POSITION ERROR ANALYSIS OF A 3D TRACKING SYSTEM

Aranda, J. (†), Gibert, K. (*), Climent, J(†). and Grau, A(†).
(†) Dep. of Automatic Control & Computer Engineering.
(*) Dep. of Statistics and Operations Research
Universitat Politècnica de Catalunya
Pau Gargallo,5. 08028. Barcelona. SPAIN.
E-mail: aranda@esaii.upc.es

Commission V, Working Group IC V/III

KEY WORDS: Tracking, real time, image processing hardware, error modelling, error propagation.

ABSTRACT

A new 3D tracking method is presented which makes use of a specific image processing hardware developed in our laboratory. This image processor performs at video rate an image transformation consisting on the computation of the distance from each pixel in the image to the contour pixels around it (if present). With the aim of minimize the cost of this processing hardware only eight distance values in a 15x15 pixels window are obtained for every pixel, corresponding to that of the eight main directions (N, NE, E, SE, S, SW, W, NW). This seems to be enough for many tracking applications as have been proved. This vector of distances identifies singular points in the contour image and it is used in their recognition process (applied both in stereo matching process and also in sequence matching process). Two main errors disturb the output of the tracking system (tridimensional position of these singular points along the time): image resolution and localization error of contour pixels. Modeling and propagation of these two main errors, both in image transformation process and also in recognition/position estimation processes is fully explained.

1. INTRODUCTION

Tracking systems based on computer vision are sensible to those errors coming from image formation and processing. Accuracy of the position measurements of the tracked target depend on these errors. However, usually little effort is made in order to evaluate how these errors disturb the output of the system. In this paper we tackle this problem for the particular case of an implemented tracking system based on a specific image processing hardware.

Efficiency of tracking systems can be measured by two parameters (usually opposed): reliability of recognition process and its execution time. The last one determines the system sampling period and obviously it has to be as short as possible. All tracking methods include a compromise solution to balance this trade-off, usually by limiting the set of targets that can be recognized and the circumstances in which they can be tracked. In order to maximize reliability without penalize the sampling period, huge and expensive computational resources are required to perform real time tracking. This circumstance limits massive application of such systems in industry [Amat,93].

In our case, a polar representation of image objects contours has been chosen for recognition [Gonzalez,87]. This polar descriptor permits to reduce contour representation from two dimensions to one. It also permits an easy way for size, position and orientation normalization of the object contour. Contour rotation appears as a translation in the transformed space, so the transformed description is easier to track in front of object rotations. For these reasons, variations on polar transform have been used by a lot of authors as a previous step in pattern recognition [Jeng,91] [Sekita,92] [Friedland,92].

However, in the presented tracking system the polar transform is only applied locally to those singular regions (*local features*) present in the object contour [Amat,92]. The polar transformation has been reduced and optimized in order to implement it with a low cost hardware. In this way the transformation has been limited to a 15x15 pixels region, from which they are selected only 8 radii in the 8 main directions (N, NE, E, SE, S, SW, W, NW). These radii represent the distance from the central pixel of the analyzed region to the first contour pixel found in the corresponding direction (figure 1).

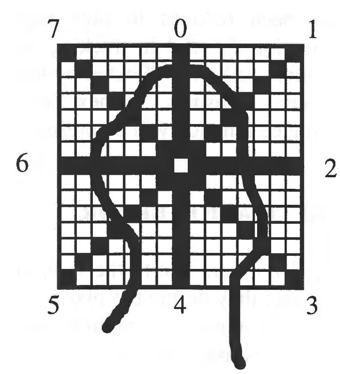


Figure 1(a). Distribution of radii in the transformation window

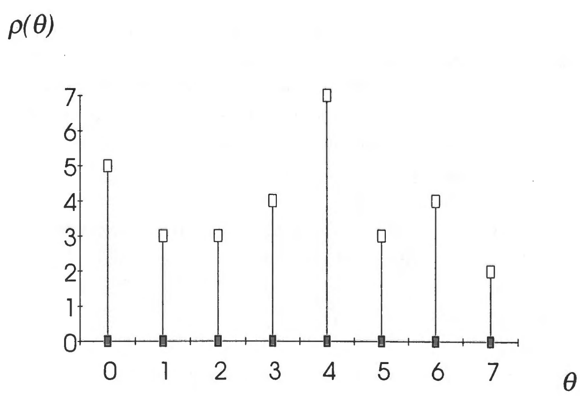


Figure 1(b). Resulting vector descriptor ρ(θ).

