Learning to predict stereo reliability enforcing local consistency of confidence maps

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

Abstract

Confidence measures estimate unreliable disparity assignments performed by a stereo matching algorithm and, as recently proved, can be used for several purposes. This paper aims at increasing, by means of a deep network, the effectiveness of state-of-the-art confidence measures exploiting the local consistency assumption. We exhaustively evaluated our proposal on 23 confidence measures, including 5 top-performing ones based on random-forests and CNNs, training our networks with two popular stereo algorithms and a small subset (25 out of 194 frames) of the KITTI 2012 dataset. Experimental results show that our approach dramatically increases the effectiveness of all the 23 confidence measures on the remaining frames. Moreover, without re-training, we report a further cross-evaluation on KITTI 2015 and Middlebury 2014 confirming that our proposal provides remarkable improvements for each confidence measure even when dealing with significantly different input data. To the best of our knowledge, this is the first method to move beyond conventional pixel-wise confidence estimation.

1. Introduction

Stereo is a popular technique to infer depth from two or more images and several approaches have been proposed to tackle this problem. However, reliability in challenging conditions still remains an open research issue and realistic datasets, such as KITTI [7, 17] and Middlebury 2014 [29], clearly emphasized this fact. Although some failures such as occlusions [5], low signal-to-noise ratio and reduced distinctiveness [14] are intrinsically related to stereo, the impact on accuracy is amplified in practical applications dealing with poor illumination conditions, reflective surfaces and so on. Therefore, determining the degree of reliability of each inferred depth point is crucial to obtain more meaningful 3D data for later processing stages. Moreover, effective confidence measures can be used for other purposes. For instance, to improve stereo accuracy [6, 23, 22, 25, 30] or for depth sensor fusion [15, 18].

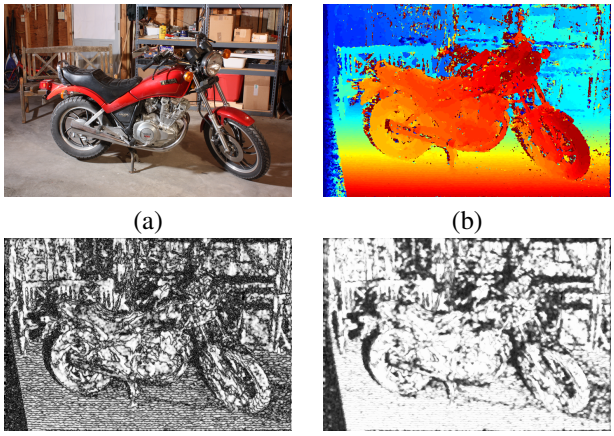


Figure 1. Confidence prediction performed by our approach processing PKRN [11]. (a) Reference image, (b) Disparity map, (c) Original PKRN measure, (d) Corresponding confidence map, PKRN$^+$, computed by our framework. More confident points are encoded with brighter values, disparity map with colormap jet.

Confidence measures, reviewed and evaluated in [11], are inferred according to different strategies: from the analysis of the input stereo pair, matching cost curve or disparity maps. Recently, some authors [9, 31, 22, 25] proposed effective confidence measures based on machine learning techniques. The common ground in these approaches is the joint use of multiple confidence measures and/or handcrafted features, extracted from disparity map and/or cost volume, fed to a random-forest classifier trained on a small set of stereo pairs with ground truth. More recently, confidence measures have been inferred [26, 30] processing disparity maps with a CNN (Convolutional Neural Network).

These facts motivated us to investigate whether a machine learning framework could be used to improve the effectiveness of confidence measures exploiting local consistency, leveraging on the information available within nearby points, as assumed by most computer vision algorithms. To this end, given an input confidence measure, our framework analyzes its local behavior by means of a CNN, trained on a subset of a dataset with ground-truth, to provide a more meaningful estimation. Specifically, by learning informative patterns on confidence maps, the network is able to infer from local patches a better estimation as shown in Figure 1. In our experimental evaluation, we consider 23 state-of-the-art confidence measures and, once trained the networks on 25 out of 194 images of the KITTI 2012 (KITTI 12) training dataset, we assess the improvements yielded by our method on the remaining images. Moreover, without retraining the networks, we perform a further cross-validation on KITTI 2015 (KITTI 15) and Middlebury 2014 (Middlebury 14). This extensive evaluation shows that exploiting local consistency enables to dramatically improve all the 23 state-of-the-art confidence measures, including those based on machine learning, on all considered datasets and even dealing with image contents never *seen* before (*e.g.*, on Middlebury 14 dataset).

To the best of our knowledge, this is the first method to exploit for confidence measures the local consistency assumption moving beyond conventional point-based strategy adopted by state-of-the-art. Experimental results, with the end-to-end CNN-based framework proposed in this paper, clearly confirm the effectiveness of this strategy.

2. Related work

Many confidence measures for stereo have been proposed in the literature [4, 5, 11]. In the review proposed by Hu and Mordohai [11], such measures are categorized into six main groups according to the to cue exploited to infer depth 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. The same authors also defined an evaluation protocol based on ROC curve analysis and reported results on indoor [28] and outdoor [32] datasets.

