AN OPTIMIZED VISUAL TRACKING SYSTEM APPLIED TO ROBOT NAVIGATION

José Maurício Santos Torres da Motta
Universidade de Brasília, Depto. Eng[a] Mecânica, - GRACO - Grupo de Automação e Controle, Brasília –DF, 70910-900, Brazil
jmmotta@unb.br

Sérgio Roberto Gonsalves Tourino
Universidade de Brasília, Depto. Eng[a] Mecânica, - GRACO - Grupo de Automação e Controle, Brasília –DF, 70910-900, Brazil
tourino@engineer.com

Antônio Piratelli Filho
Universidade de Brasília, Depto. Eng[a] Mecânica, - GRACO - Grupo de Automação e Controle, Brasília –DF, 70910-900, Brazil
pirateli@unb.br

Abstract. *This article reports the development of an optimization scheme based on factorial design analysis applied to a vision tracking system embedded on a mobile robot aiming at speeding up the vision system responses for navigation. The approach is intended to be used with an automated routine for finding optimal parameters for feature tracking and position location in real-time, according to the object/environment in sight and its distance from the camera. A review of factorial design is presented along with implementation details to identify and optimize sensor parameters. The method was applied to a model that uses SSD (Sum of Squared Differences) correlation for feature tracking and a gradient-based optical flow estimation for position calculation. Results show which are the most important parameters and the influence they have on time expense for image processing and position error estimation, including parameter interactions. A technique for finding optimal parameter values for each image is implemented and discussed.*

Keywords: *Robot Navigation, Visual Tracking, Factorial Design, Parameter Optimization*

1. Introduction

Visual Servoing of mobile robots is a thriving approach to the control of robot navigation since it emulates human sense of vision. The best control of a mobile robot would be achieved by constructing a complete three-dimensional world model, planning a path and then executing the required steps to move the robot along the path. However, many challenges are still posed ahead as object recognition, obstacle avoidance [1,2] and sensor fusion (Trucco and Verri, 1998; Klette et al, 1998).

Robot navigation based on visual tracking has had a significant amount of research work in the last years (Trucco and Verri, 1998; Klette et al, 1998; Corke and Hutchinson, 2000; Kara et al, 2000). A robust robot tracking system can provide information about the relative 3-D motion of a target relative to the observer and may simplify the retrieval of an object shape and/or localization.

Several researchers have published different approaches to the problem of visual servoing of robots and many methods have been presented for tracking a moving object over a sequence of images, based on optical flow, image correlation or deformable contours (Santos-Victor and Sentiero, 1993; Kass et al, 1998; Santos-Victor and Sandini, 1997).

Plakas e Trucco (1998, 2000) showed an uncalibrated camera model to keep robustness in underwater environment. The computer-vision algorithm was based on projective reconstruction of image points in a stereovision system calculating relative distances between points. The algorithm uses SSD correlation used by Tomasi-Kanade-Shi (1991, 1994) and an automatic scheme for rejecting spurious features using residual calculation. The system showed to be robust in artificial underwater environment. However, actual absolute distances cannot be recovered using image points without knowing the absolute positions of at least five feature points.

Espiau (1993) presented an investigation that strengthened the conclusions of Plakas and Trucco, showing results from a vision controlled system with little influence from camera calibration errors. The main conclusion was that vision systems in closed-loop can compensate camera calibration errors, meaning that camera calibration in those systems has minor importance.

Spindler (1998) presented an estimation of the apparent bi-dimensional motion induced in the image sequence by a subsea vehicle in order to compensate for it, using an affine model and a gradient-based image optic flow.

The research related to this article aims at presenting an optimization scheme based on experimental techniques, namely factorial design, to investigate sensitive parameters in an image-based tracking system, in the case of using SSD correlation and a gradient-based optical flow calculation. Experimental tests were carried out using a robot simulator, which was validated by comparing the speed curve obtained with the simulator and a physical mobile robot, a NOMAD

XR4000 with a monocular camera. A fuzzy logic controller was used to cancel the apparent motion from the captured images. This article solely reports the design of the main parameters of the vision tracking system.

