Pattern Recognition 45 (2012) 1195–1204

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Accurate real-time neural disparity MAP estimation with FPGA

Nadia Baha *, Slimane Larabi

Computer Science Department, University of Science and Technology USTHB, Algiers, Algeria

ARTICLE INFO

Article history:
Received 5 June 2010
Received in revised form
21 July 2011
Accepted 3 August 2011
Available online 30 August 2011

Keywords:
Disparity map
Disparity Space Image (DSI)
Neural network
FPGA

ABSTRACT

We propose in this paper a new method for real-time dense disparity map computing using a stereo pair of rectified images. Based on the neural network and Disparity Space Image (DSI) data structure, the disparity map computing consists of two main steps: initial disparity map estimation by combining the neuronal network and the DSI structure, and its refinement. Four improvements are introduced so that an accurate and fast result will be reached. The first one concerns the proposition of a new strategy in order to optimize the computation time of the initial disparity map. In the second one, a specific treatment is proposed in order to obtain more accurate disparity for the neighboring pixels to boundaries. The third one, it concerns the pixel similarity measure for matching score computation and it consists of using in addition to the traditional pixel intensities, the magnitude and orientation of the gradients providing more accuracy. Finally, the processing time of the method has been decreased consequently to our implementation of some critical steps on FPGAs. Experimental results on real datasets are conducted and a comparative evaluation of the obtained results relative to the state-of-art methods is presented.

1. Introduction

The issue of stereo correspondence is of great importance in the field of Machine Vision. It concerns the features matching between a pair of images of the same scene. When the stereo images are rectified, the matching points will be searched on corresponding horizontal lines and the disparity is calculated as the difference between the abscissas of matched points. The disparity values for all the image points define the disparity map. Once the stereo correspondence problem is solved the depth of the scene can be estimated. This issue is of great interest in the context of 3D reconstruction, virtual reality and robot navigation.

In general, stereo algorithms can be categorized into major classes: local methods and global methods. Local algorithms, which are based on a correlation criterion, can have very efficient implementations that are suitable for real-time application [1–7]. One of the principal factors, which influence the success of local methods, is the proper selection of a window shape and size. The windows must be large enough to capture intensity variation for reliable matching but small enough to avoid the effects of projective distortions at the same time. An appropriate window selection should improve matching accuracy but require an optimized balance between the above opposite criteria [8]. Global approaches minimize an overall cost function that involves all the pixels of the image. In these methods, calculating the disparity field leads to minimize the objective function of energy. Several optimization methods have been proposed such as dynamic programming [9], graph cuts [10], directed anisotropic diffusion [11], belief propagation [12,13] and neural network based approaches [16,17]. The global methods can generate high-quality disparity maps. However, these methods are often computationally expensive and involve difficult parameter adjustment procedures that require a lot of effort to find the optimal ones, making them unsuitable for most interactive applications. Also, there are many other methods that are not strictly included in any of these two broad classes, as example, we can cite [18,19]. A survey for the different approaches can be found in [20,21].

The real-time requirements of most robot applications complicate the realization of such vision systems. The key to success in realizing a reliable embedded real-time-capable stereo vision system is the careful design of the core algorithm. The trade-off between execution time and quality of the matching must be handled with care and is a difficult task.

However, for extracting dense and reliable 3D information from the observed scene, stereo matching algorithms are computationally intensive. To enable both accurate and fast real-time stereo vision in embedded systems, we propose a novel method for computing a dense disparity map based on the combination of Artificial Neural Network and the DSI data structure. The real-time required for such application means that a task has to be finished within an a priori defined time frame [22].

* Corresponding author.
E-mail addresses: nbahatouzene@usthb.dz (N. Baha), slarabi@usthb.dz (S. Larabi).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.005

The goal is to combine the advantages of the neural network and the DSI structure. Our approach divides the matching process into two steps: initial disparity map and refinement of the initial disparity map. Initial disparity map is first approximated by the neuronal-DSI method so called (Neural-DSI). Then a refinement method is applied to the initial disparity so that an accurate result can be achieved. In addition, in order to accomplish real-time operation, we implemented some steps of the disparity map computation on FPGAs: field programmable gate array.

The main contributions of this work are:

- the proposition of a robust matching cost based on the combination of the neural network and the DSI structure,
- the extension of matching primitives from pixel intensity to intensity, gradient magnitude and orientation of gradient vector of pixel for disparity computation of the considered window,
- taking into account of the dominant disparity to avoid any refinement and involving only pixels of the same region in case where the window contains a boundary.

This paper is organized as follows: Section 2 presents the related work in the field of real-time based stereo vision. Section 3 presents the stages followed to compute the initial disparity map. Section 4 presents the refinement method. In Section 5, experimental results obtained on real images are presented and discussed. Finally, Section 6 concludes the paper with some remarks.

2. Related work

Stereo vision is a very broad topic, which has been extensively surveyed by [20,21]. In this section, we present an overview of stereo algorithms reported in the literature. The method proposed by [23] is based on the use of ZNCC as matching cost, integrated within a neural network model. The results obtained are satisfactory, but they are not suitable for real time applications because the running time needed for standard image sets is very high. The method reported in [4] performs interval matching instead of pixel matching. The execution time of the algorithm varies from 1 to 5 s for the standard image sets. A window-based method for correspondence search is presented in [24], which use varying support-weights. The support-weights of the pixels in a given support window are adjusted based on color similarity and geometric proximity to reduce the image ambiguity. The running time for the Tsukuba image pair with 35×35 pixels support window is about 1 min. In the method based on the Bayesian estimation theory described in [25], the results are encouraging in terms of accuracy but they are not suitable for real time applications, since it takes few minutes to process a 256×255 stereo pair with up to 32 disparity levels. The method developed in [26] uses graph cuts, which produces semi-dense disparity map. The running times obtained for the Tsukuba pair is about 6 s and 13 s for the Sawtooth pair. An improvement of the aggregation strategy based on color image segmentation [7] has been proposed by [5]. The processing time achieved by this method is around 0.2 s for Tsukuba with a disparity range of 16 pixels. For the cost aggregation method presented in [6], the running time for the Tsukuba image pair is 13 s and 37 s for Teddy image pair. Another method reported by [9] uses a two-pass dynamic programming technique combined with generalized ground control points (GGCPs), which is designed to resolve the inconsistency between scanlines, which is the typical problem in conventional dynamic programming. The processing time achieved by this method is around 4.4 s for Tsukuba image pair with a disparity range of 16 pixels. In another method reported in [16] based on Self-Organizing Neural Network, the average execution time is approximately 100 s for the standard image sets.

The idea, which motivates this work, is to propose a novel aggregation cost deploying neural network and DSI data structure aiming at low computation time and at the same time as accurate as to improve the results of fast local stereo algorithms. This leads us to yield a level of accuracy comparable to that of global methods and able to meet near-real time processing requirement.

3. Initial disparity map estimation: neural-DSI

We propose in this section the steps allowing the computation of the initial disparity map using the combination of neural network and Disparity Space Image (DSI).