If more than one contour pixel appears in any direction then the corresponding radius takes as distance the nearest contour pixel to the central pixel. If no contour pixel is found in one particular direction a value equal to 7 is assigned to the corresponding radius (as radius 4 in figure 1(b)).

With the aim of providing a low cost real time tracking system for industrial applications a specific image processing board for an industrial PC has been implemented [Aranda,96]. This specific processor supplies to the host the polar descriptions of the local features included into the *tracking windows*. Two boards (one for each camera) are required in order to perform stereo tracking of the selected local features.

The host uses this set of polar descriptors to recognize and to locate the tracked targets while tracking them in an image sequence and also to perform stereo matching. Recognition is performed by looking for the minimum of the next distance function:

$$F_i = \sum_{k=0}^{7} (\rho_M(k) - \rho_i(k))^2$$

Where $\rho_M(\theta)$ is the polar description of the tracked local feature which acts as a model, and $\rho_i(\theta)$ is the polar description associated to every i local feature in the tracking window.

Once the target included in a tracking window is recognized the host relocates its corresponding tracking window to a new searching position in the next image frame. This process is repeated for every target in every tracking window for both left and right images. Image processing performed by the specific processors and target recognition computed by the host by software, are overlapped in time obtaining a total computation time of 20 ms (*video rate*).

As polar transform has been reduced to only eight radial samples, the distance function F_i used in tracking and stereo matching of the local features, is highly affected by localization error included in the radial measures. In next sections the expected error on this distance function will be presented.

2. DISCRETIZATION ERROR

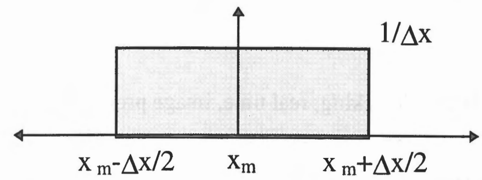
Let be Δx *i* Δy the *resolution errors* due to the sampling in the two coordinates of the image; they define the pixel dimensions. Squared pixels are achieved by adjusting sampling period on the image processing boards, so is assumed that $\Delta x = \Delta y$. In the following only dimension X is considered. Results are the same for dimension Y.

The exact position of a certain image point (*i.e.* a contour pixel) along dimension X, namely x, is measured by a discrete value x_m, which may be different from x, since a discretization error is produced. In fact, given x_m (measured position) for a certain image point, it is known that the real position (x) satisfies: $x \in [x_m - \Delta x/2, x_m + \Delta x/2]$.

The discretization error (εx) is defined as the difference between the real and the measured position: $\varepsilon x = x - x_m$ and its value is contained in the interval $\varepsilon x \in [-\Delta x/2, +\Delta x/2]$. Therefore, εx is upper-bounded by $\varepsilon x_{max} = \Delta x/2$.

However, εx_{max} is not a good measure for the localization error made in the image acquisition, since it is very infrequent to make so big errors. It is preferable to characterize the discretization error by means of its *expectation* (or mean value) and its *standard deviation*.