2. Factorial Design

Factorial Design is a technique to design an experiment or to estimate how changes in input parameter may affect the system output (Coleman and Montgomery, 1993; Montgomery, 1991). The ISO Guide to the Expression of Uncertainty in Measurement (1995) mentions the usefulness of analysis of variance techniques to determine measurement uncertainty.

Before conceiving the experimental design and applying a performance test, it is necessary to consider the amount of time spent to perform experimental runs. In factorial design, the total number of runs (N) is determined using the expression $N = (L)^V$, where L is the number of levels of each parameter and V is the number of experimental parameters investigated. As an example, when studying four parameters with three levels each, the total number of runs is $3^4 = 81$ (Montgomery, 1991).

To achieve a generic factorial design, a fixed number of levels (or values) can be selected for each of the parameters (or factors) and the experiments are carried out with all possible combinations. A level of a parameter refers to the discrete values of that parameter domain. For example, if an experiment is such that the studied temperature has the values of 20°C, 50°C and 100°C then the experiment has 3 levels associated to the variable temperature (parameter). If there are l_1 levels for the first parameter, l_2 levels for the second,..., and l_n levels for the n-th parameter, the complete arrangement of $l_1 \times l_2 \times ... \times l_n$ runs of the experiment is called a factorial design $l_1 \times l_2 \times ... \times l_n$.

If only two levels l_n of the parameter n is considered, then it is referred to as a factorial design of the type 2^n. The factorial design in two levels: a) requires a low number of runs per studied factor and, even so it is not possible to get full information about the effects due to each factor, it allows future experiments to be better planned; b) when combined with the concept of fractional factorial design allows few experiments to be carried through even with a large number of factors; c) can be used as blocks of experiments whose complexity can follow the requirements established during the proper experimentation; d) the interpretation of its results can be accomplished whether using the common sense or by means of elementary calculations and e) enables the average main effect of each factor to be estimated along with the interaction effects between factors.

Factorial design possesses an important faculty of showing up the interaction between parameters, but this does not mean that these interactions are numerically appreciable. The main effects of a single parameter tend to be larger than the effects of interactions between two parameters and those larger than three, and so on.

The number of runs or experiments performed in a factorial design 2^n increases geometrically as n increases. However, in many cases the information desired from the experiments can be obtained with only a part or fraction of the total runs, leading to the concept of fractional factorial design.

Fractional factorial design disregards the effects from interactions of higher orders to reduce the number of experiments. In this case, the fractional factorial design may be defined on the basis of the original exponent minus the fraction factor. For an original factorial design of type 25 using 1/4 of the experiments it is shown that $1/4.2^5 = 2^5.2^{-2} = 2^{5-2}$, referred to as a factorial design of the type 2^{5-2}. However the greater the fractionating the less confidence there will be in the results, since some interaction used as fraction generator may be highly insignificant.

3. Vision-Based Tracking System Design

The mobile robot whose vision tracking system in concern is mounted on can be shown in Fig. 1, comprising a control system, sensors, communication and programming, designed aiming at R & D in robot manipulation, computer vision, sensor navigation and learning. As a subsystem the robot includes a monocular vision system with a colored TV camera and a video capture plate with 45 Mbytes/s of transfer speed, generating images with resolution of 256x256 pixels to this application.

Software for the Image System are not available. The navigation system that is intended to be developed in this research is composed by the four main modules that are pictured in Fig. 2.

This article reports the second and third blocks conception and their parameter design. Pre-processing includes filtering, image thresholding and gradient estimation. The motion calculation step includes optical flow estimation using SSD and correlation between image frames in sequence. Position estimation is the output.

