

Adaptive Census Transform: A novel hardware-oriented stereovision algorithm<sup>☆</sup>Stefania Perri, Pasquale Corsonello<sup>\*</sup>, Giuseppe Cocorullo*Department of Electronics Computer Science and Systems, University of Calabria, Rende, Italy*

## ARTICLE INFO

## Article history:

Received 21 March 2012

Accepted 6 October 2012

Available online 17 October 2012

## Keywords:

Stereovision algorithm

VLSI

Disparity Map Calculation

## ABSTRACT

This paper presents a new hardware-oriented approach for the extraction of disparity maps from stereo images. The proposed method is based on the herein named Adaptive Census Transform that exploits adaptive support weights during the image transformation; the adaptively weighted sum of SADs is then used as the dissimilarity metric. Quality tests show that the proposed method reaches significantly better accuracy than alternative hardware-oriented approaches. To demonstrate the practical hardware feasibility, a specific architecture has been designed and its implementation has been carried out using a single FPGA chip. Such a VLSI implementation allows a frame rate up to 68 fps to be reached for  $640 \times 480$  stereo images, using just 80,000 slices and 32 RAM blocks of a Virtex6 chip.

© 2012 Elsevier Inc. All rights reserved.

## 1. Introduction

Stereovision is a widespread sensing technique able to calculate depths of the objects in an observed scene by capturing and processing 2D images of the scene. As discussed by Nalpantidis and Gasteratos [1,2], Munoz-Salinas et al. [3], Yu and Xu [4], the ability to reconstruct 3D information is crucial for a large variety of applications, such as robot navigation, surveillance, obstacle detection, autonomous vehicles, and many others.

A stereo vision system generally captures different perspective images and computes the relative displacement (usually called the disparity) between corresponding points in the two acquired images. By using the triangulation theory [5], the obtained disparity map leads to the depth information.

A complete survey and taxonomy of previously published approaches to compute disparity maps are furnished by Scharstein and Szeliski [6]. There, so called global and local methods are deeply analyzed. Global methods are often formulated in an energy-minimization framework, thus the objective is to find a disparity function that minimizes a global energy. Such methods are appreciated for their high quality, even though they are achieved at the expense of high computational complexity. Hardware implementations of global methods presented by Yu et al. [7], Wang et al. [8], McCullagh [9] and Jensen et al. [10], require special platforms based on Graphic Processing Units (GPUs). Thus, they appear to be unsuitable for the realization of low-cost stand-alone stereo vision systems. The semi-global methods recently proposed by Banz's and Haller's research groups [11,12] renounce part of the accuracy and

become more affordable for direct hardware implementations based on FPGA devices and/or Application Specific Integrated Circuits (ASICs).

In contrast to global methods, local approaches choose the disparity associated with a minimum cost function at each pixel, performing a “winner-takes-all” optimization. They are undoubtedly the most appropriate to design cheap and fast embedded hardware systems able to satisfy the real-time requirements of most stereo vision applications. Local methods generally locate corresponding pixels in the stereo images by comparing windows of pixels, usually named support or aggregation windows, in terms of a dissimilarity metric.

Several previous works have discussed the design of accurate local stereo vision algorithms and architectures. Nalpantidis et al. [13] presented a thorough review of stereo matching methods, especially those oriented to real-time hardware implementations. We agree with the Authors' claim that local methods could be better benefited by the use of purposely designed parallel structures rather than global ones. However, there are still plenty of options in the design space to reach more favourable tradeoffs.

In 2003, one of the first attempts at the ASIC realization of a complete stereo vision system was presented by Kuhn's research group [14]. Such an implementation simultaneously applies the Sum of Square Differences and the Census Transform (CT). The prototype achieves a frame rate of 50 fps for  $256 \times 192$  greyscale stereo images.

A 3D system based on the DeepSea processor that implements the Census algorithm and allows a frame rate of 200 fps to be reached with  $512 \times 480$  stereo images input was proposed by Woodfill et al. [15] 1 year later.

In recent years the interest in this topic has significantly increased, also because of the availability of more powerful hardware

<sup>☆</sup> This paper has been recommended for acceptance by Shahriar Negahdaripour.<sup>\*</sup> Corresponding author.E-mail address: [p.corsonello@unical.it](mailto:p.corsonello@unical.it) (P. Corsonello).

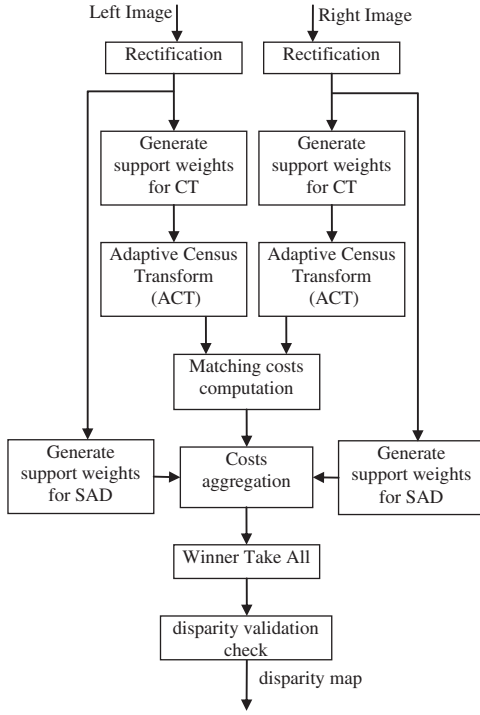


Fig. 1. The computational flow of the proposed algorithm.

platforms. Georgoulas's group presented a complete FPGA-based architecture designed for real-time disparity map computation in [16]. Their design processes up to 275 frames per second for  $640 \times 480$  image resolution and makes use of Cellular Automata filters for eliminating false correspondences from the disparity map. Chen and Jia [17] demonstrated a miniature stereo vision machine based on a parallel reconfigurable architecture that implements the adaptive aggregation based cooperative algorithm. Such a system is able to process up to  $30\,320 \times 240$  disparity maps per second running at 120 MHz clock frequency.

The fast stereo matching algorithm, proposed in 2010 by Humenberger et al. [18] for real-time systems, exploits a sparse Census mask and reaches high accuracy and a considerable speed performance especially if implemented on GPUs.

Ambrosio and Kubinger [19] proposed a novel algorithm that merges gradient-based Census Transform and the Sum of Absolute Differences method. A prototype was constructed using Altera Stratix FPGAs and it reaches a frame rate up to 60 fps for  $750 \times 400$  stereo image resolution.

An architecture implementing the Mini-Census adaptive support weight algorithm has been proposed by Chang et al. [20]. Its ASIC prototype was realized by using the UMC 90 nm technology process and it is able to achieve a frame rate up to 42 fps for  $325 \times 288$  images.

Jin's research group [21] demonstrated a complete stereo vision system based on the Census transform and able to reach 230 fps for  $640 \times 480$  stereo images, when made using a Xilinx Virtex-4 FPGA chip.

Starting from the observation that classical matching measures based on intensity similarity produce imprecise results, De-Maeztu et al. [22] designed an alternative measure that relies on the gradient fields of the neighbourhood of a pixel instead of its intensity value. Theoretical analysis is provided, but they do not present hardware implementations.

This paper proposes a novel hardware-oriented local algorithm for computing accurate disparity maps and shows how an adaptive

**Adaptive Census Transform**

Input: the  $N \times M$  image  $I$ ,  $W_T$ ,  $\gamma_c$

Output: the  $N \times M$  transformed image  $WCV$

```

s_T = (W_T - 1) / 2;
for x = 0 to M - 1
  for y = 0 to N - 1
    k = 0;
    W = ComputeWeights(x, y, s_T);
    for i = -s_T to s_T
      for j = -s_T to s_T
        if I(x, y) > I(x + i, y + j)
          CT = 1;
        else CT = -1;
        WCV(x, y, k) = W(i, j) * CT;
        k = k + 1;
      end;
    end;
  end;
end;
end;

```

(a)

**ComputeWeights(I, x, y, s)**

Input: the image  $I$

Output: the weights  $W$

```

for i = x - s to x + s
  for j = y - s to y + s
    diff = floor(abs(I(x, y) - I(i, j)) / \gamma_c);
    if diff == 0 then
      W(i, j) = 64;
    elseif diff == 1
      W(i, j) = 48;
    elseif (diff >= 2) && (diff <= 3)
      W(i, j) = 32;
    elseif (diff >= 4) && (diff <= 7)
      W(i, j) = 16;
    elseif (diff >= 8) && (diff < 12)
      W(i, j) = 8;
    elseif (diff >= 12) && (diff < 16)
      W(i, j) = 4;
    elseif (diff >= 16) && (diff < 20)
      W(i, j) = 2;
    elseif (diff >= 20) && (diff < 24)
      W(i, j) = 1;
    else W(i, j) = 0;
  end;
end;
end;

```

(b)

Fig. 2. The pseudo-codes related to: (a) the ACT; (b) the approximated exponential weights generator.

Table 1  
Some approximation functions.

	Approx1	Approx2	Approx3	Approx4	Approx5
$p_0$	64	64	64	64	64
$p_1$	48	48	32	32	32
$p_2$	40	32	32	32	32
$p_3$	32	32	32	32	32
$p_4$	24	16	16	24	16
$p_5$	20	16	16	16	16
$p_6$	16	16	16	16	16
$p_7$	12	16	16	16	12
Quantization levels	13	9	8	9	9

approach can be adopted in conjunction with the Census Transform, thus leading to the herein named Adaptive Census Transform.

