Even More Confident predictions with deep machine-learning

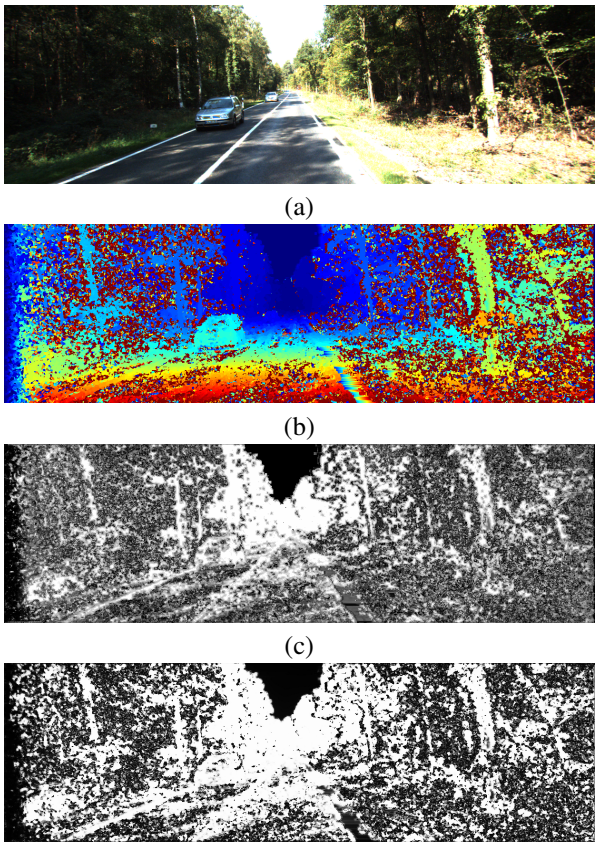
Matteo Poggi, Fabio Tosi, Stefano Mattoccia
University of Bologna
Department of Computer Science and Engineering (DISI)
Viale del Risorgimento 2, Bologna, Italy
matteo.poggi8@unibo.it, fabio.tosi5@unibo.it, stefano.mattoccia@unibo.it

Abstract

Confidence measures aim at discriminating unreliable disparities inferred by a stereo vision system from reliable ones. A common and effective strategy adopted by most top-performing approaches consists in combining multiple confidence measures by means of an appropriately trained random-forest classifier. In this paper, we propose a novel approach by training an n-channel convolutional neural network on a set of feature maps, each one encoding the outcome of a single confidence measure. This strategy enables to move the confidence prediction problem from the conventional 1D feature maps domain, adopted by approaches based on random-forests, to a more distinctive 3D domain, going beyond single pixel analysis. This fact, coupled with a deep network appropriately trained on a small subset of images, enables to outperform top-performing approaches based on random-forests.

1. Introduction

Stereo is a well-known methodology to estimate depth from multiple images. Although many algorithms dealt with this problem, with different degrees of effectiveness, performance in difficult environments characterized by specular or transparent surfaces, uniform regions, sunlight, etc remains an open research problems as clearly witnessed by recent datasets [25, 4, 15]. Therefore, regardless of the stereo algorithm, it is essential to detect its failures to filter-out wrong unreliable points that might lead to a not correct interpretation of depth data. To this aim, recent works focused on the formulation of meta-information capable to discriminate whether a disparity assignment has been correctly inferred by the stereo algorithm or not. Confidence measures encode this property by means of an estimated reliability score assigned to each pixel of the disparity map. Several measures obtained by processing different cues from the cost volume, disparity maps or input images have been proposed. Hu and Mordohai provided [10] an exhaustive review categorizing confidence measures according to the input features used, showing the strengths and weaknesses of each one. Following this observation, state-of-the-art approaches focused on combin-



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

Figure 1. Comparison between confidence measures obtained by [19] and by our proposal processing the same input features. (a) Left image, (b) disparity map, (c) confidence map computed by a random forest, (d) confidence map computed by our CNN-based method. In disparity maps, warm colors encodes closer points. In confidence maps brighter values encode more confident disparity.

ing multiple, possibly *orthogonal*, confidence measures by means of machine-learning frameworks based on random-forests.

These results, and the effectiveness of deep machine-learning applied to computer vision problems motivated us to inquire about the opportunity to achieve more accurate confidence estimation leveraging on Convolutional Neural Networks (CNNs). Figure 1, considering a sample from the KITTI 2015 dataset, shows the disparity map computed by a local stereo algorithm and two confidence maps obtained processing the same input features, respectively, by means of a state-of-the-art approach [19] based on a random-forest and our CNN-based proposal. We can observe from the figure how the confidence map obtained with deep-learning provides "*Even More Confident*" (EMC) predictions. In particular, the random-forest approach in (c) sets a large amount of points to intermediate scores being not sure enough about their actual reliability. On the other hand, our proposal (d) clearly depicts much more polarized scores. In section 4 we'll report quantitative results confirming the advantages yielded by our strategy.

Differently from approaches relying on random-forest classifiers that infer, for each point, an estimated match reliability by processing a 1D input feature vector made of point-wise confidence measures and features, our proposal relies on a more distinctive 3D input domain. Such input domain, for the point under analysis, is made of patches extracted from multiple input confidence and feature maps around the examined point as shown in Figure 2. Leveraging on a CNN, our proposal is able to infer more meaningful confidence estimations with respect to a random forest fed with the same input data. Doing so, our approach moves from the single pixel confidence strategy adopted by most state-of-the-art methods to a patch-based domain in order to exploit more meaningful local information.

