A Fault-Robust SPMD Architecture for 3D-TV Image Processing

Article · January 1997

DOI: 10.1007/978-1-4615-5449-3_13

3 authors, including:

Bruno Ciciani

Sapienza University of Rome

67 PUBLICATIONS **429** CITATIONS

SEE PROFILE

Milton Romero

Universidade Federal de Mato Grosso do Sul

12 PUBLICATIONS **26** CITATIONS

SEE PROFILE

A Fault-Robust SPMD Architecture for 3D-TV Image Processing

Armando Chiari[°] , Bruno Ciciani*, Milton Romero*

°Fondazione "Ugo Bordoni", Viale Europa 190, 00144 Roma, Italy; e.mail: chiari@fub.it
* Department of Computer and System Engineering, University of Rome "La Sapienza",
Via Salaria 113, 00198 Roma, Italy; e.mail: {ciciani,miltonr}@dis.uniroma1.it

Abstract

A fault-tolerant embedded SPMD architecture to support a pyramidal Quad-Tree motion/disparity estimation algorithm for stereoscopic image processing is proposed in this paper. The fault tolerance is obtained by using spares only for the image quality bottleneck modules (which are 6 on 90 modules) and by data derouting. Given the structure of the architecture that permits error confinement, the data derouting allows a graceful degradation of the image quality, progressively decaying with the number of faulty modules.

1. Introduction

In this paper we present an embedded computing system for stereoscopic image processing applications, that in case of fault occurrence degrades its performance to a still acceptable quality level for most video services.

Typical applications of interest range from robot vision image processing for critical tasks such as the remote manipulation of objects in hostile or otherwise unaccessible environments [5], to multimedia real-time video codecs [6, 7, 8, 9]. In these applications a prediction of image areas displacements is usually performed between two images adjacent in time (motion vectors) or/and in space (disparity vectors), to minimize the transmission bit-rate over telecommunications networks for a given image quality level. According to such applications, it is essential to provide some recovery mechanisms that in the case of faults may control the spreadout of the noise on the corrupted images.

Various algorithms have been published in the literature to face the problem of the depth estimation of objects represented by three-dimensional imaging techniques. In particular stereoscopically genereted image pairs can be processed to extract such an information.

As well known, a Quad-Tree approach to the motion/disparity estimation of stereo image pairs acts at different image resolution levels, according to a hierarchical model [1, 2, 3, 4]. At each resolution level the algorithm processes a suitably decimated (sub-sampled) version of the original stereo pairs in order to perform a local disparity estimation, also taking into account the estimation processed at the lower resolution level (if any). The number of different resolution levels can be optimized according to both subjective (visual) quality assessment and algorithm processing load.

In this work the Quad-Tree algorithm has been specialized to the evaluation of the "dense" disparity map (DM) of 3D-TV images, involving a disparity estimation for each pixel of the original images. Our system is designed to support a pyramidal Quad-Tree motion/disparity estimation algorithm for stereoscopic image processing, in an extended form of the basic version proposed in the literature [1, 2, 3, 4] to take into account the presence of faults. This architecture is oriented to the real-time video signal processing and it is also designed for on-line testability.

[°] Work carried out in the framework of the Agreement between the Italian PT Administration and the Fondazione "Ugo Bordoni".

The basic idea that guides us in the design of the computing system is to tolerate faults by replacing faulty processing components acting as image quality bottlenecks and allowing performance degradation in case of faults in all other processing elements. The performance degradation is achieved by an error confinement strategy that, in turn, is obtained through suitable de-routing resources.

The performance of our system in case of fault-free conditions is evaluated by means of the quality of the output depth (or "disparity") map (DM) image produced by the processing system in response to a set of input test stereoscopic image pairs. Such DM shall in turn be regarded as a reference image to assess the performance of the system in the presence of faults. To this purpose a set of DM's is produced according to various fault location assumptions, so that the system performance can be evaluated by both an objective criterium, i.e. the mean square error (MSE) quantity, and by a subjective assessment of the output image quality. Such measures confirm theoretical assumptions about the robustness of our structure to noise propagation due to faulty PE's. Namely, a set of experimental results are reported for both single and multiple fault assumptions and it is shown: 1) how the structure can tolerate multiple faults, by confining errors on the branches of the sub-tree under the faulty PE (fault- root); 2) how it is possible to predict the amount of error energy on the final DM in the assumption of multiple faults by a simple model based on the superposition of effects. The error confinement allows for a graceful degradation of the image quality, progressively decaying with the number of faulty PE's.

The paper is organized as follows: in Section 2 an outline of the quad-tree algorithm is described as applied to a typical problem in the three-dimensional imaging technology, namely the evaluation of the disparity field associated with a stereoscopic image pair. In Section 3 a pyramidal architecture is proposed to support the quad-tree algorithm, and a description is given for the functions performed by the Processing Elements (PE's), as well as their communication paths; moreover, the performance of such basic system is evaluated by means of the quality of the output dense disparity map (DM) images. This will be used as a reference means to assess the performance of the system in the presence of faults, which is carried out in Section 4. Finally, in Section 5 some future research developments will be anticipated.

2. Quad-Tree Algorithm

The two side-images of a stereo-pair are representations of the same scene captured from slightly different view-points, so that a parallax is generated for each real point projected onto the two image planes (Fig. 1). The horizontal displacement between the two pixels on the two images is referred to as "disparity", and the two stereoscopically coupled points are termed "correspondent". It can be shown that the disparity can take positive, null or negative values depending on the position of the shot point relative to the focal plane of the shooting system [5]; this, in turn, will affect the sensation of depth given at the display level, where objects are reproduced between the observer and the screen, on the screen plane or behind the screen, respectively. In Fig. 2 the case is depicted of the reproduction of a real point P by means of its projections P_L, P_R onto the image planes to yield the virtual point V.

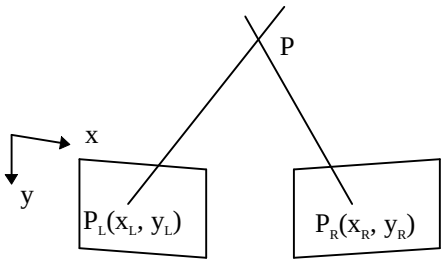


Fig. 1 - Generation of the stereo-pair: (horizontal) disparity: $d_h = x_R - x_L$; vertical disparity: $d_v = y_R - y_L$.

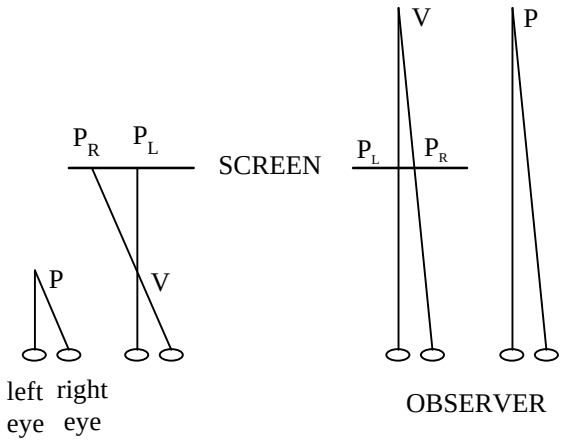


Fig. 2 - Stereoscopic image representation.

