# Planar 3D Object Detection by Using the Generalized Hough Transform *

N. Guil, J. R. Cózar and E.L. Zapata
Dept. of Computer Architecture. University of Málaga.

## Abstract

*In this work we propose a new method to detect arbitrary planar shapes from a previous template and calculate the parameters that define the transformations between the new image and the template. The image contains a perspective projection of the template subjected to two angles transformation, called tilt and pan, a displacement a rotation and a scaling. The method uncouples parameter calculation to improve computational requirements by comparing invariant information from the template and the image. The Generalized Hough Transform is used to compare this information and to vote into a parameter space.*

## 1 Introduction

The detection of planar objects is an interesting problem to be solved in industrial environments where these kinds of shapes are very common. If the object is planar, the modification of its position in the 3-D space produces a deformation on its shape (perspective transformation). However, all points remain visible and a single camera could be enough to detect the object.

Some authors, [1], try to represent projected shapes in a canonical form that facilitates their comparison.

Lo et al., [2], applied the Generalized Hough Transform to planar shape detection. A new perspective reference table, containing information from all viewing directions and positions, was built. This table is superimposed on each edge point for each inverse perspective transformation of the image shape. The problem with this method is its high computational complexity.

Pei and Liou, [3], use scaled-orthographic projection to find the movement of a planar patch in 3-D space. However, their approach is based on having previously solved the recognition problem . Futhermore, the projection model has several restrictions that limit its range of application [4].

*This work was supported in part by the Ministry of Education and Science (CICYT) of Spain under contract TIC96-1125-C03 and EC proyect BRPR-CT96-01070.

Pizlo et al., [5], introduce a new method to recognize planar shapes based on contour properties deduced from the effect of the tilt on an inverse perspective transformation of angles and lengths.

Aloimonos, [6] relaxes perspective projection conditions by introducing two new approaches that produce a negligible error in some recognition tasks.

In this paper we undertake a new approach to planar object detection in order to accelerate the detection process. Our method is based on the Generalized Hough Transform, but it reduces the computational complexity by uncoupling parameter detection and avoiding a high number of calculations during the inverse projection of the image shape.

The paper is organized as follows. The next section introduces the mathematical expressions involved in perspective transformation. New expressions for the modification of tangent vectors when the previous transformation is applied are also shown. In section 3 invariant transformations are introduced. Section 4 presents the new method for planar shape detection. Finally, in section 5, several real experiments have been carried out in order to test the algorithm's behavior.

## 2 Perspective transform

### 2.1 Point transformation

The necessary transformations to project a planar object into the image plane are shown in figure 1, where $f$ is the focal distance of the camera lens, $d$ is the distance between the focal point and the intersection of the object plane, with the $z-axis$ and $\vec{n}$ being the normal vector to the object plane. The projection of this vector onto $x-y$ and $x-z$ planes allows us to determine the pan, $\tau$, and the tilt, $\delta$, angles, respectively. Thus, $\tau$ is the angle between the projection of $\vec{n}$ onto the $x-y$ and the $x-axis$. On the other hand, $\delta$ is the angle between the projection of $\vec{n}$ onto $x-z$ and the $z-axis$.

The relationship between an object point $(x_i, y_i, z_i)$ and its corresponding image point $(u_i, v_i, f)$ can be expressed using the perspective transformation:

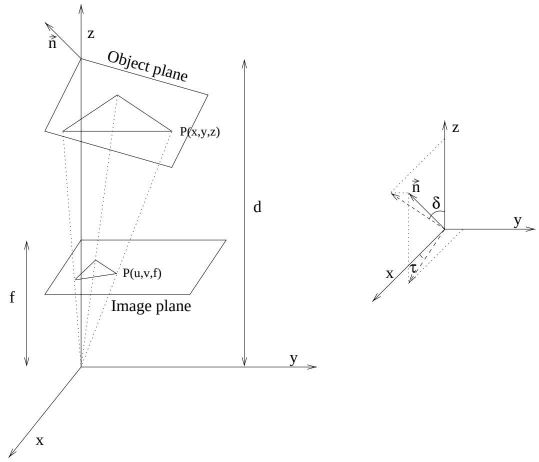



**Figure 1. Relations between a planar object and its projection into the image plane (the object plane origin lies on the $z - axis$ of the camera).**

$$
\begin{aligned}
u_i &= f \cdot \frac{x_i \cdot cos\delta \cdot cos\tau - y_i \cdot sin\tau}{x_i \cdot sin\delta + d} \\
v_i &= f \cdot \frac{x_i \cdot cos\delta \cdot sin\tau + y_i \cdot cos\tau}{x_i \cdot sin\delta + d}
\end{aligned} \tag{1}
$$