Since there is not any *a priori* information for the distribution of the real position (x) in the interval $[x_m - \Delta x/2, x_m + \Delta x/2]$, in this paper x will be considered as a random variable with a *uniform distribution* between the limits of the pixel (see figure 2).



$$x \sim U\,[x_m - \Delta x/2, x_m + \Delta x/2]$$

Figure 2. Uniform distribution for the real position x

Using this uniform distribution, the probability that x is situated in an interval of longitude dx around xm is $dx/\Delta x$, if $dx \in [x_m - \Delta x/2, x_m + \Delta x/2]$, and 0 otherwise. The expectation of the discretization error $\varepsilon x = x - x_m$ is then calculated as:

$$\mu = E[\varepsilon x] = \frac{1}{\Delta x}\int_{x_m-\Delta x/2}^{x_m+\Delta x/2} (x - x_m).dx = \frac{1}{\Delta x}\int_{-\Delta x/2}^{+\Delta x/2} (x).dx = 0$$

The variance of εx is given by:

$$\sigma^2[\varepsilon x] = E[\varepsilon x^2] - E[\varepsilon x]^2$$

$$E[\varepsilon x^2] = \frac{1}{\Delta x}\int_{x_m-\Delta x/2}^{x_m+\Delta x/2} (x - x_m)^2.dx = \frac{1}{\Delta x}\int_{-\Delta x/2}^{+\Delta x/2} (x)^2.dx = \frac{\Delta x^2}{12}$$

The resolution error Δx is *1 pixel.* So the standard deviation of the discretization error is, finally:

$\sigma(\varepsilon x) = \sqrt{\sigma^2(\varepsilon x)} = \Delta x/\sqrt{12} = 0{,}288$ *pixels.*

It can be seen that this value is much smaller as the maximum which was previously calculated (about the half of it):

$\varepsilon x_{max} = \Delta x/2 = 0{,}5$ *pixels.*

3. ERROR ON POLAR TRANSFORM

In this section the effect of the discretization error on the longitude of the radii of the polar transformation of the image is analyzed. Two possibilities will be distinguished: when the radii are on the image coordinate axes directions (horizontal and vertical radii) and diagonal cases. Again, horizontal and vertical radii need similar treatment, since squared pixels are considered ($\Delta x = \Delta y$); only the horizontal case is detailed here.

For **horizontal radii**, their longitude (r) is given by the relative position of the contour pixel with respect to the central pixel of the image transformation window. Then $r = x - xc$, being x the exact position of the contour on dimension X and xc the exact position of the transformation window center.

However, it is only possible to measure a discrete longitude of the radii, which is given by the expression $r_m = x_m - xc$, where x_m is the discrete position of the image point on the X axis.

The discretization induces also an error r_m, namely εr, which depends only on the error of x_m, since the position of the window center (xc) is known without uncertainty:

$\varepsilon r = r - r_m = (x - xc) - (x_m - xc) = x - x_m = \varepsilon x$

$|\varepsilon r_{max}| = |\varepsilon x_{max}| = \Delta x/2 = 1/2$ *pixel*

$[\varepsilon r] = E[\varepsilon x] = 0$

$\sigma(\varepsilon r) = \sqrt{\sigma^2(\varepsilon r)} = 1/\sqrt{12} = 0{,}288$ *pixels.*

For **diagonal radii**, we proceed in the same way. The longitude of those radii is defined as the distance between the contour pixel $(x1,y1)$ and the central pixel of the transformation window (xc,yc).

The real longitude of any diagonal radius is given by $r = \sqrt{(x-xc)^2 + (y-yc)^2}$, with (x,y) exact position of the contour pixel. The measured one is $r_m = \sqrt{(x_m - xc)^2 + (y_m - yc)^2}$, with $(x,y)_m$ discrete position of the contour. Then, the error associated to the longitude of diagonal radii is $\varepsilon r = r - r_m$, and using this:

$$E[r] = \frac{1}{\Delta x.\Delta y}\int_{x_m-\Delta x/2}^{x_m+\Delta x/2} dx \int_{y_m-\Delta y/2}^{y_m+\Delta y/2} r(x,y).dy = r_m \qquad (1)$$

it can be shown that the expectation of εr is null:

$E[\varepsilon r] = E[r - r_m] = E[r] - r_m = r_m - r_m = 0$

The variance of εr is:

$\sigma^2(\varepsilon r) = \sigma^2(r) + \sigma^2(r_m) + 2.\,cov(r, r_m) = \sigma^2(r)$

