Real Time Hardware Architecture for Visual Robot Navigation

F. Marino, E. Stella, N. Veneziani, A. Distante

Istituto Elaborazione Segnali ed Immagini - C.N.R.
Via Amendola 166/5, 70126 Bari, Italy
tel: +39 80 5481969 fax: +39 80 5484311
e-mail: [marino, stella, nicola, distante]@iesi.ba.cnr.it

Abstract

A specialized hardware architecture to permit a real time visual navigation is proposed. The navigation is performed by a two-stage approach to extract visual features and to match them over an image sequence acquired during the mobile robot motion in order to estimate motion parameters. The paper describes a hardware implementation of the first stage (the burdensome stage) of the method for egomotion parameter computation. The hardware performance permits a processing rate of 40 Mhz.

1. Introduction

Passive navigation is the ability of an autonomous agent to determine its motion respect to the environment. In order to guide a vehicle, some information about its motion must be estimated. Though the required information can be obtained using odometers and gyros (but with an unbounded incremental error) the task can be performed by visual sensors with a lower uncertainty. In [1,2] a two-steps algorithm to estimate the heading direction evaluating the displacement vector field on two successive images of the scene is shown. In fact, in the context of passive navigation, the main goal for recovering egomotion parameters, is efficiently solved by analyzing a displacement vector field where the correspondences between 2D features extracted in successive images of a sequence and corresponding to the same 3D feature in the space are represented. A small number of such displacement vectors on the image plane is sufficient to obtain useful information on egomotion parameters. In literature, two frameworks seem to approach the matching problem: direct and optimization methods. Both frameworks consider a low level stage in which features are extracted from images. Then, direct methods use local constraints on features in order to find correspondences [3,4,5]. The optimization methods use global constraints on features to formulate an energy or cost function and the correspondences are found by minimization of that functional, generally, using iterative techniques.

While, the direct methods are fast but more sensitive to noise; the optimization based techniques are more reliable but have the drawback to require a burdensome processing. In [2] the authors propose a feature-based approach to solve the correspondence problem by minimizing an appropriate energy function where constraints on radiometric similarity and projective geometric invariance of coplanar points are defined. The method consists of three different steps: 1) a low-level step in which features are selected and matched by correlation; 2) a verification step in which

matched features are verified by minimizing a cost function; 3) a determination step in which egomotion paramete are estimated.
An analysis of technique performances has shown as most of processing time is lost in the first step, so, in this paper we propose a novel specialized hardware to speed-up the feature extraction and the correlation stage in order to permit fast mobile robot navigation.
In literature, some other hardware implementation to resolve the matching problem can be found [6].
In section 2 a detailed description of matching technique is given; in section 3 the hardware is described, while in section 4 performance evaluations are shown.

2. Matching technique overview

Features correspond to points having a high directional variances in intensity images of the scene. We use the interest operator introduced by Moravec to isolate N points with minimal autocorrelation values [1]. The variance among neighboring pixels in four directions (vertical, horizontal and two diagonal) is computed over a window (the smallest value is called the interest operator value). In our implementation windows have a size of 7x7 pixels. Points, where the interest measure has local maxima are chosen as candidate feature. Each feature consists of a window (6x6 pixels). Initial matches for each features are computed maximizing the radiometric similarity (correlation) among features in the first image and areas in the second image. Matching features by correlation produces unavoidable false match, so matches computed by correlation can represent only an initial guess to be improved through an optimization approach. We verify the goodness of these matches and correct them imposing the cross-ratio invariance constraint.
Our experimental setup is based on our autonomous vehicle S.AU.RO, a VME 68040 based system, having a CCD camera COHU 6510 on the top. The TV camera optical axis is oriented along the forward motion direction of the vehicle. An ELTEC frame grabber digitizes image of 512x512x8 pixels. Two consecutive images acquired during the robot motion are used to estimate the heading direction the mobile robot. In fact, features are extracted on the first image and matched on the second, in order to estimate the heading of the vehicle, using the technique described above. Then, a new set of features is extracted on the second image and matched on the next acquired image. So, considering for each image (excluding the first and the last) acquired during the vehicle motion two stage must be performed: the feature extraction and the correlation based matching. The next section described a hardware implementation of these steps while the optimization stage is left to future developments.