On the other hand, the expression for the inverse perspective transformation yields:

$$
\begin{aligned}
x_i &= d \cdot \frac{(u_i \cdot cos\tau + v_i \cdot sin\tau)/cos\delta}{f - u_i tan\delta cos\tau - v_i tan\delta sin\tau} \\
y_i &= d \cdot \frac{-u_i \cdot sin\tau + v_i \cdot cos\tau}{f - u_i tan\delta cos\tau - v_i tan\delta sin\tau}
\end{aligned} \tag{2}
$$

## 2.2 Tangent angle transformation

A function, $y = g(x)$, can be defined to describe the shape in the object plane. Taking into account that the variables $y$ and $x$ depend on $u$ and $v$ (2), if the first derivate function is applied:

$$d(y(u,v) - g(x(u,v))) = 0 \tag{3}$$

that is

$$g' = \frac{\frac{\sigma y}{\sigma v} v' + \frac{\sigma y}{\sigma u}}{\frac{\sigma x}{\sigma u} + \frac{\sigma x}{\sigma v} v'} \tag{4}$$

Using the previous expressions, the modification of the tangent angle values between the shapes in the image and the object planes can be calculated:

$$g' = \frac{-f sin\tau + (f cos\tau - u tg\delta) v' + v tg\delta}{f / cos\delta (cos\tau + sin\tau v')} \tag{5}$$

Thus, the previous transformation gives the tangent angle values for the shape edge points in the object plane with respect to the corresponding ones in the image plane.

# 3 Image transformation

The detection is based on the bidimensional process developed for the GHT [7]. Originally, the GHT was applied to detect bidimensional shapes in a 2-D space by using a template. The edge points of the template are transformed, according to the number of parameters that have to be calculated, and the resulting vectors are applied to each image edge point. Some authors, [8, 9, 10], applying invariant transformations, have calculated the orientation, the scaling, and the displacement of the image shape in relation to the template. In this paper we will extend the application of invariants to also detect the tilt and pan angles of planar shapes in 3-D space. In this situation, the template is located in the object plane and the method has to compare this shape with the image edge points in order to find all the parameters.

The edge points of the image are characterized by the parameters $< x, y, \theta >$ where $x$ and $y$ are the coordinates of the points in a two-dimensional space and $\theta$ is the angle of the gradient vector associated with this edge point. An angle, $\xi$, called *difference angle* is also defined. Its value indicates the positive difference between the angles of the edge point gradient vectors that will be paired.

From this description we can derive a transformation from the original image that generates new invariant information for the displacement and the scaling, based on point pairings. Thus, let $p_i$ and $p_j$ be two edge points, $< x_i, y_i, \theta_i >, < x_j, y_j, \theta_j >$ their associated information, and $\xi$ the difference angle to generate the pairing. Then, the transformation $\mathcal{T}$ can be expressed as follows:

$$\mathcal{T}(p_i, p_j) = \begin{cases} (\theta_i, \alpha_{ij}) & : \theta_j - \theta_i = \xi \\ \emptyset & : elsewhere \end{cases} \tag{6}$$

where

$$\alpha_{ij} = \left( \arctan \frac{y_i - y_j}{x_i - x_j} \ \angle \ \theta_i \right) \tag{7}$$

that is, $\alpha_{ij}$ is the positive angle formed by the line that joins $p_i$ and $p_j$ and the gradient vector angle of the point $p_i$.