3.1. Computation of the Disparity Space Image (DSI)

3.1.1. Disparity computation

Assuming that images pairs are rectified, the search for correspondence of each feature in one image will be done in the same horizontal line of the other image. For each pixel $p_l(x_l, y_l)$ in the left image (reference image), the disparity computation will concern all pixels of the windows W_l of the left image centered on p_l and W_r^d of the right image centered on p_r instead of the use only of p_l and its match p_r. The position of W_r^d depends on the disparity $\mathbf{d}$ associated to the pair (p_l, p_r), which varies from zero to $d_{\max}$, where $d_{\max}$ represents the highest disparity value of the stereoscopic images (disparity range). The match $p_j(x_j, y_j)$ of each pixel $p_i(x_i, y_i)$ of W_l will be searched in the window W_r^d (see Fig. 1) so as $x_j=x_i+sd$, $y_j=y_i$, $s=\{+1,-1\}$ is a sign of disparities.

3.1.2. Disparity Space Image

Disparity Space Image (DSI) is an explicit representation of the matching space introduced by Bobik and Intille [14]. It plays an essential role in the development of the overall matching algorithm, which uses the occlusion constraints. Thus, it has the advantage of improving disparities in occluded areas.

For a given value of $\mathbf{d}$, disparity space image for the pixel p_i is defined as the score $DSI^d(p_i)$ computed using pixels attributes.

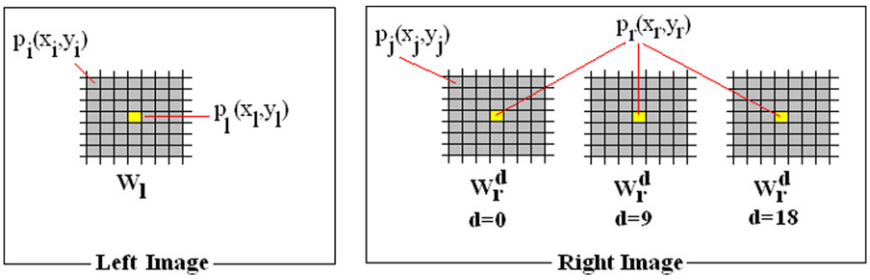

Fig. 1. Disparity computation.

Many measures have been proposed in the literature [28] using the module and orientation of the gradient vector.

What we propose in this work is a linear combination of the three intensity, gradient magnitude and gradient orientation as features for computing matching measure between p_i and the pixel p_j translated by the distance d relatively to p_i.

$$DSI^d(p_i) = (DSI_I^d(p_i) + DSI_G^d(p_i) + DSI_O^d(p_i))n_2 \quad (1)$$

where

$$DSI_I^d(p_i) = ((I_l(x_i, y_i) - I_r(x_i + sd, y_i))n_1)^2 \quad (2)$$

$$DSI_G^d(p_i) = ((G_l(x_i, y_i) - G_r(x_i + sd, y_i))n_1)^2 \quad (3)$$

$$DSI_O^d(p_i) = ((O_l(x_i, y_i) - O_r(x_i + sd, y_i))n_1)^2 \quad (4)$$

where (I_l, I_r), (G_l, G_r), (O_l, O_r) are, respectively, the intensities, gradient magnitudes and gradient orientations values of the pixels on the left and right images, n_1, n_2 are the input weights of the neural network, which will be computed in the step of learning of the neural network as will be presented in Section 3.3.

The intensity of an arbitrary pixel is given by $I(x, y)$, the gradient is defined as

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \quad (5)$$

Its magnitude (module) is defined as

$$|G| = |G_x| + |G_y| \quad (6)$$

The orientation $O(x, y)$ of the gradient vector is

$$O = \tan g^{-1}(G_x/G_x) \quad (7)$$

3.1.3. Initial disparity computation

To determine the initial disparity d^* (p_l) of the central pixel $p_l(x_l, y_l)$ of the window W_l, we calculate for all neighboring pixels p_i of p_l the score $DSI^d(p_i)$ using different values of $\mathbf{d}$ ($\mathbf{d}=0,\ldots,d_{\max}$) and we choose the value of d^* giving to p_l the minimal cost among all computed $DSI^{d^*}(p_i)$.

3.2. Neural network architecture

The Artificial Neural Network (ANN) is a network of neurons that is trained to provide the right output for giving some inputs. The neurons have some weighted inputs and are responsible for simple operations, but the whole network can make parallel calculations due to its wide parallel structure [30]. The neural network derives its computing power from its massive parallel distributed structure and from its ability to learn and, then to generalize. The generalization refers to the production by the network of reasonable outputs for inputs not encountered during training [30].

In our previous work [31], we proposed a multilayer feed forward perceptron model, trained with the supervised back propagation learning algorithm [32] to compute disparities. However, the results obtained are satisfactory in terms of accuracy, but they are not suitable for real time applications because the processing time needed for standard image sets is very high (see Fig. 7). In the present study, a multilayer feed forward model based on simple supervised learning procedure was adopted in order to improve the processing time. This procedure can be found in detail in Section 3.3

The proposed neural network is composed of four layers (see Fig. 2). Each layer is responsible for a specific task (input, score computation, decision and output). As the neurons perform simple operations, their input weights and transfer functions are adjusted as follows.

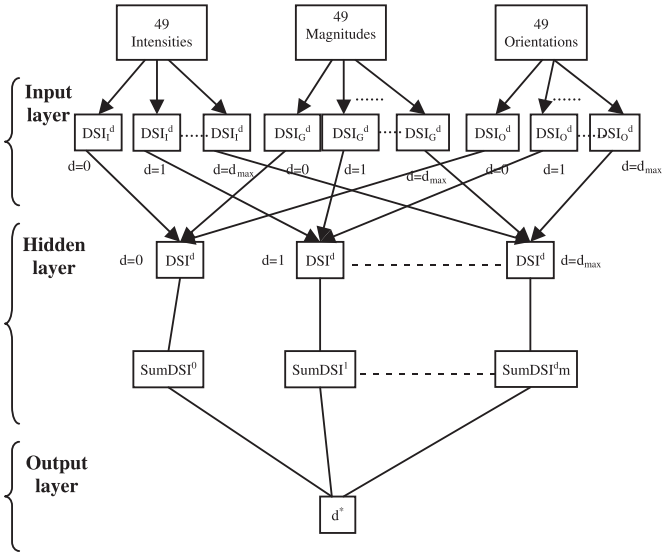

Fig. 2. Neural-DSI network architecture.

The input layer associated to a windows W_l ($7 \times 7 = 49$ pixels) is constituted by 147 neurons of (49 neurons intensities, 49 neurons gradient magnitudes and 49 neurons orientations) has the function to compute the scores DSI_I^d, DSI_G^d and DSI_O^d for each one pixel of the window W_l as given in Eq. (2), (3) and (4). The transfer function is $f(x) = sqr(x)$ and has as input weight the value n_1. We obtain then for each value of the disparity $\mathbf{d}$ $(0,\ldots,d_{\max})$ three (7×7) matrices of scores.