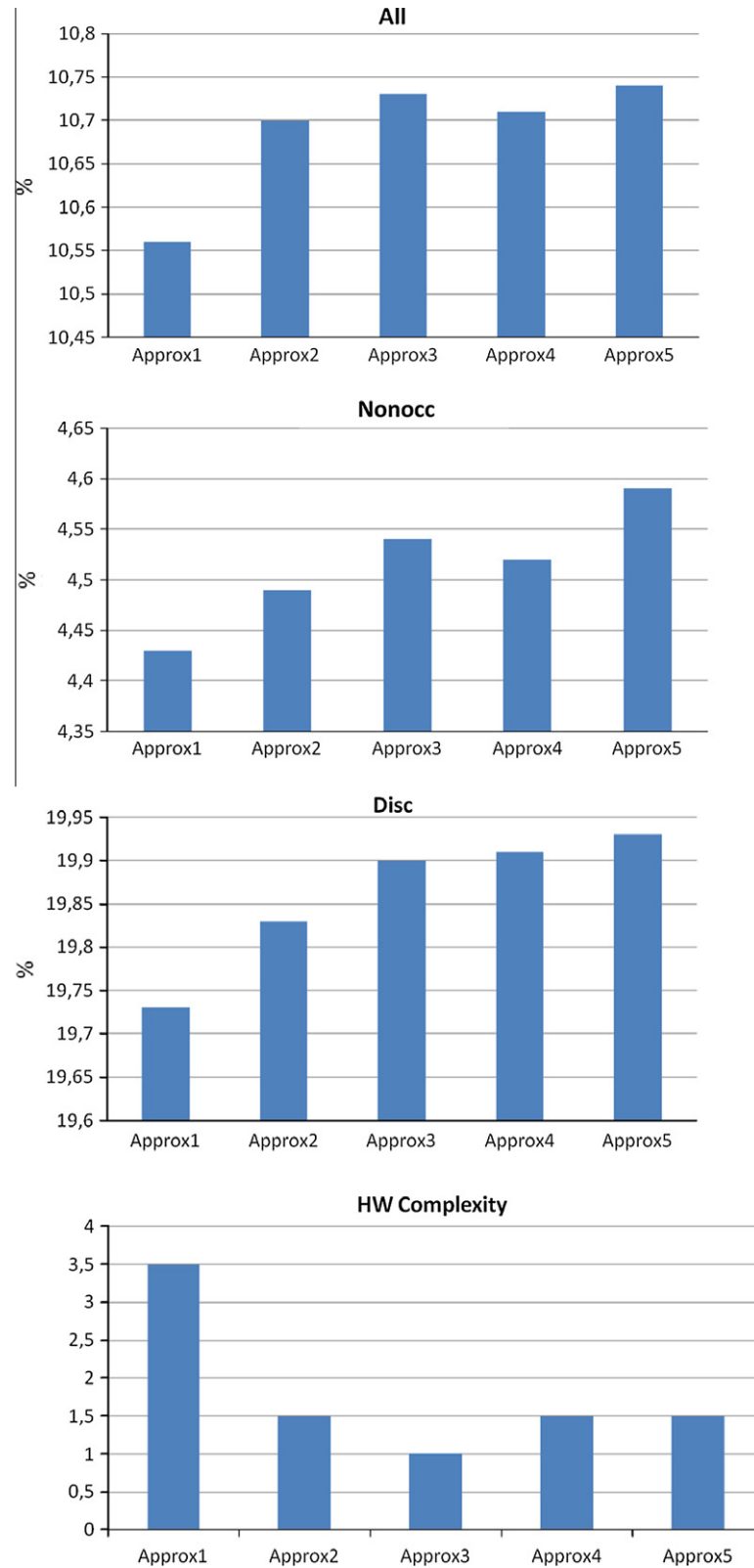


Fig. 3. Average error rates and normalized hardware complexity of analyzed approximation functions.

Quality measurements made referring to typical Middlebury benchmark image sets [23] demonstrate that, when compared to the methods described by Ambrosch and Kubinger [19], Chang et al. [20], and Jin et al. [21], the proposed algorithm can achieve average error rates up to 50%, 10% and 28% lower, respectively.

To demonstrate the feasibility, a hardware architecture of an entire stereo vision system based on the proposed algorithm and able to process VGA greyscale stereo images is also presented. It was created within a Xilinx Virtex6 XC6VLX760 FPGA device considering several sizes of support windows and it achieves a frame rate up to 68 fps.

**Disparity computation**

Input: the transformed reference image WCVR,  
the transformed candidate image WCVL,  $W$ ,  $W_T$ ,  $\gamma_c$

Output: the disparity map disp

```

s=(W-1)/2;
for xr=0 to M-1
  for y=0 to N-1
    for i=-s to s
      for j=-s to s
        for k=0 to  $W_T^2-1$ 
          RefWin(i,j,k)=WCVR(xr+i,y+j,k);
        end
      end;
    end;
    WR=ComputeWeights(xr,y,s);
    for d=MinD to Maxd
      xl=xr+d;
      for i=-s to s
        for j=-s to s
          for k=0 to  $W_T^2-1$ 
            CandWin(i,j,k)=WCVL(xl+i,y+j,k);
          end
        end;
      end;
      WL=ComputeWeights(xl,y,s);
      Num=0;
      Denom=0;
      for i=-s to s
        for j=-s to s
          MC=0;
          for k=0 to  $W_T^2-1$ 
            MC=abs(RefWin(i,j,k)-CandWin(i,j,k))+MC;
          end;
          Num=Num+MC*WL(i,j)*WR(i,j);
          Denom=Denom+ WL(i,j)*WR(i,j);
        end;
      end;
      Diss=floor(Num/Denom);
      if Diss is the first computed OR Diss<minDiss
        minDiss=Diss;
        disp(xr,y)=d;
      end;
    end;
  end;
end;

```

Eq. (10)

Eq. (8)

Fig. 4. The pseudo-code related to the disparity computation.

The remainder of the paper is organized as follows: Section 2 presents common background information about the stereo vision process and briefly describes the state-of-the-art local algorithms chosen as the counterparts; in Section 3, the novel algorithm here proposed is detailed and its accuracy is discussed; a possible hardware implementation is then presented and characterized in Section 4; finally, Section 5 draws conclusions.

## 2. Background

In order to compute disparity maps, stereo images acquired by two distinct cameras placed at a distance  $b$ , called baseline, from each other, are preventively rectified. Details on the computational steps required by the rectification can be found in Faugeras' book [5]. Then, the matching problem has to be solved: for each pixel in the right image its corresponding (matching) point in the left image must be found, or vice versa.

As thoroughly described by Scharstein and Szeliski [6], area-based local methods compare in terms of some dissimilarity metrics (such as Sum of Absolute Differences, Cross Correlation, Sum of Squared Differences, and Hamming distance) a  $W \times W$  reference aggregation window centred at the pixel  $P_{(xr,y)}$  in the right image to  $N_c$   $W \times W$  candidate aggregation windows centred at pixels

$Q_{(xl,y)}$  in the left image, where  $xl = xr + d$ , with  $d$  varying between the minimum value  $d_{min}$  and the maximum value  $d_{max}$ , and the disparity range  $N_c = d_{max} - d_{min} + 1$ . The candidate window which is the least dissimilar from the reference window is called the matching candidate window and its central pixel is the matching point that gives the disparity value  $\partial_{(xr,y)} = xl - xr$ . The disparity values calculated for each reference pixel are collected in the so-called disparity map, which is further processed to improve its overall quality through the validation check that can be executed as reported in their papers by Fua [24] and by Di Stefano's group [25,26].

In 2006, Yoon and Kweon [27] proposed an original highly accurate area-based algorithm able to exploit the adaptive support-weight approach. It measures the similarity between reference and candidate pixels taking into account that neighbouring pixels provide different supports according to their similarity and geometric proximity with respect to the central pixel in the aggregation windows. This means that,  $P_{(xr,y)}$  being the central pixel of a  $W \times W$  aggregation window in the right image, with  $W = 2s + 1$  and  $s$  being the radius of the window, the support weight  $w(R,P)$  of the generic pixel  $R_{(xr',y')}$  (with  $xr' = xr - s, \dots, xr + s$  and  $y' = y - s, \dots, y + s$ ) in the same window is computed as given in (1), where  $\gamma_c$  and  $\gamma_p$  are tuning constants,<sup>1</sup> whereas  $\Delta c$  and  $\Delta g$  are the Euclidean colour and spatial distance, respectively, between  $P_{(xr,y)}$  and  $R_{(xr',y')}$ .

$$w(R,P) = e^{-\left(\frac{\Delta c + \Delta g}{\gamma_c + \gamma_p}\right)} \quad (1)$$

$Q_{(xl,y)}$  being the central pixel of a  $W \times W$  aggregation window in the left image, for the neighbouring pixels  $L_{(xl',y')}$  (with  $xl' = xl - s, \dots, xl + s$ ) in the same window, the support weights  $w(L,Q)$  can be computed in the same way.