Two new transformations will be defined to carry out the complete recognition process. The $\mathcal{S}$ transformation uses a similar expression to that in (7) to calculate the distance between the two pair points:

$$\mathcal{S}(p_i, p_j) = \begin{cases} (\theta_i, \alpha_{ij}, d_{ij}) & : \quad \theta_j - \theta_i = \xi \\ \emptyset & : \quad elsewhere \end{cases} \tag{8}$$

where

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{9}$$

Finally, for the third transformation, called $\mathcal{D}$, an arbitrary reference point, $O = (o_x, o_y)$ has to be defined and two vectors are generated:

$$\mathcal{D}(p_i, p_j) = \begin{cases} (\theta_i, \alpha_{ij}, \vec{r_i}, \vec{r_j}) & : \quad \theta_j - \theta_i = \xi \\ \emptyset & : \quad elsewhere \end{cases} \tag{10}$$

where

$$\vec{r_k} = O - p_k, k = i, j \tag{11}$$

The information generated by the application of the transformations is stored in different tables in order to improve the detection process speed. The multivalued characteristics of the previous transformations will be apparent during table building. Next, we show the contents of these tables:

**Distance Table ($DiT$)** . A bidimensional table that uses the information generated by the $\mathcal{S}$ transformation to increment the position indicated by $\alpha_{ij}$ (row) and $d_{ij}$ (column). Note that this information is invariant to the orientation and the displacement of the bidimensional shape.

**Orientation table ($OT$).** This is a bidimensional table that contains the information generated by the $\mathcal{T}$ transformation. The $\alpha_{ij}$ and $\theta_j$ values are stored in rows and columns, respectively. When a pairing with $\alpha_{ij}$ and $\theta_j$ value is calculated, the $OT[\alpha_{ij}][\theta_j]$ position is set. Because different pairings might coincide with the same $\alpha_i$ and $\theta_{ij}$ values, the content of $OT[\alpha_{ij}][\theta_j]$ will indicate how many of them have these values.

**Displacement table ($DT$).** The $\mathcal{D}$ transformation is applied in order to build this table. Linked lists are created in a similar way to the $OT$, but the data stored at each list position are the $\vec{r_i}$ and $\vec{r_j}$ reference vectors for the points $p_i$ and $p_j$, respectively.

## 4 Planar object detection

The planar object detection process uses the information of a previous template to calculate the position of a perspective projected, displaced, scaled, and rotated template from the object to the image plane. The aim of the detection process is to identify the new position of the template and calculate the applied transformation from the original template position.

The transformations applied to the original template to generate the new image are as follows:

- Displacement along the object plane: $d_x$ and $d_y$.
- Rotation in the object plane: $\beta$.
- Pan angle around object plane $z - axis$: $\tau$.
- Tilt angle around object plane $x - axis$: $\delta$.

Futhermore, we assume that the $z - axis$ of the object plane and the camera are parallel, but not coincident, as shown in figure 2, in order to obtain a more general expression.

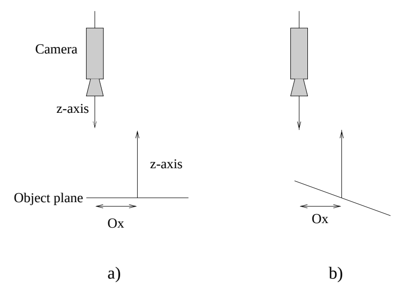



**Figure 2. Bidimensional description of parameter detection when the z-axis of the camera is parallel, but not coincident, with the z-axis of the object plane: a) Initial situation with no tilt and pan angles, and b) tilt and pan angles are applied to the object plane.**

Assuming that the edge points of the template are given by the function $t(x, y)$, each $(x, y)$ point of the template is transformed according to the equation (1):

$$\begin{aligned} u_i &= f\frac{(x'_i - o_x + d_x)cos\delta cos\tau - (y'_i - o_y + d_y)sin\tau + o_x}{(x'_i - o_x + d_x)sin(\delta) + d} \\ v_i &= f\frac{(x'_i - o_x + d_x)cos\delta sin\tau + (y'_i - o_y + d_y)cos\tau + o_y}{(x'_i - o_x + d_x)sin\delta + d} \end{aligned} \tag{12}$$

where $o_x$ and $o_y$ represent the displacement in the $x - y$ dimensions of the object plane z-axis with respect to the camera z-axis and

$$\begin{aligned} x'_i &= (x_i - x_r)cos\beta - (y_i - y_r)sin\beta \\ y'_i &= (x_i - x_r)sin\beta + (y_i - y_r)cos\beta \end{aligned} \tag{13}$$

indicates the rotation of the template in relation to a reference point $(x_r, y_r)$ in the object plane.