To compute the final DSI^d, the second layer adds the three correlation scores for each one pixel of the window (see Eq. (1)). We obtain $d_{\max}+1$ matrices of 7×7 scores. The transfer function is linear and the input weights are identical and equal to n_2. In the third layer, for each value of $\mathbf{d}$, all scores of the W_l pixels are added and constitutes the score $SumDSI^d$ of the central pixel. Then, to the central pixel of the window is associated a vector of costs (Aggregation cost) $AC = (SumDSI^0, SumDSI^1, \ldots, SumDSI^{d_{\max}})$. The input weights are the same and equal to 1 and the transfer function is linear. In the fourth layer, the minimum cost amount of the $d_{\max}+1$ costs is chosen as the best score and defines the disparity d^* of the central pixel of the window. The input weights are the same and equal to 1 and the transfer function is linear $f(x) = \mathrm{Min}(x)$.

3.3. Learning procedure

The neural correlation network must be trained with a supervised learning procedure before computing the minimum of AC (best score) for each pixel. In general, neural supervised learning is based on the presentation to the network of a set of training examples having the structure [(features); (expected value)]. In our context, the feature designates a given training example of matching pixels (p_l, p_r). The expected value is the correlation score of (p_l, p_r). The first goal of learning procedure can be formulated as the search for the appropriate weights.

To prepare the training data, some points of interest are extracted and their attributes (gradients magnitude and orientations) are computed from the stereoscopic pair of images Cones, Barn2, Sawtooth, Teddy available on the Middlebury stereo evaluation website [35]. These points are selected depending on their high values of intensity, gradient magnitude and orientation of the gradient vector. The matching of these points is done using normalized correlation *ZNCC* [29] method considering the left image to the right image and vice versa. A valid match is considered only for those points that yield the best correlation score. 150 unmatched

pixels and 50 matched pixels are selected to train offline the network. During training, the differences of intensities, gradient magnitudes and orientations between two local windows (one for the left image, the other for the right) are fed to the network. Our method is based on exhaustive search of the best weights, in such a manner to minimize the error of the matched pixels and to maximize the difference between the error of the matched and unmatched pixels. In our case, we defined two input weights: n_1 used in the first layer and n_2 used in the second layer. After the training, the network should have the ability to differentiate the matched pairs from unmatched ones.

4. Disparity map refinements

The resulting disparity map described above is not the optimal one because it contains still some noise and errors. We propose in the following our approach for refinement of the disparity map. Even if there exist more accurate techniques for the sub-pixel refinement in the literature [33,34], they are computationally too expensive for real-time stereo vision. In the next, we present our approach, which has a sufficient accuracy for our purpose and a low complexity for its implementation on FPGA circuit.

4.1. Refinement method

The proposed refinement method is inspired from the one proposed in [15]. Unlike conventional techniques whose further steps of matching algorithm are based on the minimal computed cost, Binaghi et al. use all scores obtained [15]. Instead of selecting in the window W_l only one disparity having the best score, they propose to select for each pixel **n** disparities corresponding to the best scores. In particular, for each pixel in the reference image the costs are ranked and winners are identified. For a selected pixel, the number of disparity winners within a given neighbor (window) is computed. The disparity with the highest number of confirmation is finally selected.

Assuming that initial disparities of all pixels of the left image are computed, we extended the Binaghi et al. method [15] by adding two improvements in order to reduce the computation time and to obtain a better accuracy.

4.1.1. First improvement

In contrast to the work of [15], where the authors compute systematically the disparity value using **n** disparities, for each pixel p_l in the left image, we first verify if the disparity is dominant in the window W_l centered on p_l. If it is the case, this disparity will be considered as the final disparity and not necessitates any refinement. Otherwise, we do a refinement, which consists of selecting in W_l window three best scores of $SumDSI^d$ for each pixel. Three best disparities are then associated to the pixel p_l instead of one disparity. The proposed process consists of applying a vote in order to choose the dominant disparity in the associated W_l using the three disparities of the central pixel p_l and of the 48 neighboring pixels.

Fig. 3 illustrates how we compute the new disparity using the three best disparities for each pixel when we apply the vote process. The disparity that will obtain the highest number of points will be considered as the new disparity.

For the definition of the parameter **n**, we decided to use an experimental evaluation of over existing stereo datasets [35]. The number **n**=3 corresponding to the three best disparities values has been selected. If we take **n** greater than 3 it did not lead to an increase in accuracy but an increase in processing time. Fig. 4 shows a dependence of accuracies achieved with different numbers of disparities on the cone image. For **n**=4 the accuracy

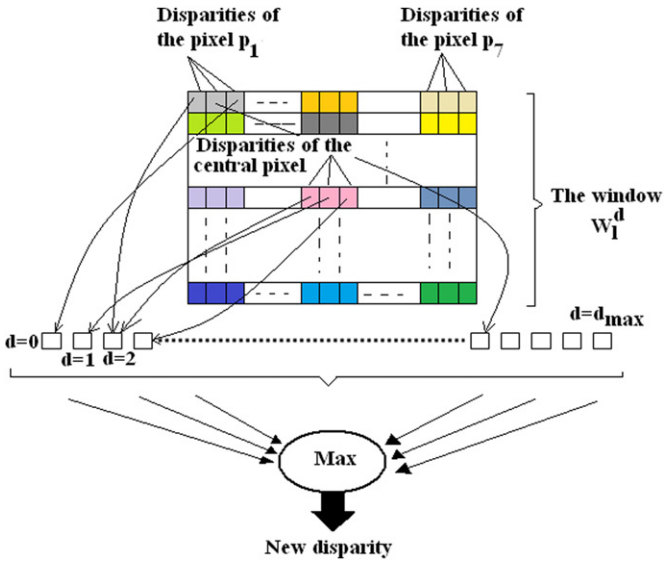

Fig. 3. Disparity computing with refinement method.

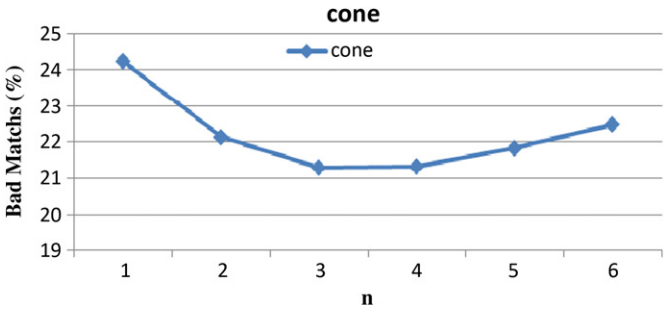

Fig. 4. Accuracy obtained with different values of *n*.

increases just 0.029% while the processing time increases of 0.07 s. This is the main reason why increasing the **n** value beyond a certain value does not improve accuracy any further.

4.1.2. Second improvement: processing of region boundaries problem

The problem of region boundaries has not been addressed in [15]. The region boundaries problem occurs when the pixels of the same window belong to two different regions (see Fig. 5). Indeed, the pixels of the two sides of the contour have usually different disparities. Consequently, we must consider in computation of the new disparity (final disparity) only the pixels belonging to the same region. The boundary between two regions is detected using the gradient magnitude.

