Design of Interpolation Functions for Sub-Pixel Accuracy Stereo-Vision Systems

Istvan Haller, Sergiu Nedevschi, *Member, IEEE*

***Abstract*—Traditionally sub-pixel interpolation in stereo-vision systems was designed for the block matching algorithm. During the evaluation of different interpolation strategies a strong correlation was observed between the type of the stereo algorithm and the sub-pixel accuracy of the different solutions. Sub-pixel interpolation should be adapted to each stereo-algorithm to achieve maximum accuracy. In consequence it is more important to propose methodologies for interpolation function generation, than specific function shapes. We propose two such methodologies based on data generated by the stereo algorithms. The first proposal uses a histogram to model the environment and applies histogram equalization to an existing solution adapting it to the data. The second proposal employs synthetic images of a known environment and applies function fitting to the resulted data. The resulting function matches the algorithm and the data as best as possible. An extensive evaluation set is used to validate the findings. Both real and synthetic test cases were employed in different scenarios. The test results are consistent and show significant improvements compared to traditional solutions.**

***Index Terms*—Stereo Vision, Sub-pixel accuracy, Function fitting, Interpolation function**

I. Introduction

SUBPIXEL accuracy is a very important component in stereo vision systems. Using the stereo imaging model the distances measured in the scene are inversely proportional with the pixel disparity in the two images. Sub-pixel level disparity calculation is required to maintain accuracy over a large metric range.

Stereo-vision is the process of extracting depth information from the environment by using 2 or more images from different viewpoints. The 2D projection of a point from space is related to its distance and the imager position. By matching the projections in multiple positions, the depth component can be extracted. The disparity represents the number of pixels by which a given point is displaced between two images. This is the only parameter estimated by the stereo algorithm in terms of depth estimation. Since the disparity is inversely proportional to the metric distance, long range applications of stereo vision require accurate sub-pixel level disparity estimate. To have an idea about the necessary accuracy, let us consider the stereo setup used for this paper and deployed as part of an automotive system. For an object located at 60 meters any disparity error larger than 0.1 pixels will result in a relative distance error greater than 2.5%, well beyond the required specifications. Thus results the necessity to improve the disparity error beyond the capabilities of currents systems.

Traditionally short-baseline stereo systems are considered to lack the long range accuracy necessary for such systems and as a result larger baselines are used. However this brings other issues such as difficult matches, larger occlusions. If we look at how the disparity is transformed into distance, we can observe that there is a linear correspondence between the pixel error and the baseline. Thus if sub-pixel error can be reduced by a significant enough factor, the solution can become competitive with current wide-baseline setups.

Equation 1 shows the relation where Z is the real depth Z_{err} is the depth error, FB is the combination of focal and baseline. The disparity is denoted by D and its error by D_{err}, while k represents the improvement factor.

$$Z + Z_{err} = \frac{FB}{D + D_{err}} = \frac{k \cdot FB}{k \cdot D + k \cdot D_{err}} \tag{1}$$

The original taxonomy proposed by Scharstein and Szeliski [1] classifies stereo algorithms into two main groups, local and global methods. The group of local algorithms uses a finite support region around each point to calculate the disparities. The methods are based around the selected matching metric and usually apply some matching aggregation for smoothing. The window aggregation allows a local smoothing of the disparity values. Larger windows reduce the number of mismatches but also reduce the detection rate at object boundaries. Different aggregation strategies were proposed to handle this issue. The main advantage of local methods is the small [2] computational complexity which allows for real-time implementations [3], [4]. The main disadvantage is that only local information is used at each step. As a result these methods are not able to handle featureless regions or repetitive patterns.

Global algorithms are able to improve the quality of the

This work was supported by CNCSIS – UEFISCSU, projects PNII – IDEI 1522/2008 and PNII - PCCE 100/2010.

I. Haller was a junior researcher for the Image Processing and Pattern Recognition Group at the Technical University of Cluj Napoca. He is now a master student in the Parallel and Distributed Computer Systems program at the VU University Amsterdam (e-mail: hal_ler@yahoo.com).

S. Nedevschi is a professor of the Technical University of Cluj Napoca, also the head of the Image Processing and Pattern Recognition Group (e-mail: Sergiu.Nedevschi@cs.utcluj.ro).

disparity map by enforcing several global constraints in the disparity selection phase. These constraints can include the ordering constraint, the uniqueness constraint and also a smoothness constraint. The resulting stereo-matching problem is modeled as a global energy function which is required to be minimized. For the general 2D case the problem is considered to be NP and different approximations were proposed such as simulated annealing, belief propagation or graph-cut to reduce the running time [1], [5]. Although benchmarks [6] show a significant improvement in the disparity map quality, these methods are not applicable for real-time applications, because the running times are several magnitudes larger than those achieved by local methods, usually in the range tens of seconds even on current hardware [7]. There are also issues when using these methods for driver assistance systems where imaging errors are frequent [8].

In 2005 Hirschmüller proposed the Semi-global matching (SGM) [9] stereo algorithm as an alternative to existing solutions which achieves high quality results while maintaining a reduced execution time. This algorithm cannot be classified using the original taxonomy, thus a new group was created, the group of semi-global algorithms. The method performs multiple 1D energy optimizations on the image. The different 1D paths run at different angles to approximate a 2D optimization. By using multiple paths instead of a single one, it can avoid a streaky behavior common with previous algorithms such as dynamic programming or scan-line optimizations. The energy optimization is based on a correlation-cost and a smoothness constraint. The smoothness is enforced by two components, a small penalty, P1, used for small disparity differences and a larger penalty, P2, used for disparity discontinuities. The larger penalty is adaptive and based on intensity changes to help with object borders. The form of the energy function is:

$$E(D)=\sum_{p}\left(\begin{array}{l} C\left(p,D_p\right)+\sum_{q\in N_p} P1*T\left[\left|D_p-D_q\right|=1\right]+ \\ \sum_{q\in N_p} P2*T\left[\left|D_p-D_q\right|>1\right] \end{array}\right), \quad (2)$$

$$P2\sim\frac{1}{local\_variance}$$

where D is the set of disparities, C is the cost function and N_p is the neighborhood of the point p in all directions. The function T turns the values true and false into 1 and respectively 0. D_p and D_q represent the selected disparities in the points p and q. The Middlebury benchmark [6] shows the results achieved using this. The algorithm consistently achieves results similar to the computationally most expensive methods while clearly differentiating itself from other real-time solutions. Several real-time implementations [10], [11], [12] were also proposed for smaller resolution images. These results show that the method represents a good compromise between speed and accuracy for real-time systems such as automotive applications.

Generally stereo algorithms use a simple parabola interpolation [1], [3], [4]. The method uses the smallest matching value and its neighbors to interpolate a parabola around the three points [13], [14]. The location of the minimum point for this parabola will represent the sub-pixel shift. This solution is mathematically accurate if the matching function can be modeled at least locally as a 2nd degree polynomial. However in 2001 Shimitzu and Okutomi [15] have highlighted that this solution presents a serious issue for the simple window based stereo algorithm, namely the pixel-locking effect where given sub-pixel ranges are favored and large errors can accumulate.