Confidence measures can be used for several purposes; for instance to detect uncertain disparity assignments [23, 27] and occlusions [10, 19], improve accuracy near depth discontinuities [6], improve overall disparity map accuracy [12, 20, 8, 22, 25] and for sensor fusion [15, 18]. More effective confidence measures, leveraging on machine learning techniques, significantly outperform conventional stand-alone approaches evaluated in [11]. In particular, in [9, 31, 22, 25] the reliability of disparity assignments is inferred by feeding a random forest with a features vector containing multiple confidence measures [9, 31, 22] and/or hand-crafted clues extracted from the disparity map [31, 22, 25]. Compared to stand-alone confidence measures, *Ensemble* [9], *GCP* [31], *Park* [22] and *O1* [25] achieved significant improvements with O1, based on features extracted only from the disparity map, outperforming other methods based on random-forests [25].

Deep learning techniques have also been recently deployed to deal with confidence prediction and stereo matching. Concerning the first goal, in [30] a confidence measure is inferred with a CNN analyzing hand-crafted features extracted from *left-right* and *right-left* disparity maps. In [26] this abstraction strategy is pushed forward inferring, from scratch with a CNN, a confidence measure from the the raw left-right disparity map. Both approaches outperform Park [22]. Finally, in [21] is described a methodology aimed to infer training data from stereo sequences by exploiting multiple viewpoints and contradictions in depth maps. Concerning stereo with CNNs, in [34] is proposed how to learn a general-purpose similarity function and in [35, 36] a patch-based matching cost. This latter strategy turned out to be very effective and, coupled with an adaptive cost aggregation strategy [37] and disparity refinement steps based on SGM [10], has excellent performance on KITTI 12 and 15 datasets. The architecture proposed in [36] is about 80 times faster than the accurate one with an increase in error rate smaller than 1% on both KITTI datasets. Other fast architectures for patch-based cost computation with CNNs are [1, 13] while Mayer et al. [16] proposed the first end-to-end architecture for stereo matching. The large amount of training samples required by this latter method is addressed deploying a large, yet realistic, synthetic dataset. Finally, in [24] a CNN was trained to combine the outcome of multiple stereo algorithms in order to obtain more accurate results.

Recently has been proved that the joint use of effective confidence measures and stereo enables to improve accuracy. In [31] the matching costs of points with the higher estimated reliability are modified in order to appear like an *ideal* cost curve and then the entire cost volume is refined by means of a MRF framework. In [22], the cost curve is modulated according to the estimated reliability of each point and, in [25], the estimated confidence along each SGM scanline is deployed to weight cost aggregation accordingly. Finally, in [30], the inferred confidence measure is *plugged* into SGM [10] to dynamically change parameters $P1$ and $P2$.

3. Proposed method

This work aims at improving the reliability of standalone confidence measures, learning from their local behavior effective informative patterns making the assumption that, as for most computer vision algorithms, locality matters. Considering that the reference image and the disparity map

are locally consistent, we expect a similar behavior for the confidence maps. Moreover, we expect different confidence measures to expose specific local patterns that can be identified with an *ad hoc* training. To this end we leverage on a deep network, appropriately trained on a dataset with ground-truth, aimed at learning and detecting effective informative patterns for each examined confidence measure. Exhaustive experimental results on challenging stereo pairs confirm that the proposed strategy enables to dramatically improve the effectiveness of state-of-the-art confidence measures.

3.1. Enforcing local consistency

A confidence measure k assigns a value to a pixel p of the disparity map computed with respect to the reference image according to C_k, a function taking as arguments one or more of the following cues: the matching cost curve c, reference left L and right image R of the stereo pair, the disparity maps D_L and D_R obtained, respectively, using as reference L and R.

$$C_k(p) = f(c(p), L, R, D_L, D_R) \tag{1}$$

Excluding more recent approaches based on machine-learning, a conventional confidence measure can be obtained [11] analyzing matching costs, local properties of the cost curve or of the entire curve, local minima, consistency between left and right disparity maps and distinctiveness among image pixels. Typically, a more complex analysis allows to achieve a more accurate correctness prediction. For example, the Matching Score Measure (MSM) [11], which is the simplest confidence measure, only relies on the minimum matching cost value. It has been adopted as baseline method, showing that most of the other confidence measures outperform it [11]. Another one based on very simple analysis is the Left-Right Consistency (LRC) [11], aimed at detecting inconsistent points between left and right disparity maps. This measure performs very well near depth discontinuities, and is mainly useful to detect occluded pixels. However, it is not very informative due to its discretized nature. Both measures typically fail in presence of some well-known issues of stereo matching, such as low textured areas or repetitive patterns, where multiple local minima concurring to the role of minimum would yield to high confidence according to MSM. Similarly, the absence of discontinuities might lead LRC, to label a pixel as confident even if it has wrong disparities on both maps.