The association of a disparity value with each pixel in one image of the stereo-pair defines a disparity field, or map, which of course takes the same dimensions as the images themselves of the stereo-pair. A disparity map (DM) can be graphically represented as an artificial intensity image where the disparity is represented by the gray level. With this position, objects off the pick-up devices appear darker (of course independently of their actual luminance values) than nearer objects, which will appear brighter instead.

2.1. Hierarchical scheme

As well known, a Quad-Tree approach to the motion/disparity estimation of stereo image pairs acts at different image resolution levels, according to a hierarchical model [1, 2, 3, 4]. At each resolution level the algorithm processes a suitably decimated (sub-sampled) version of the original stereo-pairs in order to perform a local disparity estimation, also taking into account the estimation processed at the lower resolution level (if any). The number of different resolution levels can be optimized according to both subjective (visual) quality assessment and algorithm processing load.

We have adopted a hierarchical scheme to handle multiresolution versions of the input images, in order to improve the efficiency of the matching process addressed by the algorithm [1]. The basic structure of the algorithm is reported in Fig. 3, taking in input one image stereo-pair (left and right images) and producing its DM, by two basic functional blocks: a bank of low-pass anti-alias FIR filters (LPF_k) [10, 11] and a cross-correlation processor (CORR_k) array [10, 11]. The filters are fed in parallel by a pre-processed version of the input images, whose luminances are compensated for possible gain unbalances of the pick-up devices. In order to produce different resolution versions of the recovered left (L_k) and right (R_k) images a sub-sampling device has been connected in tandem with each filter block. The cross-correlation processors take the function of computing a number of correlations for each image pixel according to the algorithm specifications.

2.2. Area-based matching

The correspondence problem is one of the most critical tasks in 3DTV imaging technology, in that it is not possible in the general case to associate a point in the right image stereoscopically matched with every point in the left one; this is due to several reasons, dominated by the presence of occluded parts which appear in one image only of the two, because of the different perspectives of the two pick-up sensors. When the automation of the correspondence problem is attempted, other phenomena such as camera noise, camera misalignments and different gains, make the matching process critical. For example, a non-perfect vertical alignment of the two cameras may be responsible for a relative vertical shift of the two images, thus producing a vertical component for the disparity vector (it can be shown that in an ideal pick-up geometry the vertical disparity is null).

An estimation of the best candidate as a correspondent pixel in the left image $P_L(x+dx, y+dy)$ to match a given pixel in the right image $P_R(x,y)$ can be obtained by a block-matching process, which is a well established component of the standard coding schemes for digital video signals. This is based on the optimization of a cost function representing the similarities between a (fixed) block of pixels centered on $P_R(x,y)$ and a set of blocks of pixels suitably selected within a search area in the left image (Fig. 4): the point $P_L(x+dx, y+dy)$ is selected as the central point within the block in the set belonging to the left image which is most similar to the reference block in the right image. Similarity between such image (usually square) blocks can effectively be measured by a suitable two-dimensional cross-correlation function. Various cross-correlation functions have been experimented in this work, trading different levels of accuracy for the hardware complexity involved. A reasonable trade-off has experimentally proved to be given by a Mean Absolute Difference (MAD) expression:

$$c(m, n) = \Sigma_{h, k} \text{abs} (P_R(x+h, y+k) - P_L(x+m+h, y+n+k)) \quad (1)$$

where indexes are let to take all values in the following ranges:

$h=-B_H..+B_H$; $k=-B_V..+B_V$;
$m=-S_H..+S_H$; $n=-S_V..+S_V$;

assuming linear dimensions (width, height) equal to $2B_H+1$, $2B_V+1$ for the blocks and equal to $2S_H+1$, $2S_V+1$ for the search area. The disparity vector (dx, dy) is evaluated as the (m, n) pair such that (1) is minimal:

$$c(dx, dy) = \min \{c(m, n)\} \; m=-S_H..+S_H; \; n=-S_V..+S_V \quad (2)$$

Compared to more complex cross-correlation functions [11], this choice dramatically reduces the complexity of an embedded system implementation.

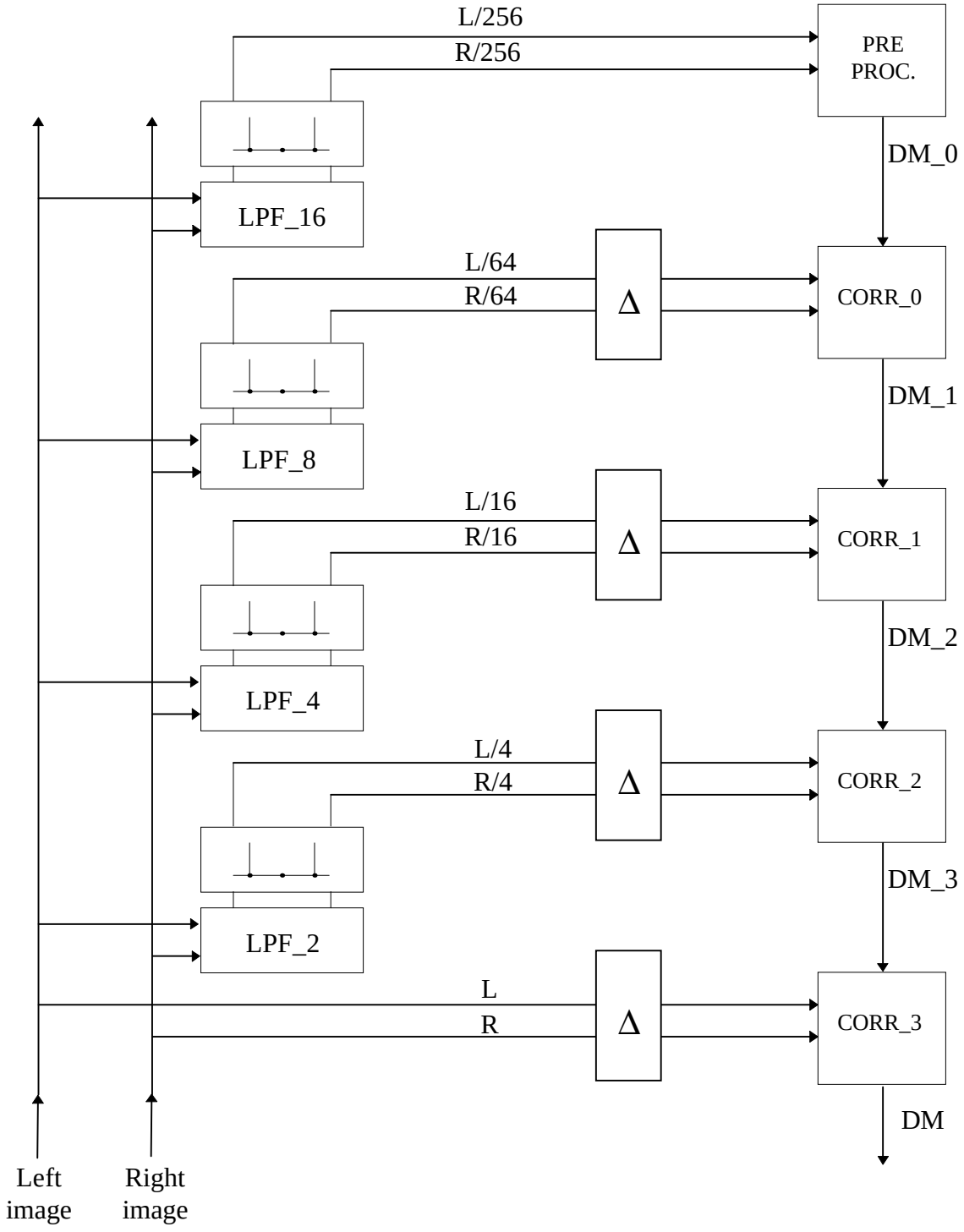


Fig. 3 - Basic structure of the quad-tree algorithm.