4. Optical Flow Calculation

Optical flow calculation is implemented on the basis of the correlation method. The correlation algorithm used is based on SSD, in accordance with the equation below (Anandan and Bergen, 1992):

$$S(c, d) = \sum_{j=-n}^{n} \sum_{i=-n}^{n} W(i, j)[I(c + (i, j), t) - I(c + (i, j), t + 1)]^2 \tag{1}$$

where *W(i, j)* represents a window of weights, *I(c, t)* is the image intensity in *c* = (x, y) in time *t* and *d* is restricted to a square shaped neighborhood equal to $(2n + 1)^2$, centered in *x*.

S: correlation between images in time t and t + 1

W: window of weights

c: image position (x,y) in image coordinates

d: neighborhood of the point (x,y) in the image

n: half-width of neighborhood d, in pixels

I: image intensity value

t: time of capturing the previous image

The speed of the correlation calculation is influenced mainly by two factors: the correlation window size and search window size.

The search window defines the search area on the image inside of which the correlation window calculates the maximum similarity between points of two captured images within a time interval. The search window defines a set of N points (pixels) in the image (square) that will be temporarily the central point (pixel) C(x, y) of the correlation window. Once the local correlation in (x,y) is calculated its value is stored and a new correlation is calculated in the next point inside the search window. The point P(x, y, t) with the largest correlation (minimum S in Eq. (1)) is set to be the central point of the search window, B(x, y, t+1), of the next image.

The correlation window is a square shaped matrix of neighboring points of a central point C(x, y). These points are used in Eq. (1) for the calculation of the correlation values between two images in a sequence.

Figure 1. Nomad XR4000 mobile robot (Nomadic Technologies)

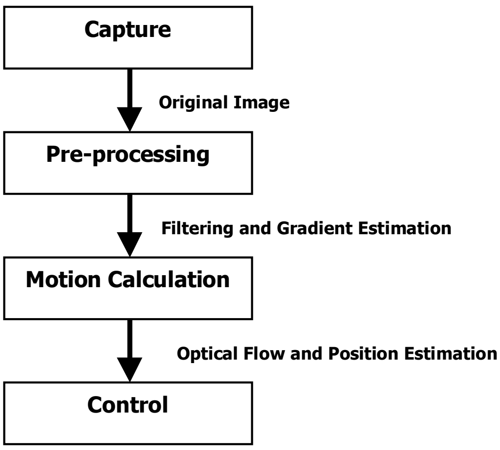

Figure 2. Vision tracking system fluxogram.

Since all points with the largest correlation between two images are calculated the velocity field of each position (x,y) can be estimated ($v = (v_x, v_y)$), producing the optical flow. The velocity is calculated from the approximated temporal derivative using finite differences. The velocity data are used to anticipate the next position of a feature point in the image, locating the new central position of the search window in the next image.

5. Image Motion Estimation

To calculate the image motion feature points had been defined all over the image. The method used to specify those points was to calculate the gradient of the thresholded image and to select the points with larger gradient values. For feature points to be the most uniformly distributed within the image area the image was divided in rectangles and the number of feature points to be selected in each rectangle had been limited. Subsequently, the excess of points was eliminated aiming at a limited number of points in the whole image. In Fig. 3 an image is shown with several feature points indicated by squares. The centroid of all the selected feature points in the image was calculated and a scattering number was defined as a dispersion around the average, as:

$$x_e = \sum_{i=0}^{N} |\bar{x} - x_i| \qquad\qquad y_e = \sum_{i=0}^{N} |\bar{y} - y_i| \qquad (2)$$

where $(\bar{x}, \bar{y})$ is the centroid of the feature points on the image, (x_i, y_i) is the position of point i, N is the total number of points of the set and (x_e, y_e) is the scattering number of the points in directions *x* and *y*.

The image motion in the *X* direction was estimated by the centroid positions of the feature points and in the Z direction by the scattering numbers indicating the object gets closer to the camera when it increases and farther when it decreases.

5.1 Results Analysis

In order to verify the method effectiveness to calculate image motion tests with real images in laboratory had been carried through with motion along the X and Z directions.

Figure 4 shows captured images of both directions of motion estimated in the tests. Table 1 shows the centroid positions and scattering numbers on several feature points for these movements.

