

**AN IMAGE ANALYSIS METHOD
FOR VEHICLE STABILIZATION**

by

MENACHEM ZUCKER

A Thesis Submitted for the Degree of

Doctor of Philosophy of the University of London

and

The Diploma of Imperial College of Science, Technology and Medicine

March 1990

Department of Electrical Engineering

Imperial College of Science, Technology and Medicine

University of London

ABSTRACT

This thesis describes a new approach for vehicle stabilizing in rotation, by using video information from a camera mounted in the vehicle. We assume that the camera has two degrees of freedom, pitch and yaw, and both are stabilized. The main goal is to estimate the roll angle in real time for closing the roll loop, and as a by-product to get the scale-factor and the camera drifts in pitch and yaw.

The video input data is natural uniform terrain, representing deserts and other desolate areas, without any particular target or object. These assumptions present a novel problem, which led us to develop three new algorithms.

The first is a procedure for 'automatic window location' (AWL) which eventually locates two reference windows, on the left and right sides of the picture respectively, in the 'best place' possible .

The second algorithm is the 'improved mean absolute differences' (IMAD) which is operated in parallel on both side of the picture, involving weight-function methods in order to be able to estimate $\epsilon_{v,l,r}$ and $\epsilon_{h,l,r}$, the vertical and horizontal displacements respectively , on both sides of the picture .

The third algorithm is the 'picture estimation process' (PEP) , which utilizes the information from the IMAD process in order to estimate the roll angle ($\hat{\phi}$), the vertical and horizontal displacement, ($\hat{d}_v$ and $\hat{d}_h$) , and the scale-factor (sf) .

By using this approach we get a fast and simple estimation process.

The process sensitivity to noise and other distortions is discussed .

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	2
LIST OF FIGURES	7
LIST OF SYMBOLS AND ABBREVIATIONS	9

CHAPTER

1.	INTRODUCTION	11
	1.1 Background	11
	1.2 Report Organization and Summary	11
2.	SYSTEM OVERVIEW	14
	2.1 The Systems Properties	14
	2.2 Desirable System Features	15
	2.3 Algorithms Necessary for Successful System	15
	2.4 The System Main Block Diagram	16
3.	THE HUMAN VISUAL SYSTEM (HVS)	17
	3.1 The HVS As a Basic Approach to the Proposed System	17
	3.2 Texture Analysis	19
4.	SUMMARY OF PREVIOUS WORK	21
	4.1 Previous Research Efforts - Map matching	21
	4.1.1 Statistical Decision Theory	22
	4.1.2 The <i>A Posteriori</i> Probability	22
	4.1.3 The Extremals Metric	24
	4.2 Comments on Previous Efforts	25
5.	FUNDAMENTAL ASSUMPTIONS	27
	5.1 The Video Signal	27
	5.2 The Search Areas	28
	5.3 The Reference Window	29
6.	THE AUTOMATIC WINDOW LOCATION (AWL) ALGORITHM	31
	6.1 The AWL Process	31
	6.2 The AWL Performance	35

6.3	The AWL Sensitivity to Noise	43
6.4	The AWL Sensitivity to Data-Sample.....	46
7.	THE MATCHING PROCESS	48
7.1	The Current Window.....	49
7.2	The Matching Problem.....	50
7.3	The Mean Absolute Difference (MAD) Algorithm	51
7.3.1	The MAD with Overlapping	54
7.3.2	The MAD with Overlapping and a Weight Function	56
7.3.3	The MAD+ with a Decision Criterion.....	58
7.3.4	The MAD Process in the Horizontal Direction	61
7.4	The IMAD Process	62
7.4.1	The IMAD and Skip Technique	64
8.	THE ESTIMATION PROCESS	66
8.1	The Displacement Model	66
8.2	The Estimator Algorithm	68
8.3	The PEP performance	73
9.	CLOSING THE LOOP.....	78
9.1	The Plant Model.....	79
9.2	The Model of the other Blocks	80
9.3	The Close-Loop Performance	82
10.	THE COMPUTATIONAL REQUIREMENTS	85
10.1	Trends in Calculation Capability	85
10.2	The Desired Calculation capacity	88
10.2.1	The IMAD and PEP Burden of Calculation	88
11.	CONCLUSIONS AND SUGGESTIONS FOR FURTHER STUDY	91
	REFERENCES	94

APPENDICES

1.	A Sample of Uniform Pictures.....	96
2.	Bilinear Interpolation	99
3.	Examples of the Reference-Window and the Current-Window	101
4.	The Spatial Resolution versus the Line Of Sight Angle (λ).....	103

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor M.H.A. Davis for his constant guidance and support during the research period and for invaluable assistance during the preparation of this thesis .

I would like to thank Dr J.M.C. Clark for many hours of discussion and useful advice at the beginning of the research and Professor A.G .Constantinides and his staff for giving me advice and useful references..

Many thanks go to all the staff and students of the control section.

Very special thanks go to my wife , Rachel who shared my everyday problems and whose help and understanding proved truly invaluable. Many thanks go also to my lovely children Chaim , Motti , Noam and Koby who enable me to carry on my research without too many disturbances.

To the remembrance of my dear father...

**To the remembrance of my father's
family who had been exterminated
in the holocaust.....**

List of Figures

<u>Figure</u>	<u>Page</u>
1.	Some sort of uniform terrain 14
2.	The major system software components 16
2-a.	Basic scene formulation using model 18
2-b	The HVS block-diagram 19
3.	The terrain 3-D illustration 27
4.	The two search area in the FOV 28
4-a.	The reference templates - $R_{l,r}(j,i)$ inside the two search-areas 30
5.	An illustration of the Extended Sobel Filter (ESF).....32
6.	The AWL algorithm block-diagram 34
7.	The AWL algorithm operate on picture (c) in A_L 35
8.	The horizontal(S_h) and vertical(S_v) components of $S(1,i)$ 36
9.	The PQF definition 37
10.	The model of synthetic pictures 38
10-a.	$S(j,i)$ and $PQF(j,i)$. vs. picture quality 39
11.	Picture (c) the search-areas A_L , A_R and the ref-windows R_l R_r 40
12.	Picture (c) - A_L , the histograms of reference-windows, $R_l(1,4)$ and $R_l(1,6)$ 41
13.	The comparison between the AWL results and the left template, in picture (b) 41
14.	Histograms for several types of $S(j,i)$ 43
15.	The AWL algorithm without filtering and with the ESF 45
16-a.	The AWL algorithm with noise and without the ESF 45
16-b.	The AWL algorithm with noise and with ESF 45
17-a.	PQF .vs. the template size $k_r(\text{pix})$, picture (a) 46
17-b.	PQF .vs. the template size $k_r(\text{pix})$, picture (c).....46
18.	The reference and the current windows 49
19.	The initial step (a) and the final step (b) along the , matching process 52
20.	MAD - nominal case , pictures (c_l) and (a_l) 54
21.	MAD , $\phi=1^\circ$, picture (a_l) without overlapping.....55
22.	MAD nominal , $\phi=1^\circ$, picture (a_l).....55

23.	MAD and MAD+weight function, picture (b_l), $\phi=2.84$	57
23-a.	MAD, $\phi=2.84^\circ$, picture (a_r)	58
24.	The matching process and the decision criterion	60
25.	The “minimum line” (L_7)	62
26.	IMAD, horizontal matching	63
27-a.	IMAD _h , picture (a_r), $\phi=90^\circ$, skip=1(pix)	64
27-b.	IMAD _h , picture (a_r), $\phi=90^\circ$, skip=2(pix)	64
28.	The IMAD block-diagram	65
29.	The ‘swing effect’	66
29-a.	The derivative model	66
30.	The Picture Estimation Process(PEP), block-diagram	72
31.	The estimation process for the roll angle- $\hat{\phi}(i)$ picture (a)	73
32-a.	The estimation process for the vertical displacement- $\hat{d}_v(i)$, picture (a)	74
32-b.	The estimation process for the horizontal displacement, $\hat{d}_h(i)$ picture (a)	74
33.	The estimated closing-range $\hat{X}_c(i)$.vs. r_o picture (c)	75
34-a.	The estimated roll angle - $\hat{\phi}(i)$.vs. pictures (a) and (c)	76
34-b.	The estimated closing range - $\hat{X}_c(i)$.vs. pictures (a) and (c)	76
35.	The scale-factor (sf) estimation interval - l_e	77
36.	The closing-loop block diagram	78
37	The roll and roll rate angle	80
38	The roll angle ϕ , for $\Delta T=0.016$ (sec)	82
39	The roll angle ϕ , for $\Delta T=0.033$ (sec)	83
40	The roll rate $\dot{\phi}$, for $\Delta T=0.033$ and $\Delta T=0.016$ (sec)	84
41	Pipeline architecture of a AUTO-Q preprocessor	86
42	The pixel processing per second .vs. time	87
43	Parallel pipeline configuration	88
44	An example of a very uniform picture from a medium range	96
45	A comparision between a picture and its digitized print	97
46	An example of a very uniform picture	98
47	A example of the ‘improvement’ .vs. the range	98
48	The transformed coordinates	99
49	The geometrical connections and the LOS definition	103

List of symbols and abbreviations

AWL	The Automatic Window Location algorithm.
IMAD	The Improved Mean Absolute Differences algorithm.
PEP	The Picture Estimator Process algorithm.
$(a)_{l,r}$	Picture (a), left or right side.
$(b)_{l,r}$	Picture (b), left or right side .
$(c)_{l,r}$	Picture (c), left or right side .
ϕ	The roll angle.
d_v	The vertical displacement.
d_h	The horizontal displacement.
sf	The Scale Factor.
FOV	Field Of View.
$\dot{\phi}$	The roll rate.
τ_p	The plant time-constant.
$\Delta\phi$	The roll angle error.
X_c	The closing -range .
$O_p(k,l)$	The picture coordinates origin.
A_l, A_r	Left and right search area repectively .
L_a, K_a	The axis origin coordinates of the search areas .
k_{sl}	The horizontal size of the left search area .
k_{sr}	The horizontal size of the right searcg area .
l_{sl}	The vertical size of the left search area .
l_{sr}	The vertical size of the right search area .
$O(x,y)$	The axis origin of the FOV .
$R_l(j,i)$	The reference window located in coordinates (j,i) in A_l .
$R_r(j,i)$	The reference window located in coordinates (j,i) in A_r .
k_{rl}	The horizontal size of the left reference window .
l_{rl}	The vertical size of the left reference window .
k_{rr}	The horizontal size of the right reference window .
l_{rr}	The vertical size of the right reference window .
$I(k,l)$	The video intensisty in coordinates (k,l) .
$x_{v,h}$	The vertical and horizontal shift in the AWL process .

PQF	The picture quality factor .
N	The number of pixels invoved in the process.
N_H	The number of pixels involved in the histogram .
C_L, C_R	The left and right current window .
k_{c_l}, l_{c_l}	The left current window dimension in the horizontal and vertical directions .
k_{c_r}, l_{c_r}	The right current window dimensions in the horizontal and vertical directions .
$O_c(k,l)$	The axis origin of the current windows .
$\epsilon_{h_{l,r}}$	The estimated horizontal error in the left and right matching process .
$\epsilon_{v_{l,r}}$	The estimated vertical error in the left and right matching process .
i	The frame number .
r_o	The initial range between the camera and the obseved terrain .
$\bar{v}$	The average velocity .
τ	The time cycle of the whole loop .
J	The shift number along the matching process .
J^*	The shift number for the best matching with the MAD(J) process.
J^+	The shift number for the best matching with the IMAD(J) process
m_a, m_b	The vertical bounderies in the matching process.
n_a, n_b	The horizontal bouderies in the matching process .
$\eta(J)$	The decision criterion .
$\bar{P}(i)$	The estimated error vector .
T_h	The PQF threshold .
I_x	The moment of inertia
q	The dynamic pressure
Sd	A aerodynamic normalized factor
$C_{LP}, C_{L\delta}$	The damping moment coeficinent and the actuator monent coefficient respectively .
α_o	Half angle of the vertical FOV.
λ	The line of sight angle .
R_λ	The range between the camera and the terrain for an arbirary λ .

CHAPTER 1

1. INTRODUCTION

1.1 Background

In recent years most research in video tracking systems has focussed on scenes which contain objects or targets in a time-varying environment. Their main goal was to find an algorithm in real time which will estimate the displacements of the object in the field of view (FOV). Furthermore, several systems which have been proposed in the relevant literature [4], [10] have utilized the estimated variables to generate control signals. Interesting and outstanding work is reported in [5] and [6].

This paper focuses on uniform pictures which represent uniform terrain, such as deserts with no particular objects or targets; these areas are observed by a video camera mounted on an airborne vehicle, stabilized in pitch and yaw . Our main goal is to present a method which uses the video information to estimate the roll angle in real time and then to use this information for closing the rool loop.

Since we have not any particular object in the uniform terrain , we have to define a template which will be a reference window for the matching process. The decision where to locate these templates and the updating process have to be taken automatically by the system, depend on the matching quality. The location of those templates has to be in the 'best place' in the search area in order to facilitate successful matching, bearing in mind that we tackle two major problems, namely the uniform landscape and the vehicle dynamic performance ,with a roll rate which could reach 120 (deg/sec) and more, with a very short time constant (i.e., 50 msec). Furthermore the desired accuracy of the estimated roll angle must be better than $\pm 1^\circ$, and in real time. These requirements cause a significant change in the regular image processing method and in the estimation process and led us to a quite new approach , as described below .

1.2 Report Organization and Summary

The report is organized in a serial procedure which means that in order to understand chapter N you must read chapter N-1 . Moreover, usually the output of chapter N-1 is used as a data base for chapter N .

Therefore the reader must follow the chapters as they appear in the report, in order to get a comprehensive understanding. We now give a short summary of the contents of each chapter.

In chapter 2 we describe the problem we are dealing with in our research . We emphasize the unique properties of the air-born vehicle, the terrain and the effects they have on our approach . At this stage we illustrate the desirable features needed to solve the problem , we submit a general block-diagram which describes the major algorithms in the process .

In chapter 3 we describe several major aspects of the Human Visual System (HVS) , which are related to our problem, since our approach for vehicle stabilization based on picture analysis is an attempt to emulate the HVS and therefore our problem is basically a part of the artificial intelligence field.

In chapter 4 we describe briefly previous research which is related to our work. Since we do not deal with specific targets or objects but with uniform terrains , the closest subject to our research is 'map matching'. Unfortunately most work in this subject is classified, which cause some difficulties to find good references, but still we have several papers which emphasize the uniqueness of dealing with terrain instead of well defined objects , and the implications it has on the matching and decision process.

Chapter 5 gives the basic assumptions which lead us in the next chapter to the first stage of our approach .

In chapter 6 ,we propose our new approach for the first stage of the problem, which is 'where is the best place for locating a reference window ? ' , which later will be used for the matching process. This process is called the Automatic Window Location (AWL) which is a sort of segmentation and is described in detail in this chapter.

Chapter 7 like the previous chapter, is one of the three fundamental chapters (6,7 and 8) in this research. In this chapter we describe our approach to the matching process which is quite different from other matching processes described in the relevant literature , and the reason is the sort of

pictures we are dealing with. This process is called the 'Improved Mean Absolute Difference' (IMAD) and is described in detail including the system parameter sensitivity.

Chapter 8 is the third part of the core of our research and describes a method for fast and simple parameter estimation which is suited to our approach since it is based on parallel processing in two windows and time sharing in the estimation process . This process is called the Picture Estimation Process (PEP) and results of this technique are shown .

In chapter 9 we describe the implications of closing a roll loop with image processing method like the one we describe in the previous chapters. We show the effects of using a time cycle of 16(msec) or 33(msec) and the performance of the air-born vehicle to disturbances.

In chapter 10 we discuss the computational requirements. We show that the way our algorithms are operated is very efficient from the real time demands aspect. We describe the current capability in pixel processing and we show that actually we are well within real time limits.

Chapter 11 Summarizes the major results from our research and recommends further developments , which might improve our results.

CHAPTER 2

2. SYSTEM OVERVIEW

The system we are dealing with is composed of two major subsystems :

1. The airborne vehicle.
2. The observed landscape.

Both subsystems have special properties, which necessitate a unique comprehensive design.

2.1 The system properties :

The airborne vehicle has some outstanding characteristics :

- a very high roll rate $\dot{\phi}_{\max} = 3$ (rad/sec).
- a small time-constant $\tau_p = 250$ (msec)
- a maximum roll angle error allowed $\Delta\phi \leq 1^\circ$

The observed landscape is a uniform one ,which has the following properties :

- few details without any particular targets or objects.
- small variations on the video intensities among the pixels in the neighbourhood of an arbitrary pixel.
- high probability for similar regions in the same field of view - FOV .
- low probability of finding sharp lines in the FOV.

Examples of the terrain we are dealing with are shown in Fig. 1. (a) ,(b) and (c).

More pictures are shown in Appendix 1.

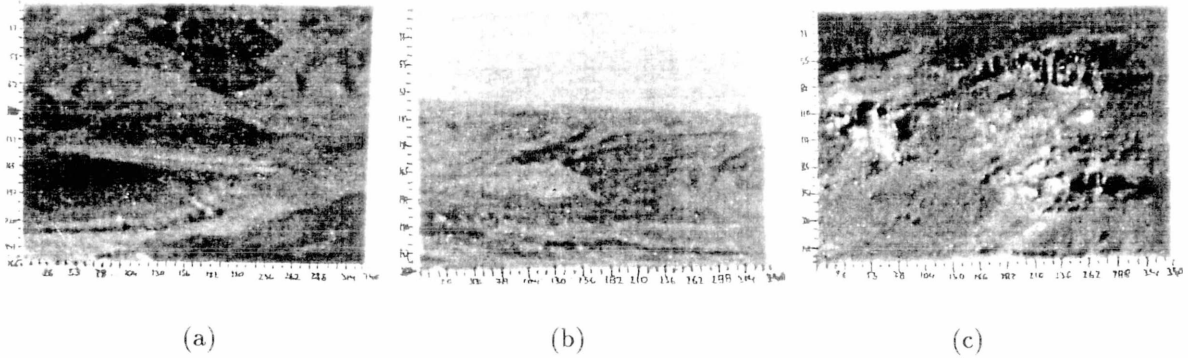


Fig. 1. Examples of uniform terrain

It is easy to appreciate that picture (a) is more uniform than (b) and that (c) contains several detailed areas relative to (a) or (b); each of the pictures represents a different type of terrain. The pictures were televised from different ranges: picture (a) was taken from a middle range, about 4 Km, picture (b) was taken from a long distance, about 10 Km, and picture (c) was taken from a short range about 0.2 Km, each of the pictures being represented by a FOV of $3^\circ \times 4^\circ$. The resolution is 260×340 pixels in the vertical and horizontal directions.

2.2 Desirable system features

The system should be designed with the following general performance :

- (1) Ability for real-time operation in complex monochromatic scenes. This constraint will affect the sampling rate as well as the complexity and the implementation of the algorithm.
- (2) Adaptability to the time-varying distance between the vehicle and the observed landscape. This constraint will generally affect the matching process and the estimation process.
- (3) Minimum probability of false matching. Later it will be shown that a suitable approach is based on methods of weight-functions and a decision algorithm.

2.3 Algorithms necessary for successful system

Referring to Fig. 2, the minimum software necessary for a successful system may be subdivided into three parts :

- (1) An automatic window location algorithm (AWL), which locates a reference window in the 'best place' in the search regions. This is done by using a gradient method in two dimensions in order to find the most detailed area.
- (2) A matching algorithm, the so called "improved mean absolute differences" (IMAD) algorithm, which processes the actual information from the current window as well as from the reference window, in order to generate a displacement vector in 2-D.
- (3) An estimation algorithm, the "picture estimation process" (PEP) in order to get the four desired variables :
 ϕ - the roll angle
 d_v - the vertical displacement

d_h - the horizontal displacement

X_c - the range between the air born vehicle and the observed train.

It should be noted that the system overall success is highly dependent upon the success of algorithm (1) and (2), and the major emphasis will therefore be on the development of the first two algorithms , i.e., AWL and IMAD. The third one - PEP should be a simple and a fast one.

2.4 The system main block diagram

As we just explained, we have two pipe-lines, on the left and right hand sides of the picture. These two pipe-lines include the two main process, the AWL and the IMAD. The pipe-line information is transfered to the estimator (PEP) in real time; this system is described is the next block-diagram

===== a real-time process.
 ----- not a real-time process.

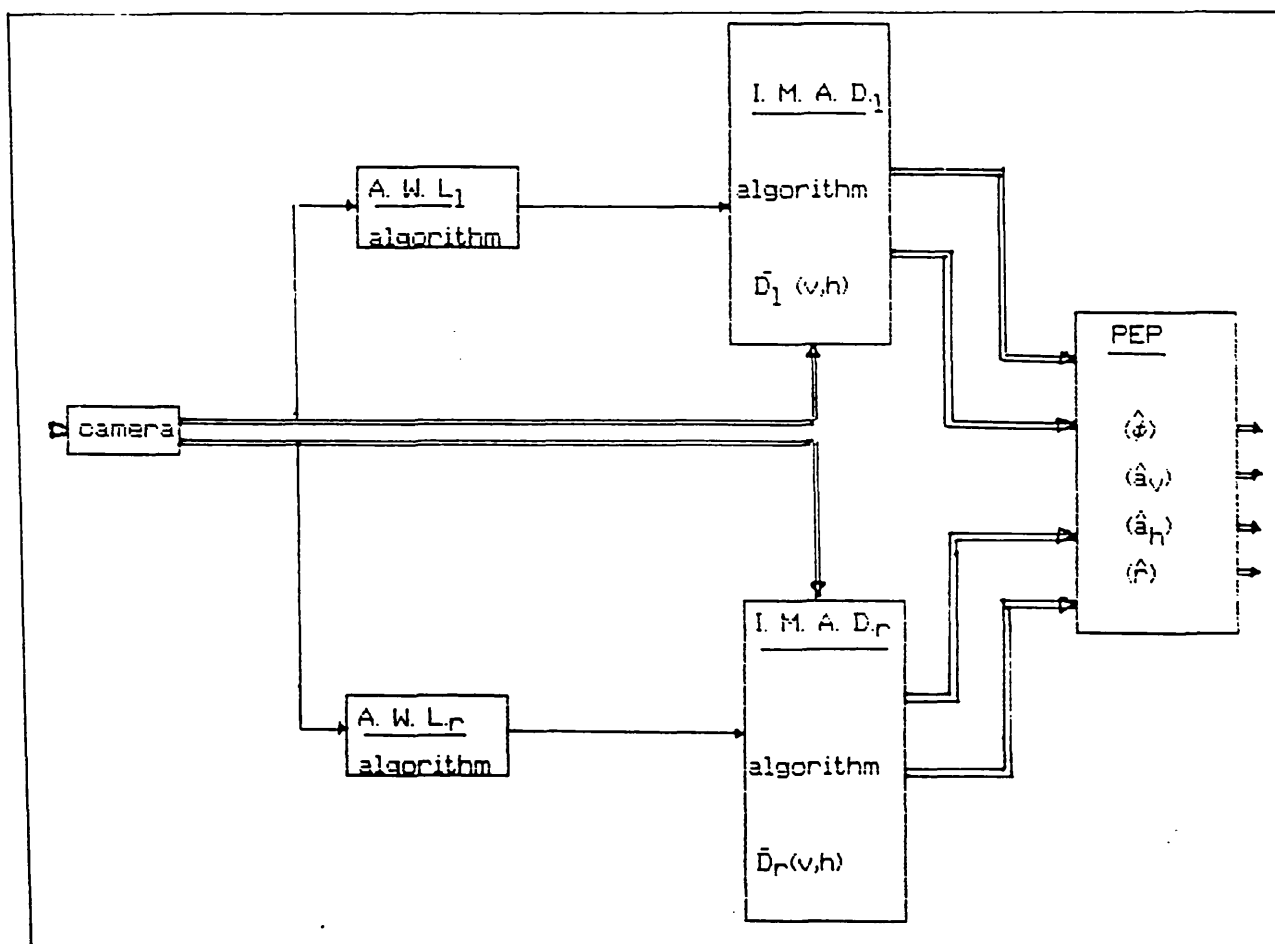


Fig. 2. The major system software components

CHAPTER 3

3. THE HUMAN VISUAL SYSTEM (HVS)

3.1 The HVS as a basic approach to our system

In developing artificial intelligence for an automatic video tracking system, the obvious comparison should be the human visual system (HVS). Although it is quite clear that the overall operation of the HVS system is not totally understood, certain basic principles of operation may provide useful design guidelines. [1]

One of the first steps in any sort of image processing is segmentation. Approaches to image segmentation are usually dependent on special applications. The algorithms are derived from optimization of mathematically defined quality criteria and are extremely dependent on the terrain and noise model, which are usually assumed to have Gaussian statistics. These algorithms perform very successfully for those models, but happen to work with much less success on scenes which are different from those for which they have been developed.

A nice example to illustrate this is the “basic (continuous) scene model formulation (2 - D)”, shown in [5] (for the case of one object).

The scene is modelled as follows :

$$P(x,t) = f(x,t) \cdot \lambda(x,t) + [1-\lambda(x,t)] \cdot b(x,t) \quad (3.1 - 1)$$

where the following functions are defined as:

$P(x,t) \equiv$ entire picture function at time t

$b(x,t) \equiv$ background function at time t

$f(x,t) \equiv$ target textural function at time t

$\lambda(x,t) \equiv$ scene characteristic or replacement function, where

$$\lambda(x,t) = \begin{cases} 1 & \text{where object exists in scene} \\ 0 & \text{elsewhere} \end{cases}$$

Thus, $\lambda(x,t)$ is the binary valued function which, by “turning on” and “turning off” the object and background function, respectively, gives the object its outline. A simple example of scene modelling, using the above concepts, is shown in Fig. 2-a.

Several points should be noted :

- (1) The scene model is a non-linear equation, and
- (2) The estimation of $\lambda(x,t)$, in a complex scene , is a very difficult problem

A system which works extremely well on a great variety of scenes, but of course not optimal in the mathematical sense, is the HVS . Therefore we study properties and processing methods of the human visual system in order to arrive at new algorithms and concepts for image segmentation .

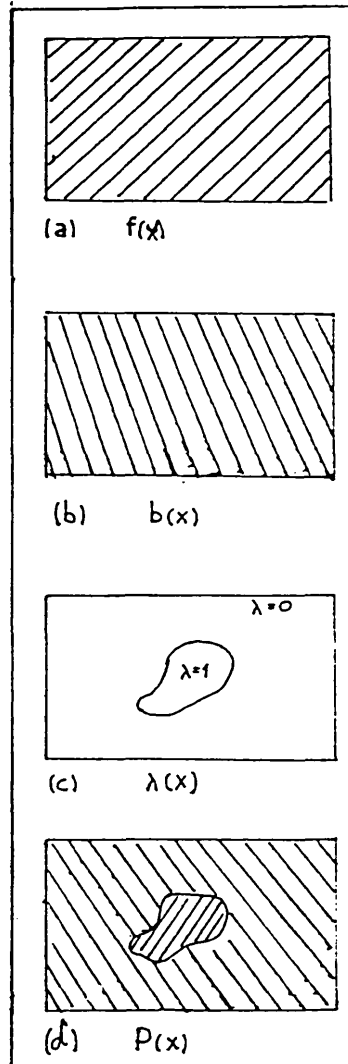


Fig. 2-a Basic scene formulation using model

Figure 2-b shows a block diagram of the human visual system which has been published earlier by Liedtke [14] . Due to nonlinear processes in the retina and

lateral inhibition of neighbouring cells the visual system performs as a bandpass filter on the incoming image and shows adaptive properties to slowly changing intensity values. On its way from the retina to the CGL (corpus geniculatum laterale) the original gray level is gradually reduced to its binary contours . In the visual cortex the binary image is filtered by various two-dimensional filters of different spatial extent and orientation sensitivity in different directions . Further processing is done in higher centres of the brain, inaccessible to direct physiological measurement, where lines and homogeneous areas are detected are combined to form meaningful objects partially guided by "experience" from an earlier learning phase .

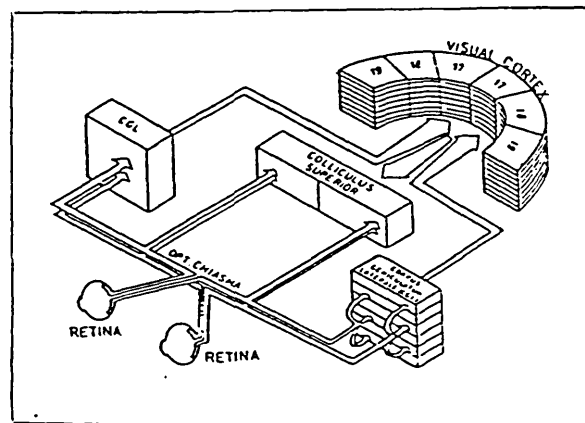


Fig. 2-b. The block-diagram of the human visual system

3.2 Texture analysis

In many cases objects in an image can be described as those subimages which are homogeneous in some local property. This local property is called texture. Objects can be separated from each other and from background by the detection of textural differences . The ability to detect a bright region depends not only on geometrical properties , but also on the brightness of the background against which the region appears, which means that there must be a certain

difference between the region and the background in order to enable detection. If the differences between the region and the background, ΔI , is low, the HVS detectability is reduced and is very sensitive to noise, and vice-versa.

Our approach to texture analysis is to develop an algorithm which automatically indicates textural differences in all those cases where in high likelihood the HVS will indicate them as well.

The visual field is usually seen as consisting of a small number of regions. The definition of a region is given by the 'Gestalt laws of organization' [1]

1. The law of similarity - states that small sets of the same properties such as brightness, colour, shape, slope and size tend to group together.
2. The law of proximity - states that closely clustered subsets tend to group together.
3. The law of good continuation - states that when curves cross or branch, parts that smoothly continue one another, are seen as one part; the HVS tends to integrate similar regions.
4. The law of closure - states that closed figures tend to be seen as units.

It is important to point out that the law of similarity has a significant effect on the design of the matching process, since in uniform pictures the main problem is the ability to define a subset as a region although we usually do not have curves or sharp changes and therefore in order to define a subset or a texture, we must find a local property in order to "get experience from an earlier learning phase", which is precisely the HVS property. In a sense correlation methods are an attempt to emulate the HVS.