3. Hardware architecture description

Our aim is to complete the feature extraction and correlation processing stages almost at the image acquisition rate. In the proposed architecture, these stages, because of their independence, work parallely on two different computing blocks. In fig. 1 a logical scheme of our architecture is shown.

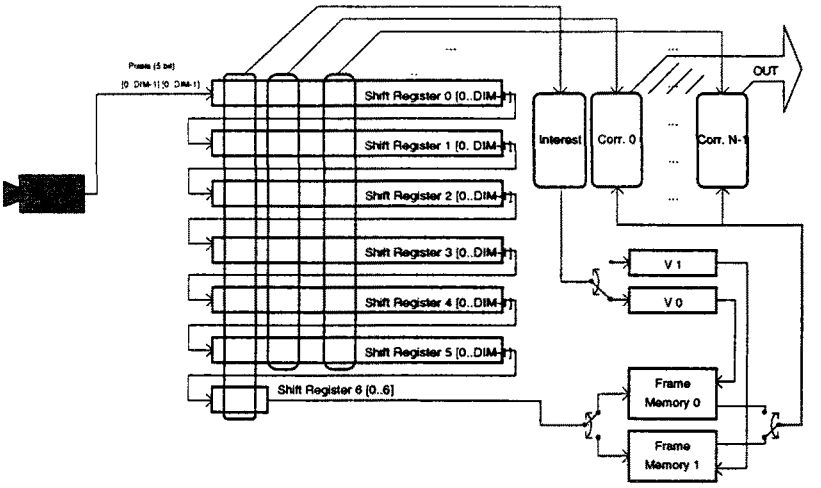


Fig. 1: Functional scheme of the proposed architecture.

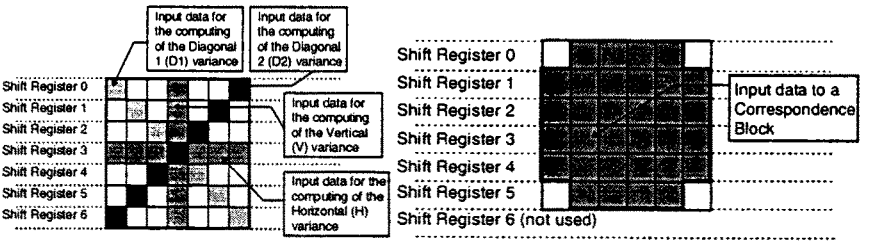
At the i^{th} step of a sequence an image (DIMxDIM) pixels is acquired. The processing, at the same time, performs:

- extraction of N features to be matched over the $(i+1)^{th}$ image (to be acquired)
- detection of the corresponding points for each extracted feature over the $(i-1)^{th}$ image.

The Computing Blocks performing these operations are respectively: the "**Interest Block**" and the "**Correspondence Block**" (fig. 1).

The image pixels, provided by an external frame grabber, flow into a pair of interleaved frame memories, passing across a pipe of 7 Shift Registers (because 7x7 is the size of area for feature extraction).The frame memories store, allternatively, images for next processing phases (correspondence stage).

The **Interest Block** computes the variances among neighboring pixels in four directions (vertical, horizontal and two diagonal) over a window of 7x7 pixels (stored in the 7 Shitf Registers). The lowest variance is selected as interest value and it is associated to the pixel in the middle of the window (fig 2.a).



a) b)

Fig. 2. a) the cells that are considered as input in the Interest Block. b) the cells that are considered as input to a Correspondence Block.

The final aim of the Interest block is to estimated N features (as described in sect. 2) so, the input image is partitioned in N regions (DIMxDIM/N pixels). For each region is estimated the local maximum among all Interest values computed on the region and the address of this maximum is stored in register V_i where i cam be 0 or 1 depending on the current reference frame memory.
The **Correspondence Block** computes the sum of absolute errors between the 6x6 window relative to the extracted feature and the data currently into the Shift Registers (fig. 2.b). At each shifting a new sum is evaluated and compared with the current minimum. When all shifts are performed (mean that the whole image has been acquired) the current minimum represents the best match. This value is the output of each Correspondence Block.