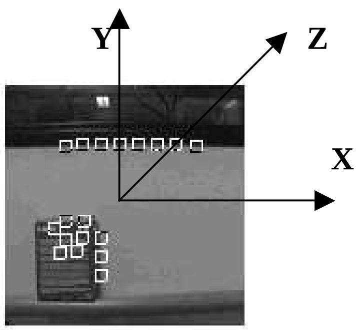

Figure 3. Feature points on an image and the reference coordinate system.

(a)

(b)

Figure 4. Observed movements: a) x direction b) z direction

Table 1. Centroid and scattering numbers of feature points for motion in X and Z directions

MOTION ALONG THE X AXIS					MOTION ALONG THE Z AXIS				
NPI	x_c	y_c	x_e	y_e	NPI	x_c	y_c	x_e	y_e
19	48	62	332	504	19	48	62	332	504
19	51	62	267	511	19	46	57	379	568
19	55	63	256	512	8	40	75	40	23
17	56	62	200	444	8	38	75	30	22
17	60	63	218	448	8	40	75	30	21
16	67	64	188	405	8	42	74	29	21
16	71	63	205	398	8	42	76	17	20
16	75	64	218	403	8	43	75	17	24
16	80	64	235	397	8	44	75	15	21
16	84	63	251	395	8	45	74	15	21

NPI: Number of feature points in the image

(a) (b)

It can be seen in Table 1a that the value of y_c is remained relatively stable. This confirms the fact that the object motion is predominantly in the X direction. It is possible to observe in Table 1b that y_c and x_c have a small variation during the sequence, showing that there is no significant motion in direction X. On the other hand, the values of x_e and y_e have a clear reduction, showing that the object is moving away (distance increases) in relation to the robot.

It can be seen from the tests that the algorithm for motion estimation is capable, in the condition of small linear displacements, to follow the motion of the feature points and to generate data of centroid positions and scattering numbers as predicted. It is important to emphasize that the tests were not conceived to check motion precision but to highlight important parameter interactions/effects by using a factorial design model.

6. System Parameter Optimization

In order to examine the parameters of larger importance in the vision tracking system a study using the factorial design methodology was planned aiming at getting a deepened knowledge of the effect of different parameters on the system effectiveness and moving towards to develop an optimization scheme to speed up image processing. Five parameters were used in this study, showed in Table 2.

Table 2. Variables studied by Factorial Design: (-) inferior limit (+) superior limit

Variable (parameter)	Meaning	(-)	(+)
MAX_FEAT_NUM (1)	Maximum number of feature points	50	200
THRES_PERCENT (2)	Threshold for feature points recording	0,1	5,0
SEARCH_SIZE (3)	Search window width	10	30
CORR_SIZE (4)	Correlation window width	5	15
UNGROUP_SIZE (5)	Minimum distance between feature points	5	20

6.1 Experiments Design and Results Analysis

In order to perform the experiments in a simpler form a fractional factorial design was specified to be in the form of 2^{5-1} with no repetitions, totting up 2^{5-1} = 16 runs. The order of the runs was set to be random.

The sequence of experiments was carried through checking four values of outputs: t_1 (time for processing the first image), t_f (time for processing the following images), e_x (position error in direction *X*) and e_y (position error in direction *Y*).

The time to process an image was measured using internal functions of the programming language especially designed for this purpose, as the *clock()* function, that returns the number of time units spent since the processor started. Recording the initial time unit and subtracting the final time unit yields the span of time needed to process an image.

The position errors had been calculated using as reference the system position operating with large search and correlation windows (values of 50 and 30 respectively), in such a way that the error of this "reference" would be minimum in relation to the errors calculated in the tests.

The results of the analysis of variance can be seen in Table 3, showing variables and interactions that are significant at 99% and 95%. All variables or interactions that are not shown have a significance level smaller than 95%.

Table 3. Results of the Analysis of Variance, showing which input variables and their interactions affect each output variable.