In our proposal, in order to predict the correctness of a disparity assignment enforcing the locality constraint, it is useful to encode match reliability with a confidence map. That is, given a confidence measure k, for each pixel p belonging to the reference image L, the confidence map $M_k \in [0, 1]$ is obtained as follows:

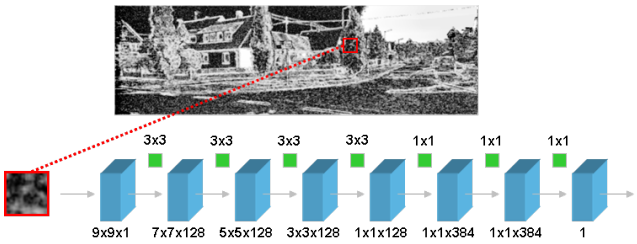


Figure 2. Proposed CNN architecture to prediction match reliability enforcing local consistency on the input confidence map.

$$M_k(p) = \frac{C_k(p) - \min_{p \in L} C_k(p)}{\max_{p \in L} C_k(p) - \min_{p \in L} C_k(p)} \tag{2}$$

Observing confidence maps we can notice that some measures apparently do not show distinctive patterns, looking like noisy images to human observers. Conversely, some others clearly present such distinctive patterns, related to particular features of the disparity map. Starting from these observations, we assume that local properties of confidence maps can be exploited to improve their reliability with respect to their original counterpart by learning specific image patterns of each measure. Such properties, within the neighborhood of a pixel p, are sought in the confidence map M_k analyzing a $N \times N$ patch centered on p with a CNN, trained to infer a new confidence estimation for the examined point.

3.2. Deep network architecture

To learn a locally consistent confidence prediction, we propose to train a custom CNN to assign the new value for the pixel under investigation, using image patches extracted from confidence maps. For this purpose we rely on a deep network architectures structured as in Figure 2.

In order to infer the final pixel-wise confidence score, in our experiments we evaluated different CNN architectures made of different convolutional layers, depending on the perceptive field of the network, and fully-connected layers. Convolutional layers extracts f feature maps by applying 3×3 kernels from the input feature maps fed by the previous layer, fully-connected containing n neurons. The single final neuron is in charge of the regression stage. Each layer is followed by activation operators, in particular we used Rectifier Linear Units (ReLU) and we applied a Sigmoid operator on the output of the last neuron. Following the successful deployment of CNNs for stereo [36] and confidence estimation [26], we chose convolutional kernels of fixed 3×3 size and we did not include any pooling operator. The remaining hyper-parameters of our architecture, such as the size of the perceptive field and the number of neurons, have been tuned during the experimental phase.

Given a patch of size $N \times N$, referred to as $P^{N\times N}_{M_k(p)}$, extracted from a confidence map M_k centered on pixel p, the value predicted by the network is:

$$M_{k^+}(p) = F(P^{N\times N}_{M_k}(p)) \in [0, 1] \quad (3)$$

where $F(P^{N\times N}_{M_k(p)})$ is the output of the network processing $P^{N\times N}_{M_k(p)}$. According to this terminology, we will refer, for example, to the learned version of the PKRN confidence measure as PKRN$^+$ (PKRN *plus*).

In testing, after the network has been trained, we replace the fully-connected layers with convolutional layers made of 1×1 kernels. This new model is functionally identical to the one used for training but, with the same network, it allows to process input of different size enabling a single forward pass of the full resolution confidence map M_k rather than forwarding all the single $P^{N\times N}_{M_k}$ patches. This strategy greatly reduces the time required to obtain the final confidence map M_{k^+}. The absence of pooling allows us to maintain full resolution output by applying zero-padding to the original M_k according to the size of the perceptive field.

4. Experimental results

In this section we describe in detail the methodology adopted for the training phase on a subset of the KITTI 12 [7] dataset. Then, we compare, on KITTI and Middlebury datasets, the learned confidence measures to their original counterparts[1]. In particular, we evaluate the performance in terms of correctness prediction by analyzing the Area Under Curve (AUC) [11] on the remaining images of the KITTI 12 [7] dataset as well as on the whole KITTI 15 [17] and Middlebury 14 [28] datasets without re-training the networks.

Since the ground-truth is required for training and for AUC evaluation, as common in this field [9, 22, 26, 25], for each considered dataset we rely on the evaluation training sets of KITTI 12 (194 images, 25 for training and 169 for testing), KITTI 15 (200 images) and Middlebury 14 (15 images). Moreover, we compute confidence measures according to the output of two algorithms: AD-CENSUS, aggregating matching costs (computed with the Hamming distance on 5×5 census transformed image patches) on a fixed support region of size 5×5, and MC-CNN algorithm [36].

4.1. Training phase

For each confidence measure we trained the CNN, on a subset of the KITTI 12 dataset, according to *stochastic gradient descend*, in order to minimize the *binary cross entropy*, with batch size set to 128 patches. Each network ran 15 training *epochs* with a *learning rate* equal to 0.003, reduced by a factor 10 after the 11^{th} epoch, a *momentum* of 0.9 and shuffled the training examples before the training phase. Network models and training phase have been implemented with the Torch 7 framework [2].