3.1 Correspondence block

In order to reduce the computational load and the hardware complexity, the image has been quantized at 5 bits. This assumption does not reduce the capability of the hardware because the features, being points having high directional variances, do not change when the image dynamic decrease.
Furthermore, environmental influences (like lightning changes or occlusions) can be considered negligible when images are processed at TV acquisition rate (50Hz), so simple correspondence operators can be adopted.
Finally, in order to reduce the number of clock pulses needed by each phase of the pipe, the Residue Number System representation [13] has been used.
The correspondence measurement used to implement the Correspondence Block is the least absolute error. So, the N Correspondence Blocks (one for each feature to be matched) have to compute the least absolute error over a 6x6 pixels window centered on the feature.

- An example of these Blocks is shown in fig. 3 and, mainly, it is composed by:
  two kernels: the first one represents the input data, extracted from the shift registers and the second one is centered on the feature *i* whose address is stored in register Vj (j=0,1 depending on the current frame memory).

- an array computing the absolute error between the two kernels, defined as:

$$E_k = |I_k - I_k| \qquad (k=0,...,31) \qquad (1)$$

  In (1) k in range [0..32] depends on an implementation choice to use a kernel 6x6 pixels without the four corner elements.

- a tree computing :

$$LSE = \sum_{k=0}^{31} E_k \qquad (2)$$

- a terminal block comparing the current LSE result with the current minimum valued (stored in register MIN (fig. 3)) in order to find the smallest one, and consequently the resulting match for the processed feature.

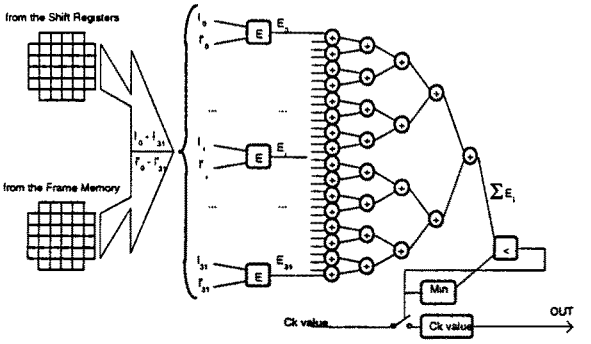


Fig. 3: the Correspondence Block.

The nodes at first level (fig. 3) operate on 5-bits data (raw image intensities), so they can be implemented using a 1k-word Look Up Table.
The nodes at next levels have to sum the previous 5-bit values (data dynamic is D =32 · 31=992) so, they can be implemented by LUTs using RNS [13].
The final LSE needs to be compared with the current minimum LSE stored in MIN, in order to find the match. The "<" block, is detailed in [13].

3.2 Interest block

A functional description of the Interest Block is shown in fig. 4.

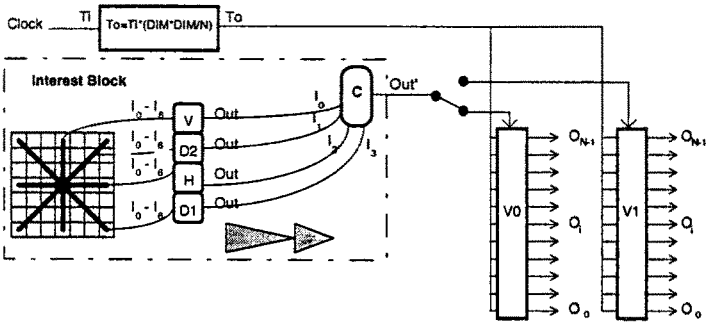


Fig. 4: Interest Block.

The more significative components of this block are:

- a kernel (on the left side) coming from the rows of the Shift Registers;
- V, H, D1 and D2 computing respectively the vertical, horizontal and two diagonal variances;
- a module of comparison C (see fig. 5a), where the variances are compared in order to find the smallest value (the Interest value) and the candidate feature is selected (">-Block").