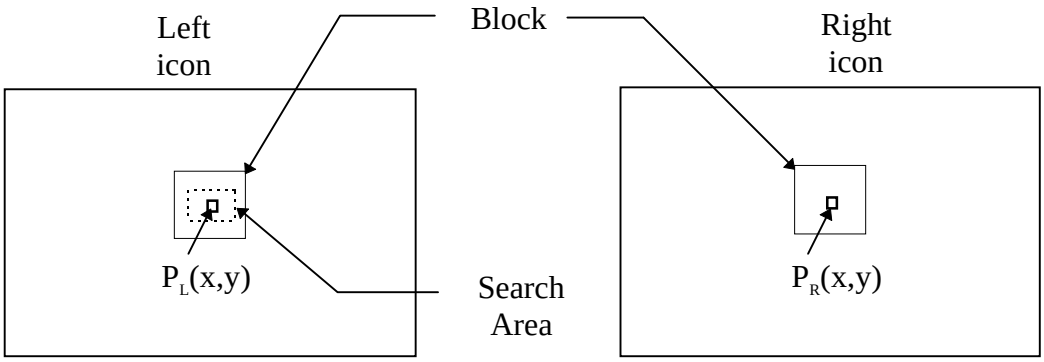


Fig. 4 - Geometrical parameters for the disparity vectors estimation at the pre-processing level. Block size: 9 x 9 points; search area range: ± 1 x ± 4 points.

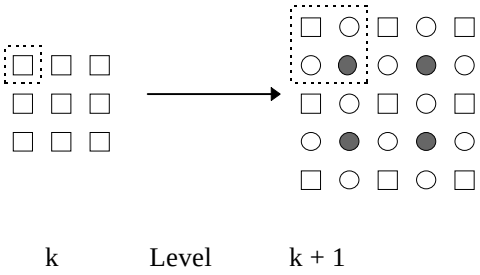


Fig. 5 - Quad-tree expansion of each pixel into a square block of 2x2 pixels.

The pre-processor module is fed by two iconical representations (45x36 pixels) of the original images input to the system, thus producing a first-estimation disparity map (also 45x36 pixels) for the benefit of the next level (processor CORR_0). Each element of such (iconical) DM is a disparity vector evaluated by the computation of the cross-correlation function between the reference block on the right icon and all the blocks placed on a search area within the left icon (Fig. 4). According to experimentation outcomes, a reasonable trade-off between disparity vector accuracy and computational load has been obtained by a 15x15 pixels square block and a search area range of ± 1 x ± 4 points. It should be noticed that the latter corresponds to a search area of ± 16 x ± 64 points at the highest resolution level of the images, thus meeting the requirements on the displacements of corresponding pixels due to a stereoscopic parallax [12]. With such positions, each pixel of the right icon need be matched with 27 (i.e. 3x9) pixels of the left icon by the selected cross-correlation function.

While the pre-processor module provides the first (most coarse) estimation of an iconical DM, within next levels, the computation of the disparity vectors is made according to the estimation performed at the previous level, thus yielding DM's with increasing resolution with the processing levels. In Fig. 5 the principle is outlined of the quad-tree expansion of each pixel into a square block of 2x2 pixels.

In our algorithm such an expansion has been designed to support some degree of computational parallelization, by the following provisions:

1) Pixels heired from the previous estimations (represented as squares in Fig. 5) can be updated independently;
2) Pixels to be generated for the first time at a given level (represented as circles in Fig. 5) are classified into two types:
   - pixels located on a new lattice shifted one position right-downwards (represented as grey circles in Fig.

5); such pixels are set in a central position to four heired pixels;
- pixels set on intermediate rows/columns (represented as light circles in Fig. 5).

With such positions, the four pixels are produced at each level according to the following steps (Fig. 6):

step 1: update disparity map of lower grid points;

step 2: estimate disparity of central points, and new columns/rows points.

The update step is performed according to the scheme reported in Fig. 7 (top), where a set of blocks of 15x15 pixels are cross-correlated with the block centered upon the pixel to estimate, so that their central shift across a 3x3 square window.

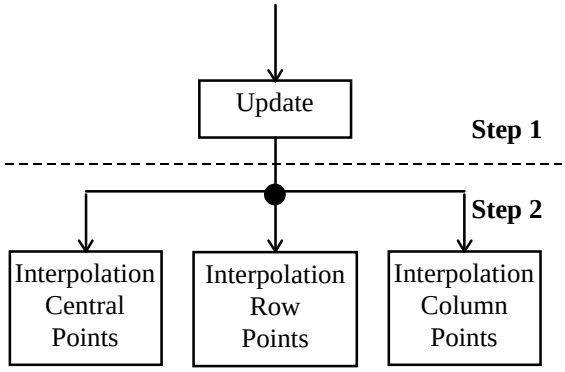


Fig. 6 - Time-sequencing of computation steps of the algorithm.

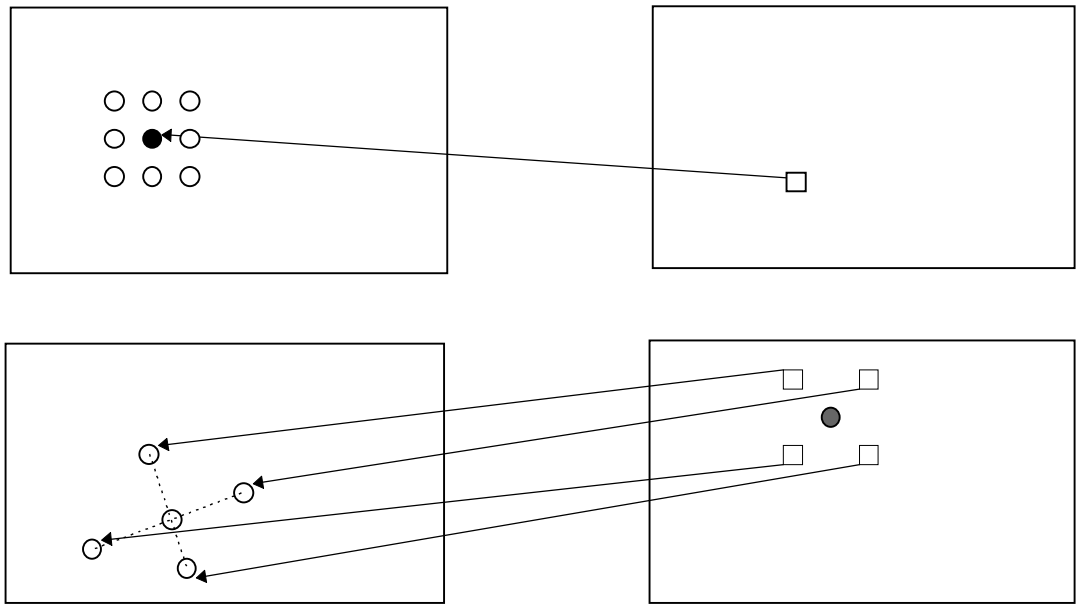


Fig. 7 - Image domain for the update of heired pixels (top) and the production of centered pixels (bottom).

