Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1993-V02

Locating 3D Edges by Direct Methods in Motion Based Vision

Sagüés C., Guerrero J.J.

If you want to cite this report, please use the following reference instead:
Locating 3D Edges by Direct Methods in Motion Based Vision, Sagüés C., Guerrero J.J., *IEEE International Conference on Systems, Man and Cybernetics - ISBN 0-7803-0911-1*, pages 511-516, October 1993.

This work was partially supported by project ROB91-0949 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

Locating 3D Edges by Direct Methods in Motion Based Vision

C. Sagüés & J.J. Guerrero
Dpto. de Ingeniería Eléctrica e Informática
Centro Politécnico Superior, UNIVERSIDAD DE ZARAGOZA
María de Luna 3, E-50015 ZARAGOZA, SPAIN
(Phone 34-76-517274, FAX 34-76-512932, email: csagues@cc.unizar.es)

Abstract **Mobile vision leads to more robust methods of recovering structure than those based on binocular stereo. To solve the structure from motion problem, direct methods avoid both optical flow and correspondence computation. In this paper, we propose to obtain the location of 3D straight edges from a camera placed in the hand of a robot using these methods. The algorithm is based on image regions supporting the geometric features and can be applied with any known camera motion.**

1 Introduction

A robot works with objects in a real, prone to uncertainty environment. To the end of reducing the engineering which supplies the objects in prefixed locations, the robot has to be provided with sensorial capabilities.

Vision is the sensor which provides more information, but as with other sensors, its information is incomplete and noisy. It is expected that mobile vision research will lead to more robust methods of recovering depth than those used for binocular stereo. When the camera can be moved some advantages are introduced. So, third dimension can be extracted; camera can be directed towards the best position of observation; observed features can be fixed in the image; motion can be made as short or large as desired.

Methods for extracting shape and motion information from vision can be classified as optical flow-based, correspondence-based, and direct methods. Besides those, a sequence of images and some integration mechanism can be considered to improve the results of the perception.

In the first two approaches, after the computation of optical flow or the correspondence between features, the structure and motion is determined. Direct methods avoid both, correspondence and optical flow computation, and the structure and motion is extracted without this intermediate step.

In [4] some special cases to extract structure from motion with direct methods are addressed. In [7] the direct approach by fixation of a small patch of approximately constant depth is presented. In [1] the task of obstacle avoidance is solved without reconstruction of range. In [3] the direct approach is used combined with a Kalman filter algorithm to reduce the noise effects.

We present a direct approach to extract the 3D parameters of a straight line using the information from two images at least, which are taken by a camera with known motion located in the hand of a robot. We propose to achieve this goal with the following ideas:

- The use of geometric information better than the use of parametric information.
- Macro-features (straight lines, arcs of circle, ...) instead of points are extracted. Macro-features are more robust to occlusions, they are easier to extract and to match. Besides that, they provide

more information. Anyway, points can also be included (edge with tip).

- With the aim of reducing the location uncertainty, the information from regions better than from pixels themselves in the image is evaluated. So, feature support regions are the perception basis.
- The link of the different steps in the computation avoids to throw away information in intermediate results.
- The shape from motion and shape from contour may be the two most powerful ways to extract the structure when dealing with visual information. We propose the features to be contours of the object.

In §2 we present the background of the proposed algorithm. In §3 we present the method to locate straight lines making some considerations on edge orientation. Finally §4 is devoted to expose some conclusions and future developments.

2 Background of the approach

We adopt a pinhole camera model with a planar screen. The camera model is illustrated in figure 1. The origin of the camera coordinate system $OXYZ$ is on the projection center of the camera. The Z axis is aligned with the optical axis and the focal length is considered to be the unit. Without loss of generality, the image plane is in front of the pinhole to avoid dealing with inverted images. A point in the scene with $\mathbf{X} = (X, Y, Z)$ coordinates in that system is projected in the image with $(x, y, 1)$ coordinates, being:

$$x = \frac{X}{Z} \quad , \qquad y = \frac{Y}{Z} \tag{1}$$

The optical flow is the apparent motion of the brightness pattern in a sequence of images. Ideally the optical flow will correspond with the motion field. However local measurements of movement in the changing image generally provide only one component of local velocity, rather than the full 2D velocity vector. Therefore an additional constraint to compute the two-dimensional velocity field would be required.