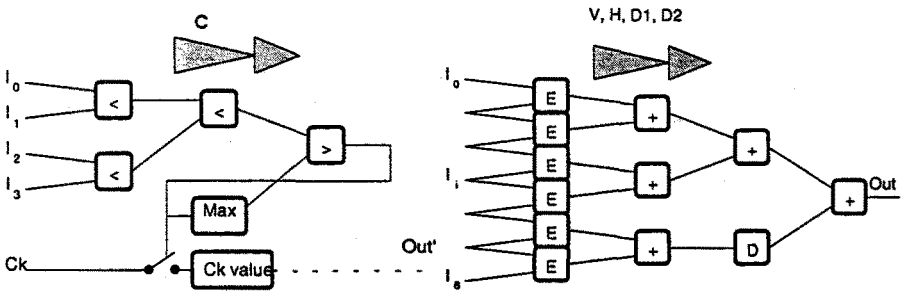


Fig. 5: a) tree of comparison: it is used to detect the Interest value and to find the maximum value identifying the feature to be extracted; b) Vertical, Horizontal and Diagonal variance computation.

V, H, D1 and D2 are four similar modules (from a computational point of view (fig. 5b) whose architectures are similar to hardware described in sect. 3.1. Each of them computes the sum $\sum_{j=0}^{5}\left|I_j - I_{j+1}\right|$, where I_j and I_{j+1} are two adjacent pixels along the same direction (vertical, horizontal or diagonal) .
Furthermore, a tree estimating the minimum value among the four results of V, H, D1 and D2 trees needs to be implemented (see fig. 5a). The "<" and ">" blocks are the same of the similar block described in sect. 3.1.

4. Hardware Performance Evaluation

In order to evaluate hardware performance and the necessary resources, we mainly examine the characteristics of each Computing Block.

Tab.1 and tab. 2 resume LUTs required by a Correspondence Block and by the Interest Block.

Quant.	Type	Function, Collocation
32	1K-word, 5-bit/word	absolute error, first level
32	32-word, 7-bit/word	Binary->S* conversion, second level
62	1K-word, 5-bit/word	sum, other levels (two parallel layers)
31	16-word, 2-bit/word	sum, other levels (a third layer)
2	1K-word, 5-bit/word	subtraction (comparison)
1	16-word, 2-bit/word	subtraction (comparison)
1	4K-word, 1-bit/word	S*->sign conversion (comparison)

Tab. 1: Quantity, type, function and collocation of the LUTs required by a Correspondence Block.

Quant.	Type	Function, Collocation
6*4	1K-word, 5-bit/word	absolute error, first level
6*4	32-word, 4-bit/word	Binary->S' conversion, second level

5*4	1K-word, 5-bit/word	sum, other levels (first layer)
5*4	256-word, 4-bit/word	sum, other levels (second layers)
3+1	1K-word, 5-bit/word	subtraction (tree of comparison)
3+1	256-word, 4-bit/word	subtraction (tree of comparison)
3+1	512-word, 1-bit/word	S'-> sign conversion (comparison)

Tab. 2: Quantity, type, function and collocation of the LUTs required in the Interest Block.

These LUTs have been compiled as ROM in 0.7 mm CMOS technology (ES2 Standard Cells Library) and the data sheets are reported in tab. 3. and in tab. 4.

#	Type	tacc [ns]	tcyc [ns]	size (mm)	area	#*area
96	1K-word,5-bit/word	12.40	24.75	0.77*0.58	0.443	42.528
32	32-word,7-bit/word	12.17	22.32	0.52*0.37	0.196	6.272
32	16-word,2-bit/word	11.93	21.79	0.43*0.36	0.154	4.928
1	4K-word,1-bit/word	13.95	24.40	0.69*0.58	0.399	0.399
					Total Area	**54.127**
					Max.Freq. [Mhz]	**40.4**

Tab. 3: Data sheets of the LUTs required in the Correspondence Block. They have been compiled as ROM in 0.7 mm CMOS technology (ES2 Standard Cells Library).