The values for $(u_i, v_i)$ pairs correspond to the coordinates of the projected shape into the image plane after applying the previous transformation. Thus, the detection of the template in the object plane requires the calculation of a large number of parameters. If the traditional formulation of the GHT is applied to this situation, the computational time could be very long. Computational complexity can be reduced by using invariant information from the images in such a way that the parameter calculation can be uncoupled. Obtaining the invariant information is based on the following theorem.

**Theorem:** Let $t(x, y)$ be the description of a template in the object plane and $T(\beta, d_x, d_y, \tau, \delta)$ the direct perspective transformation to be applied to this template, where $(d_x, dy)$ represents a displacement of the template along the object plane, $\beta$ is a rotation in the same plane, and $\tau$ and $\delta$ are the pan and tilt angles, respectively. Let $i(u, v)$ be the projection of $t(x, y)$ in the image plane after applying the direct perspective transformation $T$ in relation to a coordinate axis in an arbitrary position $(o_x, o_y)$. Then, the backprojected image $t'(x, y)$ from $i(u, v)$ using both a coordinate axis in the origin and the inverse perspective transformation $B(\tau, \delta)$ is identical to $t(x, y)$ except for a rotation, a scaling, and a displacement.

***Proof:*** The expression for the projection of each template point, taking into account the transformation $T$, is given by expression (12). The backprojected transformation $B$ can be applied to these points by using expression (2). Then, the backprojected points are calculated as follows:

$$x_i'' = d' \frac{x_i' - o_x + d_x - \frac{(d_x+o_x)cos\tau-(d_y+o_y)sin\tau}{cos\delta}}{d - (d_x + o_x)tan\delta cos\tau - (d_y + o_y)tan\delta sin\tau} \quad (14)$$
$$y_i'' = d' \frac{y_i' - o_y + d_y - \{(d_x + o_x)sin\tau + (d_y + o_y)cos\tau\}}{d - (d_x + o_x)tan\delta cos\tau - (d_y + o_y)tan\delta sin\tau}$$

where $d'$ is an arbitrary distance value used to carry out the backprojection.

If the $(x_i'', y_i'')$ coordinates of the backprojected points are compared to those of the original template $(x_i, y_i)$, it can be deduced that the backprojected image is similar to the template's except for a rotation, $\beta$, a scaling of the same value in the $x$ and $y$ coordinate, $s_t$, and a displacement in the $x$ and $y$ axis, $(d_{tx}, d_{ty})$ with the following values:

$$s_t = \frac{d'}{d - (d_x + o_x)tan\delta cos\tau - (d_y + o_y)tan\delta sin\tau} \quad (15)$$
$$d_{tx} = -o_x + d_x - \frac{(d_x + o_x)cos\tau - (d_y + o_y)sin\tau}{cos\delta}$$
$$d_{ty} = -o_y + d_y - \{(d_x + o_x)sin\tau + (d_y + o_y)cos\tau\}$$

## 4.1 Tilt and pan calculation

Invariant information for the scaling, rotation and displacement must be generated for the template and the backprojected image in order to perform the tilt and pan calculation. This information is revealed by using the $\mathcal{S}$ transformation that produces the $TDiT$ ($DiT$ for the template) and $IDiT$ ($DiT$ for the image).

Notice that the information generated by $\mathcal{S}$ transformation in the $DiT$ is invariant to the orientation and displacement of the shape. Thus, the detection of the pan and tilt angles can be carried out by comparing the $TDiT$ with the $IDiT$ of the shape for different tilt and pan values. If the dimensions of the shape in the image and the template were the same, this process could be carried out by comparing identical positions of the template and image tables and voting into a parameter space. The highest peak would indicate the best values for the tilt and pan angles.

However, in the most general case, the scalings of the template and the image shape are different as shown in expression (15). Then, the comparison has to be carried out taking into account a different scaling factor for the template. In this way, the highest peak will indicate the tilt and pan angles and the scaling factor.

