Efficient confidence measures for embedded stereo

Matteo Poggi, Fabio Tosi, and Stefano Mattoccia

University of Bologna
Department of Computer Science and Engineering (DISI)
Viale del Risorgimento 2, Bologna, Italy
{matteo.poggi8,fabio.tosi5,stefano.mattoccia}@unibo.it

Abstract. The advent of embedded stereo cameras based on low-power and compact devices such as FPGAs (Field Programmable Gate Arrays) has enabled to effectively address several computer vision problems. However, being the depth data generated by stereo algorithms affected by errors, reliable strategies to detect wrong disparity assignments by means of confidence measures are desirable. Recent works proved that confidence measures are also a powerful cue to improve the overall accuracy of stereo. Most approaches aimed at predicting match reliability rely on cost volume analysis, an information seldom available as output of most embedded depth sensors. Therefore, in this paper we analyze and evaluate strategies compatible with the constraints of embedded stereo cameras. In particular, we focus our attention on methods to infer match reliability inside depth sensors based on highly constrained computing architectures such as FPGAs. We quantitatively assess, on Middlebury 2014 and KITTI 2015 datasets, the impact of different design strategies for 16 confidence measures from the literature, suited for implementation on such embedded systems. Our evaluation shows that, compared to the confidence measures typically deployed in this context and based on storing intermediate results, other approaches yield much more accurate predictions with negligible computing requirements and memory footprint. This enables for their implementation even on highly constrained architectures.

1 Introduction

The recent availability of embedded depth sensors paved the way to a variety of computer vision applications for autonomous driving, robotics, 3D reconstruction and so on. In these application depth is crucial and several approaches have been proposed to tackle this problem following two main strategies. On one hand *Active* sensors infer depth by perturbing the sensed scene by means of structured light, laser projection and so on. On the other hand, *passive* depth sensors infer depth not altering at all the sensed environment. Although sensors based on active technologies are quite effective they have some limitations. In particular, some of them (e.g., Kinect) are not suited for outdoor environments during daytime while others (e.g., LIDAR) provide only sparse depth maps and are quite expensive, cumbersome and containing moving mechanical parts.

Stereo vision is the most popular passive technique to infer dense depth data from two or more images. Many algorithms have been proposed to solve the stereo correspondence problem, some of them particularly suited for hardware implementation, thus enabling the design of compact, low-powered and real-time depth sensors [7], [10], [2], [4], [24], [22], [26]. Despite the vast literature in this field, challenging conditions found in most practical applications represent a major challenge for stereo algorithms. Popular benchmarks Middlebury 2014 [21] and KITTI 2015 [11] clearly highlighted this fact. Therefore, regardless of the stereo algorithm deployed, it is essential to detect its failures to filter-out wrong unreliable points that might lead to a wrong interpretation of the sensed scene. To this aim, confidence measures have become a popular topic on recent works concerning stereo. Some recent confidence measures combine multiple features within random forest frameworks to obtain more reliable confidence scores while an even more recent trend aims to infer confidence prediction leveraging on Convolutional Neural Networks (CNN) [19], [23]. Despite their effectiveness, the latter strategies are often not compatible with the computing resources available inside the depth sensor, typically a low cost FPGA or a System-On-Chip (SoC) based on ARM CPU cores and an FPGA (e.g., Xilinx Zynq). Moreover, the features required by most of these machine-learning frameworks are not available as output of the embedded stereo cameras being in most cases computed from the cost volume (often referred to as disparity space image (DSI) [20]).

Therefore, in this paper we consider a subset of confidence measures compatible with embedded devices evaluating their effectiveness, on two popular challenging datasets and two algorithms typically deployed for real-time stereo for embedded systems, focusing our attention on issues related to their FPGA implementation. Our study highlights that some of the considered confidence measures, appropriately modified to fit with typical hardware constraints found in the target architectures, clearly outperform those currently deployed in most embedded stereo cameras.