There are two principal issues in which we always compare the HVS and artificial intelligence :

1. The ability of object identification in real-time, in a time varying scene
2. The ability to identify and estimate translation rotation and magnification (scale factor).

From what is published in the relevant literature, there is still a big gap in favour of the HVS regarding the subject of object identification, but regarding the ability for displacement identification and estimation, it is possible that we are now at the stage where the "artificial intelligence system" is doing it better.

CHAPTER 4

4. SUMMARY OF PREVIOUS WORK

4.1 Previous research efforts - map matching

Due to the novel nature of this subject, there are less than a handful of specific references. Perhaps the earliest unclassified reference is that of H.H. Bailey [3] in 1976. This paper outlines the overall problem of “visual map matching”, by using image correlation methods for the purpose of weapon guidance which provides autonomous flight. Most of the papers dealing with this subject are classified, but the ability to analyse natural terrain in real time is of increasing interest in artificial intelligence, particularly in connection with autonomous robotic vehicle road following [12], which uses a general terrain typing vision algorithm, in order to segment the image and find the road region. Road segmentation uses a maximum likelihood pixel classification technique to segment the image into road and nonroad regions, which is obviously a significant difference from our research, since we cannot use segmentation techniques to differentiate between one sort of terrain and the other one, and therefore it is useless in our case to operate a general terrain typing vision system [11]. Both references [3] and [12] consider that a good metric can be the likelihood ratio, but only for the extremely simple case of two possibilities or two pixels. We will refer to this issue later.

The problem of “map-matching” is the closest to our work. As mentioned in [3] the essential step in image correlation guidance is to find the position of the “best fit” between two similar but nonidentical images templates or maps, for guidance systems. The match point is found [3],[4] and [6] by systematically displacing one map relative to the other and computing for each of the many possible displacements, the value of a “comparison function” or “metric” that, ideally, has an extremum (max or min) value at the match point. The particular displacement, suitably scaled, that produces the extremum become the correction signal for the guidance system.

In scenes which represent real terrains, the two maps are not identical, which lead to the conclusion that the extremum does not always correspond to the correct match point, moreover it is possible to get a “region” of extremum without

any outstanding results and that refer particularly to uniform pictures. In [3] Bailey points out the confusion which sometimes happens when the problem of map matching is mixed with the problem of signal detection, and as a result the solution to the signal detection problem, i.e., the classical correlator, has sometimes been erroneously interpreted as the theoretically optimum solution to the map-matching problem. The starting point in many analyses of map matching has therefore been the calculation of the classical correlation function. The fundamental difference between the two problems is that in the case of map-matching we have to distinguish between signal plus noise and background plus noise while in the case of signal detection it is necessary to distinguish between signal plus noise and noise alone.

4.1.1 Statistical decision theory

Map matching is fundamentally a problem in statistical decision theory. During the acquisition phase it is necessary to decide whether or not the current map and the reference one are matched. But because of noise, errors, distortions, etc., possibly in each map, there is no foolproof way to assure that the maps have been matched correctly. The most that can be done is to determine the probability that the maps are in a certain geometrical relationship to each other based on the available data.

The optimum map-matching, therefore, is by definition the system whose output is the set of *a posteriori* probabilities describing each possible relationship. Once these probabilities are determined and costs are assigned to each kind of wrong decision, the decision rule is implicitly defined by the requirement that some measure of the total cost, e.g., the average total cost of a decision, be a minimum.

The decision rule, whatever its form, is necessarily based on the values of the *a posteriori* probabilities, and our main task is to determine these probabilities.

4.1.2 The *a posteriori* probability

For simplicity only two possibilities are considered, namely that the maps are, or are not, matched. Let these two conditions be denoted by the symbols S (signal) and B (background), respectively. If a current map consisting of N pixels is represented by the N-dimensional vector x , then Bayes' formula for the matched hypothesis states that

$$P(S | x) P(x) = P(x | S) P(S) \quad (4.1.2-1)$$

where:

$P(S | x)$ - is the *a posteriori* conditional probability that the maps are matched, given that the current map has the value x ,

$P(x)$ - is the *a priori* probability that the current map has the value x ,

$P(x | S)$ - is the conditional probability that the current map has the value x , given that the maps are matched,

$P(S)$ - is the *a priori* probability that the maps are matched.

In the same way we define the formula for the unmatched hypothesis, with B replacing S .

Since there are only two possibilities, the probability that the current map has the value x is simply

$$P(x) = P(x | S) P(S) + P(x | B) P(B) \quad (4.1.2-2)$$

The likelihood ratio $L(x)$ is defined by

$$L(x) = \frac{P(x | S)}{P(x | B)} \quad (4.1.2-3)$$

so that , after substituting Eq. (4.1.2-2) and Eq. (4.1.2-3) into Eq. (4.1.2-1) , the *a posteriori* probability of a match can be written as

$$P(S | x) = \frac{1}{1 + \frac{P(B) / P(S)}{L(x)}} \quad (4.1.2-4)$$

In a purely formal sense, Eq. (4.1.2-4) represents a complete solution to the problem. However the essential difficulty in applying Eq. (4.1.2-4) to a practical situation is that the N -dimensional joint probability functions needed to form the likelihood ratio are usually unknown . A typical current map consist of tens to thousands of spatially correlated pixels with a non-Gaussian joint probability distribution . In the absence of a theoretical framework to describe the form of the distribution, the only recourse for determining the distribution is a direct measurement , which , for maps (or templates) containing even a few pixels , is

practically impossible. Because the probability distributions needed to form the likelihood ratio are generally unknown, and , in a practical sence, essentially unmeasurable, another approach to map matching is required.

The approach most frequently encountered in the literature is one based on the extremals of a metric.

4.1.3 The extremals metric

A metric F is a class of pairs of the form $(x, F(x))$, where x is a vector and $F(x)$ is a number.* The first member of the pair is often called the object and the second member , its value. Many metrics have extremal properties, i.e. for some particular vector, the value of the metric is either a maximum or a minimum. Familiar examples of such metrics are given below.

The first one is the Normalized Inner Product (NProd) Metric, sometimes called the nomalized Product , is defined by

$$F(x) = \frac{x \cdot y}{||x|| ||y||} \quad (4.1.3-1)$$

where y is some given vector, $x \cdot y$ denotes the inner product , and $|| \quad ||$ denotes the norm of a vector defined by

$$||x|| = (x \cdot x)^{\frac{1}{2}} \quad (4.1.3-2)$$

The extremal property of this metric follows from the familiar Cauchy-Schwarz inequality. Thus , $|F(x)| \leq 1$ for all vectors $x \neq y$, with equality occurring only when $x=Cy$, where C is an arbitrary constant.

The second metric is the squared difference, defined by

$$F(x) = ||x - y||^2 \quad (4.1.3-3)$$

where y is some given vector. Thus , $F(x) \geq 0$ for $x \neq y$, with equality only when $x = y$.

*This is a standard terminology in image processing, and is synonymous with "comparison function" ; it is different from the usual mathematical notion of a "metric"

The third metric is the Mean Absolute Differences ,this metric is defined by

$$F(x) = \sum |x_i - y_i| \quad (4.1.3-4)$$

wher x_i and y_i denotes i th components of the vector x and y , respectively , and the sum extends over all components. As with the squared difference, $F(x) \geq 0$ for $x \neq y$, with equality only when $x = y$.

If the vectors x and y are associated with the current map and the reference map, respectively, then a physically reasonable basis for attempting to adapt such metrics to the problem of map matching immediately suggests itself. If it is assumed that the two maps coincide, then it is reasonable to assume that the value of the metric in that case will be “closest” to the extremal value. Because of noise and other distortions, however, a perfect match is never to be expected . If it is known beforehand that the current map is contained at least partly within the reference map, the decision rule may simply decide that the extremal-producing map defines the match point . If the current map is not known to be contained within the reference map the decision about the matching point is almost impossible.

4.2 Comments on previous efforts

The main results of previous efforts are as follows:

- (1) There must be a tradeoff between robustness, generality, and real time performance, in particular when we deal with realistic scenes.
- (2) The two most commonly used approaches to determine the probability of a correct match, P_c , for map matching, are the so-called Product ,which means the sum of the products, which is related to classical correlation and the MAD which is the mean absolute difference algorithm.
- (3) It is known that in all cases P_c increases with the size of the data sample and with the elemental signal-to-noise ratio (S/N) and decreases slowly with increasing search-area.
- (4) It is also known that at low S/N , the Product algorithm is the preferred one, but in high S/N , the MAD algorithm is preferred.

(5) The likelihood ratio computed for each match and then choosing the match point at the place where the likelihood ratio is the maximum, could be the optimum solution. Unfortunately, that computation requires a knowledge of N dimensional joint probability distributions functions, that are unknown and in practical sense unmeasurable.

(6) The theoretical model of random Gaussian scene is known to be not completely realistic, but it appears to err on the conservative side. Thus, a "floor" for P_c can be established, which could be used in further design.

(7) The last and the most significant conclusion is that there ought to be better algorithms than those that have usually been used in the past. Since (a) there is at present little theoretical basis for the commonly used comparison metrics, and (b) most real terrain contains features beyond those describable by simple Gaussian statistics, it seems both reasonable and not inconsistent with theory to search for more efficient ways to carry out the initial map-matching or target-acquisition function. In particular, drastic preprocessing to extract special features of a given scene, using techniques currently being developed and exploited in the field of pattern recognition, should lead to more efficient algorithms.

CHAPTER 5

5. FUNDAMENTAL ASSUMPTIONS

5.1 The video signal

Our pictures were taken by a video-camera with a FOV of 3° in the vertical dimension and 4° in the horizontal, each line contains 340 pixels and each column contains 260 pixels. The video signals are composed of 256 grey levels, but in order to decrease the level of noise, we use an A/D converter of 6 bits, i.e. 64 grey levels, which is adequate regarding the information we get from such a sampler, since the pictures we are dealing with are characterized by uniformity, so that eventually we do not get a better resolution than 32 grey levels in any case. By using a 6-bit sampler our main advantage is an outstanding decrease in the noise level; the loss of information is negligible since above 64 grey levels the signals are anyway mixed with atmospheric noise, particularly in the cases when we observe the terrain from long ranges. In Fig. 3 (a) and (b) we show two

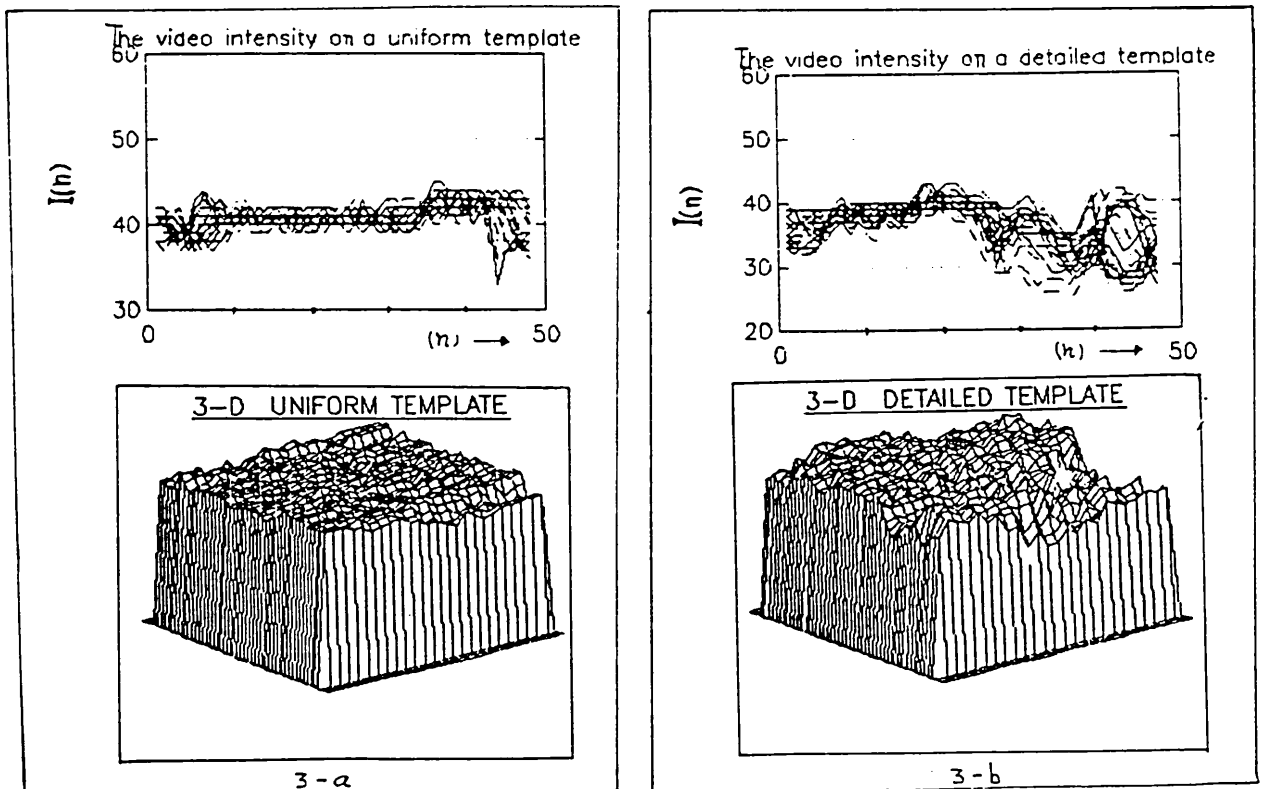


Fig. 3 The terrain 3-D illustration.

types of different templates, illustrating different areas of the same picture. Fig. 3-a depicts a template which has been taken from a very uniform region: we can see that the intensity function $I(n)$ is approximately constant in both directions, and this is illustrated clearly in the 3 dimensional graph. In such templates the likelihood for a successful matching is low. In Fig 3-b we show another template from the same picture, which clearly has very different qualities. Since the template in Fig 3-b is more detailed in both directions then the template in Fig 3-a, we would prefer template 3-b for the matching process. Our first aim in this stage is to propose an algorithm which will automatically find the most detailed region, called the 'best place', for locating a reference window. This process will be operated in two specific parts of the picture called the 'search areas' .

5.2 The search areas

Following the previous explanation and the system description in § 2 ,the inevitable conclusions are: (1) An automatic process in order to identify the 'best area' is essential, (2) the roll angle estimation has an overall priority over the other three unknown variables. A successful stabilization in rotation will directly simplify the estimation of the other variables, since the system model will become linear, at least for most of the vehicle trajectory .

In order to find the 'best place' for locating the reference-windows and to succeed in the roll angle estimation, we define two "search areas" (see Fig. 4) A_L , A_R on the left and right side of the FOV ,(their location is defined in picture

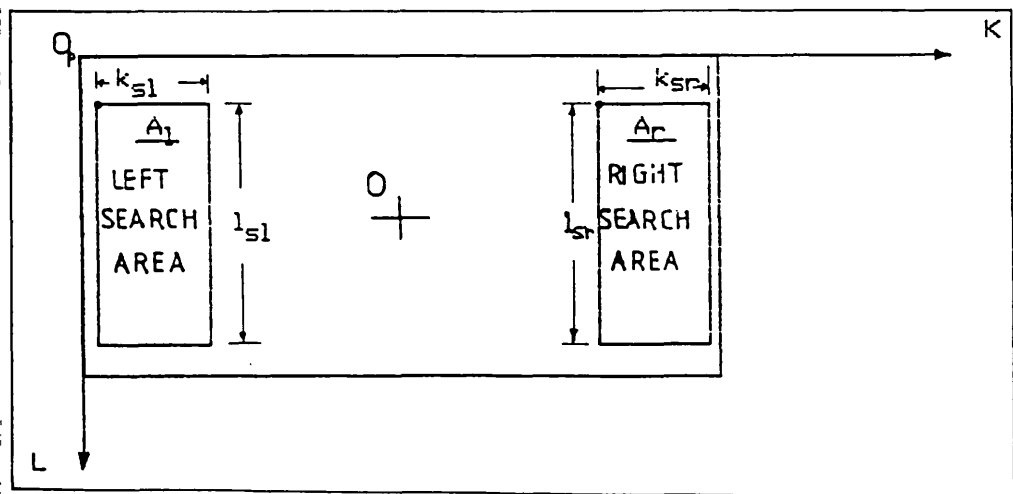


Fig. 4 The two search area in the FOV

coordinates $O_p(k,l)$ as shown in Fig. 4), aiming to find within them the 'best place' for locating the reference-windows. The basic idea where to locate the search areas is based on the fact that for high sensitivity in the roll angle estimation and in the scale-factor estimation process, we must locate the search-areas in both sides of the picture as far as possible from the FOV center, and in the horizontal direction (Fig. 4). In the vertical direction we might find the sky, and anyhow it might be a less detailed area;(see Appendix 4) therefore we prefer the horizontal direction. However by locating the search areas at remote distances from the FOV center,we are able to utilize the 'swing affect' in rotation since we get a significant displacement in the vertical direction even for small roll angles .

From the scale-factor aspect remote distances cause high sensitivity in the horizontal direction , because when the range between the camera and the observed terrain become shorter, the pixels will spread out faster. The further from the FOV center the chosen templates are and the more detailed part of the picture the templates cover, the more reliable the estimation process results will be . In order to avoid areas of nonlinearity and other distortions we define the search areas, 30 pixels from the outer boundaries of the FOV. Since we would like to get a resolution of 0.5° in rotation, the horizontal centre of the search area must be at least 100 pixels from the FOV centre, which leads to the decision to choose the search area size ($k_{s_{l,r}} \times l_{s_{l,r}}$) as 48×216 pixels. The search-areas dimensions in picture coordinates (O_p) are :

$$\begin{array}{ll} 30 \leq k_{sl} \leq 78 & 262 \leq k_{sr} \leq 310 \\ 30 \leq l_{sl} \leq 246 & 30 \leq l_{sr} \leq 246 \end{array}$$

The spin axis origin is the FOV center(O), since we assume that the camera has been calibrated properly and the misalignment is better then $1mR$.

3.3 The reference template

The reference template - $R_{l,r}$ is shifted inside the search areas $A_{l,r}$ in order to find there the most detailed region, (Fig. 4-a). Its size is a compromise between the desirability of a large number of pixels, which increases the probability (P_c) of correct matching [3] (particulary important for uniform pictures) and the necessity for real time processing which demands the lowest

possible number of pixels involved in the process. This is to be expected ,since the total signal-to-noise ratio represented by the sum of the contributions from each pixel in the reference-window is increased by either an increase in the pixel signal-to-noise ratio or an increase in the total number of pixels included in the reference-window. On the other hand by increasing the data-sample we increase the calculation burden which is of course undesirable from the aspect of real time.

The reference template - $R_{l,r}(j,i)$ size is defined as : $k_{r,l,r} \times l_{r,l,r}$ with indexes l,r for the left and right search-areas . $k_{r,l,r}$ and $l_{r,l,r}$ are chosen as 16×24 pixels which is a compromise between the above demands and other demands which will be discussed later on.

These templates - $R_{l,r}(j,i)$ are shifted vertically i_m times and horizontally j_n times, inside the search-areas A_l and A_r as indicated in Fig. 4-a

where : $i = 1.....i_m$ and $j = 1.....j_n$

and : $i_m = \frac{l_{s,l,r}}{l_{r,l,r}}$ and $j_n = \frac{k_{s,l,r}}{k_{r,l,r}}$

For each shift we operate a new algorithm , named as the Automatic Window Location (AWL) , in order to find the 'best place' inside the search-areas for locating the ref-template . This ref-template is defined as the reference-window which will be utilized later in the matching process.

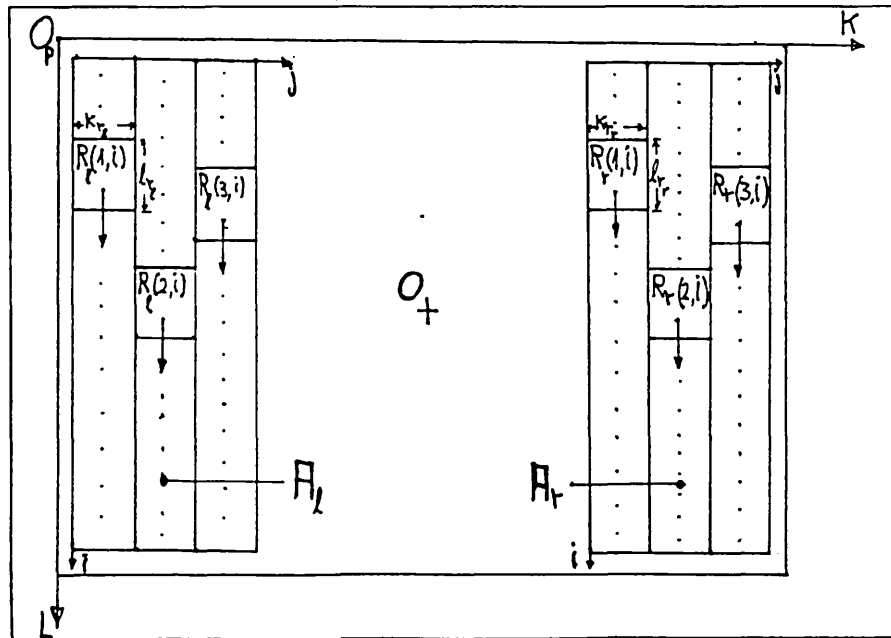


Fig. 4-a. The reference-templates - $R_{l,r}(j,i)$ inside the search-areas - A_l and A_r .

CHAPTER 6

6. THE AUTOMATIC WINDOW LOCATION ALGORITHM (AWL)

6.1 The AWL process

The AWL algorithm is based on two preliminary assumptions:

1. The 'best area' for locating a reference window is the most detailed area in the search areas A_l and A_r .
2. In pictures representing uniform terrains the differences between two successive pixels are not significant and could contain only noise .

Based on these assumptions we look for the 'best place' by using a gradient method in two dimensions, in other words we take the sum of $|\nabla_{h,v} I(k,l)|$, in every sub-region defined by the size of the reference-window. $\nabla_{h,v}$ is the two dimension differential operator and $I(k,l)$ is the video intensity around point (k,l) . The main idea is that the maximum of the sums represents the most detailed area. We operate this process along the search area by shifting the reference window every x_h pixels in the horizontal direction and x_v pixels in the vertical one, inside the search areas . However since on the one hand gradient methods are very sensitive to noise and on the other, according to assumption 2, there is no significant difference between two successive pixels , we transform the each template composed of regular pixels to a new template composed of 'huge pixels' called "hpix". This transformation is done by extending the Sobel filter [1] to two dimensions. The intensity of hpix is basically a result of a weight function which depends on the neighbouring pixels.

An illustration of the way we build the extended Sobel filter is shown in Fig. 5 , where the size of the circles represents the relative weights put on the intensity of the corresponding pixels (see Eq (6.1-1) and (6.1-2)).

The advantages of using this method are both a decrease in the noise level, and a real modification of the video intensities around (k,l) which does not absorb the interesting features inside the template. By operating the extended Sobel filter we give a real proportional weight to the vertical and to the horizontal vicinities around an arbitrary point (k,l), which allows us to use a gradient method without the fear of increasing the noise effect on the results .

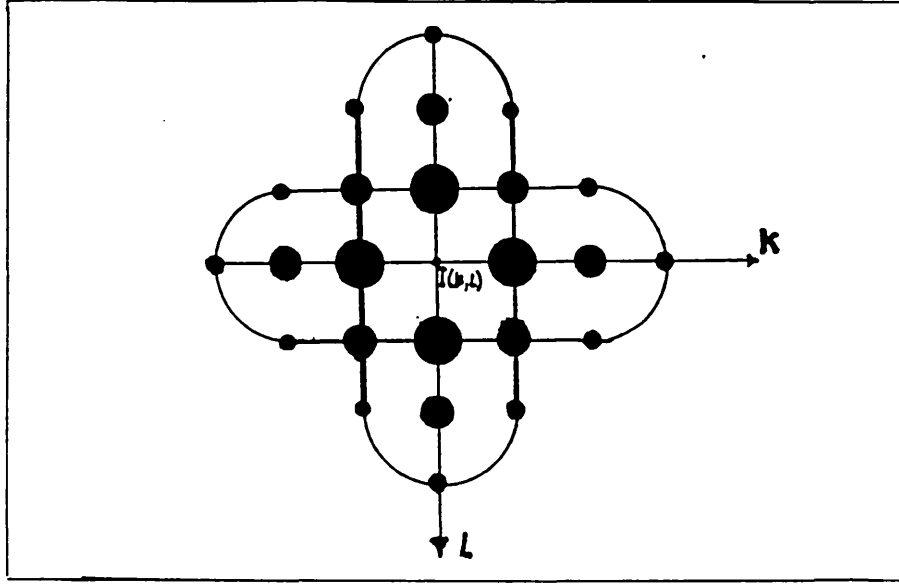


Fig. 5 An illustration of the Extended Sobel Filter

Let $I_h^{(+)}(k,l)$ represent the weighted average intensity in the horizontal area to the right of pixel (k,l) and $I_h^{(-)}(k,l)$ represent the same thing on the left of pixel (k,l) then :

$$I_h^{(+)}(k,l) = 2 \cdot I(l-1,k+1) + 3 \cdot I(l,k+1) + 2 \cdot I(l+1,k+1) + \\ I(l-1,k+2) + 2 \cdot I(l,k+2) + I(l+1,k+2) + \\ I(l,k+3) \quad (6.1-1)$$

$$I_h^{(-)}(k,l) = 2 \cdot I(l-1,k-1) + 3 \cdot I(l,k-1) + 2 \cdot I(l+1,k-1) + \\ I(l-1,k-2) + 2 \cdot I(l,k-2) + I(l+1,k-2) + \\ I(l,k-3) \quad (6.1-2)$$

$$\nabla I_h(k,l) = \frac{1}{N} \times (I_h^{(+)}(k,l) - I_h^{(-)}(k,l)) , \quad (6.1-3)$$

where N is the total number of pixels in the reference window.

The same process is carried out in the vertical direction and $\nabla I_v(k,l)$ is :

$$\nabla I_v(k,l) = \frac{1}{N} \times (I_v^{(+)}(k,l) - I_v^{(-)}(k,l)) \quad (6.1-4)$$

Then we take the sum of the gradients in the horizontal and vertical directions for each template - $R_{l,r}(j,i)$, (see Fig. 4- a).

$$S_h(j,i) = \sum_{l=1}^{l_{rw}} \left(\sum_{k=1}^{k_{rw}} (|\nabla I_h(k,l)|) \right) \quad (6.1-5)$$

$$S_v(j,i) = \sum_{k=1}^{k_{rw}} \left(\sum_{l=1}^{l_{rw}} (|\nabla I_v(k,l)|) \right) \quad (6.1-6)$$

$$S(j,i) = S_h(j,i) + S_v(j,i) \quad (6.1-7)$$

We perform this process J1 times in each search area, where J1 is :

$$J1 = \text{int} \left\{ i_m \times j_n \right\} = \text{int} \left\{ \frac{l_{s_{l,r}} \cdot k_{s_{l,r}}}{l_{r_{l,r}} \cdot k_{r_{l,r}}} \right\} \quad (6.1-8)$$

$$S_{\max}(K_a, L_a) = \max \left\{ S(i, j) \right\} \quad (6.1-9)$$

The coordinates of $S_{\max}$, k_a and l_a are the axis origin of the reference-window in picture-coordinates (O_p), (see Fig 4-a). The connections between picture coordinate (K, L) and template coordinate (j, i) are :

$$K = 30 + k_r \times (j-1) \quad (6.1-9a)$$

$$L = 30 + l_r \times (i-1) \quad (6.1-9b)$$

This process is performed in both search areas in parallel, but it is not performed in real time since there is no rapid change in the FOV. The updating of the AWL algorithm will be adaptive to the matching quality: as soon as the matching quality is degraded the control unit will operate the whole AWL process in order to find a new 'best place'. This process will be discussed later.

The whole process is shown in Fig. 6

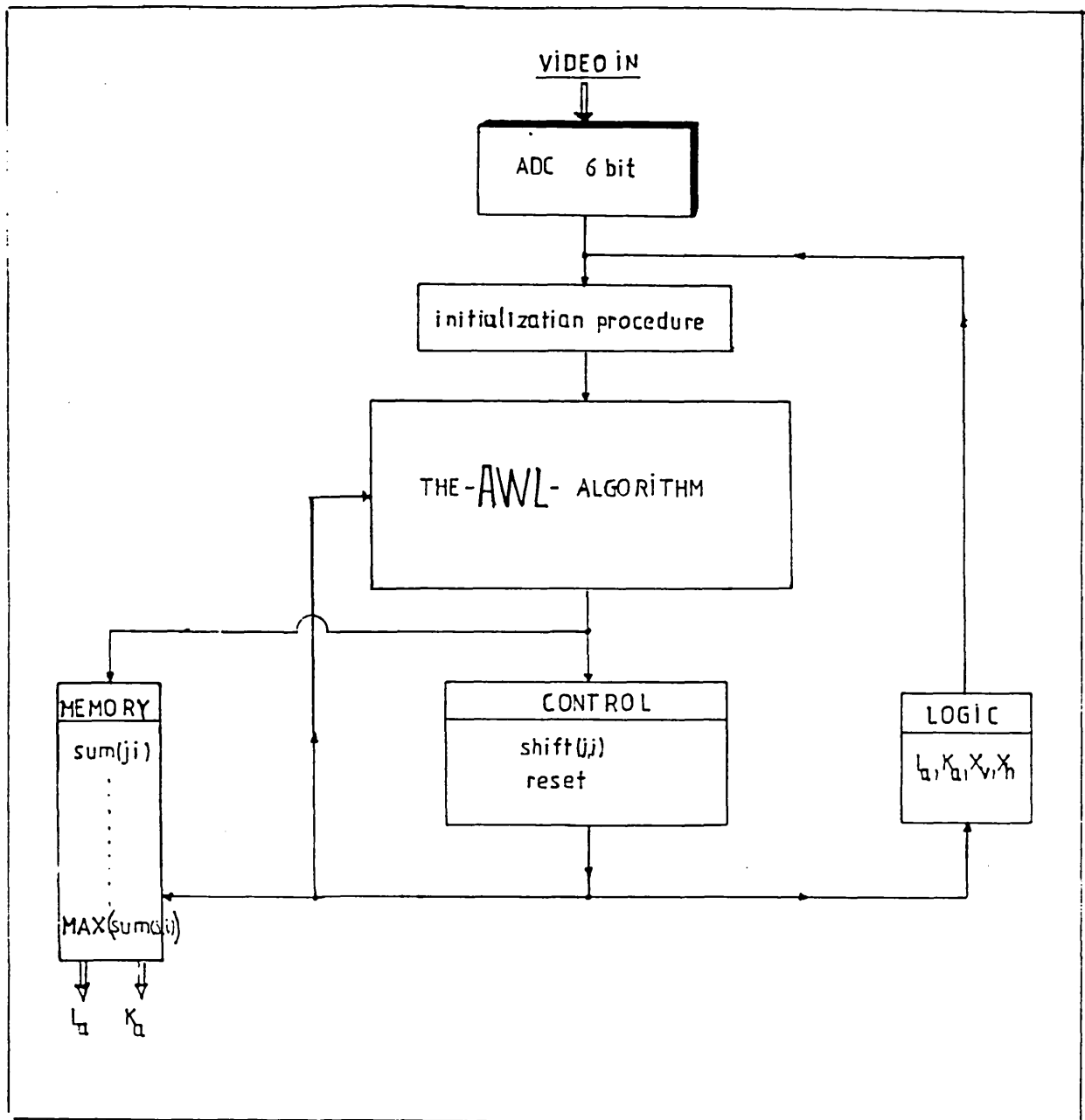


Fig. 6 The AWL algorithm block diagram

The block diagram describes the whole AWL procedure. The initialization procedure deal with the axis origin coordinates of the search areas in both side of the picture (L_a, K_a); these coordinates might be changed as a function of the matching process degradation, this decision being taken in the logic unit. The AWL unit performs the whole process, include the hpix definition. The control unit organizes the whole process as shown in Fig.4-a and the AWL output is stored in the memory unit. L_a, K_a are the final result for the best location.