To take into account this problem, we propose a second improvement of the method proposed in [15], which uses all pixels of the window W_l, by adding a criterion, which eliminates from the vote process described above all pixels of the window W_l located in the second region at the right of the boundary, which is detected using the gradient magnitude information. With this improvement, the disparity map is more accurate because only the pixels of the window belonging to the same region (object) are involved in the calculation of the new disparity. Since, the pixels belonging to two different regions have different disparities. Fig. 5 illustrates how we compute the new disparity in the case the window W_l is positioned on two regions (objects).

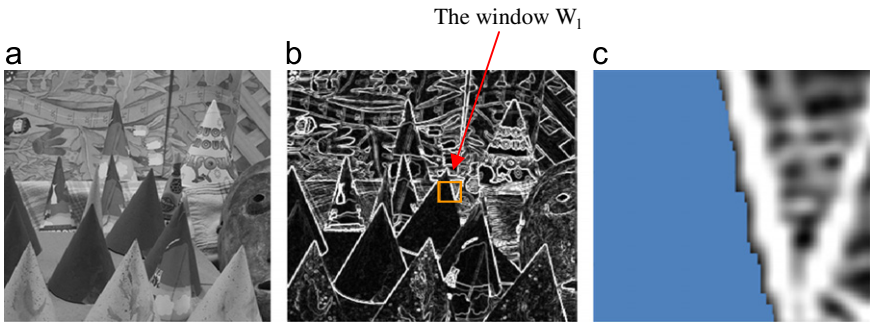

Fig. 5. Example of region boundaries problem: (a) reference image, (b) case where the window W_l contains pixels of two regions, (c) blue color represents the pixels of the same region of the window W_l involved in the computation of the new disparity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Disparity map smoothing

Finally, in order to keep the good trade-off between accuracy and processing time, a simple median filter is applied for smoothing the disparity map. The median filter is a robust method, often used to remove the impulsive noise known for its salt and pepper noise from an image [36–40]. This is the type of noise, which is present in disparity maps. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel. Processing time increases considerably when the size of the mask increases giving noticeable image blur for large mask sizes. This is the main reason the block size must be limited because it does not improve the accuracy any further.

5. Experimental results

In this section, we describe the experiments conducted to evaluate the performance of the proposed method. Our aim is to obtain an accurate disparity map and a fast runtime, which is the requirement for any obstacle detection system of autonomous mobile robot navigation. For this purpose, an extensive performance evaluation and comparison between different methods is proposed. The two criteria used for the evaluation are then accuracy and computation cost.

Several parameters have been mentioned and discussed in the next sub-sections. Qualitative tests through disparity map observation were carried out with four stereo couples [35] to find the influence and appropriate values of those parameters.

5.1. Initial disparity map results

In order to study the efficiency of the combination of the neural network and DSI concept, two other approaches were implemented: the neuronal method [31] called (Neural) and the DSI method described in [14,15] called (DSI). We applied these methods on four images (Cones, Barn2, Sawtooth, Teddy) of standard datasets available on the Middlebury stereo evaluation website [35]. Figs. 6 and 7 illustrate, respectively, the results of the initial disparity map obtained for three selected methods and the correspondent processing time (s). The three selected methods were implemented using the C++ language and the timing tests were performed on a Personal computer PC, 2.5 GHZ. We can clearly see (Fig. 7) that our approach (Neural-DSI) is relatively the fastest among DSI and neural methods.

5.2. Refinement disparity map results

We implemented our proposed method for disparity map refinement as the neural refinement method [31]. Fig. 8 shows the disparity maps obtained for the two methods. We can see that, as depicted, our refinement method gives a better map than the neural refinement method [31]. Fig. 9 illustrates the processing time of the two methods applied on the five image pairs. Also, the proposed refinement is faster compared to the neural refinement method.

We studied also the influence of window size on the accuracy of the proposed method. Fig. 10 shows the disparity map obtained after applying our refinement method for the Barn1 image pair for different sizes of the window. Greater the size of the window W, better the computed map disparity. Nerveless, the computation time is also very high when the size of W is large. Fig. 11 illustrates the variation of time processing for three methods Neural, DSI and our method (Neural-DSI).

Experiments are conducted in order to study the influence of d_{max} values on the performance of the Neural-DSI method. We use different values of this parameter for disparity map estimation of four stereo images pairs with 1×7 window. Fig. 12 illustrates the processing times obtained and shows that the Neural-DSI method is faster. Indeed, the processing time obtained is less than 0, 2 s for the Map image.

5.3. Discussion and comparison

This section presents a comparison between the proposed method (Neural-DSI) and other state-of-art methods. The results obtained by our algorithm are better than some methods reported in the literature [21,41]. Table 1 shows a comparison of stereo vision implementation reported in the literature in terms of computation time. The description of the systems introduced here is restricted to the system platform, the basic matching strategy, the image size and the processing time achieved. A ranking of each method according to the computation time is shown in the second column.

Scharstein and Szeliski [20] have developed an online evaluation platform, the Middlebury Stereo Evaluation [35], which provide a number of stereo image datasets consisting of the stereo image pair and the appropriate ground truth image. To evaluate an algorithm on this website, disparity maps of all datasets have to be generated and uploaded. The disparity maps have to correspond to the left stereo image and the disparities have to be scaled by a certain factor. The evaluation engine calculates the percentage of bad matched pixels, within a certain error threshold, by pixel-wise comparison with the ground truth image. Many stereo algorithm developers use this platform for evaluation. This gives a significant overview of how the developed algorithm performs in comparison to other algorithms. The platform is up-to-date and constantly growing. Parameters ALL and NOCC are defined according to the Middlebury website [35].