To calculate $\sigma^2(r)$ expression *(1)* is used:

$$\sigma^2(r) = E[r^2] - E[r]^2 = E[r^2] - r_m^2 \qquad (2)$$

$E[r^2]$ can be calculated as:

$$E[r^2] = E[(x-xc)^2] + E[(y-yc)^2] \qquad (3)$$

Developing the sum of squares and using that E[εx]= 0:

$E[(x-xc)^2] = E[(x - x_m + x_m - xc)^2] = \sigma^2(\varepsilon x) + (x_m - xc)^2$

Similarly, for y: $E[(y-yc)^2] = \sigma^2(\varepsilon y) + (y_m - yc)^2$, and using these results, expression *(3)* is reduced to:

$E[r^2] = \sigma^2(\varepsilon x) + \sigma^2(\varepsilon y) + r_m^2$

Going back, expression *(2)* is:

$\sigma^2(\varepsilon r) = \sigma^2(\varepsilon x) + \sigma^2(\varepsilon y) + r_m^2 - r_m^2 = \sigma^2(\varepsilon x) + \sigma^2(\varepsilon y)$

Since squared pixels are considered, it is known that $\sigma^2(\varepsilon x) = \sigma^2(\varepsilon y)$. Then:

$\sigma^2(\varepsilon r) = 2.1/12 = 0{,}16$ *pixels.*

The standard deviation of the error on the longitude of diagonal radii is:

$\sigma(\varepsilon r) = \sqrt{\sigma^2(\varepsilon r)} = 1/\sqrt{6} = 0{,}408$ *pixels.*

It can be seen that the value of the standard deviation is about the half of the maximum expected error:

$|\varepsilon r_{max}| = \sqrt{(|\varepsilon x_{max}|^2 + |\varepsilon y_{max}|^2)} = \sqrt{((1/2)^2 + (1/2)^2)} = 0{,}707$ *pixels*

There results agree with the intuitive perception (given by geometry) that the error on diagonal radii due to the discretization has a factor √2 with respect to that produced in the horizontal (or vertical) radii (which corresponds to the relationship between the diagonal and the side of a 1 pixel square).

4. ERROR ON RADII COMPARISON

Two possibilities have to be considered: on the one hand, horizontal and vertical radii; on the other hand, diagonal radii.

Differences between two **horizontal** (or vertical, taking again advantage of the squared pixels) radii are given by the equations:

$f = r1 - r2 = (x1 - xc) - (x2 - xc) = x1 - x2$

$f_m = r1_m - r2_m = (x1_m - xc) - (x2_m - xc) = x1_m - x2_m$

where $x1$ and $x2$ are the coordinates of the contour pixels inside the image, which determine the longitude of $r1$ and $r2$ respectively; $x1_m$ and $x2_m$ are their discrete values and xc is the x coordinate of the center of the polar transformation window.

The error on f_m due to the discretization, namely εf, depends on the errors of $x1_m$ and $x2_m$. Remembering that $\varepsilon x = x - x_m$.

$\varepsilon f = f - f_m = (x1 - x2) - (x1_m - x2_m) = (x1 - x1_m) - (x2 - x2_m) = \varepsilon x1 - \varepsilon x2$

Since εf is the difference of two random variables of uniform distribution, its probability function follows a triangular distribution (figure 3):

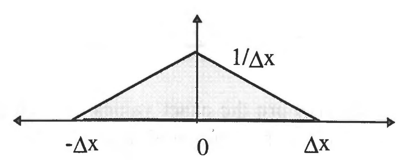


Figure 3. Triangular distribution for the error on radii comparison

The maximum value is given by the expression:

$|\varepsilon f_{max}| = |\varepsilon x1_{max}| + |\varepsilon x2_{max}| = \Delta x/2 + \Delta x/2 = \Delta x = 1$ *pixel*

Again, this is an infrequent value for the error, especially considering that this error follows a triangular distribution. The expectation may be calculated on the basis of the errors associated to $x1_m$ and $x2_m$ as:

$E[\varepsilon f] = E[\varepsilon x1 - \varepsilon x2] = 0$

$$E[\varepsilon x1 - \varepsilon x2] = \frac{1}{\Delta x.\Delta x}\int_{-\Delta x/2}^{+\Delta x/2} dx1 \int_{-\Delta x/2}^{+\Delta x/2} (x1 - x2).dx2 = 0$$

and its variance can be evaluated using the independence of $x1$ and $x2$ [Sanchez, 89]:

$\sigma^2(\varepsilon f) = \sigma^2(\varepsilon x1) + \sigma^2(\varepsilon x2) = 1/12 + 1/12 = 0{,}16$ *pixels*^{2}

So, the standard deviation for the error associated to the difference of horizontal (or vertical) radii is:

$\sigma(\varepsilon f) = \sqrt{\sigma^2(\varepsilon f)} = 1/\sqrt{6} = 0{,}408$ *pixels.*

For the case of **diagonal radii**:

$r = \sqrt{(x-xc)^2 + (y-yc)^2}$

Where (x,y) is the exact position of the pixel contour.

$f = r1 - r2$ and $f_m = r1_m - r2_m$

To calculate εf, previous $\varepsilon r1$, $\varepsilon r2$ are used:

$\varepsilon f = f - f_m = (r1 - r2) - (r1_m - r2_m) = (r1 - r1_m) - (r2 - r2_m) = \varepsilon r1 - \varepsilon r2$

Its maximum value is: $|\varepsilon f_{max}| = |\varepsilon r1_{max}| + |\varepsilon r2_{max}| = 1{,}41$ *pixels*

The expected value for εf is, as usual:

$E[\varepsilon f] = E[\varepsilon r1 - \varepsilon r2] = E[\varepsilon r1] - E[\varepsilon r2] = 0-0 = 0$

and the variance for the error associated to differences between diagonal radii is:

$\sigma^2(\varepsilon f)=\sigma^2(\varepsilon r1-\varepsilon r2)=2.\sigma^2(\varepsilon r)=4.\sigma^2(\varepsilon x)=4.1/12=0,33$ *pixels*

so the standard deviation is:

$\sigma(\varepsilon f)= \sqrt{\sigma^2(\varepsilon f)} = 0,577$ *pixels*

5. ERROR ON LOCAL FEATURES RECOGNITION

In this section the effect of the discretization error on the recognition of the local features of the image is quantified. The recognition process is used by the proposed system in order to locate the tracked targets while tracking them in an image sequence and also to perform stereo matching.

The distance function used to compare two polar transforms has an exact value (F) and a calculated value from given data (F_m):

$$F = \sum_{\theta=0}^{7}(r1(\theta) - r2(\theta))^2 = \sum_{\theta=0}^{7} f^2(\theta)$$

$$F_m = \sum_{\theta=0}^{7}(r1_m(\theta) - r2_m(\theta))^2 = \sum_{\theta=0}^{7} f_m^2(\theta) \qquad (4)$$

Where $r1(\theta)$ and $r2(\theta)$ are the exact values of each radii of the polar transformations to be compared, and $r1_m(\theta)$, $r2_m(\theta)$ are the discrete values associated to them.

Then, $\varepsilon F = F - F_m$. This error depends on the accumulated error from the radii comparison due to the discretization error.

For each element of the summation of expression *(4)*, there can be defined another error term:

$\varepsilon R = f^2 - f_m^2 = (f_m + \varepsilon f)^2 - f_m^2 = 2.f_m.\varepsilon f\ (+\ \varepsilon d^2)$

From this expression, the last term can be disregarded. Then, for **horizontals** and **verticals** radii ($\theta=0,2,4,6$):

$|\varepsilon R_{max}| = 2 . f_m . \max(\varepsilon f) = 2 . f_m .1 = 2.f_m$ *pixels*^{2}

$E[\varepsilon R] = 2 . f_m .E[\varepsilon f] = 0$

$\sigma(\varepsilon R)= \sqrt{(2.f_m)^2.\sigma^2(\varepsilon f)}= \sqrt{(2/3)}.f_m= 0,816.f_m$ *pixels*^{2}.

