A Comparative Analysis of Cross-Correlation Matching Algorithms Using a Pyramidal Resolution Approach*

Nuno Roma[1], José Santos-Victor[2], and José Tomé[3]

[1] Instituto Superior Técnico / INESC
Rua Alves Redol, No. 9 - 1000-029 Lisboa - PORTUGAL
Tel. +351 21 3100000 - Fax: +351 21 3145843,
E-mail: Nuno.Roma@inesc.pt, URL: http://sips.inesc.pt/~nfvr

[2] Instituto Superior Técnico / ISR
E-mail: jasv@isr.ist.utl.pt, URL: http://www.isr.ist.utl.pt/~jasv

[3] Instituto Superior Técnico / INESC
E-mail: Jose.Tome@inesc.pt

Abstract. Disparity map estimation is one of the most critical operations in several applications of the computer vision area. Many different algorithms have been proposed to solve this problem. However, such number of distinct approaches has often raised the question of choosing the most suitable one for a given application. Few and limited resources can be found in the literature covering this problem. The main objective of this paper is to present a comparative analysis of the performance and characteristics of a set of similarity measure algorithms proposed in the literature in the past few years. The obtained results can be regarded as an extremely valuable basis for selecting the most suitable registration algorithm to be used in a given application. The presented analysis was focused on the study of two distinct aspects: the final matching error and the computational load of each of the considered correlation functions. Besides this comparative study, the advantages of using a pyramidal resolution approach was also considered, in order to obtain fast running times and few arithmetic operations, with insignificant loss in the final matching error.

Keywords: Stereo Matching, Disparity, Correlation, Similarity Measure, Pyramidal Processing, Coarse-to-Fine.

1 Introduction

During the past few years, the estimation of the disparity field of an image sequence has been playing an increasingly important rule in a large number of applications of the computer vision area. Some examples of such applications can be found in video coding, multi-view image generation, online camera

* This research work has been partially funded by project PRAXIS 2/2.1/TPAR/2074/95, SIVA

self-calibration, 3D reconstruction from stereo image pairs, object recognition, mobile robots, visual control and motion estimation [1, 2]. The main purpose of this paper is to present a quantitative and comparative analysis of several cross-correlation similarity measure functions used in disparity estimation, based on a pyramidal resolution approach. Matching error and computational load measures will be used to compare the several registration algorithms, as well as the different possible configurations of the hierarchical processing scheme.

By definition, disparity is the difference between two matched point's coordinates. The disparity map is composed by a dense field of disparity vectors, one for each matched pixel or group of pixels. However, this definition can have some slightly different interpretations, depending on the considered application. While in video coding it is associated to a set of motion vectors, computed using two distinct images corresponding to two different time instants, in 3D reconstruction this field is computed using two images, corresponding to two different points of view of the same scene. The obtained vectors allow 3D perception by inferring depth information from the scene, thus enabling 3D reconstruction of objects [3]. To make this possible, a high-resolution estimation of disparity vectors of all image pairs is required, leading to the computation of high accuracy disparity maps.

Several different approaches have been developed to solve this correspondence search problem [1]. Most of them can be classified in three distinct groups:

- *Feature based algorithms*, consisting in first extracting predefined features from both images, such as, edge segments or contours, followed by the computation of the correspondence matching field;
- *Area based algorithms*, where registration algorithms using cross-correlation based similarity measures [4] are used to find the block of image pixels which best matches the one being analyzed in the other image;
- *Optical-flow algorithms*, relying on the relation between photometric correspondence vectors and spatiotemporal derivatives of luminance in an image sequence [5].

Although feature based algorithms and optical flow algorithms have been object of an intense research during the last years, conventional area based algorithms still remain very popular and will assume an important position in the next future in correspondence computation tasks. As an example, area based algorithms have been recently applied for use within digital topographical mapping systems of terrain imagery via satellite platforms or from aircrafts [6]. The main reason for this fact relies on their simple and straightforward implementation, well suited for parallel implementations based on VLSI circuits or Digital Signal Processors (DSP), and their robustness against certain image transformations. Furthermore, they directly generate the dense disparity map, whereas in feature based approaches and interpolation step is required if a dense map of the scene is desired. However, these algorithms present some serious drawbacks. The most serious one is concerned with the high computational load, usually associated to the dense matching field computation. To circumvent this disadvantage, a

hierarchical processing scheme is proposed in this paper to compute dense disparity maps. Furthermore, several different similarity measure functions have been proposed in the past. However, few and limited resources can be found in the literature comparing and characterizing the various aspects of the several approaches, as well as their advantages and disadvantages. Since the final performance can be greatly influenced by a correct selection of the used registration algorithm, a complete and exhaustive comparison of the several possible functions urges to be performed. With such a study, such as the one presented in this research work, it will be easier to select the most suitable similarity function to be used in a given application, depending on the desired final matching error level or on the available computational resources.

This paper is organized as follows. In Section 2 area based matching algorithms will be described, as well as their major advantages and disadvantages. Section 3 will present the several registration algorithms covered by this research. The pyramidal processing scheme will be described in Section 4. In Section 5 it will be presented the used evaluation methods and the achieved experimental results. Section 6 concludes the presentation.

2 Area Based Matching Algorithms

In area based matching algorithms, rectangular blocks of pixels from a set of two $M \times N$ images (left and right images) are compared and matched (see figure 1). For each block of the right image, designated by reference window, a correspondent block in the left image is sought using a given similarity measure as main criteria (see figure 2). During the search process, the correspondent block of the left image is moved by integer increments (c, l) around a predefined allowed region, designated by *search window*, and an array of similarity scores $d(c, l)$ is generated for integer disparity values (see eq.1). The position (c_M, l_M) of the moving block, corresponding to the maximum computed value of the considered similarity function $\odot$ for that search window, is selected and chosen to compute the optimum disparity vector corresponding to that reference window. Hence, a matching dense field $D(x, y)$ is obtained by using as many overlapping search windows as the number of pixels that compose the image, thus obtaining a single disparity vector for each pixel (see equation 3).

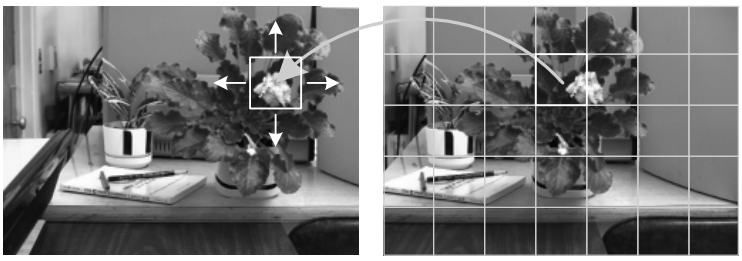

Fig. 1. Disparity map estimation process.

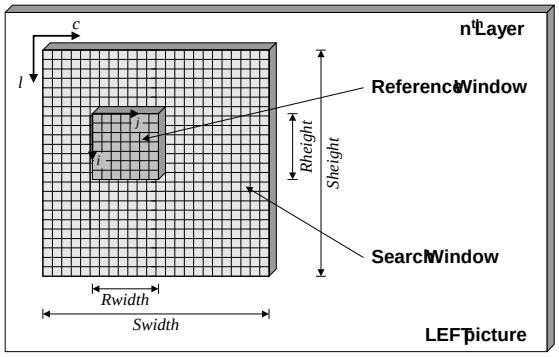

Fig. 2. Searching procedure.

$$d(c,l) = \sum_{v=0}^{Rwidth} \sum_{u=0}^{Rlenght} R(u,v) \odot S(c+u, l+v) \tag{1}$$

$$d(c_M, l_M) = \arg\max_{(c,l)} \{\, d(c,l) \;:\; 0 \le c < Swidth\,;\; 0 \le l < Slenght \,\} \tag{2}$$

$$\mathbf{D}(x,y) = \{\, d_{xy}(c_M, l_M) \;:\; 0 \le x < M\,;\; 0 \le y < N \,\} \tag{3}$$

Many different similarity measures have been referred in the literature [4]. The reasons for a particular selection are usually related to the computational load, algorithmic simplicity and achieved performance. Often, cross-correlation, normalized cross-correlation, sum of squared differences and sum of absolute differences are chosen. However, other similarity measures based on co-occurrence matrices have also been recently proposed [8].