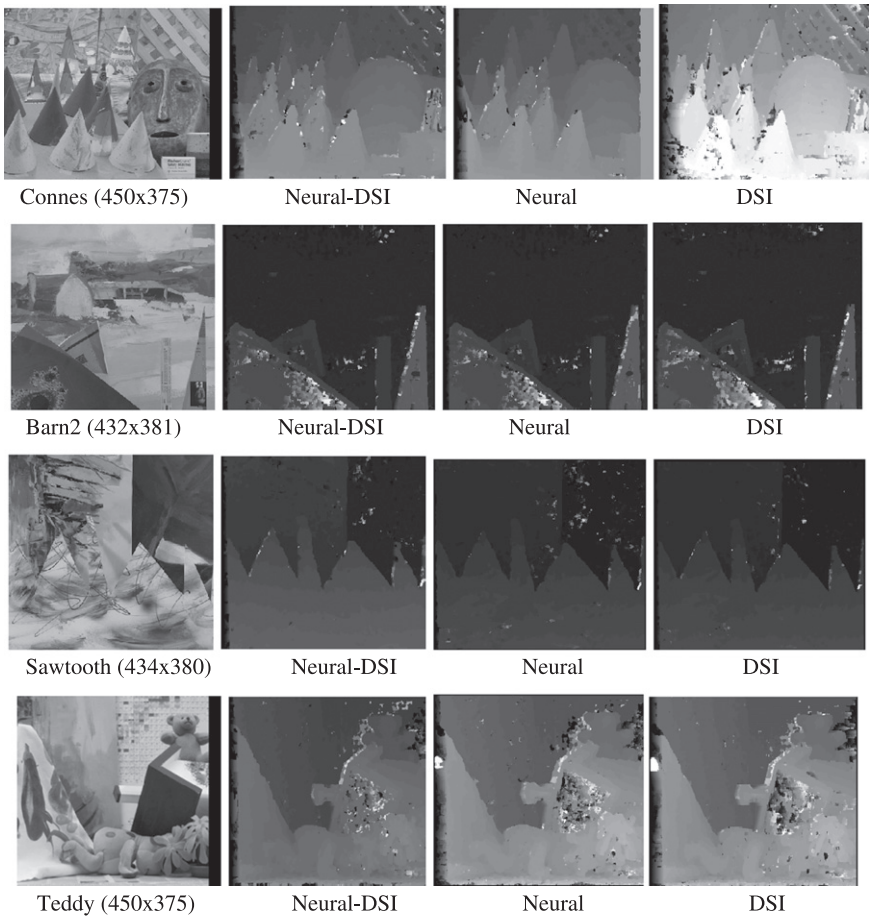

Fig. 6. Initial disparity maps obtained by the Neural-DSI, Neural and DSI methods: top to bottom the reference images; left to right: our method, Neural and DSI methods.

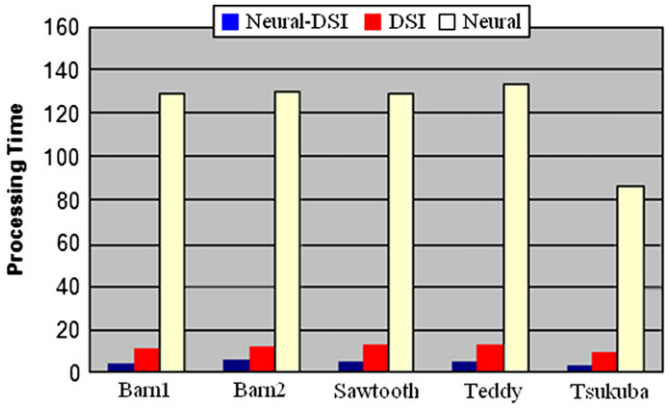

Fig. 7. Processing time (s) of Neural-DSI, Neural and DSI methods on the five image pairs.

ALL is the error computed on the whole image and NOCC is the error computed on the whole image excluding the occluded region. Among the quality measures proposed by [20] in their paper we adopted the percentage of bad matching pixels between the computed disparity map $d_C(x, y)$ and the ground truth map $d_T(x, y)$

$$PBP = (1/N \sum(|d_C(x,y) - d_T(x,y)| > \delta_d)) \quad (8)$$

where δ_d is the error disparity deviating from the ground truth by more than 1 pixel.

In this new real-time disparity map estimation method, which is the extension of our previous work [27], we propose a qualitative evaluation of our method using the Middlebury Stereo Evaluation [35]. Fig. 13 illustrates the influence of the window size on the accuracy of our method for four images.

Table 2 shows the comparisons results in terms of accuracy of the disparity maps obtained by some stereo vision methods reported in the literature. We use four reference stereo pairs and for each of them evaluate the error rates on the two ground truth maps Nocc and All. Similar to the evaluation of computation time, Table 2 shows the ranking of methods according to the accuracy. Accuracy corresponds to the percentage of the correct matched pixels. Finally, Table 3 reports in the rightmost column the ranking obtained by averaging the overall accuracy ranking and the time ranking, so as to highlight the methods that better trade-off between accuracy and computational efficiency. Hence, overall our approach can be regarded as an interesting trade-off between accuracy and speed.

5.4. FPGA implementation

Due to the computational complexity of many stereo algorithms, a number of attempts have been made to implement such systems using reconfigurable hardware in the form of Field programmable gate Arrays (FPGAs) [43–45]. These devices consist of programmable logic gates and routing, which can be re-configured to implement practically any hardware function. Hardware implementations on one hand allow the application of the parallelism that is common in image processing and vision algorithms, and on the other hand the building of systems to

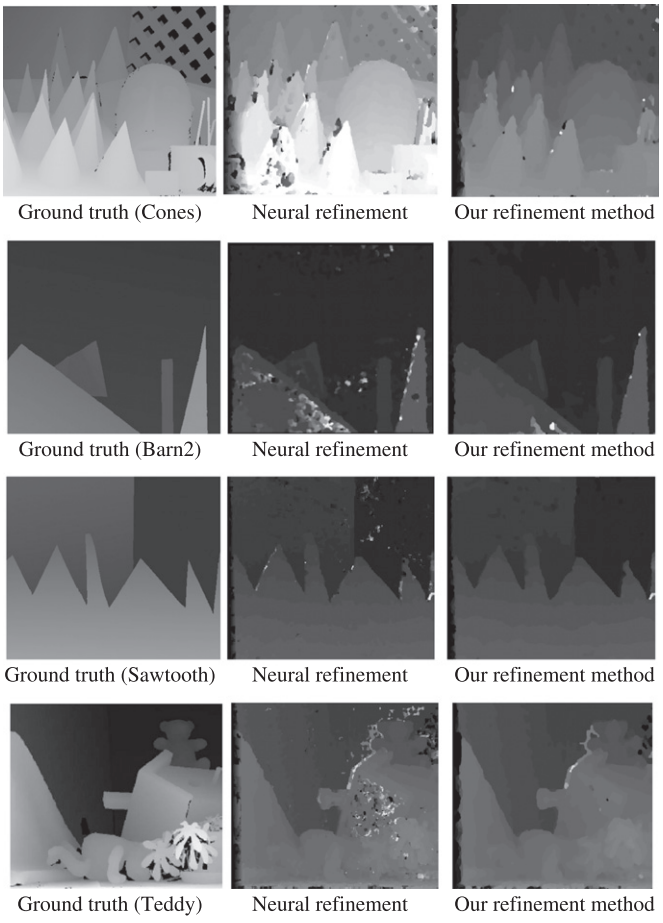

Fig. 8. Final disparity maps obtained by the neural refinement method and our refinement method.

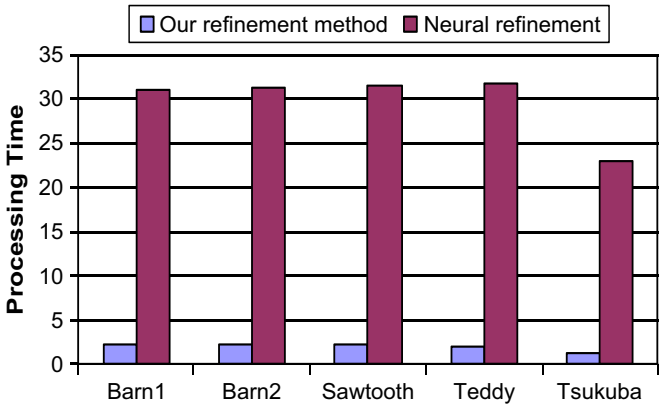

Fig. 9. Processing time (s) obtained by our refinement method and Neural method.

perform specific computation quickly compared to software implementations [46].