for **diagonal radii** ($\theta=1,3,5,7$):

$|\varepsilon R_{max}|=2 . f_m . \max(\varepsilon f) = 2 . f_m . \sqrt{2} = 2\sqrt{2}.f_m$ *pixels*^{2}

$E[\varepsilon R] = 2 . f_m .E[\varepsilon f] = 0$

$\sigma(\varepsilon R) =\sqrt{(2. f_m)^2.\sigma^2(\varepsilon f)}=\sqrt{(4/3)}. f_m = 1,154.f_m$ *pixels*^{2}

For any radius direction, the standard εR is proportional to f_m. Therefore, the relative error $\varepsilon R/f_m^2$ decreases when f_m^2 increases. It has to be considered f_m is only defined between values 0 and 7.

The total error of function F is:

$$\varepsilon F = F - F_m = \sum_{\theta=0}^{7} f^2(\theta) - \sum_{\theta=0}^{7} f_m^2(\theta) = \sum_{\theta=0}^{7}\left[f^2(\theta) - f_m^2(\theta)\right] = \sum_{\theta=0}^{7}\varepsilon R(\theta)$$

$$\left|\varepsilon F_{max}\right| = \sum_{\theta=0}^{7}\left|\varepsilon R_{max}(\theta)\right| = \sum_{\theta=0}^{7} w_\theta . f_m(\theta) \qquad w_\theta = \begin{cases} 2 & \text{if } \theta \text{ even} \\ 2.\sqrt{2} & \text{if } \theta \text{ odd}\end{cases}$$

$$E\left[\varepsilon F\right] = \sum_{\theta=0}^{7} E\left[\varepsilon R_\theta\right] = 0$$

$$\sigma^2\left[\varepsilon F\right] = \sum_{\theta=0}^{7}\sigma^2\left[\varepsilon R_\theta\right] = \sum_{\theta=0}^{7} w_\theta . f_m^2(\theta) \qquad w_\theta = \begin{cases} 2/3 & \text{if } \theta \text{ even} \\ 4/3 & \text{if } \theta \text{ odd}\end{cases}$$

$$\sigma\left[\varepsilon F\right] = \sqrt{\sum_{\theta=0}^{7}\sigma^2\left[\varepsilon R_\theta\right]} = \sqrt{\sum_{\theta=0}^{7} w_\theta . f_m^2(\theta)} \qquad w_\theta = \begin{cases} 2/3 & \text{if } \theta \text{ even} \\ 4/3 & \text{if } \theta \text{ odd}\end{cases}$$

From this results, it can be observed than expected value for error in local feature recognition due to discretization of image coordinates is equal to 0. Standard deviation of this error grows up in the same proportion as the square root of the measured distance : $\sigma(\varepsilon F) \cong \sqrt{F_m}$. In fact, as a first approach, the standard deviation of this error can be upper-bounded by the expression: $\sigma(\varepsilon F) < 1,15.\sqrt{F_m}$.

That is, the greater is the real similitude between two polar transformations (small F_m), the smaller is the error made in the measurement of this similitude.

6. ERROR ON 3D DISTANCE ESTIMATION

In this section the effect of image discretization on measuring the disparity between corresponding local features from both left and right cameras is analyzed. Disparity determines the z coordinate of the tracked local features.

The 3D distance (or depth) is measured from disparity by means of the following expression:

$Z = \lambda.B / (xe - xd) \quad Z_m = \lambda.B / (xe_m - xd_m) \quad \varepsilon Z = Z - Z_m$

where λ is the focal distance (equal to both cameras) and B is the camera separation (*baseline*). Once cameras are calibrated, these quantities act as constants. Variables xe and xd are the x coordinates of a certain local feature of the scene in the left and right image respectively.

The *disparity (g)* is determined by the difference between these values: $g = xe - xd$ so $Z = \lambda.B / g$ (with g>0 since xe>xd).