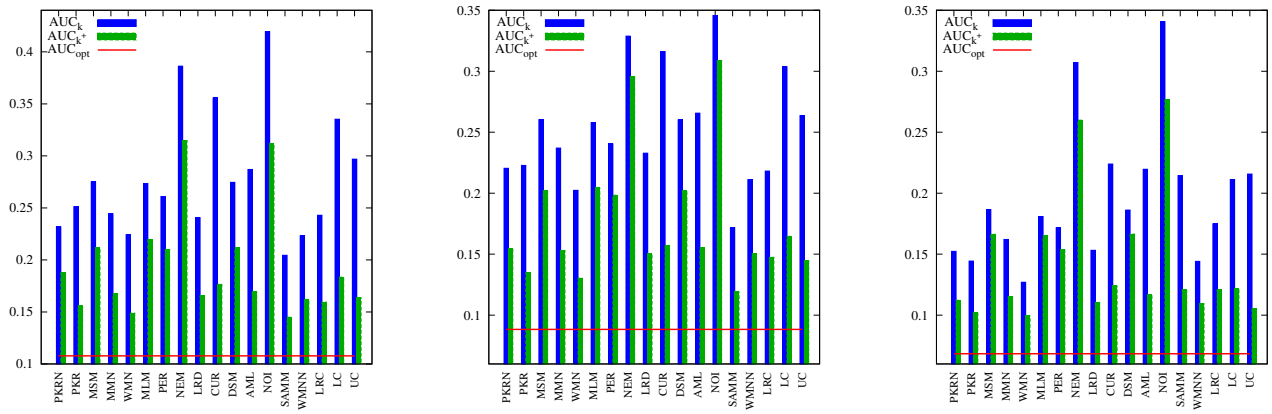
In our experiments we tested different amounts of training data to generate learned confidence maps and we achieved the best results considering 25 stereo images (i.e., from frame 000000 to 000024) of the KITTI 12 dataset [7]. Increasing the training set did not improve noticeably the quality of the learned confidence measures. From these 25 frames, we extracted patches centered on pixels with available ground-truth, obtaining approximatively 2.7 million samples for each confidence measure. Patches centered on points having a disparity error ≤ 3 (following the threshold suggested in [7, 17]) are labeled as $confident$ and encoded as ones, the remaining as zeros.

In our evaluation we considered 18 state-of-the-art standalone confidence measures and 5 approaches based on machine-learning. Regarding the first group, they are: Matching Score Measure (MSM), Peak Ratio (PKR) and Peak Ratio Naive (PKRN), Winner Margin (WMN) and Winner Margin Naive (WMNN), Negative Entropy Measure (NEM), Number Of Inflection points (NOI), Maximum Margin Naive (MMN), Maximum Likelihood Measure (MLM), Attainable Likelihood Measure (AML), Curvature (CUR), Local Curve (LC), Left Right Consistency (LRC), Left Right Difference (LRD), Distinctive Similarity Measure (DSM), Uniqueness Constraint (UC), Self-Aware Matching Measure (SAMM) and Perturbation (PER). Excluding PER [9], UC [3] and LC [33] the other confidence measures have been reviewed in [11]. Regarding the specific parameters setting, we set $\sigma_{MLM} = 0.3$ and $\sigma_{AML} = 0.1$ as suggested in [11]), $s_{PER} = 120$, $\gamma = 480$ for LC as suggested in [9]. SAMM has been computed in its symmetric version, within the range $[-\frac{d_{max}}{2}, \frac{d_{max}}{2}]$, as suggested by the authors.

Regarding confidence measures based on machine-learning we considered Ensemble [9] (in its more effective configuration with 23 features), GCP [31], Park [22] (in its more effective configuration with 22 features) and the two methods proposed in [25] and [26] referred, to as, respectively, *O1* and *CCNN*. We implemented these 5 approaches following exactly the guidelines reported in each paper and trained, as for our proposal, each one on the same 25 images of the KITTI 12 dataset. Before being fed to the deep network, each confidence map was normalized according to equation 2.

The AUC values reported in Section 4.2 and Section 4.3 for AD-CENSUS and in Section 4.4 for MC-CNN were obtained tuning the previously described hyper-parameters of our network as follows: 9×9 perceptive field, $f = 128$ kernels per convolutional layer, $n = 384$ neurons (*i.e.* 1×1 kernels at test time) per fully-connected layer. The 9×9 perceptive field enabled to achieve on average the best per-

[1]Source code and trained networks available on http://vision.disi.unibo.it/~mpoggi/code.html



(a) (b) (c)

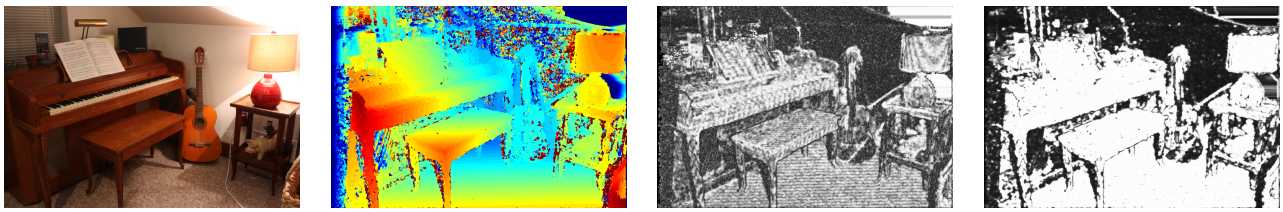
Figure 3. Average AUC for the 18 stand-alone confidence measures on the 3 considered datasets with AD-CENSUS. (a) Evaluation on KITTI 12 images excluded from training (169 frames, from 000025 to 000193), (b) evaluation on KITTI 15 dataset (200 frames), (c) evaluation on Middlebury 14 dataset (15 frames). In blue AUC related to the original confidence measure (*e.g.*, AUC_{PKRN}), in green the AUC related to its learned counterpart (*e.g.*, AUC_{PKRN+}). The red line shows the optimal AUC value (AUC_{opt}), computed according to 5.