Output Variables	**Input Variables and Interactions**	
	Significant 99%	**Significant 95%**
t_1	1 , 2	12, 13, 23, 34
t_f	3, 4, 34	5, 12, 35, 45
e_x	4, 5, 45	3, 34
e_y	1, 2, 12, 34	3, 4

To show up the importance of the input variables 3 and 4 a sensitivity analysis was carried out to determine optimal values for these variables reducing the time for processing images without losing too much in the code robustness (keeping small errors, e_x and e_y). For this purpose, experiments had been run varying the search window width between 10 and 20 pixels and the correlation window width between 5 and 15 pixels, according to the ranges defined in the previous factorial design runs. The values assigned to the other variables are shown in Table 4. Figures 5, 6 and 7 show these results.

Table 4. Fixed values used on the sensitivity analysis of variables 3 and 4.

Input Variables	**Fixed Value**
VAR 1 = MAX_FEAT_NUM	100
VAR 2 = THRES_PERCENT	2
VAR 5 = UNGROUP_SIZE	5

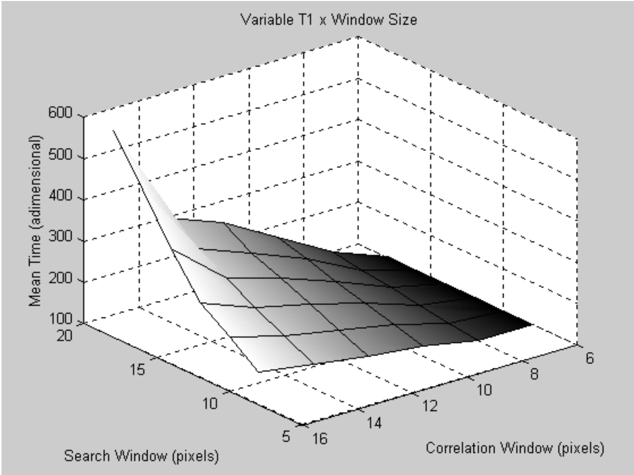

Figure 5. Graph showing the effect of the width of the search and correlation windows on variable t_1.

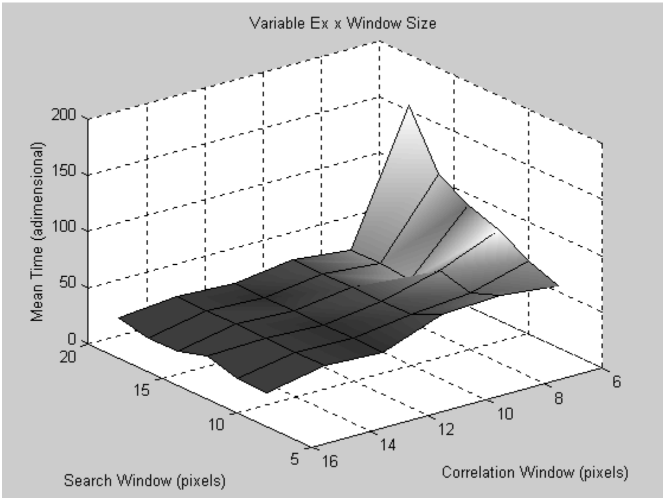

Figure 6. Graph showing the effect of the width of the search and correlation windows on variable e_x.

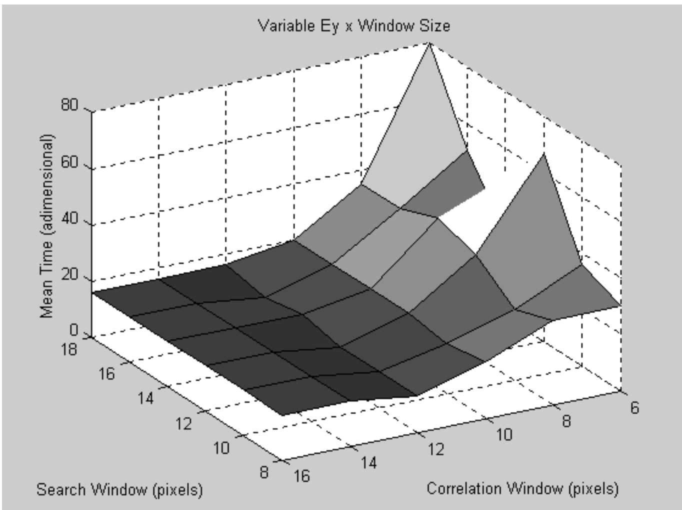

Figure 7. Graph showing the effect of the width of the search and correlation windows on variable e_y.