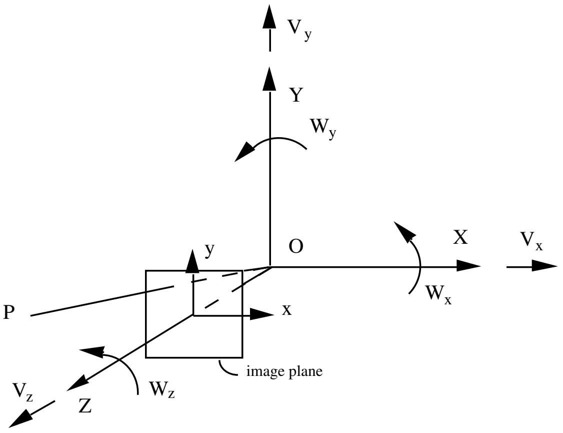

Figure 1: Pinhole camera model

Assuming that any given point on the object appears in the successive image frames with a constant brightness (which is a reasonable assumption in many situations), we can formulate the brightness constraint equation [5].

$$\frac{dE}{dt} = 0$$

where E denotes the brightness of a point in the image and t the time.

This equation can be rewritten as:

$$\frac{\delta E}{\delta x}\frac{dx}{dt} + \frac{\delta E}{\delta y}\frac{dy}{dt} + \frac{\delta E}{\delta t} = E_x\, u + E_y\, v + E_t = 0 \tag{2}$$

Motion field can be originated by object or camera motion. Let us suppose the camera moves with respect to the scene and its motion to be composed by a translation $\mathbf{t} = [V_x, V_y, V_z]$ and a rotation $\mathbf{w} = [W_x, W_y, W_z]$ in the camera reference system. The expressions relating the displacement in the image (u, v) of the point (X, Y, Z), with the camera motion are:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \tag{3}$$

$$+ \begin{bmatrix} x\,y & -(x^2+1) & y \\ y^2+1 & -x\,y & -x \end{bmatrix} \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix}$$

Instead of computing optical flow, the brightness constraint equation (2) and the motion field equation (3) are taken and one equation that links image brightness gradients to motion and structure parameters is obtained [6].

$$\frac{\mathbf{s}\cdot\mathbf{t}}{Z} + \mathbf{v}\cdot\mathbf{w} + E_t = 0 \qquad (4)$$

being $\mathbf{s} = [-E_x, -E_y, x\ E_x + y\ E_y]^T$ and $\mathbf{v} = [E_y + y(x\ E_x + y\ E_y), -E_x - x(x\ E_x + y\ E_y), y\ E_x - x\ E_y]^T$.

This equation relates the observer motion ($\mathbf{t}$, $\mathbf{w}$), the depth of the scene (Z) and certain measurable quantities of the images ($\mathbf{s}$, $\mathbf{v}$ and E_t).

In our approach we use the depth of the scene computed from (4) to replace it in the straight line equations.

3 Extraction of 3D features with a mobile camera

We extract the 3D parameters of a straight edge from two images, taken with a camera of known motion by direct methods.

The 3D features are represented as in the *Symmetries and Perturbation Model* [8] by a reference system attached to them and their location by the transformation from this reference to a global reference. The frame attached to an edge has the X axis in the direction of the line and the origin on any of its points. Thus, to obtain edge location, the direction (n_x, n_y, n_z) of the edge and the position (P_x, P_y, P_z) of a point on the edge must be given.

The first step in the procedure proposed is the extraction of spatial gradients to segment the image. Pixels are grouped into line support regions (LSR) of similar direction of brightness gradient [2], with the gradient magnitude larger than a threshold.