$\mathbf{h}_0$	0.63528158	$\mathbf{h}_4$	0.58646997 E -1	$\mathbf{h}_8$	0.17655253 E -1	$\mathbf{h}_{12}$	0.37617773 E -2
$\mathbf{h}_1$	- 0.20802693	$\mathbf{h}_5$	- 0.43450143 E -1	$\mathbf{h}_9$	- 0.12703447 E -1	$\mathbf{h}_{13}$	- 0.21157524 E -2
$\mathbf{h}_2$	0.12030548	$\mathbf{h}_6$	0.32435280 E -1	$\mathbf{h}_{10}$	0.87930853 E -2	$\mathbf{h}_{14}$	0.10766329 E -2
$\mathbf{h}_3$	- 0.81213146 E -1	$\mathbf{h}_7$	- 0.24119453 E -1	$\mathbf{h}_{11}$	- 0.58282715 E -2	$\mathbf{h}_{15}$	- 0.49893671 E -3

Tab. 1 - Coefficients of the basic filtering function impulse response h(n).

Filter	Filtering function
LPF_2	$h_2(n) = h(n)$
LPF_4	$h_4(n) = h(n) * h(n)$
LPF_8	$h_8(n) = h(n) * h(n) * h(n)$
LPF_16	$h_{16}(n) = h(n) * h(n) * h(n) * h(n)$

Tab. 2 - Impulse responses of the filters LPF_k. Symbol * is used for convolution.

In Fig. 7 (bottom) it is outlined the selection of the pixels used in the vector estimations for the points in central positions. Namely, five points are selected in the left image for the cross-correlation operator, which are the four pixels pointed to by the four disparity vectors originating at the four pixels surrounding the one being considered, plus a fifth point which is evaluated as the barycenter of them. Experimentation has evidenced that the former points are mainly selected for points lying on object edges, whereas the weighted vector is selected by the algorithm for points belonging to smoothly varying regions.

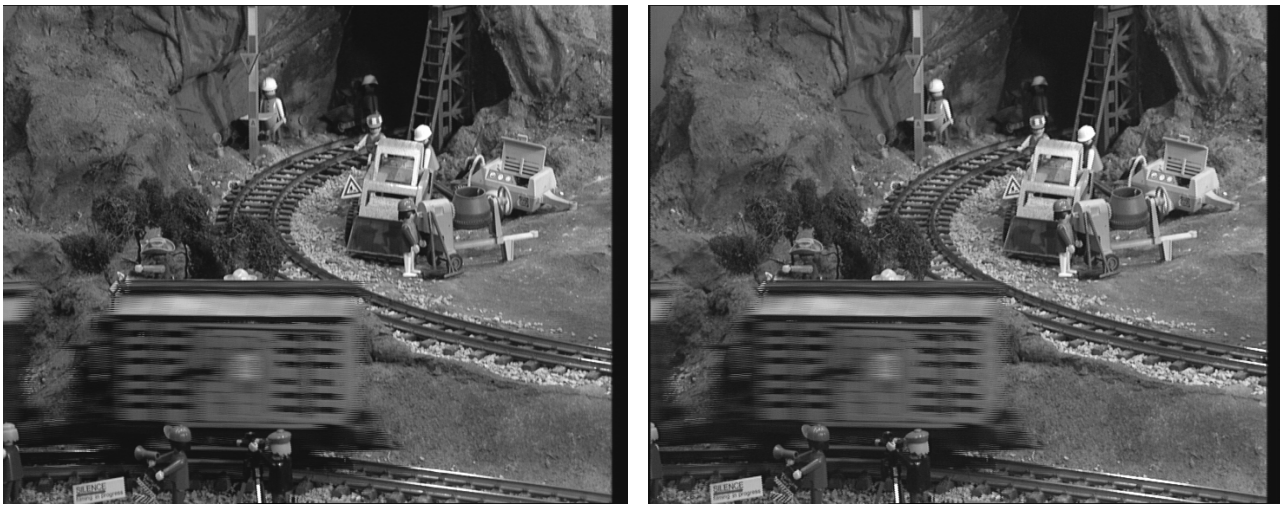
In order to protect the lower resolution images from aliasing artefacts [10] due to the decimation process, very accurate filters have been designed, which meet the specifications recommended by the digital video standard, i.e. 0.05dB in-band ripple [13]. Filter LPF_2 in Fig. 3 features a half-band transfer function, so that it produces a smoothed version of the original images; therefore these can be decimated by a factor 2 along both vertical and horizontal dimensions without introducing aliasing effects on the sub-sampled images L/4, R/4 prior to feeding the processor CORR_2. The two-dimensional transfer function has been made separable [10], so that a simpler linear filtering function may be used to process the images in two separate steps for the vertical and the horizontal directions. In Tab. 1 the 16 coefficients are listed of the resulting digital filter 32 taps symmetrical impulse response h(n).

The following stages act iteratively in such a way that a further band reduction and a decimation by 2 is introduced at each stage. Therefore, the related filtering impulse responses may be obtained by convolutions of h(n) on itself, after the relationships reported in Tab. 2.

2.3. Performance evaluation

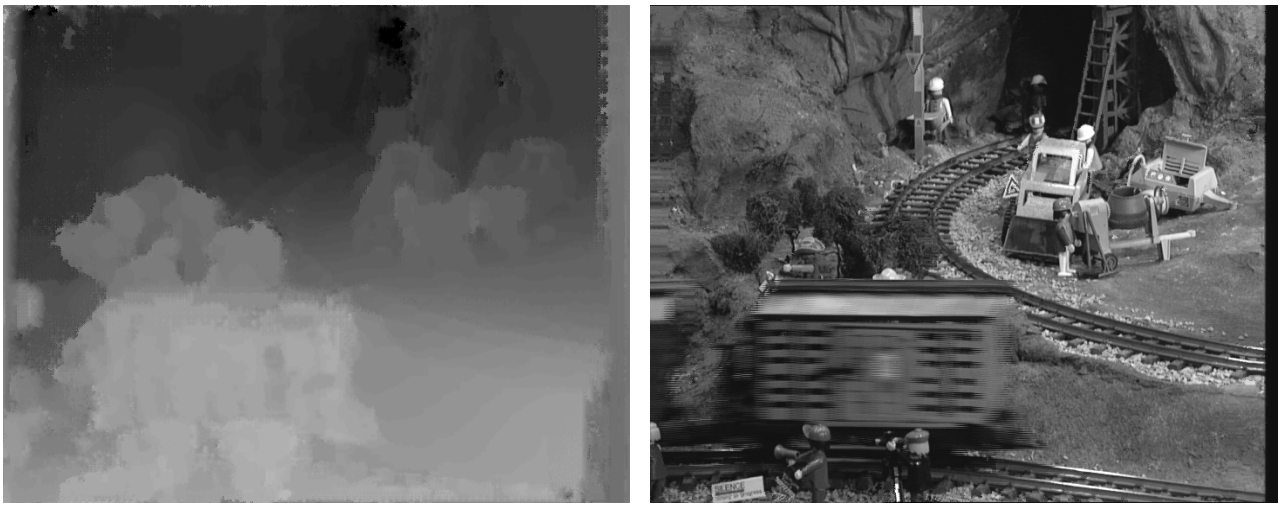
For the sake of a verification of the effectiveness of our algorithm, several stereo-pairs have been selected as test vectors from a data-base available to the scientific community for results interchange. For demonstration purposes in Fig. 8 a still stereo-pair from the stereo-sequence “Train” (courtesy of C.C.E.T.T. - F) is reported. Full resolution images are assumed to be 720 pels x 576 lines/pel, according to the European standard digital video scan format [13].