From the graphs above it is easy to choose ideal values for the variables 3 and 4 as 15 and 9 respectively. A strategy to calculate automatically an optimal value for the width of the windows is straightforward.

7. Conclusions

The article presented an optimization scheme based on factorial design analysis techniques to show the effects of important variables in an image-based tracking system for a mobile robot using SSD correlation and a gradient-based optical flow calculation. Results showed the effect of all possible combinations of input variables on the system performance and a sensitivity analysis of the most important variables, namely the correlation and search window sizes for tracking feature points in the image. The routines involved can be easily automated. The method presented can be very useful in a non-structured environment where choices of usually fixed variables may strongly affect the system performance.

8. References

Anandan, P & Bergen, J. R. (1992). *Hierarchical model-based motion estimation*, Proceedings of the European Conference on Computer Vision, Santa Margherita Ligure, Italy.

Coleman, D. F. & Montgomery, D. C. (1993). *A systematic approach to planning for a designed industrial experiment*, Technometrics, 35(1), pp. 1-27.

Corke, P. & Hutchinson, S. A. (2000). *Real-Time Vision, Tracking and Control*, Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco-CLA, USA.

Espiau, B. (1993). *Effect of Camera Calibration Errors on Visual Servoing in Robotics*, 3rd International Symposium on Experimental Robotics, Kyoto, Japan.

ISO. (1995). *Guide to the Expression of Uncertainty in Measurement (GUM)*, Geneva: International Organization for Standization (ISSO).

Kara, R., Wira, P. & Kihl, H. (2000). *Robot Vision Tracking with with a Hierarchical CMAC Controller*, Fourth International Conference on knowledge-Based Intelligent Engineering Systems & Allied Technologies, Brighton, UK, Sept. 2000.

Kass, M., Witkins, A. and Terzopoulos, D. (1998). *SNAKES: Active Contour Models*, IJCV, Vol. 1, N. 4.

Klette, R., Schlüns, K. & Koschan, A. (1998). *Computer Vision: Three-Dimensional Data from Images,* Springer-Verlag, Singapore.

Montgomery, D. C. (1991). *Design and analysis of experiments,* New York, Wiley.

Plakas, K. & Trucco, E. (2000). *Developing a real-time, robust, video tracker,* OCEANS 2000 MTS/IEEE Conference and Exhibitions, Vol. 2, pp. 1345-1352.

Plakas, K., Trucco, E. & Fusiello, A. (1998). *Uncalibrated Vision for 3D Underwater Applications*, Proceedings IEEE Oceans´98, pp. 272-276, Nice, France.

Santos-Victor, J. and Sandini, G. (1997). *Visual Behaviours for Docking*, Computer Vision and Image Understanding.

Santos-Victor, J. and Sentiero, J. (1993). *A 3D Vision System for Underwater Vehicles: Na Extended Kalman-Bucy Filtering Approach,* Acoustic Signal Processing for Ocean Exploration, 567-580.

Shi, J. & Tomasi, C. (1994). *Good features to track*, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600.

Spindler, F. & Bouthemy, P. (1998). *Real-time estimation of dominant motion in underwater video images for dynamic positioning*, Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Louvain, Belgium, pp. 1063-1068.

Tomasi C. & Kanade T. (1991). *Detection and tracking of point feature,* Technical Report CMU-CS-91-132, Carnegui-Melon University, Pittsburg, PA.

Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall, New Jersey, USA.