Usually, the selection of the size of the reference and search windows is not a simple and trivial task [7, 1]. It has been shown that the probability of a mismatch usually decreases as the size of the reference window increases. However, using large windows leads to an accuracy loss, since the influence of image differences increases greatly with the increase of the considered area, as will be shown in section 5. In what concerns the size of the search window, this parameter influences the maximum allowed value of the resulting disparity vector. The greater the search window, the greater the allowed mobility of a given pixel. Therefore, this parameter must be great enough so that the search window comprises the correspondent block of a given reference window. Furthermore, a difficult and important trade-off must be done when selecting these window sizes, since the computational load and processing time usually increase quadractically with the size of these windows. Often, a compromise must be made, by adjusting these parameters according to the image size and contents.

Consequently, disparity estimation of a dense geometric matching field is usually considered to be one of the most challenging tasks of 3D reconstruction. This fact can be explained by several reasons, such as:

- The dimension of the solution space is extremely large, since each image pixel can, in principle, match any pixel of the other image.
- It is associated to very computational demanding algorithms.
- Its accuracy is extremely dependent on the photometric characteristics of the images being analyzed, such as texture, luminance and contrast, and the noise conditions associated to the acquisition camera system. This task can be particularly influenced in images with lack of information, e.g., regions with constant bright, horizontal edges and repetitive patterns.
- The existence of partially or totally occluded objects in the image pair can give rise to disparity errors with difficult solution.

The estimation of dense disparity maps is usually performed by taking in consideration a set of constrains relating the two images being analyzed. One of these constrains is the so-called *Constant Image Brightness* (CIB) assumption, which states that a corresponding pixel pair should have an equal luminance value [1]. Therefore, the main problem consists in finding the pixel pair whose neighborhood best matches the region being analyzed (see figure 3). Theoretically, all points of the other image can be considered as possible candidates to this matching process. However, there is at most only one pixel, which corresponds geometrically. Thus, the CIB constraint alone is considered to be insufficient for the estimation of dense disparity maps. For this reason, this estimation is often considered an ill-posed problem. One way of simplifying this task is to use some more constrains, making it possible to decrease the dimension and reduce the ambiguity of this problem, thus decreasing the total processing time. Some of the constrains more frequently used are [2]:

- *Epipolar Constraint*: makes it possible to convert a 2D search into a 1D search, by imposing that the matched points must lie on the corresponding epipolar line of the two images;
- *Unicity*: imposes that each pixel can have, at most, one correspondent pixel in the other image;
- *Smoothness*: imposes a continuous and smooth variation of the disparity field;
- *Order Constraint*: forces the order of the points belonging to an epipolar line to remain the same;
- *Disparity Gradient*: limits the allowed variation of the disparity values.

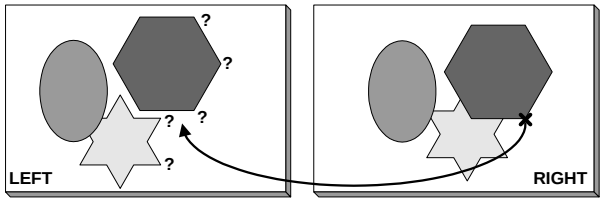

Fig. 3. Any point of the left image is considered to be a possible candidate to the matching process.

3 Cross-Correlation Algorithms

As it was previously referred, one of the main purposes of the present research is to establish a comparative and quantitative analysis of several registration algorithms, used as similarity functions in disparity map estimation. This research work was mainly focused on the study of correlation-type algorithms, due to their potentially robust real-time operation and their moderate requirements in terms of hardware and software resources. In table 1 it is presented the complete set of the studied correlation algorithms, as well as their definition expressions [4]. In this table, $R(u,v)$ denotes a reference window pixel, $S(c,l)$ a search window pixel, $\overline{R}$ the local mean of the reference window: $\overline{R} = \sum_{v=0}^{Rlenght} \sum_{u=0}^{Rwidth} R(u,v)$, and $\overline{S(c,l)}$ the pixel mean in the block of the search window being compared: $\overline{S(c,l)} = \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} S(c+u, l+v)$.

The functions SCC, NCC, $ZNCC$ and MOR are pure similarity measures. The best match corresponds to the maximum value of these functions. In contrast, the functions $NZSSD$, SSD, SAD, $NSSD$, $ZSSD$, $ZSAD$, $LSSD$ and $LSAD$ represent difference or dissimilarity functions. Therefore, the best match is obtained when the returned value of these functions is the minimum.

Among the several studied functions, some represent normalized versions with respect to mean and standard deviation of the SCC, SSD and SAD functions. The objective is to make these registration algorithms insensitive to changes of the brightness and contrast of $R(u,v)$ and $S(c,l)$ values [9]. Furthermore, in order to overcome some distortions on these measures in the vicinity of the image bounds, it was also performed, in all considered algorithms, a size normalization, by dividing each correlation result by the area of the correspondent reference window:

$$\hat{d}(c,l) = \frac{1}{Rwidth \times Rheight} \cdot d(c,l) \tag{4}$$

Although the described functions present evident analogies, the correspondent computational load and the requirements of each one can have significant differences. While with the SCC, the simplest function, it is only necessary to perform $Rlenght \times Rwidth$ multiply-and-accumulate (MAC) operations, in other functions it is necessary to use arithmetic units capable of performing squared-root operations ($ZNCC$, $NZSSD$, $NSSD$), absolute-value operations (SAD, $ZSAD$, $LSAD$) or integer divisions (NCC, $ZNCC$, MOR, $NZSSD$, $NSSD$, $LSSD$, $LSAD$). These requirements are often an important aspect when selecting the most suitable similarity measure function to be used in a given implementation, as will be further illustrated in section 5.

4 Pyramidal Processing Scheme

As it was referred in section 2, in order to obtain a high-accuracy disparity computation, it is important to use reference and search window sizes large enough to enable the computation of the correct match, even when pixel pairs present

Table 1. Cross-Correlation Algorithms.