The masks used in the computation of the gradient are selected as a function of the detail of the lines to be searched. Large masks tend to smooth the image and to reduce the clarity of fine details.

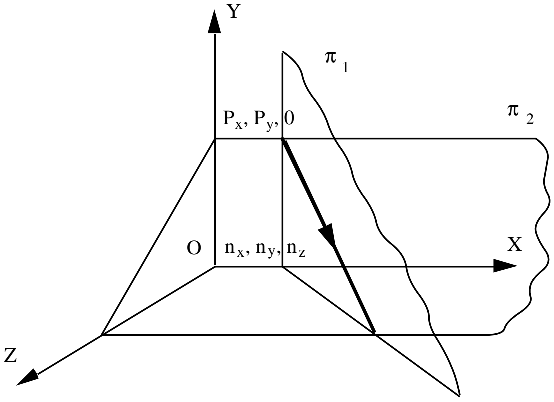

Figure 2: The intersection of the planes π_1 and π_2 gives the line

The segmentation in function of the gradient orientation is made using fixed and overlapped partitions. Two overlapped partitions of constant intervals (usually eight 45° intervals or sixteen 22.5° intervals) are used to group pixels according to the gradient direction. This partition overlap avoids that a straight line provokes separated LSR, when its orientation is close to the partition boundary.

So, we obtain line-support regions, containing all information available of the straight contours.

After the image segmentation, instead of computing directly the depth of each pixel or patch of pixels, we combine straight line equations and (4) to obtain relations linking image gradients and motion parameters to edge parameters.

3.1 Basis of the proposed method

Let us name $(P_x, P_y, 0)$ to the intersection point of the edge with the XY plane of the camera reference system.

If that point exists, n_z is not zero and the normality constraint (${n_x}^2 + {n_y}^2 + {n_z}^2 = 1$) can be applied, giving the following relation:

$${n_z}^2 = \frac{1}{{N_x}^2 + {N_y}^2 + 1} \qquad (5)$$

where:

$$N_x = \frac{n_x}{n_z} \quad , \quad N_y = \frac{n_y}{n_z} \tag{6}$$

So, the edge location is given with N_x, N_y, P_x, P_y parameters.

The 3D straight edge can be expressed as the intersection of two planes π_1 and π_2 being parallels to the Y and X axis of the camera reference system respectively (Figure 2).

From the direction vector of the edge, a normal vector to π_1 can be $\mathbf{n} = (n_z, 0, -n_x)$. Besides that, one of its points can be $\mathbf{P} = (P_x, 0, 0)$. Therefore the equation of π_1 plane expressed as $\mathbf{n} \cdot (\mathbf{X} - \mathbf{P}) = 0$, is:

$$n_z\ X - n_x\ Z - P_x\ n_z = 0 \tag{7}$$

With (1) and (6) equations, we have:

$$(x - N_x)\ Z - P_x = 0 \tag{8}$$

The expression $[(x - N_x)\ Z - P_x]^2$ is a fitting measure of generic points to this plane. Substituting the expression of the depth (Z) from equation (4) in this measure we obtain:

$$[(x\ -\ N_x)\ (\frac{-\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) - P_x]^2 \tag{9}$$

Similarly a normal to the π_2 plane can be $(0, n_z, -n_y)$ and one of its points $(0, P_y, 0)$. Operating as above, we arrive to:

$$[(y\ - N_y)\ (\frac{-\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) - P_y]^2 \tag{10}$$

The (9) and (10) expressions are two measures of the fitting of the line-support region to π_1 and π_2 planes. This fitting is expressed in function of the parameters of the straight line, the camera motion and data extracted from the images (spatial and temporal brightness gradients).