6.2 The AWL performance

The AWL algorithm is operated on the three pictures shown in Fig. 1. We have three sorts of pictures: picture (c) which was televiewed from a short range, about 0.5 (Km) . picture (a) which was televiewed from a medium range about 4 (Km) and picture (b), which was televiewed from a long distance , about 10 (Km). It is obvious that in picture (c) the detailed area is well defined , while in picture (a) , it is more complicated , bearing in mind that we are confined to the search-area .

In Fig. 7 we show results of the AWL algorithm (represented by the function $S(j,i)$) operated on the left search area (A_1) of picture (c).

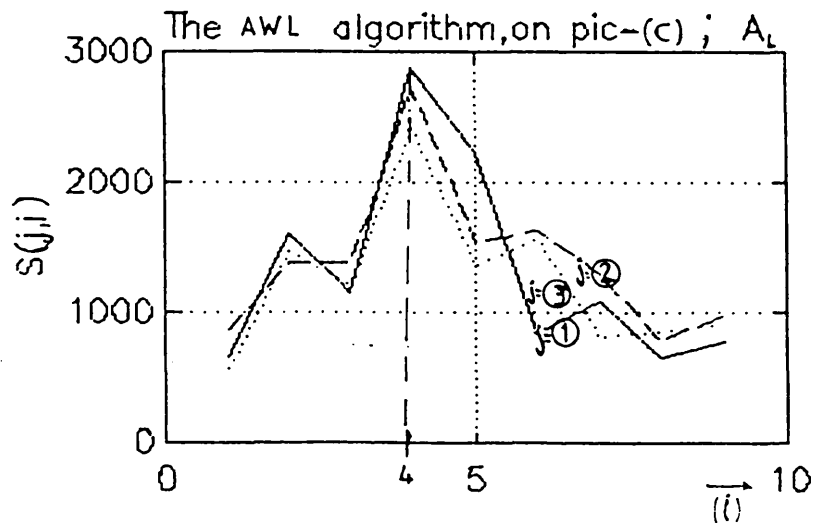


Fig. 7

We can see an outstanding maximum of function $S(j,i)$ in location $R_1(1,4)$, which means ,that in picture coordinates (O_p), (see Fig. 4-a) the maximum is located in (30,102) (see § 6.1 , Eq 6.1-9a and 6.1-9b , by replacing $k_r = 16$ and $l_r = 24$). However this graph gives us more information: for graph $j=1$ we can see that the horizontal vicinities of the peak value $S(1,4)$,i.e., $S(2,4)$ and $S(3,4)$ are detailed templates as well , but its vertical vicinity e.g., $S(1,3)$ is much less detailed while the other one $S(1,5)$ is quite a detailed one. Such cases might lead to the next conclusion, if the peak value of $S(j,i)$ is received at coordinates (j_m, i_m) and :

$$S(j_m, i_m) \simeq S(j_m, i_m \pm 1) \quad (6.2-1)$$

then the coordinates of the reference window will not be at (j_m, l_m) but at $(j_m, i_m \pm \frac{l_{1,r}}{2})$. This criterion ensures us that for any disturbance, the current window which is a part of the reference window (this issue will be explained latter on) will be located in the 'best place' possible. The same approach can be implemented in the horizontal direction ($i=\text{constant}$). The maximum of $S(1,i)$ in Fig. 7 is 2877 which means that this is a relatively detailed area. From this picture and from others, we get a criterion for the terrain quality, regarding the detail it contains. The results are summarized in table-1.

sum(j,i)	Terrain character
$S(j,i) \leq 1200$	very poor terrain
$1200 < S(j,i) < 2100$	medium terrain
$S(j,i) \geq 2100$	detailed area

Table-1

These results lead us to define a picture quality factor (PQF) and that might be very important in the matching process regarding the real time constraints. We can actually update the number of pixels involved in the matching process according to the PQF. We decrease the number of pixels involved in the matching process if PQF increases and vice-versa. In order to define the PQF let us look at the way graph 1 in Fig. 7 is composed. We show in Fig. 8 $S_h(1,i)$ and $S_v(1,i)$ which are the components of $S(1,i)$, shown in Fig. 7.

The horizontal(S_h) and vertical(S_v) components of $S(1,i)$

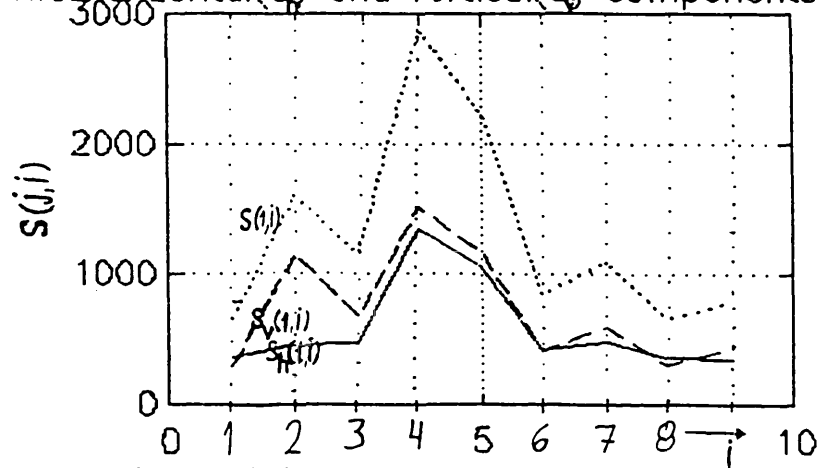


Fig. 8 $S(1,i)$ and its components

We can see that the maximum of $S(j,i)$ is reached when the maximum of its components S_h and S_v are reached , which means that the chosen template in picture (c) ,is not only the 'best place' , but it has the same quality in the horizontal and vertical directions. This result is not a surprise, since the pictures we are dealing with represent natural uniform terrains and therefore the likelihood of finding a significant difference between $S_h(j,i)$ and $S_v(j,i)$ is low. But small differences are possible and they are very important i.e., the way the pixels are distributed might affect the decision about the quantity of the sample-data involved in the matching process. Bearing in mind that rotation might occur while the reference-window is located near to the line of 45° compared to the horizontal axis (while the origin axis is in O, Fig. 4), which might cause a sharp displacement in both directions in the horizontal and the vertical one. In such a case the decision whether to use maximum data in the matching process or making an attempt to save calculation time by skipping every other pixel along the columns or the lines is highly dependent on the way the detailed area is composed. When we shift two templates relative to each other and we reduce the number of pixels involved we at the same time increase the likelihood of false matching (as we will show in detail later). Therefore we will modify the criterion assumed in (6.1-9) in a way that will take into consideration the directional distribution of the pixels in the templates, by defining a picture quality factor (PQF). The PQF is defined in terms of vectors, means amplitude $A(j,i)$ and phase $\Theta(j,i)$:

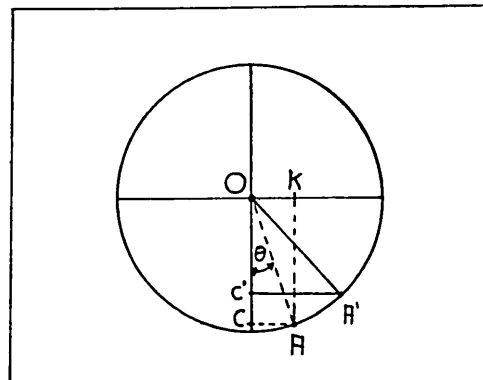
$$A(j,i) = \sqrt{\left\{ \left(\frac{S_h(j,i)}{1000} \right)^2 + \left(\frac{S_v(j,i)}{1000} \right)^2 \right\}} \quad (6.2-2)$$

$$\theta(j,i) = \tan^{-1} \left(\frac{S_h(j,i)}{S_v(j,i)} \right) \quad (6.2-3)$$

$$PQF(j,i) = A^2(j,i) \cdot \sin(2\theta) \quad (6.2-4)$$

$$PQF_{\max}(k_a, l_a) = \max \{ PQF(j,i) \} \quad (6.2-5)$$

Fig. 9. The PQF definition :



Eq-(6.2-5) represents the area of a rectangle OCAK shown in Fig. 9. The PQF_{max} gives the optimum place for locating the reference window, and the reason is because when the ref-window is located at the edges of the search area, even a small roll angle might shift the current template in vertical and horizontal direction, therefore we prefer to locate the ref-window in a region that is detailed in both direction rather than in a region which is very detailed in one direction but less in the other one. In order to select the right location we suggest the criterion shown in Eq-6.2-4. It is assumed that since we deal with natural terrains represented by pictures like Fig. 1 and others like in Appendix-1, we do not get templates similar to the pictures in Fig. 10

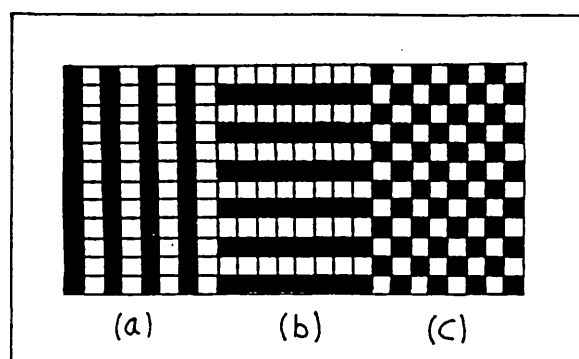


Fig. 10 The model of synthetic pictures

This picture can be modeled in the next way ; for case (a)

$$\begin{aligned}
 I(l,k) &= I(l+1,k) = \dots\dots\dots I(l+m,k) = I_v \quad \text{for all } l, \quad k=1,3,5,\dots,2n-1 \\
 I(l,k) &= I(l+1,k) = \dots\dots\dots I(l+m,k) = I_h \quad \text{for all } l, \quad k=2,4,6,\dots,2n
 \end{aligned}$$

I_v and I_h are different arbitrary video intensities. In the same way we can define the model of (b) and (c). Theoretically if we operate our decision criterion (Eq 6.2-4) in Fig. 10, it will prefer case (c), although (c) is a 'bad-case' just like the others. In order to eliminate even a slight chance of templates like the one in (c), we use the extended Sobel filter(ESF) (see §6.1) which gives a different weight to each area inside the template and eventually we get 'huge pixels' instead of ordinary pixels. This might help to reduce the likelihood of an accidentally small regions with the same properties as shown in Fig. 10.

Another point is the fact that we operate the algorithm in quite big templates , so there is a very low possibility to find there pixels which are organized in the same way as illustrated in Fig. 10.

As we explained previously, usually there will not be an outstanding difference between the results from Eq 6.1-9 and the results from Eq. 6.2-5 regarding the location decision (the AWL algorithm). But the decision about the sample-data quantity for a successful matching process, that might require a more sensitive algorithm in order to be able to identify the template's quality.

In the case of very poor terrain , described in tab-1(§ 6.2), $\text{sum}(j,i) \leq 1200$, the difference between the two methods is not significant since the picture is uniform in all directions , but when we are dealing with a more detailed picture , the differences between the two methods might be significant as it is shown in Fig. 10-a.

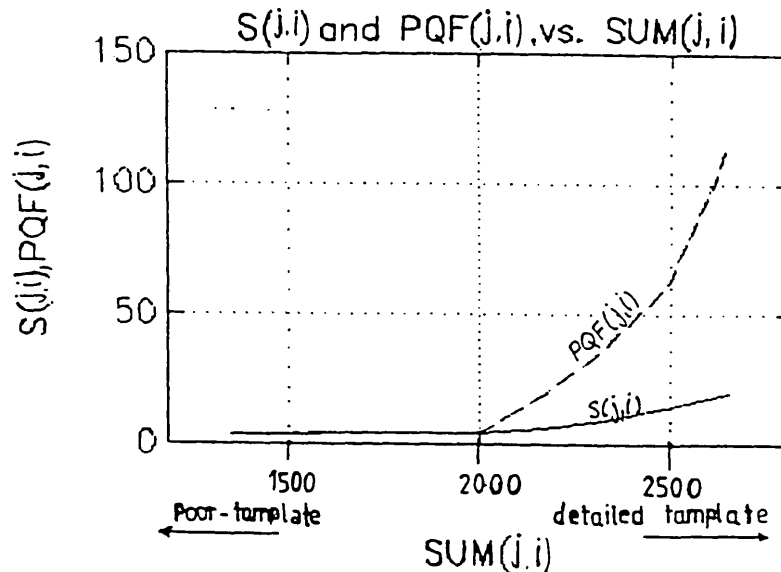


Fig 10-a. $S(j,i)$ and $PQF(j,i)$.vs. the picture qualitys

We can realize that the decision based on $PQF(j,i)$, is more robust then the one based on $S(j,i)$, therefore we choose the PQF function as the criterion for the reference window location and for the matching process as well .

We should stress that the analysis of the AWL process, must anticipate the matching process needs as well as the estimation algorithm desirable features ,

which means that the output of the AWL process must be significant. In Fig. 11. we show a picture which demonstrates the whole process. The two search areas are marked as A_L and A_R and the reference windows are located in the right place after the AWL process had been operated .

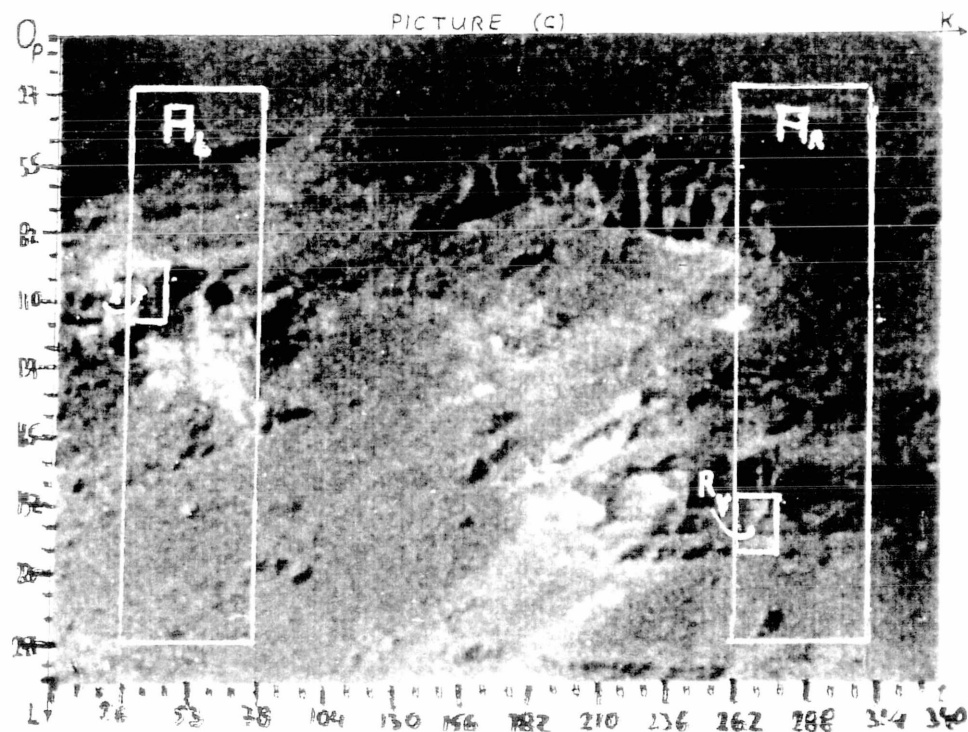


Fig. 11. Picture (c) - The search areas A_L , A_R and the ref-windows , R_L , R_R

It is interesting to examine the chosen locations for the reference windows , and the best way to do this is to look to their histograms.

In Fig. 12. we compare two histograms describing two different reference-windows ,the first describes the choosen template $R_L(1,4)$ (see Fig. 7) and the second describes template $R_L(1,6)$, both from the same left search area- A_L of picture (c). We can see that the template chosen by the AWL algorithm as a ref-window($R_L(1,4)$) includes quite a wide range of video intensities actually 18 out of 64 grey levels comparing with only 7 grey levels in $A_L(1,6)$. These results show the robustness of the AWL algorithm , since in templates with a small range of grey levels the likelihood to find a similar regions is very high which means that the matching process is impossible and vice-versa .

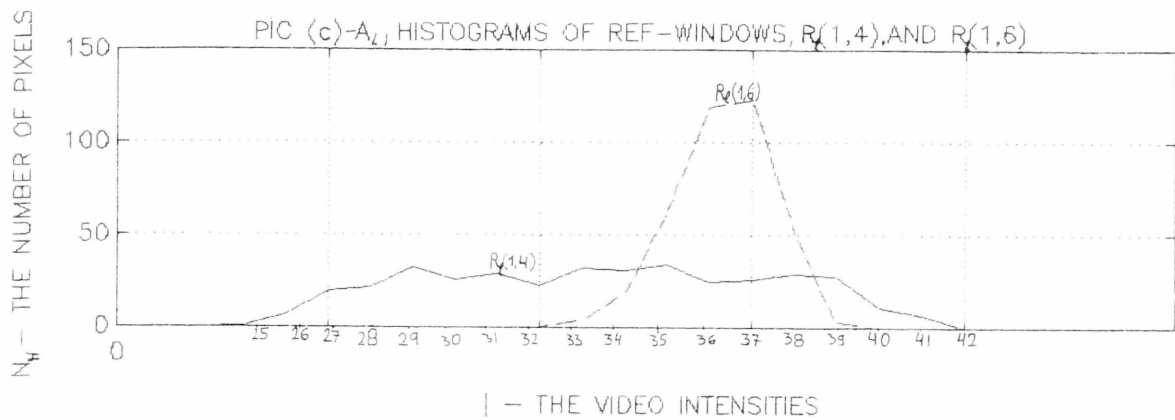


Fig. 12.

Based on the previous discussion it is clear that we cannot use segmentation technique and then sharpen several regions in the template, since our histogram has no peaks and therefore it is impossible to find such a threshold that will separate the peaks.

In order to demonstrate the robustness of the results coming from the AWL algorithm, we choose the left search-area (A_1), in picture (b) (see Fig. 1), which according to Tab-1 is a poor template and operate the AWL algorithm on A_1 . From Fig. 13 it is obvious to see the remarkable matching between the graph and the picture itself.

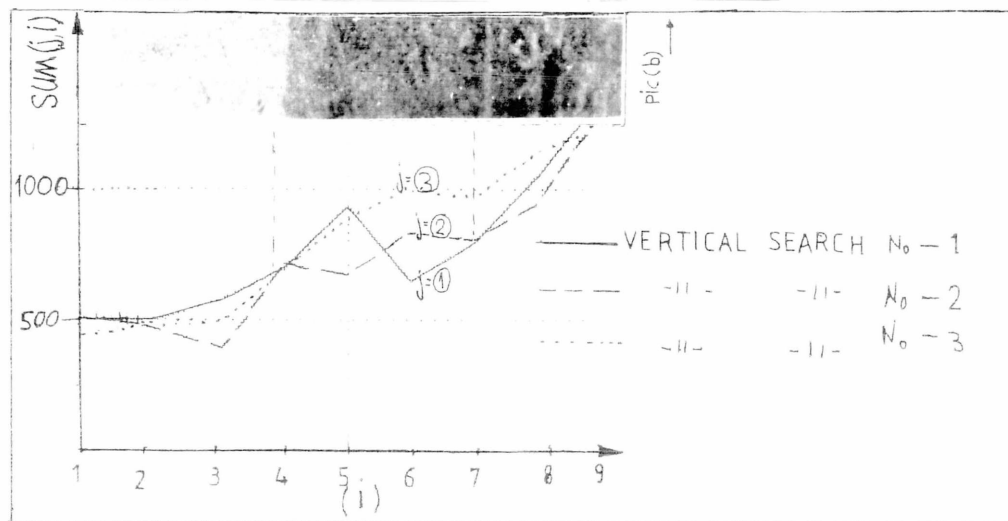


Fig. 13. The comparison between the AWL results and the left template in picture (b)

Basically the template is divided into three parts :

1. The sky - a very uniform region.
2. The mountains - a uniform region.
3. The flat area - a less uniform region

This division can be seen better in Fig. 1-b , where the whole picture is shown. The three graphs show a gradual improvement as we move towards the bottom of the picture which might be observed easily from the picture itself ; it is not so easy with other pictures we work with .

What is very interesting is the fact that even along the region which represents the sky there are differences in the $\text{sum}(j,i)$ function between the three graphs. Each represents a vertical search along the search-area and since the search-area width is 3 times that of the ref-window width , we get the three graphs in Fig. 13. We might expect quite constant values for the $\text{sum}(j,i)$ graphs in Fig. 13 particularly for this very uniform picture (b) and because we use the extended Sobel filter as a pre-filter for the AWL process, but in fact the results are not like that and the AWL process seems to be a sensitive one. The reason for this remarkable result is basically the combination between the gradient method and the weighted filter(ESF). The ESF is chosen as prefilter to the AWL process since when the Signal to Noise ratio (S/N) is very low, even very small differences among the video intensities can distort the expected results, in particular when gradient methods are implemented. Another important property of the ESF is its ability to maintain the video-intensities distribution since it is a weighted one.

From the fact that we get differences in the 3 graphs of Fig. 13 regarding the 'part of sky' , we might conclude that variations on $\text{sum}(j,i)$ less than 20 percent are not significant, and can be caused by noise or by an accidental distortion .

In Fig. 14. we can see the various histograms represent different types of terrains and the sky . The sky histogram is different even from the poor terrain, in particular in the fact that it includes only two or three grey levels while the very uniform terrain (called, poor terrain in Tab-1) includes at least 10 grey level in the same size of templates . That is the AWL robustness: though the terrain is very uniform it is still possible to find a region that includes more than just a few grey levels . Another conclusion from Fig. 14 is that usually for increasingly uniform templates the grey levels tend to shift towards the high frequencies , namely the white region. Grey levels of 42-43, out of 64 represent this region.

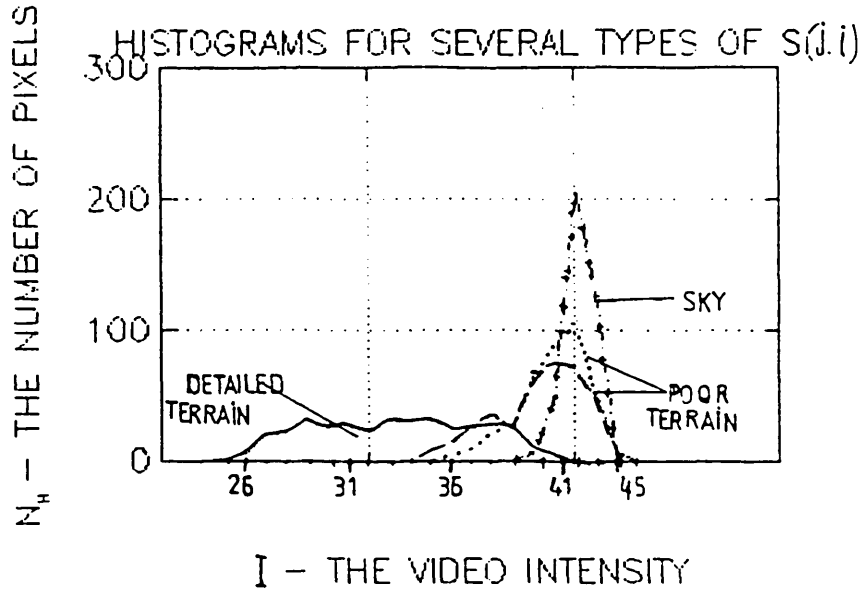


Fig. 14

6.3 The AWL sensitivity to noise

Low sensitivity to noise in the AWL process is vital, since otherwise the location of the reference-window is not necessarily the 'best place' in the search areas.

It is well known that gradient methods are sensitive to noise in particular when the frequency spectrum is narrow, and that is precisely what happens in uniform pictures (see Fig. 14.). Usually pictures are subject to many different types of noise like electrical sensor noise, channel noise, atmospheric transmission and others [7]. In our case when the terrain is scanned by a vidicon television camera it is assumed that the noise is uncorrelated and has zero average value [1],[2]. Therefore we might consider an arbitrary video intensity $I(l,k)$ which is formed by the addition of noise $\eta(l,k)$ to the original intensity $f(l,k)$:

$$I(l,k) = f(l,k) + \eta(l,k) \quad (6.3-1)$$

The differences between two successive pixels are:

$$\delta(l,k) = I(l,k) - I(l,k-1) + \eta(l,k) \quad (6.3-2)$$

In uniform templates it is easy to find that:

$$I(l,k) - I(l,k-1) \sim \eta(l,k) \quad (6.3-3)$$

and that certainly might distort the results from the AWL process!

One way to alleviate the problem is by using a new template $\hat{I}(l,k)$ which is formed by averaging N different noisy templates $I(l,k)$:

$$\hat{I}(l,k) = \frac{1}{N} \sum_{i=1}^N I_i(l,k) \quad (6.3-4)$$

Then it follows that the expected value of $\hat{I}(l,k)$ is :

$$E\{ \hat{I}(l,k) \} = f(l,k) \quad (6.3-5)$$

and the variances of $\hat{I}(l,k)$ and $\eta(l,k)$ are :

$$\sigma_{\hat{I}(l,k)}^2 = \frac{1}{N} \sigma_{\eta(l,k)}^2 \quad (6.3-6)$$

where the standard deviation at any point in the average template $\bar{I}$ is :

$$\sigma_{\bar{I}(l,k)} = \sqrt{\frac{1}{N}} \sigma_{\eta(l,k)} \quad (6.3-7)$$

From equation (6.3-7) we can see that as we increase N , $\sigma_{\hat{I}(l,k)}$ is decreased which means that $\hat{I}(l,k)$ is close to $f(l,k)$ with high probability. But there are two main reasons why we do not use this method: The first is that since we do not know *a priori* the ref-window location, we will have to register to the frame-buffer N times the whole search area, and only afterwards begin the AWL process. This will take a lot of memory capacity and calculation time. Secondly, this process does not take into consideration the sort of picture we are dealing with. The principle difficulty in pictures which represent uniform terrain is the lack of information from two successive pixels and an average filtering does not help in this case. We use instead the extended Sobel filter (ESF).

Another way to filter most of the noise is by operating a sampler of 6 bit (64 grey levels). By using only 64 grey-levels we lose information which is basically noise, the reason being that usually pictures which represent deserts and similar terrains, are composed of less than 64 grey levels.

There is a significant point which makes the noise filtering so important regarding the AWL process and that is the fact that, once we choose the 'best place' for locating the reference window, it will remain in the same place for several video frames (at least 6 frames of 33 msec as we will show later) and if its location is affected by random noise, the whole matching process might fail.

A comparison between the AWL algorithm without pre-filtering (i.e., replacing $I_h^{(+)}$, $I_h^{(-)}$ in (3) by $I(k,l)$, $I(k,l-1)$ etc.) and the same algorithm with the Extended Sobel Filter (ESF) is shown in Fig. 15 where we operate both methods on a uniform template. It is obvious that even without noise the use of the ESF gives us an advantage: we got an outstanding maximum which has not been identified by the AWL algorithm without the ESF. Furthermore the maximum value is almost 30% higher than the maximum reached by the AWL algorithm without the ESF.

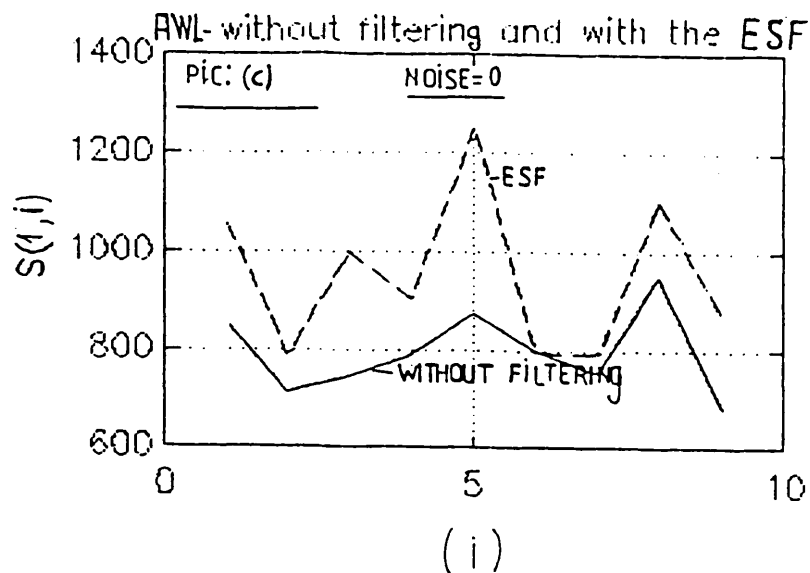


Fig. 15

In Fig. 16-a and 16-b we examine the sensitivity of the AWL algorithm to noise, which is actually a pseudo random normal distribution, with a parametric standard deviation $S.D = 1, 2, 3$. In Fig 16-a we show high sensitivity to noise while we do not operate the ESF, although there is no change in the location of the maximum, there is still a significant effect on the results of the AWL algorithm. In Fig 16-b we can see an outstanding improvement when we operate the ESF: the changes in $S(1,i)$ caused by the noise are much smaller in particular around the maximum area which is a very important fact, since it means that the decision of the 'best place' is invariant to noise.