Another solution proposed for sub-pixel interpolation is the use of a linear function [13], [14]. The linearity is motivated for simple stereo algorithms which are based on aggregation. The symmetric V interpolation proposed for the Tyzx DeepSea development system is one such solution [16]. This system shows high accuracy thanks to the synergy between the stereo algorithm and the sub-pixel interpolation function.

This paper describes in detail two new methodologies that can extract new interpolation functions based on the behavior of the stereo setup. This allows the interpolation to be hand-crafted for the setup to make sure that maximum accuracy is achieved.

The first proposal is based on the histogram model of a real scene. Using histogram equalization an existing interpolation model can be adapted to reduce the sub-pixel errors. Although the histogram equalization is difficult in the continuous domain, this solution allows the use of real images.

For the second proposal function fitting is used to estimate the shape of the interpolation function more accurately. This methodology requires extensive knowledge about the scene, which is difficult to obtain in a real setting. But this paper shows that synthetic images work well as a work-around. In the latter case, this methodology should be validated using real images, to make sure that there are no differences in the imaging processes.

An exhaustive battery of tests is used to validate the results. Results are tested both on synthetic and real images with different configurations in terms of relative angle and texture characteristics. Even the parts of the Middlebury benchmark are included to show the behavior on well-known reference images. Evaluation focuses on planar surfaces since the main motivation was to improve consistent sub-pixel errors introduced by the current interpolation functions. In the case of complex shapes the sub-pixel values is affected by multiple sources of errors which may lead to inconsistent results. For a modular design solutions to handle these errors should be decoupled from the interpolation function, the latter based on the model without geometric information. In this case the scene complexity is not relevant for evaluating the interpolation function accuracy.

II. Related Work

A. Fractional Disparities

The idea of using fractional disparities was first proposed by

Shimitzu and Okutomi [15]. They observed that the sub-pixel errors can be cancelled through the use of the cost function of images shifted by 0.5 pixels. The shifted images will have the error function inverted compared to the regular image-pair. Although this solution proved to be quite effective, its main disadvantage is that the stereo matching has to be performed 2 times resulting in a significant waste of computing resources.

Szeliski and Scharstein [21] performed a thorough evaluation of this idea using Fourier analysis and different up-sampling techniques. Their results show that an up-sampling using the sinc interpolator and a factor of 2 can result in significant reduction in errors. Unfortunately the paper evaluates the solution only for a simple window based stereo algorithm.

B. Solutions for Long Range Stereo Accuracy

Gehrig and Franke [22] have also proposed two solutions to improve the accuracy for the Semi-Global stereo vision algorithm. The first solution extends the disparity range through the use of fractional disparities. This method is based on the work presented by Szeliski and Scharstein [21], but for some reason the up-sampling factor was increased to 4. This may be due to the inherent complexity of the stereo algorithm. Using this up-sampling the sub-pixel range covered by each cost matrix step will be reduced to 0.25. The disadvantage of this solution is the significant increase in execution time and memory requirements.

To further improve the accuracy for planar surface, Gehrig and Franke also propose the use of adaptive smoothing. It is based on the local homogeneity of the distance values on local patches. The paper shows that planar surfaces are well reconstructed when multiple iterations are used, but the computational cost is significant for a real-time system. It is also unknown, how the smoothing affects the 3D points for non-planar objects and discontinuities. This is highlighted by the fact that the error percentages increased for some of the scenarios.

III. STEREO SETUP

Modern stereo methods such as the Semi-Global method [9] use multiple non-linear transforms. Describing the complete mathematical model of the sub-pixel interpolation is difficult in this case. Examples of such transformations are the census transform and also global and semi-global optimizations. The distribution of the matching values also varies between the solutions and as such it is important to mention the stereo algorithm for which we propose an interpolation function.

The stereo algorithm selected for this paper is a variation of the basic Semi-Global method [17]. These modifications concern both the running time and the sub-pixel accuracy. The configuration selected for this paper uses only 4 directions for reducing the computational complexity and improving hardware integration. Using only the horizontal and vertical directions the memory access pattern and parallelization pattern can be optimized for the GPU architecture. The original description [9] specifies that the recommended number of directions is at least 8 to achieve quality, but previous tests [17] show that the difference is insignificant for automotive applications. Another test [18] also supports similar results for generic scene, even though the authors' view was different. The system used for this paper is optimized for automotive scenes where the object surfaces are usually tilted around the image axis. Consequently diagonal directions introduce no extra information.

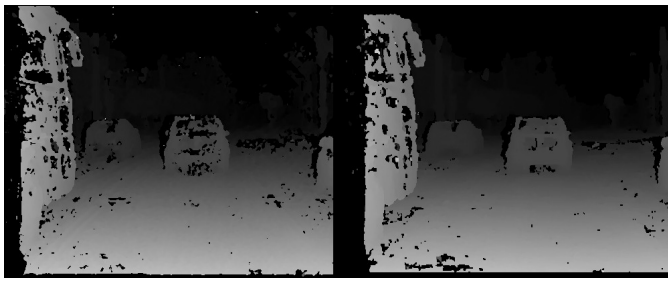

Fig. 1. Intersection scene. Comparison of different solutions, left is SGM+ZSAD, and right is using SGM + Census.

An issue was also observed concerning the sub-pixel accuracy of the original system. The P1 component affects the matching values used in sub-pixel interpolation. The values at the positions -1 and +1 may be shifted with the constant P1. As a result some of the sub-pixel values are corrupted and point scatter is increased. We proposed the elimination of this component from the equation. The new equation is:

$$E(D) = \sum_{p} \left(C\left(p, D_p\right) + \sum_{q \in N_p} P2 * T\left[D_p \neq D_q\right] \right). \tag{3}$$

For the correlation metric the proposed solution uses the Census transform. This metric has the main advantage of being independent of luminosity and contrast differences between cameras. Other papers [19], [20] evaluated the different metrics and the Census transform was consistently one of the best solutions especially in the presence of radiometric errors. These features are important for an automotive system where the precise calibration of cameras is difficult. The original metrics proposed for the Semi-Global method were shown to be not effective in such systems. Another solution [12] proposed uses ZSAD, but in previous tests [17] the Census based solution presented a reduction in disparity errors. Fig. 1 presents a comparison of the two solutions on a typical scenario.

IV. INTERPOLATION FUNCTION THEORY

In this paper we focus on the different interpolation function shapes as a means to improve the sub-pixel accuracy. The shapes have a significant effect on the final distribution of points, and it should match the mathematical model of the matching cost distributions. We propose a common framework to define and compare different shapes.