2 Related Work

Stereo represents a popular and effective solution for depth estimation. It exploits epipolar geometry to find corresponding pixels on two or multiple synchronized frames, thus enabling to infer distance of the observed points by means of triangulation. According to the taxonomy by Scharstein and Szeliski [20], algorithms can be grouped into local and global methods. Algorithms belonging to the former group are usually very fast algorithms but typically less accurate than global ones. The Semi-Global Matching (SGM) [6] algorithm represents a very good trade-off between speed and accuracy and for this reason one of the most popular approach to infer depth even with embedded devices. The core of SGM algorithms consists of multiple and independent *scanline optimization* (SO) [20] along different directions. Each SO is fast, but affected by *streaking* artifacts near discontinuities. However, by combining multiple SOs as done by SGM significantly softens this issue. Moreover its computational structure allows for different optimization strategies and simplifications that enabled to implement it on almost any computing architecture (e.g., CPUs, GPUs, SoC, FPGAs). In particular, low power and massively parallel devices such as FPGAs represents a very good design choice for depth sensors with optimal performance/Watt. Examples of stereo pipeline based on SGM mapped on FPGAs are [2], [4], [24], [22], [7], [26], [10]. Some of them deploy hardware-friendly implementations, based on census transform [28] and 4 or 5 scanlines computed in a single image scan from top-left to bottom-right. On FPGAs a smart design is crucial in order to achieve accurate real-time results without violating the limited logic resource available.

Despite the good accuracy of SGM and state-of-the-art algorithms [29], stereo is still an open problem, as witnessed by recent, challenging datasets [21], [11]. Thus, detecting failures of the stereo algorithm is a desirable property to achieve a more meaningful understanding of the sensed environments.

Several confidence measures have been proposed to tackle match reliability. In [8] the authors highlighted how different cues available inside the pipeline of general-purpose stereo algorithms implemented in software lead to different degrees of effectiveness on well-known ill-conditions of stereo such as occlusions, lack of texture and so on. Most recent proposals in this field proved that machine-learning can be effectively deployed to infer more accurate confidence measures, capable to better detect disparity errors. The very first work [5] trained a random forest classifier on multiple measures or features extracted from the DSI. More recent and effective proposals based on this strategy were proposed in [25] and [15], while in [18] was shown that a confidence measure could be effectively inferred by processing cues computed only from the disparity map. In [14] was proposed a data generation process based on multiple view points and contradictions, to select reliable labels to train confidence measures based on random forests. Latest works on confidence measures rely on deep-networks: [19] and [23] address confidence estimation by means of a CNN processing patches, respectively, from the left disparity map and from both left and right disparity maps.

Finally, we conclude this section observing that confidence measures have been deployed to detect occlusions [6], [13] and sensors fusion [9], [12]. Moreover, they were also plugged inside stereo pipeline to improve the overall accuracy by acting on the initial DSI [16], [25], [15], [18].

3 Hardware strategies for confidence implementation

When dealing with conventional CPU based systems confidence measures are generally implemented in C, C++ and to maintain the whole dynamic range single or double floating point data types are deployed. However, floating point arithmetic is sometimes not available in embedded CPU and generally unsuited to FPGAs. In particular, transcendental functions and divisions represent major issues when dealing with such devices. To overcome these limitations, fixed point arithmetic is usually deployed [1]. Fixed point represents an efficient and hardware-friendly way to express and manipulate fractional numbers with a fixed number of bits [1]. Indeed, fixed-point math can be represented with an integer number split into two distinct parts: the integer content (I), and the fractional content (F). Through the simple use of integer operations, the math can be efficiently performed with little loss of accuracy taking care to use a sufficient number of bits. The steps required to convert a floating point value to the corresponding fixed representation with F bits - the higher, the better in terms of accuracy - are the following:

1. Multiply the original value by 2^F

2. Round the result to the closest integer value
3. Assign this value into the fixed-point representation

Fixed point encoding greatly simplifies arithmetic operations with non-integer values, but integer divisions can be demanding - in particular on FPGAs - except when dealing with divisors which are powers of 2. In fact, in this case division requires almost negligibly hardware resources being carried out by means of a simple right shift. Thus, a simplified method to avoid integer divisions consists in rounding the dividing value to the closest power of 2, then shifting right according to its $\log_2$. This strategy will be referred to as *pow*.