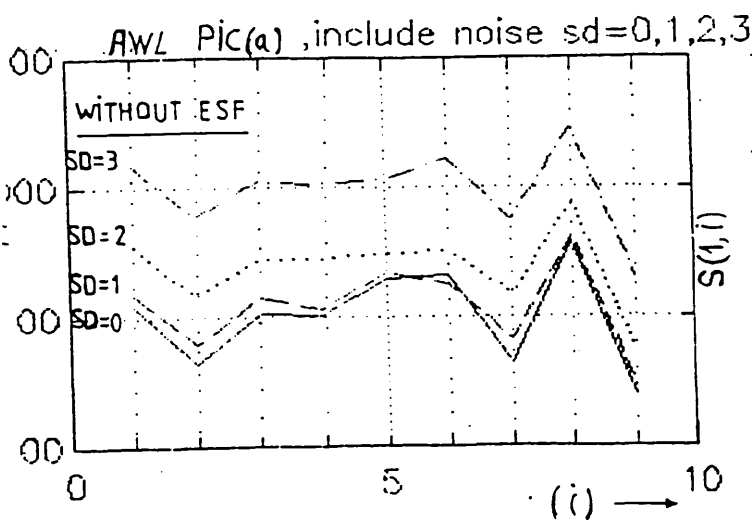


Fig. 16-a

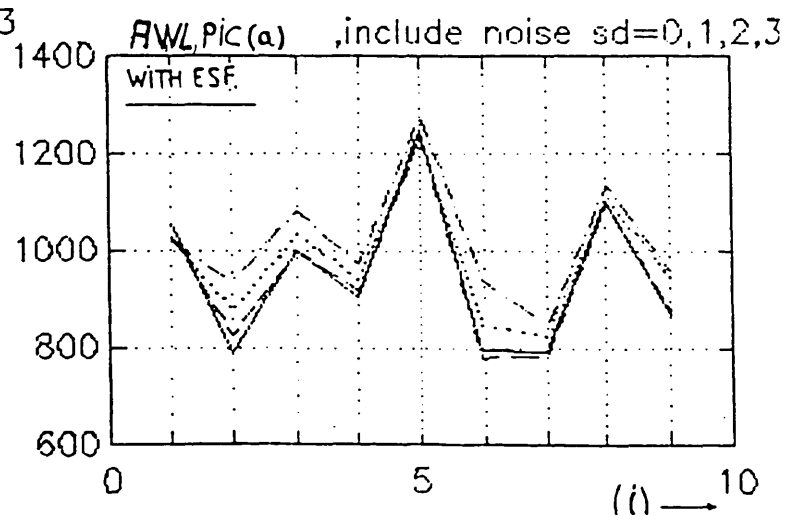


Fig. 16-b

6.4 The AWL algorithm sensitivity to the data-sample

One of the basic parameters in the AWL process is the decision about the data-sample quantity, i.e. the size of the reference-windows $k_{r,l,r}$ and $l_{r,l,r}$. Their minimum size is determined by the maximum displacements expected from the video camera shifts between frame to frame. We will show later that this is about ± 12 pixels in the vertical dimension and ± 8 pixels in the horizontal one, which leads to $k_{r,l,r} = 16(\text{pix})$ and $l_{r,l,r} = 24(\text{pix})$. From the matching aspect regarding the S/N factor, and according to reference [3] the probability of correct matching (P_c) increases with an increase of the elemental signal-to-noise ratio, and with an increase in the size of the data-sample, since the total signal to noise ratio represented by the sum of the contribution from each pixel is increased by either an increase in the pixel signal to noise ratio or an increase in the total number of pixels, which means an increase in the ref-window size. But from the real time aspect it is obvious that a smaller size reference window is desirable. Another important parameter is the effect of the window size on the ability to identify the 'best place'. In Fig. 17 we show the PQF, which is a robust criterion regarding the quality of the chosen template as the 'best-place'.

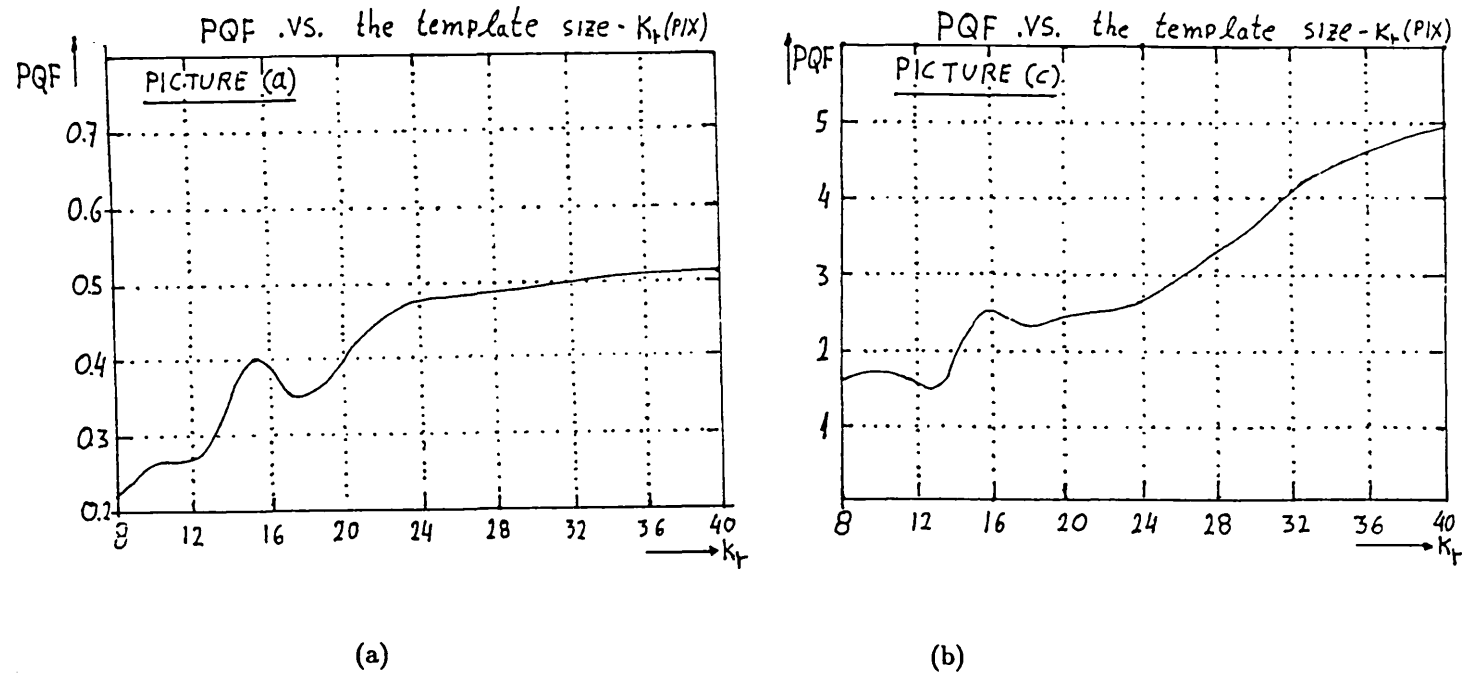


Fig. 17

The results in Fig. 17 are shown for poor terrain picture (a) , and for more detailed terrain picture (c). (Note the different vertical scales on the two graphs.) In both cases, particularly for the more detailed regions (picture (c)), increasing of the data sample ,by increasing the horizontal size of the reference window - k_r , cause an increase in the PQF , (all the results are normalized to the templates size) .These results are in agreement with those in [3] which means that in pictures representing deserts or other uniform terrains the bigger the templates are the higher the probability for detailed regions is .

But another conclusion from Fig. 17 (a) and other tests we have made is, that there is a finite size for each uniform picture so that above this size it is useless to increase the template size, we can see that for the case in Fig. 17 (b) the gradient of $PQF(k_r)$ is decreasing after a certain size of the template (in our case , for $k_r = 32(\text{pix})$, l_r for all cases is $24(\text{pix})$) , the reason is since by increasing the size of templates in the case of very uniform pictures we inevitably include inside the templates regions which are totally 'flat' , which means those regions do not add any information except of noise. Therefore we strongly recommend not to increase the templates size much above the size needed for covering the maximum displacements expected, bearing in mind that an unnecessary increasing in the template size will increase the calculation burden which might cause realtime problems. A reasonable value for PQF in order to ensure a successful matching is $PQF_{SM} = 2 - 3$ (As we show in the next chapter). So for very uniform pictures like picture (a), we cannot reach this values of PQF_{SM} anyway. Therefore it is useless to increase the template size above $k_r = 16(\text{pix})$ which is needed for covering the displacements of the current window. For less uniform pictures like (c) we don't need to increase the template size above $k_r = 16(\text{pix})$ since with this size we already reached the PQF desirable value - PQF_{SM} .

CHAPTER 7

7. THE MATCHING PROCESS

One of the basic ideas in matching techniques is to make an attempt to emulate the human visual system (HVS) (see Appendix 3) , by matching two successive video frames and finding the differences between them. While the HVS is able to identify almost any displacement, but not to give a precise measure of the displacement , the normal requirement from an “artificial intelligence” system is to identify the displacement and to give a precise measurement as well .

It is quite clear that the overall operation of the HVS system is not totally understood , but certain basic principles of operation are known e.g., several of the “Gestalt laws” and these may provide useful design guidelines . Two of the Gestalt laws are of particular importance: the ‘organization law’ and the ‘common fate’ law . The first law means that areas with common properties are most naturally seen as a unit, and the second that if a collection of parts move in harmony , we tend to see them as a single figure. The importance of the first law concerning uniform pictures is the understanding that in order to be able to extract “features” from a uniform picture we must find a template containing small sub-areas which have differing properties, otherwise the whole template will be seen as ‘one unit’. This has been precisely implemented in the AWL algorithm . The second principle is used in the matching process , since we may assume that any part of the template has roughly the same displacement, and this facilitates the matching process. Of course the HVS has many other factors as well e.g. , the extremely large amount of *a priori* information which has been acquired by a long term learning process. Clearly it is impractical to incorporate memory sufficient for this data and then to utilize it in real time as the HVS does. Another important HVS property is the ‘background factor’ i.e., without differences between the background and the observed area no displacements could be identified. This property has a significant use in the matching process: when we get two templates, one is used as a reference-window and is located at the ‘best place’ as we explained previously. This reference window is actually the background. The other template is used as an actual window (or current-window) and represents the observed area. In order to be able to identify the displacement between the actual-window and the reference one we must have enough differences between the two; that is the reason that the actual window will be a part of the reference one ,

since it is located in the most detailed area which increases the likelihood of differences between the shifted current-window and the reference one.

7.1 The current window

The current window is a part of the reference window . We have two current windows, on the left and on the right . In the initialization of the matching process both windows are located as shown in Fig. 18.

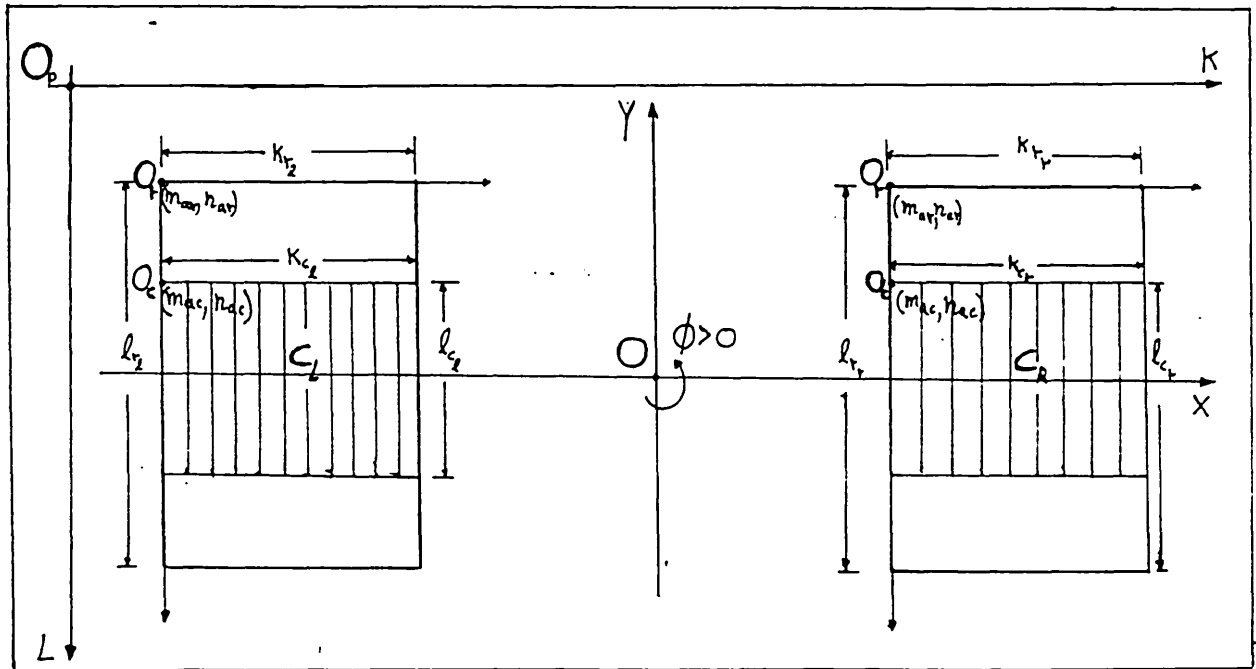


Fig. 18. The reference and the current windows

The horizontal size of the current window $k_{c_{l,r}}$ is equal to the horizontal size of the reference window namely $k_{c_{l,r}} = 16(\text{pix})$, the vertical size of the current window is , $l_{c_{l,r}} = 16(\text{pix})$. The current window is updated every video frame at a rate of 60 frames/sec and its axis origin (O_c) is located at the upper left-hand side of the current-window. For each frame we sample the current window , where its origin coordinates $O_c(m_{ac}, n_{ac})$ in FOV coordinates ($O_p(L,K)$) are :

$$ma_c = 30 + ma_r + \frac{l_{r,l,r}}{2} \quad (7.1-1)$$

$$na_c = 30 + na_r \quad (7.1-2)$$

Here ma_r and na_r are the coordinate of the reference window origin (O_r) relative to the FOV origin (O_p) and are calculated in the AWL process. As we can see from Eq 7.1-1 and Eq 7.1-2 the current window coordinates are changed only if there is a change in the reference window location, which happens if there is a decrease in the matching quality, (this issue is discussed later on), but relative to the reference window the current window is sampled always in the same place.

(As described in Fig. 18) .

The size of the current window 16×16 (pixels), is defined by the fact that the maximum displacement caused by the worst case of the disturbances is: ± 8 (pixels). Another factor is the necessity for a significant overlapping (as we show later) particularly in the vertical direction .

7.2 The matching problem

Our main task is to estimate the vector displacements $D(\hat{\epsilon}_{v,l,r}, \hat{\epsilon}_{h,l,r})$ which is caused by 4 disturbances :

- (1) Rotation (ϕ)
- (2) Vertical drift (d_v)
- (3) Horizontal drift (d_h)
- (4) Scale-factor (sf)

The mathematical model describing the connection between the vertical and horizontal displacements $\epsilon_{v,l,r}(i)$ and $\epsilon_{h,l,r}(i)$ and the disturbance is :

$$\epsilon_{h,l,r}(i) = sf_i \times [k(i) - d_h(i)] \times \cos(\phi_i) - sf_i \times [l(i) - d_v(i)] \times \sin(\phi_i) \quad (7.2-1)$$

$$\epsilon_{v,l,r}(i) = sf_i \times [k(i) - d_h(i)] \times \sin(\phi_i) + sf_i \times [l(i) - d_v(i)] \times \cos(\phi_i) \quad (7.2-2)$$

where sf_i is the scale-factor in each frame. The scale-factor is a variable which represents the 'blow up' phenomenon, caused by the camera approaching the observed terrain , the pixels around the centre of the FOV begin to move towards the edges. The pixel displacement rate is very

low at the beginning of the approach and grows sharply when the 'closing range' (namely the range between the camera and the observed terrain) is short. Equation 7.2-3 is the scale-factor model.

$$sf_i = \frac{r_0}{(r_0 - \bar{v} \times i \times \tau)} = \frac{r_0}{X_c(i)} \quad , \quad (7.2-3)$$

where $\bar{v}$ is the average velocity of the airborne vehicle, τ is the calculation time per clock cycle, r_0 is the initial range between the airborne vehicle and the observed terrain and $X_c(i)$ is the 'closing-range' which is decreased from frame to frame . In our case we assume that the average velocity $\bar{v}$ is roughly 0.6 Mach which is 200(m/sec) , this value is quite a reasonable value for airborne-vehicles. The calculation time interval ΔT is 16 (msec) which is caused by the fact that our video camera based on 60 Hz video frequency . We assume quite a wide variation for the initial range - r_0 , between 1 (Km) and 6 (Km) , since the pictures in Fig. 1 and in Appendix-1 were taken from those distances; however these ranges are suitable for observation ranges from an air-born vehicle equipped with electro-optics facilities .

Since between two successive video frames ϕ is a small angle , we assume that :

$$\sin(\phi_i) = \phi_i \quad (7.2-4)$$

$$\cos(\phi_i) = 1 \quad (7.2-5)$$

but still equations (7.2-1), (7.2-2) are nonlinear because of the nonlinearity of the scale-factor (sf) equation ; moreover, the sf non-linearity increases as $X_c(i)$ decreases and vice-versa .

7.3 The Mean Absolute Difference (MAD) algorithm

There are several well-known algorithms which estimate the displacement of moving objects between two successive television frames; they are divided into three groups : Fourier method ; Correlation method ; Differentials method.

In our case , the second class is of particular interest ! The first group, the Fourier method needs signal transformations and in our case this is useless and too complicated, because our sample data has no certain spatial frequencies for suppressing or enhancing, since the video signals are uniform. To operate

the Fourier transformation on the whole search area, which is a relatively big one, seems to be a very complicate approach. The third class is a popular one [6] in particular when one displacement estimate is calculated for the entire moving area, or when the displacement is estimated recursively or when the estimation is for small rectangular image segments, which of course is not our case. Correlation methods are analyzed in detail in [3]. In order to avoid excessive computational burden, the Mean Absolute Difference (MAD) method is the preferred one .

As is explained in §5.1 the initial step in the matching process is defining the current window. We then locate the current window for the initialization of the matching process, so that in the horizontal direction there is a total overlapping and in the vertical direction the lower half of the current window is overlapped by the reference window as shown in Fig. 19-a. At this stage , which is defined as $J=1$ the first video line of the reference window - l_{1r} is overlapped with the ninth line of the current window - l_{9c} and in the same way l_{2r} is overlapped with l_{10c} etc . (see Fig. 19-a) , so for $J=2$ the total size of the overlapped area is $8 \times k_{c,l,r}$ which is 128 pixels (since $k_{c,l,r} = k_{r,l,r} = 16$).

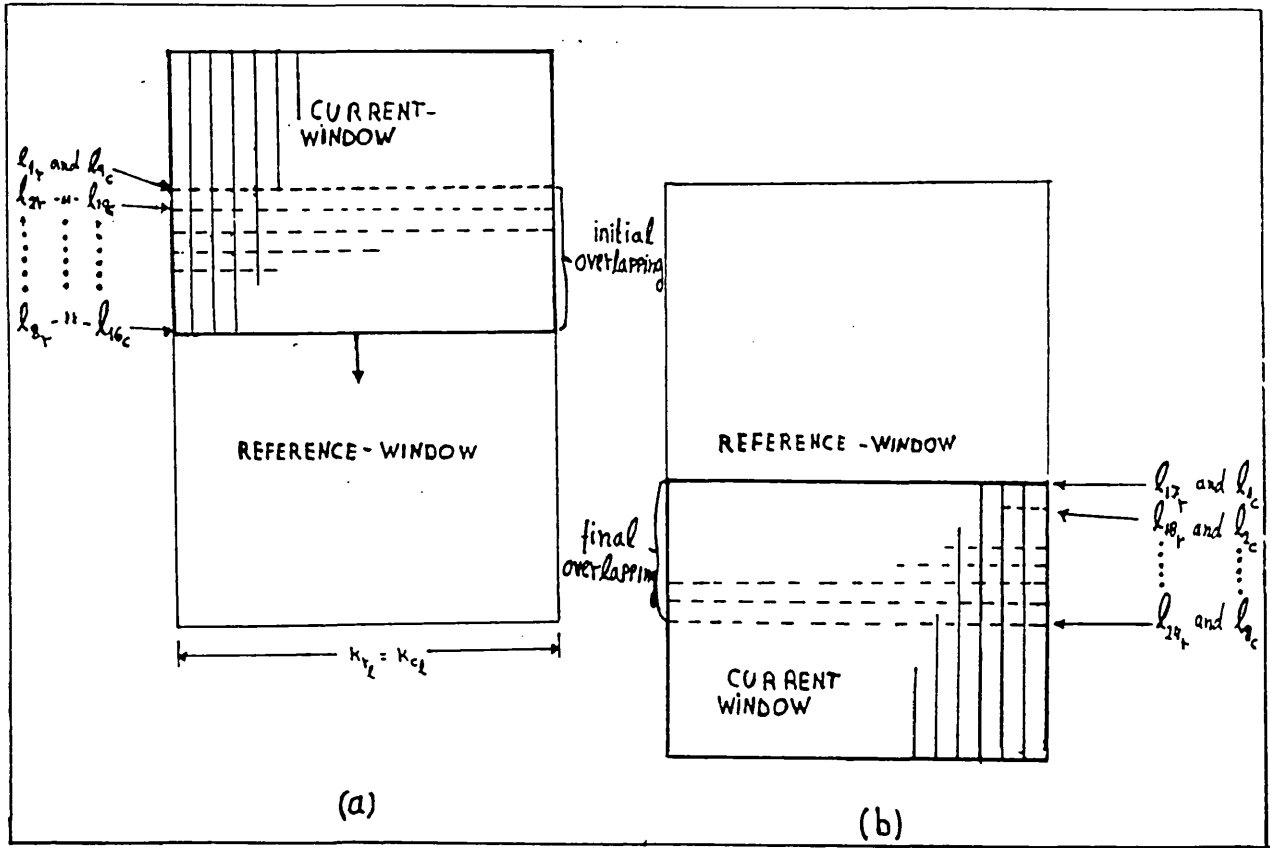


Fig. 19 The initial step (a) and the final step (b) along the matching process

This is the lowest number of data-sample (pixels) involved in the matching process, the same number of data-samples we get at the end of the process (see Fig. 19-b) . This is quite a small number bearing in mind that we might have a horizontal displacement in addition to a vertical one, but for the vertical estimation displacement we shift the current window J times in the vertical direction only , where $J_{\max}$ is equal to the number of lines in the reference window , therefore we get $J_{\max}=24$, which means that along the vertical direction we get 24 steps in which we operate the MAD algorithm and therefore 24 results from the matching process. The MAD algorithm is defined as :

$$\text{MAD}(J) = \nu^*(J) = \frac{1}{N} \times \left\{ \sum_{l=m_a}^{m_b} \left\{ \sum_{k=n_a}^{n_b} |I_c(l', k) - I_r(l, k)| \right\} \right\} \quad (7.3-1)$$

Where N is the total number of pixels involved in the matching process, m_a and m_b , n_a and n_b are the vertical and horizontal boundaries of the actual process - J , defined in reference window coordinates. For the vertical matching process we always get $n_a = 1$ and $n_b = 16$, while m_a and m_b are modified according to J , e.g., for first step $J=1$ $m_a=1$ and $m_b=8$, for the second step $J=2$ $m_a=1$ and $m_b=9$, for the tenth step $J=10$ $m_a=3$ and $m_b=19$ (here is a case of maximum overlapping) and finally for the last step $J=24$ $m_a=17$ and $m_b=24$ (see Fig 19-b). $I_c(l',k)$ and $I_r(l,k)$ are the video intensities in the current and reference window respectively , l' is the l_c modified to reference window coordinates (since in Eq 7.3-1 we must work with one axis system), according to Eq 7.1-1, the connection is :

$$l' = l + \frac{l_{r,l,r}}{2} - J \quad (7.3-2)$$

As an example let us take $J=1$, therefore the vertical boundaries are $m_a=1$ and $m_b=8$, by substitute them into Eq 7.3-1 , we get for the first pixel in the reference window $I_r(1,1)$ and for the first pixel in the current window (in reference window coordinates) $I_c(12,1)$, in the same way for $J=2$ the first pixel in the ref-window is $I_r(1,1)$ and the first one in the current window is $I_c(11,1)$ etc.

At the end of this process we get 24 values which are the results of operating Eq (7.3-1) 24 times on different areas as we have just explained. The decision about the best vertical matching is based on the assumption that for a successful matching : $I_c(l',k) - I_r(l,k) \rightarrow 0$, therefore we look for the minimum of $\text{MAD}(J)$

$$J^* = \arg_J \{ \min \{ \text{MAD}(J) \} \} \quad (7.3-3)$$

J^* describes the number of shifts we have done in order to get the minimum of $\text{MAD}(J)$, this number is related to the nominal case (where $l_{12c} = l_{12r}$) as described in Fig. 18 which means that if for example we get $J^*=8$, the vertical displacement ϵ_v is -4 , since ϵ_v is equal to $J^* - 12$. The same procedure we operate in the horizontal direction, but along the J^* line only as we will discuss later.

7.3.1 The MAD algorithm with overlapping

Theoretically we expect, $\text{MAD}(J^*) = 0$, which happens in the nominal condition, defined as: noise=0, $\phi(i) = 0$, $d_h = 0$, $d_v = 0$, and we operate the MAD algorithm for a freezing point, namely the scale-factor is constant.

We test the MAD algorithm efficiency at the first stage in nominal conditions in order to analyse the results when they are affected only by the uniform data. We operated the method on two pictures: (c_l) (index 1 indicates that the results are related to the left search area) which has a good PQF (PQF=41.2) and picture (a_l) which has a poor PQF (PQF=6.4), both without initial overlapping at the beginning of the process (i.e. the "initial overlapping" in Fig. 19(a) is reduced to 2 lines)! The results are shown in Fig. 20.

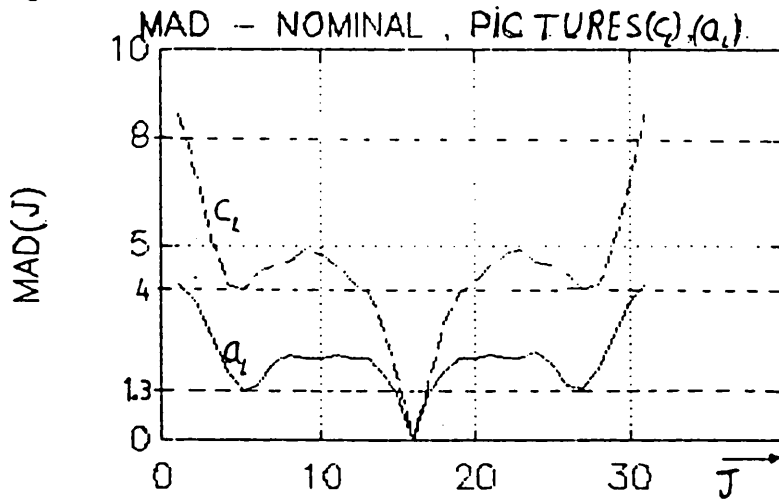


Fig. 20

It is quite obvious that, while we have a clear minimum at $J^*=16$, we have two side-lobes as well, which means that the likelihood to have false matching is high. Although this is a nominal case, we can see from pic: (a_l) in Fig. 20 that the differences between the side-lobes values and the minimum value at J^* are small. These results are a reason for great concern, and indeed when we tried the MAD

algorithm with a small disturbance like $\phi_c = 1^\circ$ for $\Delta T = 0.033$ (sec) we did not get any outstanding minimum. The results are shown in Fig. 21 , and we can see that the two side-lobes become local-minima .

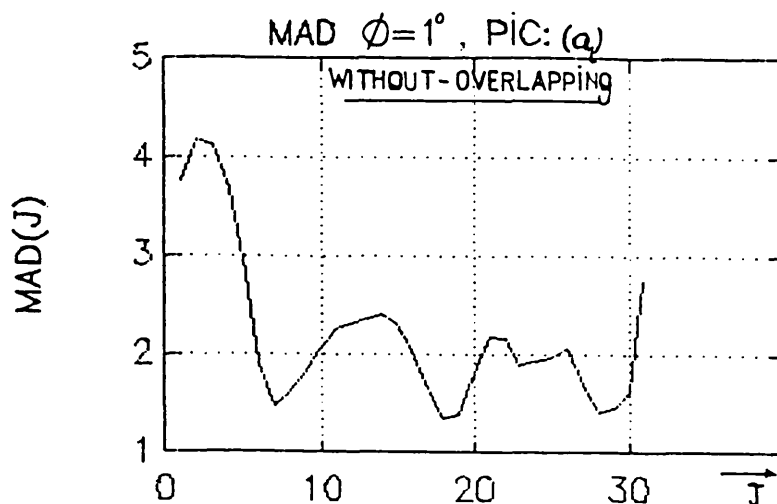


Fig. 21

One of the important reasons for this phenomenon is the lack of overlapping, in particular at the beginning of the process when the sample-data is relatively small, and the picture is a uniform one, with low value of PQF . The significance of the initial condition referring to Fig 19. is the big overlapping between the current window and the reference one. The size of the overlapping is : $\frac{K_{c1,r}}{2} \times K_{c1,r}$ and that is 50% of the maximum sample-data size . This method of overlapping is essential in order to decrease false matching . In Fig. 22 we show the results from the MAD algorithm with overlapping ,operated on pic: (a_1) . The improvement is remarkable, and the side-lobes are almost completely eliminated.