We use the classic function prototype for sub-pixel interpolation, the same as legacy solutions:

$$d_{Final} = d + f\left(m_{d-1}, m_d, m_{d+1}\right), \tag{4}$$

where d is the integer disparity, $f(m_{d-1},m_d,m_{d+1})$ generates the sub-pixel disparity, and m is the matching cost for the different disparity steps. We believe that the input parameters contain enough information for an accurate interpolation, while preserving simplicity.

But having 3 independent input parameters is too difficult when modeling. By finding correlation between the parameters the dimensionality of the problem can be reduced. The first observation is the invariance of the sub-pixel position to any translation applied on the matching cost values. All 3 values are translated, such that m_d becomes 0. As a result the number of independent variables becomes 2.

$$\begin{aligned} leftDif &= m_{d-1} - m_d \\ rightDif &= m_{d+1} - m_d \end{aligned} \tag{5}$$

Finding the correlation between these variables is more difficult. A proper mathematical model has never been described, thus it was important to work with empirical observations. A synthetic benchmark was used for this purpose. The scene contains a large surface parallel to the imager plane. A non-repetitive pattern is used to reduce stereo uncertainty (Fig. 2). The stereo system is chosen to have similar parameters as a real system with a baseline of 44cm and a focal length of 6mm. The imaging resolution is 512x383.

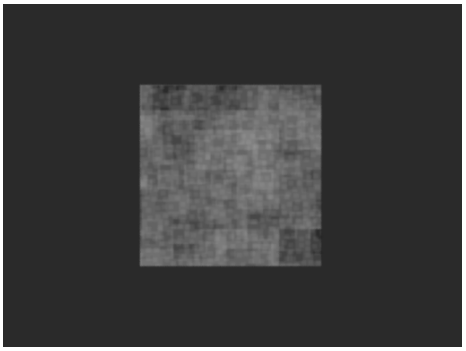

Fig. 2. Example image (right camera, distance is 62.17m)

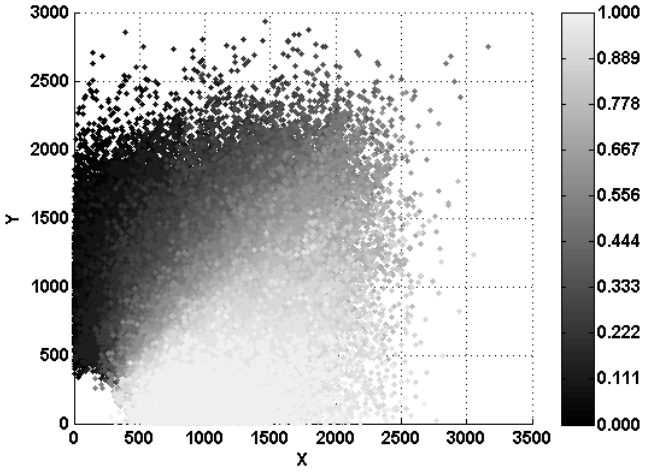

Fig. 3. X-leftDif, Y-rightDif, gray-sub-pixel value scaled 0 to 1.

The position of the plane is set to distances corresponding to disparity values ranging from 3.5 to 4.5 pixels using a step of 0.05.

A careful analysis of the data (Fig. 3) shows a correlation between the polar angle, described by leftDif and rightDif, and the expected sub-pixel value. Since the polar angle is based on the ratio between the two parameters, the latter will be used for the proposed model. Taking into account the symmetricity of the problem, the ratio can also be limited to the range [0, 1] (equations 6 and 7). The final interpolation function (equation 8) maps this ration to the sub-pixel value.

Considering: $leftDif \le rightDif$

$$x = \frac{leftDif}{rightDif} \tag{6}$$

$$d_{Final} = d - 0.5 + \text{interpFunction}(x)$$

Considering: $leftDif > rightDif$

$$x = \frac{rightDif}{leftDif} \tag{7}$$

$$d_{Final} = d + 0.5 - \text{interpFunction}(x)$$

where:

$$\begin{aligned} &\text{- interpFunction} : [0,1] \to [0,0.5] \\ &\text{- interpFunction is monotonic increasing} \\ &\text{- interpFunction}(0) = 0 \\ &\text{- interpFunction}(1) = 0.5 \end{aligned} \tag{8}$$

The proposed model can also be used to describe both traditional interpolation functions. The resulted interpolation functions are simple and straightforward, suggesting that the model is general and suitable for designing new interpolations functions. The following equations use basic transformations to bring the parabola interpolation into the required template.

$$\begin{aligned} f(m_{d-1},m_d,m_{d+1}) &= \frac{m_{d-1} - m_{d+1}}{2*(m_{d-1} - 2*m_d + m_{d+1})} \\ &= \frac{leftDif - rightDif}{2*(leftDif + rightDif)} = \\ &= \begin{cases} -0.5 + \dfrac{leftDif}{leftDif + rightDif}, & if\ leftDif \le rightDif \\ 0.5 - \dfrac{rightDif}{leftDif + rightDif}, & if\ leftDif > rightDif \end{cases} \end{aligned} \tag{9}$$

depending on how the fraction is simplified:

$(0 = leftDif - leftDif), (0 = rightDif - rigthDif)$.

The interpolation function shape:

$$\text{interpFunction}(x) = \frac{x}{x+1} \tag{10}$$

Applying the model to the linear interpolation is even easier, since it is also based on the ratio of matching cost differences.

$$f(m_{d-1},m_d,m_{d+1}) = \begin{cases} -0.5 + \dfrac{1}{2} * \dfrac{leftDif}{rightDif}, & if\ leftDif \le rightDif \\ 0.5 - \dfrac{1}{2} * \dfrac{rightDif}{leftDif}, & if\ leftDif > rightDif \end{cases} \tag{11}$$

The interpolation function shape:

$$\text{interpFunction}(x) = x/2 \tag{12}$$

V. Interpolation Function Based on Data Histogram

A. Histogram, a Known Model for Real Data

The first proposed approach [23] uses real images to extract knowledge about the interpolation functions. The problem with using real images is the lack of detailed ground truth information. The solution is to work on a higher abstraction level then raw pixel data, for example a histogram of sub-pixel values. The latter models a planar surface with a flat histogram shape. This information is available even when other knowledge about the environment is missing. By comparing the resulted histogram to the reference model, problem areas can be highlighted and corrected.

This experiment used a set of real images containing a segment of road surface covered with featureless pavement. A rectangle of interest is applied to consider only road points from the scene. These points are part of a single planar surface and cover a multiple disparity values. As presented previously, the sub-pixel range should be covered homogenously in the resulting histogram bins. Although matching errors may exist, their effect is insignificant from a statistical point of view.