Although fixed point increases the overall efficiency, some confidence measures rely on transcendental functions (in particular, exponentials and logarithms) which represent an a further major issue even when dealing with CPU based systems. An effective strategy to deal with such functions consists in deploying Look-Up Tables (LUTs) to store pre-computed results encoded with fixed point arithmetic. That is, given a function $\mathcal{F}(x)$, with x assuming n possible values, a LUT of size n can store all the possible outcome of such function. Of course, this approach is feasible only when the size of the LUT (proportional to n) is compatible with the memory available in the device.

4 Confidence measures suited for hardware implementation

In this section we describe the pool of confidence measures from the literature suited for implementation on target embedded devices. Figure 1 shows the matching cost curve for a pixel of the reference image. Given a pixel $\mathbf{p}(x, y)$, we will refer to its minimum cost as c_1, the second minimum as c_2 and the second local minimum as c_{2m}. The matching cost for any disparity hypothesis d will be referred to as c_d while the disparity corresponding to c_1 as d_1, the one corresponding to c_2 as d_2 and so on. If not specified otherwise, costs and disparities are referred to the reference left image (L) of the stereo pair. When dealing with right image (R), we introduce the ^{R} symbol on costs (e.g., c_1^R) and disparities. We denote as $\mathbf{p}'(x', y')$ the homologous point of $\mathbf{p}$ according to d_1 (i.e., $x' = x - d_1$, $y' = y$). It is worth to note that, assuming the right image as reference, the matching costs can be easily obtained by scanning in diagonal the cost volume computed with reference the left image without any further new computation. Nevertheless, adopting this strategy would require an additional buffering of $\frac{d_{max} \cdot (d_{max}+1)}{2}$ matching costs with d_{max} the disparity range deployed by the stereo algorithm.

We distinguish the considered pool of confidence measures in two, mutually exclusive, categories:

- *Hardware friendly*: confidence measures whose standard implementation is fully compliant with embedded systems.
- *Hardware challenging*: confidence measures involving transcendental functions and/or floating point divisions not well suited for embedded systems in their conventional formulation.

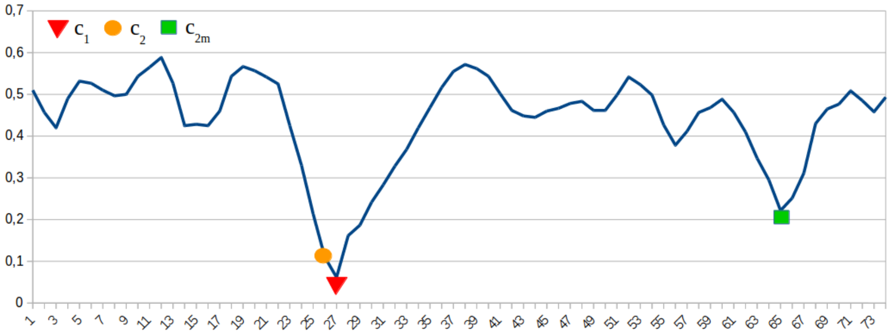


Fig. 1. Example of cost curve, showing the matching cost c_1, the second minimum c_2 and the second local minimum c_{2m}. On x axis the disparity range, on y magnitude of the costs.