Correlation Name	Definition
Simple Cross-Correlation **SCC**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R(u,v) \cdot S(c+u, l+v)$
Normalized Cross-Correlation **NCC**(**c**, **l**)	$\frac{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R(u,v) \cdot S(c+u,l+v)}{\sqrt{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R^2(u,v) \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} S^2(c+u,l+v)}}$
Zero Mean Normalized Cross-Correlation **ZNCC**(**c**, **l**)	$\frac{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v)-\overline{R}\right) \cdot \left(S(c+u,l+v)-\overline{S(c,l)}\right)}{\sqrt{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v)-\overline{R}\right)^2 \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}}$
Moravec **MOR**(**c**, **l**)	$\frac{2 \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v)-\overline{R}\right) \cdot \left(S(c+u,l+v)-\overline{S(c,l)}\right)}{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v)-\overline{R}\right)^2 + \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}$
Normalized Zero Mean Sum of Squared Differences **NZSSD**(**c**, **l**)	$\frac{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left[\left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right]^2}{\sqrt{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v)-\overline{R}\right)^2 \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}}$
Sum of Squared Differences **SSD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v) - S(c+u, l+v)\right)^2$
Sum of Absolute Differences **SAD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left\lvert R(u,v) - S(c+u, l+v)\right\rvert$
Normalized Sum of Squared Differences **NSSD**(**c**, **l**)	$\frac{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} (R(u,v)-S(c+u,l+v))^2}{\sqrt{\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} R^2(u,v) \cdot \sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} S^2(c+u,l+v)}}$
Zero Mean Sum of Squared Differences **ZSSD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left[\left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right]^2$
Zero Mean Sum of Absolute Differences **ZSAD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left\lvert \left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right\rvert$
Locally Scaled Sum of Squared Differences **LSSD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left(R(u,v) - \frac{\overline{R}}{\overline{S(c,l)}} \cdot S(c+u, l+v)\right)^2$
Locally Scaled Sum of Absolute Differences **LSAD**(**c**, **l**)	$\sum_{v=0}^{Rlength} \sum_{u=0}^{Rwidth} \left\lvert R(u,v) - \frac{\overline{R}}{\overline{S(c,l)}} \cdot S(c+u, l+v)\right\rvert$

significant disparity values. However, as it was also referred, the computational effort of this correspondence search increases significantly when these window sizes increase. Furthermore, larger windows are usually associated to longer computation times.

One way of solving these implementation issues is to use a hierarchical approach, by using a pyramidal processing scheme as the one depicted in figure 4 [1, 6]. With this technique, the matching process is done in a multi-layered fashion based on a coarse-to-fine approach, providing significant functional and computational advantages [10]. Each direction of the left and right images is down-sampled by a factor of 2, using a decimation function to achieve lower resolution versions of the image pair. The original images represent level 0 and images resolution decreases with the level number. This way, the pyramid may be viewed as a 4D data structure, where the pixels' intensity is a function $f(l, x, y)$ with 3 arguments: a level designator (l) and a pair of coordinates (x, y).

The matching estimation process starts at level L. This ensures that the earlier correlations are performed with the gross image features rather than with the details. The matching of these gross features will be used to guide the later high-resolution search, to achieve more accurate matches of these features and surrounding details. After this set of low-resolution pictures has been processed,

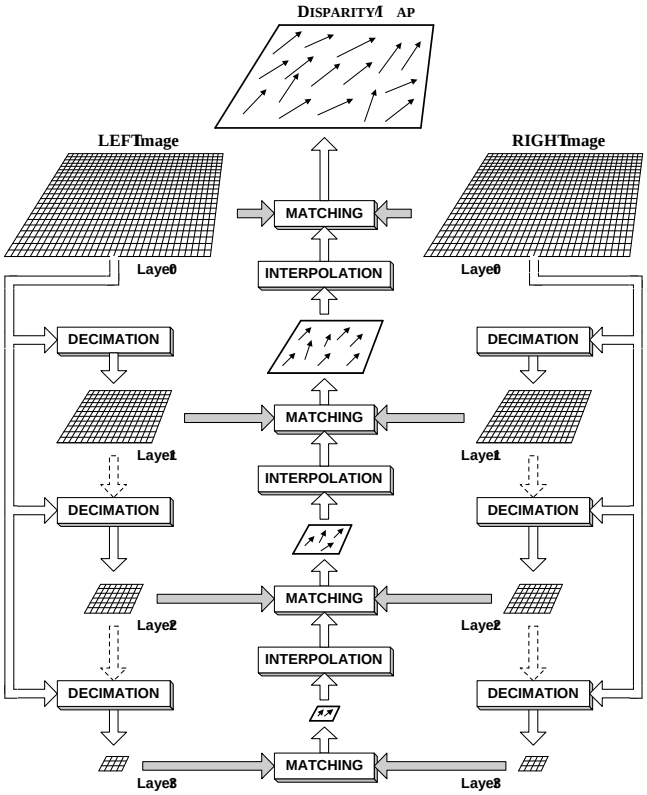

Fig. 4. Pyramidal Processing Scheme.

the obtained disparity map is interpolated to the resolution correspondent to level $L-1$. These disparity values are then used as an initial estimate for the computation of the disparity map of this level (see figure 4). This process continues until estimation of the disparity map corresponding to full resolution, at level 0, is performed. This way, for the estimation of the disparity field using several different levels, the same algorithm is repetitively applied to each of the considered levels.

Using this scheme, it is possible to use the same small size of the search and reference windows along the layered processing scheme. This way, each time the images resolution is increased, the image coverage of these windows is reduced by a factor of 4, thus enabling a gradual refinement of the matching process and a greater treatment of the picture's details. This also enables the achievement of accurate disparity values and a significant coverage area, only possible with the use of larger and more time consuming window sizes in a single layered processing architecture.

Some research works, referred in the literature, use an additional strategy to speed up the matching estimation, by working with sub-images rather than processing the entire image [11]. However, although with this solution the required memory space is lower, it involves an additional overhead in the whole processing scheme.

In the next subsections it will be described, in a more detailed way, several important aspects of the pyramidal processing scheme.

4.1 Number of Layers

One of the most critical decisions that arise when using a pyramidal processing scheme is related to the number of layers used in the architecture. Increasing the number of layers of the structure enables the use of smaller reference and search windows, thus leading to faster estimations of the dense disparity field. However, significant features and other kind of image information necessary to the matching process can be lost or distorted when too coarse resolutions are used, giving rise to critical problems in the search process [12]. Furthermore, multiscale image representations should be consistent, since features at different resolutions may be related [6, 12]. As resolution increases, significant features at different layers should not randomly appear or disappear. However, with careful design of the decimation and interpolation blocks of the hierarchical scheme, satisfactory results can still be achieved. Therefore, image size and its contents should always be considered in the decision of the number of layers to be used. Some applications referred in the literature [6] report the use of 4 or 5 layered pyramid structures derived from sets of 512×512 pixels images, achieving good results.

4.2 Decimation Function

A pyramidal structure is usually implemented by sub-sampling the original image. However, in order to satisfy the Nyquist theorem, a low-pass filtering stage

is necessary to be first performed in the original image. The filter implemented in this system is a gaussian filter, with mean centered at $\overline{m} = (m_x, m_y)$ and variance σ^2, and with an impulse response given by eq. 5.

$$h(\overline{x}) = \frac{1}{\sqrt{2\pi}.\sigma} \cdot e^{-\frac{(\overline{x}-\overline{m})^2}{2.\sigma^2}} \tag{5}$$

The frequency response of this filter can be shown to be given by eq. 6, having a 3dB bandwidth given by eq. 7.

$$H(\overline{f}) = e^{-2\pi^2\sigma^2\overline{f}^2} \tag{6}$$

$$BW_{-3dB} = \left.\frac{-\ln(\alpha)}{2\pi^2\sigma^2}\right|_{\alpha=\frac{1}{\sqrt{2}}} = \frac{\ln(2)}{4\pi^2\sigma^2} \tag{7}$$

Therefore, the set of $(M_L \times N_L)$ pixels, $f_L(i,j)$, that compose the image layer corresponding to level L can be obtained by performing a 2D convolution of the image matrix of size $(M_{L-1} \times N_{L-1})$, $f_{L-1}(i,j)$, corresponding to the layer of level $L-1$, with the impulse response of the gaussian filter $h(x,y)$, followed by the sub-sampling process.