Fig. 9 represents the DM produced by our algorithm (a), as well as the right image reconstructed by such DM (b). The subjective good quality of the reconstructed image (Fig. 9) is matched by the objective measurement of the Mean Square Error (MSE), evaluated with reference to the original right image: the calculated value is 220, which is still within an acceptable range, especially considering that most error energy is concentrated on the image left and right borders (due to the occlusions generated by the parallax of the pick-up geometry), which are visually not very relevant.



(a) **(b)**

Fig. 8 - Original stereo-pair “Train”; (a): left image, (b): right image.



(a) **(b)**

Fig. 9 - (a): Disparity Map for the stereo-pair of Fig. 8; (b): reconstructed right image.

3. Architecture of the Processing system

3.1. Pyramidal scheme

The quad-tree structure is essentially oriented to a pyramidal implementation which is attractive for the real-time processing of video image signals; in fact it is scalable in its dimensions to accommodate different image resolutions, and may be designed so as to feature an equally distributed processing load on all the pyramid levels.

As appears from the algorithm description, it is possible to perform the computations onto a parallel scheme, in which the disparity values heired at a given level can be in principle be updated independently and therefore simultaneously; moreover also the points set on the central shifted lattice can be interpolated simultaneously, as can the points defining new rows and columns (Fig. 6). This suggests that the computation steps of the quad-tree algorithm can be partitioned on a hierarchical architecture supporting the various levels of resolution of the stereo-pair and DM images [12].

The proposed computing system is derived to support the iterative structure of the algorithm in a pipeline/parallel organization, thus closely resembling the structure of Fig. 3. In this work it is proposed that each correlation processor is implemented by a regular replica in space of an elementary Processing Element (PE) charged with the basic functions above identified. Moreover, in order to cope with the high data-rates involved by real-time video signals, a uniform computation workload should be delivered to each level of the pyramidal structure.

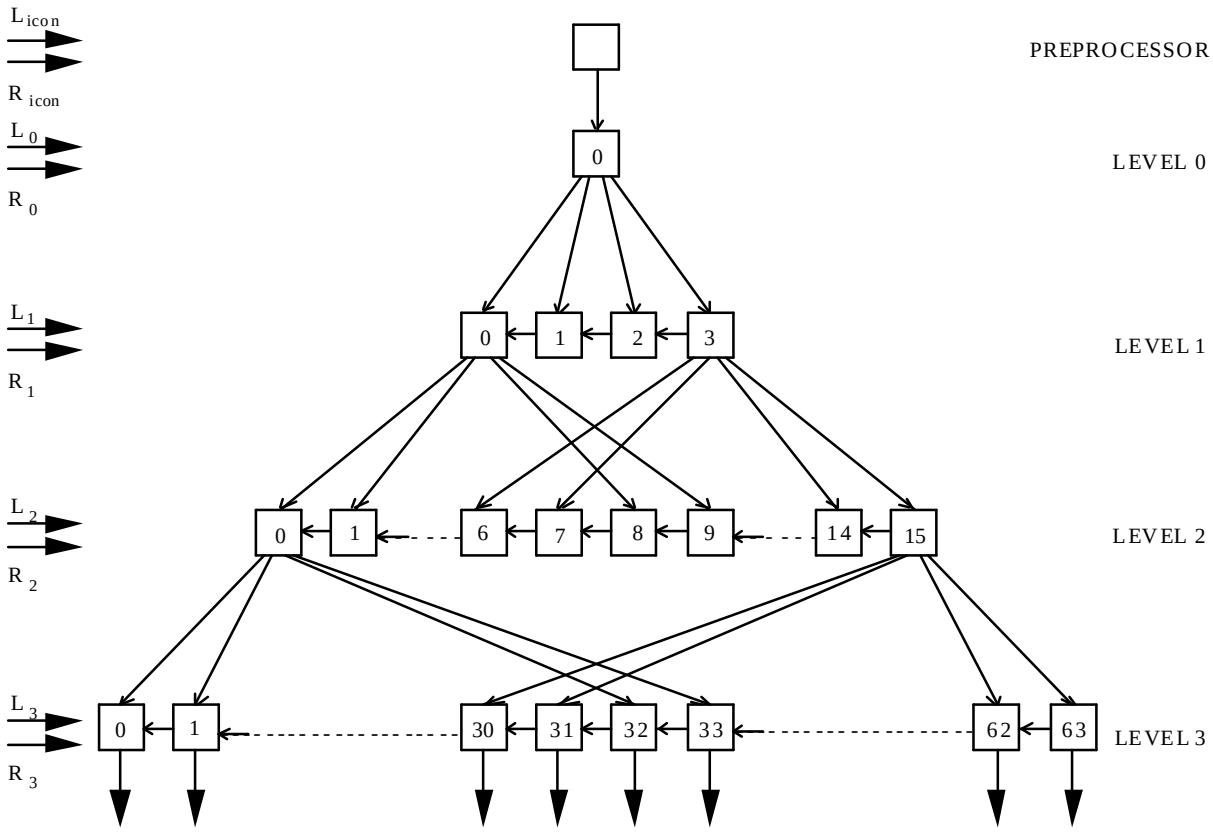


Fig. 10 - Pyramidal computing structure.

Due to the different image resolutions or pixel quantities handled at each level, the resulting structure accommodates increasingly larger linear arrays of PE's with the level depth of the tree. The deriving pyramidal structure is reported in Fig. 10, which builds on five levels: the first one represents the preprocessor, which is appended to a four-level quad-tree, i.e. connecting its root (level zero) to four nodes on level 1, also having sixteen nodes on level 2 and sixty-four nodes on level 3.

It can be observed that since all PE's perform the same computation on different data sets, such a structure classifies as a Single Program Multiple Data (SPMD) scheme. In fact, within each node computations include the estimation of the minimum of a cost function, whose execution of course is data dependent, and so its full evaluation can be stopped as soon as partial results exceed the current minimum candidate.

3.2. PE's communications

In our algorithm implementation, PE's are interconnected so that the disparity vectors evaluated at a level I are delivered to PE's on the level I+1 after the arrangement shown in Fig. 10. In particular, it is supposed that each PE receives one disparity vector at a time from one PE at the lower level, and it produces the four vectors after the arrangement of Fig. 5 (one vector updated plus three new estimated). Such vectors are vertically arranged within the PE into two pairs, which are then passed to two adjacent PE's on the next level, as shown in Fig. 11. Since four PE's are available at level I+1 for each PE at level I, the next group of four vectors produced by the PE at level I will be delivered to a different pair of PE's on level I+1. Due to the quad-tree structure, such a PE pair is in a position off the pair involved previously by half the number of PE's available at the level I+1 (Fig. 11).

From Fig. 11 it is apparent that $Pe_{i,j}$ has in charge the processing of all the vectors located on columns k of the Dm_i (see Fig. 3) such that: k mod 4^i = j. For example, $PE_{3,0}$ (the leftmost PE on the output level of the tree in Fig. 3) handles all the vectors of DM_3 (containing 360 columns x 288 rows of pixels) lying on columns 0, 64, 128, 256, 320. Moreover, according to our algorithm, to produce new vectors each PE needs to know the vectors updated by its right hand side neighbour only. Such an information flow is supported by the direct right-to-left interconnections shown in Fig. 10 and Fig. 11.