Confidence	KITTI 12 (169/194)			KITTI 15 (200/200)			Middlebury 14 (15/15)		
measure	AUC_k	AUC_{k+}	Δ_k	AUC_k	AUC_{k+}	Δ_k	AUC_k	AUC_{k+}	Δ_k
PKRN	0.231682	**0.187407**	35.74%	0.220458	**0.154534**	49.90%	0.152359	**0.112248**	47.76%
PKR	0.251132	**0.155664**	66.61%	0.222827	**0.134693**	65.54%	0.144349	**0.101848**	55.94%
MSM	0.274919	**0.211803**	37.77%	0.260329	**0.202062**	33.88%	0.186604	**0.166312**	17.16%
MMN	0.244250	**0.167334**	56.37%	0.236990	**0.153026**	56.49%	0.162109	**0.115097**	50.15%
WMN	0.224146	**0.148876**	64.70%	0.202390	**0.130410**	63.12%	0.127015	**0.099424**	47.05%
MLM	0.273479	**0.219593**	32.52%	0.257940	**0.204421**	31.56%	0.180903	**0.164901**	14.22%
PER	0.260978	**0.210076**	33.23%	0.240324	**0.198303**	27.65%	0.171692	**0.153460**	17.65%
NEM	0.386211	**0.314742**	25.67%	0.328761	**0.295701**	13.75%	0.307148	**0.259922**	19.78%
LRD	0.240665	**0.165342**	56.69%	0.232831	**0.150244**	57.16%	0.153181	**0.110457**	50.38%
CUR	0.355582	**0.176552**	72.25%	0.316048	**0.157221**	69.76%	0.223898	**0.123904**	64.30%
DSM	0.274579	**0.211731**	37.68%	0.260062	**0.202075**	33.77%	0.186157	**0.166489**	16.70%
AML	0.287019	**0.169239**	65.72%	0.265626	**0.155299**	62.23%	0.219605	**0.116534**	68.16%
NOI	0.419441	**0.311631**	34.59%	0.345756	**0.308789**	14.36%	0.340609	**0.276457**	23.57%
SAMM	0.204491	**0.150287**	56.06%	0.171475	**0.12176**	59.81%	0.214449	**0.133298**	55.55%
WMNN	0.223139	**0.162058**	52.96%	0.211146	**0.150363**	49.50%	0.144132	**0.109271**	46.01%
LRC	0.242911	**0.159512**	61.73%	0.218156	**0.147458**	54.47%	0.174806	**0.120645**	50.89%
LC	0.335298	**0.183496**	66.73%	0.303691	**0.164670**	64.56%	0.211085	**0.121464**	62.80%
UC	0.296917	**0.165900**	69.28%	0.263651	**0.146081**	67.07%	0.215678	**0.104459**	75.50%
Optimal	0.107802			0.088357			0.068375		

Table 1. Average AUC for the 18 stand-alone confidence measures on the 3 considered datasets with AD-CENSUS. Last row reports the optimal AUC. The table is split into three blocks: left block reports evaluation on KITTI 12 images excluded from training (169 frames, from 000025 to 000193), middle block reports evaluation on KITTI 15 dataset (200 frames), right block reports evaluation on Middlebury 14 dataset (15 frames). Each block contains AUC for the original measure (AUC_k), its learned counterpart (AUC_{k+}) and the improvement (Δ_k) yielded by our proposal, with respect to AUC_{opt}, computed according to equation 5.

formance. The resulting CNN architecture has more than 600 thousand parameters and, with a full resolution confidence map of the KITTI dataset, it requires just 5 GB of memory and about 0.1 sec to infer a new confidence estimation with a Titan X GPU.

Finally, we stress the fact that in our experimental evaluation we performed a single training procedure on 25 images of the KITTI 12 dataset even when dealing with different datasets (i.e, KITTI 15 and Middlebury 14) and the remaining 169 images of KITTI 12.



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

Figure 4. Qualitative comparison of three stand-alone confidence measures and their learned counterparts. (a) Reference image, (b) Disparity map computed by AD-CENSUS, (c) PKR and (d) learned PKR$^+$. Higher confidence values are brighter. The disparity map is encoded with colormap jet.

4.2. Evaluation of stand-alone confidence measures

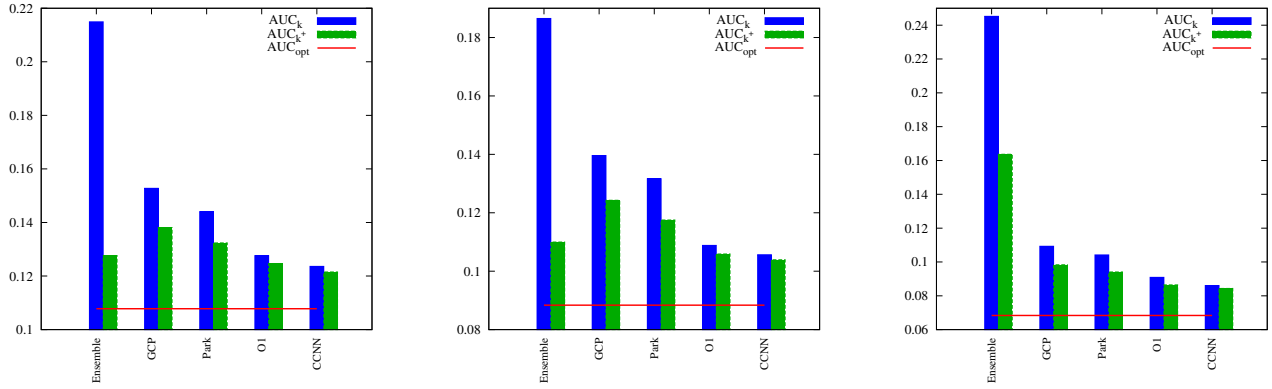
We assess the effectiveness of confidence measures performing ROC curves analysis, a commonly adopted evaluation protocol in this field [11, 9, 31, 22, 26, 25]. In particular, given a confidence map, the image points are sorted in descending order according to their confidence values. Then, top 5% are extracted and the error rate is computed as the ratio between the number of pixels with disparity errors larger than 3 (the standard threshold suggested for KITTI datasets [7, 17], maintained also on Middlebury 14 to be compliant with the training protocol) and the currently processed points, repeating this phase for the top 10%, 15% and so on. Ties are managed by including all pixels having the same confidence value (resulting in an horizontal curve). The AUC encodes the effectiveness of a confidence measure: the lower the AUC, the better is the estimation. Given the percentage ϵ of erroneous pixels in the disparity map, setting in our experiments threshold 3, the optimal AUC value can be obtained [11] as:

$$AUC_{opt} = \epsilon + (1 - \epsilon) \ln (1 - \epsilon) \tag{4}$$

Figure 3 summarizes the experimental results with AD-CENSUS on the 3 datasets involved in our evaluation. On the left we report results concerning the KITTI 12 dataset (the remaining 169 stereo pairs out of 194, being 25 used for training), in the middle concerning KITTI 15 dataset (200 stereo pairs, none involved in training), on the right concerning Middlebury 14 dataset (15 stereo pairs, none involved in training). Given a confidence measure k belonging to the pool of 18 stand-alone measures considered, two bars are depicted, related to the average AUC achieved by the original measure (AUC_k, in blue) and the one obtained after being processed by our framework (AUC_k^+, in green). The red line represents the optimal value (AUC_{opt}), computed according to equation 4. The closer the AUC is to AUC_{opt}, the more effective the confidence measure is. The charts in Figure 3 show that our method always improves the effectiveness of each confidence measure, achieving a lower AUC on all the datasets. To perceive more clearly the benefits yielded by our framework, we report in detail the AUCs in Table 1. Each row is related to a single stand-alone confidence measure, the final row contains AUC_{opt} values. The table is organized into three main blocks, each one related to one of the charts shown in Figure 3 (left: KITTI 12, middle: KITTI 15, right: Middlebury 14). For each dataset, each row reports the original confidence measure AUC_k, the learned counterpart AUC_{k^+} and the the improvement Δ_k, defined in 5, yielded by our frameworks with respect to the optimal AUC (*i.e.* AUC_{opt}, last row of the table).

$$\Delta_k = \frac{AUC_k - AUC_{k^+}}{AUC_k - AUC_{opt}} \tag{5}$$

According to 5, given a confidence measure, a Δ_k = 100% improvement would be achieved by our framework obtaining the optimal AUC_{opt}. Concerning the evaluation on KITTI 12 dataset, we can observe how Δ_k is always greater than 25%. In particular, the worst case is represented by NEM measure, being the AUC of NEM$^+$ 25.67% closer to AUC_{opt} with respect to the original version. For 6 measures (*i.e.*, PKRN, MSM, MLM, PER, DSM, and NOI) our framework yields an improvement between 30% and 50% and for the remaining 11 measures we report major improvements, up to 72.25% comparing CUR with CUR$^+$. Extending the analysis to the remaining datasets, the same behavior is confirmed for all the examined confidence measures. In particular, observing the results concerning KITTI 15 dataset, NEM and NOI yield the smaller improvements, respectively with a Δ_k of 13.75% and 14.36%, PER$^+$ achieves an improvement close to 30%, 5 measures (*i.e.*, PKRN, MSM, MLM, DSM and WMN) obtain a Δ_k between 30% and 50% and the remaining measures yield major gains, up to 69.76% deploying CUR$^+$. Finally, we report a further cross validation on Middlebury 14, the most challenging dataset being made of indoor scenes completely different from the 25 outdoor scenes of KITTI 12 *seen* during the training phase. In this case there are 6 measures (*i.e.*, MSM, MLM, PER, NEM, DSM and NOI) with a Δ_k between 14% and 30%, PKRN, WMN and WMNN between 30 and 50% and the remaining 9 measures showing major improvements, up to 74.91% achieved by UC$^+$.



(a) (b) (c)

Figure 5. Average AUC for the 5 confidence measures based on machine-learning on the 3 datasets with AD-CENSUS. (a) Evaluation on KITTI 12 images excluded from training (169 frames, from 000025 to 000193), (b) evaluation on KITTI 15 (200 frames), (c) evaluation on Middlebury 14 (15 frames). In blue the AUC for the original confidence measure (*e.g.*, AUC_{GCP} [31]), in green the AUC related to its learned counterpart (*e.g.*, AUC_{GCP+}). In red, optimal AUC values (AUC_{opt}) computed according to 4.