The efficiency of this algorithm can be greatly improved. In fact, during the described process it is performed the computation of $(M_{L-1} \times N_{L-1})$ filter results and, after the sub-sampling process, only $(M_L \times N_L) = (M_{L-1} \times N_{L-1})/4$ pixels are used. If we consider this 25% efficiency in a K layered pyramid, it is possible to conclude that only $(0.25)^{K-1}$ % of the computations are used. To circumvent this limitation, it was decided to use a different approach, consisting in only performing the necessary filter computations corresponding to the pixels which are going to be in the sampling process to obtain the new desired layer. This procedure can be regarded as a combination of the filtering and sub-sampling phases, thus avoiding the computations of unnecessary results.

Another decision considered at this stage is related to the precision of the filter results. In order to avoid a gradual decrease of the precision as a consequence of multiple and cascaded quantization steps, it was decided to use always the original image as the input of all filter operations at all implemented layers. This implies a gradual and correspondent decrease of the filter bandwidth, as the current layer number increases. As it was shown in eq. 7, this can be easily done by increasing the filter variance σ^2 as the layer number is increased. In the current system, it was considered $\sigma_L^2 = 2 \times \sigma_{L-1}^2$ and a filter window of size $6 \times \sigma_L^2$ was used in the computation of the filtered layers. This window size guarantees that more than 99% of the area of the 2D gaussian filter of eq. 5 lies within this window.

In figure 5 it is presented the result obtained with the application of the described decimation function to the 256×256 *lenna* test image.

4.3 Matching Process

The matching process was performed using the set of correlation based similarity functions described in section 3 and presented in table 1. However, in order to

Fig. 5. Result of the application of a 4 layered decimation function using the *lenna* test image.

achieve an efficient implementation of each of these algorithms, some manipulations of the expressions presented in table 1 were performed, in order to obtain the final result in the minimum number of steps, using the minimum number of operations [11]. This is illustrated, for the general case of a zero-mean normalized cross-correlation, in eq. 8 through 11, where it was used the following simplified nomenclature, described in section 3: $R = R(u,v)$, $S = S(c+u, l+v)$, $\overline{S} = \overline{S(c,l)}$, $L = Rlength$, $W = Rwidth$, $\sum\sum\alpha = \sum_{v=0}^{L}\sum_{u=0}^{W}\alpha$.

$$d(c,l) = \frac{cov_{c,l}(f,g)}{var_{c,l}(f) \times var_{c,l}(g)} \tag{8}$$

$$\begin{aligned} var(f) &= \sum\sum\left(R-\overline{R}\right)^2 \quad = \quad \sum\sum\left(R^2 - 2R\overline{R} + \overline{R}^2\right) \\ &= \sum\sum R^2 - 2\cdot\frac{\left(\sum\sum R\right)^2}{L.W} + \left(\frac{\sum\sum R}{L.W}\right)^2 \times L.W \\ &= \sum\sum R^2 - \frac{\left(\sum\sum R\right)^2}{L.W} \end{aligned} \tag{9}$$

$$var(g) = \sum\sum S^2 - \frac{\left(\sum\sum S\right)^2}{L.W} \tag{10}$$

$$\begin{aligned} cov_{c,l}(f,g) &= \sum\sum\left(R-\overline{R}\right)\left(S-\overline{S}\right) \\ &= \sum\sum R.S - \overline{R}\sum\sum S - \overline{S}\sum\sum R + L.W \times \overline{R}.\overline{S} \\ &= \sum\sum R.S - \frac{\sum\sum R}{L.W}\cdot\sum\sum S - \frac{\sum\sum S}{L.W}\cdot\sum\sum R + \\ &\quad L.W \times \frac{\sum\sum R}{L.W}\cdot\frac{\sum\sum S}{L.W} \\ &= \sum\sum R.S - \frac{\sum\sum R\cdot\sum\sum S}{L.W} \end{aligned} \tag{11}$$

Hence, for the majority of these algorithms, it is only necessary to compute the values of $\sum_{v=0}^{L}\sum_{u=0}^{W}R$, $\sum_{v=0}^{L}\sum_{u=0}^{W}S$, $\sum_{v=0}^{L}\sum_{u=0}^{W}R.S$, $\sum_{v=0}^{L}\sum_{u=0}^{W}R^2$ and $\sum_{v=0}^{L}\sum_{u=0}^{W}S^2$ in one single step.

Besides these manipulations, a size normalization was also performed, as illustrated in eq. 4.

4.4 Interpolation

As it was previously described, the several disparity maps estimated in the lower levels of the pyramid are used as an initial estimate of the disparity fields of subsequent higher levels, following a classic coarse-to-fine approach. However, before these initial estimates can be used in the computation, a scaling up operation is necessary to be performed on the disparity map of the previous layer, in order to conform its dimension and its disparity vectors with the new layer resolution.

To implement this function it was decided to use a bilinear interpolation algorithm based on the computation of the mean disparity value of the group composed by 4 or 2 neighbor disparity vectors corresponding to the set of pixels belonging to a 3×3 interpolation window.

4.5 Disparity Maps

The final result of this pyramidal processing scheme is a dense disparity map. This map can be seen as a $(M_0 \times N_0)$ sized array, where each element of this array is a data structure composed by 3 values:

- Disparity value along the xx axis.
- Disparity value along the yy axis.
- Similarity measure value of the correspondent pair of pixels.

Hence, this data structure can be viewed as a set of 3 $(M_0 \times N_0)$ arrays. Since there is a direct relation between the calculated correlation values and the achieved matching performance, the similarity measure array can be regarded as a confidence map of the final result. Therefore, it can be used to select the pixel coordinates corresponding to the best match of the whole process.

5 Experimental Results

The comparative analysis presented in this research was based in a software implementation of the described algorithms using the object oriented language C++. The several computations, performed along this experiment, were done using general purpose Linux and Windows NT workstations. In the following subsections, it will be described the experiment layout and presented the achieved experimental results.

5.1 Experiment Layout

In the performed comparative analysis, it was used a set of two image pairs representing, respectively, a scene taken at planet Mars and an aerial view of the Pentagon (see figures 6 and 7). These scenes are considered to be good examples of high textured pictures, well suited for evaluating area-based matching algorithms. In the several computations carried out along this research, it was considered two main aspects to assess the several algorithms: *matching error* and *computational load*.

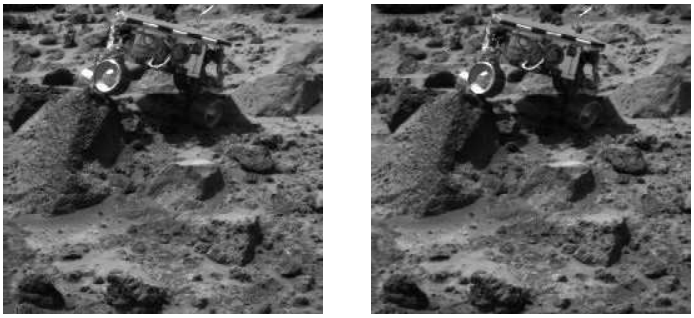

Fig. 6. Left and right images of Sojourner, taken from Pathfinder lander camera at planet Mars.

Fig. 7. Left and right images of an aerial view of the Pentagon.