We validate our method as follows. Once selected a subset of stereo pairs from the KITTI 2012 [4] training dataset, we run a fast local stereo algorithm, using as matching cost the census transform plus Hamming distance, a cost function common to previous works [19, 21]. From the outcome of the previous phase we compute a pool of confidence measures and features training a random forest and our CNN framework on such data. In particular, we choose as input confidence measures and features the same adopted by state-of-the-art methods [27], [19] and [21] based on random-forest frameworks. Then, we evaluate the effectiveness of our proposal with respect to [27], [19] and [21] by means of ROC curve analysis [10], on the remaining portion of KITTI 2012. Moreover, we cross-validate without re-training on KITTI 2015 and Middlebury 2014.

2. Related work

Stereo has been tackled, with different degrees of effectiveness, by many works in literature. Almost any algorithm deployed to address it belongs to one of the two categories defined by Scharstein and Szeliski [24]: local and global methods. Currently, most state-of-the-art stereo pipelines [4, 15] leverage on the point-wise matching cost MC-CNN [28] inferred on image patches with a CNN and by refining the obtained cost volumes with adaptive local cost aggregation and Semi-Global Matching (SGM). Concerning CNN-based stereo algorithms, Chen et al. [1] and Luo et al. [12] follow a similar strategy. Conversely, Mayer et al. [14] proposed a deep architecture for end-to-end disparity estimation.

In this field, detecting wrong assignments is important for different purposes and in particular to improve overall disparity accuracy in challenging conditions. This is carried out exploiting confidence measures that, with different formulations and effectiveness, allow to estimate match reliability. Hu and Mordohai [10] reviewed, evaluated and categorized such measures according to the input cues: matching cost, local properties of cost curve, local minima, entire cost curve, left-right consistency between disparity maps and distinctiveness. They report a complete benchmark, by defining a protocol based on ROC curve analysis, deploying different matching cost functions and evaluating confidences for different tasks such as detection of correct matches, occlusions and disparity selection. In addition to their standard deployment, confidence measures proved to be very effective for others purposes. In [8, 17] for occlusion detection, in [23] for error detection, and in [13, 16] to combine depth data from multiple sensors. Moreover, such measures can also be used to improve disparity accuracy by enhancing the raw cost curve [20, 18, 5, 27, 19]. These methods turned out to be very effective when dealing with very challenging scenarios as reported in [19].

A recent trend concerning confidence measures consists in improving the effectiveness of stand-alone approaches within machine-learning frameworks. Hausler et al. [6] proposed to train a random forest classifier, fed with a set of stand-alone confidences and features computed at different scales, to distinguish correct matches from wrong ones. Inspired by the results yielded by such strategy, in other works the problem was addressed similarly such as in [27] and [19] enabling to obtain results closer to optimality. Both methods also proposed original methodologies, driven by confidence measure, to improve the accuracy of stereo algorithms. In [27], by detecting a subset of reliable *ground control points* processed by a global optimization framework [11]. In [19], by modulating raw cost curve before aggregating them with methods based on the guided-filter [7], [9, 3], or performing a disparity optimization with SGM. Moreover, in [21] a random forest classifier has been

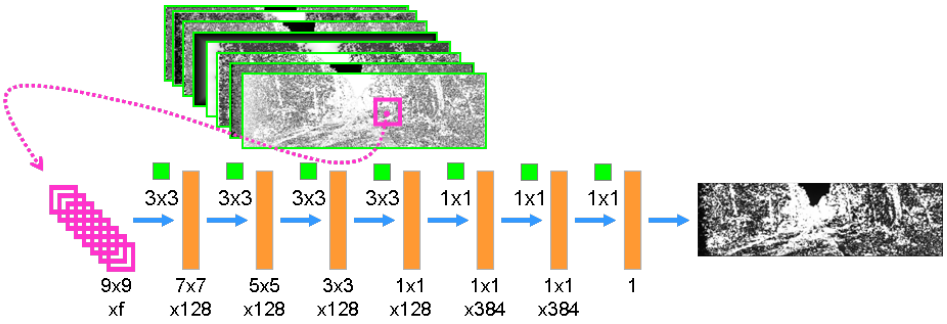


Figure 2. Architecture of CNN with highlighted in purple the confidence measures and features processed in a 3D domain by our method.

trained only on features obtained from the disparity map, making the entire cost volume no longer required to effectively predict the reliability of each pixel, proving to outperform [19] and establishing as the most effective confidence measure based on random forest. Moreover, this latter measure has been deployed to improve SGM results by weighting the contribution of the different scanlines according to the confidence of their respective WTA maps. In this field, Mostegel et al. in [2] proposed a process to generate disparity labels exploiting multiple view points and contradictions between depth maps, in order to perform unsupervised training of confidence measures based on machine-learning [6, 27, 19]. Finally, more recent deep-learning based confidence measures have been proposed. In particular, Seki and Pollefeys [26] deployed a CNN inferring confidence by working on patches obtained from left and right disparity maps, while Poggi and Mattoccia in [22] trained a deep architecture to predict confidence only from the reference disparity map.