In fact, what is known is : $g_m = xe_m - xd_m$ and $Z_m = \lambda.B / g_m$.

In this case, g_m can be null (equal to 0) owing to the discretization error. If so, the distance cannot be determined. Singular cases will not be treated here.

The error associated to the disparity is already known, since it is the same as the error on the difference of horizontal radii calculated in section 4:

$\varepsilon g = g - g_m = (xe - xd) - (xe_m - xd_m) = \varepsilon xe - \varepsilon xd$

$|\varepsilon g_{max}| = |\varepsilon xe_{max}| + |\varepsilon xd_{max}| = \Delta x/2 + \Delta x/2 = \Delta x = 1$ *pixel*

$E[\varepsilon g] = E[\varepsilon xe - \varepsilon xd] = 0$ *pixels*

$\sigma(\varepsilon g) = 1/\sqrt{6} = 0,408$ *pixels.*

using that, the error made when calculating Z is:

$\varepsilon Z = Z - Z_m = (\lambda.B / g) - (\lambda.B / g_m)$

$$\varepsilon_Z = \frac{\lambda.B}{g_m + \varepsilon g} - \frac{\lambda.B}{g_m} = -\frac{\lambda.B.\varepsilon g}{(g_m + \varepsilon g).g_m} = -\frac{\lambda.B}{g}.\frac{\varepsilon g}{g_m} = -Z.\frac{\varepsilon g}{g_m}$$

which depends on the Z itself. So, it is more interesting the relative error on Z, which corresponds to the following expression: $\varepsilon Z / Z = -\varepsilon g / g_m$

It is possible to quantify this relative error, using the measured disparity (g_m) and its associated error (εg):

$|\varepsilon Z/Z_{max}| = |\varepsilon g_{max}| / g_m = 1/g_m$

$E[\varepsilon Z/Z] = -(1/g_m).E[\varepsilon g] = 0$

$\sigma(\varepsilon Z/Z) = (1/g_m).\sigma(\varepsilon g) = 0,408/g_m$

Figure 4 shows the evolution of the standard deviation of this relative error depending on the disparity. It can be seen that disparities less than four pixels give a standard deviation of the relative error on the measure greater than the 10%. The user of our system can limit the maximum distance (minimum disparity) to which objects can move. A reasonable value for this minimum distance seems to be four pixels, considering the results of the error analysis.

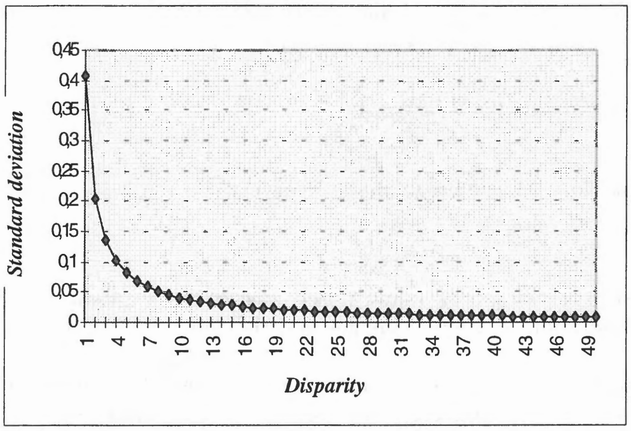


Figure 4. Evolution of standard deviation of relative error on Z, $\sigma(\varepsilon Z/Z)$, as a function of disparity.

An easy way to increase the disparity (decreasing the effect of the discretization error on the distance measure), is to increase the distance between cameras (*B*). However, this involves other problems; among them, a complexity increase in the data association process (stereo matching), major difficulty for camera calibration and increase of hidden local features (without corresponding point) in some left or right projection.

7. CONCLUSIONS

With the aim of providing a low cost real time tracking system for industrial applications a specific image processing board for an industrial PC has been implemented. This specific processor supplies to the host with the polar descriptions of the local features to be tracked. This information is used then to compute a recognition and stereo matching algorithm so that 3D tracking and data acquisition processes are overlapped in time.