#	Type	tacc [ns]	tcyc [ns]	size mm*mm	area	#*area
48	1K-word,5-bit/word	12.40	24.75	0.77*0.58	0.443	21.264
24	32-word,4-bit/word	12.09	22.07	0.47*0.37	0.174	4.176
24	256-word,4-bit/word	12.86	23.01	0.54*0.46	0.250	6.000
4	512-word,1-bit/word	12.42	22.52	0.54*0.40	0.218	0.872
					Total Area	**32.312**
					Max .Freq. [Mhz]	**40.4**

Tab. 4: data sheets of the LUTs required in the Interest Block. They have been compiled as ROM in 0.7 mm CMOS technology (ES2 Standard Cells Library).

Because of the pipe structure, the highest operating frequency of the structure is the same of the slowest LUT: so the whole architecture may process an image having a pixel frequency of about 40 MHz.

5. Conclusions

A VLSI architecture ables to select and to match a set of features over a time varying sequence of images is described. Both, extraction and matching steps are performed, independently, on each acquired frame.
Our architecture can process images at a rate of about 40Mpixel per second (a 512*512 TV frame at 50 Hz has a rate of 13M pixel per second). This computing power has been reached, mainly, by mean of Look Up Tables (LUTs) whose sizes were optimized by using Residue Number System (RNS).
A medium size chip is sufficient to integrate in 0.7 mm CMOS technology (ES2 Standard Cells Library) the LUTs that are required both by the Interest Block (performing the extraction of the features) and by the Correspondence Block (finding the correct matches for the extracted features).
The study and the design of the described hardware start from the need of realtime image processing for passive navigation tasks of our mobile robot SAURO. As soon as, hardware will be available it will be test on SAURO architecture.

REFERENCES

1. A.Branca, G.Cicirelli, E.Stella, A.Distante "Mobile Vehicle's Egomotion Estimation from Time Varying Image Sequences " Proc. of ICRA97,New Mexico (USA), 1997 .
2. A.Branca, E.Stella, A.Distante "Passive Navigation using Focus of Expansion", Proc. of Workshop on applications of computer vision, Sarasota (USA),1996.
3. H.P.Moravec, "The Stanford Cart and the CMU Rover",Proc. IEEE,1983.
4. D.Marr, T.Poggio, "A computational theory of human stereo vision",Proc. of Royal Society of London B, Vol.204.
5. N.Ayache, "Artificial Vision for Mobile Robots" ,MIT Press, 1991
6. G.Erten, R.M. Goodney, "Analog VLSI Implementation for Stereo Correspondence Between 2-D Images" IEEE Trans. Neural Net. vol.7 (2), pp. 266-277 (1996).
7. Alia and E. Martinelli, "A VLSI Algorithm for Direct and Reverse Conversion from Weighted Binary Number System to Residue Number System" IEEE Trans. Circuits Syst. CAS-31 (12), pp. 1033-1039 (1984).
8. Vu "Efficient Implementations of the Chinese Remainder Theorem for Sign Detection and Residue Decoding" IEEE Trans. Circuits Syst. CAS-34 (7), pp. 646-651 (1985).
9. Shenoy and R. Kumaresen, "Residue to Binary Conversion for RNS Arithmetic Using Only Modular Look-up Tables" IEEE Trans. Circuits Syst. CAS-35 (9), pp. 1158-1162 (1988).
10. Capocelli and R. Giancarlo "Efficient VLSI Networks for Converting an Integer from Binary System to Residue Number System and Vice Versa" IEEE Trans. Circuits Syst. CAS-35 (11), pp. 1425-1430 (1988).
11. Elleithy and M. A. Bayoumi "Fast Flexible Architecture for RNS Arithmetic Decoding" IEEE Trans. Circuits Syst. CAS-II 39 (4), pp. 226-235 (1992).
12. J.Martin and J.L.Crowley "Comparison of Correlation techniques" , Intelligent Autonomous Systems,U.Rembold et al. (Eds.), IOS Press, 1995.
13. F.Marino, E. Stella, N. Veneziani,A.Distante "Real Time specilized hardware for visual feature matching", IESI internal report, 1996.