4.1 Hardware friendly

This category groups confidence measures involving simple math operations that do not represent issues when dealing with implementation on embedded systems. The *matching score measure* (MSM) [8] negates the minimum cost c_1 assuming it related to the reliability of a disparity assignment. *Maximum margin* (MM) estimates match uncertainty by computing the difference between c_{2m} and c_1 while its variant *maximum margin naive* (MMN) [8] replaces c_{2m} with c_2. Given two disparity maps computed by a stereo algorithm assuming as reference L and R, the *left-right consistency* (LRC) [8] sets as confidence the negation of the absolute difference between the disparity of a point in L and its homologous point in R. This method represents one of the most widely adopted strategy by most algorithms even for those implemented on embedded devices. Another popular and more efficient strategy based on a single matching phase is the *uniqueness constraint* (UC) [3]: it assumes as poorly confident those pixels colliding on the same point of the target image (R) with the exception of the one having the lowest c_1. *Curvature* (CUR) [8] and *local curve* (LC) [27] analyze the behavior of the matching costs in proximity of the minimum c_1 and its two neighbors at (d_1-1) and (d_1+1) according to two similar strategies. Finally, *number of inflections* (NOI) [8] simply counts the number of local minima in the cost curve assuming that the lower, the more confident is the disparity assignment.

4.2 Hardware challenging

Confidence measures belonging to this category can not be directly implemented in embedded systems following their original formulation. We consider *peak ratio* (PKR) [8] which computes the ratio between c_{2m} and c_1 and its variant *peak ratio naive* (PKRN) [8] which replaces c_{2m} with the second minimum c_2. According to the literature, these measures are quite effective but seldom deployed in embedded stereo cameras. Another popular measure is *winner margin measure* (WMN) [8] which normalizes the difference between c_{2m} and c_1 by the sum of all costs. Its variant *winner margin measure naive*

(WMNN) [8] follows the same strategy replacing c_{2m} with c_2. The *left-right difference* measure (LRD) [8] computes the difference between c_2 and c_1 divided by the absolute difference between c_1 and the minimum cost of the homologous point in R (c_1^R). For these confidence measures the major implementation issue on embedded systems is represented by the division. For the remaining confidence measures the main problem is represented by transcendental functions: exponentials and logarithms. *Maximum likelihood measure* (MLM) [8] and *attainable maximum likelihood* (AML) [8] infer from the cost curve a *probability density function* (pdf) related to an ideal c_1, respectively, equal to zero for MLM and to c_1 for AML. A more recent and less computational demanding approach *perturbation* (PER) [5], encodes the deviation of the cost curve from a Gaussian function ant its implementation requires a division by a constant value suited for a LUT-based strategy. Finally, we also mention two very effective confidence measures based on distinctiveness, namely *distinctive similarity measure* (DSM) and *self-aware matching measure* (SAMM) and one *negative entropy measure* (NEM) [8] that infers the degree of uncertainty of each disparity assignment from the negative entropy of c_1. However, they require additional cues (e.g., self-matching costs on both reference and target images for SAMM) not well suited to embedded systems and thus not included in our evaluation.

5 Experimental results

In this section we evaluate the 16 confidence measures previously reviewed and implemented following the design strategies outlined so far. We test their effectiveness with the output of two popular stereo algorithms well-suited for implementation on embedded systems:

- AD-CENSUS: aggregates matching costs according to the Hamming distance computed on 5×5 patches with census transform [28]. A further aggregation step is performed by a 5×5 box-filter. To reduce the amount of bits required by the single matching cost, we normalized aggregated costs by the dimension of the box-filter (to be more hardware-friendly, by 16), with negligible reduction of accuracy according to [17].
- SGM [6]: four scanline implementation using as data term the same AD-CENSUS aggregated costs and for parameters P1 and P2, respectively, 11 and 110. The four directions are those processed by scanning the image from top-left to bottom-right as suggested in [2, 10, 17].

We encode matching costs with, respectively, 6 and 8 bit integer values, being this amount enough to encode the entire ranges. Regarding parameters of the confidence measures: for LC, we set the normalization factor γ to 1 to avoid division, while for PER, MLM and AML we set s_{PER} to 1.2 and σ_{aml}, σ_{mlm} to 2 before initializing the LUTs. The other 12 confidence measures do not have parameters.