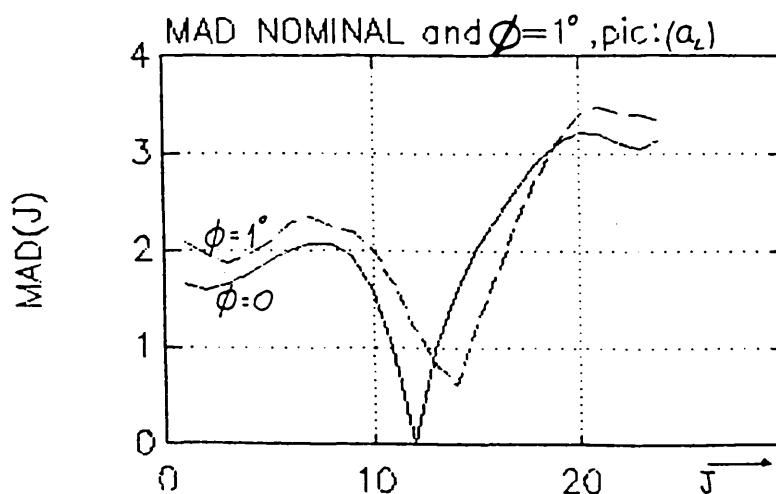


Fig. 22.

7.3.2 The MAD with overlapping and with weight function

The contribution of the overlapping process is significant, as we showed above. But in order to get an outstanding minimum we add a weight function , which gives the same weight to each vertical shift in the matching process. The new MAD process will be $\nu(J)$.

$$\nu(J) = \left(\Gamma - \frac{N_p}{N_t} \right) \times \nu^*(J) \quad \Gamma \geq 1 \quad (7.3.2-1)$$

where Γ is a constant the choice of which will be discussed below, N_p is the partial number of pixels involved in the process in every shift , and N_t is the total number of pixels for which there is a maximum overlapping.

$$N_p = (m_b - m_a) \times k_{r,l,r} \quad (7.3.2-2)$$

$$N_t = k_{c,l,r} \times l_{c,l,r} \quad (7.3.2-3)$$

m_a and m_b are the horizontal boundaries of the current window . The minimum of $\nu(J)$ (Eq 7.3.2-1) is, the matching point.

It is easy to see from Eq 7.3.2-1, that at the beginning of the matching process we have $N_p \ll N_t$ and therefore we get :

$$\nu(J) \leq \Gamma \cdot \nu^*(J) \quad ; \quad N_p \ll N_t \quad , \quad (7.3.2-4)$$

When the matching process meets its maximum sample data condition (maximum- overlapping) i.e., $N_p = N_t$, we get :

$$\nu(J) = \nu^*(J) \quad ; \quad N_p = N_t \quad (7.3.2-5)$$

The advantage of this weight function is its ability to avoid accidental matching caused by similar sub-regions in the reference-window and the current one. The greatest likelihood to find such sub-regions, happens when a relatively small data sample is involved while we compare the two windows, as usually happens on the first shifts in the matching process. Practically this process never reaches a theoretical minimum of zero, its minimum value is the lowest value among a group containing the MAD results and therefore it could happen that a sub-region in the

reference-window, which accidentally coincides with a sub-region in the current one, might easily cause a false minimum. But by operating Eq 7.3.2-1 we modify the MAD algorithm in such a way that the minimum received is more affected by the differencing procedure i.e., $(I_C(l',k) - I_T(l,k))$ rather than N i.e., the total number of pixels involved in the process. By using simulation methods we find that $\Gamma = 2$ is adequate for this process. Increasing Γ above 2 might decrease the sensitivity of the matching, particularly when a strong disturbance is expected and the picture is a uniform one.

In Fig. 23. we show the improvement in the results with the MAD+WEIGHT function compared to results with the MAD (which includes the overlapping) without the WEIGHT function. In this case the system is excited by a roll command $\dot{\phi}_c = 90$ ($^{\circ}/\text{Sec}$).

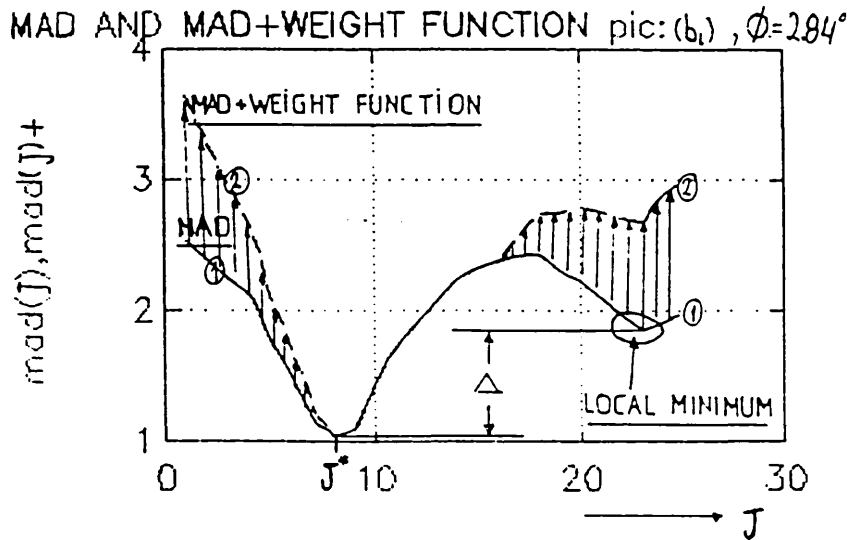


Fig. 23

The local-minimum in graph (1), almost vanished in graph (2) and the real minimum is not affected. It is important to point out that if the difference Δ between a local minimum and the real minimum is small, the likelihood of a false matching in the worst case is increased. In the example shown in Fig. 23 the use of the weight function increases Δ by a factor of almost 2. Although in practice a displacement more than 9 pixel per frame (as in Fig. 23.) is unlikely to happen, we don't at this stage put in any logical constraints, since we would like to examine the process in a wide spectrum of displacements.

7.3.3 The MAD with decision criterion

One of the features which characterizes this matching process is the absence of an outstanding minimum for very uniform scenes, particularly when there is a significant disturbance. Actually we get a 'minimum domain' and there is no sharp point which defines the real minimum. The reason for this phenomenon is that scenes representing desert, wilderness and other uniform areas, have the property of very slight gradients in the video intensities, in vertical and in horizontal directions around an arbitrary pixel, and this causes similar adjacent regions. In Fig. 22 we can see an example of an outstanding minimum, whereas in Fig. 23-a we show the 'minimum domain', the disturbance is $\phi_c = 2.84^\circ/\text{frame}$ and the picture is (a_r) . There is a wide range (m.d.) of pixels which might include the real minimum, since the differences among the MAD values in the 'minimum domain' are very small.

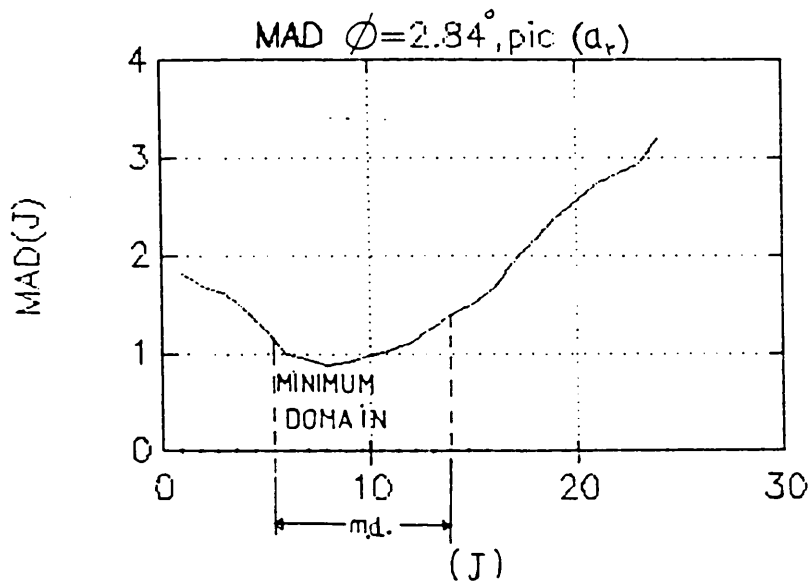


Fig. 23-a

The problem of identifying the real minimum inside the 'minimum domain' is intensified by the fact that we shift the current window vertically along the reference window with a complete overlapping. That means that we do not shift at this stage horizontally, whereas even rotation by itself might cause a horizontal displacement, so that the total number of pixels for a successful matching is

decreased, which leads to quite a flat minimum (the ‘minimum domain’). As discussed previously we need an accurate identification of the displacements , namely we are looking for an outstanding minimum.

In order to be able to identify the real minimum we build a decision criterion based on the values of $\nu(J)^2$ and the derivative values before and after each value of $\nu(J)$. The idea is based on the assumption that the ‘minimum domain’ is composed of several local-minima and only the right weight between the ‘shape’ and the value of the local-minimum will identify the real minimum. The output from a successful matching is a minimum in the values of the $\nu(J)$ function and an inevitable change in the ‘shape’ of the $\nu(J)$ function. The change in the graph shape might be small and therefore we take the differences between the adjacent derivatives around each arbitrary value of $\nu(J)$. We define the right side as : $\dot{\nu}(J^+)$ and the left side derivative as : $\dot{\nu}(J^-)$, and we get :

$$\dot{\nu}(J^+) = \nu(J+1) - \nu(J) \quad (7.3.3-1)$$

$$\dot{\nu}(J^-) = \nu(J) - \nu(J-1) \quad (7.3.3-2)$$

and the difference $\Delta\dot{\nu}(J)$ is :

$$\Delta\dot{\nu}(J) = \dot{\nu}(J^+) - \dot{\nu}(J^-) \quad (7.3.3-3)$$

Our decision function $\eta(J)$ is therefore :

$$\eta(J) = \frac{\Delta\dot{\nu}(J)}{\nu^2(J)} = \frac{\nu(J+1) - 2\cdot\nu(J) + \nu(J-1)}{\nu^2(J)} \quad (7.3.3-4)$$

and the decision criterion $\eta_{\max}$ is :

$$\eta_{\max} = \text{Max}\{ \eta(J) \} = \eta(J^+) \quad (7.3.3-5)$$

where J^+ is the value at which the maximum is attained. It is easy to see that the maximum of $\eta(J)$ is reached when $\nu(J)$ is minimum and $\nu(J+1) + \nu(J-1)$ is maximum, which means that we get simultaneously a low value for $\eta(J)$ and a sharp “V” shape . Moreover this process enables us to filter the results from Eq 7.3.2-1. Basically the results of this process are similar to a ‘peak detector’. Fig. 24 illustrates how this decision process is working: we operate the MAD(J) algorithm and the decision criterion $\eta(J)$ on picture (a₁), the disturbance are: $\phi_c=3^\circ$ and a vertical displacement $d_v = -3$ (pix) .

In Fig. 24 we show both graphs $MAD(J)$ and $\eta(J)$ and the results are very interesting. While the minimum of the $MAD(J)$ process is at point J^* , the minimum identified by the decision criterion $\eta(J)$ is at point J^+ which is the real matching line, and from Fig. 24 we can see that J^+ is very different from J^* . This case illustrates quite clearly that we cannot rely on the minimum values of the $MAD(J)$ process only, but we need a more sophisticated criterion which is the $\eta(J)$ criterion.

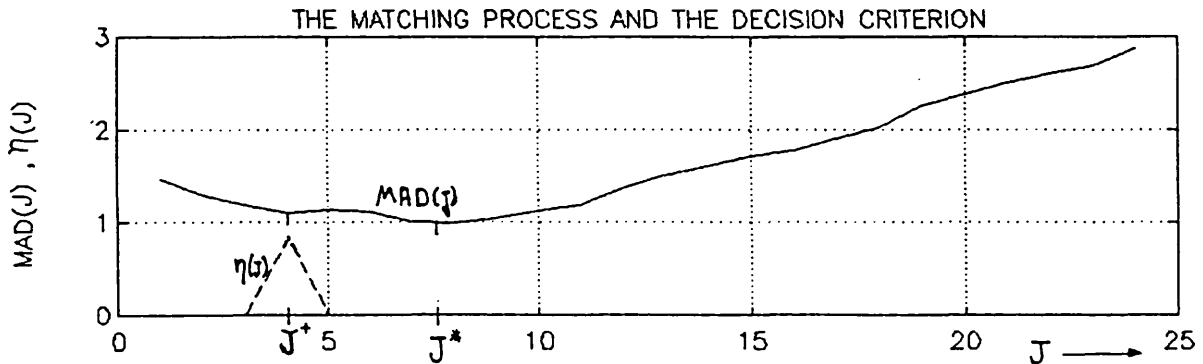


Fig. 24

It is interesting to realize that the significance of the location J^* of the minimum value of $MAD(J)$, is reduced by involving other effects such as the gradients along the $MAD(J)$ curve (Eq 7.3.3-4). The difference between point J^+ and J^* is 4 (pixels) which means that the contribution of the decision criterion is very significant (4 pixels might be translated to 2° , in terms of rotation angle!). The flatter the curve of the $MAD(j)$ function, the greater the significance of the decision criterion, since the differences among the values of the MAD function are small. By operating the suggested decision criterion $\eta(J)$ particularly in such cases we are looking for minor changes along the gradients of the curve, and that might cause the whole difference between a wrong identification based only on the MAD values themselves, and the perfect identification based on our decision criterion.

It is worthwhile to mention here that noise and other distortions have a negligible effect on the decision criterion, since the data used on this process namely - $\nu(J)$ are filtered variables, which have been smoothed by the overlapping process and the weight function, as we discussed previously. In the next paragraph we will discuss the horizontal matching process, and the special modification implemented in this process. We will see that the decision criterion in the horizontal matching is even more vital than in the vertical one.

7.3.4 The MAD process in the horizontal direction.

In order to identify the displacement in the horizontal direction, we operate the same process as in the vertical direction. The new idea is that the horizontal shifts are preformed only along the line which had reached the minimum in the vertical matching process - J^+ , so that instead of having $L_{r_{l,r}} \times K_{r_{l,r}}$ shifts, we get only: $L_{r_{l,r}} + K_{r_{l,r}}$ which is a huge reduction in calculations, and has an enormous impact on real-time performance; in fact in our case we reduce the number of shifts and by that the total number of calculations by almost 90% !! There are three main reasons which enable us to go only along one special-line, J^+ :

- (1).The use of quite a big number of data samples for the reference-window.
- (2).The use of total overlapping in the vertical direction together with a weight function and a criterion decision.
- (3)The picture character.

By involving a big number of pixels in the matching process, we eliminate the possibility of a 'total failure without any hint' about the right place of matching. Although the results from the first stage of the matching process might be inadequate, the use of a weight function and a decision criterion will force the matching point to be in the right place. Therefore, even when a displacement in the horizontal direction occurs together with a vertical shift, this process will identify the right vertical shift - $\epsilon_{v_{l,r}}$, based only on the information from the vertical process. Even more significant is the picture character. Since we deal with uniform pictures which represent real terrain the information 'transmitted' from such video lines or columns does not change sharply or abruptly but dies away smoothly and slowly. That means that by utilizing a big number of data-sample (pixels) in the vertical matching process we might get good information about the right place for a successful matching in the horizontal process. Although for the vertical process even when disturbances occur and cause a horizontal displacement, we still shift the current window precisely along the centre column (total overlapping) which is certainly not the best column to shift along. The last assumption is unique to pictures representing uniform terrain, and is; There are no sharp lines or sharp changes along the video line or columns, like in the pictures in Fig. 1 or in appendix 1.

In Fig. 25 we illustrate this principle discussed above, we can see that the

information from a horizontal matching process, which is shifted from the left side of the picture to the right side, has no advantage compared with a vertical matching process with any horizontal shifts, since we get a 'minimum line' (L_7 in Fig. 25) which means that all minima created by the vertical matching process are approximately located along the same horizontal line , with any significant dependence on an horizontal shifting. Even in very uniform pictures, the variability on the points composed the 'minimum line' is $\pm 1(\text{pixel})$, therefore it is useless to operate those horizontal shifts for each vertical matching process .

So we operate the vertical matching process in an optimum way by shifting the current window along the reference one with a total overlapping and in order to identify the horizontal displacement $\epsilon_{h,l,r}$ we operate the same process along the 'minimum line' which has been identified in the vertical process.

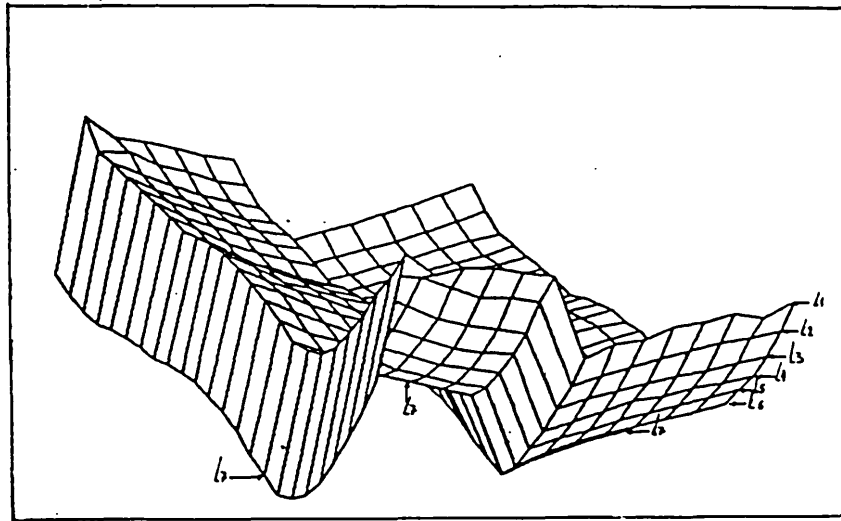


Fig. 25 The 'minimum line' - L_7

7.4 The IMAD process

We now call our overall matching process the "IMAD" ('Improved Mean Absolute Difference') process , which is actually the MAD process together with those 4 outstanding supplements which have been discussed above, namely :

- (1)The overlapping technique.
- (2)The weight function.
- (3)The decision criterion.
- (4)The 'minimum line' method .

The IMAD process is performed on both sides of the picture as we already

explained. Eventually we get two estimated error-vectors, from the left side process $\bar{D}_L(\hat{\epsilon}_{hl}, \hat{\epsilon}_{vl})$ and from the right side process $\bar{D}_R(\hat{\epsilon}_{hr}, \hat{\epsilon}_{vr})$. These 4 errors must be estimated in real-time and are the data base for the estimation of $\bar{P}(i)$, where $\bar{P}(i)$ is :

$$\bar{P}(i) = [\hat{\phi}(i), \hat{d}_v(i), \hat{d}_h(i), \hat{sf}(i)] \quad (7.4-1)$$

The vertical errors $(\hat{\epsilon}_{vl}, \hat{\epsilon}_{vr})$ are estimated by the vertical matching process as explained previously (§7.3.3) and the horizontal errors $(\hat{\epsilon}_{hl}, \hat{\epsilon}_{hr})$ are estimated by the horizontal matching process along the 'minimum line' as is explained in (§7.3.4). There is only one difference between the vertical and the horizontal matching process and that is the fact that clearly the result quality from the horizontal process depends on the accuracy of the vertical one. But it is amazing to realize that under the constraints defined in §(2), the results from the horizontal process are satisfactory ! In Fig. 26 results of the horizontal IMAD process are illustrated.

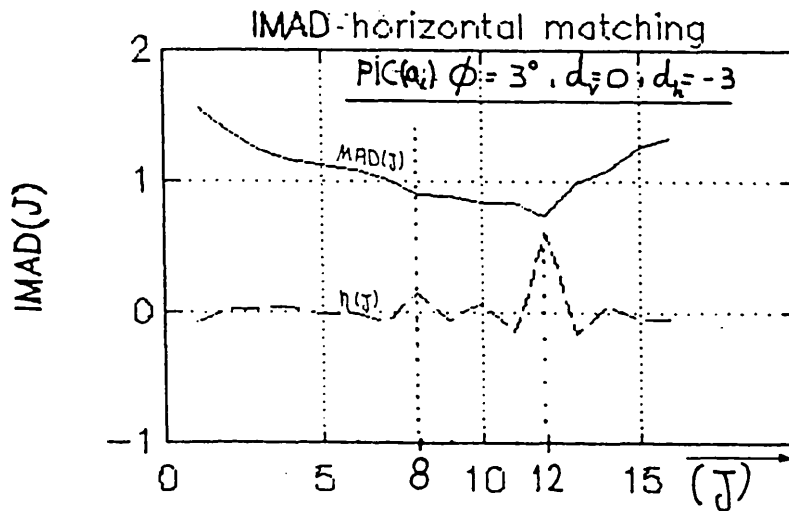


Fig. 26

In this complex example, the rotation angle is $\phi_c = 3^\circ$ per frame, and the horizontal displacement is $d_h = -3$ pixels, the picture (a) is a very uniform one and the $MAD(J)$ minimum is easily identified by the decision criterion function $\eta(J)$, although the $MAD(J)$ function is quite flat. This is another example which illustrates the advantage of taking in consideration the gradients of the $MAD(J)$ (Eq 7.3.3-4).

In this case the vertical matching displacement is identified as 7 pixels instead of 6 pixels and the horizontal displacement is identified as 4 pixels instead of 3 pixels, which means that the horizontal results are not affected by the vertical results ,since there is no increase in the error estimation. Furthermore a one- pixel error might easily be a result of noise or of the bilinear process(appendix 2) and not a result of the IMAD process.

7.4.1 The IMAD and skip technique

The IMAD process performance is highly dependent on two main factors, the picture quality factor (PQF) of the reference-window, and the total number of data samples (pixels) involved in the matching procedure. In order to utilize the information from the PQF , we define a PQF threshold called T_h and then operate the following procedure :

$$PQF \leq T_h ; \text{SKIP} = 1 \quad (7.4.1-1)$$

$$PQF > T_h ; \text{SKIP} = 2 \quad (7.4.1-2)$$

which means that if the reference-window is located in a poor region and therefore the PQF is less or equal to the threshold T_h , we are not allowed to skip any pixel while the matching process is operated. But if the PQF is higher then the T_h we are allowed to skip one pixel in the matching process . The purpose of this procedure is to shorten computing time whenever possible. In Fig. 27 (a)

we

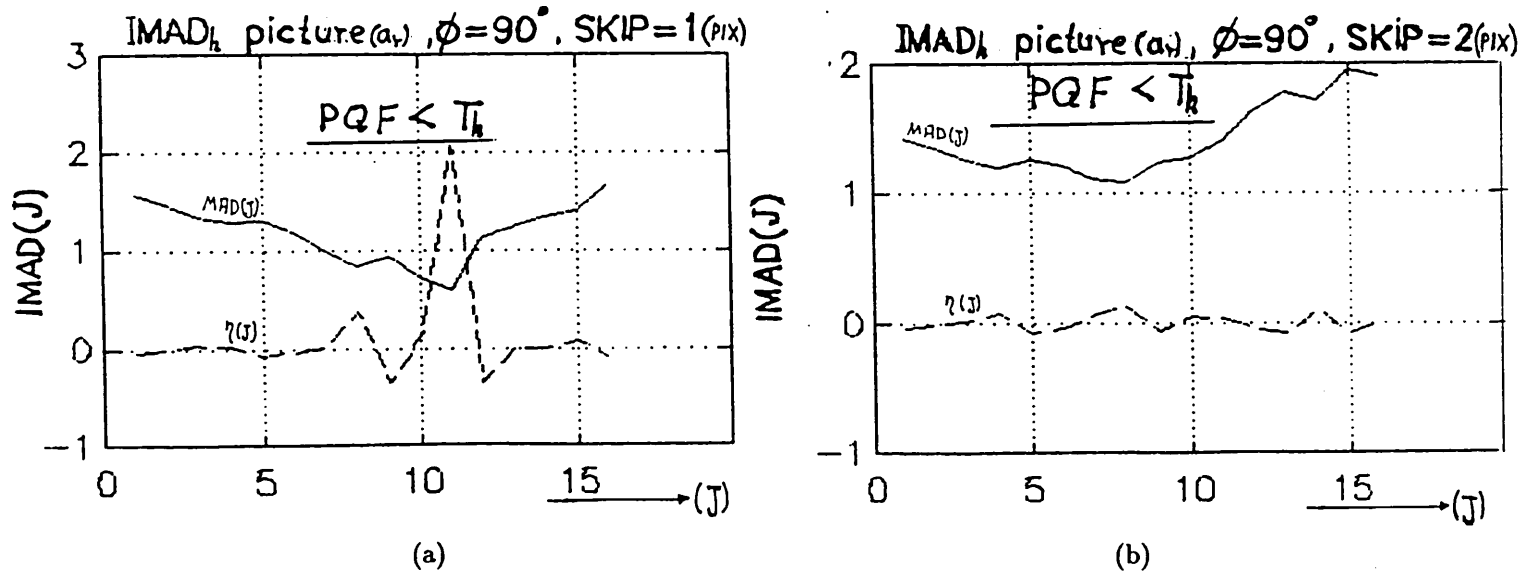


Fig. 27

show the result from the IMAD process for a very uniform picture ($PQF \leq T_h$). By using each pixel available the ultimate results of the process are good, but if we ignore the procedure in Eq-7.4.1-1 and Eq-7.4.1-2 and skip one pixel in the IMAD process, it is easy to realize from Fig. 27 (b) that the results are disrupted. For pictures with PQF greater than T_h , the results using the skipping procedure or not using it are the same.

The skipping decision is operated in the horizontal direction only, and that is because the vertical matching process eventually absorbs almost all the distortions occurring in a current frame, moreover the horizontal process has a certain dependency on the vertical one and therefore we cannot afford to lose any information in the vertical process.

The whole IMAD procedure is shown in block diagram form in Fig. 28.

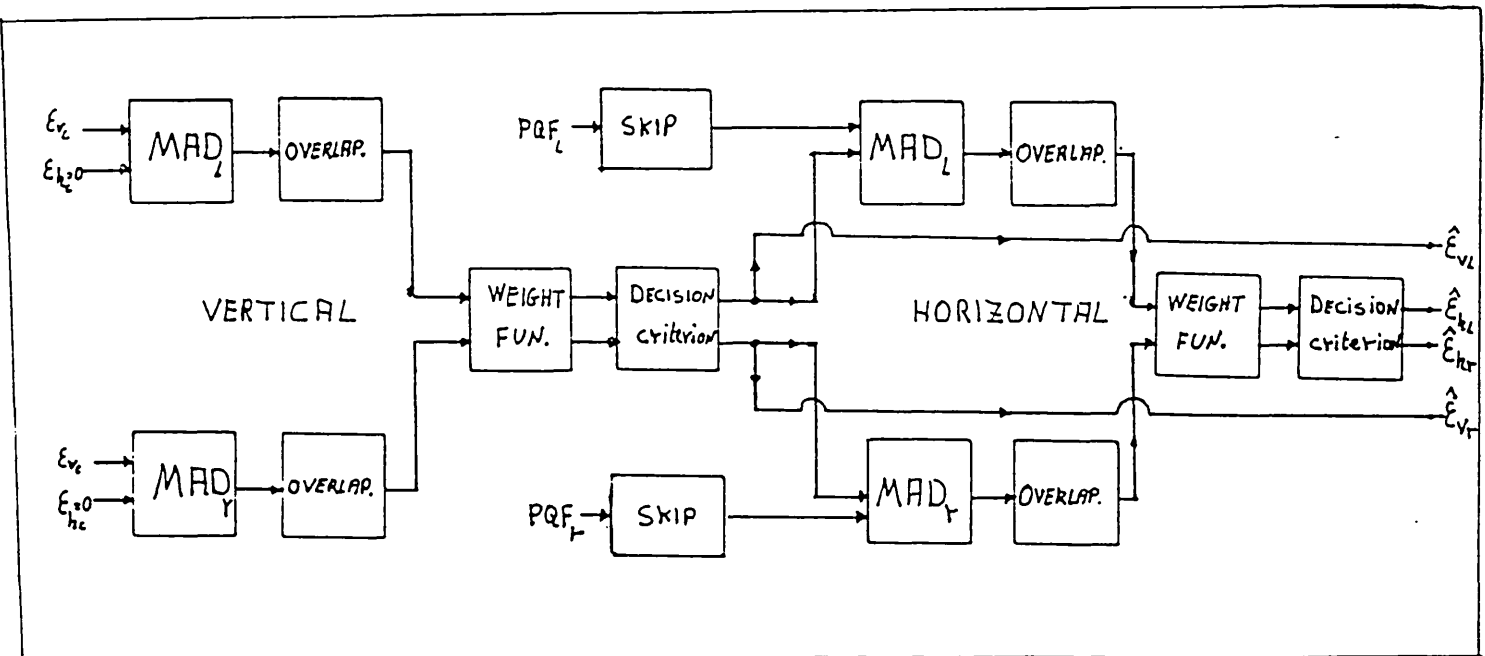


Fig. 28 The IMAD block diagram

CHAPTER 8

8. THE ESTIMATION PROCESS .

8.1 The displacement model

The estimation process is executed serially to the IMAD process: we have to complete both processes in real time and therefore a fast and simple algorithm is required .