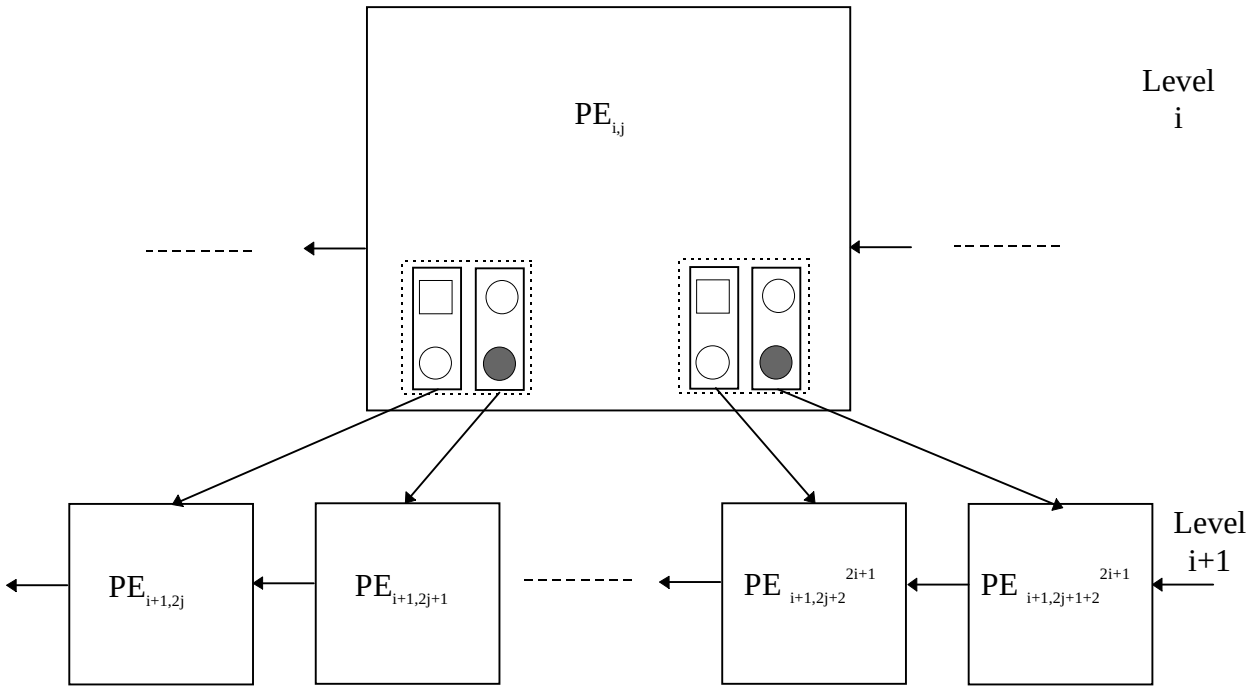


Fig. 11 - Inter-PE disparity vector communication.

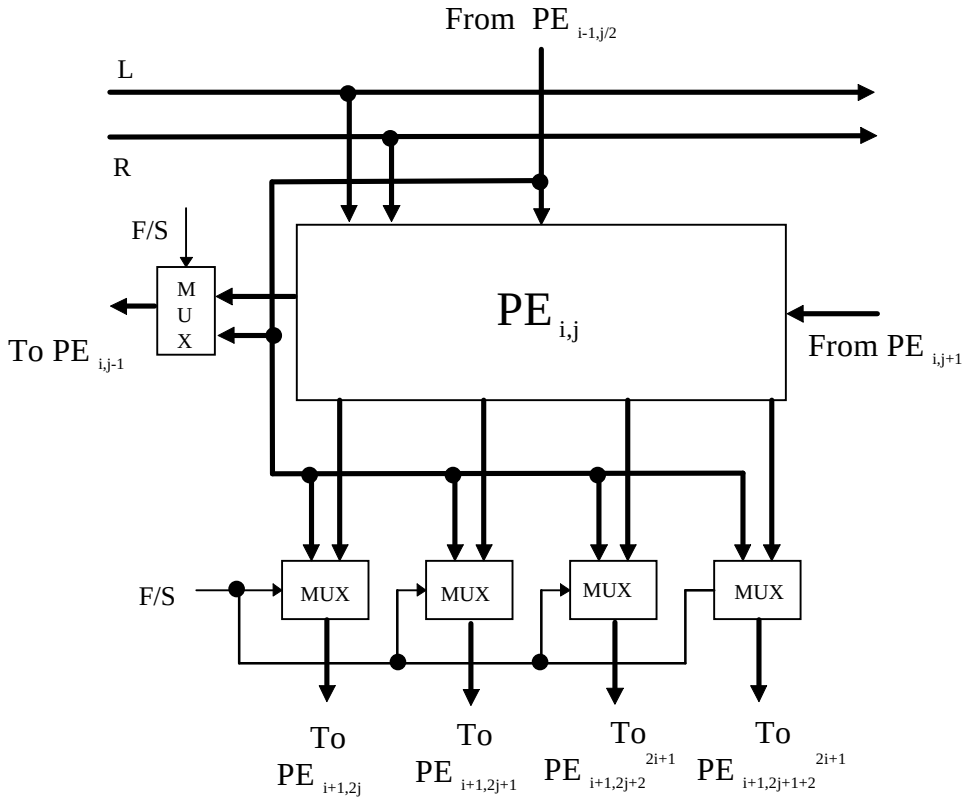


Fig. 12 - PE enhancement for data de-routing support.

The proposed tree organization offers such relevant advantages as:
- simple direct one-way interconnections between adjacent PE's are needed. This highly contributes to making our structure suitable for a VLSI implementation;
- any misbehaviour of the system due to faults on PE's results in the confinement of errors on regularly spaced vertical stripes on the DM. Moreover in this case the width of the noisy stripes is correlated with the level of the faulty PE; this consideration lays the basis for the fault-tolerance analysis discussed in the next sections.

4. Fault-tolerant characteristics

The fault-robustness characteristics of a DM computing support can be evaluated by comparing the images produced by it with and without faults. In our application, the fault-robustness can be qualified by both visual subjective assessment and objective measurement in terms of MSE. A set of experiments has therefore been carried out in order to detect the sensitiveness of our computing structure to the faults. Namely we investigated how a fault degrades the performances of the images versus both its function (filter module, preprocessor, PE) and position. This preliminary study has driven the placement of the redundancies, if necessary. In the following we first analyse what happens in case of single fault and then in case of multiple faults.

As discussed, the proposed architecture is composed of three subsystems: the bank of the filters, the DM preprocessor and the cross-correlation tree. The bank of filters is composed only of four components, and in case of single fault in the bank filter we found, by simulation experiments, that the data de-routing techniques realizes an acceptable performance degradation [12]. Given the relatively low complexity of the filters and the presence of only four components in the bank, the hypothesis of only one fault in the filter bank is not a bad assumption; however redundancy can easily be incorporated if more than one fault are expected. More details can be found in [12].

From the analysis of the data relating to the cases of single faults, it appears that a fault on the DM preprocessor prevents the algorithm from being correctly

started. Therefore it is essential to introduce spares to make as high as possible the reliability of that stage.