In order to obtain a comparison of the resultant disparity maps as fair as possible, it was decided to define a measure which equally assessed the result of the several algorithms. This common measure is the final registration error of each similarity function and was estimated by computing, for each pixel of the right image, the sum of the squared differences, e_{xy}, between all pixels belonging to a rectangular neighborhood window of size $(K \times L)$ of the right image, $f(x, y)$, and all pixels belonging to the correspondent window of the left image, $g(x, y)$, defined with the correspondent disparity vector (d_x, d_y) (see eq. 12).

$$e_{xy} = \sum_{i=-\frac{K}{2}}^{+\frac{K}{2}-1} \sum_{j=-\frac{L}{2}}^{+\frac{L}{2}-1} \left[f(x+i, y+j) - g(x+i+d_x, y+j+d_y) \right]^2 \qquad (12)$$

By evaluating the square root of this sum and by dividing it by the area of the considered window, it was obtained a value which enables us to quantify the resultant matching error in the pixel domain (eq. 13). Moreover, by performing the sum of all these $E(x, y)$ values and by dividing it by the total image area, it is possible to obtain a value which best characterizes the performance of a given algorithm (eq. 14). These values were then used in the several comparative charts presented in the following subsections. With the set of all $E(x, y)$ values, it was also possible to obtain a matrix $\mathbf{E}(x, y)$, denominated by error map (eq. 15).

$$E(x,y) = \frac{\sqrt{e_{xy}}}{K \times L} \tag{13}$$

$$E = \frac{\sum_{x=0}^{M} \sum_{y=0}^{N} E(x,y)}{M \times N} \tag{14}$$

$$\mathbf{E}(x,y) = \{ E(x,y) \ ; \ 0 \leq x < M \ ; \ 0 \leq y < N \} \tag{15}$$

In what concerns the computational load, the implemented program was designed in order to provide a statistical study of all arithmetic operations performed along the disparity map estimation, namely, *multiplications*, *additions* and *other* similarity measure specific functions, such as, square roots, absolute values and integer divisions.

Furthermore, the several estimations were also focused on the analysis of several different aspects related to matching estimation using a pyramidal structure. All experiments were tested using the complete set of similarity measure functions presented in table 1 in order to evaluate the specific characteristics of each of the described functions. The performed tests intend to study the influence of several aspects in the final disparity error and computational load, such as:

- *Pyramid depth*, i.e., the number of layers used in the hierarchical processing;
- *Pixel mobility*, concerning the maximum value of the disparity vector allowed for each pixel, controlled by adjusting the *search window size* parameter;
- *Reference window size*;
- *Computational load distribution* among the several stages of the system.

In the following subsections, it will be presented the achieved comparative results, obtained using the Mars stereo image pair. These results are similar to those obtained using other test images, in what concerns all of the studied aspects described in the previous paragraphs.

5.2 Disparity Maps

In figure 8 it is presented a graphical representation of the computed matrices, composed by the components of the disparity vectors along the xx and yy axis. These matrices were obtained applying the $ZNCC$ similarity measure function with a 2 layered pyramidal structure, a 64 pixels sized search window and a 13 pixels sized reference window to the Mars stereo image pair. In this figure, it is possible to see a gradual increase of the amplitude in the y direction of the disparity vectors along the xx axis. This fact conforms with the expected behaviour, since these vectors correspond to image points farther from the camera system position, thus presenting more significant coordinates differences. The representation shown in figure 9 presents these matrices in a more intuitive way.

In figure 10 it is presented the correlation map correspondent to this configuration and the error map obtained by using eq. 14. In this figure it is possible

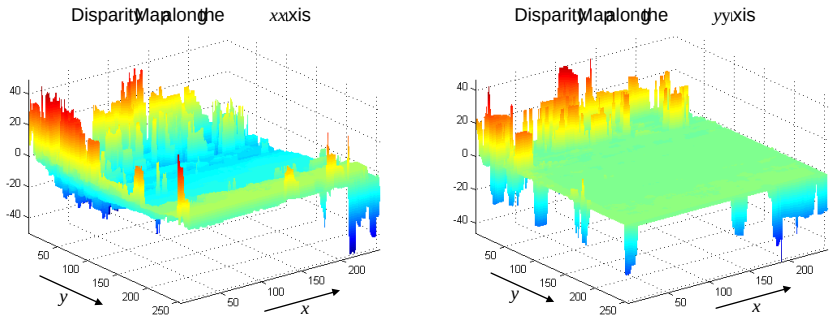

Fig. 8. Disparity values along the xx and yy axis.

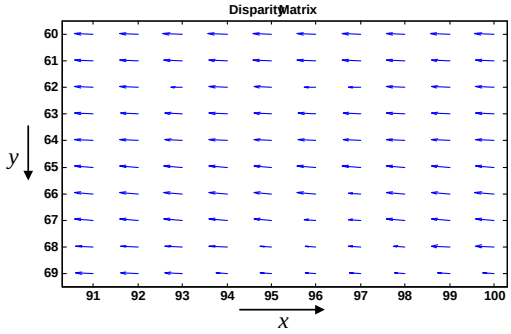

Fig. 9. Disparity Matrix.

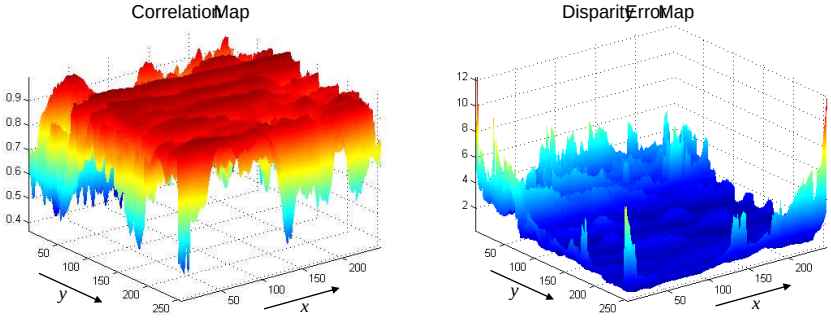

Fig. 10. Correlation and Error Maps.

to distinguish image areas with significant higher values of the disparity error, namely, at the borders of the graph. This higher values are usually due to non-overlapping regions of the image pair. Some of the peaks found in the central area of the disparity error map are usually due to matching mistakes or occluded objects. Several different solutions have been proposed in the literature to eliminate these matching mistakes. As an example, Sun et al. have proposed a dynamic programming algorithm [11]. In the present research it was studied

a different approach based on the previous described *Smoothness* constraint of the disparity field. The main idea consists in performing a median filtering stage before applying the interpolation step of the obtained disparity field at each level of the pyramid scheme. The objective is to eliminate abrupt maximum and minimum peaks of this field. Unfortunately, the achieved results demonstrated effective improvements of only 5% on the final disparity error. Therefore, it was decided to disable this intermediary block on the overall processing scheme in the subsequent study of the required computational load presented in the next subsections.

5.3 Disparity Error

In figure 11 it is represented the variation of the disparity error with the allowed pixel mobility. In this and in the following charts it has been defined the measure *Full-Mobility (FM)*, designating the largest allowed value of any disparity vector of the estimated map. This value corresponds to the limit situation, such as the one where there is a match between the lower-left pixel of one image and the upper-right pixel of the other image. In a non-hierarchical scheme this limit situation would give rise to the usage of a search window area equal to twice the original image area.

As it can be seen, for the majority of the studied configurations, it is possible to distinguish one configuration corresponding to the best match. In the presented case, it corresponds to the situation where the maximum allowed disparity value is $FM/8$, which corresponds to a 32 pixel sized search window. In fact, for smaller windows, the increase of the error value was already expected, since when using too small windows the probability to find the correct correspondent pixel in the other image decreases. The increase of the disparity error when using larger windows is probably due to an increase of the ambiguity of the image features in larger search areas.