From the IMAD process we get two error vectors, $\bar{D}_L$ and $\bar{D}_R$ on the left and right sides of the picture. Both contain two components , which are the estimated displacements in the horizontal and the vertical directions.

$$\bar{D}_L(i) = [\hat{\epsilon}_{hl}(i) , \hat{\epsilon}_{vl}(i)]^T \quad (8.1-1)$$

$$\bar{D}_R(i) = [\epsilon_{hr}(i) , \epsilon_{vr}(i)]^T \quad (8.1-2)$$

The outstanding advantage from operating the IMAD process in parallel on both sides of the picture, is the fact that we get the 'swing effect' , in other words since the rotation axis passes through the centre of the picture, the vertical error components on both sides $\hat{\epsilon}_{vl}(i)$ and $\hat{\epsilon}_{vr}(i)$ will have a different sign when rotation occurs, since their gradients are in opposite direction. Moreover since we locate the windows at a remote distance from the picture centre (O), the effect of rotation will be significant , as illustrated in Fig. 29 :

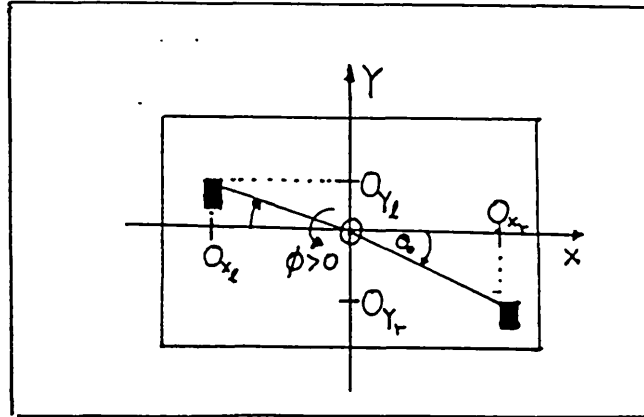


Fig. 29 The 'swing effect'

In order to simplify the estimation process we examine the displacements caused by each of the four unknown variables:

The 'swing effect' in the vertical direction is caused by the roll angle $\hat{\phi}(i)$, which means :

$$\frac{\partial \epsilon_{y_l}}{\partial y_l} = - \frac{\partial \epsilon_{y_r}}{\partial y_r} \quad (8.1-3)$$

We take into consideration the vertical components only, although there might be horizontal displacements as a result of the rotation disturbance, particularly if we get θ_0 close to $\pi/4$, but anyway the more significant displacement is the vertical one.

The scale-factor - $sf(i)$, is due to the fact that the airborne vehicle is approaching the observed terrain. In this case there is a blow-up of the picture and the pixels escape from the picture centre towards the edges of the picture. This phenomenon is characterized by the following model in the horizontal direction :

$$\frac{\partial \epsilon_{x_l}}{\partial x_l} = - \frac{\partial \epsilon_{x_r}}{\partial x_r} \quad (8.1-4)$$

In the vertical direction the model is the same. We prefer the horizontal model for two reasons :

- (1) The horizontal model is unique for the scale-factor model, while the vertical one, is similar to the roll model.
- (2) The 'pixels escape rate'- $\dot{\alpha}(i)$ of an arbitrary pixel is given by:

$$\dot{\alpha}(i) = \frac{a_{oh} \times \bar{v}}{X_c^2(i)} \quad (8.1-5)$$

where $X_c(i)$ is the close-range :

$$X_c(i) = r_0 - \bar{v} \times i \times \tau \quad (8.1-6)$$

a_{oh} - is the initial condition for $Ox_{l,r}$ ($i = 0$), and $\bar{v}$ is the average velocity of the airborne vehicle.

From Eq 8.1-5 it is easy to see that at the beginning of the process ($r_0 = X_c$), the 'escape rate'- $\dot{\alpha}(i)$ is very slow, which might cause difficulties in the scale-factor estimation. The only way to increase the rate at this stage is by increasing a_{oh} . In the horizontal direction we can assure it, by locating the search-areas at the maximum distance from the centre of the field of view- FOV, but the vertical distance- a_{ov} is not under our control, since it is determined by the AWL algorithm. Therefore the estimation of the scale-factor ($\hat{sf}(i)$) is based on the horizontal derivatives (Eq 8.1-4).

The vertical displacement - d_v , model is :

$$\frac{\partial \epsilon_{v_l}}{\partial v_l} = \frac{\partial \epsilon_{v_r}}{\partial v_r} \quad (8.1-7)$$

since both windows on the left and right sides of the picture move vertically up or down in the same direction. The same applies to the horizontal direction and therefore we get :

$$\frac{\partial \epsilon_{h_l}}{\partial h_l} = \frac{\partial \epsilon_{h_r}}{\partial h_r} \quad (8.1-8)$$

The derivative model is illustrated in Fig. 29-a

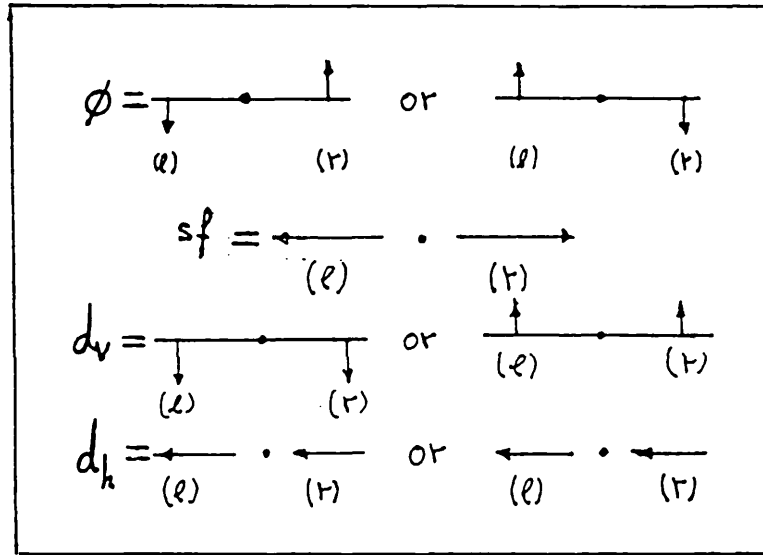


Fig. 29-a The derivative model

8.2 The estimator algorithm

Based on the previous explanations, the estimation process begins with the evaluation of the scale-factor-sf. But the estimation of the $sf(i)$ is very sensitive to noise and other distortions particularly at the beginning of the process when the initial range- r_0 is big ($r_0 \geq 3000(m)$), and since the $\hat{sf}(i)$ affects all other estimated variables, we put three constraints on the scale factor estimation process in order to avoid false estimation . The constraints are :

- (1) $\epsilon_{h_l} \leq 0$
- (2) $\epsilon_{h_r} \geq 0$

$$(3) \quad \left| \epsilon_{h_l} + \epsilon_{h_r} \right| \leq 1$$

if these constraints are not fulfilled then we set

$$\hat{sf}(i) = 2 \cdot \bar{sf}(i-1) - \bar{sf}(i-2) \quad (8.2-1)$$

(i.e., linear extrapolation) and the initial conditions are :

$$\bar{sf}(1)=\bar{sf}(0)=\bar{sf}(-1) = 1 , \quad (8.2-2)$$

where $\bar{sf}(i-1)$ and $\bar{sf}(i-2)$ are smoothed by the least-square method as we will explain later. The reason for relying on the previous smoothed information $\bar{sf}(i-1)$ and $\bar{sf}(i-2)$ for the current estimated scale factor $\hat{sf}(i)$ is that since the actual horizontal results from the matching process ϵ_{h_l} and ϵ_{h_r} do not fulfill the scale-factor physical model as derived from Eq-8.1-4 and 8.1-5 as illustrate in Fig. 29-a, we cannot rely on them for the sf estimation process. In such a case we prefer to predict the current $\hat{sf}(i)$ from the previous smoothed information ($\bar{sf}$). We are not concerned about the error this method may cause to $\hat{sf}(i)$ by delaying the updating of $\hat{sf}(i)$ with the actual matching results, because a real-time estimation process is not vital for the scale-factor evaluation since its change rate is quite slow relative to the other variables except when the close-range $X_c(i)$ is very small (Eq 8.1-5) , but then the high sensitivity of sf causes the ϵ_{h_l} and ϵ_{h_r} to be always under the constraints mentioned in 1,2, and 3. In terms of time-constant (t_τ), the scale factor time-constant- $t_{\tau_{sf}}$ is bigger than the drifts (d_h, d_v) and certainly bigger than the rotation(ϕ) time-constant. Therefore for most of the air-born trajectory we get :

$$t_{\tau_{sf}} \geq t_{\tau_{d_{h,l}}} \geq t_{\tau_{\phi}} \quad (8.2-3)$$

which means the estimation process for the scale-factor($\hat{sf}(i)$) will 'wait' one or two video frames, in order to get ϵ_{h_l} and ϵ_{h_r} inside the constraints domain.

When the above constraints are satisfied (which usually happens for most of the flight-time), the sf estimation equation is :

$$\hat{sf}(i) = 1 + \frac{1}{2} \times \left(\frac{\epsilon_{h_l}}{O_{x_l}} + \frac{\epsilon_{h_r}}{O_{x_r}} \right) \quad (8.2-4)$$

Here we implement the method shown in Fig 29: by taking the sum of the left and right matching errors, we reduce the displacement effect on the sf estimation. As we mentioned earlier the importance of an accurate estimation of the sf is

enormous. In order to smooth the results from Eq 8.2-4, we use a linear least squares approach . The scale-factor physical model is :

$$sf(i) = \frac{1}{1 - (\frac{\bar{v} \times \tau}{r_0}) \times i} \quad (8.2-5)$$

Where i is the number of time-cycles ($t = \tau \times i$) . For simplicity , define

$$\gamma = \frac{\bar{v} \times \tau}{r_0} \quad (8.2-6)$$

Then the linear approximation of Eq 8.2-5 is :

$$sf(i) = 1 + \gamma \times i \quad (8.2-7)$$

Now, the least-square error is :

$$S = \sum_{i=1}^{i_t} (sf(i) - \hat{sf}(i))^2 = \sum_{i=1}^{i_t} (1 + \gamma \times i - \hat{sf}(i))^2 \quad (8.2-8)$$

By taking $\frac{\partial S}{\partial \gamma} = 0$, we get :

$$\hat{\gamma}(i_t) = \frac{\sum_{i=1}^{i_t} (\hat{sf}(i) - 1) \times i}{\sum_{i=1}^{i_t} i^2} \quad (8.2-9)$$

and by using the formula of Eq 8.2-7 we get :

$$\bar{sf}(i) = 1 + \hat{\gamma}(i) \times i \quad (8.2-10)$$

The range between the camera and the observed terrain $X_c(i)$ is :

$$\hat{X}_c(i) = \frac{\bar{v} \times \tau \times i}{\bar{sf}(i) - 1 + \epsilon} \quad (\epsilon = 0.001) \quad (8.2-11)$$

Now we transform the matching errors , in such a way that the scale-factor effect will be diminished .

We define :

$$\epsilon_{h_{lc}} = \epsilon_{h_l} - O_{x_l} \times (\overline{sf}(i) - 1) \quad (8.2-12)$$

$$\epsilon_{h_{rc}} = \epsilon_{h_r} - O_{x_r} \times (\overline{sf}(i) - 1) \quad (8.2-13)$$

$$\epsilon_{v_{lc}} = \epsilon_{v_l} - O_{y_l} \times (\overline{sf}(i) - 1) \quad (8.2-14)$$

$$\epsilon_{v_{rc}} = \epsilon_{v_r} - O_{y_r} \times (\overline{sf}(i) - 1) \quad (8.2-15)$$

The 'new' matching errors, $\epsilon_{h_{lc}}$, $\epsilon_{h_{rc}}$, $\epsilon_{v_{lc}}$, $\epsilon_{v_{rc}}$ will be utilized for estimating

the other three variables $\hat{\phi}$, $\hat{d}_v$, $\hat{d}_h$.

The estimated roll angle $\hat{\phi}(i)$ is :

$$\hat{\phi}(i) = -\frac{1}{2} \times \left(\frac{\epsilon_{v_{lc}}}{O_{x_l}} + \frac{\epsilon_{v_{rc}}}{O_{x_r}} \right) \quad (8.2-16)$$

The estimated vertical drift is :

$$\hat{d}_v(i) = \frac{1}{2} \times \left(\epsilon_{v_{lc}} + \epsilon_{v_{rc}} + \hat{\phi}(i) \times [O_{x_l} + O_{x_r}] \right) \quad (8.2-17)$$

The estimated horizontal drift is :

$$\hat{d}_h(i) = \frac{1}{2} \times \left(\epsilon_{h_{lc}} + \epsilon_{h_{rc}} + \hat{\phi}(i) \times [O_{y_l} + O_{y_r}] \right) \quad (8.2-18)$$

This process is operated in each video frame, and is implemented for stabilizing the air-born vehicle in roll and for closing the drift loop. Another by-product is the 'close range' - X_c , and we might utilize this information for adaptive proportional navigation. Anyhow the most important variable regarding stability aspects is of course $\hat{\phi}(i)$ and our concern is to get $\hat{\phi}(i)$ as accurate as possible in each video-frame, while the convergence of $\hat{d}_v(i)$ and $\hat{d}_h(i)$ to their real values may take even three or four video-frames. The whole system is called the Picture Estimator Process - PEP and is shown in Fig. 30

THE PICTURE ESTIMATION PROCESS - PEP , BLOCK-DIAGRAM

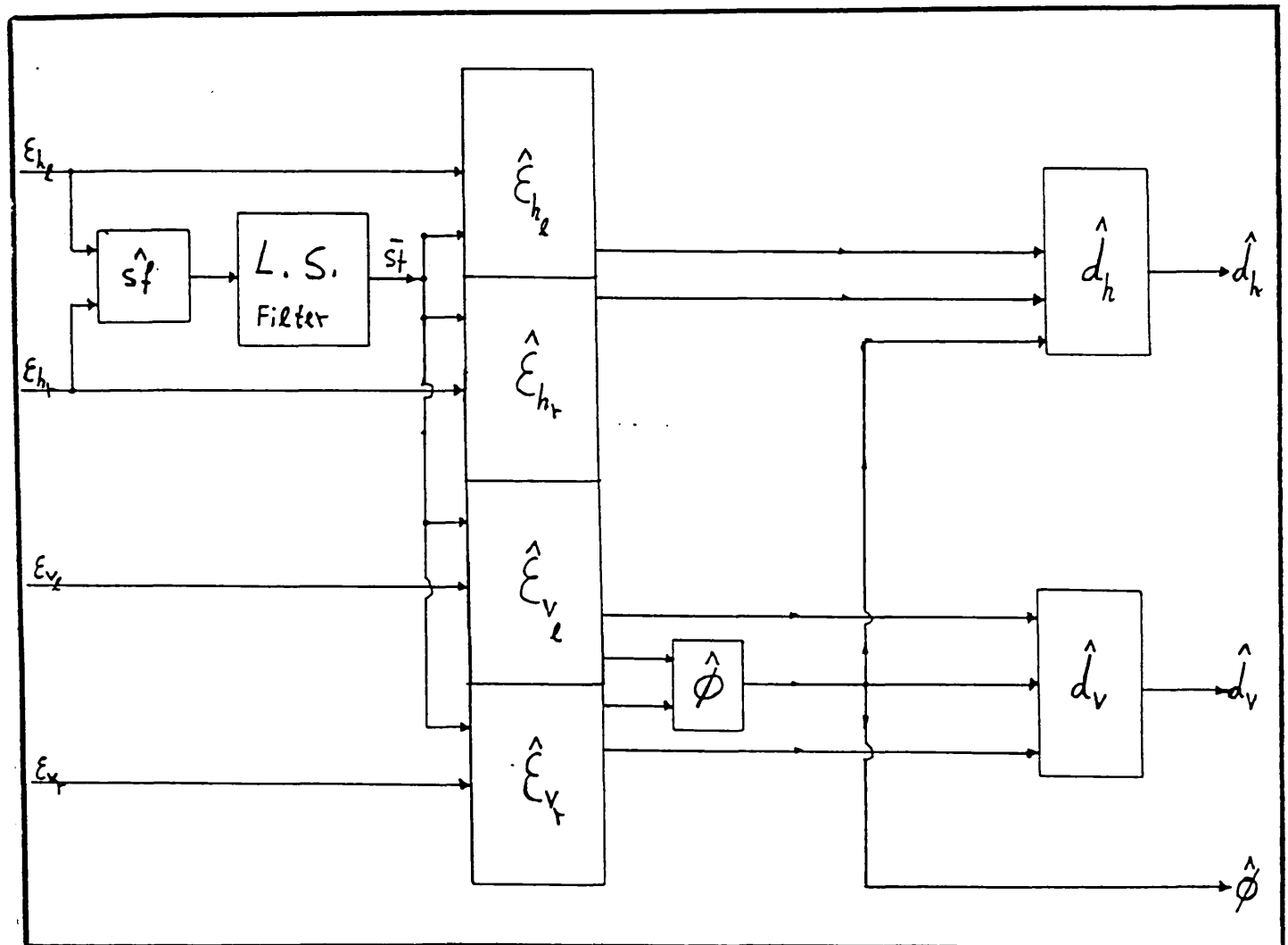


Fig. 30

8.3 The PEP performance

In order to examine the picture estimator process - PEP performance, we operate the PEP on picture (a) which has a low PQF, with the following data :

- (1) $r_0 = 6000(m)$
- (2) $\dot{\phi} = 90 (^{\circ}/Sec)$
- (3) $d_v = -3 (pixel)$
- (4) $d_h = 2 (pixel)$

We operate the PEP during 20 video-frames (0.66 (sec)), in Fig. 31 we show the results for $\hat{\phi}(i)$:

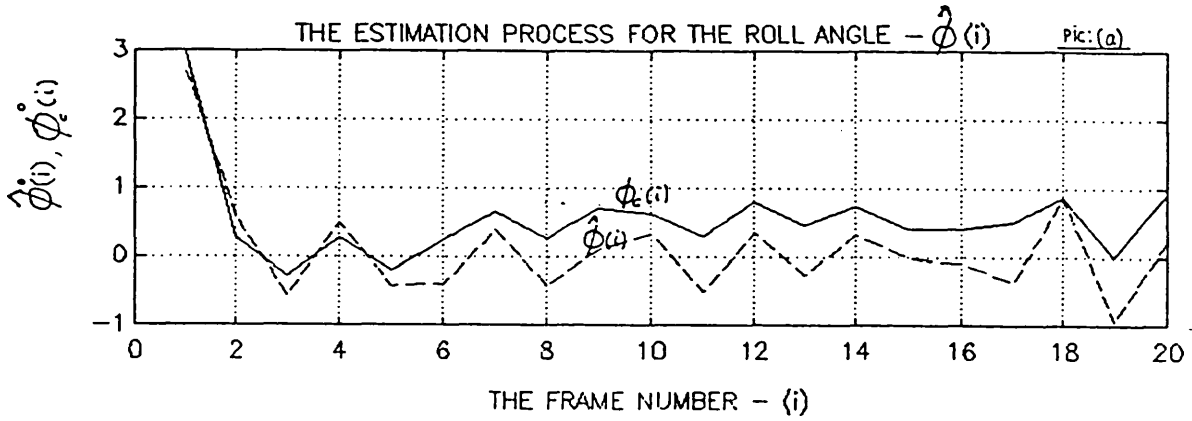


Fig. 31

We can see that the results are excellent: the PEP identifies the initial disturbance at the first frame, in fact we get $\hat{\phi}(1) = 2.63^{\circ}$ instead of $\phi_c(1) = 2.97^{\circ}$, where $\phi_c(i)$ is the roll command in each frame and $\phi_c(i)$ is defined as :

$$\phi_c(i) = \phi_c(i-1) - \hat{\phi}(i-1) \quad (8.3-1)$$

We assume that the system response in roll is immediate and that the time-delay between the (i) command and the (i-1) command is the calculation cycle which in this case is $\Delta T = 0.033 (sec)$, and therefore $\phi_c(i)$ is the rotation disturbance in each frame. (more about the integrated system response is described in chapter 9). The maximum error in each frame is less than $\pm 1^{\circ}$ which is adequate.

In Fig. 32-(a) we show the estimated vertical displacement - $\hat{d}_v(i)$ and in Fig. 32-b the estimated horizontal displacement - $\hat{d}_h(i)$. The initial vertical and horizontal displacements are 3(pixels) and -2 (pixels) respectively which is quite a big disturbance for one frame. The results in Fig. 32 (a) and (b) are quite satisfactory, bearing in mind that the picture is a very uniform one (picture (a)) and that the disturbance cause by the scale-factor is a continuous one, since we approach the observed terrain in the process. It is obvious to see that the estimated values follow the command values, particularly when the command values are significant. For small commands i.e., when $d_{vc}(i)$ and $d_{hc}(i)$ are less then 1(pixel) the PEP follows the commands with an error less or equal to one pixel. The command values per frame are:

$$d_{vc}(i) = d_{vc}(i-1) - \hat{d}_v(i-1) \quad (8.3-2)$$

$$d_{hc}(i) = d_{hc}(i-1) - \hat{d}_h(i-1) \quad (8.3-3)$$

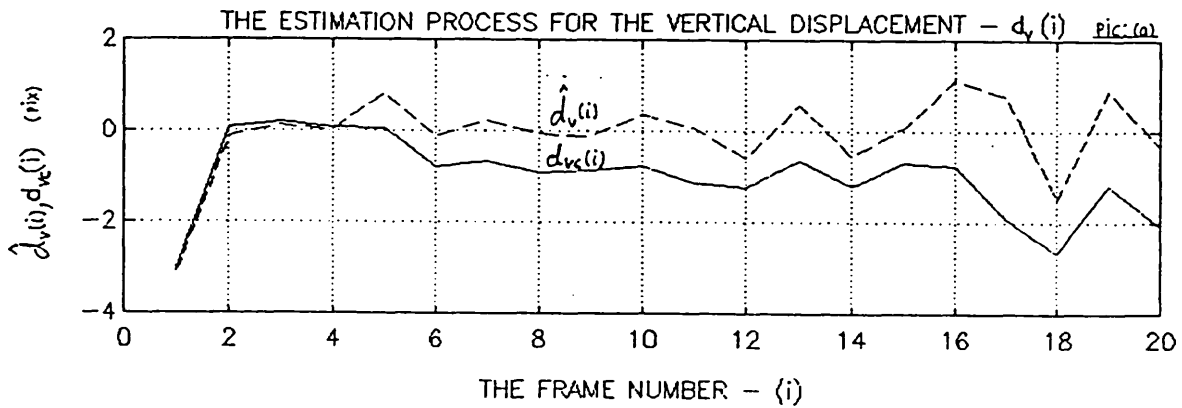


Fig. 32 - a

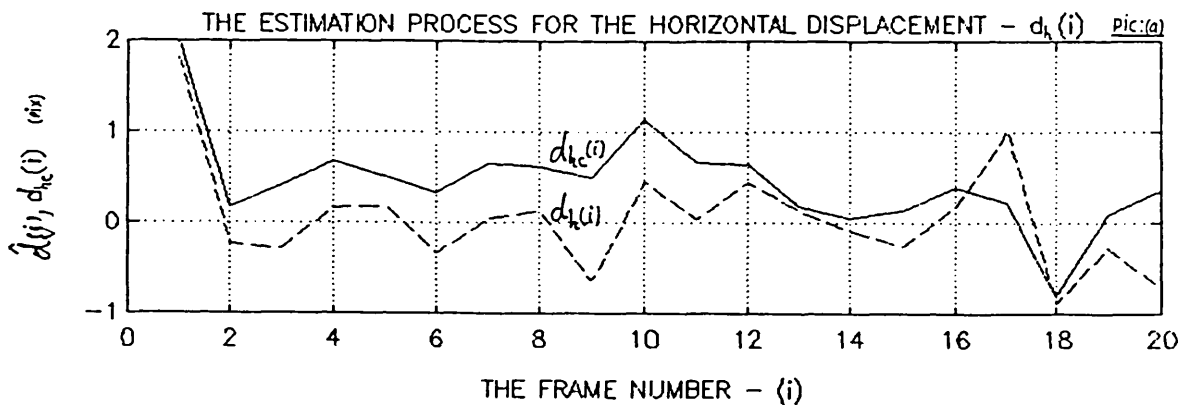


Fig. 32 - b

The last estimated variable is $\hat{X}_c(i)$ - the closing range. From Eq 8.2-11, it is clear that the quality of the PEP regarding $\hat{X}_c(i)$ is highly dependent on the accuracy of the $\hat{s}f(i)$, which in turn is highly dependent on the initial range - r_0 . For long ranges such as our case ($r_0 = 6000$ m) the sensitivity to the airborne vehicle approaching the observed terrain is low, which means that the 'picture blowing up' effect is not significant. Therefore it is very difficult for the IMAD process to identify any displacement. Actually we tried to improve the situation by locating the ref-window almost at the picture edges and we can see in Fig. 33 that after 6 frames eventually $X_c(i)$ converges.

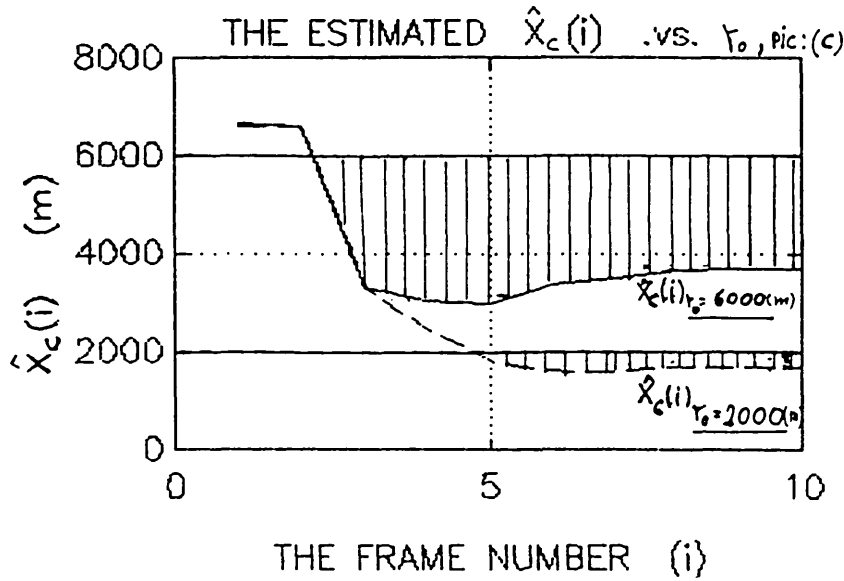


Fig. 33

For the same data except that the initial range is $r_0 = 2000$ (m), we can see in Fig. 33 that $\hat{X}_c(i)$ convergence is much better then for $r_0 = 6000$ (m), which demonstrates the highly sensitivity to r_0 .

For fast convergence, the decision about initial condition is vital, but since for short ranges the convergence of the process is fast, and is less affected by the other initial conditions, we always choose $\hat{s}f(0)=1$, and $\hat{X}_C(0) = 6600$ (m), which enable us to get a 'good-guess' for the first iteration. For long-ranges this is essential, and at the same time it does not have a significant effect on short-ranges, as we show in Fig. 33

Another important factor in the PEP is, the picture quality factor-PQF, which we explained in §4.2. It is reasonable to assume that since picture (c) has higher

value of PQF, the PEP results are better than in pictures with smaller values of PQF. In Fig. 34-a and 34-b we show a comparison between picture (c) which has a high value of PQF and picture (a) which has a poor value of PQF. We can see that while $\hat{\phi}(i)$ has a better initial estimated value for the case of picture (c), $\hat{X}_c(i)$ estimation is deteriorating in the case of high PQF values, particularly for long ranges, the reason is that pictures with high values of PQF are sensitive to every shift even ± 1 pixel which sometimes is caused by noise. This might lead to a wrong estimated values for a 'weak-variable' as X_c is, for long ranges (this phenomenon is well known in image processing and sometimes one uses a 'smoothing filter' aimed at reducing the algorithm sensitivity when the variable is a 'weak' one). But as we explained there is no vital necessity for a fast estimation regarding $\hat{X}_c(i)$ when the initial range is big.

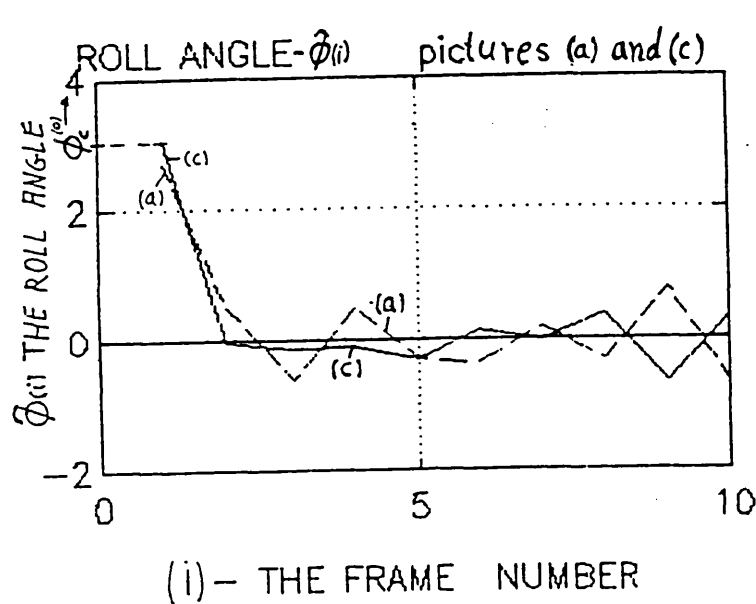


Fig 34-a

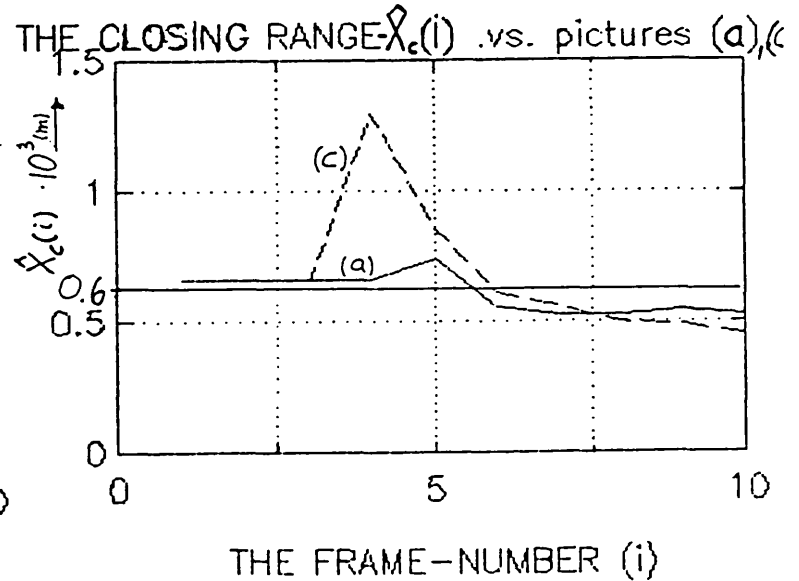


Fig 34-b

The PEP regarding the vertical and horizontal translations respectively d_v and d_h might converge after 2 or 3 frames since a displacement of one or two pixels per frame does not affect the stabilization problem and therefore they have no priority. The PEP automatically selects the right order in the estimation process, the priority is given to the roll angle estimation - $\hat{\phi}(i)$, and afterwards it depends on the initial range- r_0 . For long-ranges the PEP prefers first the horizontal and vertical translations and then the $\hat{s}_f$, and for short ranges, the PEP prefers first

the $\hat{s}f(i)$ and then the translations $\hat{d}_h(i)$ and $\hat{d}_v(i)$. The reason is obvious: since the sensitivity of the sf depends on $\frac{1}{\bar{X}_c(i)}$ as we previously explained, the shorter the range, the more the displacements caused by the scale-factor - sf are outstanding. In Fig. 35 we show an example of this phenomenon, we can see that only in the I_e interval is the effect of the scale-factor significant, which means that the first stage was utilized to estimate other variables, and only after four or five frames the estimation process begins and then converges after another five or six frames.