3. Deep learning for confidence measures

In this work, we follow the successful strategy of combining multiple confidence measures through supervised learning, by exploiting CNN. Such solution greatly increases the amount of information processed when predicting confidence with respect to conventional random-forest classifiers. In particular, by processing confidences and other hand-crafted features as images, our approach moves from the 1D features domain of the random forest classifiers to a more distinctive 3D domain, encoding local behavior of features and, thus, going beyond single pixel confidence analysis . Two dimensions are given by the image domain and one by the features domain as shown in Figure 2.

3.1. Hand-crafted features layer

In [6] the random-forest classifier is fed with a feature vector F containing f different features, obtained according to f functions (*e.g.*, multiple confidence measures computed at different scales). Although this strategy and the others inspired by this method [27, 19, 21] enabled remarkable improvements, the random forest classifier takes as input a 1D feature domain made of elements of F, encoding pixel-wise properties.

By moving into the deep learning domain, we can imagine this feature vector F as a set of f general purpose feature maps that might be generated by a generic convolutional layer C_i and fed as input to the following one C_{i+1}. According to this observation, we model our framework as a CNN with a first layer H in charge of extracting a set of hand-crafted feature maps. Excluding the front-end layer H, the remaining portion of the deep architecture is trained according to the number input feature maps provided by such layer. For example, adopting the same input features of [27] in our framework, the H front-end would provide to the first convolutional layer of the deep network the following eight feature maps described in [27]: MSM, MMN, AML, LRC, LRD, distance to border, distance to discontinuities and median deviation of disparity.

3.2. Deep network architecture

This section describes the design of the architecture proposed to infer a learned confidence measure. Excluding the H front-end, in charge of providing multiple feature maps from the available input cues (*e.g.*, cost curve, disparity maps, etc), we rely on a deep-network architecture made of 7 convolutional layers trained to infer a point-wise confidence measure processing 3D input features. Specifically, we deploy a patch-based fully-convolutional architecture, as shown in Figure 2.

A patch-based approach, as proposed in [28, 22], requires a significantly lower amount of data for training compared to an end-to-end deep network architecture working on full-resolution images like the one proposed in [14]. In fact, in this second case, the dataset required to train such deep-network for the same purpose would be much more larger. Considering this fact, our model is made of four convolutional layers, each one followed by Rectifier Linear Units (ReLU). Each layer applies 128 kernels of size 3×3, applied to each pixel (stride equal to 1). Two additional

convolutional layers, made of 384 1×1 kernels followed by ReLU, increase the amount of extracted features, leading to the final output layer. This model counts more than one half million parameters and was chosen in our experiments, after a preliminary testing, as the one yielding more accurate results. According to this architecture, a single point-wise confidence measure is obtained by processing a 9×9 perceptive field after the front-end H. According to Figure 2 this means that the 3D input domain processed by our network has size $9 \times 9 \times f$.

Being our architecture a fully-convolutional model, any input of size greater than the perceptive field can be processed by the network. This means that it is capable of computing a full resolution confidence map by processing the feature maps forwarded by the H front-end. The deep network, excluding H, performs on a full-resolution KITTI 2012 image a confidence prediction in a few seconds on an i7 CPU, dropping to 0.8 seconds with a Titan X GPU, with an overall memory footprint of about 4.5 GB.

4. Experimental Results

To evaluate our proposal, we feed our network with multiple stand-alone confidence measures and hand-crafted features comparing the results with state-of-the-art confidence measures [27, 19, 21] based on random-forest frameworks. We perform a single training on a portion of the KITTI 2012 dataset (25 out of 194 total images), then we test the methods on the remaining stereo pairs available, deployed as evaluation set. Moreover, we further cross-validate the confidence measures on KITTI 2015 (200 images) and Middlebury 2014 datasets (15 images). We will release source code and trained networks on a public repository.

4.1. Training phase

We trained our network according to *stochastic gradient descend*, we choose the *binary cross entropy* as loss function, according to the regression problem we are dealing with. We trained on nearly 3.5 million samples, obtained from the first 25 stereo pairs of the KITTI 2012 training dataset. Each sample corresponds to a volume of $9 \times 9 \times f$ patches output of the H layer, each one centered on a pixel with provided ground-truth available in the dataset. We define a batch size of 128 training samples, training for 5 *epochs*, corresponding to nearly 135 thousand iterations, with a 0.002 learning rate and 0.8 momentum. We applied training samples shuffling.

The stereo algorithm used to generate matching costs for the training phase consists of a 5×5 census based data term, aggregated on a fixed local window of size 5×5. We set as error threshold the value 3, commonly adopted to compute the error rate of the stereo algorithms on the most popular datasets [4, 15]. Samples concerning pixels with a disparity assigned by the fixed window aggregation lower than the threshold are labeled with high confidence (1 values). For a fair evaluation, we compare the proposed methodology with random-forests trained on the same amount of data. In our experiments, we choose [27], [19] and [21], representing state-of-the art confidence measures inferred by random-forest frameworks. During the validation, these three methods will be referred to as, respectively,

- GCP (Ground Control Point) [27], processing a feature vector of cardinality 8 by means of a random-forest. Such vector contains MSM, MMN, AML, LRC, LRD confidence measures reviewed in [10], DTB (distance to border), DTD (distance to discontinuities) and MED (median deviation of disparity) computed on a 5×5 patch.