In order to reduce the computing time, we implemented the disparity map computation approach on FPGA, the processing time is reduced considerably. The architecture corresponding to our method (Neural-DSI) is split into three major pipeline stages: the pre-processing, calculation and post-processing stage. The first stage is the input stage, which supplies the image data for the computation. At this stage, the Sobel operator is used to calculate the gradient value in x and y directions. The calculation stage

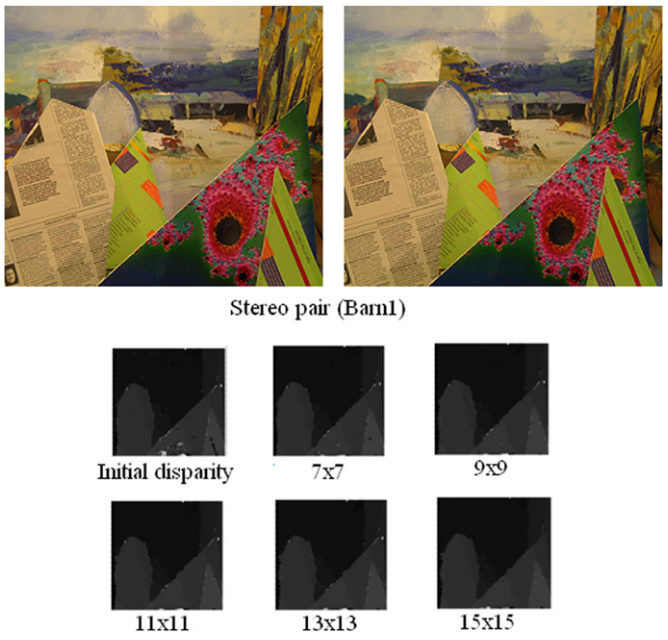

Fig. 10. Application of the refinement method on Barn1 image for different window sizes.

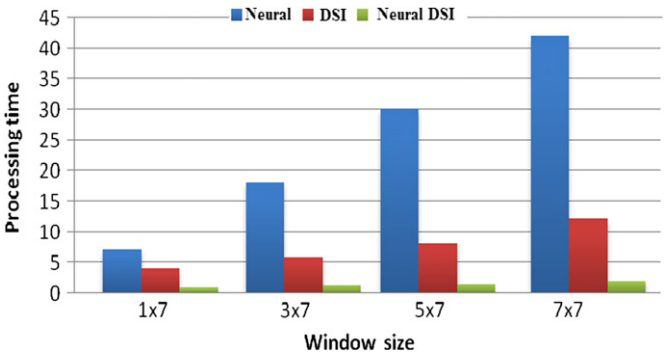

Fig. 11. Processing time (s) of the Neural, DSI and Neural-DSI methods for different window sizes.

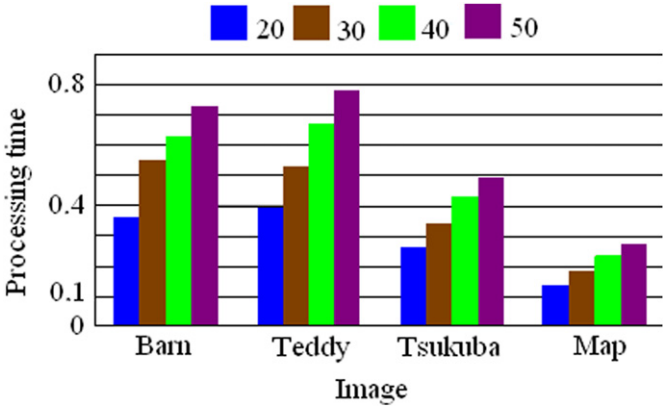

Fig. 12. Processing time (s) of disparity map computing of Neural-DSI for different values of d_{max} on four images pairs for 1×7 window.

computes the Neural-DSI algorithm, which will give us the initial disparity map. Finally, the initial disparity map is refined in the post-processing stage using a median filter [47].

Table 1
Comparison of stereo vision implementations ("/" means not available).

Author	Time (s)	Algorithm	Image size	Machine
Tombari et al. [5]	0.2 s 1	Aggr. Stra. based on color segm.	Tsukuba	2.4 GHz Intel Core Duo
Our method	**0.26 s**	**Neural-DSI Neural-DSI+Refinement**	**Tsukuba**	**2.5 GHz Centrinon Microprocessor**
	0.7 s 2			
Gerrits and Bekaert [7]	2 s 3	Segment based	Teddy	2.4GHz Intel Core Duo
Kim et al. [9]	4.4 s 4	Dyn. Prog.	Tsukuba	2.4 GHz Pentium IV
Ogale and Aloimonos [4]	1–5 s 5	/	All images	/
Veksler [26]	6 s 6	Graph-cut	Tsukuba	0.6 GHz, Pentium III
Tappen and Freeman [12]	183 s 11	Accelerated belief prop.	Map	2.4 GHz Pentium IV
Mattocia [6]	13 s 7	LC locally consist.	Tsukuba	2.5 GHz Intel Core Duo
Yoon and Kweon [24]	60 s 8	Window-based	Tsukuba	AMD 2700
Vanetti et al. [16]	100 s 9	Self org. map	All images	1.8 GHz AMD processor
Venkatesh et al. [17]	120 s 10	Self org. map	256×256	1.4 GHz Pentium IV
Gutierrez and Marroquin [25]	Few minutes 12	Bayesian estimation	256×255	/
Tombari et al. [42]	33 mn 34 s 13	Segment support	Teddy	2.4 GHz Intel core Duo