Integrating these measures by a least-squares based method along the LSR, we obtain the parameters of the location of the straight edge. Thus, the integrals are:

$$J_1 = \int\int_{LSR} [(x - N_x)(\frac{-\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) - P_x]^2\ dx\ dy$$

$$J_2 = \int\int_{LSR} [(y - N_y)(\frac{-\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) - P_y]^2\ dx\ dy$$

In the first measure expression, N_x and P_x are the unknowns. In the second measure N_y and P_y

For an extreme of J_1, we have:

$$\frac{\delta J_1}{\delta N_x} = 0 \quad , \quad \frac{\delta J_1}{\delta P_x} = 0$$

That is:

$$\int\int_{LSR} [(x - N_x)(\frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) + P_x][\frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}]\ dx\ dy = 0$$

$$\int\int_{LSR} [(x - N_x)(\frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}) + P_x]\ dx\ dy = 0$$

which constitutes a system of two equations in the two parameters of the edge unknowns (N_x, P_x), whose solution is:

$$N_x = \frac{S\ S_{v2x} - S_{vx}\ S_v}{S\ S_{v2} - {S_v}^2} \tag{11}$$

$$P_x = \frac{S_v\ S_{v2x} - S_{vx}\ S_{v2}}{S\ S_{v2} - {S_v}^2} \tag{12}$$

being:

$$\begin{aligned}
S &= \int\int_{LSR} dx\ dy \\
S_v &= \int\int_{LSR} \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}\ dx\ dy \\
S_{v2} &= \int\int_{LSR} \frac{(\mathbf{s} \cdot \mathbf{t})^2}{(\mathbf{v} \cdot \mathbf{w} + E_t)^2}\ dx\ dy \\
S_{vx} &= \int\int_{LSR} x\ \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \mathbf{w} + E_t}\ dx\ dy \\
S_{v2x} &= \int\int_{LSR} x\ \frac{(\mathbf{s} \cdot \mathbf{t})^2}{(\mathbf{v} \cdot \mathbf{w} + E_t)^2}\ dx\ dy
\end{aligned}$$

Similarly minimizing J_2:

$$\frac{\delta J_2}{\delta N_y} = 0 \quad , \quad \frac{\delta J_2}{\delta P_y} = 0 \tag{13}$$

We have a two equations system in the other location parameters of the straight line (N_y, P_y). Operating, we obtain:

$$N_y = \frac{S\ S_{v2y} - S_{vy}\ S_v}{S\ S_{v2} - {S_v}^2} \tag{14}$$

$$P_y = \frac{S_v\ S_{v2y} - S_{vy}\ S_{v2}}{S\ S_{v2} - {S_v}^2} \tag{15}$$

where:

$$\begin{aligned} S_{vy} &= \int\int_{LSR} y\ \frac{\mathbf{s}\cdot\mathbf{t}}{\mathbf{v}\cdot\mathbf{w}+E_t}\ dx\ dy \\ S_{v2y} &= \int\int_{LSR} y\ \frac{(\mathbf{s}\cdot\mathbf{t})^2}{(\mathbf{v}\cdot\mathbf{w}+E_t)^2}\ dx\ dy \end{aligned}$$

Finally n_z can be obtained from the (5), (11) and (14) equations, and n_x, n_y from (6) and the above equations. The location of a point on the edge to fix its reference system is obtained with (12) and (15) expressions and P_z equal to zero. The origin of the frame attached to the line is translated to a point in the middle of the projection of the line.

3.2 Considerations on edge orientation

The method is proposed as a general one. However, depending on edge orientation some considerations can be made on the planes describing the edge.

Edge ‖ XY plane

In this case the edge do not intersect the image plane, and the two planes π_1 and π_2 defined as in the precedent paragraph will be the same.

However, the depth of the line is constant. Its n_z parameter will be null and its n_x and n_y parameters will be those of its projection in the image.

The edge in the image can easily be extracted from the intersection of two brightness planes [2]. One of the planes is obtained by a least-squares fitting of the intensity slope in the LSR, weighting the brightness of pixels with gradient magnitude. The other plane has a constant height which is the mean brightness E_m (weighted with the gradient module) in the line support region.