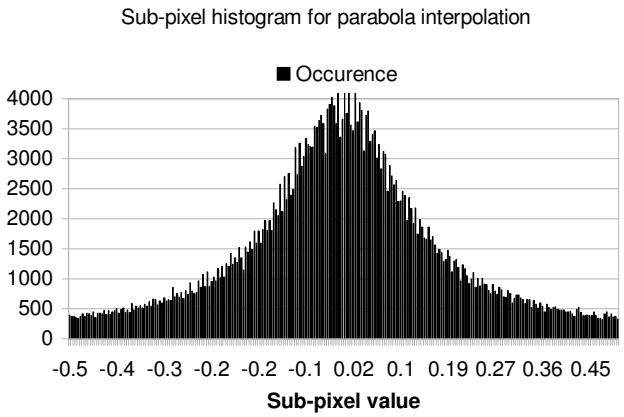

Fig. 4. Histogram of sub-pixel values using parabola interpolation. X axis is the sub-pixel value compared to the closest integer. Y axis is occurrence.

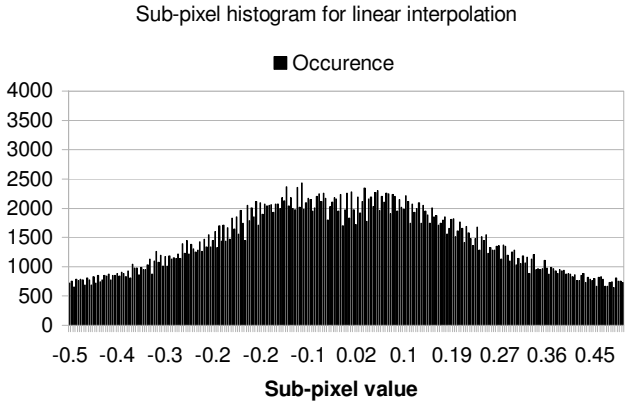

Fig. 5. Histogram of sub-pixel value using linear interpolation. X axis is the sub-pixel value compared to the closest integer. Y axis is occurrence.

Using road textures increases the amount of uncertainty, leading a significant spread in the 3D points. The histogram will be better covered, leading to a smoother shape and helping analysis.

B. Histogram Equalization and Resulting Function

Besides visual feed-back, this model allows a systematic correction through histogram equalization. Although histogram equalization was proposed for discrete values, the mathematical model can also be used for a continuous range.

Suppose p(x) is the probability that the sub-pixel shift is equal with x. This value is the real continuous probability, which is only approximated in the measurements. The interpFunction is used for the equalization.

$$\begin{aligned} &p:[-0.5,0.5]\to[0,1] \\ &p_{Transformed}:[0,0.5]\to[0,1] \\ &p_{Transformed}(x) = p(x-0.5)+p(0.5-x) \\ &cdf(x) = \int_0^x p_{Transformed}(t)dt \\ &\text{interpFunction}_{Corrected} = cdf\left(\text{interpFunction}\right) \end{aligned} \tag{13}$$

The probability function is transformed to take into account the symmetricity of interpFunction. The function *cdf* represents the cumulative distribution function in the continuous domain. The difficulty lies in the estimation of the probability density function, based on the available measurements. After applying the integral operator in the function, any errors will be magnified.

Fig. 4 and Fig. 5 present the occurrences of different sub-pixel shift values for the two legacy solutions. From these figures we try to estimate the shape of the continuous probability function p.

Unfortunately the shape for the parabola interpolation is quite complex making it hard to determine the function *p,* comparatively the linear interpolation histogram shows a linear behavior in each of the symmetric sub-halves. It can be described by the linear function:

$$p_{Transformed}(x) = a*x+b \tag{14}$$

For this paper the model parameters are estimated empirically. This solution works well since the general shape of the interpolation function can also be deduced without knowledge about the parameters. The large amount of noise in the data made it difficult to perform the entire process automatically. Future work may be able to provide a more robust work-flow.

The chosen parameters are $a=1; b=0.5$. Integrating the probability distribution function and composing it with the original linear interpolation functions yields:

$$\text{interpFunction}(x) = \frac{x^2+x}{4}. \tag{15}$$

The histogram resulted with the new function is presented on Fig. 6. The distribution is significantly improved compared to the legacy solutions. This method is the first proposal for an improved sub-pixel interpolation function.

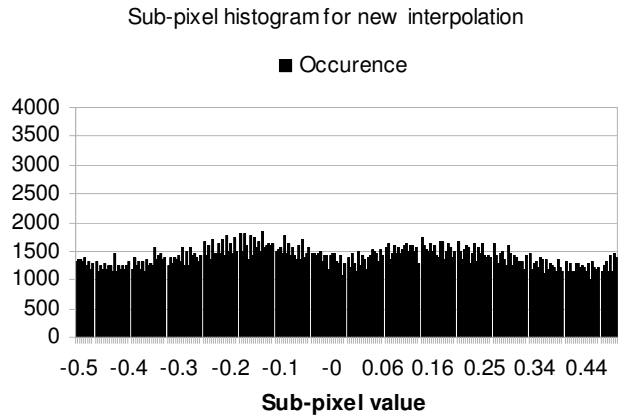

Fig. 6. Histogram of sub-pixel value using proposed interpolation. . X axis is the sub-pixel value compared to the closest integer. Y axis is occurrence.

VI. Interpolation Function Based on Fitting

A. Basic Methodology

The second proposed approach [23] is to use synthetic images to model the sub-pixel interpolation functions. The synthetic images have the advantage of an accurate representation for a predefined scene. The same benchmark is used as in section IV. Each image contains a vertical surface at a distance corresponding to a given sub-pixel location. For each image we log the data used by the sub-pixel interpolation. Because the proposed interpolation model uses the same data for all of the methods, we only need to save the triplet (*leftdif, rightDif, expectedSubpixel)* for each point. Using this data-set we can model the sub-pixel interpolation function through function fitting. This solution allows us to devise an interpolation function that is a perfect match for the extracted data. But we still need to validate if the data distribution is representative of the stereo algorithm in different scenarios. A thorough evaluation of the results is presented in the sections VII and VIII.

B. Function Fitting

As the metric for function fitting we choose the maximum error. Compared to using the sum of errors, this metric reduces the error peaks. For a robust system we consider that it is much more important to consider this worst case error. The fitting method uses non-linear regression to handle different component functions. The components are based on the preliminary analysis of the data [23]. For this paper different polynomial and trigonometric functions were combined, with the final results being generated by the following model:

$$\text{interpFunction}(x) = A*x + B*x^2 + C*x^3 + D*\cos\left(x*\frac{\pi}{2}\right) + E \quad (16)$$

The best fit was achieved when the sinusoidal component represented 99% of the final function. We consider that the polynomial components are too small to take into account because they are within the error margin of the imaging process. The sinusoidal function has the following formula:

$$\text{interpFunction}(x) = 0.5 - \frac{1}{2}*\cos\left(x*\frac{\pi}{2}\right). \quad (17)$$

Fig. 7 compares the shape of the interpolation functions across the input domain. While the parabola is concave and the linear interpolation is straight, the two new functions are both convex. The output of the last function is less then half of the parabola in the entire first half of the input domain, resulting in a significantly different point distribution in the final depth image.