When the Absolute Difference (AD) is used as the pixel-based matching cost, the dissimilarity  $D(xr,xl,y)$  between the pixels  $P_{(xr,y)}$  within the right image and  $Q_{(xl,y)}$  within the left image, with  $xl = xr + d$ , is computed as given in (2). The disparity  $\partial_{(xr,y)}$  is finally obtained as the displacement for which the minimum dissimilarity value is computed.

$$D(xr,xl,y) = \frac{\sum_{j=-s}^s \sum_{i=-s}^s |R_{(xr+j,y+i)} - L_{(xl+j,y+i)}| \cdot w(R,P) \cdot w(L,Q)}{\sum_{j=-s}^s \sum_{i=-s}^s w(R,P) \cdot w(L,Q)} \quad (2)$$

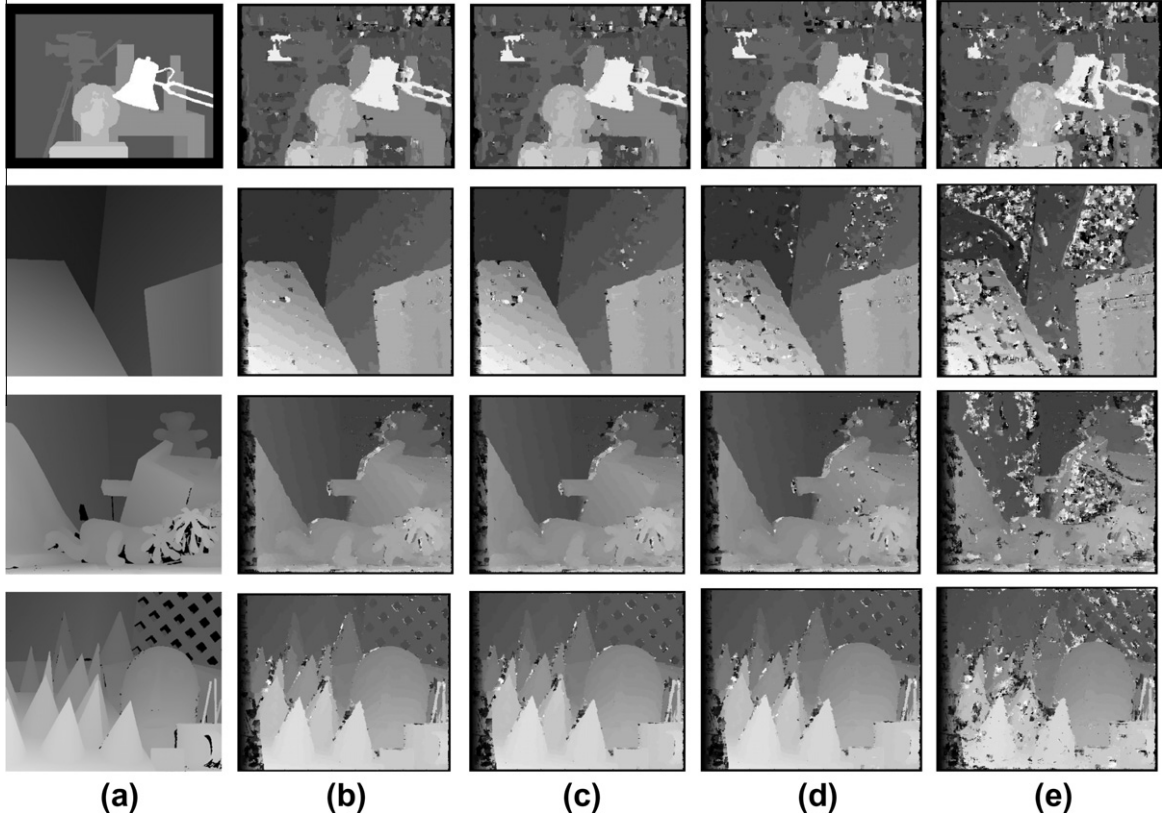
Apart from the SAD, one of the most popular dissimilarity metrics is that proposed by Zabih and Woodfill [28]. It is based on the CT and the Hamming Distance (HD). As the preliminary step, the CT is performed on both the left and right images. For each pixel  $P_{(xr,y)}$ , a  $W_T \times W_T$  support window is considered, with  $W_T = 2s_T + 1$  and  $s_T$  being the radius of the window, to perform the CT thus producing the  $(W_T^2 - 1)$ -bit census vector  $CV_{(xr,y)}$  with the  $k$ th element defined by the piecewise function reported in (3), where  $xr' = xr - s_T, \dots, xr + s_T$  and  $y' = y - s_T, \dots, y + s_T$ .

$$CV_{(xr,y)}(k) = \begin{cases} 0 & | P_{(xr,y)} > R_{(xr',y')} \\ 1 & | P_{(xr,y)} \leq R_{(xr',y')} \end{cases} \quad (3)$$

Analogously, for each candidate pixel  $Q_{(xl,y)}$  the  $(W_T^2 - 1)$ -bit census vector  $CV_{(xl,y)}$  is computed. The pixel-based matching cost of the pixel  $P_{(xr,y)}$  is then evaluated by (4) as the Hamming distance  $HD(xr,xl,y)$  between  $CV_{(xr,y)}$  and  $CV_{(xl,y)}$  (i.e. the number of different elements in the CVs).

$$HD(xr,xl,y) = \sum_{k=0}^{W_T^2-1} (CV_{(xr,y)}(k) \oplus CV_{(xl,y)}(k)) \quad (4)$$

<sup>1</sup> As discussed by Yoon et al. in [27],  $\gamma_c$  is an experimental parameter determined by the perceptual difference between colours, whereas  $\gamma_p$  depends on the radius of the support window.



**Fig. 5.** Disparity maps computed using  $W_T = W = 9$ : (a) ground truth; (b) computed without noise; (c) computed with noise SNR = 50 dB; (d) computed with noise SNR = 40 dB; (e) computed with noise SNR = 30 dB.

In the subsequent computational step,  $W \times W$  aggregation windows are considered to compute the dissimilarity as defined in (5). The displacement leading to the minimum dissimilarity value gives the disparity  $\hat{d}_{(x,y)}$ .

$$D(xr, xl, y) = \sum_{j=-s}^s \sum_{i=-s}^s HD(xr+j, xl+j, y+i) \quad (5)$$

As an alternative to the CT, the algorithm presented by Chang et al. [20] exploits the Mini-Census Transform (MCT) to produce  $(W-1)$ -bit CVs, and the HD as the pixel-based matching cost. In the aggregation step, adaptive support weights, computed using the Manhattan instead of the Euclidean colour distances, are used to calculate dissimilarities as shown in (6).

$$D(xr, xl, y) = \frac{\sum_{j=-s}^s \sum_{i=-s}^s HD(xr+j, xl+j, y+i) \cdot w(R, P) \cdot w(L, Q)}{\sum_{j=-s}^s \sum_{i=-s}^s w(R, P) \cdot w(L, Q)} \quad (6)$$

### 3. The proposed algorithm

Before describing in detail the algorithm proposed here for computing disparity maps, the main innovations introduced with respect to state-of-the-art algorithms are briefly pointed out.

1. Adaptive support weights are exploited not only in the cost aggregation step, but also, for the first time, in the Census Transform computation. The novel approach is here named Adaptive Census Transform (ACT).
2. The adaptive weights exploited in both steps of the algorithm are computed through a simple approximation function that generates only integer values. The latter, exploiting an approach

similar to that used by Chang et al. [20], are chosen not only to achieve a good approximation of the original definition given by Yoon and Kweon [27] for adaptive weights, but also to make the algorithm hardware-friendly.

3. Since the transformation produces vectors of integer numbers, neither the AD nor the HD can be trivially adopted as the pixel-based matching cost that instead is defined as the SAD.
4. Consequently, for the first time, the adaptively weighted sum of SADs is used as the dissimilarity metric in place of the conventional metrics.

#### 3.1. Overview

The block diagram depicted in Fig. 1 summarizes the overall computational flow of the proposed algorithm. Support weights for the CT are generated. Then, the ACT is performed on both the left and right rectified images considering  $W_T \times W_T$  support windows. Then, the pixel-based matching costs are computed and adaptively weighted over  $W \times W$  aggregation windows to calculate dissimilarities between reference and candidate pixels. For each reference pixel,  $N_c$  candidates are examined and the disparity leading to the minimum aggregated matching cost (i.e. the minimum dissimilarity) is selected through a winner-takes-all method. Finally, computed disparities are checked for validation.