Figure 12 represents the variation of the disparity error with the number of levels of the used pyramidal scheme. From the obtained values, it is possible to conclude that better results can be achieved with fewer hierarchical levels. The reason for this fact can be explained because of the distortions and feature

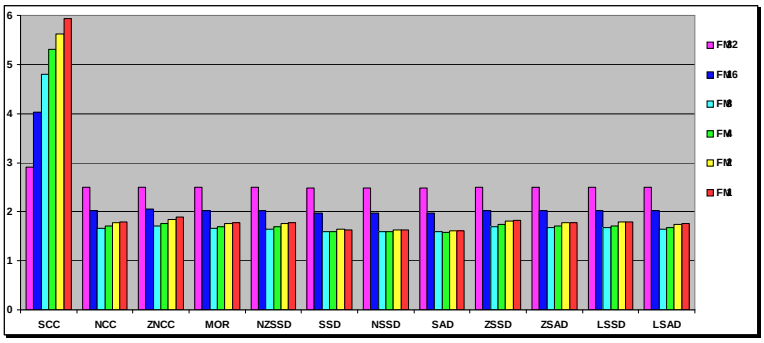

Fig. 11. Influence of the allowed pixel mobility in the final disparity error.

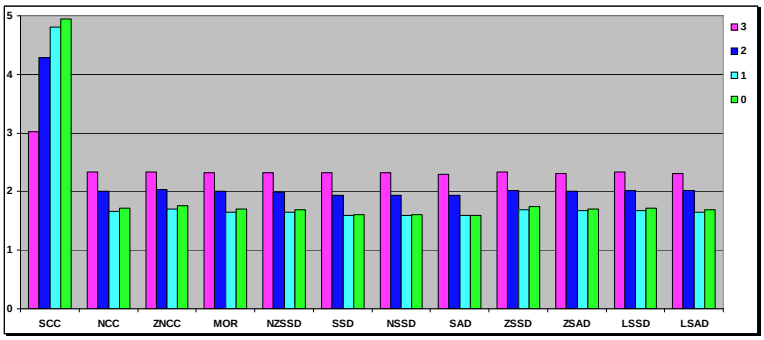

Fig. 12. Influence of the pyramid depth in the final disparity error.

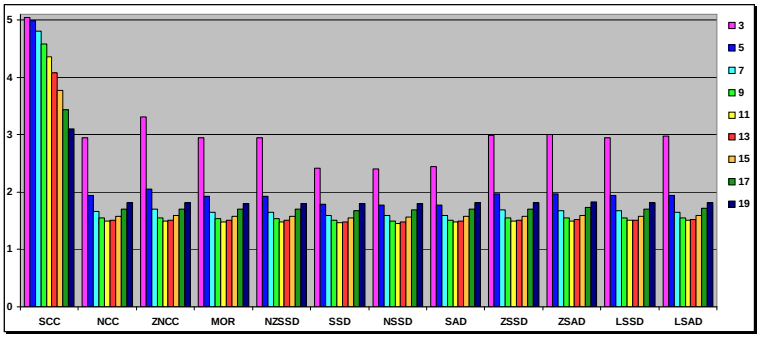

Fig. 13. Influence of the reference window size in the final disparity error.

losses on the images when using too coarse resolutions, giving rise to a critical loss of very important reference points, essential to perform the matching of image regions.

Figure 13 describes the influence of the used reference window size in the final disparity error. From the achieved results, it is possible to conclude which there is an optimal value for this parameter that corresponds to the best configuration. For this considered case, the reference window size should be defined between 11 and 13 pixels. In fact, for smaller windows, the increase of the error value was already expected, since with too small reference windows such as 3×3 windows, it becomes very difficult to distinguish between the several image features, giving rise to a substantial increase of the matching ambiguity. The increase of the disparity error associated to larger windows can be justified because, when using too large reference windows, it becomes more difficult to find the correct match due to the increasingly influence of the differences encountered in the image pair, due to the different points of view of the acquisition camera system.

The previous charts enable also to perform a general comparison of the several studied similarity measure functions. From these charts, it is possible to conclude that using a convenient value for the search window size and reference window size, the final results can be very similar. However, it is possible to distinguish better results obtained using SSD, $NSSD$ and SAD similarity measure functions. The simple cross-correlation function (SCC) presented the worst set

of disparity error values. Consequently, its disparity error was not considered in some of the previous comparisons.

5.4 Computational Load

In figure 14 it is presented a statistical comparison of the required number of *multiplications*, *additions* and *other* operations, when using several different values for the maximum allowed pixel mobility. In these and in the following graphs, the general designation *Other* function corresponds to the computation of:

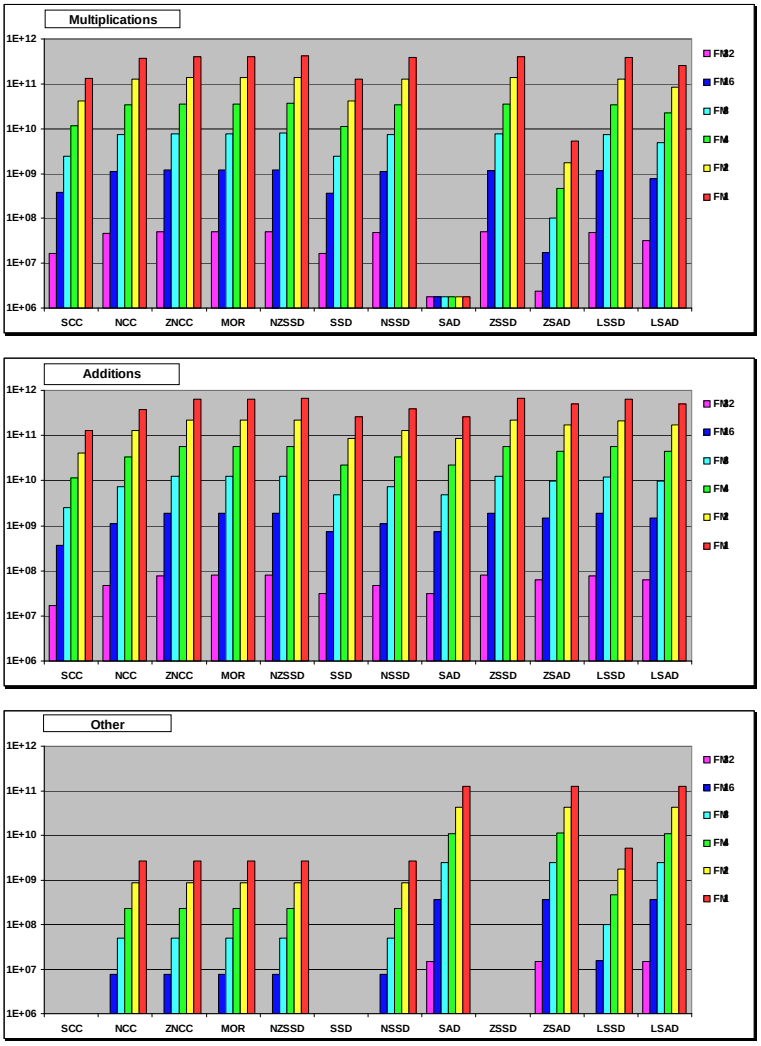

Fig. 14. Influence of the allowed pixel mobility in the number of arithmetic operations.

- $Other\,(a,b) = \frac{a}{\sqrt{b}}$ in NCC, $ZNCC$, $NZSSD$ and $NSSD$ similarity functions;
- $Other\,(a,b) = \frac{a}{b}$ in MOR and $LSSD$ similarity functions;
- $Other\,(a) = |a|$ in SAD, $ZSAD$ and $LSAD$ similarity functions.

From these charts it is possible to conclude that the increase of the mobility factor (reflected in a correspondent increase of the search window size) is responsible for a significant increase of the number of arithmetic operations performed along the estimation process. As an example, it is possible to verify that using a search window corresponding to $FM/1$ can mean a computational load 8000 times higher than the one corresponding to $FM/32$.