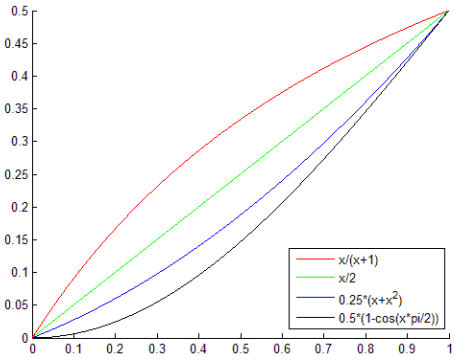

Fig. 7. Plot of interpolation function shapes.

VII. Evaluation Using Synthetic Images

A. Vertical Surfaces

The first test uses the synthetic images generated for function fitting. Although this selection favors the sinusoidal, we use this test to have a baseline before the detailed evaluation. The disparity range corresponds to a metric range from 48 to 62 meters. For measuring the distance of the surface from the camera we use the mean distance of the 3D points. The numerical results are presented in table I.

TABLE I
Errors for Vertical Surfaces

Method	Average (pixel)	Average (rel)	Max (pixel)	Max (rel)
Parabola	0.124	3.10 %	0.215	5.60 %
Linear	0.080	2 %	0.138	3.65 %
Histogram	0.045	1.12 %	0.081	2.17 %
Fitting	0.026	0.64 %	0.053	1.38 %

PIXEL – Error in pixels / REL – Relative distance error
Histogram – Function generated using histogram equalization
Fitting – Function generated using function fitting

The results show that traditional solutions are a poor match to the stereo algorithm and they present significant errors. Both of the proposed solutions are based on the stereo algorithm and the errors are reduced accordingly. The sinusoidal function resulted from the fitting process has the lowest errors by far compared to the other results. These results could be dismissed since the same image sequence is used for fitting and evaluation. Still all of the further tests show the similar results concerning the pixel errors.

B. Surface at Different Angles

For this evaluation we wanted to see the effect of the surface tilt on the error rates. We use the same methodology to generate a synthetic scene containing a surface at 60 meters tilted at 30/45/60 degrees in the YZ coordinate system. The middle of the camera baseline is centered compared to the surface. For evaluation the averages of the Y and Z values are measured along the image rows. As a result we can calculate the error between the measured Z and the expected Z based on the Y value. The results for the three scenes are compared in tables II, III and IV.

TABLE II
ERRORS FOR TILTED SURFACE (30 DEGREES)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.113	2.8 %	0.217	5.17 %
Linear	0.063	1.58 %	0.133	3.13 %
Histogram	0.025	0.64 %	0.087	1.92 %
Fitting	0.011	0.28 %	0.051	1.13 %

TABLE III
ERRORS FOR TILTED SURFACE (45 DEGREES)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.101	2.65 %	0.208	4.65 %
Linear	0.053	1.38 %	0.113	2.61 %
Histogram	0.017	0.46 %	0.047	1.25 %
Fitting	0.012	0.32 %	0.041	1.01 %

TABLE IV
ERRORS FOR TILTED SURFACE (60 DEGREES)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.107	3.05 %	0.180	4.96 %
Linear	0.059	1.68 %	0.103	2.83 %
Histogram	0.022	0.64 %	0.052	1.47 %
Fitting	0.010	0.27 %	0.027	0.77 %

The results for all of the scenarios are consistent and similar with the results found for the vertical surfaces. Looking at the average error we can observe a factor of 2 improvements for the sinusoidal function compared to the other proposal and a factor of 5 compared to the linear interpolation.

C. Horizontal Surface

Besides angled surfaces we also evaluate a horizontal surface. The scene contains a large horizontal surface 2 meters below the level of the cameras. The same texture is used as in the previous tests. For estimating the surface once more we project the points in the 3D metric space. In this case it is hard to estimate the real Z distance for each image row. In consequence we observe the deviation of the Y values from the real height of 2 meters. Again we average the values along the image rows to reduce the spread. Although the differences between the interpolation algorithms are reduced, the order between them remains as presented in table V.

TABLE V
DEVIATION IN Y VALUES FOR HORIZONTAL SURFACE

Method	AVERAGE (ABS)	MAX (ABS)
Parabola	8.05 mm	21.66 mm
Linear	7.09 mm	17.93 mm
Histogram	6.6 mm	16.3 mm
Fitting	6.5 mm	15.7 mm

ABS – Absolute error

D. Vertical Surface with road specific texture

To verify that the results are not specific to the texture, we generate the same scenario, but using road texture taken from the real world. The source of the texture is a tarmac segment of a real image. Compared to the highly detailed pattern used for the previous test, this texture contains very weak features. Road surface was specifically selected because it is one of the scenarios encountered by the stereo system when deployed in an automotive environment. An example image is presented in Fig. 8.

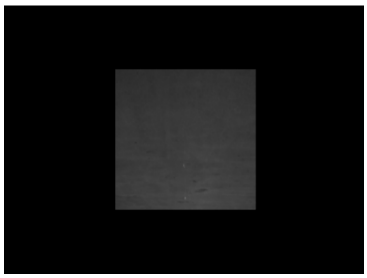

Fig. 8. Vertical surface textured with road segment. Left image.

For the evaluation we used only the range of disparities from 3.5 to 4. Table VI presents the results using the new image-set.

TABLE VI
ERRORS FOR VERTICAL SURFACES (ROAD TEXTURE)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.150	3.79 %	0.264	6.81 %
Linear	0.112	2.82 %	0.192	5.03 %
Histogram	0.081	2.03 %	0.136	3.6 %
Fitting	0.065	1.64 %	0.113	2.73 %

The results show that the increased uncertainty amplifies the erroneous behavior for all of the solutions. Although the effect is different for each solution, the order is unaffected and the newly proposed methods still far better than the traditional ones.

E. Effects of Up-Sampling

We also verified the claims of using up-sampling to improve sub-pixel quality [8], [12], [13]. Again we used the fitting image-set and increased the linear resolution of the images by a factor of 2. For each image the middle was cropped to yield a new image of the original resolution. For this test again we used only the sub-range of disparities from 3.5 to 4. The new error rates are presented in table VII.

TABLE VII
ERRORS FOR VERTICAL SURFACES (UP-SAMPLING)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.061	1.45 %	0.134	3.26 %
Linear	0.045	1.06 %	0.103	2.52%
Histogram	0.031	0.74 %	0.080	1.94 %
Fitting	0.025	0.62 %	0.066	1.67 %

As observed in previous work [8], [12], [13] the oversampling significantly reduces the errors for traditional interpolation methods. A small improvement is also obtained for the histogram based solution. In the case of the sinusoidal the maximum error is increased, but the average error is almost unchanged. It seems that the up-sampling affects this solution negatively. Even in this case the sinusoidal presents the lowest errors, but when using this solution we do not recommend combining it with up-sampling to improve the results.