Confidence measure	KITTI 12 (169/194)			KITTI 15 (200/200)			Middlebury 14 (15/15)		
	AUC_k	AUC_{k+}	Δ_k	AUC_k	AUC_{k+}	Δ_k	AUC_k	AUC_{k+}	Δ_k
Ensemble [9]	0.214929	**0.127682**	81.44%	0.186504	**0.109991**	77.96%	0.245227	**0.163656**	46.12%
GCP [31]	0.152764	**0.138078**	32.66%	0.139611	**0.124286**	29.90%	0.109302	**0.098367**	26.71%
Park [22]	0.144077	**0.132393**	32.21%	0.131662	**0.117529**	32.64%	0.104146	**0.094084**	28.13%
O1 [25]	0.127645	**0.124695**	14.87%	0.108812	**0.105893**	14.27%	0.090908	**0.086444**	19.81%
CCNN [26]	0.123612	**0.121257**	14.90%	0.105645	**0.103645**	11.59%	0.086082	**0.084485**	9.01%
Optimal	0.107802			0.088357			0.068375		

Table 2. Average AUC for the considered 5 confidence measures based on machine-learning based on the 3 datasets with AD-CENSUS. The table is split into three blocks: left block reports evaluation on KITTI 12 images excluded from training (169 frames, from 000025 to 000193), middle block reports evaluation on KITTI 15 (200 frames), right block reports evaluation on Middlebury 14 (15 frames). Each block contains AUC for the original measure (AUC_k), the outcome of our framework (AUC_{k+}) and the improvement (Δ_k) yielded by our proposal, with respect to AUC_{opt}, computed according to equation 5.

Figure 4 provides a qualitative comparison between PKR confidence measure and its learned counterparts PKR$^+$ on the *Piano* stereo pair from Middlebury 14. Observing the figure we can clearly notice the improvements yielded by our framework exploiting local consistency. Confidence values are much more smooth and consistent (*e.g.*, the floor, the lampshade, the piano and its bench). Moreover, we can also notice how our framework can recover from gross failures of the original confidence measure (*e.g.*, the portion of the wall at the top-right corner of the image).

4.3. Evaluation of confidence measures based on machine-learning

Once assessed the effectiveness of our proposal on standalone measures, we extended our evaluation considering 5 state-of-the-art confidence measures based on machine-learning: Ensemble [9], GCP [31], Park [22], O1 [25] and CCNN [26]. As already pointed out, we adopt for this evaluation the same protocol for training and testing. In this case, we train the original 5 considered confidence measure on the same 25 images used to train our framework (frames from 000000 to 000024 of KITTI 12).

Figure 5 shows the results on the three datasets with AD-CENSUS, reported in detail in Table 2, according to the same methodology described in Section 4.2. Observing the figure we can clearly notice that our proposal always outperforms significantly the 5 original confidence measures on all the three datasets. The improvements are remarkable also for top-performing confidence measures O1 and CNN being Δ_k, respectively, greater than 14% and 9% in the worst case. For the other 3 confidence measures the improvement is, in the worst case, greater than 28% for Park, almost 27% for GCP and greater than 46% for Ensemble that, in the best case, improves by more than 81% with our framework. Interestingly, the learned Ensemble$^+$ confidence measure is able to outperform the original GCP and Park approaches on KITTI 12 and KITTI 15. This further evaluation confirms the effectiveness of our proposal even with the 5 confidence measures based on machine learning.

Moreover, comparing the results reported in Table 1 and

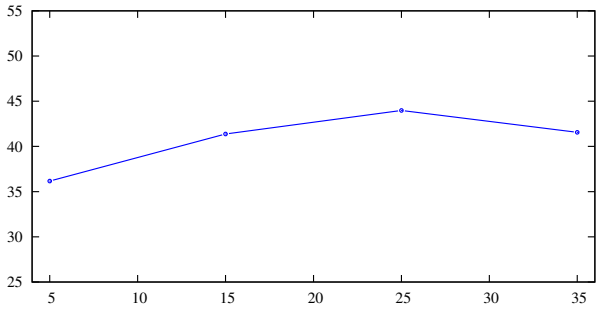


Figure 6. Average improvement Δ_k (%) on Middlebury 14 with different amount of training data (first 5, 15, 25 and 35 frames) from KITTI 12 with AD-CENSUS.

2, we can notice how with our proposal some stand-alone confidence measures are able to outperform approaches based on machine-learning. In particular, Ensemble is outperformed by all the learned confidence measures, except MLM$^+$, NEM$^+$ and NOI$^+$ on KITTI 12, MSM$^+$, MLM$^+$, PER$^+$, NEM$^+$, DSM$^+$ and NOI$^+$ on KITTI 15, NEM$^+$ and NOI$^+$ on Middlebury 14. GCP is outperformed by WMN$^+$ and SAMM$^+$ on KITTI 12, by PKR$^+$, WMN$^+$ and SAMM$^+$ on KITTI 15, by PKR$^+$, WMN$^+$, WMNN$^+$ and UC$^+$ on Middlebury 14. Park is outperformed by WMN$^+$ and SAMM$^+$ on KITTI 15, by PKR$^+$ and WMN$^+$ on Middlebury 2014. This means that the proposed framework is not only able to significantly improve the effectiveness of each considered confidence measure, but in many cases it enables to achieve even more accurate prediction by processing a single confidence measure rather than by combining multiple ones as done by the three machine-learning approaches Ensemble [9], GCP [31] and Park [22].

Finally, we report in Figure 6 the average improvement Δ_k achieved by our networks on Middlebury 14 as a function of the amount of training data. Observing the figure we can notice that we obtain the best performance with 25 frames and, more interestingly, our networks trained only on 5 frames achieve an average improvement greater than 35%.

