JOURNAL OF ENGINEERING
annals OF
FACULTY OF ENGINEERING HUNEDOARA
Tome V (year 2007), Fascicole 3, (ISSN 1584 – 2673)

IMAGE STABILIZATION ALGORITHM OPTIMIZED FOR A HOMOGENEOUS COMMUNICATING PROCESSOR NETWORK INTEGRATED ON SoPC

[1]Dorin ROIBAN, [2]Lionel DAMEZ, [3]Jean-Pierre DERUTIN

[1] Blaise Pascal University, LASMEA/Laboratory of Sciences and Materials for Electronics and of Automatic, VIRO / VIsion and RObotic, CLERMONT-FERRAND, FRANCE
[2] Blaise Pascal University, LASMEA/Laboratory of Sciences and Materials for Electronics and of Automatic, GRAVIR/GRoup of Automatic: VIsion and RObotic, CLERMONT-FERRAND, FRANCE
[3] Blaise Pascal University, LASMEA/Laboratory of Sciences and Materials for Electronics and of Automatic, CLERMONT-FERRAND, FRANCE

ABSTRACT
The electronic stabilization of video images consists in compensating the effects of a parasitic movement in a sequence of video images. The output of the stabilization system is a modified image sequence in which unwanted components of image motion have been reduced or removed.
We have to calculate the amplitude of the parasitic movements then to compensate them. For real-time application we need a very simple but efficient algorithm.
Keywords
Electronic Digital Image Stabilization, algorithm, network integrated

1.INTRODUCTION

Electronic Digital Image Stabilization

- Context

The electronic stabilization of video images consists in compensating the effects of a parasitic movement in a sequence of video images. The output of the stabilization system is a modified image sequence in which unwanted components of image motion have been reduced or removed.

- Applications

The applications are numerous and we can quote:

- Tele operation of robots simplified by stabilized images
- Help driving system vehicle (Road line detection system)
- Autonomous Navigation System (3D localization system by vision sensor)
- Old movies restoration (black and white pictures)

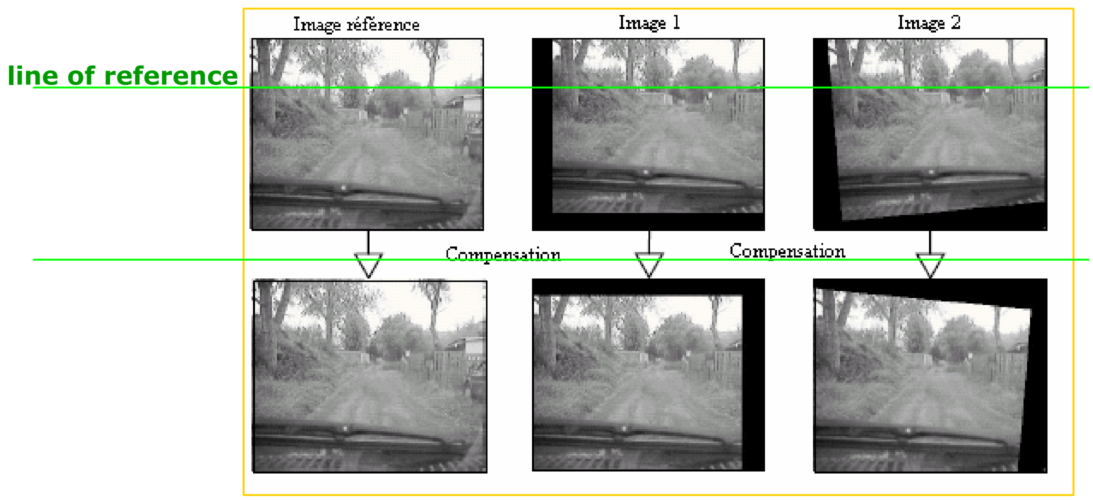
The three first points concern the domain of robotics with constraints of the same value either from a temporal aspect or from the size of the image. These three types of application could require the definition of a video sensor integrating this algorithm.

The fourth point concerns the restoration of old films to correct, in particular ,the changes of images linked up to the initial mechanism of recording. According to the size of image, this last one may need very big computation capacities.

- Problematic

We have to calculate the amplitude of the parasitic movements then to compensate them. For real-time application we need a very simple but efficient algorithm.