- LEV (Leveraging-Stereo) [19], processing a feature vector of cardinality 22 by means of a random-forest. The vector contains PKR, PKRN, MSM, MMN, WMN, MLM, NEM, LRD, CUR and LRC confidence measures reviewed in [10], PER confidence measure proposed in [6], DTBL (distance to left border), DTE (distance to edges), HGM (horizontal gradient magnitude), MED (median deviation of disparity) and VAR (variance of disparity) on 5×5, 7×7, 9×9 and 11×11 neighborhood.

- O1 (O1) [21], processing a feature vector of cardinality 20 by means of a random-forest. The vector contains DA (disparity agreement), DS (disparity scattering, median disparity, VAR (variance of disparity) and MED (median deviation of disparity), each one computed on 5×5, 7×7, 9×9 and 11×11 neighborhood.

4.2. EMC vs random-forest

A common procedure to evaluate the effectiveness of a confidence measure is the ROC curve analysis, proposed by Hu and Mordohai [10] and adopted by subsequent works [6, 27, 19, 22]. The ROC curve is drawn by iterative sub-sampling of pixels from the image, according to descending order of confidence. Starting from a small subset of points (*i.e.*, 5% most confident), the error rate on such group is plotted, then more pixels are included into the subset and the new error is plotted, and so on until all pixels have been included into the set. This leads to a non-monotonic curve, whose area (AUC) is an indicator of the effectiveness of the confidence measure. Given a disparity map with $\varepsilon\%$ wrong pixels, an optimal confidence measure should draw a curve which is zero until $\varepsilon\%$ pixels have been sub-sampled. The area of this curve represents the optimal AUC achievable by a confidence measure and can be obtained, according to [10], as $AUC_{opt} = \int_{1-\varepsilon}^{\varepsilon} \frac{p-(1-\varepsilon)}{p} dp = \varepsilon + (1-\varepsilon) \ln(1-\varepsilon)$

To be compliant with the training protocol, ε is obtained by fixing a threshold value on disparity error of 3.

