003. IEEE Computer Society.

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

[32] Stereolabs. Zed camera. URL https://www.stereolabs.com/.

[33] Christoph Strecha, Wolfgang von Hansen, Luc J. Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

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

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

[36] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016.