Non Stabilized Images i, i+1 and i+2

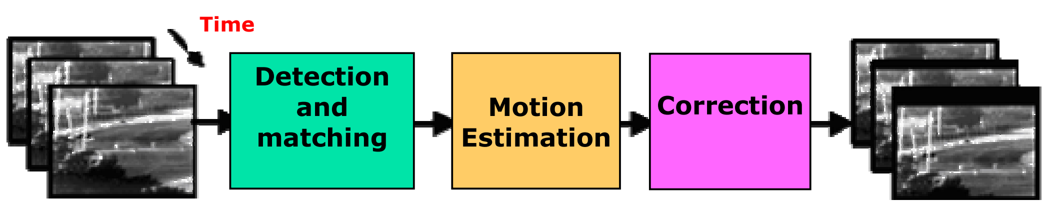


Stabilized Images i, i+1 and i+2

2. STABILIZATION METHOD

The proposed method is decomposed in three parts:

2.1 Detection and matching of features in two consecutive images
2.2 Global Movement Estimation
2.3 Correction/compensation of the parasite motion



2.1. Detection and matching of features in two consecutive images

In the **first** step we search for *feature selection* in image i using Harr's wavelet because of its simplicity of calculation, approach introduced by Papageorgiou.

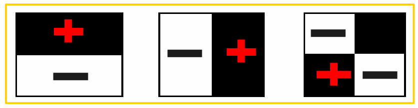
The *feature selection* is divided into three parts:

→ The first stage consists in determining a set of points of interest in the image *i* by using a detector ,then to realize a matching of these points of interest in the following image, *i+1;*

→ From this matching, the second stage consists in considering the global movement from an image to the other one;

→ And finally, from this estimation, the third stage will allow to compensate totally or partially this movement.

We use three detectors according to 3 different orientations: *Horizontal*, *Vertical* and *Diagonal* and this allow us to obtain n points of interest



The value of « a two rectangle feature » is the difference between the sums of the pixels within two rectangle regions.

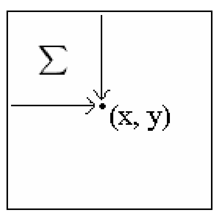
From convolution of a ROI (region of interest) with a 10x10 mask, with mask coefficients +1 or -1 results in a grey level gradient and the maximum for each mask in each ROI is kept.

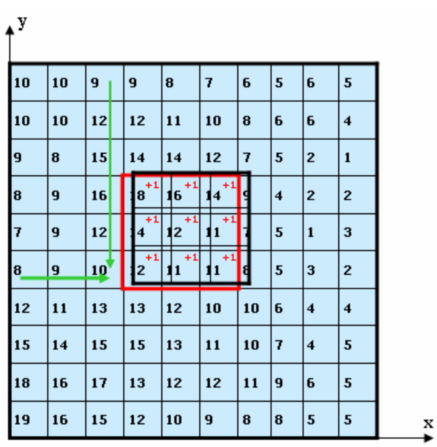
To define the number of ROI, we define first the number of points of interest we want to use and then we divide it by 3 in order to obtain the number of ROI.

For this example we have 42 features=14 ROI.

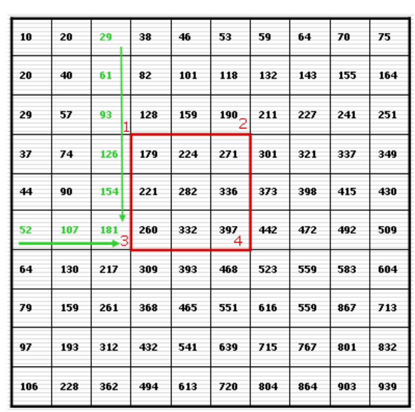
Convolutions are very easily computed with the use of a transformed image called *Integral Image* [1]

$$ii(x,y) = \sum_{x' \le x,\; y' \le y} i(x', y')$$





10	10	9	9	8	7	6	5	6	5
10	10	12	12	11	10	8	6	6	4
9	8	15	14	14	12	7	5	2	1
8	9	16	18	16	14	9	4	2	2
7	9	12	14	12	11	7	5	1	3
8	9	10	12	11	11	8	5	3	2
12	11	13	13	12	10	10	6	4	4
15	14	15	15	13	11	10	7	4	5
18	16	17	13	12	12	11	9	6	5
19	16	15	12	10	9	8	8	5	5