From the projection in the image, a point (x_0, y_0) is obtained. It remains to obtain the P_z, that is also evaluated with a least-squares method. A measure of the fitting of the line-support region to the plane containing it and being parallel to the image plane $(Z = P_z)$ is minimized:

$$J = \int\int_{LSR} [P_z - (\frac{-\mathbf{s}\cdot\mathbf{t}}{\mathbf{v}\cdot\mathbf{w}+E_t})]^2\ dx\ dy$$

to obtain:

$$P_z = -\frac{S_v}{S} \tag{16}$$

The location of the reference system attached to the 3D straight edge will be given by a point (x_0P_z, y_0P_z, P_z) and the direction (n_x, n_y, 0) obtained previously.

Edge ‖ XZ or YZ planes

When the edge is parallel to XZ plane or YZ plane, its n_y or n_x component is, respectively, null. Anyway, the planes describing the edge are well defined and the algorithm described in 3.1 works correctly.

Edge ‖ X axis or Y axis

When the edge is parallel to X axis or Y axis its n_y, n_z or n_x, n_z are, respectively nulls but they can be considered as particular cases of the first one considered (edge ‖ XY plane).

Edge ‖ Z axis

When the edge is parallel to Z axis n_x and n_y are nulls and the edge will nearly appear in the image as a point. Anyway, the planes are also well defined.

3.3 Complete algorithm

The exposed method allows to obtain the 3D straight edge from two images taken with a camera moved without constraints in motion direction. Obviously with this direct formulation the camera motion must be small.

The complete algorithm can be summarized as follows:

- Segment the first image. Extract line-support regions.
- Evaluate the spatial and temporal brightness gradient in each pixel of the line-support region using at least two images.
- Obtain N_x, N_y from the expressions of §3.1.

  If ${N_x}^2 + {N_y}^2$ is less than a threshold we evaluate P_x and P_y as in §3.1.

  If ${N_x}^2 + {N_y}^2$ is greater than the fixed threshold, the straight line is considered to be parallel to the image plane or particularly to X or Y axis. In this case the procedure presented in §3.2 is used.

3.4 Using a different minimization norm

In the least-squares process a weighting norm can be considered. Thus, we propose the use of the temporal gradient due to translation ($E_t + \mathbf{v} \cdot \mathbf{w}$) which adds the following advantages:

- Avoids the problem that could origin a pixel having a null translational temporal gradient ($E_t + \mathbf{v} \cdot \mathbf{w} = 0$) in the evaluation of the integral terms.
- Pixels having larger temporal gradient due to translation are given more weight. These pixels correspond to closest points to the image plane which are the pixels adding less error (accuracy decreases with depth). Besides that, temporal gradient will be greater in pixels with greater spatial gradient (pixels closest to the theoretic edge).

A second factor can be added as weighting norm. It is interesting pixels near the theoretical edge to contribute more in the fitting than pixels far from it. Thus we introduce a function (N_w) proportional to gradient magnitude which reduces the contribution of pixels far from the edge. The integrals remains to be:

$$J_{1n} = \int\int_{LSR} [(x - N_x)(\mathbf{s} \cdot \mathbf{t}) + (\mathbf{v} \cdot \mathbf{w} + E_t) P_x]^2 \; N_w \; dxdy$$

$$J_{2n} = \int\int_{LSR} [(y - N_y)(\mathbf{s} \cdot \mathbf{t}) + (\mathbf{v} \cdot \mathbf{w} + E_t) P_y]^2 \; N_w \; dxdy$$

For an extreme of J_{1n} and J_{2n}, we obtain the line parameters.