The presence of 85 cross-correlation processing elements (PE's) makes difficult to add massive redundancies, so we have to investigate the sensitiveness of our system to multiple fault occurrences. In a previous work [12] we found that data de-routing techniques can be very effective to recover faults on cross-correlation processors. In that paper the tasks of cross-correlation were performed by a single processor per level, while in this case the cross-correlation activities are executed by 4^i specialized processing elements at the i-th level. Therefore we have introduced some local resources to support the data de-routing in the form of multiplexers, as outlined in Fig. 12. Here it is shown how a faulty PE can be bypassed and the incoming data are driven to the right-hand side neighbour.

4.1. Fault analysis with de-routing

In Tab. 3 the MSE figure is reported for the output DM's (with reference to the DM produced by the fault-free system) output by the system affected by various cases of a single fault on the tree.

The position of the faulty PE is identified by the two indeces (i,j): the first index identifies the level and the second one the position of the PE within that level. For example, the first (leftmost) PE in the level 3 is identified by the pair i = 3 and j = 0, instead the last (rightmost) PE in the level 2 by the pair i = 2 and j = 15.

From Tab. 3 it is possible to see that the MSE is very high in case of the PE at level 0, whereas it is still acceptable in the case of a faulty PE at level 1 and it is very low in case of a faulty PE at level 2 or 3. However, we found a value near 40 for the MSE in case of presence of two faulty PE's at level 1; therefore we use spares to tolerate faults not only in the DM preprocessor, but also at levels 0 and 1. This choice does not involve a high hardware investment considering that the number of PE's at level 0 is just one and at level 1 it is four only, i.e. very low compared to the eighty PE's at the levels 2 and 3. Conversely, due to opposite reasons, i.e. low MSE figures and high connectivity, we propose to provide PE's at the levels 2 and 3 with de-routing mechanisms only. Hence in the following section we will focus on the performance degradation of our structure for the cases of multiple faults on levels 2 and 3.

4.2. Fault superposition of effects

In principle, to understand the fault-robustness of the proposed architecture we need to know the performance degradation for all possible fault patterns localized in the levels 2 and 3: i.e. $2^{16} \times 2^{64}$, thus making such a study unfeasible. For this reason it is necessary to investigate if there are some characteristic fault patterns that can be used to identify the expected performance degradation of the system given a generic fault pattern (FP). To this purpose, we first study the performance degradation due to FP's localised within the same level and then we analyse the performance degradation due to FP's affecting both levels.

Faulty PE	MSE	Faulty PE's	MSE
(0,0)	187	(2,6)	2.77
(1,0)	21.44	(2,15)	3.72
(1,2)	23.64	(3,0)	0.47
(1,3)	19.15	(3,7)	0.42
(2,0)	2.87	(3,8)	0.49
(2,2)	2.97	(3,9)	0.40
(2,3)	3.36	(3,10)	0.43
(2,4)	2.44	(3,12)	0.50

Tab. 3: MSE in case of single faults.

FP	Faulty PE's	MSE
FP1	(2,2) (2,3)	5.53
FP2	(2,0) (2,3)	6.23
FP3	(2,2) (2,3) (2,4) (2,6)	10.11
FP4	(2,2) (2,4) (2,6) (2,15)	11.90
FP5	**All** PE's of level 2	44.95
FP6	(3,7) (3,8)	0.91
FP7	(3,0) (3,12)	0.97
FP8	(3,7) (3,8) (3,9) (3,10)	1.44
FP9	(3,0) (3,7) (3,9) (3,12)	1.79
FP10	**All** PE's of level 3	28.19

Tab. 4 - MSE in case of multiple faults in the same level.

FP	Faulty PE's	MSE
FP11	(2,2) (3,0) (3,7) (3,10) (3,12)	4.79
FP12	(2,0) (3,0) (3,1) (3,32) (3,33)	6.24
FP13	(2,1) (3,2) (3,3) (3,34) (3,35)	6.76
FP14	(2,4) ((3,8) (3,9) (3,40) (3,41)	5.75
FP15	(2,0) (2,1) (3,0) (3,1) (3,2) (3,3) (3,32) (3,33) (3,34) (3,35)	11.50
FP16	(2,0) (2,4) (3,0) (3,1) (3,8) (3,9) (3,32) (3,33) (3,40) (3,41)	11.99

Tab. 5 - MSE in case of multiple faults in the level 2 and 3.

In Tab. 4 the MSE's are reported corresponding to some fault patterns involving multiple faults at the same level. For the sake of conciseness this table reports some significant FP's, which allow us to characterize the effect of the faults in the same level. From data of Tab. 4 it is possible to note that the MSE figures pertaining to FP2, FP4, FP7 and FP9 equal the sum of the MSE's related to the single faults. For example, the MSE associated with FP4 is equal to 11.90, and it is equal to the sum of the MSE's related to PE(2,2), PE(2,4), PE(2,6) and PE(2,15). Instead, the MSE's associated with FP1, FP3, FP5, FP6, FP8 and FP10 are less than the sums of the MSE's related to the single faulty PE's in the cluster. This different behavior is due to the fact that in the first case the PE's process data that are independent, instead in the second case there is a correlation between the data processed by the PE's. So in the first case there is a superposition of the fault effects (denoted as *horizontal superposition* in the sequel), whereas in the second case there is an intereference between the noisy subtrees, which reduces the number of the image columns affected by the noise with respect to the case of data independence. In this case therefore an *upper bound* can be formally found for the noisy image columns; nevertheless, as far as the MSE is concerned, experimental results have shown that such a property is *de facto* extended to the MSE.

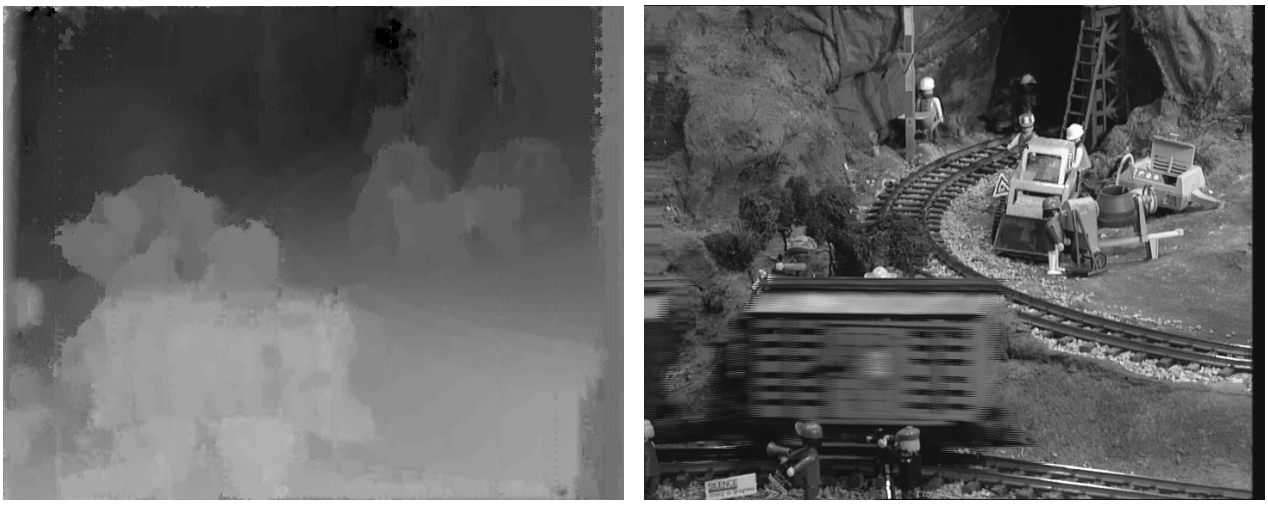
To confirm the validity of our choice to provide levels 2 and 3 with de-routing resources only, the two further cases have been reported in Tab. 4 of all faulty PE's in each of those levels (FP5, FP10). In both cases it can be appreciated that still acceptable figures are obtained.