4.4. Evaluation with MC-CNN

In Table 3 we provide additional experimental results concerned with state-of-the-art cost function MC-CNN [35, 36]. We trained our networks on the same amount of data (*i.e.*, 25 images of KITTI 12 dataset) and followed the same cross validation protocol adopted with the AD-CENSUS algorithm. Due to the lack of space, we report for MC-CNN only the average improvement Δ_k on the three datasets. The table confirms that, even with the more accurate MC-CNN algorithm, our proposal achieves notable improvements on each of the 23 examined confidence measures with Δ_k ranging from $\approx$10% (LC$^+$ in the worst case) to more than 77% (CUR$^+$ in the best case). Focusing on approaches based on machine-learning we can also notice that our proposal yields improvements from 10.6% (CCNN$^+$ in the worst case) to more than 74% (Ensemble$^+$ in the best case).

Measure	KITTI 12	KITTI 15	Middlebury 14
PKRN$^+$	66.5%	60.8%	29.1%
PKR$^+$	69.2%	54.7%	23.4%
MSM$^+$	34.4%	21.9%	23.4%
MMN$^+$	52.5%	41.4%	40.6%
WMN$^+$	73.1%	59.4%	23.7%
MLM$^+$	17.8%	13.5%	14.4%
PER$^+$	43.6%	33.9%	42.3%
NEM$^+$	46.6%	32.5%	34.3%
LRD$^+$	51.8%	41.1%	44.8%
CUR$^+$	11.4%	49.9%	77.1%
DSM$^+$	36.2%	23.6%	24.3%
AML$^+$	63.5%	53.4%	51.1%
NOI$^+$	46.1%	33.9%	28.9%
SAMM$^+$	70.9%	64.0%	61.4%
WMNN$^+$	57.2%	53.0%	23.1%
LRC$^+$	73.3%	63.7%	30.9%
LC$^+$	9.8%	25.8%	65.6%
UC$^+$	75.0%	71.0%	72.3%
Ensemble$^+$	74.3%	70.5%	38.5%
GCP$^+$	27.1%	18.5%	26.0%
Park$^+$	33.5%	28.5%	36.3%
O1$^+$	26.2%	22.0%	38.9%
CCNN$^+$	15.6%	10.6%	21.5%

Table 3. Average improvement Δ_k yielded by our proposal on the three datasets with MC-CNN [36].

5. Conclusions

In this paper we have proposed a methodology aimed at improving the effectiveness of confidence measures for stereo by exploiting local consistency. Our framework, leveraging on a deep network, is able to learn and improve the local behavior of confidence measures and, to the best of our knowledge, it is the first method to move beyond single pixel-wise confidence estimation performed by other approaches. The exhaustive experimental evaluation with two stereo algorithms, including a cross-validation on two additional datasets, shows that our method enables remarkable improvements on each of the 23 state-of-the-art confidence measures and on each dataset. This confirms the assumption made in this paper: confidence maps are locally consistent and a deep network can learn how to exploit this fact. In particular, results reported with state-of-the-art confidence measures based on machine-learning set the bar a further step closer to optimality paving the way to further improvements in this field.

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] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 4

[3] L. Di Stefano, M. Marchionni, and S. Mattoccia. A fast area-based stereo matching algorithm. *Image and vision computing*, 22(12):983–1005, 2004. 4

[4] G. Egnal, M. Mintz, and R. P. Wildes. A stereo confidence metric using single view imagery. In *PROC. VISION INTERFACE*, pages 162–170, 2002. 2

[5] G. Egnal and R. 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. 1, 2

[6] F. Garcia, B. Mirbach, B. E. Ottersten, F. Grandidier, and A. Cuesta. Pixel weighted average strategy for depth sensor data fusion. In *ICIP*, pages 2805–2808. IEEE, 2010. 1, 2

[7] 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, 4, 6

[8] 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

[9] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *CVPR. Proceedings*, pages 305–312, 2013. 1. 1, 2, 4, 6, 7, 8

[10] 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

[11] 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, 3, 4, 6

[12] D. Kong and H. Tao. A method for learning matching errors in stereo computation. In *British Machine Vision Conference (BMVC)*, 2004 2004. 2

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

[14] 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. 1

[15] 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. 1, 2

[16] 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

[17] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 4, 6

[18] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. Frahm, R. Yang, D. Nistér, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8, 2007. 1, 2

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

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

[21] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof. Using self-contradiction to learn confidence measures in stereo vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[22] 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, 4, 6, 7, 8

[23] 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. 1, 2

[24] M. Poggi and S. Mattoccia. Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In *Proceedings of the 4th International Conference on 3D Vision, 3DV*, 2016. 2

[25] 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. 1, 2, 4, 6, 7

[26] M. Poggi and S. Mattoccia. Learning from scratch a confidence measure. In *Proceedings of the 27th British Conference on Machine Vision, BMVC*, 2016. 1, 2, 3, 4, 6, 7

[27] 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

[28] 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, 4

[29] 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

[30] A. Seki and M. Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016. 1, 2

[31] 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. 1, 2, 4, 6, 7, 8

[32] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. 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. 2

[33] A. Wedel, A. Meiner, C. Rabe, U. Franke, and D. Cremers. Detection and Segmentation of Independently Moving Objects from Dense Scene Flow. In *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 14–27, Bonn, Germany, August 2009. Springer. 4

[34] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[35] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 8

[36] 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, 4, 8

[37] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19(7):1073–1079, jul 2009. 2