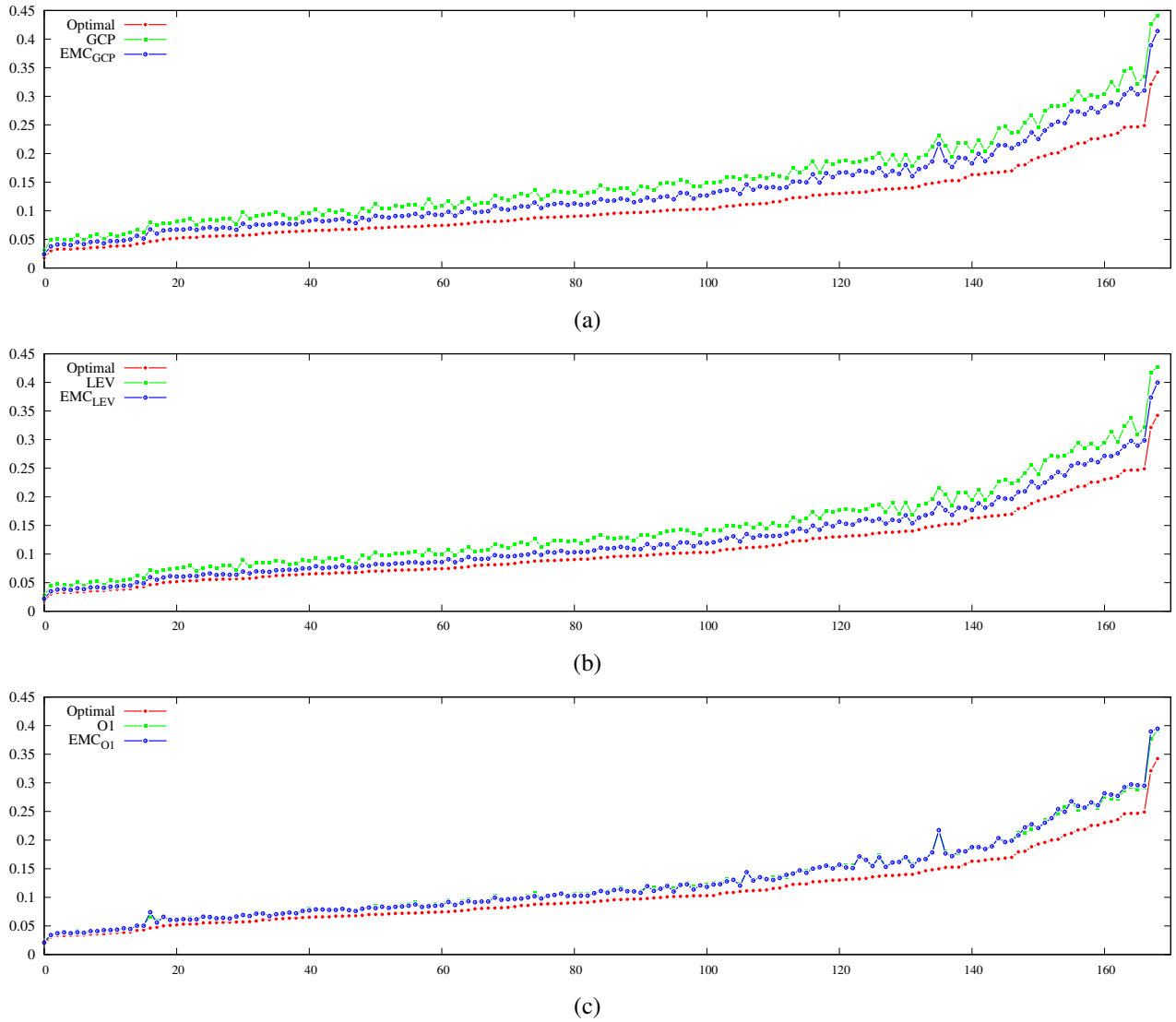


Figure 3. AUC values on the KITTI dataset. Each value on the plot represent the AUC on a single image of the dataset, sorted in non-descending order according to their optimal values. We report, from top to bottom, comparison between GCP and EMC_{GCP} (a), LEV and EMC_{LEV} (b), O1 and EMC_{O1} (c). Cost volumes obtained by census based fixed window algorithm.

Figure 3 depicts three plots, containing the AUC values computed over the entire KITTI 2012 (excluding the images processed during training) of both the EMC approach and the corresponding random forest counterpart, for GCP [27], LEV [19], O1 [21]. The curves are plotted in non-descending order according to optimal values (red), together with curves related to random forest implementation (referred to as GCP, LEV and O1, plotted in green) and our method processing the same inputs (referred to as EMC_{GCP}, EMC_{LEV} and EMC_{O1}, plotted in blue). In particular, from top to bottom, (a) concerns with GCP versus EMC_{GCP}, (b) with LEV versus EMC_{LEV}, (c) with O1 vs EMC_{O1}. As we can observe, for the first two experiments the EMC implementations achieves lower AUC values, thus closer to optimal values. From the AUC curve, it's evident how the EMC framework outperforms the random forest on each image of the dataset. Concerning O1, our implementations performs very similarly to the original proposal [21], but on average it achieves a better AUC on the entire dataset.

Figure 4 depicts the three plots for the entire KITTI 2015, comparing the EMC approach with the corresponding random forest counterpart, for GCP [27], LEV [19], O1 [21]. Optimal values are plotted in red, curves related to random forest implementation (referred to as GCP, LEV and O1, plotted in green) and our method processing the same inputs (referred to as EMC_{GCP}, EMC_{LEV} and EMC_{O1}, plotted in blue). In particular, top graph (a) concerns with GCP versus EMC_{GCP}, the second one (b) with LEV versus