In the sequel we study the effect of multiple faults when the faulty PE's are located contemporaneously in the levels 2 and 3. Also in this case in Tab. 5 the FP's have been included that permit us to characterize the presence of multiple faults in different levels.

Tab. 5 shows the MSE's for six FP's. Namely, FP11 is characterized by PE's that process indepedent data. FP12, FP13 and FP14 are characterized by a set of PE's in which there is a dependency between the data produced by the PE at the level 2 and those received by the PE's at level 3. FP15 is the union of FP12 and FP13. FP16 is the union of FP12 and FP14.

FP11 exhibits an MSE that is exactly equal to the sum of the MSE's of the FP's characterized by a single fault, i.e. PE(2,2), PE(3,0), PE(3,7), PE(3,10) and PE(3,12); instead, FP12, FP13 and FP14 produce MSE's that are greater than the sums of the MSE's of the FP's involving a single PE. Therefore also in this case there is a superposition of fault effects when the faulty PE's process independent data (denoted as *vertical superposition* in the sequel); instead, in this case, when there is a data dependency between faulty PE's of different level there is an amplification of the effects. However, we can note that FP15 (that is the union of the FP12 and FP13) presents an MSE that is less than the sum of the MSE's generated by FP12 and FP13; also the FP16 has an MSE that is equal to the sum of the MSE's generated by FP12 and FP14. Therefore there is a superposition of the fault effects depending on the data processed by a cluster of faulty PE's being independent on the data processed in the other cluster (denoted as *subtree superposition* in the sequel). Instead, in case of data dependency between faulty clusters there is only a partial superposition of the effects, much the same as the case discussed above for the horizontal superposition.

From data of Tab. 3, Tab. 4 and Tab. 5 we can conclude that the horizontal, vertical and subtree superpositions of the effects on the data corruption is verified only if the faulty clusters process independent data, however, if there is some data dependency between faulty clusters then the superposition of the effects is experimentally conservative. So to characterize the possible $2^{16} \times 2^{64}$ FP's it is enough to know the MSE's associated with only 96 FP's: of these 16 FP's involve a single faulty PE at level 2, 64 FP's involve a single faulty PE at level 3 and finally 16 faulty clusters involve a single faulty PE at level 2 and four faulty PE's at level 3 that are data dependent on the faulty PE at level 2.



(a) **(b)**

Fig. 13 - (a): Disparity Map produced by the FP (2,0) (3,30); (b): reconstructed right image.

Moreover from Tab. 5 it is possible to see that even in the case of ten faulty PE's of which two are located at level 2 and other eight at level 3, the MSE is still less than 12. Given the superpositions of the effects on data corruption, in the case of four faulty PE's at level 2 and sixteen faulty PE's at level 3 (corresponding to 25% faulty elements on the last two levels), the expected MSE can be evaluated as low as about 24; this allows to conclude that our structure more generally performs a high robustness to random faults. To give a visual insight of the adequate performance of our system, in Fig. 13 we report (a) the disparity map produced by the FP (2,0) (3,30), and (b) the reconstructed right image.

5. Concluding remarks

In this work a Quad-Tree algorithm has been specialized for the evaluation of the disparity map (DM) of 3D-TV images, which is attractive for the depth estimation of objects within images related to industrial processes as well as multimedia applications. A multi-layer hierarchical scheme has been adopted to handle multiresolution versions of the input image stereo-pairs so as to improve the efficiency of the matching process involved in the disparity estimation for each pixel of the original images. To cope with the critical tasks potentially requested to an embedded system implementation, a fault-tolerant SPMD hierarchical architecture has been introduced for the PE's (processing elements) on the tree nodes to support the iterative structure of the algorithm in a pipeline/parallel organization, which is attractive for the real time processing of video signals.

The basic idea in the design of this computing system is to tolerate faults by the selective provision of spare redundancies and de-routing resources, with the target of letting the system degrade the quality of the processed images gracefully with the number of faults. The placement of the redundancies as well as of the data de-routing resources to tackle faults has been driven by simulation results: several stereoscopic image pairs have been used as test patterns to qualify our system by both an objective measurement (MSE) and a subjective assessment (output images) in the presence of faults with respect to a faultless behaviour. Experimental outcomes confirm that any misbehaviour of the system due to faults on PE's results in the confinement of errors on regularly spaced vertical stripes on the output images. This architecture property allowed us to identify a set of characteristic fault patterns which allow us to predict the performance evaluation of the computing structure in the assumption of random fault configurations by the combination of the superposition of effects and upper bound estimations.

Further work has been started to provide the structure with on-line self-test capabilities. This includes the concept of the insertion of test vectors within the vertical blank interval of the video signals to periodically stimulate the PE's hardware. Additional work shall address the granularity of the parallel architecture, that will strongly affect the processor VLSI implementation.

References

[1] L. Alparone, F. Argenti, V. Cappellini, “A robust coarse-to-fine least squares stereo matching”, 4-th European Workshop on 3DTV, Oct. 1993, Rome (I).

[2] V. Cappellini et al., “Digital processing of stereo images and 3-D reconstruction techniques”, International Journal of Remote Sensing, Vol. 12, No. 3, 1991.

[3] Marr, D., “Vision”, Freeman, New York, 1982.

[4] Rosenfeld A. et al. “Digital Picture Processing”, Vol. 2, Academic Press, 1982.

[5] Horn B.K.P.“Robot Vision”, Mc Graw-Hill, 1986.

[6] M. Ziegler, et al. “An object-based stereoscopic coder”, Proc. of the Int. Workshop on Stereoscopic and Three-Dimensional Imaging”, Sept. 1995, Santorini (GR).

[7] ISO/IEC/JTC1/SC29 Doc. MPEG 93/254, “Stereoscopic Video Transmission: A Future Application of Spatial Embedded Coding”, March 1993.

[8] Tseng, B., Anastassiou, D., “Perceptual Adaptive Quantization of Stereoscopic Video Coding Using MPEG’s Temporal Scalability Structure”, Proc. of the Int. Workshop on Stereoscopic and Three-Dimensional Imaging”, Sept. 1995, Santorini (GR).

[9] D. Tsovaras, et al., “Depth Map Coding for Stereo and Multiview Image Sequence Transmission”, Proc. of the Int. Workshop on Stereoscopic and Three-Dimensional Imaging”, Sept. 1995, Santorini (GR).

[10] Oppenheim, A.V., Schafer, R.W., “Digital Signal Processing”, Prentice Hall, Englewood Cliffs 1975.

[11] Pratt, W.K., “Digital Image Processing”, Wiley, New York, 1978.

[12] M. Cappiello, A. Chiari, B. Ciciani, “A Graceful Degradation Embedded Computing System for the Disparity Evaluation of Stereoscopic Video Signals”, Proc. of the IEEE International Workshop on Embedded Fault-Tolerant Systems, Sept. 1996, Dallas, Texas (USA).

[13] ITU-R, Rec. 601, “Encoding parameters of digital television for studios”, 1990, Vol. XI.