#### 3.2. The Adaptive Census Transform

$P_{(xr,y)}$  being the central pixel of a  $W_T \times W_T$  support window in the right image, the ACT generates the  $W_T^2$ -element weighted census vector  $WCV_{(xr,y)}$ , with the  $k$ th element defined by the piecewise function reported in (7), where  $R_{(xr,y')}$  is the generic pixel in the window and  $w(R, P)$  is the support weight computed referring to



the window under transformation. Analogously, applying the ACT to each pixel  $Q_{(xl,y)}$  in the left image, the  $W_T^2$ -element vector  $WCV_{(xl,y)}$  is computed. A possible method to implement Eq. (7) is shown in the pseudo-code of Fig. 2a.

$$WCV_{(xr,y)}(k) = \begin{cases} -w(R,P) & | P_{(xr,y)} > R_{(xr',y')} \\ w(R,P) & | P_{(xr,y)} \leq R_{(xr',y')} \end{cases} \quad (7)$$

Taking into account that the elements of each computed vector are integer numbers, the matching cost  $MC(xr,xl,y)$  of the pixel  $P_{(xr,y)}$  at the displacement  $d$  is defined by (8) as the SAD calculated on the weighted census vectors  $WCV_{(xr,y)}$  and  $WCV_{(xl,y)}$ , with  $xl = xr + d$ .

$$MC(xr,xl,y) = \sum_{k=0}^{W_T^2-1} |WCV_{(xr,y)}(k) - WCV_{(xl,y)}(k)| \quad (8)$$

The adaptive support weights  $w(R,P)$  and  $w(L,Q)$  used to perform the ACT on the right and left images are calculated referring to (1) and neglecting the proximity distance  $\Delta g$ . This simplification leads to a more compact architecture without worsening the quality of the maps. In fact, tests performed on Middlebury benchmark images, demonstrated that, when the windows sizes are between  $5 \times 5$  and  $13 \times 13$ , the accuracy achieved with the novel algorithm does not benefit by the proximity support weights. This confirms the results obtained by Chang's group [20] that verified an accuracy improvement due to the proximity weights only for aggregation windows larger than  $19 \times 19$ .

In the case of greyscale images, the colour distance  $\Delta c$  is the Absolute Difference between the pixels intensities. In order to choose the appropriate  $\gamma_c$  value, the new algorithm was analyzed measuring error rates for several values of  $\gamma_c$ . Results have shown that best average accuracies are obtained for  $\gamma_c = 16$ .

In order to make the novel algorithm accurate and hardware-friendly, several simplifications were introduced to reduce its computational complexity. The first simplification consists in calculating the support weights through an approximation of the exponential function. Eq. (9) describes a set of possible integer approximations of the scaled exponential function  $64 \times e^{-\frac{\Delta c}{\gamma_c}}$ . The scaling factor has been introduced to manipulate the quantization approximation reasonably well.

$$w(R,P) = \begin{cases} p_0 & | \lfloor \Delta c / \gamma_c \rfloor = 0 \\ p_1 & | \lfloor \Delta c / \gamma_c \rfloor = 1 \\ p_2 & | \lfloor \Delta c / \gamma_c \rfloor = 2 \\ p_3 & | \lfloor \Delta c / \gamma_c \rfloor = 3 \\ p_4 & | \lfloor \Delta c / \gamma_c \rfloor = 4 \\ p_5 & | \lfloor \Delta c / \gamma_c \rfloor = 5 \\ p_6 & | \lfloor \Delta c / \gamma_c \rfloor = 6 \\ p_7 & | \lfloor \Delta c / \gamma_c \rfloor = 7 \\ 8 & | 8 \leq \lfloor \Delta c / \gamma_c \rfloor < 12 \\ 4 & | 12 \leq \lfloor \Delta c / \gamma_c \rfloor < 16 \\ 2 & | 16 \leq \lfloor \Delta c / \gamma_c \rfloor < 20 \\ 1 & | 20 \leq \lfloor \Delta c / \gamma_c \rfloor < 24 \\ 0 & | \lfloor \Delta c / \gamma_c \rfloor \geq 24 \end{cases} \quad (9)$$

All the approximations expressed in (9) share the weights ranging between 8 and 0 that approximate the portion of the exponential function with its slighter slope. This choice was justified by preliminary simulations, which have shown that the number of quantization levels used to approximate the exponential function when  $\lfloor \frac{\Delta c}{\gamma_c} \rfloor \geq 8$  does not significantly influence the overall accuracy of the process. On the contrary, weights  $p_0, \dots, p_7$  are involved in modelling the steep portion of the exponential function. Thus, they impact the overall accuracy of the computed disparity maps much more significantly. For these rea-

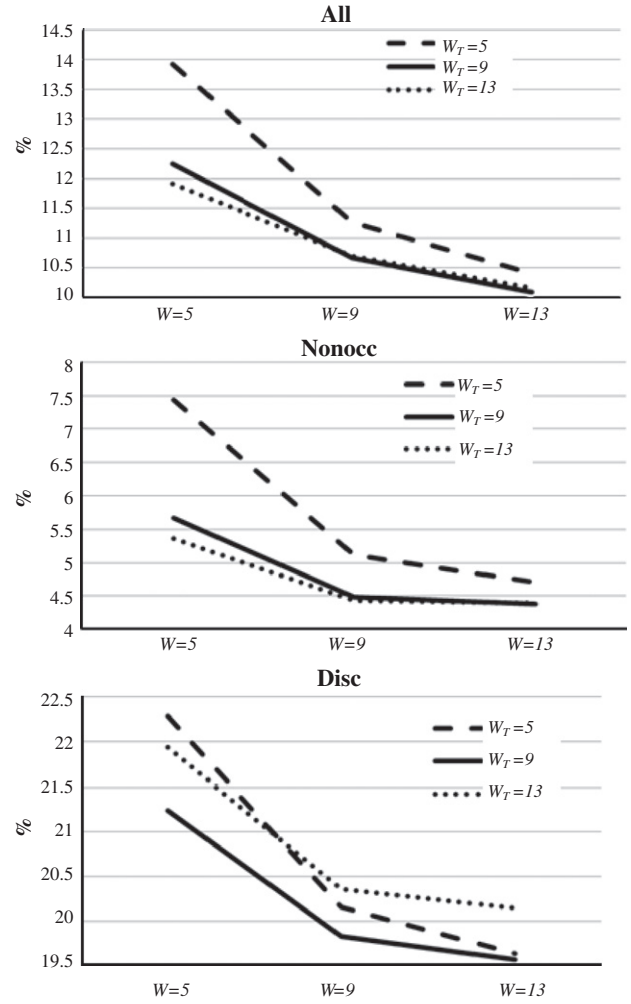


Fig. 6. The average error rates of the novel algorithm versus the windows sizes.

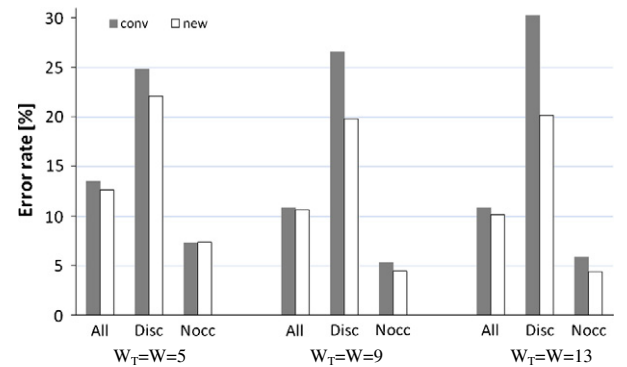


Fig. 7. Error rates achieved by the new algorithm and by the conventional CT-HD matching algorithm.

sons, we evaluated several representative approximation functions measuring their performances in terms of quality and hardware complexity. The chosen functions are reported in Table 1, where the respective number of quantization levels is also reported. Average error rates achieved with  $9 \times 9$  aggregation windows are summarized in Fig. 3, together with the normalized hardware complexities. The error rates represent the percentage of bad disparity values in the nonoccluded regions (Nonocc), in

**Table 2**  
Comparison with existing algorithms.

	$W_T$	$W$	Tsukuba			Venus			Teddy			Cones		
			All	Nonocc	Disc	All	Nonocc	Disc	All	Nonocc	Disc	All	Nonocc	Disc
[18]	16	5	6.25	5.08	19.2	2.42	1.58	14.2	13.8	7.96	20.3	9.54	4.10	12.2
[19]	5	5	23.7	22.7	26.2	15.9	15.1	25.3	20.7	13.9	29	12.7	5.96	15.5
		13	15.1	13.8	25.3	7.33	6.56	25.2	19.4	12.1	29.5	12.4	5.37	15.3
		19	13.4	12.1	28.2	4.86	4.06	25.9	19.7	12.3	31.3	14	6.91	19.2
[20]	5	18	4.18	n.a.	n.a.	3.41	n.a.	n.a.	20.6	n.a.	n.a.	16	n.a.	n.a.
[21]	11	15	11.6	9.79	20.3	5.27	3.59	36.82	21.5	12.5	30.6	17.6	7.34	21.01
[27]	–	33	1.85	1.38	6.9	1.19	0.71	6.13	13.3	7.88	18.6	9.79	3.97	8.26
NEW	5	5	11.8	11.4	39.1	7.02	5.49	15.5	18.9	8.09	21	18	4.79	13.54
		9	8.9	8.5	36.3	3.37	1.91	12.55	16.4	5.98	18.9	16.4	4.06	12.88
		13	7.6	7.2	35.7	2.65	1.33	10.9	15.5	5.73	18.4	15.9	4.54	13.53
	9	5	8.6	8.2	36.2	4.91	3.28	15.55	17.6	6.55	19.4	17.9	4.64	13.8
		9	7.23	6.84	34.9	2.96	1.46	12.6	16.1	5.64	18.71	16.4	4	13.09
		13	6.7	6.28	35.1	2.55	1.2	10.7	15.2	5.43	18.5	15.9	4.63	13.96
	13	5	7.5	7.1	35.9	4.15	2.5	15.8	17.6	6.48	20.2	18.4	5.36	15.84
		9	6.76	6.34	35.2	2.89	1.31	12.87	16.3	5.68	19.3	16.84	4.4	14.06
		13	6.32	5.92	34.9	2.63	1.17	11.38	15.3	5.5	19.2	16.4	5	15.1

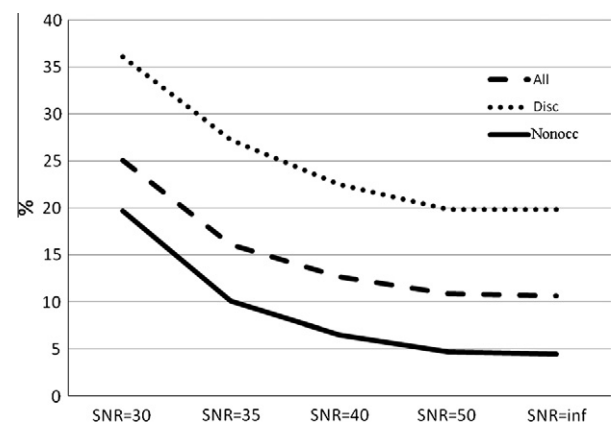
**Table 3**  
Average error rates of the compared algorithms.