F. *More Traditional SGM*

The last synthetic test concerns the applicability on a more traditional SGM implementation. Table VIII shows that results when applying all 8 optimization directions. This shows that even though the functions were adapted for a specific stereo framework, they can be reused in other variants. For the best performance it is still recommended to apply the proposed function generation strategies for each algorithm configuration.

The results show that traditional solutions are a poor match to the selected stereo algorithm and they present significant errors. Both of the proposed solutions are based on the stereo algorithm and the errors are reduced accordingly. The sinusoidal function resulted from the fitting process has the lowest errors, especially the average values which is almost 4 times better than even the histogram based solution. Part of this result is due to using the same image sequence for fitting and evaluation. Still all further tests show the same tendency even if the error for the sinusoidal increases slightly.

TABLE VIII
ERRORS FOR VERTICAL SURFACES

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.127	3.16 %	0.220	5.70 %
Linear	0.082	2.05 %	0.141	3.72 %
Histogram	0.046	1.17 %	0.083	2.21 %
Fitting	0.027	0.68 %	0.054	1.40 %

VIII. VALIDATION USING REAL IMAGES

A. *Vertical Surfaces*

For the first test concerning real images, we use vertical surfaces textured with the same pattern used for the synthetic images. The pattern was printed on a large canvas surface spanning 1.5x2 meters. The canvas was hung from a height slightly greater than 1 meter to create a well textured vertical surface for the evaluation (Fig. 9). The distance between the camera system and the canvas was measured using a laser rangefinder for maximum accuracy. Here we present the results from two scenarios, one at 25.31 meters from the canvas and one at 30.27 meters. For this system these correspond to disparities of 8.76 and respectively 7.3 pixels.

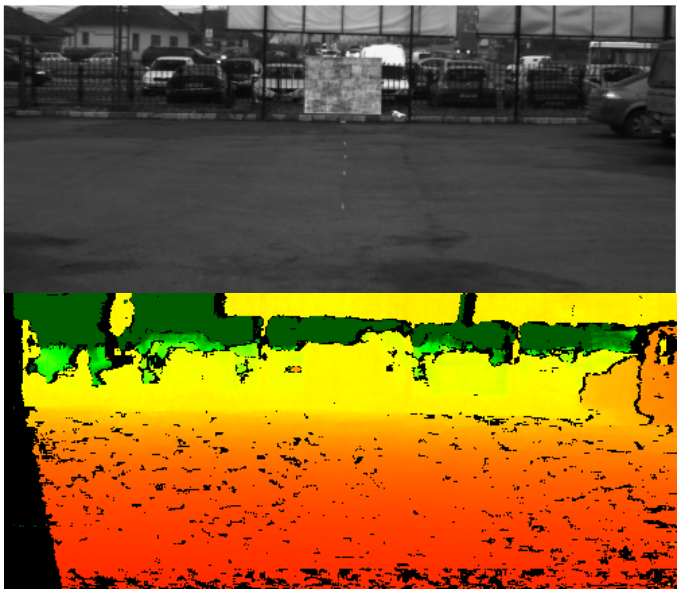

Fig. 9. Parking lot scene. Top image is original left image. Bottom image was generated using Sub-pixel Optimized Real-Time SGM algorithm.

To limit the effects of the imaging errors, we selected a rectangle of interest for both scenarios where the reconstructed surface was homogeneous. The distance values were averaged along the image row to reduce the spread. Table IX includes the distance deviations from the reference values provided by the rangefinder. The values are consistent with the previous evaluations. There is little difference between maximum and average values because each scenario covers a single disparity and the errors are similar for each image row.

TABLE IX
PIXEL ERRORS FOR REAL VERTICAL SURFACES

Method	AVERAGE 30.27M	AVERAGE 25.31M	MAX 30.2M	MAX 25.31M
Parabola	0.168	0.156	0.180	0.165
Linear	0.094	0.090	0.109	0.103
Histogram	0.036	0.037	0.051	0.051
Fitting	0.008	0.010	0.018	0.026

B. *Tilted Surface*

For a second test we used the same canvas to generate a tilted surface. A panel having a width of 2 meters and a height of 1 meter was used for support. The test scenario is similar with the tilted synthetic test. The surface ranges from 17.1 to 17.8 meters, corresponding to a disparity range from 8.03 to 7.71. Once more the results (table X) correspond to the synthetic tests.

Both of the real world tests validate the previous evaluations

and they prove that the proposed synthetic benchmark can replace real images when detailed information is needed about the environment.

TABLE X
ERRORS FOR REAL TILTED SURFACE

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.081	1.15 %	0.159	2.25 %
Linear	0.050	0.7 %	0.098	1.39 %
Histogram	0.024	0.34 %	0.050	0.73 %
Fitting	0.013	0.18 %	0.027	0.41 %

C. *Standard Benchmark*

The last validation uses the Middlebury benchmark [6] to measure the number of erroneous matches at sub-pixel level. A suitable image for sub-pixel interpolation is the Venus sequence with a number of tilted surfaces. The pixel locking effect is highlighted for this sequence as the surface looses its continuity with the traditional solutions. Images 2 and 6 are selected as the input pair because ground-truth is available for them with resolution of 0.125 pixels. For a further validation the Teddy and Cones sequences were also analyzed since they contain complex objects. Unfortunately ground truth is available only with an accuracy of 0.25 pixels.

TABLE XI
PERCENTAGE OF FALSE MATCHES

Method	VENUS (threshold of 0.125)	TEDDY (threshold of 0.25)	CONES (threshold of 0.25)
Parabola	37.6 %	17.9 %	24.59 %
Linear	29.0 %	15.4 %	22.44 %
Histogram	24.6 %	14.3 %	21.00 %
Fitting	23.9 %	14.3 %	21.40 %

In the case of the Venus images the number of erroneous matches is reduced significantly since the error threshold is low enough to highlight the problems of classical approaches. The improvements are also visible for the Teddy and Cones sequences, but are not as significant due to the higher error threshold. Fig. 10 presents the resulting disparity maps.

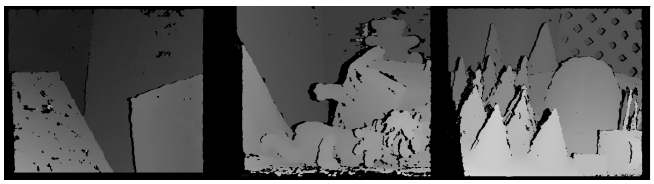

Fig. 10. Left to right reconstructed Venus, Teddy and Cones images.

D. *Advantages in environment perception*

The reduced depth error also improves the performance of associated environment perception systems. First and foremost object distance estimates will be more accurate since they are based on the individual point distances. The removal of the pixel locking effect also improves the homogeneity of the point distribution. Algorithms based on clustering or statistical sampling which uses this data will thus be more efficient and work at longer distances. One example of this behavior is visible in the case of the elevation map algorithm [24] which is now able to generate a more refined classified occupancy grid (Fig.11). High accuracy also allows a better delimitation between side-walk and road surfaces, increasing the curb-detection range.