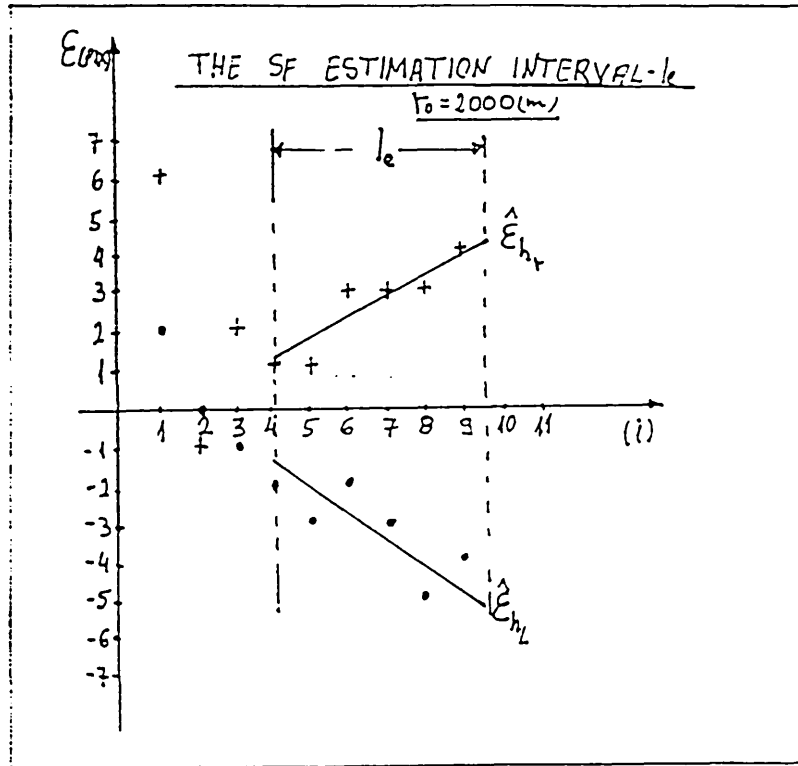


Fig. 35

CHAPTER 9

9. THE ROLL LOOP.

As we explained previously we utilize the video information for stabilizing an air-born vehicle in rotation. The closed loop block diagram is shown in Fig 36 .

We can see that the new 'block' in Fig. 36 is an 'image processing block', replacing a gyro as a feedback measuring device .

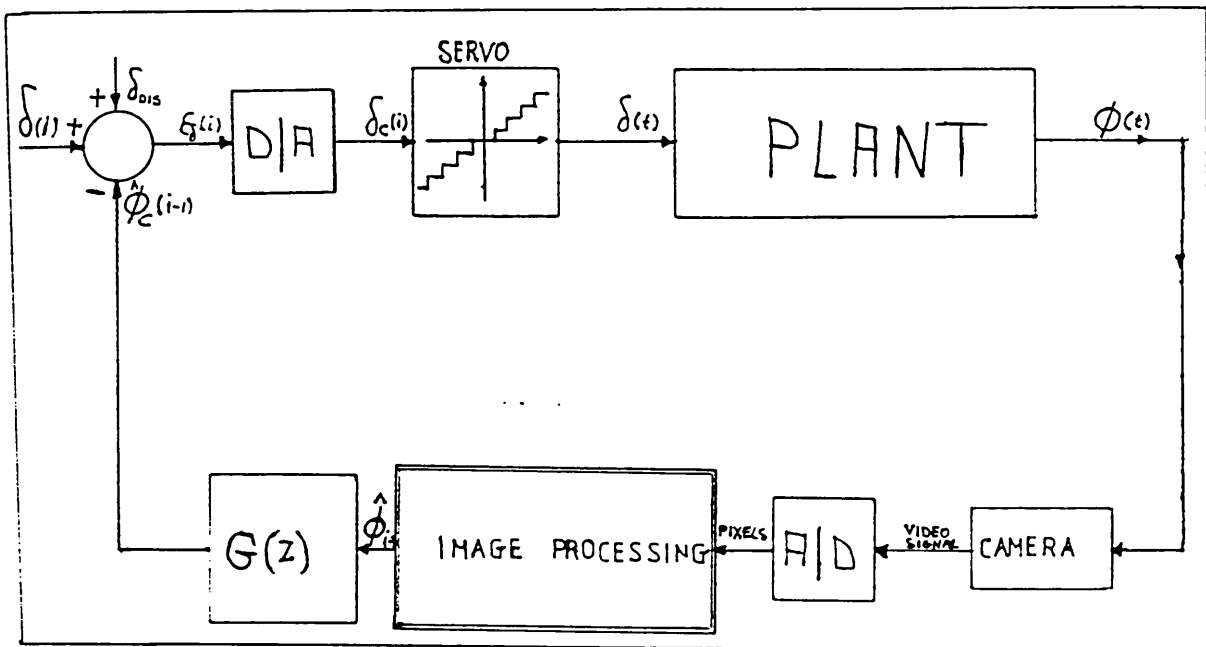


Fig. 36 The closed loop block diagram

There are several demands from the closed loop, which are unique to systems which use the video information:

1. $\dot{\phi}_{\max} < 3(\frac{\text{rad}}{\text{sec}})$
2. $\Delta\phi_{\max}|_{s.s} < 0.0174(\text{rad})$

The reason for the first demand is to avoid blur since above this rate we might face blur problems in particular at the edges of the FOV , since the pixels there

get high speed translations, which might cause a significant diminution in the video quality and eventually lead to a total failure in the image processing procedure. In order to avoid this phenomenon the maximum of the actuator angle should be designed as a compromise between stability demands and blur problems .

The second demand ($\Delta\phi \leq 0.0174$ (rad)) is for two reasons; the first concern human-engineering requirements, since the video information might be displayed on screen for human watching, and it is always more convenient for an observer to watch a horizontal picture then a rotated one. Secondly from the processing system aspect a rotated picture as a steady state condition is a very undesirable phenomenon ,it might cause some problems in the matching process since we get a steady-state error between the reference window and the current one, and that might lead to a failure in the estimation process. Therefore we prefer to keep the roll angle - ϕ as small as possible. Another assumption we have made concerns the actuator $\delta(t)$:

$$3. \delta(t)_{\max} < 0.2 \text{ (rad)}$$

This is a reasonable value as we will explain soon.

9.1 The plant model

The plant is an air-born vehicle which is characterized by the following aerodynamic equation :

$$I_x \times \ddot{\phi}(t) - q \times S_d \times C_{LP} \times \dot{\phi}(t) = q \times S_d \times C_{L\delta} \times \delta \quad (9.1-1)$$

Where qS_d is the normalized dynamic pressure, C_{LP} and $C_{L\delta}$ are the aerodynamic coefficients which depend on the air-born configuration and are constant for our airborne vehicle . In our case Eq 9.1-1 is reduced to :

$$\ddot{\phi}(t) = -4 \times \dot{\phi}(t) + 130 \times \delta(t) \quad (9.1-2)$$

Thus the aerodynamic transfer function in roll, $G_a(s)$, is

$$G_a(s) = \frac{\dot{\phi}}{\delta} = K_a \times \frac{1}{s/b + 1} \quad (9.1-3)$$

where K_a is the aerodynamic gain and b is the plant pole. In our case $K_a = 30$ and $b = 4$.

from Eq 9.1-3 we can see that the steady-state gain is K_a :

$$\frac{\dot{\phi}}{\delta} |_{s.s} = K_a \quad (9.1-4)$$

and therefore

$$\dot{\phi}_{\max} = K_a \times \delta_{\max} . \quad (9.1-5)$$

From Eq 9.1-5 , we can see that the maximum roll rate is 150 ($^{\circ}/\text{sec}$) . In Fig. 37 we show the ϕ and $\dot{\phi}$ response to a step function in $\delta_{\max}$.

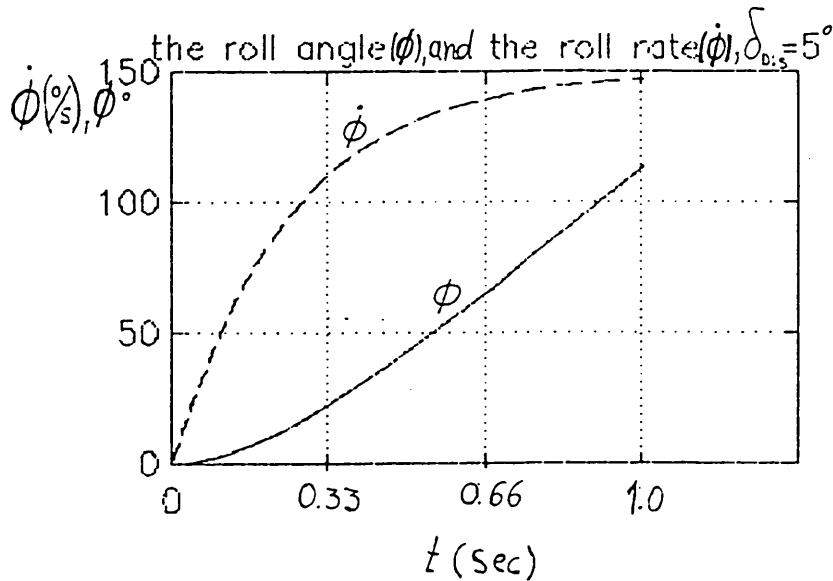


Fig. 37.

9.2 The other block models

The servo (Fig. 36) is described by a quantization function which represents the finite accuracy of such systems, usually servo systems are very fast compared with the whole system , i.e., in our case the plant time constant is $\tau_p = 250$ (msec) while the servo time constant might be $\tau_s = 25$ (msec) . Therefore we may neglect the servo dynamics and its effect on the closing loop design.

The image-processing is described by a time delay - ΔT , which is the time-cycle needed for the completion of the image processing calculation. From the IMAD and PEP aspect we prefer to work with 33(msec) time-cycle which means a maximum

of 5° roll angle per time cycle for the maximum actuator disturbance , which the image-processing procedure is still able to cope with. This time-cycle allow us to save calculation time for further development in the future without major changes. But from the closing-loop performance aspect we would like to decrease the time delay - ΔT to 16(msec), which is synchronized with the video fields and might reduce the sampling effect to a minimum , so that the system will be almost a continuous one .

The filter $G(z)$ is basically a lead-lag filter whose aim is to increase the stability of the close-loop. If the input variable to $G(z)$ is $\phi(i)$ and the output variable is $\phi_c(i)$ then the $G(z)$ recurrence formula is :

$$\dot{\phi}_c(i) = k_1 \times \dot{\phi}(i) + k_1 \times z_1 \cdot \phi(i) - p_1 \times \phi_c(i-1) \quad (9.1-6)$$

where k_1 is the filter gain, z_1 and p_1 are the zero and pole location(in the s domain) respectively.

$$\phi_c(i) = \phi_c(i-1) + \dot{\phi}_c(i-1) \times \Delta T \quad (9.1-7)$$

where ΔT is the calculation time cycle, we will show the results for $\Delta T = 16(\text{msec})$ and for $\Delta T=33(\text{msec})$. It is worthwhile to mention here that $\dot{\phi}_c(i)$ can be calculated easily from the image-processing procedure. Since we want a small steady-state roll angle error $\Delta\phi(i)_{\max} |_{s.s} \leq 1^\circ$ (see § 9 and Fig. 36) we must increase the feedback gain, so that the D.C gain K_T is :

$$K_T = \frac{\delta_{\max}^{\text{dis}}}{\Delta\phi(i)_{\max} |_{s.s}} \times f(\Delta T) \quad (9.1-7)$$

The disadvantage of increasing the D.C gain in non-linear systems is, that it might cause a high amplitude limit-cycle. This is certainly an undesirable effect in systems which are based on video information and we therefore try to reduce the non-linearity by choosing very fast and accurate A/D and D/A converters. The ADC(analog to digital converter) is a 8 bit flash video converter and the DAC is a general purpose 8 bit convertor.

The image processing block is charactrized by a time delay - ΔT and a noise generator with a standard deviation(s.d) of 0.5° which emulates the estimation process(PEP) as shown in Fig. 34-a

9.3 The closed-loop performance

The closed-loop system as shown in Fig. 36 is a non linear system which is quite sensitive to the time delay - ΔT , we compensate the close loop performance by a significant lead in $G(z)$.In Fig. 38 we show the roll angle response to a disturbance in the aerodynamic actuator $\delta_{dis} = 3^\circ$ (step fuction), while the time delay is 16(msec) which means the rate updating is 60(Hz)

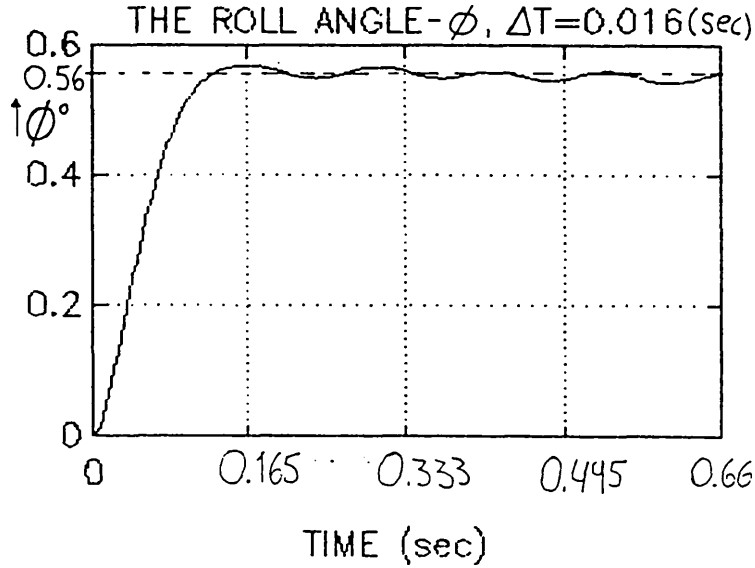


Fig. 38

From Fig. 38 we can see that in the case of $\Delta T = 0.016(sec)$ the closed loop performances are very good, the system time constant is $\tau_{cl} = 120(msec)$, the overshoot and the limit-cycle are negligible , the stability margin (SM) is better then 6(db) and the system sensitivity to noise and other distortions is low. The steady-state error in the roll angle- ϕ is 0.56° for 3° disturbance in the aerodynamic actuator which is quite good. The maximum roll rate in this case is $7^\circ/sec$ (shown in Fig. 40) which is really good performance.

In Fig. 39 we show that the roll angle time-response for the same disturbance while in this case we increased the time delay (ΔT) to 0.033 , namely the feedback updating frequency is 30 (Hz) only. In this case the system is much more sensitive to disturbances, we can see that the overshoot is high (about 40%), this is a result of the low stiffness of the closed-loop system which allows the development of quite a big error in $\epsilon_\delta(i)$ (see Fig. 36) .

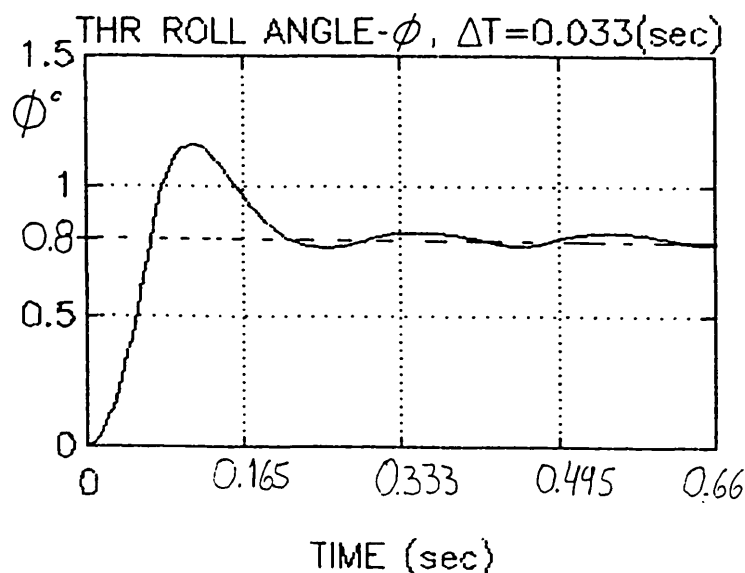


Fig. 39

The closed loop low stiffness is caused by the low DC gain K_T which is reduced by 3 compared with the case of $\Delta T = 0.016(\text{sec})$. The reason that the DC gain is modified is because we must keep a 6(db) stability margin at least, in particular when we deal with air-born vehicle in which the air-speed - V_a is not always constant, which means that the variation of the dynamic pressure - q (Eq 9.1-1) is significant since q is a function of squared velocity ($q = f(V^2)$) and that might have an outstanding affect on the system gain. Therefore eventually we must find a compromise between the stability demands and the closed loop performance like overshoot and limit-cycle. Problems with such tradeoffs occurs usually when the time-delay ΔT is increased as in our case. The system performances as shown in Fig. 39 are the best available with these parameters: although on the one hand we get a high overshoot and the time constant τ_{cl} increases to 250(msec), on the other hand we get a reasonable stability margin, a low amplitude in the limit-cycle and as we can realize from Fig. 40 we get a low value for the maximum roll rate compared with the maximum roll rate allowed (§ 6).

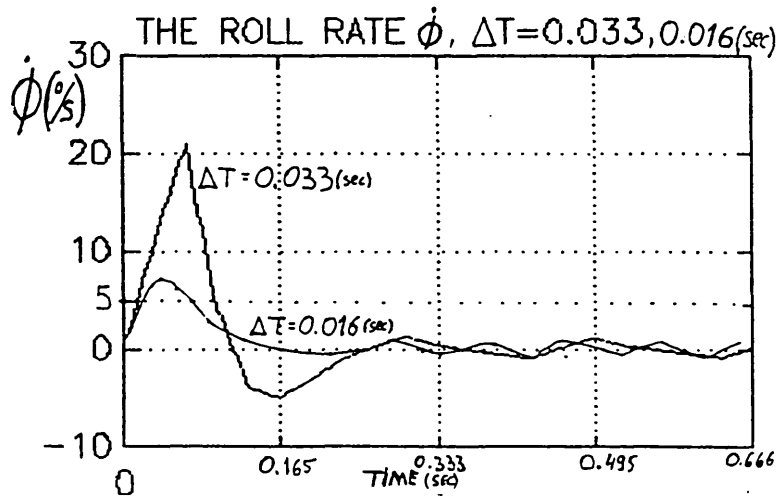


Fig. 40

The conclusion from this chapter is that in our case it is preferable to work with an updating rate of 60(Hz), which means that all calculations regarding the image processing process should be completed in less than 16(msec) . However we strongly recommend to carry on with more research regarding the system updating at a frequency of 30(Hz), in case the calculation burden at 60 Hz exceeds the real time limits . Moreover the updating cycle of 30(Hz) enable us to upgrade the system in the future, for example the next step might be a navigation process based objects identification simultaneously to the stability process as shown in our approach.

CHAPTER 10

10. THE COMPUTATIONAL REQUIREMENTS

The required calculation capability is the main parameter which characterizes the hardware structure. In this work we are not going to deal with hardware implications, but we give an idea of the calculation burden per second. One might expect that with very fast chips and the available technology there is no real problem of implementation .

10.1 Trends in calculation capability.

The first real-time vision systems began to appear at the beginning of the 70's. The hardware occupied about thirteen cubic foot volume and could process a 100-by-100 pixel window at a rate of 10^5 pixels per second (1 frame per second). In order to capture the incoming video signals , an analog storage tube was used , which certainly was a major source of noise and instability.

The next progress was made by using an A/D converter and a 128-by-128 digital frame buffer, replacing the unstable analog storage tube. The system occupied a volume of five cubic feet and could process 10^6 pixels per second , which means 60 fields per second.

In the beginning of the eighties processors with 512-by-512 digital frame buffers began to appear , this processor can perform $400 \cdot 10^6$ operations per second. The processing rate is 10^6 pixels per second, the processor occupied a volume of one cubic foot. Most of those processors consisted of an Intel 8086 single board computer.

The latest version of a preprocessor(AUTO-Q) is described in [8] . The AUTO-Q preprocessor was completed in 1984. A block diagram of the preprocessor is shown in Fig. 41. The preprocessor performs 10^9 operations per second to process $4 \cdot 10^6$ pixels per second. From Fig. 41 we can see the use of software instead of complicated wired hardware, e.g., filtering ,contrast enhancement etc. All stages are under software control frame-by-frame, and each of the blocks enables several options e.g., edge detection includes the gradient or the Sobel method.

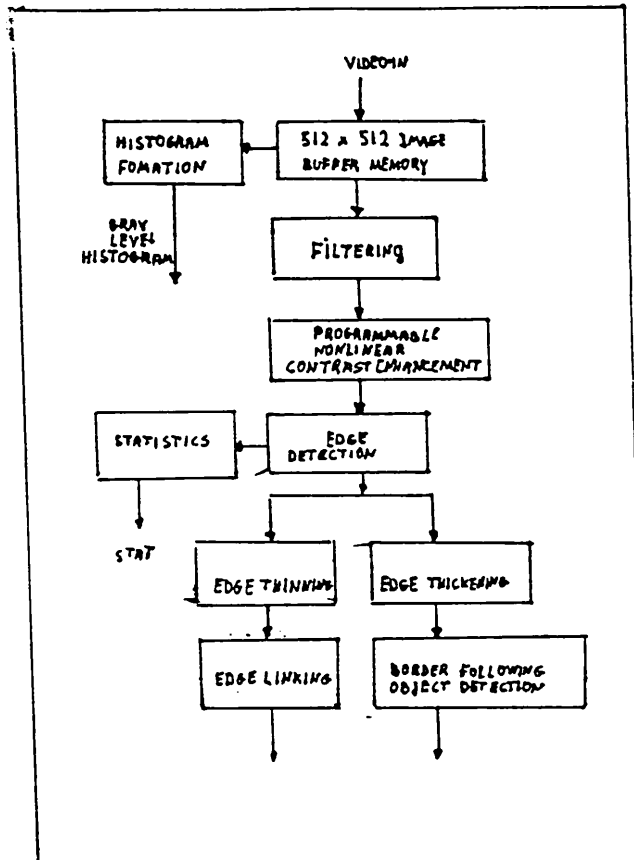


Fig. 41 Pipeline Architecture of a AUTO-Q preprocessor

Image processing now will incorporate VLSI technology. Algorithms will no longer be hardwired but will be programmed via microcode. This processor will occupy a volume of 2 cubic feet and perform $32 \cdot 10^9$ operations per second. And we anticipate that it will be able to process $20 \cdot 10^6$ pixels per second. Because the preprocessor is fully programmable many of the postprocessing functions can be offloaded to the preprocessor eliminating the traditional processing bottleneck.

As hardware technology becomes more sophisticated so too will the algorithms. In the past, hardware limitations allowed only simple algorithms to be implemented. These algorithms have not demonstrated adequate performance, often being erratic and inconsistent.

As we explained previously our approach is based on the attempt to emulate the HVS . We estimate that in order to surpass human detection and recognition capability a machine will have to perform in excess of $100 \cdot 10^9$ operation per second. A throughput of 60 - 100 million pixels per second is anticipated. This level of processing power will be available in the not-too-distant future ! In Fig. 42. we show the development in calculation capacity vs. time . The conclusion from this graph must be that actually the hardware is no longer the main obstacle in improving performance in image processing and artificial intelligence applications.

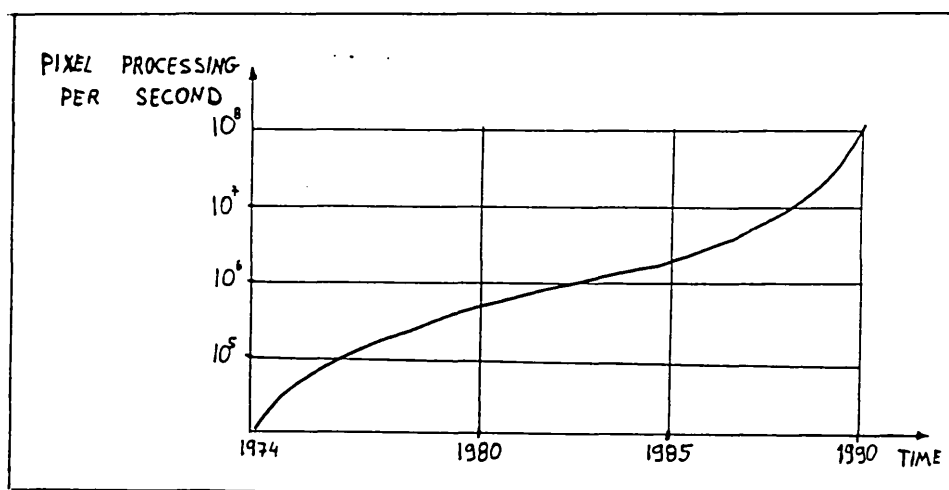


Fig. 42. The pixel processing per second vs. time

10.2 The desired calculation capacity .

The desired calculation capacity is given roughly , by calculating the numbers of additions and multiplications in one time-cycle (ΔT) , we assume that one multiplication is an equivalent to three additions. The number of video fields in one second is 60 and the real-time cycle is therefore $\Delta T = 16(\text{msec})$. There are other parameters which effect the total calculation capacity e.g. , the integration of fast hardware devices which might take a part of the calculation burden, or the decision about the necessity of double precision in each stage, which is dependent in the system hardware itself as well.

10.2.1 The IMAD and PEP burden calculation.

As we can see from Fig. 42 the main processes which must be performed in real-time are the IMAD and the PEP algorithms. The AWL is performed in parallel at a lower speed, but all the processors have to be synchronized . The general configuration for our image processing procedure is the parallel-pipeline configuration.(Fig. 43.)

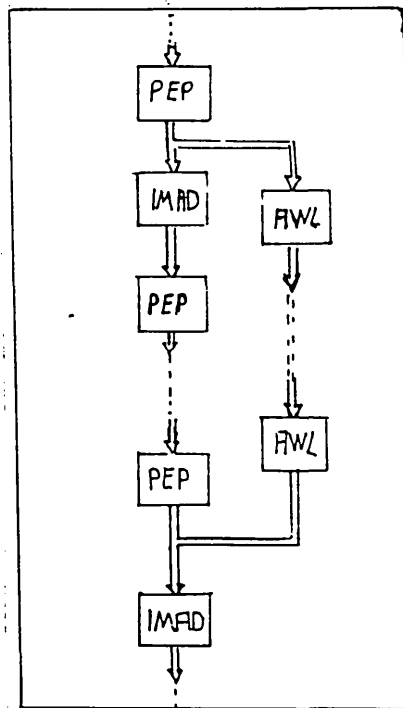


Fig. 43. PARALLEL-PIPELINE CONFIGURATION

The number of instructions in the IMAD procedure in the vertical direction is $N_{V_{ml}}$:

The number of pixels involved $N_{pix} = 4992$

The number of instructions per pixel in the matching process is $N_{ins} = 3$

The total number of instructions per second in the matching loop is $N_{V_{ml}}$

$$N_{V_{ml}} = 60 * N_{pix} * N_{ins} = 898560$$

There are more calculations like the weight function, but their calculation burden is negligible .

The number of instructions in the IMAD procedure in the horizontal direction is the same as in the vertical one.

$$N_{H_{ml}} = 898560$$

Therefore the total number of the IMAD process N_{imad} is roughly :

$$N_{imad} = N_{V_{ml}} + N_{H_{ml}} = 1797120$$

The number of instruction in the PEP procedure N_{pep} per second is :

$$N_{pep} = 7200$$

If we assume that the same amount of calculations have to be performed in both sides of the picture, we get that in the worst case , we need no more then $6 \cdot 10^6$ instructions per second (6MIPS) , which is quite easy to achieve. It is relatively a small number compared with the capability as shown in Fig. 42.

The advantage of our method is the fact that we have to process in real-time only a very small part of the scene which is a result of the AWL process ability to define the 'best area'. For the IMAD+PEP algorithms we roughly require 350 "operations" per pixel compared with 1000 "operations" per pixel estimated

by Reddy [15] , only for segmentation ! This remarkable result is achieved by taking in¹⁵ consideration the real-time aspects from the beginning of the research and in every step along the development of the algorithms, by trying to simplify the algorithms as much as possible. e.g, the 'minimum line' technique as mentioned in §5.3.6 which saves a huge calculation burden . Another major point in real time calculation is to try to find an intelligent way to process only the "most interested templates" in the FOV namely a sort of a 'selective segmentation' algorithm which automatically will refer only to the most interested templates e.g., in our case the AWL algorithm is doing this "job". For controlling some mechanism or navigating a vehicle through some environment , like driving a car on a highway , we suggest to process only half of the FOV. [Appendix- 4].

CHAPTER 11

11. CONCLUSION AND FURTHER DEVELOPMENT

A comprehensive approach for vehicle stabilization based on image analysis methods is presented. We assumed that the observed terrains are uniform and that the roll rate could be very high, i.e., $\dot{\phi}_c = 120^\circ/\text{Sec}$, which makes the problem more severe. We have successfully proposed three algorithms to tackle the problem: the AWL which is a decision criterion, designed to answer the question 'where is the best place for the reference-template?', the second is the IMAD which is a matching process combined with weight function, which is essential in uniform pictures and the third is the PEP which is a fast and simple estimator, utilizing information from both sides of the picture. We tested successfully the whole process with real data representing deserts and others undetailed landscape, while the disturbance were in rotation, translations and scale-factor, from short ranges to long ones. The major conclusion is that by using the suggested method we are able to get an accurate estimation of the roll angle $\hat{\phi}$, and are able to use it for closing the roll loop. As a by product we get the vertical and horizontal displacements $\hat{d}_v$ and $\hat{d}_h$ respectively and the scale-factor sf , i.e., the closing-range $\hat{X}_c$ which might be used for navigation purposes. There are several others secondary conclusions from our research which we want to point out:

1. The first is a general one; it is worthwhile to define before the beginning of the research the priorities between the desired results regarding the estimation process, and their time-constant. By doing it we might simplified the whole process, because the whole approach is adapted to the specific model defined and to the priorities between the estimated variables.
2. In order to get substantial information from pictures representing uniform terrains, it is better to process a group of pixels (hpix), rather than each pixel.
3. The use of the Extended Sobel Filter (ESF), proved to be very useful, particularly when the histogram function is very narrow.
4. The technique of a simultaneously parallel processing the left and right sides of the picture enables us to get highly reliable results, and therefore is highly recommended.