10	20	29	38	46	53	59	64	70	75
20	40	61	82	101	118	132	143	155	164
29	57	93	128	159	190	211	227	241	251
37	74	126	179	224	271	301	321	337	349
44	90	154	221	282	336	373	398	415	430
52	107	181	260	332	397	442	472	492	509
64	130	217	309	393	468	523	559	583	604
79	159	261	368	465	551	616	559	867	713
97	193	312	432	541	639	715	767	801	832
106	228	362	494	613	720	804	864	903	939

In order to get the values for the *Integral Image* we just have to use pair of recurrences:

$s(x,y) = s(x, y-1) + i(x,y)$ the cumulative row sum (x constant)

ii(x,y) = ii(x_1,y) + s(x,y)

Example:

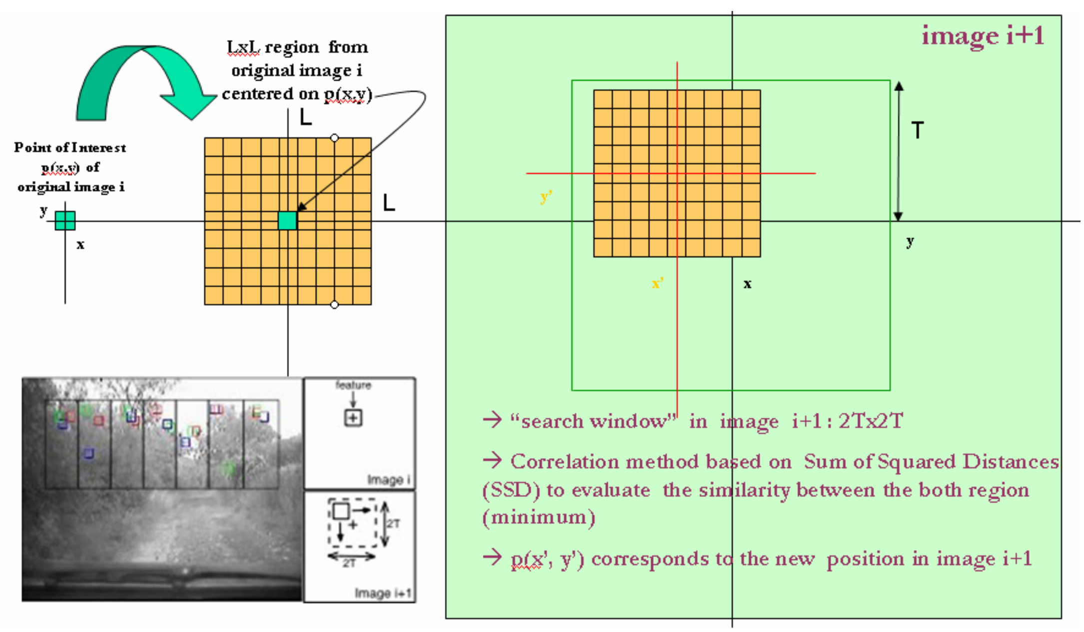
- 107 + (9 + 12 + 15 + 16 + 12 + 10) = 181
- Sum of pixels within a region is evaluated with 3 +/- operations :
- (val_4 + val_1 – (val_2 + val_3)) = 119

For the second step we need to measure the similarity between the searched feature and its potential matching. Calculation cost: (2Tx2T SSD correlations with an 8x8 or 16x16 region) x n features.

The calculation method proposed to reduce processing load impose a multi-resolution approach.

- ¼ sub-sampled image with a « search window » of size T/2 x T/2 (first estimation, LxL region)
- ½ sub-sampled image with a « search window » of size 3 x 3, centered on estimated position at the previous stage, (2nd estimation, LxL region).
- Original image with a "search window" of 2x2, centered on estimated position at the previous stage, (final estimation LxL region)
- ½ and ¼ sub-sampled images are directly calculated from Integral Image

With *n* selected points of interest, the second step consists in building *n* regions of LxL size, centered on every selected point and using a method of correlation (Sum of Squared Distances) which allows a measure of similarity between this region of reference and a potential region in the image *i+1*.