For CUR, LRC, LC, MM, MMN, MSM, NOI and UC we provide experimental results with the conventional implementation since their mapping on embedded devices is totally equivalent. Moreover, regarding PER, we do not report results concerned with

measure	standard
Opt.	0.08891
CUR	0.24377 (14)
LRC	0.19933 (7)
LC	0.24377 (15)
MM	0.17765 (6)
MMN	0.19933 (8)
MSM	0.23182 (13)
NOI	0.39053 (16)
UC	0.20974 (10)

measure	standard
Opt.	0.08891
AML	0.21173 (11)
LRD	0.17004 (3)
MLM	0.22413 (12)
PER	0.20687 (9)
PKR	0.16250 (1)
PKRN	0.17185 (5)
WMN	0.16503 (2)
WMNN	0.17169 (4)

(a)

measure	standard
Opt.	0.04367
CUR	0.11602 (11)
LRC	0.16853 (15)
LC	0.11602 (12)
MM	0.09371 (5)
MMN	0.12920 (14)
MSM	0.10181 (7)
NOI	0.32028 (16)
UC	0.10347 (9)

measure	standard
Opt.	0.04367
AML	0.08843 (3)
LRD	0.11725 (13)
MLM	0.09567 (6)
PER	0.08766 (1)
PKR	0.08813 (2)
PKRN	0.10527 (10)
WMN	0.08898 (4)
WMNN	0.10232 (8)

(b)

Table 1. Experimental results, in terms of AUC, on Middlebury 2014 dataset with AD-CENSUS (a) and SGM (b) algorithms for the 16 confidence measures using a conventional software implementation. In red, top-performing measure. We also report the absolute ranking.

division by the closest power of two being the divisor a constant value and thus such operation can be addressed with a LUT. Finally, it is worth observing that most embedded stereo vision systems rely on LRC [7, 2] and UC [10, 2] for confidence estimation.

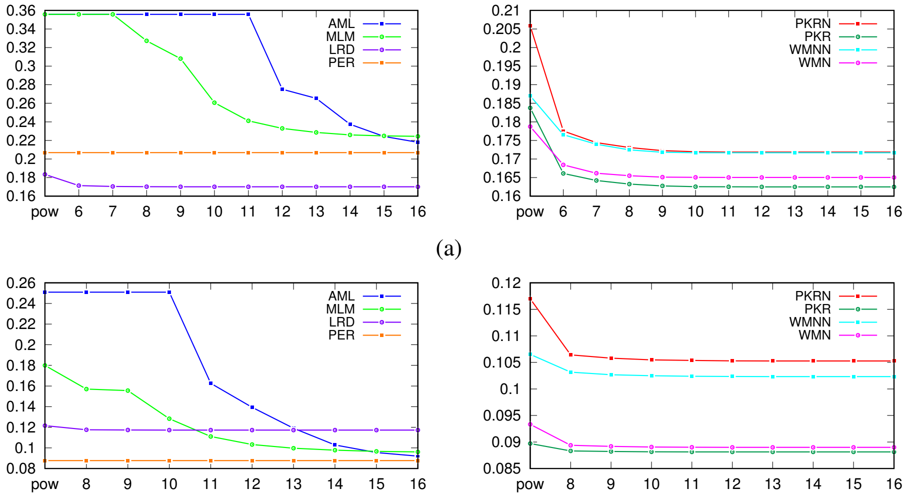
In section 5.1 we describe the evaluation protocol and in section 5.2 we report experimental results on Middlebury 2014 (at quarter resolution) and KITTI 2015 datasets for AD-CENSUS and SGM algorithms.

5.1 Evaluation protocol

The standard procedure to evaluate the effectiveness of a confidence measure is the ROC curve analysis, proposed by Hu and Mordohai [8] and adopted by all recent works [5], [25], [15], [19], [18], [23] in this field. By extracting subsets of pixels from the disparity map, according to descending order of confidence, a ROC curve is depicted by computing the error rate, starting from a small subset of points (i.e., 5% most confident) and then increasing the pool of pixels iteratively, up to include all pixels. This leads to a non-monotonic ROC curve, whose area (AUC) is an indicator of the effectiveness of the confidence measure. Given a disparity map with $\varepsilon\%$ of pixels being erroneous, an optimal confidence measure should draw a curve which is zero until $\varepsilon\%$ pixels have been sampled. The area of this curve represents the optimal AUC achievable by a confidence measure and can be obtained, according to [8], as:

$$AUC_{opt} = \int_{1-\varepsilon}^{\varepsilon} \frac{p-(1-\varepsilon)}{p} dp = \varepsilon + (1-\varepsilon)\ln{(1-\varepsilon)} \tag{1}$$