	$W_T$	$W$	Average All	Average Nonocc	Average Disc
[18]	16	5	8	4.68	16.48
[19]	5	5	18.25	14.42	24
		13	13.56	9.46	23.83
		19	12.99	8.84	26.15
[20]	5	18	11.1	n.a.	n.a.
[21]	11	15	14	8.3	27.2
[27]	–	33	6.53	3.49	9.97
NEW	5	5	13.93	7.44	22.28
		9	11.27	5.11	20.16
		13	10.4	4.7	19.63
	9	5	12.2	5.7	21.24
		9	10.7	4.49	19.83
		13	10.1	4.39	19.57
	13	5	11.9	5.36	21.94
		9	10.7	4.43	20.36
		13	10.16	4.39	20.14

the entire images the borders except (All), and in the regions near to depth discontinuities (Disc), whereas, the hardware complexity takes into account the number of comparators and adders required in the physical implementations.

From results reported in Fig. 3, it can be seen that the approximation Approx1 is the most accurate, but it also requires 13 quantization levels and use five weights that are not to the power of 2. Owing to this, it shows the highest hardware complexity. Conversely, the approximation Approx3 leads to the lowest hardware complexity, but it achieves the least accuracy.

The approximation Approx2 has been adopted in the novel algorithm since it represents a good trade-off between accuracy and complexity. A possible implementation of the Approx2 function is given in the pseudo-code of Fig. 2b. It must be noted that all weight values, upper bounds and lower bounds used in the chosen approximation are integer numbers. Therefore, complex floating point operations are avoided. This is crucial to make low-cost real-time hardware implementations actually approachable. In fact, as detailed in Section 4, when a Xilinx Virtex6 FPGA device is chosen as the target hardware platform, the adopted approximation function can be easily implemented within only 14 slices. Whereas, an efficient floating point circuit, such as that described



**Fig. 8.** Effects of the Gaussian noise on the average error rates when  $W_T = W = 9$ .

by Dinechin and Pasca [29], that directly implements the exponential function requires at least  $\sim 400$  slices, one DSP48 module and 18 Kbits of dual-port memory.

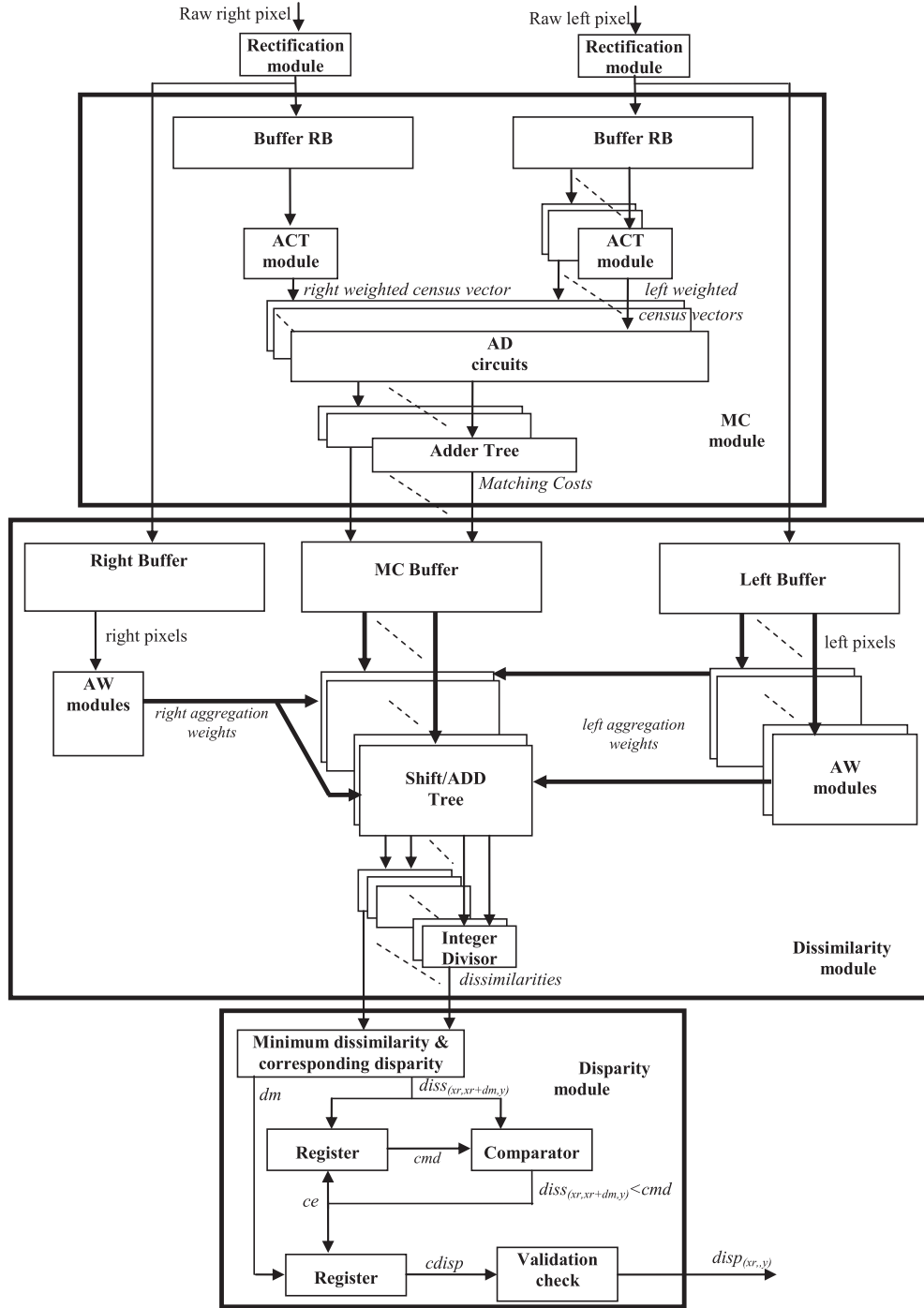


Fig. 9. Demonstration hardware architecture.

### 3.3. Costs aggregation

The final matching cost of the pixel  $P_{(xr,y)}$ , is obtained by aggregating the pixel-based matching costs, above computed by (8), over a  $W \times W$  aggregation window centred at the vector  $WCV_{(xr,y)}$ . In other words, the dissimilarity at the displacement  $d$  is computed as shown in (10), where  $w_A(R,P)$  and  $w_A(L,Q)$  are the adaptive aggregation weights of the pixels  $R_{(xr+j,y+i)}$  and  $L_{(xl+j,y+i)}$ , respectively, within the  $W \times W$  windows centred at  $P_{(xr,y)}$  and  $Q_{(xl,y)}$ , with  $xl = xr + d$ . Also these aggregation weights are computed by exploiting the approximation function above defined in (9).

$$D(xr, xl, y) = \frac{\sum_{j=-s}^s \sum_{i=-s}^s MC(xr+j, xl+j, y+i) \cdot w_A(R,P) \cdot w_A(L,Q)}{\sum_{j=-s}^s \sum_{i=-s}^s w_A(R,P) \cdot w_A(L,Q)} \quad (10)$$

For each pixel  $P_{(xr,y)}$  in the right image, the disparity value  $\partial_{(xr,y)}$  is selected through a winner-takes-all approach as the displacement leading to the minimum dissimilarity value.

The computations required to calculate disparity values are detailed in the pseudo-code reported in Fig. 4, where the portions written to implement Eqs. (8) and (10) are highlighted by continuous and dashed box, respectively.

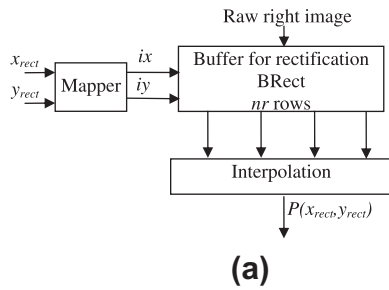


### 3.4. Accuracy of the proposed algorithm

The accuracy of the proposed algorithm was evaluated by means of the well-known benchmarks Tsukuba, Venus, Teddy and Cones [23]. The error rates were measured as the percentages of bad disparity values in the nonoccluded regions (*Nonocc*), in the entire images the borders except (*All*), and in the regions near to depth discontinuities (*Disc*). Disparity values that are not within 1 pixel when compared with the ground truth disparity map are considered incorrect. The accuracy was measured for several windows sizes  $W_T$  and  $W$ . In Fig. 5b disparity maps computed using  $W_T = W = 9$  are shown.