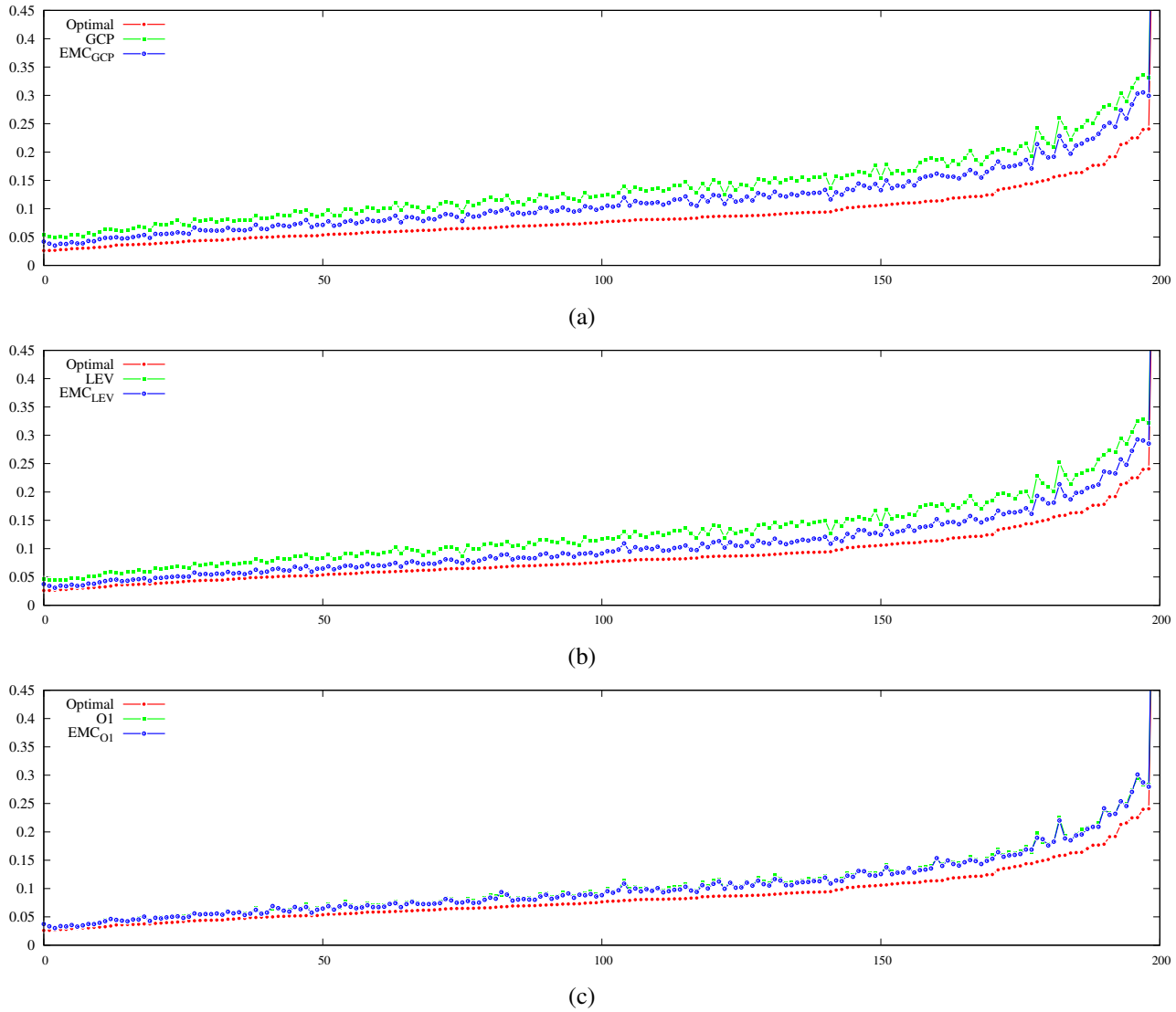


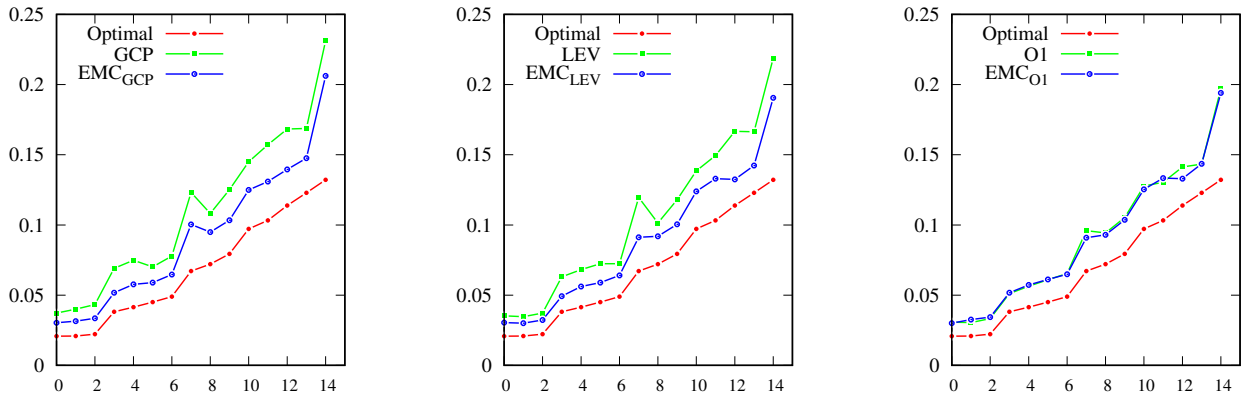
Figure 4. AUC values on the KITTI 2015 dataset. Each value on the plot represent the AUC on a single image of the dataset, sorted in non-descending order according to their optimal values. We report, from top to bottom, comparison between GCP and EMC_{GCP} (a), LEV and EMC_{LEV} (b), O1 and EMC_{O1} (c). Cost volumes obtained by census based fixed window algorithm.

EMC_{LEV}, the final (c) with O1 vs EMC_{O1}. The behavior observed on KITTI 2012 is confirmed, GCP and LEV features achieve major improvements when processed within EMC framework with respect to random forest, while we can observe a minor improvement concerning O1.

Figure 5 shows three plots concerning the evaluation on the Middlebury 2014 dataset. As for the previous figures, optimal values are plotted in red, curves related to random forest implementation are in green (referred to as GCP, LEV and O1) and those related to EMC processing the same inputs (referred to as EMC_{GCP}, EMC_{LEV} and EMC_{O1}). In particular, from left to right, (a) concerns with GCP versus EMC_{GCP}, (b) with LEV versus EMC_{LEV}, (c) with O1 vs EMC_{O1}. The three confidence measures confirm the behaviors already highlighted on the KITTI datasets.

To further perceive the improvements lead by our framework (and, concerning O1, to highlight its behavior more clearly), we report AUC values averaged over each of the three datasets for the three confidence measures, for both random forest and EMC implementations. We report two aspects allowing for such comparison. The first is the variation of average AUC achieved by EMC implementation of confidence measure k with respect to its random forest counterpart and optimal value, referred to as Δ_k and obtained as:

$$\Delta_k = \frac{AUC_k - AUC_{EMC_k}}{AUC_k - AUC_{opt}} \quad (1)$$



(a) (b) (c)

Figure 5. AUC values on the Middlebury dataset. Each value on the plot represent the AUC on a single image of the dataset, sorted in non-descending order according to their optimal values. We report, from top to bottom, comparison between GCP and EMC_{GCP} (a), LEV and EMC_{LEV} (b), O1 and EMC_{O1} (c). Cost volumes obtained by census based fixed window algorithm.

	KITTI 2012			KITTI 2015			Middlebury 2014		
	GCP	LEV	O1	GCP	LEV	O1	GCP	LEV	O1
Optimal	0.107802			0.088357			0.068375		
RF	0.152764	0.144077	0.127645	0.139611	0.131662	0.108812	0.109302	0.104146	0.090908
EMC	0.133684	0.125211	0.126898	0.117551	0.107969	0.106523	0.091749	0.088473	0.089928
Δ_k	-42.44%	-52.01%	-3.76%	-43.04%	-54.71%	-11.19%	-42.88%	-43.81%	-4.35%

Table 1. Average AUC values on the three dataset, KITTI 2012, KITTI 2015 and Middlebury from left to right respectively. First row reports optimal AUC values according to [10], second row shows values concerning the random forest implementation of GCP [27], LEV [19] and O1 [21], third row shows results achieved by EMC implementation. Final row shows the improvement Δ_k led by EMC with respect to optimal AUC values. Cost volumes obtained by census based fixed window algorithm.