Recognition and stereo matching processes are disturbed by the fact that data provided by image processor includes localization error of contour pixels. In this paper it has been demonstrated that the expected error on this distance function is equal to zero and its standard deviation is upper-bounded by the squared root of its magnitude ($\sigma(\varepsilon F) < 1{,}15.\sqrt{F_m}$).

Using the error that appears when measuring the position (*x,y*) in the image, the errors associated to all the measures realized in the proposed tracking method have been calculated. This has been done by modeling the localization error of a given pixel by means of an uniform distribution function and propagating this position error. Results are summarized in table 1.

These results can be generalized to all those errors in contour position with a mean value of zero. If only the discretization error is considered, the following values can be assigned to the position errors:

$\varepsilon x_{max}=1/2\ pixel \quad \sigma(\varepsilon x)=0{,}408\sqrt{\varepsilon x_{max}}=0{,}2884\ pixels$

From this data, all other values can be calculated as shown in the table.

This work is the base for a more accurate analysis, which has also be done, taking into account the localization error produced in the contour extraction module.

The demonstrated accuracy of the presented system permit their generalized application in a lot of different industrial applications including robot control feedback and teleoperation.

	ε_{max}	$E[\varepsilon]$	$\sigma(\varepsilon)$
εx	εx_{max}	0	$\sigma(\varepsilon x)$
εr	εx_{max}	0	$\sigma(\varepsilon x)$
$\varepsilon r(d.)$	$\sqrt{2}.\varepsilon x_{max}$	0	$\sqrt{2}.\sigma(\varepsilon x)$
εf	$2.\varepsilon x_{max}$	0	$\sqrt{2}.\sigma(\varepsilon x)$
$\varepsilon f(d.)$	$2.\sqrt{2}.\varepsilon x_{max}$	0	$2.\sigma(\varepsilon x)$
εR	$4.f_m.\varepsilon x_{max}$	0	$2.\sqrt{2}.f_m.\sigma(\varepsilon x)$
$\varepsilon R(d.)$	$4.\sqrt{2}.f_m.\varepsilon x_{max}$	0	$4..f_m.\sigma(\varepsilon x)$
εF	$4.\varepsilon x_{max}[\sum f_m+\sqrt{2}.\sum f_m]$	0	$2.\sqrt{2}.\sigma(\varepsilon x).[\sum f_m+\sqrt{2}.\sum f_m]$
εg	$2.\varepsilon x_{max}$	0	$\sqrt{2}.\sigma(\varepsilon x)$
$\varepsilon Z/Z$	$(2/g_m).\varepsilon x_{max}$	0	$(\sqrt{2}/g_m).\sigma(\varepsilon x)$

Table 1. Summarized results of localization error incidence.

8. REFERENCES

Amat, J., Aranda, J., A. Casals, 1992. A Tracking System for Robot Servopositioning in moving parts operations. ISIR'92. Barcelona. Catalunya.

Amat, J., Aranda, J., A. Casals, 1993. Tracking capabilities in Computer Vision for Industrial Applications. Optical Engineering (SPIE). Vol. 32, num. 11, pp. 2796-2804.

Aranda, J., 1996. Image Processing Card for Real Time Target Tracking. DCIS. Sitges, Catalonia.

Friedland, N.S. and Rosenfeld, A., 1992. Compact object recognition using energy-function-based optimization. IEEE Trans. on Pattern Analysis and Mach. Int., vol. 14, n. 7, pp. 770-777.

Gonzalez, R. C., 1987. Digital Image Processing. Addison-Wesley.

Jeng, S. and Tsai, W., 1991. Scale and orientation invariant generalized Hough Transform - A new approach. Pattern Recognition, vol. 21, n. 11, pp. 1037-1051.

Sanchez, C. 1989. Error Analysis (*in spanish*). Eudema S.A.

Sekita, I., Kurita, T. and Otsu, N., 1992. Complex autoregressive model for shape recognition. IEEE Trans. on Pattern Analysis and Mach. Int., vol.14, n. 4, pp. 489-496.