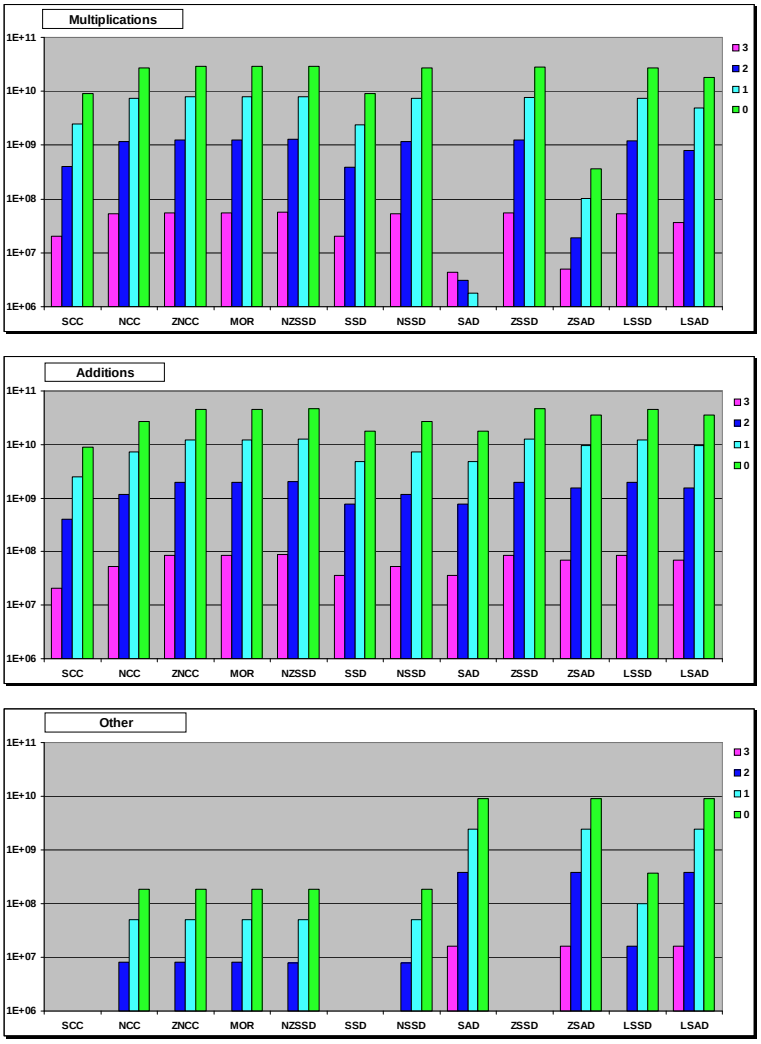

Fig. 15. Influence of the pyramid depth in the number of arithmetic operations.

In figure 15 it is shown the statistical comparison of the number of required arithmetic operations to estimate the dense disparity field using 0 (original resolution), 1, 2 and 3 hierarchical levels. With these results, it is evident the significant advantages of using a hierarchical approach. The achieved results allow us to conclude that the computational load of a 3 layered scheme can be 500 times lower than the one based on a 0 level approach. The exceptional lower number of multiplications correspondent to SAD similarity function was already expected since, for this registration algorithm, this operation is only performed in the filter and interpolation stages. In figure 16 it is represented this relation using a percentual analysis. These graphs evidence the significant difference of

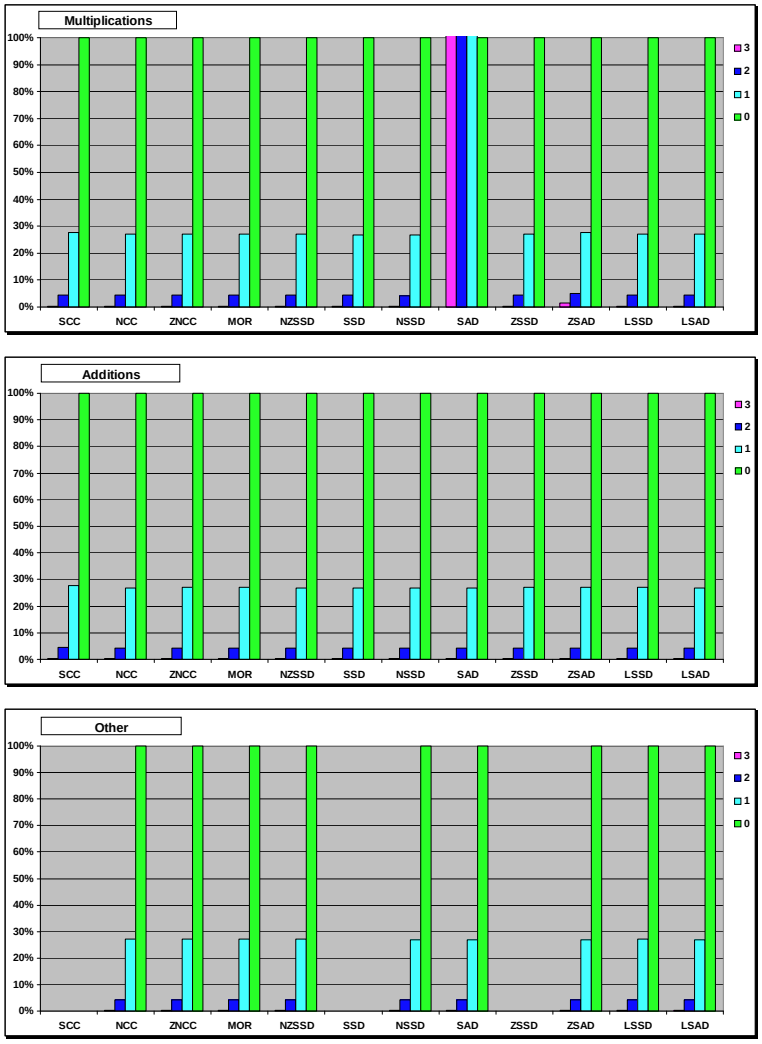

Fig. 16. Influence of the pyramid depth in the number of arithmetic operations (percentage analysis).

the number of performed operations when using a pyramidal scheme with 0, 1, 2, or 3 levels.

In figure 17 it is presented the variation of the computational load with the size of the reference window. In these charts, it is evident the general increase of the number of arithmetic operations performed in the matching estimation stage with the increase of the reference window size.

Finally, in figure 18, it is presented the distribution of the global computational load in the three main stages of the system: *Similarity computation*, *Gaussian filtering* and *Disparity map interpolation*. From these charts, it is evident that the similarity measure computation is responsible for the major part of the set of performed operations. Therefore, these results demonstrate that the