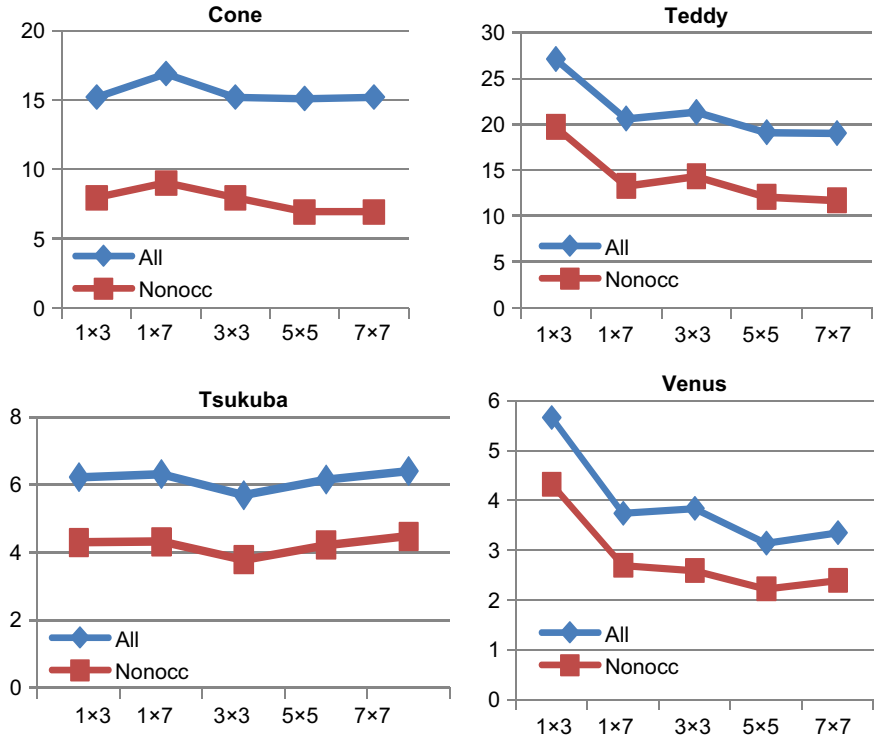

Fig. 13. PBP obtained by our method (Neural-DSI) on four image pairs for different window sizes.

Table 2
Accuracy according to the methodology defined by the Middlebury website (/ means not available).

Method	Cones		Teddy		Tsukuba		Venus		Accuracy (%)
	ALL	NOCC	ALL	NOCC	ALL	NOCC	ALL	NOCC	
Vanetti et al. [16]	12.4	6.31	15.73	10.41	3.76	3.38	1.42	0.98	93.21 2
Our method	**15.2**	**7.97**	**21.3**	**14.3**	**5.69**	**3.78**	**3.84**	**2.6**	**90.7 5**
Mattocia [6]	15.1	4.75	18.3	9.3	3.44	1.77	1.74	0.27	92.43 3
Yoon and Kweon [24]	16	5.5	21.6	12.7	6.68	4.66	6.18	4.61	91.08 4
Veksler [26]	27.3	29.6	25.5	25.9	4.86	3.12	3.87	2.42	84.68 8
Tombari et al. [5]	/	/	/	/	/	/	/	/	86.4 7
Tombari et al. [42]	3.77	9.87	8.43	14.2	1.25	1.62	0.25	0.64	94.9 1
Venkatesh et al. [17]	32.83	27.93	28.11	23.62	23.6	22.97	15.91	15.17	76.23 9
Gerrits and Bekaert [7]	/	13.22	/	15.78	/	8.18	/	8.06	88.7 6
Tappen and Freeman [12]	/	/	/	/		4.1	/	/	/
Kim et al. [9]	/	/	/	/	1.53	/	0.94	/	/

Table 3
Comparison of stereo vision implementations in terms of accuracy and computation time.

Method	Rank time	Rank accuracy	Average rank
Our method	**2**	**5**	**3.5**
Mattocia [6]	5	3	4
Tombari et al. [5]	1	7	4
Tombari et al. [42]	9	1	5
Vanetti et al. [16]	7	2	4.5
Yoon and Kweon [24]	6	4	5
Veksler [26]	4	8	6
Venkatesh et al. [17]	8	9	8.5
Gerrits and Bekaert [7]	3	6	4.5

Table 4
Processing time (ms) for software implementation.

Method	1	2	3
Pair 1	147	73	468
Pair 2	163	780	470
Pair 3	107	490	312.5

Table 5
Processing time (ms) for FPGA implementation.

Method	1	2	3
Pair 1	2.99	14.99	3.46
Pair 2	3.07	15.36	3.55
Pair 3	2.01	10.07	2.32

To demonstrate the importance of the use of FPGA circuits, Table 4 illustrates the processing time obtained for each component in traditional implementation (Soft) where:

- Pair 1, pair 2 and pair 3 are, respectively, Barn1 (432×381), Teddy (450×375) and Tsukuba (384×288).
- Methods 1, 2 and 3 correspond, respectively, to Gradient (Sobel), Neural-DSI (with 1×7 window and $d_{max}=50$) and Median Filter.

Table 5 illustrates the processing time obtained for each component using FPGA implementation. Not surprisingly the running times obtained with the use the FPGA are better. The processing time for one image pair was 490 ms, which is around 49 times slower than our hardware implementation and it seems obvious that even with algorithmic and software optimizations, the processor-based system cannot outperform the FPGA-based solution. All the reasons make FPGA implementation preferable.

6. Conclusion

In this paper, we proposed a novel disparity map estimation algorithm based on the combination of neural network and DSI data structure. Disparity map computing process is divided into two main steps. The first step deals with computing the initial disparity map using a neuronal method and DSI structure. The second step presents a simple and fast method to refine the initial disparity map so an accurate result can be achieved. Using this method (combination of neural network and DSI structure), we reached the results of global methods without sacrificing the simplicity, flexibility and speed of local aggregation methods. Thus, our method can be regarded as an interesting trade-off between accuracy and speed.

The computation time mainly depends on the image size, the size window and the value of highest disparity in the image (d_{max}).

As this work is intended to be used for obstacle detection system of autonomous mobile robot navigation, we implemented our algorithm on FPGA decreasing the processing time considerably.