As reported on Middlebury 2014 and KITTI 2015 benchmarks, ε is obtained by fixing a threshold value on disparity error of, respectively 1 and 3 for the two datasets following the guidelines. Confidence measures achieving lower AUC values (closer to optimal) better identify wrong disparity assignments.



(a)

(b)

Fig. 2. Average AUC values on the Middlebury 2014 dataset for hardware challenging measures, varying the implementation settings (i.e., *pow* and number of bits of fixed-point arithmetic). (a) AD-CENSUS, (b) SGM algorithm.

5.2 Experimental evaluation on Middlebury 2014 and KITTI 2015

In this section we report results on Middlebury 2014 and KITTI 2015 datasets in terms of average AUC values achieved by confidence measures implemented in software. For hardware challenging measures of section 4.2 we also report multiple AUC obtained with increasing number of bits dedicated to fixed point operations (i.e., from 6 to 16 for AD-CENSUS and from 8 to 16 for SGM, so as to handle the whole cost range). Moreover, for such measures, we also report the results obtained by rounding to the closest power of 2 and, then, shifting right (referred to as *pow* in the charts).

Table 1 shows for Middlebury 2014 that LRC and UC, confidence measures typically deployed in embedded stereo cameras, are less effective than MM, LRD, PKR, PKRN, WMN, WMNN with AD-CENSUS and MM, MSM, AML, MLM, PER, PKR, WMN, WMN with SGM. We can notice that LRC provides poor confidence estimation with SGM but achieves better results with AD-CENSUS while UC has average performance with both algorithms. Considering the more effective confidence measures in the table, we can notice that PKR and WMN, as well as their naive formulations, performs pretty well with both algorithms clearly providing much more accurate confidence estimation compared to LRC and UC. Moreover, we can notice that PER achieves the best performance with SGM but it does not perform as well with AD-CENSUS, yielding slightly better confidence predictions with respect to UC. Specularly, LRD provides very reliable predictions with AD-CENSUS but poor results with SGM. Finally, we point our that top-performing confidence measures always belong to the hardware challenging category.

measure	standard
Opt.	0.08055
CUR	0.30692 (14)
LRC	0.20018 (2)
LC	0.30692 (15)
MM	0.20601 (4)
MMN	0.24588 (11)
MSM	0.25571 (13)
NOI	0.31160 (16)
UC	0.22324 (8)

measure	standard
Opt.	0.04367
AML	0.23053 (10)
LRD	0.20706 (5)
MLM	0.25180 (12)
PER	0.22575 (9)
PKR	0.19821 (1)
PKRN	0.20931 (7)
WMN	0.20221 (3)
WMNN	0.20795 (6)

(a)

measure	standard
Opt.	0.01618
CUR	0.08585 (11)
LRC	0.10377 (15)
LC	0.08585 (12)
MM	0.06374 (8)
MMN	0.09549 (14)
MSM	0.05999 (5)
NOI	0.16308 (16)
UC	0.06310 (7)