The average values of *All*, *Nonocc* and *Disc* related to the four selected data sets versus the windows sizes are reported in Fig. 6. It can be seen that, the case in which  $W_T = W = 5$  leads to the maximum average error rates. Moreover, results suggest that at a parity of  $W_T$ , using larger  $W$  improves the accuracy. However, it is also evident that for a value of  $W = 9$  and above, a window size  $W_T$  greater than 9 increases the average *Disc* without significantly decreasing either *All* or *Nonocc*.

The novel algorithm has been preliminary compared to the conventional CT–HD approach. Comparison results are reported in Fig. 7 and clearly demonstrate the superiority of the proposed method, also when small support windows are used. The advantages offered by the novel algorithm increase with the support windows sizes.



**Rectification**

Input: the  $N \times M$  raw image  $I$ ,  
the  $3 \times 3$  camera matrix  $KK$ ,  
the  $3 \times 3$  alignment matrix  $M$ ,  
the  $1 \times 5$  distortion vector  $k$ ;

Output: the  $N \times M$  rectified image

```

for x_rect=0 to M-1
  for y_rect=0 to N-1
    (xx, yy, zz)=MatrixProduct(M,(x_rect,y_rect,1));
    x=xx/zz;
    y=yy/zz;
    r2=x2+y2;
    r4=r22;
    r6=r23;
    a1=2*x*y;
    a2=r2+2*x2;
    a3=r2+2*y2;
    xd=(1+k(1)*r2+k(2)*r4+k(5)*r6)*x+k(3)*a1+k(4)*a2;
    yd=(1+k(1)*r2+k(2)*r4+k(5)*r6)*y+k(3)*a3+k(4)*a1;
    (xraw, yraw)=MatrixProduct(KK,(xd,yd,1));
    ix=floor(xraw);
    iy=floor(yraw);
    P(x_rect,y_rect)=Interpolation(I(ix,iy),I(ix+1,iy),I(ix,iy+1),I(ix+1,iy+1));
  end;
end;

```

(b)

Fig. 10. The rectifier module: (a) the schematic block; (b) the pseudo-code.

The error rates obtained by the proposed algorithm have been also compared with the algorithms presented in [18–21,27], because those papers furnish complete accuracy tests and results allowing an unbiased direct comparison. The algorithms described in [18,27] are included in the comparison even though, as mentioned above, their hardware implementation is neither available nor easy to carry out without sacrificing the overall accuracy reached.

Data reported in Table 2 shows that, for all the examined data sets, the new algorithm achieves a *Nonocc* error rate lower than the competitors demonstrated in [19,21], even when it uses smaller windows sizes. This is very important behaviour of the algorithm, especially when the target is the design of low-cost hardware systems. In some cases, the new algorithm reaches *Nonocc* error rates either lower or at the same level as the approach demonstrated by Humenberger et al. [18]. Moreover, for the stereo images Venus and Teddy, the proposed approach exhibits *All* error rates at least 25% lower than all its counterparts (obviously those described in [18,27] except). Lower values of *Disc* are also reached for Venus, Teddy and Cones. Finally, it can be seen that, for the data set Tsukuba, while the accuracy achieved by the novel algorithm is always lower than that reached by the algorithm proposed by Chang et al. [20], *All* error rates up to 50% and 45% lower than the designs proposed by Ambrosch and Kubinger's [19] and Jin et al. [21] are obtained, respectively.

From data reported in Table 2, it can be observed that, as is widely known, the accuracy reached with a given stereo vision algorithm strictly depends on the input images and it can significantly change with the processed data set. For this reason, as suggested by Ambrosch and Kubinger [19], also the average error rates should be evaluated. Comparison results in terms of average error rates are summarized in Table 3. The latter shows that, for windows size greater than 5, the proposed algorithm assures, on average, better accuracies to be achieved with respect to the approaches proposed in [19–21]. Furthermore, it overcomes the algorithm demonstrated by Humenberger et al. [18] in nonoccluded regions.

To completely characterize the proposed algorithm, the achieved accuracy was evaluated also referring to noisy images obtained by contaminating the original benchmarks with several levels of Gaussian noise. Experiments were done for Signal to Noise Ratio (SNR) ranging from 50 dB to 30 dB. Fig. 5c–e report disparity maps obtained when the SNR is equal to 50 dB, 40 dB and 30 dB, respectively. The effect of Gaussian noise on the error rates *All*, *Nonocc* and *Disc* is summarized in Fig. 8. The latter shows that the novel algorithm achieves good accuracy also when noisy stereo images are processed.

### 4. Design of a hardware demonstrator

To demonstrate its feasibility, a hardware prototype based on the algorithm proposed was set up here. Two designs were analyzed and characterized: in the first one  $W_T = W = 5$  and  $N_c = 64$  were adopted, whereas in the second design  $W_T = W = 9$  and  $N_c = 60$  are set. Each design processes VGA greyscale images and was organized to fit in a single Xilinx Virtex6 FPGA device. The top level architecture of the designed systems is illustrated in Fig. 9.

Left and right images are firstly rectified by exploiting the interpolation process detailed by Faugeras in [5] and by Lenz and Tsai in [30]. Rectified pixels are then processed by the MC module that computes the matching costs defined in (8). The Adaptive Census Transform is performed on both the left and right rectified images processing  $W_T \times W_T$  support windows through the ACT modules,

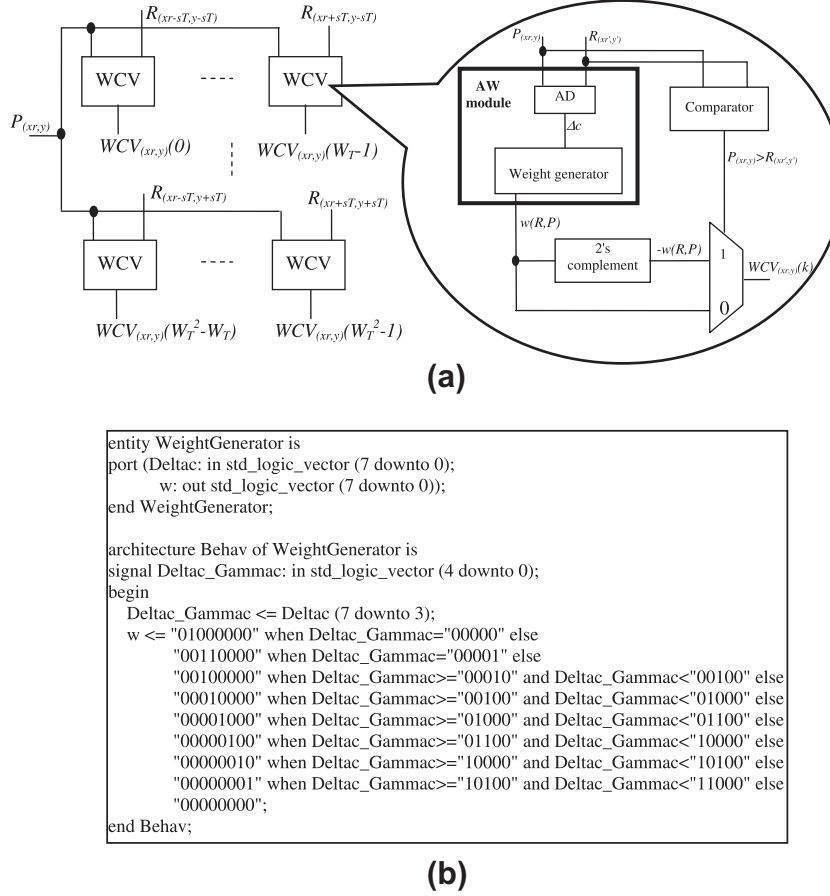


Fig. 11. The ACT module: (a) schematic diagram; (b) VHDL code of the weights generator.

each designed to produce the weighted census vectors as above defined in (7).

The disparity range  $N_c$  is separated into  $p$  partitions, each containing  $dl = N_c/p$  disparity levels. In its generic computational round the MC module processes one partition (i.e.  $dl$  disparity levels) and produces  $dl$  matching costs. The two RB buffers are used to guarantee that all the left and right rectified pixels required at any time to compute the generic Adaptive Census Transform are certainly locally available.

Weighted census vectors are then inputted to the AD circuits.  $(W_T^2 - 1)$  AD circuits<sup>2</sup> compute their element-by-element Absolute Differences. Finally, the matching costs are produced by the Adder Trees.

The next computational step is performed by the dissimilarity module that adaptively aggregates the matching costs over  $W \times W$  aggregation windows. Also this module is provided with appropriate buffer structures to ensure that all the pixels and matching costs required at any time to compute a novel dissimilarity value are locally stored. In its generic computational round the dissimilarity module produces  $dl$  dissimilarity values.