5. The decision about the most detailed template in the the search area, must consider the way the pixels are distributed in the template as well as their histogram function.
6. For a successful matching process it is essential to work with quite a big number of pixels involved , but there is a threshold which depends on the picture properties. Above this threshold it is useless to increase the number of the pixels involved in the process . As an example, for uniform pictures a reference window of 24×16 (pixels) or 24×24 (pixels) , is quite reasonable.
7. The use of overlapping technique in matching process is essential, since it decreases the likelihood of false matching .
8. It is highly recommended not to base the decision about the matching point only on the minimum value of the matching process results, but to consider the gradients along the matching process results as well .
9. There is no need for a pixel by pixel horizontal shifting , if the reference windows are as recommended and the pictures represent natural terrain .
10. It is preferable to work with a time cycle of 16(msec) although the closed-loop performance with 33(msec) time cycle are not bad. The main advantage of the 16(msec) time cycle is the stability margin of the close loop based on the plant model we work with.
11. The computational requirements in order to perform the algorithms we suggest is a system with a processing ability of 6 mips which is quite easy to achieve .

Some obvious and desirable extensions of the research reported herein are as follows:

(a) In order to achieve more accurate results from the matching process (IMAD) and from the estimation process (PEP) , it is worthwhile to utilize sub-pixel methods , i.e., as mentioned in [17] .

(b) It is possible to operate a logic procedure which will 'push' the search-area towards the centre of the picture if we cannot get a satisfactory picture quality factor(PQF) at the search area (A_L and A_R) as are defined in §5.2 .

The point here is that when the PQF is low, it may be worthwhile moving the reference window towards the centre of the FOV in search of a higher PQF, even though this reduced the "swing effect" (see §8.1).

(c) It is worthwhile to check the relationship between overlapping (see §7.3.1), namely the horizontal size of the reference windows, and the quality of the matching process results (see §7.4). It might be that under some circumstances, e.g., a very low number of the PQF, it is better to increase the size of the reference window and by that to increase the horizontal overlapping, although according to our approach we already supposed ourselves to be on the 'minimum line' (see §7.3.4). But the point is that the result from the vertical matching process might be less accurate than ± 1 pixel from the right estimation, as a result of this vertical error the 'minimum line' might be shifted from its 'true minimum line' inevitably leading to a false matching in the horizontal direction. An increase in the horizontal dimension of the current and reference windows as a function of the PQF might avoid this.

(d) In order to decrease the variability of the estimated values as a result of poor performance of the matching process (see §8.3), it is possible to find a procedure which will push the current window towards the picture edges in order to reduce the error caused by the scale-factor(sf). However this procedure must let the $\hat{X}_c(i)$ process to converge, and that might happen after five or six frames after the other variables ($\hat{\phi}(i)$, $\hat{d}_h(i)$, $\hat{d}_v(i)$) converged. the calculation of $\hat{X}_c(i)$ after the 'spread-out' shift, will be based only on its physical model described by Eq- 8.2-11.

REFERENCES

- [1] A.Rosenfeld and A.C. Kak, "Digital picture processing" Academic Press. 1982 .
- [2] R.C.Gonzales and P.Wintz, "Digital image processing" Addison Wesley publishing 1987.
- [3] H. H Bailey "Image correlation Part I and II" Rand corporation R-2075/1-pr Nov. 76
- [4] Hans C. Bergman "Analysis of different displacement estimation algorithm for digital television signals. NATO ASI series.
- [5] Robert J. Schalkoff "Algorithm for real-time automatic video tracking system. Ph.D. research, E.E. Dep. University of Virginia , May 1979.
- [6] M. Kabuka, E . Mcvey and P . Shironoshita, "An adaptive approach to video tracking" ,IEEE Trans. Robotics and Automation.vol 4 1988.
- [7] F.C. Billingsley "Noise considerations in Digital Image processing hardware" Berlin : Springer-Verlag ,1979, pp.249-280.
- [8] Richard F. Kroupa, Bruce J. Schechter "Art development at Westinghouse" Westinghouse Electric Corporation ,Baltimore, Maryland
- [9] Milton Makris, Joe Sawicki, Steve Iannetti "Real time autotracking/image processing architectures for field applications" Honeywell,Electro-optics Division , Lexington, MA 02173.
- [10] H.G.Meissner and E.D Dickmans , "Control of an unstable plant by computer v ision" ,in Image Sequence Processing and Dynamic Scene Analysis. Berlin:Springer-Verlag,1983, pp.532-548.
- [11] R. Ohlander , "Analysis of natural scenes , " Ph.D. dissertation , Dep. Comput. Sci. , Carnegie-Mellon Univ. , 1975.
- [12] D . Kuan , G . Phipps , and A. Chuan Hsueh , "Autonomous Robotic Vehicle Road Following" in IEEE , Pattern Analysis and machine intelligence Sep. 1988 ,vol , 10.
- [13] A . Hsueh "A terrain typing vision system for autonomous vehicles, "FMC Corp. , Internal Tech Rep. , 1986
- [14] Claus E . Liedtke . "Image segmentation considering properties of the human visual system" Hannover University West-Germany. NATO , ASI Series , Vol. F2.
- [15] R . Reddy , "Pragmatic aspects of machine vision" in Hanson and Riseman(ed): Computer vision systems . pp . 89-98 , Academic-Press.

- [16] W. Niblack "An introduction to digital image processing" Prentice/Hall international. 1990.
- [17] W. E. Snyder and S. A. Rajala "Track aquisition of sub-pixel target" Department of Electrical Engineering North Carolina State University, Raleigh, North Carolina 27650/USA

APPENDIX 1 : A sample of uniform pictures we work with

Some other samples of pictures we work with are shown here. All the pictures represent uniform terrain with out any particular object or target. We prefer to base our research on pictures (a), (b) , and (c) (see Fig. 1) , because each of them represents some different parameters like : the range the picture was taken, the variety of features these pictures represent as a clear line between sky and ground(pic (b)) or a narrow path between the hills etc. But we examine the suggested algorithms with some of these other pictures and the results were quite similar to the results shown for pictures (a) ,(b), and (c).

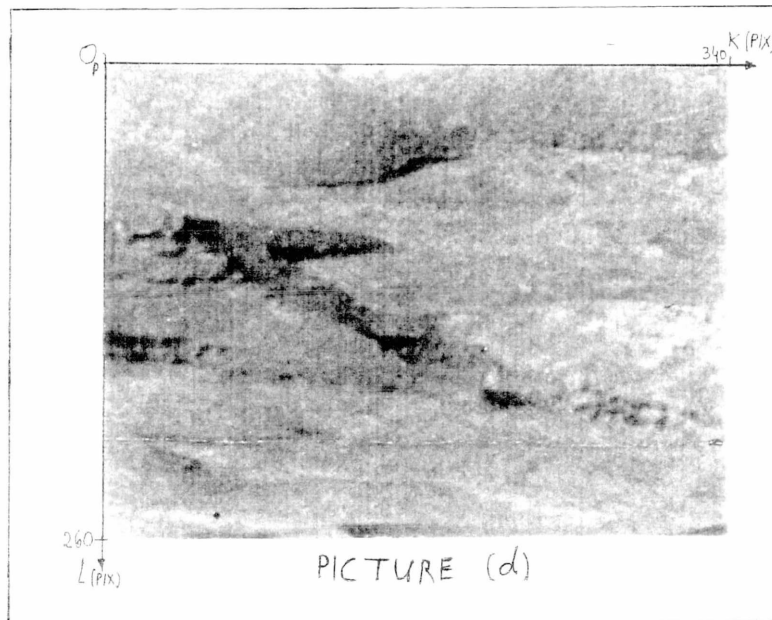


Fig. 44. An example of a very uniform picture from a medium range

Picture (d) is quite similar to picture (a) ,(in Fig. 1) since both represent uniform terrain and were televised from quite similar distances. But still we can see that the pictures have different distributions of video intensities.

Another interesting case is the terrain represented by picture (e) , this was televised from a big distance(10Km) , but it is relatevily quite a detailed picture. In Fig. 45 we compare the picture 45-a and its digitized image 45-b. Picture 45-b is digitized by a 6 bit ADC ,but the display and the print out are with a resolution of 16 grey level (4 bit) , we can see that even with 16 grey level the results are

quite good .

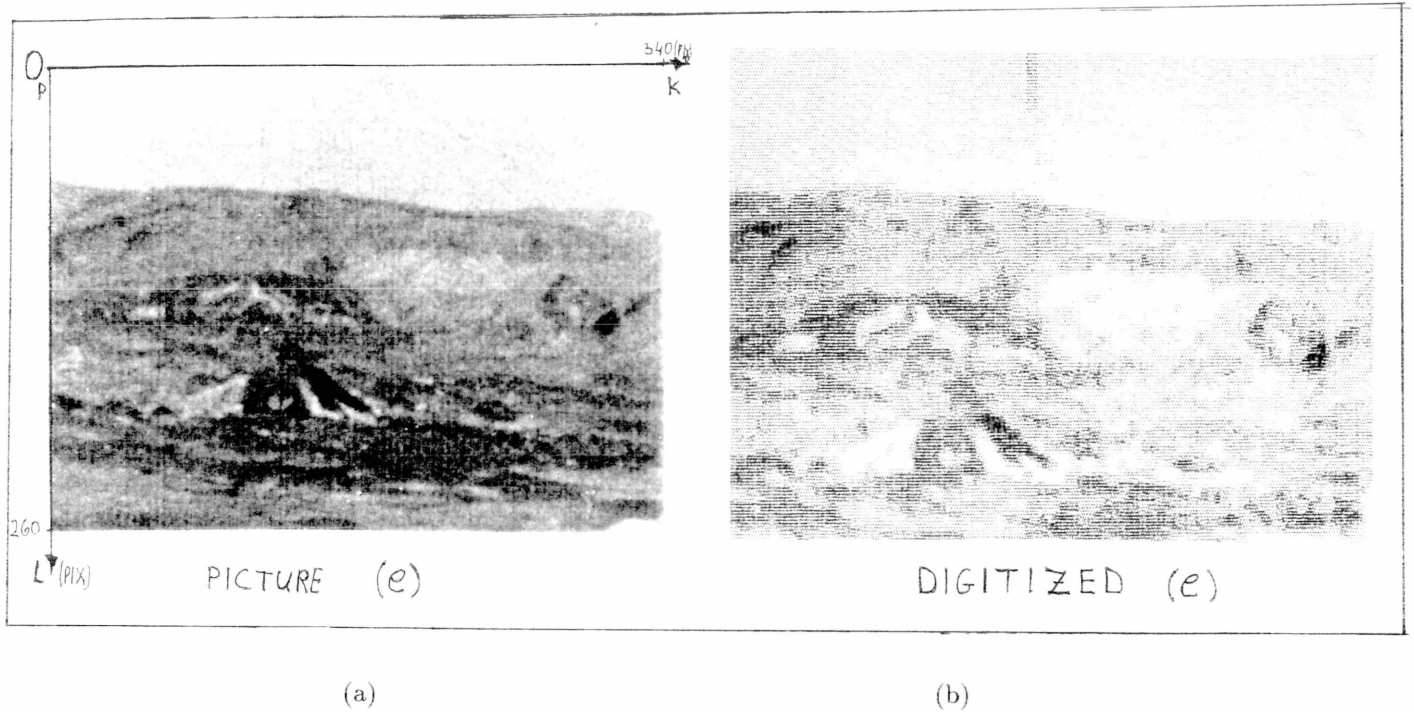


Fig. 45 A comparison between the picture and its digitized image

Other sorts of pictures we work with are represented by picture (f) and picture (g). Picture (f) is a very uniform one and was televised from a remote distance ($r_0 = 10(\text{Km})$), while picture (g) was televised from a short range ($r_0 = 2.5(\text{Km})$) and although the terrain is uniform, because of the relatively short range the AWL algorithm quite easily identifies some of the detailed regions inside the picture.

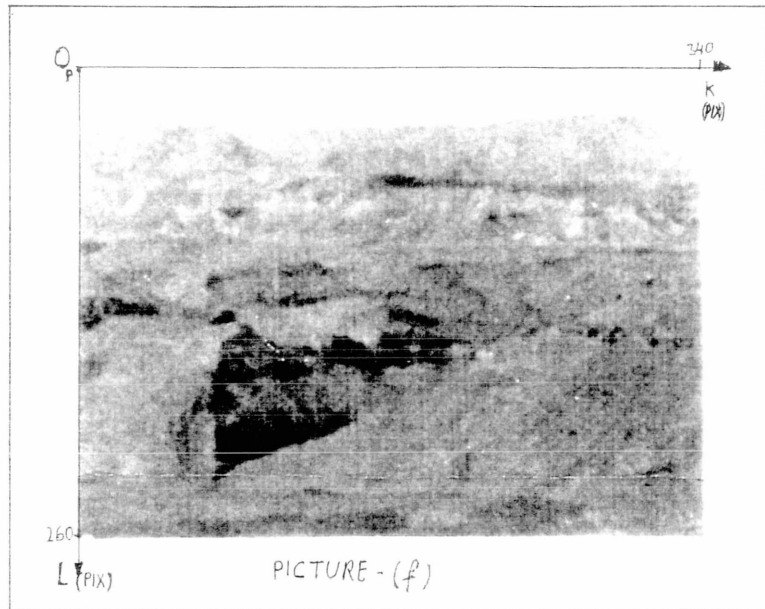


Fig. 46 An example of a very uniform picture

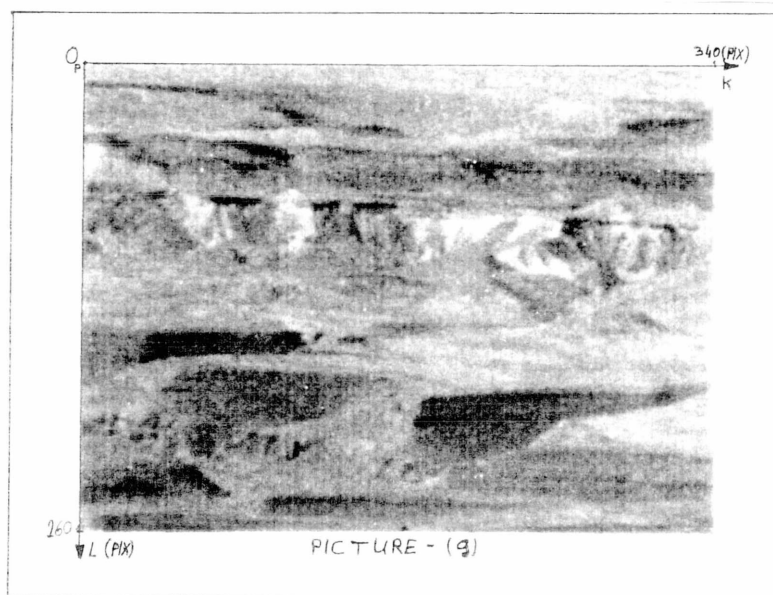


Fig. 47 An example of the 'improvement' .vs. the range

APPENDIX 2 : The bilinear interpolation .

During our research we built a program which simulates the whole system including the frame to frame disturbances. We simulate the rotation by rotating the template we work with, in the opposite direction to the camera rotation, using the equations in chapter 8.

As a result, pixel $I(l',k')$ which is located in grid (L,K) might be transferred to coordinates (l',k') , which define a new grid (L',K') , where (L',K') do not match the (L,K) grid . Therefore we operate the bilinear interpolation in order to define the new video intensity of pixel (l',k') and the other pixels as well.

Let us assume that pixel I' which has coordinates (l',k') is surrounded by four pixels which are (l,k) , $(l,k+1)$, $(l+1,k)$ and $(l+1,k+1)$ in the (L,K) coordinates. Our aim is to find the video intensity I' from those pixels.

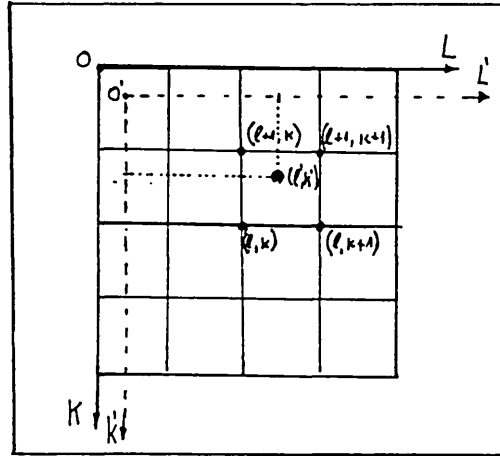


Fig. 48. The transformed coordinates

We define :

$$(l_i, k_i) = \text{int}\{ l', k' \} \quad (1)$$

where $\text{int}\{ \}$ is the integer operator.

$$\alpha = l' - l_i \quad (2)$$

and

$$\beta = k' - k_i \quad (3)$$

$$A_{11} = (1-\alpha) \times (1-\beta) \times I(l,k) \quad (4)$$

$$A_{12} = (1-\alpha) \times \beta \times I(l+1,k) \quad (5)$$

$$A_{21} = \alpha \times (1-\beta) \times I(l,k+1) \quad (6)$$

$$A_{22} = \alpha \times \beta \times I(l+1,k+1) \quad (7)$$

$$I'(l',k') = \text{nint}\{ A_{11} + A_{12} + A_{21} + A_{22} \} \quad (8)$$

Where $\text{nint}\{ \}$ denotes the nearest integer .

This method is particularly suitable for pictures we are working with, since in pictures representing natural terrain there are no sharp changes among pixels located in the same region and therefore designating the grey level (intensity) of an arbitrary pixel as a weighted function of its closest neighbors seems to be reasonable.

APPENDICES - 3 : AN EXAMPLE OF THE REFERENCE AND CURRENT WINDOWS

In this appendix we give two examples of the relative location between the current window and the reference one. the first case (3-a) describes the nominal case (see Fig. 18) , and the second case (3-b) describes the case of a rotation disturbance, $\dot{\phi} = 100(^{\circ}/\text{sec})$. Both cases relate to picture (a).

```

THIS REFERENCE WINDOW IS A PART OF PICTURE: (a)
-----
ROW NU BEGIN MA= 25
COL NU BEGIN MA= 9
TOTAL WINDOW SIZE 16 * 24 PIXELS
SFX=1.0 SFY=1.0 A= 0 B= 0 FEE= .00

42434341414141404042424141424241 42414041424241404140403939403942
41424341414141413940414241414142 40414041404341404141414040414041
41414240404141404041414142414043 40404040414241404040404039434240
40424240414141414240404041424143 41404140414142404041414040424140
40404140404343424341414241424144 40414142414142404042404142404142
403841424041424242424442434342 39424141424141414043414243404142
40394140404040414242424241424241 41404041403940403942404243424242
41424241414040424239414039404039 41414040403939404042404142434242
41414140414040404039404040393940 41414140403939414042414242434242
42404040414242404140404141404140 42394141414040424041414242434342
38393942424242414342404141434241 43414241414241424241414241414242
40414143424343414242424242434242 43424241414342444342434242404140
424242424343434343434344444444 42424242424342434441414242404141
444343424343424243434343444443 42424242424140424241414242424342
43434242434242424243424243434142 44434242424141404241394342434342
42414040414241424242424141424243 42424243414242424241414241434242
41414040404140414040404241414242 42414242434041424240414342424243
40404040393939403941403939404140 44434342434241434342414341424242
39394138383839403938383837373738 43424343434242444343434242384040
39373837383938383838383737373736 42424344444343434241423939363836
39383838383837383838373637363535 4242444343414141413839383737373837
38373737373837383838373737353536 43424340413940383938383737383838
383837262735363837383737353535 42414138393737383837383738373939
38373727363637373736363635363637 42404038393736393836373838373939

THE AVERAGE VIDEO INTENSITY IN C1 AND C2 RESPECTIVELY
40.0000000 41.0000000

THIS IS THE CURRENT WINDOW
-----
ROW NU BEGIN MAWL= 28
COL NU BEGIN MAWL= 8
ROW NU BEGIN MAWR= 28
COL NU BEGIN MAWR= 8
TOTAL WINDOW SIZE 16 * 16 PIXELS
SFX=1.0 SFY=1.0 A= 0 B= 0 FEE= .00

40404242404141414142404040414241 41414041404141424040414140404241
40404041404043434243414142414241 41404141424141424040424041424041
41403841424041424242424244424343 41394241414241414140434142434041
41403941404040404142424242414242 41414040414039404039424042434242
42414242414140404242394140394040 41414140404039394040424041424342
41414141404140404040394040403939 42414141404039394140424142424342
40424040404142424041404041414041 41423941414140404240414142424342
39383939424243424143424041414342 43434142414142414242414142414142
41404141434243434142424242434242 42434242414143424443424342424041
434242424343434343434344444444 42424242424243424344414142424041
4444434342434342434343434444343 42424242424241404242414142424243
43434342424343424242434242434341 43444342424241414042413943424343
41424140404142414242424242414342 43424242434142424242414142414342
40414140404041404140404042414142 42424142424340414242404143424242
40404040403939394039414039394041 42444343424342414343424143414342
413939413838383940393839383739 43434243434342424443434342423840

YLM DYL B DYL A EP S L YRM DY R B DY R A EP S R
---
1.07 -.35 .28 -1.00 .28 -.13 .09 -1.00

```

(3 - a)

THIS REFERENCE WINDOW IS A PART OF PICTURE: (a)

 ROW NU BEGIN MA= 25
 COL NU BEGIN NA= 9
 TOTAL WINDOW SIZE 16 * 24 PIXELS
 SFY=1.0 SFY=1.0 A= 0 B= 0 FEE= .00

```

42434341414141404042424141424241 42414041424241404140403939403942
41424341414141413940414241414142 40414041404341404141414040414041
41414240404141404041414142414043 40404040414241404040404039434240
40424240414141414240404041424143 41404140414142404041414040424140
40404140404342424341414241424144 40414142414142404042404142404142
40384142404142424242424442434342 39424141424141414043414243404142
40394140404040414242424241424241 414040414039404039424042434242
41424241414040424239414039404039 41414040403939404042404142434242
41414140414040404039404040393940 41414140403939414042414242434242
42404040414242404140404141404140 42394141414040424041414242434342
38393942424342414342404141434241 43414241414241424241414241414242
40414143424343414242424243424342 43424241414342444342434242404140
424242434343434343434344444444 42424242424342434441414242404141
44434342434342434343434344434345 42424242424140424241414242424342
43434242434342424243424243434143 44434242424141404241394342434242
42414040414241424242424241434243 42424243414242424241414241434243
41414040404140414040404241414242 42414242434041424240414342424243
40404040393939403941403939404140 44434342434241434342414341434243
39394138383839403938393838373938 4342434343424244434343424384040
393738373839383838383838373736 42424344444343434241423939363836
39383838383837383838373637363535 42424443434141414138393837373837
383737373837383838373737353536 43424340413940383938383937383838
38383736373536383637383737353535 42414138393737383838373837383939
383737373637373736383636353637 42404038393736393836373838383939

```

THE AVERAGE VIDEO INTENSISTY IN C_L AND C_R REPECTIVLY

40.0000000 41.0000000

THIS IS THE CURRENT WINDOW

 ROW NU BEGIN MAWL= 33
 COL NU BEGIN NAWL= 7
 ROW NU BEGIN MAWR= 22
 COL NU BEGIN MAWR= 10
 TOTAL WINDOW SIZE 16 * 16 PIXELS
 SFY=1.0 SFY=1.0 A= 0 B= 0 FEE= 3.03

```

40393939404242434241434041414041 41424041404242424140414040404040
38393940424243424143424041414342 41414141424141424140404040404040
41414142434243434142424141434241 41404142414141414140404041414141
43424242434343434343434243424342 41404142414041414040394040404141
44434343424343424343434443444444 40404042414040414140404041414141
43434242434342424242434344434344 40414142414040404040404142414041
41414140404142414241414242434243 41414141424040414140414141414141
4041404040404040404140404241424243 41404241414140424141424140414241
40404040393939394039404140414141 41404140404040424141434241424241
403939393838383939383939394039 414140393940404141414242424241
393838383838383838383837373636 40414039394040414141424243424242
383837373738383738383737373636 41424140404140414142424243424243
383837373736363736373737363535 42424142414242414142424142424242
383837373736363736383737353535 42424143424343424242424141414040
373738383737383837363636363636 42424243424344424142424140414140
373738383838383836363737373738 43424241404242414142424242434241

```

YLB	DYLB	DYLA	EPSL	YRM	DYRB	DYRA	EPSR
1.26	-1.19	.40	3.00	.11	-.06	.04	-5.00

(3 - b)

In case (3 - b) we can see that the axis origin of the reference window (M_a, N_a), is modified from location (28,8) in both sides (see 3-a) to (33,7) and (22,10) in the left and right side of the picture, we can see that the major displacement is in the vertical direction, in fact if we look to the bottom of (3 - b) we can realize that the estimated errors are quite accurate (EPSL=5 and EPSR=-5 instead of -6).

This displacement is caused by the rotation of 3° in one frame (ΔT in this case is 33(msec) and the scale-factor is $sf=1$, which means it is a freezing range). Another important result is the average of the video intensities, 40 and 41 for the left and right current windows. We can see that they are unchanged from the nominal case (3-a), although we get quite a significant displacement as a result of the current disturbance, due to the uniformity of the picture. (See §6.2)

APPENDICES - 4 : THE SPATIAL RESOLUTION VERSUS THE LINE OF SIGHT ANGLE (λ)

As we discussed in chapter 5 the best place for locating a reference window for matching process is : the most detailed template. The possibility to find a detailed template in the FOV depends on the observed terrain, but if we assume that our FOV includes such regions , the best way for defining the search area is through the spatial resolution definition which is, the terrain area each pixel represent [16] and that is dependent on the line of sight angle (λ) (Fig. 49.)

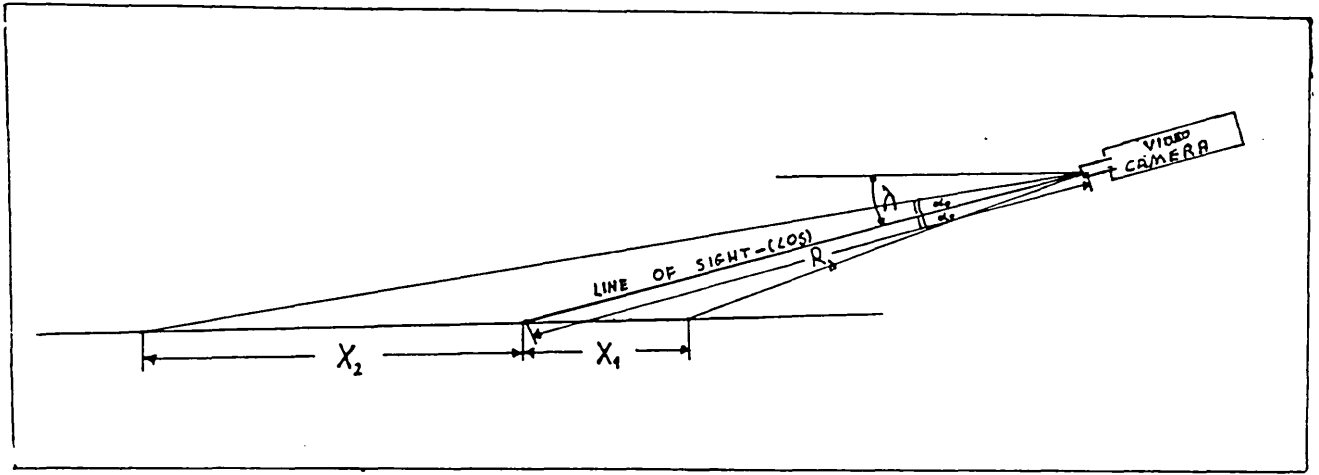


Fig. 49. The geometrical connections and the LOS definition

If the line of sight(LOS) angle (λ) is small , $\lambda < 15^\circ$ and the vertical angle FOV is $2\alpha_0$ then :

$$X_1 \simeq R_\lambda \times \frac{\alpha_0}{\lambda + \alpha_0} \quad (1)$$

$$X_2 \simeq R_\lambda \times \frac{\alpha_0}{\lambda - \alpha_0} \quad (2)$$

If for example $\lambda = 2 \times \alpha_0$, then X_1 is equal to $\frac{R_\lambda}{3}$, and X_2 is equal to R_λ .

We can see that for small LOS angle, the difference between X_2 and X_1 is significant, which means that the spatial resolution of the upper part of the screen is much less than the lower part of the screen.

If we assume for the above example $R_\lambda = 3000$ (m), then $X_1 = 1000$ (m) and $X_2 = 3000$ (m) and

since in our case we assume 260 pixels in the vertical direction, therefore both X_1 and X_2 are represented by 130 (pixels) and their spatial resolution SP_{X1} and SP_{X2} are :

$$SP_{X1} = \frac{X_1}{130} = 7.7 \text{ (m/pixel)} \quad (3)$$

and

$$SP_{X2} = \frac{X_2}{130} = 23 \text{ (m/pixel)} \quad (4)$$

We can realize from this simple example that for small angle of LOS , the ability to distinguish differences between pixels is much higher in the lower part of the picture (which is represented by X_1) , rather than in the upper part of the picture (which is represented by X_2).

As a conclusion and in particular for vehicle navigation, the preferable area for locating a reference-window is the lower part of the FOV. Of course there is a compromise between the this demand and a demand for sensitivity , therefore we suggest that the search area for this case will be composed from 1/3 upper FOV and 2/3 lower FOV which gives a total of 0.5FOV as a search area .

For the case of our research , in most of the air-born trajectory the LOS angle are high , so that the above analysis is less relevant.