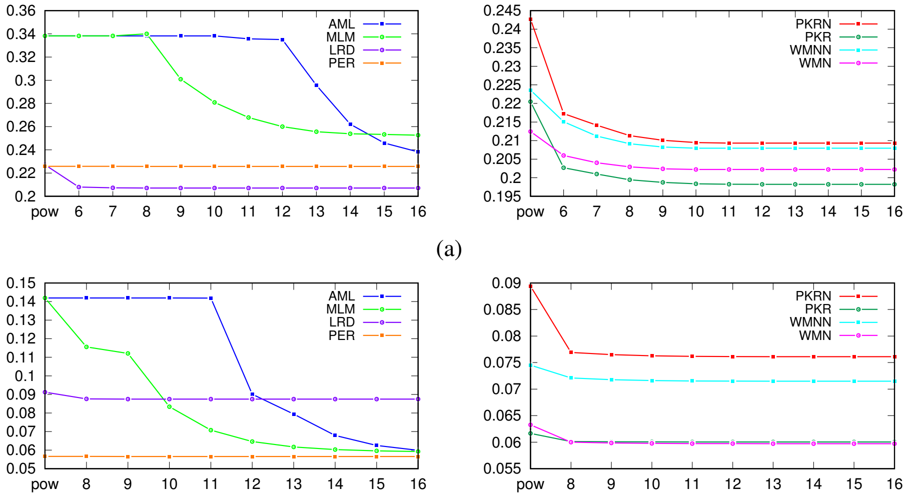
measure	standard
Opt.	0.01618
AML	0.05738 (2)
LRD	0.08744 (13)
MLM	0.05889 (3)
PER	0.05657 (1)
PKR	0.06003 (6)
PKRN	0.07611 (10)
WMN	0.05970 (4)
WMNN	0.07149 (9)

(b)

Table 2. Experimental results, in terms of AUC, on KITTI 2015 dataset with AD-CENSUS (a) and SGM (b) algorithms for the 16 confidence measures using a conventional software implementation. In red, top-performing measure. We also report the absolute ranking.

Therefore, in Figure 2 we report the performance of hardware challenging confidence measures, on Middlebury 2014 with AD-CENSUS and SGM, with multiple simplification settings. Observing the charts, PER is independent of the adopted strategy, being based on a LUT. Moreover, excluding PER, we can notice that the best performing ones (PKR, PKRN, WMN and WMNN at the right side of the figure) are those less affected by the number of bits deployed for fixed-point computations, thus resulting in reduced computational resources. In particular, we can observe that with only 8 bits, PKR and WMN achieve with both algorithms results almost comparable to their conventional software implementation. A similar behavior can be observed, with slightly worse performance, for their naive formulation PKRN and WMNN and for LRD that, excluding PER, is the approach less dependent of the number of bits. On the other hand, AML e MLM with both algorithms are significantly affected by the number of bit deployed for their implementation achieving results comparable to their traditional software formulation, respectively, only with 13 and 16 bits. Finally, excluding PER, we can observe that dividing by a power of 2 always provides poor results with respect to other simplifications. However, we highlight that even with this very efficient implementation strategy, PKR, WMN outperform LRC and UC with both stereo algorithms. Thus, trading simplified computations with memory footprint leads to design better alternatives to standard confidence measures for embedded systems.

Table 2 reports the average AUCs for the two considered stereo algorithms on KITTI 2015 for software implementation of the 16 confidence measures. Compared to Table 1 we can notice a similar behavior with a notable difference. In fact, observing Table 2 we highlight that LRC achieves almost optimal results on AD-CENSUS but yields very poor performance with SGM. Looking at the behavior of the hardware challenging measures, reported in Figure 3, we observe on KITTI 2015 a substantially similar behavior with respect to Figure 2 concerned with Middlebury 2014.



(a)

(b)

Fig. 3. Average AUC values on the KITTI 2015 dataset for hardware challenging measures, varying the implementation settings (i.e., *pow* and number of bits of fixed-point arithmetic). (a) AD-CENSUS, (b) SGM algorithm.

6 Conclusions

In this paper we have evaluated confidence measures suited for embedded stereo cameras. Our analysis shows that conventional approaches, LRC and UC, are outperformed by other considered solutions, whose implementation on embedded devices enables to achieve more accurate confidence predictions with a negligible amount of hardware resources and/or computations. In particular, according to our evaluation on Middlebury 2014 and KITTI 2015, PKR and WMN represent the overall best choice when dealing with two popular algorithms, AD-CENSUS and SGM, frequently deployed for embedded stereo systems.