$$N_x = \frac{S_{2x}\; V_2 - V_{sx}\; V_s}{S_2\; V_2 - {V_s}^2} \tag{17}$$

$$P_x = \frac{S_{2x}\; V_s - V_{sx}\; S_2}{S_2\; V_2 - {V_s}^2} \tag{18}$$

$$N_y = \frac{S_{2y}\; V_2 - V_{sy}\; V_s}{S_2\; V_2 - {V_s}^2} \tag{19}$$

$$P_y = \frac{S_{2y}\; V_s - V_{sy}\; S_2}{S_2\; V_2 - {V_s}^2} \tag{20}$$

being the integral factors:

$$\begin{aligned}
S_2 &= \int\int_{LSR} (\mathbf{s} \cdot \mathbf{t})^2 \; N_w \; dx \; dy \\
S_{2x} &= \int\int_{LSR} x \; (\mathbf{s} \cdot \mathbf{t})^2 \; N_w \; dx \; dy \\
S_{2y} &= \int\int_{LSR} y \; (\mathbf{s} \cdot \mathbf{t})^2 \; N_w \; dx \; dy \\
V_2 &= \int\int_{LSR} (\mathbf{v} \cdot \mathbf{w} + E_t)^2 \; N_w \; dx \; dy \\
V_s &= \int\int_{LSR} (\mathbf{s} \cdot \mathbf{t})(\mathbf{v} \cdot \mathbf{w} + E_t) \; N_w \; dx \; dy \\
V_{sx} &= \int\int_{LSR} x \; (\mathbf{s} \cdot \mathbf{t})(\mathbf{v} \cdot \mathbf{w} + E_t) \; N_w \; dx \; dy \\
V_{sy} &= \int\int_{LSR} y \; (\mathbf{s} \cdot \mathbf{t})(\mathbf{v} \cdot \mathbf{w} + E_t) \; N_w \; dx \; dy
\end{aligned}$$

When n_z is zero we must determine P_z. Using the new norm in the minimization, the depth is obtained from the following expression:

$$J_n = \int\int_{LSR} [(\mathbf{v} \cdot \mathbf{w} + E_t) P_z \; + \; \mathbf{s} \cdot \mathbf{t}]^2 \; N_w \; dx \; dy$$

and therefore:

$$P_z = -\frac{V_s}{V_2} \tag{21}$$

4 Summary and Conclusions

The use of geometric features in object recognition and localization for robotic tasks is noticeably interesting when working in unstructured scenes. Vision sensor provides several advantages with respect to other sensors. We propose the use of mobile vision with the camera in the hand of a robot to locate geometric features, such as straight lines and arcs of circles.

In this paper we present an algorithm to extract the location of straight edges by direct methods, from two images taken with a mobile camera . The proposed work avoids the difficulties in the matching or in the full extraction of optical flow.

Feature support regions in the image are used and with a least squares approach the 3D edge location is directly extracted. A weighting norm that improves the results is considered.

Actually we are implementing the proposed method to state its efficiency. We are also studying other approaches in the minimization process which are supposed to be more general and efficient. So we are researching the direct fitting of pixels in the LSR to a 3D straight edge and other method to locate the edge taking the information of an image completely processed as the basis and one second image with a simple processing.

The comparison of the methods we are now developing will be presented in future works.

Acknowledgements

This work was partially supported by project ROB91-0949 of the Comisión Interministerial de Ciencia y Tecnología (CICYT) and by project IT-5/90 of the Diputación General de Aragón (CONAI).

References

[1] Y. Aloimonos. Is visual reconstruction necessary? obstacle avoidance without passive ranging. *Journal of Robotic Systems*, 9(6):842–858, 1992.

[2] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

[3] J. Heel. Direct dynamic motion vision. In *IEEE International Conference on Robotics and Automation*, pages 1142–1147, 1990.

[4] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, (2):51–76, 1988.

[5] B.P.K. Horn. *Robot Vision.* MIT Press, Cambridge, Mass., 1986.

[6] S. Negahdaripour and B.K.P. Horn. Direct passive navigation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[7] M.A. Taalebinezhaad. Direct robot motion vision by fixation. In *IEEE International Conference on Robotics and Automation*, pages 626–631, Sacramento, California, April 1991.

[8] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE International Conference on Robotics and Automation*, pages 1799–1804, Nice, France, May 1992.