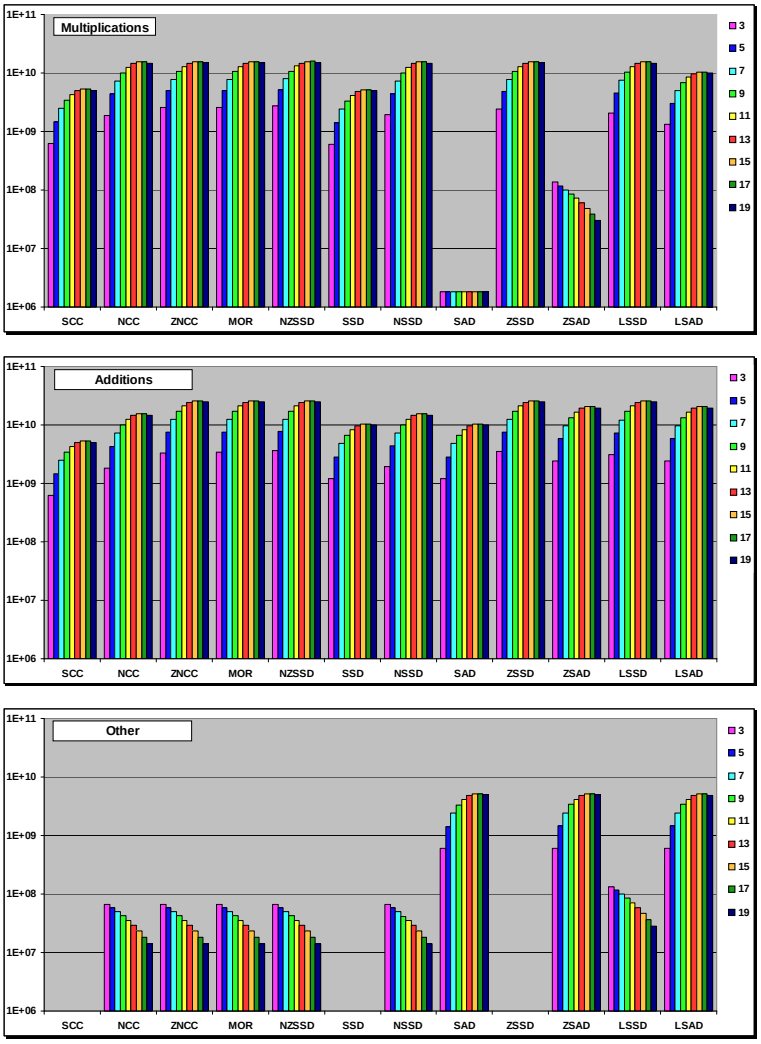

Fig. 17. Variation of the number of operations with the size of the reference window.

overhead of the several auxiliary stages required to perform a pyramidal processing scheme have an almost insignificant weight in the overall computational load.

Hence, the results presented in this subsection constitute a very important comparison basis to select the most convenient similarity function and the most suitable set of parameters of the pyramidal processing scheme to estimate the dense disparity map of a given stereo image pair. Therefore, the results presented in the several charts of this section should constitute a primordial basis of decision in the critical tradeoff between computational load saving and final disparity error minimization.

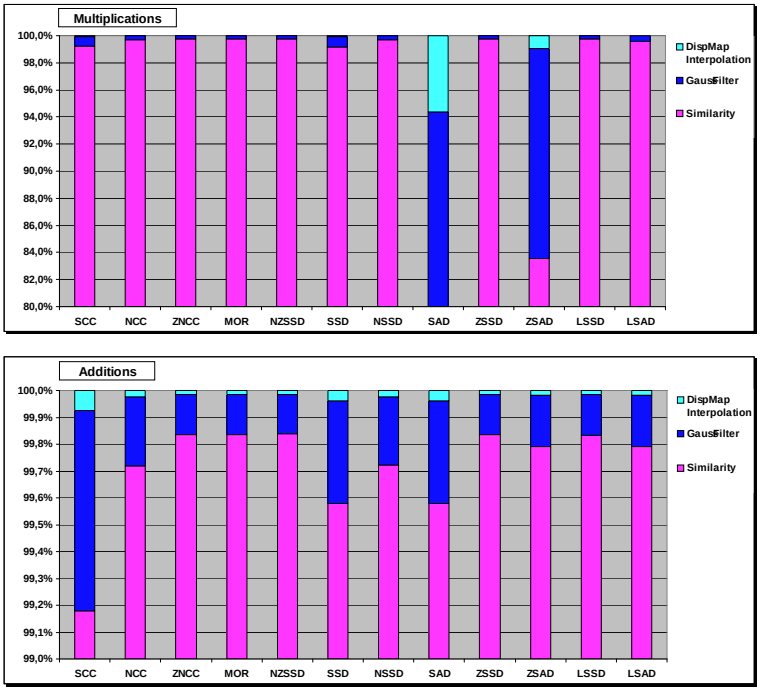

Fig. 18. Distribution of the *multiply* and *add* operations in the several stages of the system.

6 Conclusion

This paper presents a comparative analysis of several area based similarity measure functions, using a pyramidal resolution scheme. The studied similarity functions constitute a set of twelve different cross-correlation based matching algorithms. Among this set, it has been verified that better results can be obtained when using zero-mean normalized similarity functions, such as, $ZNCC$ and MOR. Dissimilarity functions like SSD and SAD have also proved to give rise to good results.

The presented research has also shown the influence of several parameters of this hierarchical scheme on the overall *Disparity Error* and *Computational*

Load performances. Among this set of parameters, the number of layers, the search window size and the reference window size have denoted as predominant parameters to achieve a convenient tradeoff between the two referred factors. Moreover, some registration algorithms based in the computation of dissimilarity measures using the absolute value operator (SAD, $ZSAD$ and $LSAD$) have proved to be very cost effective in what concerns the required computational resources. This fact is tied to the low required number of multiplication operations, usually associated to higher demanding resources, in what concerns arithmetic units and processing time. As a consequence, recent implementations based on VLSI circuits or Digital Signal Processors have adopted this set of registration algorithms.

The software package implemented along this research work is freely available at the following address: *http://sips.inesc.pt/~nfvr/research/matching*.

7 Acknowledgements

The stereo images of *Sojourner*, taken at planet Mars by Pathfinder lander were obtained at: *http://www.psc.edu/Mars/default.html*. The stereo images of *Pentagon* and *Plants* were obtained at: *http:// www.ius.cs.cmu.edu//idb/html/stereo*.

References

1. A. Redert, E. Hendriks, J. Biemond, "Correspondence Estimation in Image Pairs", *IEEE Signal Processing Magazine*, May 1999, pp 29-45.
2. A. Arsénio, J. S. Marques, "Performance Analysis and Characterization of Matching Algoritms", *Proc. of the* 5^{th} *International Symposium on Intelligent Robotic Systems*, Stockholm, Sweden, July 1997.
3. M. G. Strintzis, S. Malassiotis, "Object-Based Coding of Stereoscopic and 3D Image Sequences", *IEEE Signal Processing Magazine*, May 1999, pp 14-28.
4. P. Aschwanden, W. Guggenbühl, "Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms", *in Forstner and Ruweidel, eds.*, *Robust Computer Vision*, pp 268-282, Wichmann, (1992).
5. E. Grossmann, J. Santos-Victor, "Performance Evaluation of Optical Flow Estimators: Assessment of a New Affine Flow Method, Etienne Grossmann", *VisLab-TR 07/97 - Robotics and Autonomous Systems*, Elsevier, July 1997.
6. M. O'Neill, M. Denos, "Automated System For Coarse-To-Fine Pyramidal Area Correlation Stereo Matching", *Image and Vision Computing, vol.14*, 1996, pp. 225-236.
7. O. Faugeras et. others, "Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms", *in Forstner and Ruweidel, eds.*, *Robust Computer Vision*, pp 1-26, Wichmann, (1992).
8. H. Hseu, A. Bhalerao, R. Wilson, "Image Matching Based On The Co-occurrence Matrix", 1999.
9. R. Gonzalez, R. Woods,"Digital Image Processing", *Addison-Wesley*, 1993.
10. D. Ballard, C. Brown,"Computer Vision", *Prentice-Hall, Inc.*, 1982.
11. C. Sun, "Multi-Resolution Rectangular Subregioning Stereo Matching Using Fast Correlation and Dynamic Programming Techniques", August 1998.
12. R. Schalkoff,"Digital Image Processing and Computer Vision", *John Wiley & Sons, Inc.*, 1989.