References

1. Bailey, D.: Space efficient division on fpgas. In: Electronics New Zealand Conference (ENZCon06). pp. 206–211 (2012)
2. Banz, C., Hesselbarth, S., Flatt, H., Blume, H., Pirsch, P.: Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In: ICSAMOS. pp. 93–101 (2010)
3. Di Stefano, L., Marchionni, M., Mattoccia, S.: A fast area-based stereo matching algorithm. Image and vision computing 22(12), 983–1005 (2004)
4. Gehrig, S.K., Eberli, F., Meyer, T.: A real-time low-power stereo vision engine using semi-global matching. In: ICVS. pp. 134–143 (2009)
5. Haeusler, R., Nair, R., Kondermann, D.: Ensemble learning for confidence measures in stereo vision. In: CVPR (2013)

6. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. PAMI 30(2), 328–341 (feb 2008)
7. Honegger, D., Oleynikova, H., Pollefeys, M.: Real-time and low latency embedded computer vision hardware based on a combination of fpga and mobile cpu. In: IROS (2014)
8. Hu, X., Mordohai, P.: A quantitative evaluation of confidence measures for stereo vision. PAMI pp. 2121–2133 (2012)
9. Marin, G., Zanuttigh, P., Mattoccia, S.: Reliable fusion of tof and stereo depth driven by confidence measures. In: ECCV 2016. pp. 386–401 (2016)
10. Mattoccia, S., Poggi, M.: A passive rgbd sensor for accurate and real-time depth sensing self-contained into an fpga. In: 9th ICDSC (2015)
11. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR (2015)
12. Merrell, P., Akbarzadeh, A., Wang, L., Frahm, J., Nistr, R.Y.D.: Real-time visibility-based fusion of depth maps. In: CVPR (2007)
13. Min, D.B., Sohn, K.: An asymmetric post-processing for correspondence problem. Sig. Proc.: Image Comm. 25(2), 130–142 (2010)
14. Mostegel, C., Rumpler, M., Fraundorfer, F., Bischof, H.: Using self-contradiction to learn confidence measures in stereo vision. In: CVPR (2016)
15. Park, M.G., Yoon, K.J.: Leveraging stereo matching with learning-based confidence measures. In: CVPR (2015)
16. Pfeiffer, D., Gehrig, S., Schneider, N.: Exploiting the power of stereo confidences. In: CVPR (2013)
17. Poggi, M., Mattoccia, S.: Evaluation of variants of the sgm algorithm aimed at implementation on embedded or reconfigurable devices. In: IC3D (2016)
18. Poggi, M., Mattoccia, S.: Learning a general-purpose confidence measure based on o(1) features and asmarter aggregation strategy for semi global matching. In: 3DV (2016)
19. Poggi, M., Mattoccia, S.: Learning from scratch a confidence measure. In: BMVC (2016)
20. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision 47(1-3), 7–42 (apr 2002)
21. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: CVPR (2003)
22. Schmid, K., Hirschmuller, H.: Stereo vision and imu based real-time ego-motion and depth image computation on a handheld device. In: ICRA (2013)
23. Seki, A., Pollefeys, M.: Patch based confidence prediction for dense disparity map. In: BMVC (2016)
24. Shan, Y., Hao, Y., Wang, W., Wang, Y., Chen, X., Yang, H., Luk, W.: Hardware acceleration for an accurate stereo vision system using mini-census adaptive support region. ACM Trans. Embed. Comput. Syst. 13(4s), 132:1–132:24 (apr 2014)
25. Spyropoulos, A., Komodakis, N., Mordohai, P.: Learning to detect ground control points for improving the accuracy of stereo matching. In: CVPR. IEEE (2014)
26. Ttofis, C., Theocharides, T.: Towards accurate hardware stereo correspondence: A real-time fpga implementation of a segmentation-based adaptive support weight algorithm. In: DATE (2012)
27. Wedel, A., Meiner, A., Rabe, C., Franke, U., Cremers, D.: Detection and Segmentation of Independently Moving Objects from Dense Scene Flow. In: 7th EMMCVPR (2009)
28. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: ECCV (1994)
29. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research 17, 1–32 (2016)