### 4.1.1 Complexity of tilt and pan detection.

A matching algorithm is used to compare, by rows, the $TDiT$ and the $IDiT$ for different scaling values. The computational complexity of the method can be very high, depending on the range of the values for tilt and pan angles. However, three facts reduce computational time:

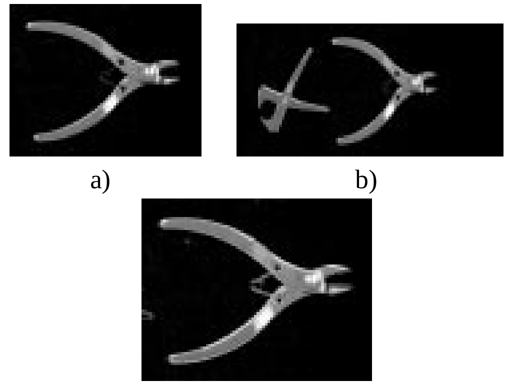
- Only the image edge points are backprojected.
- The gradient angle calculation of the backprojected points do not need the application of a contour operator. In fact, they are calculated using expression (5) for each combination of the tilt and pan angles.
- $TDiTs$ for different template scalings are generated in advance and multiresolution techniques are also applied.

## 4.2 Orientation calculation

The $OT$ information is invariant to the displacement and scaling of the shape due to angles $\alpha$ and $\theta$ remaining unchanged. Then, when the correct tilt and pan angles have been calculated with the previous step, the $OT$ of the template and the retroprojected image can be compared to find the orientation angle $\beta$. Taking into account that different orientation changes produce a rotation in the information of each $OT$ row, the comparison process has to compare similar rows of both tables for different shifts. A matching process between both tables is carried out for each shift generating a vote. The highest value indicates the correct value for the orientation. More details about this process can be found in [11].

### 4.3 Displacement calculation

In order to calculate the displacement $(d_{tx}, d_{ty})$, the value of the reference vector for each entry of the $TDT$, multiplied by the scaling, is applied to the coordinates of the pair points stored in the linked list associated with each entry of the $IDT$. The positions pointed out by these vectors are used to increment a bidimensional space. The maximum positions in this accumulator will give us the situation, in the backprojected image, of the equivalent reference point defined in the template shape. Then, the displacement can be calculated by applying the previous calculated scaling to the image reference point and subtracting this new position from the template reference point.



a) b) c)

**Figure 3. Detection of pliers: a) Template, b)Image (tilt=40°, pan=10°) and c) Superimposing the calculated retroprojected image on the template.**

## 5 Experimental results

We have used different images in order to test the behavior of the whole detection process. All the examples have been executed in an SGI O2 Workstation with a R-5000 processor at 180 Mhz. Cannys operator is applied to the template and the image to detect the edge points. We have also applied a simple multiresolution technique to reduce the number of computations, especially during the tilt and pan calculation. The resolution of the tilt and pan values has been modified from a coarse to a fine approach. The orientation calculation is carried out by using a narrow interval of tilt and pan angles around the values calculated in the previous step. In this way, more accuracy for the three parameter detection can be achieved. Finally, the displacement value is calculated.

The range of the tilt, pan and orientation angles for the experiments are (0°,50°), (-30°,30°), and (0° ,360°), respectively.

The accuracy of the detection process is checked by superimposing the retroprojected image shape on the template one. A difference angle, $\xi$, of 180° is used. However, several difference angles can be used depending on the characteristics of the templates [11].

First we have applied the detection to real images with no tilt and pan angles that have been projected using a warping process. In this manner, we can know the exact tilt and pan values which have been used and check the accuracy of the detection. This has been applied to figure 3 where the template is a plier. The projected image (pliers plus destapler) has been generated with tilt and pan values of 40° and 10°, respectively. Parameter detection coincides with the theoretical values. Following this, in figure 4, we used tilt, pan and orientation angles of 40°, 0° and 30° respectively, causing part of the pliers be left out. In this situation several pairings do not appear and the parameter detection has a slight error: the detected tilt angle is 41°, the pan angle is 2°, and the orientation is 28°.

a) b)

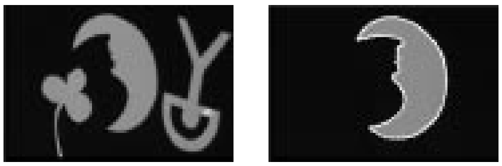
**Figure 4. Detection of pliers: a)Partially occluded image (tilt=40°, pan=0°, Orientation=30°) and b) superimposing on the template.**