	KITTI 2012		
	GCP	LEV	O1
EMC win rate	169/169	169/169	122/169
	KITTI 2015		
	GCP	LEV	O1
EMC win rate	200/200	200/200	181/200
	Middlebury 2014		
	GCP	LEV	O1
EMC win rate	15/15	15/15	8/15

Table 2. EMC win rate on the three dataset, KITTI 2012, KITTI 2015 and Middlebury (*i.e.*, number of images per dataset on which EMC outperforms the random forest) from top to bottom respectively. First row reports optimal AUC values according to [10], second row shows values concerning the random forest implementation of GCP [27], LEV [19] and O1 [21], third row shows results achieved by EMC implementation. Final row shows the improvement Δ_k led by EMC with respect to optimal AUC values. Cost volumes obtained by census based fixed window algorithm.

Negative values of this variation reflects an improvement achieved by EMC, while positive stand for a worse confidence prediction. The second is the win rate, as the number of images on which EMC achieves a lower AUC with respect to its random forest counterpart. Table 1 reports average AUC for each confidence measure (GCP, LEV, O1) on the three datasets KITTI 2012, KITTI 2015 and Middlebury. The first row reports optimal AUC, according to [10], averaged over each dataset, then AUC concerning both implementations (referred to as, respectively, RF for random forest, EMC for our approach). Finally, Δ_k highlights the effectiveness of the CNN with respect to the random forest. We can observe how on the KITTI 2012 dataset the improvement yielded by our method is, concerning GCP and LEV, higher than 40%, respectively, 42.44% with respect to GCP and 52.01% with respect to LEV. These results are confirmed on the KITTI 2015 dataset, reporting Δ_k very close to the previous ones, and on Middlebury 2014, on which LEV achieve a lower, yet important Δ_k value. Focusing on O1, the improvement is lower, between 3% and 12% (the higher is on KITTI 2015, -11.19%) on the three datasets. This may be caused by the higher accuracy of the random forest implementation compared to GCP and LEV solutions, or to the nature of the features extracted by O1, all processed from the disparity map only and, probably, encoding less different behaviors with respect to GCP and LEV features. Nonetheless, on average with O1, EMC is more effective than the random forest counterpart. Table 2 reports the win rate achieved by EMC for each confidence