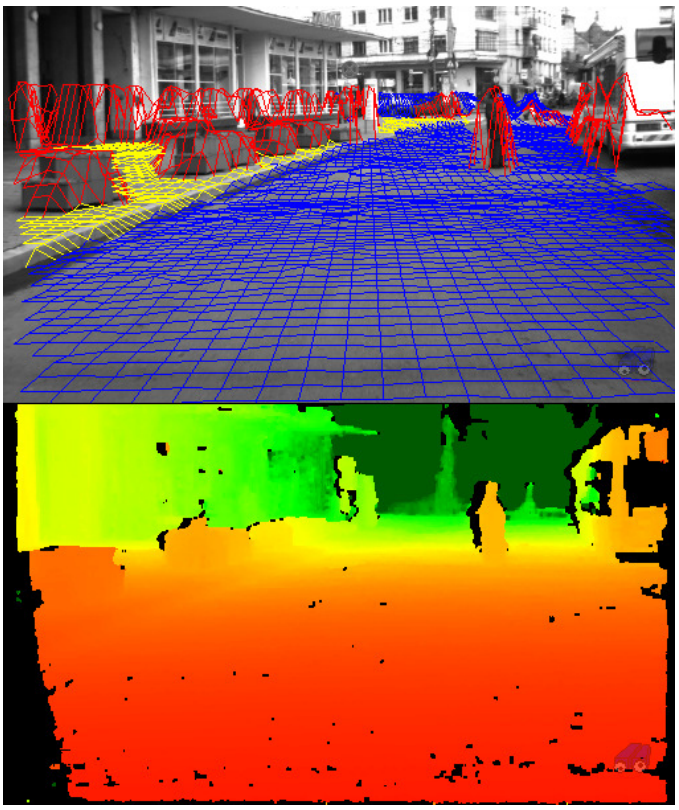

Fig. 11. Urban scene, top image is the classified occupancy grid, bottom image is the depth map.

IX. EVALUATION USING LOCAL STEREO ALGORITHM

One of the main ideas presented in this work is the dependence of the sub-pixel interpolation on the stereo algorithm. This behavior was observed during the evaluation of selected stereo system compared to a different real-time solution.

In this evaluation we use a local stereo algorithm using the Census and a multi-window setup. The system is similar to the one proposed, by Hirschmüller in 2002 [3]. Using the Census transform for the matching metric improves the pixel level quality compared to other metrics such as SAD or ZSAD [9], [10]. The multi-window setup takes into account 9 windows arranged in a 3x3 grid. The grid step is twice the window size. For the final cost we select the minimum between the original window cost and the averages along the horizontal, the vertical axis and the diagonals. The option of preserving the original window cost allows a more accurate reconstruction along object boundaries. The same confidence based filtering and left-right consistency check is used to eliminate the errors as in the original system using the SGM algorithm.

For the evaluation we use 2 images from the tilted surface set, the 30 and the 45 degree scenario. Tables XII and XIII

show the new error values.

TABLE XII
ERRORS FOR TILTED SURFACE (30 DEGREES) (LOCAL)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.061	1.49 %	0.137	2.99 %
Linear	0.012	0.29 %	0.061	1.26 %
Histogram	0.038	0.94 %	0.090	2.12 %
Fitting	0.063	1.56 %	0.140	3.02 %

TABLE XIII
ERRORS FOR TILTED SURFACE (45 DEGREES) (LOCAL)

Method	AVERAGE (PIXEL)	AVERAGE (REL)	MAX (PIXEL)	MAX (REL)
Parabola	0.052	1.36 %	0.129	2.77 %
Linear	0.011	0.29 %	0.033	0.92 %
Histogram	0.039	1.05 %	0.075	2.29 %
Fitting	0.063	1.68 %	0.116	3.37 %

For the traditional solutions the relative error is reduced by almost 2% compared to previous evaluations, with the linear interpolation being the best of all 4 options. The solutions proposed by us have the worst results, showing that they are not universal solution. Both evaluations are consistent with each other, showing that it is not an exceptional situation.

Even though local algorithms fare better in terms of sub-pixel interpolation with the traditional functions, these algorithms present pixel level deficiencies which limit their use in current systems. With the advances in hardware performance multiple real-time implementations were already presented which use the SGM algorithm for correlation. The sub-pixel advances presented in this paper are combined with the algorithm adaptations and optimizations presented in [9] to create a high performance system called Sub-pixel Optimized Real-time SGM (SORT-SGM). Fig. 9 and Fig. 12 show a comparison of this system with a high performance local algorithm based one, the Tyzx DeepSea development board. The huge difference in point density shows why modern algorithms such as the SGM are important for future development.

As a result defining a new interpolation function shape is not enough with the continuous evolution of stereo algorithms. It is more important to define clear and repeatable methodologies to adapt the sub-pixel interpolation to each stereo system. The two parts can then evolve side-by-side and sub-pixel accuracy is maintained.

X. CONCLUSION

The lack of accuracy of short-baseline stereo systems has long been considered one of its important downsides. However, by increasing the pixel accuracy by factor of 5, it becomes competitive with current wide-baseline solutions since accuracy is linearly proportional to the baseline.

One of the main ideas introduced in this paper is the correlation between the stereo algorithm and the sub-pixel interpolation. Although this correlation is expected from the mathematical model, literature has not considered it when presenting new sub-pixel interpolation models. The evaluation comparing the interpolation techniques shows different behavior when used together with different stereo algorithm. High accuracy cannot be achieved without using algorithm specific interpolation functions.

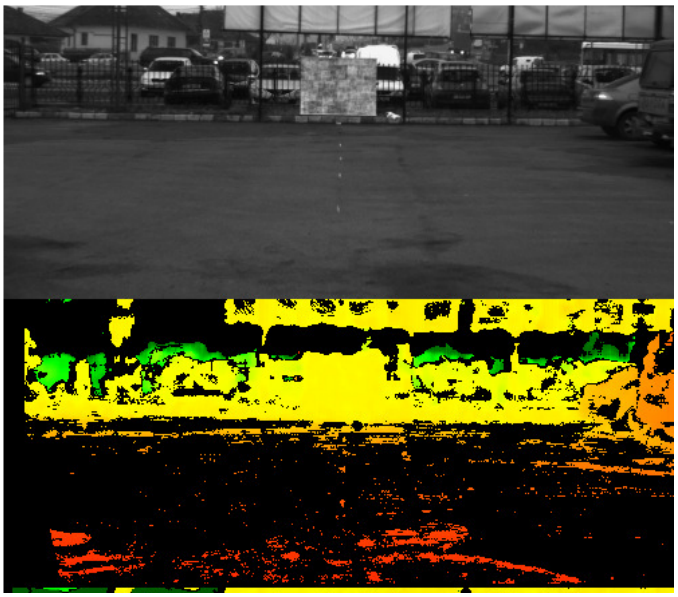

Fig. 12. Parking lot scene. Top image is original left image. Bottom image is depth image generated with Tyzx DeepSea development board.