References

[1] L. Di Stefano, M. Marchionni, S. Mattoccia, A fast area-based stereo matching algorithm, Image and Vision Computing 22 (12) (2004) 983–1005.
[2] R. Maas, B. Haar Romeny, M. Viergever, Area-based computation of stereo disparity with model based window size selection, Computer Vision and Pattern Recognition (CVPR) (1999) 106–112.
[3] S. Kumar, B. Chatterji, Stereo matching algorithms based on fuzzy approach., International Journal of Pattern Recognition and Artificial Intelligence 16 (7) (2002) 883–899.
[4] A. Ogale, Y. Aloimonos, Shape and the stereo correspondence problem, International Journal of Computer Vision (IJCV) 65 (3) (2005) 147–1758.
[5] F. Tombari, S. Mattoccia, L. Di Stefano, Near real-time based on effective cost aggregation, ICPR (2008).
[6] S.A. Mattocia, Locally global approach to stereo correspondence, in: Proceedings of the ICCV, Twelfth International Conference on Computer Vision Workshops, 2009, pp. 1763–1770.
[7] M. Gerrits, P. Bekaert, Local stereo matching with segmentation-based outlier rejection, in: Proceedings of the Conference on Computer and Robot Vision, 2006.
[8] T. Kanade, M. Okutomi, A stereo matching algorithm with an adaptive window: theory and experiment, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 920–932.
[9] J. Kim, K. Lee, B. Coi, S. Lee, A dense stereo matching using two-pass dynamic programming with generalized ground control points, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2005, pp. 1075–1082.
[10] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts? ECCV (3), Lecture Notes in Computer Science (2002) 65–81.
[11] A. Banno, K. Ikeuchi, Disparity map refinement and 3D Surface smoothing via directed anisotropic diffusion, in: Proceedings of the Twelfth ICCV Workshops, 2009, pp. 1870–1877.
[12] M. Tappen, W. Freeman, Comparison of graph cuts with belief propagation for stereo, using identical MRF, parameters, in: Proceedings of the International Conference on Computer Vision, ICCV, 2003.
[13] Q. Yang, L. Wang, R. Yang, H. Stewenius, D. Nister, Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI (2008).
[14] A. Bobik, S. Intille, Large occlusion stereo, International Journal on Computer Vision 33 (1999) 181–200.
[15] E. Binaghi, I. Gallo, C. Fornasier, M. Raspanti, Growing aggregation algorithm for dense two-frame stereo correspondence, in: Proceedings of the First International Conference on Computer Vision Theory and Application, 2006. pp. 326–332.
[16] M. Vanetti,I. Gallo,E. Binaghi, Dense two-frame stereo corresponence by self-organizing neural network, in: Image Analysis and Processing, LNCS, Springer, 2009, pp. 1035–1042.
[17] Y.V. Venkatesh, S.K. Raja, M. Raspanti, Neural disparity computation for dense two-frame stereo correspondence, IEEE Transactions on Image Processing 16 (11) (2007) 2822–2829.
[18] T.E. Zickler, J. Ho, D.J. Kriegman, J. Ponce, P.N. Belhumeur, Binocular helmholtz stereopsis, in: Proceedings of the IEEE Conference Proceedings of International Conference on Computer Vision, ICCV, vol. 2, 2003, pp. 1411–1417.
[19] M. Lhuillier, L. Quan, Robust dense matching using local and global geometric constraints, in: IEEE Conference Proceedings of the International Conference on Pattern Recognition, ICPR, vol. 1, 2000, pp. 968–972.
[20] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International Journal of Computer Vision (2002) 47 7–42.
[21] L. Nalpantidis, G. Sirakoulis, A. Gasteratos, Review of stereo matching algorithms: from software to hardware, International Journal of Optomechtonics 2 (2008) 435–462.
[22] H. Kopetz, Real-Time Systems, Design Principles for Distributed Embedded Applications, Springer, 1997.
[23] E. Binaghi, I. Gallo, C. Fornasier, M. Raspanti, Neural adaptative stereo matching, Pattern Recognition Letters 25 (2004) 1743–1758.
[24] K. Yoon, I. Kweon, Adaptive support weight approach for correspondence search, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006).
[25] S. Gutierrez, J. Marroquin, Robust approach for disparity estimation in stereo vision, Image and Vision Computing (2004) 83–195.
[26] O. Veksler, Extracting dense features for visual correspondence with graph cuts, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and pattern Recognition, 2003.

[27] N. Baha, S. Larabi, Towards real-time Neuronal disparity map estimation, in: Proceedings of the Third International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications VISAPP, 2010, pp 355–360.
[28] T. Twardowski, B. Cyganek, J. Borgosz, Gradient based dense stereo matching, Lecture Notes in Computer Science 3211 (8) (2004) 721–728.
[29] R.C. Gonzalez, R.E. Woods, Digital Image Processing, second ed., Pearson Education International, 2002.
[30] S. Haykin, Neural Network: A Comprehensive Foundation. Pentice Hall. 1998.
[31] N. Baha, S. Larabi, Disparity map estimation with neural network, in: Proceedings of the IEEE ICMWI 2010, International Conference on Machine and Web Intelligence, 3–5 October 2010, pp. 282–285.
[32] H. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representation by error propagation. Parallel Distributed Processing, 1986, pp. 318–362.
[33] M. Shimizu, M. Okutomi, Precise sub-pixel estimation on area based matching, in: Proceedings of the Eighth IEEE International Conference on Computer Vision, 2003.
[34] E.Z. Psarakis, G.D. Evangelidis, A generic implementation framework for FPGA based stereo matching, in: Proceedings of the IEEE Region Tenth Annual Conference on Speech and Image Technologies for Computing and Telecommunications, 1997.
[35] Middlebury stereo evaluation, 〈http://vision.middlebury.edu/stereo〉.
[36] G.R. Arce, Nonlinear Signal Processing: A Statistical Approach, Wiley, New Jersey, USA, 2005.
[37] G. Egnal, R.P. Wildes, Detecting binocular half-occlusions : empirical comparisons of five approaches, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 24 (8) (2002) 1127–1133.
[38] A. Koschan, Dense stereo correspondence using polychromatic block matching, in: Proceedings of the International Conference on Computer Analysis of Images and Patterns, CAIP, vol. 719 of Lecture Notes in Computer Science, 1993, pp. 538–542.
[39] K. Muhlmann, D. Maier, J. Hesser, R.M. Manner, Calculating dense disparity maps from color stereo images, an efficient implementation, in: IEEE Conference Proceedings of Workshop on Stereo and Multi-Baseline Vision, SMBV, 2001, pp. 30–36.
[40] M.P. Eklund, A.A. Farag, Robust correspondence methods for stereo vision, International Journal of Pattern Recognition and Artificial Intelligence, PRAI 17 (7) (2003) 1059–1079.
[41] F. Tombari, S. Mattoccia, L. Di Stefano, Classification and evaluation of cost aggregation methods for stereo correspondence, in: Proceedings of the CVPR, 2008.
[42] F. Tombari, S. Mattoccia, L. Di Stefano, Segmentation-based adaptive support for accurate stereo correspondence, in: Proceedings of the Pacific-Rin Symposium on Image and Video Technology, 2007.
[43] K. Ambrosch, M. Humenberger, W. Kubinger, A. Steininger, SAD-based Stereo Matching using FPGAs, in Embedded Computer Vision Part II, Springer, 2009, pp. 121–138.
[44] C. Murphy, D. Lindquist, A.M. Rynning, T. Cecil, S. Leavitt,M. Chang, Low cost stereo vision on an FPGA, in: Proceedings of the Fifteenth IEEE Symposium on FPGAs Custom Computing Machines, 2007.
[45] D.K. Masrani, W.J. MacLean, A real-time large disparity range stereo-system using FPGAs, in: Proceedings of the IEEE International Conference on Computer Vision Systems, 2006.
[46] D. Han, Real-time object segmentation of the disparity map using projection based region merging, in: R. Stolkin (Ed.), Scene Reconstruction, Pose Estimation and Tracking, 2007 (ISBN 978, 530).
[47] M.A. Vega-Rodríguez, J.M. Sánchez-Pérez, J.A. Gómez-Pulido, An FPGA-Based Implementation For Median Filter Meeting the Real-time Requirements of Automated Visual Inspection Systems, 2002.

Nadia Baha received the Magister in Computer Science at CDTA in 1991. She is currently a Ph.D. student and a researcher at the Computer Science Institute of University of Sciences and Technologies, USTHB, Algiers, Algeria. She is the author of numerous publications for conferences and proceedings. Her research interests include computer vision and mobile robot navigation.

Slimane Larabi received his Ph.D. in Computer Science from the Polytechnic National Institute of Toulouse, France, in 1991. He is currently a Professor at the Computer Science Institute of University of Sciences and Technologies. Algiers, Algeria. He is the head of the Computer Vision group at the Laboratory of Artificial Intelligence of the same university, and the author of numerous publications for conferences, proceedings and journals.