We have also performed several examples using a real tilt-pan platform. These examples are shown in figure 5 and 6. In table 1 the values of the detected parameters are shown. In general, the computational time depends on the number of edge points. This is the reason for the high computational time of figure 6.c, where a lot of edge points appear due to reflections. A lower time can be achieved if a more efficient edge detection is performed. Our future works will try to reduce the computational complexity and apply more sophisticated multiresolution techniques to all

| Images | Points | Tilt | Pan | Ori. | $d_{tx}$ | $d_{ty}$ | Time (s.) |
|---|---|---|---|---|---|---|---|
| Moon | 2148 | 40 | 2 | 22 | 39 | -1 | 35 |
| Pliers1 | 4032 | 28 | 15 | 8 | 71 | -50 | 60 |
| Pliers2 | 6213 | 25 | 11 | 7 | 63 | -35 | 95 |

**Table 1. Calculated parameters**

the stages of the process detection.

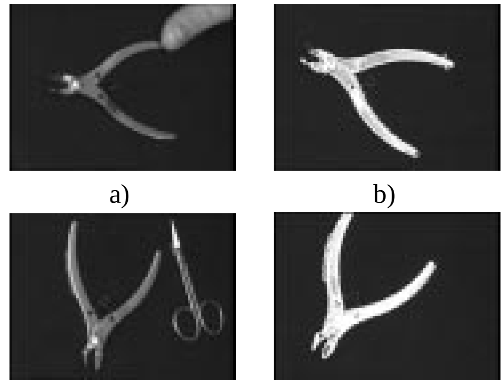


a) b)

**Figure 5. Detection of moon: a)Image and b) Superimposing the calculated retroprojected image on the template.**

## 6 Conclusions

A new method to detect planar shapes has been presented. The method copes with a situation in which the shape of the projected image presents a different displacement, orientation and scaling in relation to a template. It can also be applied to instances in which the object plane and camera $z - axis$ are parallel, but not coincident. The method is based on the GHT and generates invariant information by using gradient information that allows us to uncouple the parameter calculation and, in this way, reduce computational complexity. Important improvements have been introduced to save gradient angle calculation of the backprojected points. Several examples, that show the accuracy of the algorithm, have been also presented.



a) b) c) d)

**Figure 6. Detection of pliers: a) and c) Original images . b) and d) superimposing on the template.**

## References


[1] Sincalir. D. and A. Blake. Isoperimetric normalization of planar curves. *IEEE Transactions on Pattern Anal. and Machine Intell.*, 16(8):709–777, 1994.

[2] R. Lo and W. Tsai. Perspective-transformation-invariant generalized hough transform for perspective planar shape detection and matching. *Pattern Recognition*, 30(4):383–396, 1997.

[3] S.-C. Pei and L.G. Liou. Finding the motion, position and orientation of a planar patch in 3d space from scaled-ortographic projection. *Pattern Recognition*, 27(1):9–25, 1994.

[4] Yu Cheng. Analysis of affine invariants as aproximate perspective invariants. *Computer Vision and Image Understanding*, 63(2):197–207, March 1996.

[5] Z. Pizlo and A. Rosenfeld. Recognition of planar shapes from perspective images using contour-based invariants. *CVGIP:Image Understanding*, 56(3):330–350, November 1992.

[6] J.Y. Aloimonos. Perspective aproximations. *Image and Vision Computing*, 18(3):179–192, August 1990.

[7] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *PR*, 14(2):111–122, 1981.

[8] D.C.W. Pao, H.F. Li, and R. Jayakumar. Shapes recognition using the straight line hough transform: Theory and generalization. *PAMI*, 14(11):1076–1089, November 1992.

[9] N. Guil and E.L. Zapata. A fast generalized hough transform. In *European Robotic and Systems Conference*, pages 498–510, Malaga, Spain, August 1994.

[10] H. Chang, F.H. Cheng, W.H. Hsu, and Wu G.Z. Fast algorithm for point pattern matching - invariant to translations, rotations and scale changes. *Pattern Recognition*, 30(2):311–320, 1997.

[11] N. Guil, J.M. Gonzalez-Linares, and E.L. Zapata. Bidimensional shape detection using an invariant approach. *To appear in Pattern Recognition*.