As such two methodologies are proposed to solve this problem. Both methodologies are based on data provided by the stereo algorithm. Through this link the interpolation becomes dependant of the selected algorithm and matches its behavior.

Extensive evaluations show the improvements gained using the proposed methodologies. Traditional sub-pixel interpolation methods perform poorly when used with modern stereo solutions such as the SGM algorithm. The interpolation function resulted from the fitting process was the most accurate, having error rates several times reduced compared to the other solutions. The findings were validated through the use of both synthetic and real images take in different scenarios. The results were consistent across all evaluations.

In conclusion the proposed methods help designers generate algorithm specific interpolation functions which eliminate the pixel-locking effect. The simple 3 input function model is preserved, allowing easy integration into existing systems. The computational cost is also limited to a few arithmetic operations per pixel, similar with traditional solutions. These characteristics allow the new functions to be used as a drop-in replacement for a large range of existing stereo systems, improving accuracy with limited cost.

REFERENCES

[1] Scharstein, D., Szeliski, R., “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, International Journal of Computer Vision, vol. 47 no.1–3, pp. 7–42, April-June 2002.

[2] Gong, M., Yang, R., Wang, L., and Gong, M., "A Performance Study on Different Cost Aggregation Approaches Used in Real-Time Stereo Matching", International Journal of Computer Vision vol. 75, no. 2, pp. 283–296, Nov. 2007.

[3] Hirschmüller, H., Innocent, P. R., and Garibaldi, J. "Real-Time Correlation-Based Stereo Vision with Reduced Border Errors" International Journal of Computer Vision vol. 47, no.1–3, pp. 229–246, April–June 2002.

[4] Mark, W., Gavrila, D.M., "Real-time dense stereo for intelligent vehicles", IEEE Transactions on Intelligent Transportation Systems, vol.7, no.1, pp. 38–50, March 2006.

[5] Klaus, A. Sormann, M. Karner, K, "Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure", 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 15–18, 2006.

[6] Scharstein, D., Szeliski, R.: Middlebury stereo vision and evaluation page, http://vision.middlebury.edu/stereo

[7] Xu, Y., Chen, H., Klette, R., Liu, J. and Vaudrey, T., "Belief Propagation Implementation Using CUDA on an NVIDIA GTX 280", 22nd Australasian Joint Conference on Advances in Artificial Intelligence, Lecture Notes In Artificial Intelligence, vol. 5866, pp. 180–189, November 2009.

[8] Morales, S., Penc, J., Vaudrey, T., and Klette, R., "Graph-Cut versus Belief-Propagation Stereo on Real-World Images", 14th Iberoamerican Conference on Pattern Recognition, Lecture Notes In Computer Science, vol. 5856, pp. 732–740, November 2009.

[9] Hirschmüller, H., "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information" IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05, vol. 2, pp. 807–814, June 2005.

[10] Hirschmuller, H., Ernst, I., "Mutual Information Based Semi-Global Stereo Matching on the GPU", Lecture Notes in Computer Science, Advances in Visual Computing, vol. 5358, pp. 228–239, 2008.

[11] Gibson, J., Marques, O., "Stereo depth with a Unified Architecture GPU," Computer Vision and Pattern Recognition Workshop, pp. 1-6, 2008.

[12] Gehrig, S., Eberli, F., Meyer, T., "A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching", Lecture Notes in Computer Science, Computer Vision Systems, vol. 5815, pp. 134–143, 2009.

[13] Fisher, R.B. Naidu, D.K., "A Comparison of Algorithms for Subpixel Peak Detection", Image Technology, Advances in Image Processign, Multimedia and Machine Visio, Springer-Verlag, 1996, pp. 385–404.

[14] Bailey, D.G., "Sub-pixel estimation of local extrema", Image and Vision Computing, New Zealand, 2003, pp. 408–413.

[15] Shimizu, M., Okutomi, M., "Precise sub-pixel estimation on area-based matching", Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, Canada, pp. 90-97, 2001.

[16] Woodfill, J.I., et al., "The Tyzx DeepSea G2 Vision System, A Taskable, Embedded Stereo Camera", Embedded Computer Vision Workshop, pp. 126–132, 2006.

[17] Haller, I., Pantilie, C., Oniga, F., Nedevschi, S., "Real-time semi-global dense stereo solution with improved sub-pixel accuracy", Proceedings of the IEEE Intelligent Vehicles Symposium IV 2010, pp. 369–376, June 2010.

[18] Hermann, S., Klette, R., and Destefanis, E., "Inclusion of a Second-Order Prior into Semi-Global Matching", 3rd Pacific Rim Symposium on Advances in Image and Video Technology, Lecture Notes In Computer Science, vol. 5414, pp. 633–644, January 2009.

[19] Hirschmuller, H. Scharstein, D., "Evaluation of Cost Functions for Stereo Matching", IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07, pp. 1–8, June 2007.

[20] Hirschmuller, H. Scharstein, D., "Evaluation of Stereo Matching Costs on Images with Radiometric Differences," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 9, pp. 1582–1599, September, 2009.

[21] Szeliski R., Scharstein D., "Sampling the Disparity Space Image", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 419–425, 2004.

[22] Gehrig, S., Franke, U., "Improving Stereo Sub-Pixel Accuracy for Long Range Stereo", IEEE 11th International Conference on Computer Vision ICCV 2007, Rio de Janeiro, Brazil, pp. 1–7,2007.

[23] Haller, I., Pantilie, C., Tiberiu, M., Nedevschi, S., "Statistical method for sub-pixel interpolation function estimation", IEEE Intelligent Transportation Systems Conference, ITSC 2010, pp. 1098-1103, September 2010.

[24] Oniga, F., Nedevschi, S., "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle and Obstacle Detection", IEEE Transactions on Vehicular Technologies, vol. 59 , issue 3, pp. 1172-1182, 2010

Istvan Haller received the Bachelor Degree in Computer Science from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 2010. As student he carried out research activities in the field of image processing, stereo vision, and GPU based solutions, being involved in some national and international research projects. As student he received several awards for his research results.

Sergiu Nedevschi received the M.S. and PhD degrees in Electrical Engineering from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 1975 and 1993, respectively. From 1976 to 1983, he was with the Research Institute for Computer Technologies, Cluj-Napoca, as researcher. In 1998, he was appointed Professor in computer science and founded the Image Processing and Pattern Recognition Research Laboratory at the TUCN. From 2000 to 2004, he was the Head of the Computer Science Department, TUCN, and is currently the Dean of the Faculty of Automation and Computer Science. He has published more than 200 scientific papers and has edited over ten volumes, including books and conference proceedings. His research interests include Image Processing, Pattern Recognition, Computer Vision, Intelligent Vehicles, Signal Processing, and Computer Architecture.