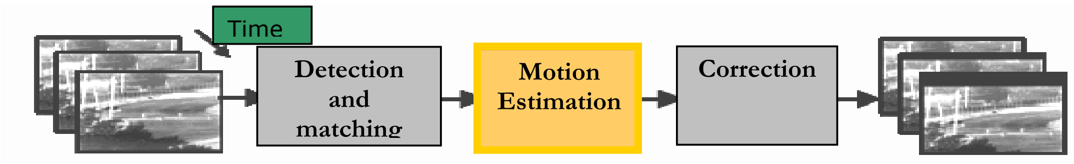


For every region, we define a 2T by 2T window of research, with T, maximum distance crossed by a pixel of the image *i* in the image *i+1*

The final estimation will be realized with a sub-pixel approach (1/4 of pixel) by using the bilinear interpolation technique.

An important point is that the processing of the sub-sampled images (by classical filtering technique) is directly made from the Integral Image (which allows the optimization of the calculations), the calculation of the SSD being naturally made on grey level values.

2.2. Motion Estimation

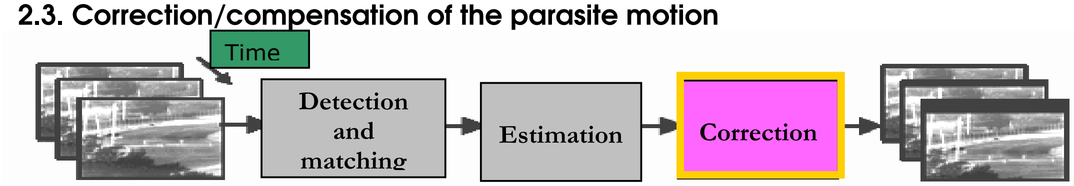


For *motion estimation* we use a 2D Model parameters, describing the movement (hypothesis of planar scene) based on a homogeneous transformation matrix composed of two translations Δx, Δy and a rotation $\Delta\theta$ around the optical axis calculated for each *n* features in image *i+1*:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & tx \\ -\sin\theta & \cos\theta & ty \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

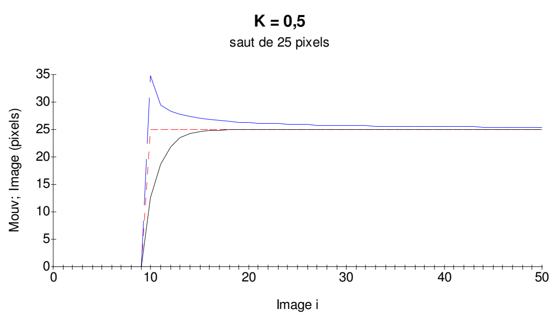
The errors are minimized using the Median Least Squares Method.

2.3. Correction/compensation of the parasite motion

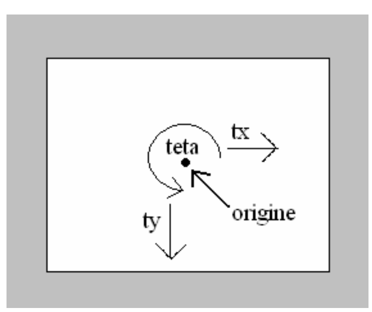


The **first step** for movement compensation is to use a digital filtering of parasitic motion (first order linear filter). We assume that:

Parasite motion → high frequency
Controlled motion of robot → low frequency



The **second step**, as a matter of fact the *movement compensation*, imply an inverse homogeneous transformation applied to image i+1.



3. HARDWARE TARGETS

FIRST VERSION

For algorithm validation it was used a *sequential* approach.

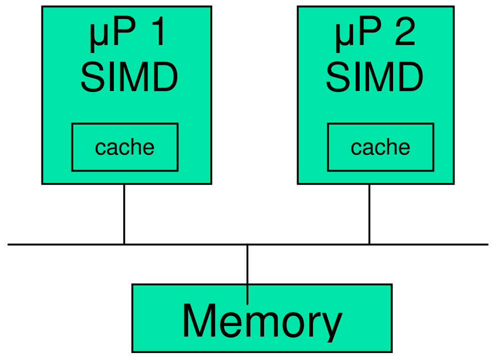
→ Compatible PC: AMD **Athlon XP 1,47 GHz**

Real time aspect validation

→ Machine Apple PowerPC **2x G4-1GHz – Mac OS 10**

→ Machine Apple PowerPC **2x G5-2GHz – Mac OS 10**

Bi processors Machine
Shared memory : SMP Distributed
Memory : MIMD



For real time application were used 3 levels of parallelism:

→ **SMP** Architecture: software using a multi-thread approach (***pthread***)

→ **SIMD** Architecture: Instructions set ***ALTIVEC***

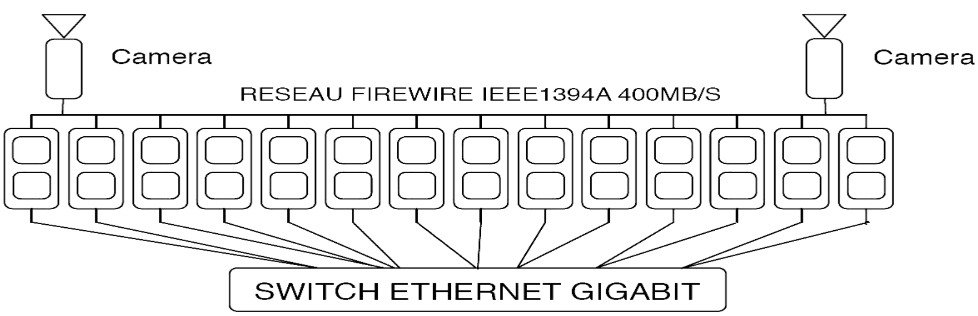
→ **MIMD-DM** Architecture (14 machines « power G5 bi µP »): message passing with use of *MPI library*

Hardware Targets-**first version**

Machine Babylone:

14 node bi-processors (PPC970)

→ Cluster double bus



This structure is used for fast "software" prototyping of real time vision applications. The CAD tools for parallelization in this case are:

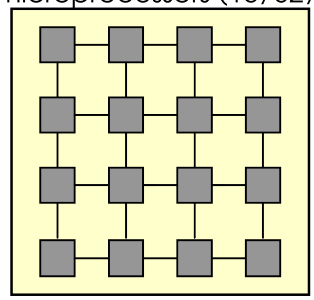
→ Software: SKiPPER (syndex)

→ Libraries: Eve, Quaff

SECOND VERSION

As hardware target we propose a homogeneous and regular network of communicating processor cores on ASIC. We use the same parallelism model:

- MIMD-DM
- Messages passing communication
- The computing power is based on a important number of microprocessors (16, 32, 64, …)



This approach involves a hardware prototyping, hardware tunning and ,also, an algorithm architecture adequacy.

The major differences between a «classic» PC or Mac and FPGA hardware are:

- *Memory dimension*
- *Clock frequency*
- *FPU, Vector unit*

The actual SoPC FPGA structures permit to implement multiple CPU on the same hardware entity.

Hardware	Memory	Clock frequency	Consumption power	Mobillity
PC&MAC	~1Go	~1,5Ghz	~150W	NO
SoPC	~1Mo	~100Mhz	~5W	YES

Important!

To implement a MIMD-DM model in a FPGA structure we have to optimize (shrink) the program code size and also to pass from float point computation to fix point computation.

4. CONCLUSION AND PERSPECTIVE

The first results I have obtained, with the help of my colleagues, until today are really encouraging me to continue my work in implementing this image stabilization algorithm on a SoPC FPGA structure.

This technology is evaluating very fast so it's a matter of months until we will have FPGA structures with big amounts of memory that will permit us to implement more and more complex and powerful image processing tools and all this with the advantage of mobility, low power consumption and, of course, the flexibility and speed of design phase.

The design tools for FPGA development are in these days friendlier and the « *barriers* » between a non IT engineer and a hardware application are vanishing.

Applying this technology, the intelligent sensors will become « smarter» in a shorter period of time.

Bibliography

(1.) Jean Pierre DERUTIN, Fabio DIAS, Lionel DAMEZ, Nicolas ALLEZARD. SIMD, SMP and MIMD-DM parallel approaches for real-time 2D image stabilization. Proceedings on Computer Architecture for Machine Perception. CAMP'05.

(2.) J.P. Derutin, L. Damez, A Desportes, J.L. Lázaro. Design of a Scalable Network of Communicating Soft Processors on FPGA. Proceedings on Computer Architecture for Machine Perception and Sensing. CAMPS'06. sept. 2006

(3.) P. Viola and M. Jones. Robust real-time object detection. In Proc. of IEEE Workshop on Statistical and Computational Theories of Vision, 2001.

(4.) Paul Viola, Michael Jones « Rapid object detection using a Boosted Cascade of Simple feature » - 2001