The aggregation weights obtained for the reference pixel  $P_{(x,y)}$  and the  $dl$  candidate pixels  $Q_{(x+d1,y)}, \dots, Q_{(x+dl,y)}$  at the disparity  $d1, \dots, dl$  considered in the current computational round are inputted to the Shift/ADD Tree modules that also receive the  $dl W \times W$  aggregation windows of matching costs corresponding to the candidates under examination. Each Shift/ADD Tree produces the

numerator and denominator of Eq. (10). The  $dl$  separate dissimilarity values related to the  $dl$  disparity levels examined in the current computational round are finally computed by the integer division circuits.

As the final step, the Disparity module computes the minimum value  $D(x_r, x_r + dm, y)$ , corresponding to the disparity  $dm$ , among the dissimilarities received as input; compares it to the currently stored minimum dissimilarity  $cmd$ , corresponding to the disparity  $cdisp$ , and, if needed, updates both these values. Once all the computational rounds required to examine the entire disparity range have been performed, the disparity corresponding to the minimum dissimilarity value is validated through the Validation check module. In order to limit the cost of the overall system, the Uniqueness Check method proposed by Di Stefano et al. [25,26] was adopted and implemented as shown by Zicari et al. [31].

In Fig. 10a, the rectifier module is illustrated. It implements an interpolation process: the intensity value of the generic pixel located at the coordinates  $(x_{rect}, y_{rect})$  within a rectified image is computed by interpolating four pixels located at the coordinates  $(ix, iy)$ ,  $(ix+1, iy)$ ,  $(ix, iy+1)$  and  $(ix+1, iy+1)$  within the original image, also named the raw image. The *mapper* circuit in Fig. 10a maps the rectified coordinates  $(x_{rect}, y_{rect})$  into the raw coordinates  $(ix, iy)$ , which are computed following the computational steps reported in the pseudo-code of Fig. 10b. The camera matrix  $KK$ , the alignment matrix  $M$  and the distortion vector  $K$  are obtained by the preliminary calibration step performed, as explained by Lenz and Tsai [30], through the Matlab calibration toolbox [32].

In order to guarantee a pipelined computational flow without introducing pauses,  $nr$  rows of the raw image are locally stored through the buffer BRect, with  $nr$  chosen taking into account the

<sup>2</sup> Note that the elements of  $WCV_{(x,y)}$  and  $WCV_{(x,y)}$  related to the central pixel of the elaborated support windows are both equal to 64. Therefore, their difference will always be equal to 0.

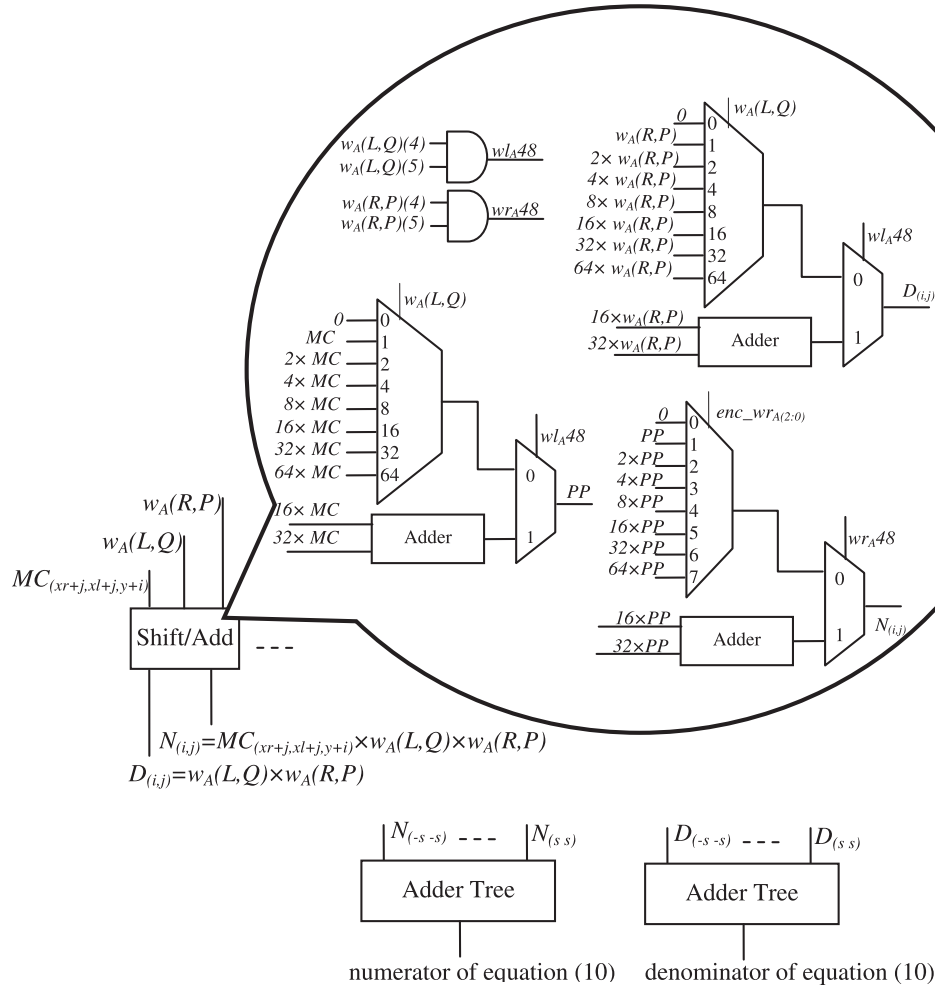


Fig. 12. The Shift/ADD tree module.

**Table 4**  
Speed performances and resources requirements.

	$W_T$	$W$	Image size	$N_c$	Platform	fps	Resources
[19]	5	5	$750 \times 400$	60	Stratix EP1S60	60	38,944 LE 557,056 bits Mem.
[20]	5	18	$352 \times 288$	64	UMC 90 nm	42	562,642 gates 21.3 KB Mem.
[21]	11	15	$640 \times 480$	64	Virtex4 XC4VLX200	230	51,191 slices 12 DSP48 322 RAMBs
NEW	5	5	$640 \times 480$	64	Virtex6 XC6VLX760	68	80,270 slices 112 DSP48 32 RAMBs
NEW	9	9	$640 \times 480$	60	Virtex6 XC6VLX760	45	102,439 slices 112 DSP48 32 RAMBs

lens distortion effects introduced by the stereo camera. In particular,  $nr$  depends on the minimum ( $n1$ ) and maximum ( $n2$ ) displacement between the row indexes  $y_{rect}$  and  $iy$ . In the experiments two kinds of cameras were used: the Bumblebee2 stereo camera [33] and a special purpose camera set up coupling two Omnivision OV9655 sensors [34]. In both cases, the calibration process showed that  $n1 \geq -50$  and  $n2 \leq 49$ , therefore  $nr = 100$  was chosen.

In Fig. 11a, the detailed architecture of the ACT module is depicted. The internal AW module computes the colour distance  $\Delta c$  of the generic pixel  $R$  within a window from the central pixel  $P$ . The corresponding adaptive weight is determined by the weight generator circuit that implements Eq. (9) as shown in the VHDL code reported in Fig. 11b. Finally, the generic element  $WCV_{(xr,y)}(k)$  of the weighted census vectors is outputted.

Implementation details related to the Shift/ADD Tree are furnished with the schematic diagram illustrated in Fig. 12. The architecture of Fig. 9 was synthesized and laid out through the Xilinx ISE

Design Suite 12.2. Several types of optimization could have been applied in the hardware design to improve speed performances and/or space requirements. However, this aspect is outside the scope of this work.

In Table 4, the characteristics of the implemented designs are reported, for  $W_T = W = 5$  and  $W_T = W = 9$ . It can be easily seen that, when smaller aggregation windows sizes are used, less hardware resources are required and higher parallelism level can be exploited, thus increasing the achieved frame rate (fps). Table 4 also shows characteristics of the hardware implementations proposed by Ambrosh and Kubinger [19], Chang et al. [20] and Jin et al. [21], which use disparity ranges equal to 60, 64 and 64, respectively. Furthermore, in Fig. 13, the average error rates are plotted versus the number of disparity evaluations computed per second (DE/s), for all the compared designs. Error rates were obtained by means of purposely designed MATLAB codes reproducing all quantization effects caused by the hardware implementation. From this