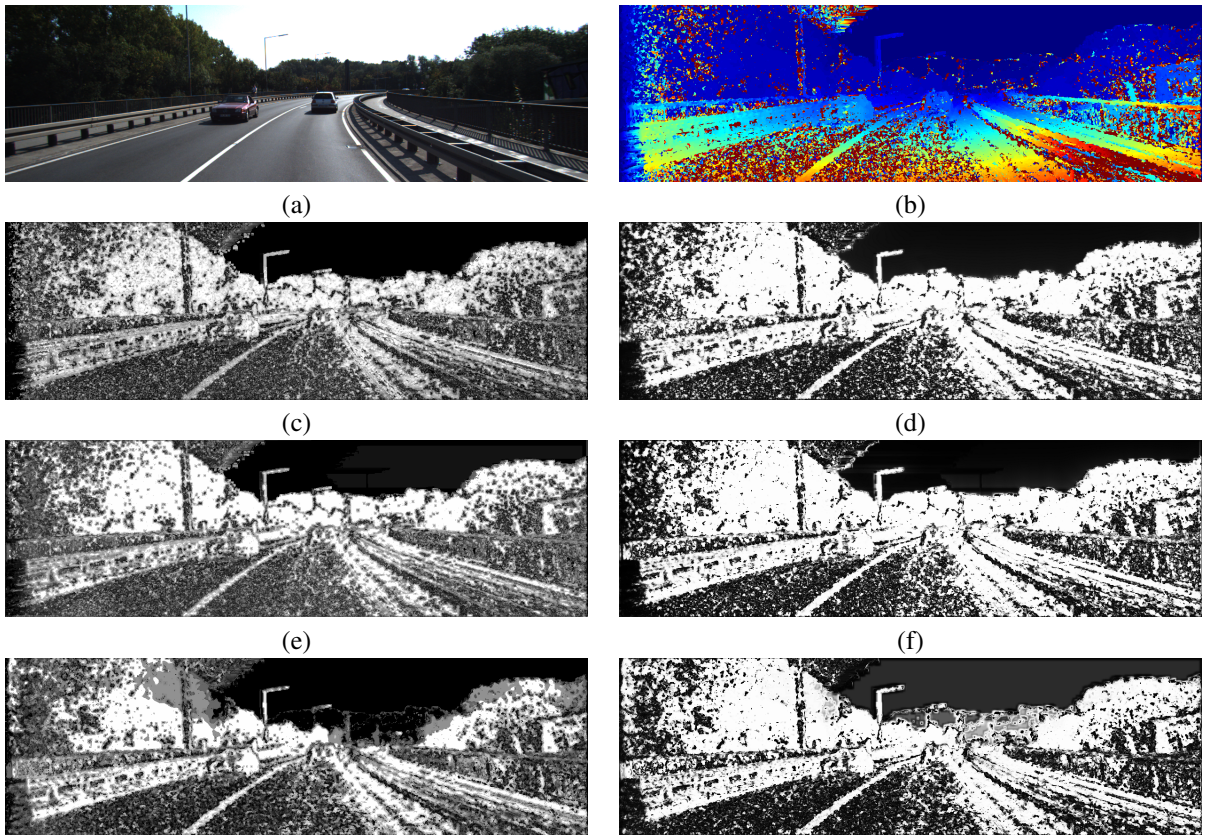


Figure 6. Confidence maps obtained by random forest and EMC. The disparity map is concerned with the considered stereo algorithm on pair 000176 of the KITTI 2015 dataset. Reference image (a) , disparity map (b), confidence map obtained by GCP [27] using a random forest (c) and EMC (d), confidence map obtained by LEV [19] using a random forest (e) and EMC (f), confidence map obtained by O1 [21] using a random forest (g) and EMC (h).

measure on the three datasets. While EMC outperforms random forests on all the stereo pairs of the three datasets for GCP and LEV (*i.e.*, 100% win rate), it wins 122 out of 169 times on KITTI 2012, 181 out of 200 on KITTI 2015 (confirming to be more effective on this dataset) and 8 out of 15 on Middlebury for O1, confirming to be less effective, but still outperforming random forest implementation on average. We would like to point-out that the training procedure did not take into account any of the KITTI 2015 nor Middlebury 2014 data for random forest approaches and EMC. This evaluation proves how the effectiveness of the CNN-based proposal implementation result is kept processing different data. This fact (*i.e.*, the capability to generalize to new data) represents a notable result for a machine-learning framework. Finally, Figure 6 reports a qualitative comparison of confidence maps obtained by random forest and EMC, respectively, with GCP (c,d), LEV (e,f) and O1 (g,h), for a stereo pair from KITTI 2015 dataset.

5. Conclusions

In this paper we tackled the confidence prediction problem exploiting a deep network to combine multiple confidence and feature maps. Differently from state-of-art approaches based on random-forest framework processing input features in a 1D domain, our proposal relies on more distinctive features in the 3D domain enabling to extract more effective confidence predictions. Extensive experimental results show that our proposal improves the effectiveness of top-performing approaches based on random-forest when fed with the same input features and trained on the same amount of data.

Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 972–980, 2015. 2

[2] M. Christian, R. Markus, F. Friedrich, and B. Horst. Using self-contradiction to learn confidence measures in stereo vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[3] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza. Linear stereo matching. In *A13th International Conference on Computer Vision (ICCV2011)*, November 6-13 2011. 2

[4] 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. 1, 2, 4

[5] R. Gherardi. Confidence-based cost modulation for stereo matching. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008. 2

[6] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *CVPR. Proceedings*, pages 305–312, 2013. 1. 2, 3, 4

[7] K. He, J. Sun, and X. 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. 2

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

[9] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(2):504 – 511, 2013. 2

[10] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 2121–2133, 2012. 1, 2, 4, 7

[11] N. Komodakis, G. Tziritas, and N. 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. 2

[12] W. Luo, A. G. Schwing, and R. Urtasun. Efficient Deep Learning for Stereo Matching. In *Proc. CVPR*, 2016. 2

[13] G. Marin, P. Zanuttigh, and S. Mattoccia. Reliable fusion of tof and stereo depth driven by confidence measures. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, pages 386–401, 2016. 2

[14] N. Mayer, E. Ilg, P. Häusser, 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[15] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2, 4

[16] P. Merrell, A. Akbarzadeh, L. Wang, J. michael Frahm, and R. Y. D. Nistr. Real-time visibility-based fusion of depth maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[17] D. B. Min and K. Sohn. An asymmetric post-processing for correspondence problem. *Sig. Proc.: Image Comm.*, 25(2):130–142, 2010. 2

[18] P. Mordohai. The self-aware matching measure for stereo. In *The International Conference on Computer Vision (ICCV)*, pages 1841–1848. IEEE, 2009. 2

[19] M.-G. Park and K.-J. Yoon. Leveraging stereo matching with learning-based confidence measures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2, 3, 4, 5, 7, 8

[20] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. In *IEEE Computer Vision and Pattern Recognition*, pages 297–304, Portland, OR, USA, June 2013. 2

[21] M. Poggi and S. Mattoccia. Learning a general-purpose confidence measure based on o(1) features and asmarter aggregation strategy for semi global matching. In *Proceedings of the 4th International Conference on 3D Vision, 3DV*, 2016. 2, 3, 4, 5, 7, 8

[22] M. Poggi and S. Mattoccia. Learning from scratch a confidence measure. In *Proceedings of the 27th British Conference on Machine Vision, BMVC*, 2016. 3, 4

[23] N. Sabater, A. Almansa, and J.-M. Morel. Meaningful Matches in Stereovision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(5):930–42, dec 2011. 2

[24] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, apr 2002. 2

[25] D. Scharstein and R. 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. 1

[26] A. Seki and M. Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016. 3

[27] A. Spyropoulos, N. Komodakis, and P. 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. 2, 3, 4, 5, 7, 8

[28] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016. 2, 3