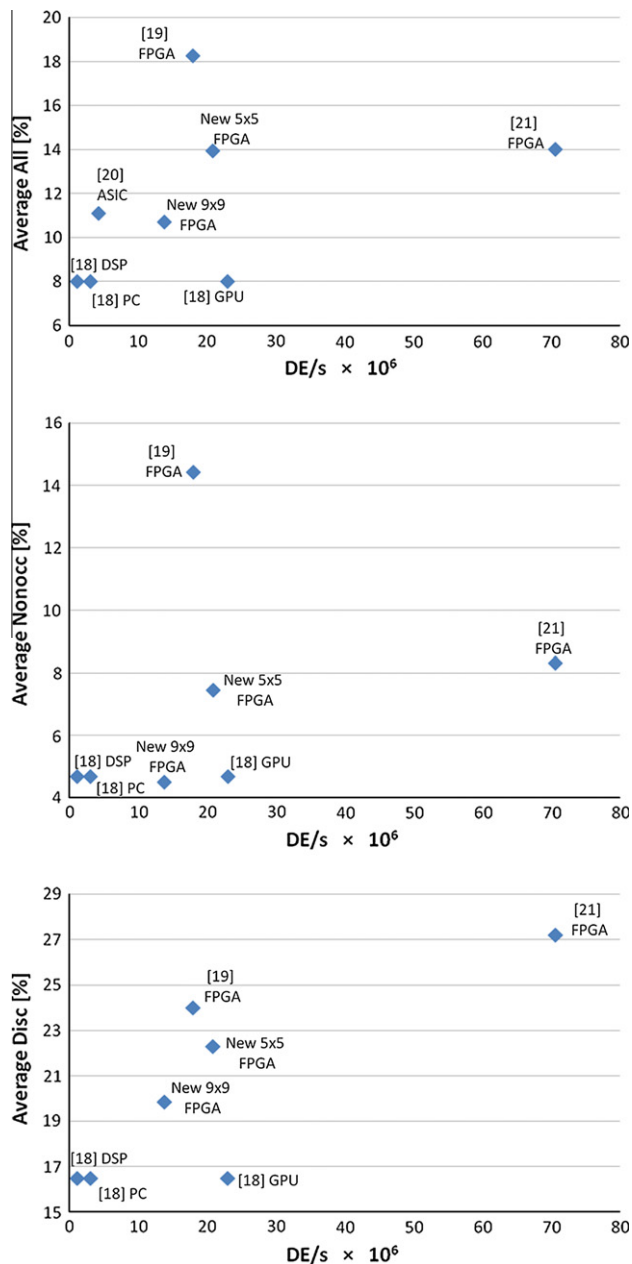


Fig. 13. Average error rates versus the number of dissimilarity per second (NDS).

comparison, it can be seen that the FPGA architecture demonstrated by Jin's research group [21] is the fastest one. The system implemented by Humenberger et al. [18] reaches the lowest *All* and *Disc* error percentages and a considerable speed performance with  $N_c = 50$ . However, it is implemented by means of a NVIDIA GeForce GPU. The architecture designed by Chang and colleagues [20] is significantly slower than competitors, even though it is created using a 90 nm process technology ASIC. The novel system proposed here represents a good trade-off between accuracy and speed. It allows error rates always lower than other FPGA-based solution with a number of dissimilarities per second not much different from that of the GPU-based implementation to be reached.

## 5. Conclusions

A new hardware-oriented approach for the computation of disparity maps based on the Adaptive Census Transform has been

presented. The adaptive support weights exploited in both the steps of the algorithm are computed through a simple approximation function that generates only integer values. This allowed the algorithm to be easily hardware implemented.

Quality tests demonstrated that the proposed method reaches significantly better accuracy than alternative hardware-oriented approaches. Also, it has been noted that the new method allows the same accuracy level as competitors to be reached, by using narrower window sizes, thus significantly reducing the hardware complexity.

An FPGA prototype has been created using a Virtex6 XC6VLX760 chip. A maximum frame rate of 68 fps has been reached for  $640 \times 480$  greyscale images.

## Acknowledgments

The Authors wish to thank Dr. Paolo Zicari for his help in the VHDL coding of the architecture. They would also thank anonymous reviewers for their stimulating observations and suggestions.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.cviu.2012.10.003>.

## References

- [1] L. Nalpantidis, A. Gasteratos, Stereo vision for robotic applications in the presence of non-ideal lighting, *Image Vis. Comput.* 28 (2010) 940–951.
- [2] L. Nalpantidis, A. Gasteratos, Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence, *Robot. Auton. Syst.* 58 (2010) 457–464.
- [3] R. Munoz-Salinas, Eugenio Aguirre, M. Garcia-Silvente, People detection and tracking using stereo vision and color, *Image Vis. Comput.* 25 (2007) 995–1007.
- [4] W. Yu, B. Xu, A portable stereo vision system for whole body surface imaging, *Image Vis. Comput.* 28 (2010) 605–613.
- [5] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.
- [6] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Comput. Vision* 47 (2002) 7–42.
- [7] W. Yu, T. Chen, F. Franchetti, J.C. Hoe, High performance stereo vision designed for massively data parallel platforms, *IEEE Trans. Circuits Syst. Video Technol.* 20 (11) (2010) 1509–1519.
- [8] L. Wang, M. Liao, M. Gong, R. Yang, D. Nister, High-quality real-time stereo using adaptive cost aggregation and dynamic programming, in: *International Symposium on 3D Data Processing, Visualization and Transmission*, Chapel Hill, North Carolina, USA, 2006, pp. 798–805.
- [9] B. McCullagh, Real-time disparity map computation using the cell broadband engine, *J Real-Time Image Proc.* (2010), <http://dx.doi.org/10.1007/s11554-010-0155-8>.
- [10] L.B.W. Jensen, A. Kjaer-Nielsen, K. Pauwels, J.B. Jessen, M. Van Hulle, N. Kruger, A two-level real-time vision machine combining coarse- and fine-grained parallelism, *J Real-Time Image Proc.* 5 (2010) 291–304.
- [11] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, P. Pirsch, Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGA-implementation, in: *International Conference on Embedded Computer Systems*, Samos, Greece, 2010, pp. 93–101.
- [12] I. Haller, C. Pantilie, F. Oniga, S. Nedevschi, Real-time semi-global dense stereo solution with improved sub-pixel accuracy, in: *International Symposium on Intelligent Vehicles*, San Diego, California, 2010, pp. 369–376.
- [13] L. Nalpantidis, G.Ch. Sirakoulis, A. Gasteratos, Review of stereo vision algorithms: from software to hardware, *Int. J. Optomechatronics* 2 (2008) 435–462.
- [14] M. Kuhn, S. Moser, O. Isler, F.K. Gurkaynak, A. Burg, N. Felber, H. Kaeslin, W. Fichtner, Efficient ASIC implementation of real-time depth mapping stereo vision systems, in: *IEEE 43th Midwest Symposium on Circuits and Systems*, Cairo, Egypt, no. 3, 2003, pp. 1478–1481.
- [15] J.I. Woodfill, G. Gordon, R. Buck, Tyzx deepsea high speed stereo vision system, in: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Washington D.C., USA, no. 3, 2004, pp. 41.
- [16] C. Georgoulas, L. Kotoulas, G.C. Sirakoulis, I. Andreadis, A. Gasteratos, Real-time disparity map computation module, *Microprocess. Microsyst.* 32 (2008) 159–170.
- [17] L. Chen, Y. Jia, A parallel reconfigurable architecture for real-time stereo vision, in: *IEEE International Conference on Embedded Software and Systems*, no. 1, 2009, pp. 32–39.

- [18] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, M. Vincze, A fast stereo matching algorithm suitable for embedded real-time systems, *Comput. Vis. Image Underst.* 114 (2010) 1180–1202.
- [19] K. Ambrosch, W. Kubinger, Accurate hardware-based stereo vision, *Comput. Vis. Image Underst.* 114 (2010) 1303–1316.
- [20] N.Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, T.-S. Chang, Algorithm and architecture of disparity estimation with mini-census adaptive support weight, *IEEE Trans. Circuits Syst. Video Technol.* 20 (6) (2010) 792–805.
- [21] S. Jin, J. Cho, X.D. Pham, K.M. Lee, S.K. Park, M. Kim, J.W. Jeon, FPGA design and implementation of a real-time stereo vision system, *IEEE Trans. Circ. Syst. Video Technol.* 20 (1) (2010) 15–26.
- [22] L. De-Maeztu, A. Villanueva, R. Cabeza, Stereo matching using gradient similarity and locally adaptive support-weight, *Pattern Recogn. Lett.* 32 (2010) 1643–1651.
- [23] Middlebury Stereo Vision page. <<http://vision.middlebury.edu/stereo>>.
- [24] P. Fua, Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuity, in: *International joint Conference on Artificial Intelligence*, Sydney, Australia, 1991, pp. 1292–1298.
- [25] L. Di Stefano, M. Marchionni, S. Mattoccia, G. Neri, Dense stereo based on the uniqueness constraint, in: *International Conference on Pattern Recognition*, Quebec, Canada, 2002, pp. 657–661.
- [26] L. Di Stefano, M. Marchionni, S. Mattoccia, G. Neri, A fast area-based stereo matching algorithm, *Image Vis. Comput.* 22 (12) (2004) 983–1005.
- [27] K.-J. Yoon, I. Kweon, Adaptive support-weight approach for correspondence search, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (4) (2006) 650–656.
- [28] R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in: *European Conference on Computer Vision*, Stockholm, Sweden, 1994, pp. 151–158.
- [29] F. De Dinechin, B. Pasca, Floating-point exponential functions for DSP-enable FPGAs, in: *International Conference on Field-Programmable Technology*, Beijing, China, 2010, pp. 110–117.
- [30] R.K. Lenz, R.Y. Tsai, Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology, *IEEE Trans. Pattern Anal. Mach. Intell.* 10 (5) (1988) 713–720.
- [31] P. Zicari, S. Perri, P. Corsonello, G. Cocorullo, Low-cost FPGA stereo vision system for real time disparity maps calculation, *Microprocess. Microsyst.* 36 (4) (2012) 281–288.
- [32] Camera Calibration Toolbox for Matlab. <[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)>.
- [33] Bumblebee Stereo Vision Camera Systems, Datasheet. <<http://www.ptgrey.com/>>.
- [34] <http://www.ovt.com/products>.