Proceedings

FIRST INTERNATIONAL CONFERENCE ON COMPUTER VISION

June 8-11, 1987
Royal National Hotel
London, England

Sponsored by
The Computer Society of the IEEE

In cooperation with
The International Association for Pattern Recognition

THE COMPUTER SOCIETY
OF THE IEEE

IEEE   THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC

IEEE
COMPUTER
SOCIETY
PRESS

Table of Contents

Structure from Motion

Matching

Object Recognition

Shape from X

On Projective Geometry and the Recovery of 3-D Structure

Ambjörn Naeve and Jan–Olof Eklundh

Computer Vision and Associative Pattern Processing Laboratory

Department of Computer Science, Royal Institute of Technology

S—100 44 Stockholm, Sweden

Abstract

Geometric properties are of key importance in the recovery of scene structure from images. It is argued that the proper formulations of the determination of scene geometry are obtained when projective geometry is used. A framework of projective geometry for computer vision is presented in brief and its advantages and applicability are demonstrated. Examples of how the framework can be used are given. It is also described how the needed features can be found with required accuracy.

Topic area: Computer vision.

Key words: Recovery of 3-D structure, contour features, projective geometry.

Introduction

The goal of computational vision is to derive descriptions of a scene from images of it. In particular, the descriptions could be in terms of primitives representing the geometric structure of the world. There are several reasons why such descriptions are important.

In a world of coherent objects and at the level of surfaces and volumes and their bounding contours important aspects of the structure are embodied in the geometry. Furthermore, it is well-known that the geometric cues, like the occluding boundaries of surfaces, impose very strong restrictions on the possible structure of the scene. This has been shown to be true both for single images of static scenes, [1], [2], for stereo images of static scenes [3], [4], and for time–varying imagery, [5] and others. In fact, geometric features tend to be much more useful than the photometric features in the computation of what is in the scene (see e.g. [1]). There is also an abundance of proposed methods for exploiting geometry in the recovery of scene structure, see [2] for an overview.

The basic paradigm in such approaches is to detect certain primitives in the images and determine their interrelationships which constrain the scene. Often the latter part requires some form of search or agglomeration procedure since the inverse problem at hand is underdetermined. Structure can therefore only be inferred from evidence obtained at different locations in the image. A classical example of this method is given by the so called Hough transform [6]. The primitives are then points, e.g. the edge points and the relation one tries to establish is that of collinearity. The problem is translated into dual space, where collinear points (points on a line) become concurrent lines (lines on a point). Discretization implies that the search can be performed by a simple counting procedure.

The Hough transform example works in the image and makes no reference to the image formation process. In most cases such references have to be made. The problem of determining the orientation of three planes intersecting at right angles in a corner has been treated by several authors e.g. by Mackworth [7], Barrow [8] and Barnard [9]. Mackworth uses the surfaces normals and utilizes certain constraining relations in gradient space in his search. Barnard uses lines as primitives and searches for orthogonality by maximizing volumes. Barrow on the other hand employs a strictly geometric construction in the image plane. The corner problem is important when scenes containing man–made objects are considered. Moreover, it is well–known that humans tend to ascribe right angle interpretations to corners when this is possible. Another, more general example is the detection of parallel lines in the scene. They map into vanishing points in the image. Hence, the primitives are the lines and the relation searched for is concurrency. An assumption has to be made: many (more than two) lines will generally not be concurrent in the image by accident. Barnard [10] suggests that the search for concurrency is carried out by mapping the image plane to the Gaussian sphere. In that way the search can be carried out in a compact space. In fact, a similar argument was put forward by Duda and Hart [6] when they explored the Hough transform. They did

not compactify the space, but they chose the polar equation for the lines and hence bounded one of the two variables.

Other problems in a similar vein concern the determination of symmetries (see e.g. [11]) and occluding contours, [12], [13], both being of paramount importance in recognition tasks. If we in particular consider a world containing man–made objects all these problems deal with issues treated in classical projective geometry. Bits and pieces of the wealth of results in geometry have also been used in computer vision. However, we contend that the approaches taken have often been somewhat arbitrary. The basic methodological principle of geometry as demonstrated by Klein e.g. in [14] states that problems should be solved by *changing the background*. In a terminology more familiar to practitioners in AI this implies that the key issue is representation. But representation is not only a question about choosing the primitives. It is also a matter of posing the problem in its right context, that is defining the solution space correctly. If we consider the problems on recovery of scene structure given above, we observe that they all deal with projective notions. The proper context in which to pose the problems is therefore in terms of projective geometry. We shall show that this indeed can be done and that several advantages accrue from that. We shall state three of the most important aspects here. First, the equations obtained are simple. Often they are linear or of the second order. This is very much at the heart of projective geometry. When the problems are solved in affine or Euclidean space unnecessary analytic complexity is introduced. Complicated trigonometric expressions are obtained when the problems are formulated in such metric settings. Secondly, the dimensionality of the solutions to these equations is often directly given by the problem formulation. It implies that the degrees of freedom of the solution are explicitly given. We know how much we don't know without making any ad hoc assumptions. This is highly desirable in computer vision tasks. Thirdly, the need to compactify the space to limit the search disappears. The projective spaces are compact!

At this point one could ask the question whether there are only mathematical arguments for such an approach, or if there exist motivations also from the point of view of biological vision. We have already observed that geometric cues are important for our understanding of the visual world. It can certainly be argued that our visual system is more qualitative (comparative) than quantitative (see e.g. Brooks [15]). Precise metric estimates are definitely not appropriate to describe the process. The use of relations, like coincidence, relative sizes etc., is more to the point. So, without claims as to how biological vision works, we can say that descriptions in projective terms are more appropriate than those in metric terms.

In this paper we shall introduce some of the basic notions of projective geometry needed to solve vision problems like those stated above. We shall then demonstrate how they can be used by giving solution to a few simple problems. Finally we shall outline a computational approach to finding the necessary primitives in image data.

Some basic notions of projective line geometry.

In [16] Naeve develops a complete theory of projective line geometry for computational vision. The gist of this theory is the way in which the imaging process is represented. Of particular importance is the coordinatization, e.g. because appropriate relations become evident and symmetries are explicit. In this paper we shall make use of this theory and the notations needed to demonstrate that problems on parallelism and orthogonality can be expressed in a straightforward way, amenable to simple solutions. Their solutions, also derived in [16], will be presented as well. It is beyond the scope of this paper to account for the theory, as a whole. The purpose here is to point to the fact that the proper mathematics exists and to show how it can be applied to real imagery.

We first introduce the coordinate systems (for more details see [16]). The projective point and plane coordinates can be defined using double ratios relative to the coordinate tetrahedron (C.T.), or, as in [16], by representing the points of projective 3–space by the lines through a fixed point in affine 4–space. If one selects a unit point and a unit plane according to Figure 1 one gets

$$\text{point}: \quad (x_1 : x_2 : x_3 : x_4) \qquad (1)$$

$$\text{plane}: \quad (\omega^1 : \omega^2 : \omega^3 : \omega^4) \qquad (2)$$

and the equation of joint position, in tensor notation

$$x_i \omega^i = 0 \qquad (3)$$

As a special case the affine coordinates are obtained when the plane at infinity is

$$\Pi_\infty : x_4 = 0 \qquad (4)$$

The affine coordinates are

$$\begin{cases} \text{point}: \ x_i/x_4 \\ \text{plane}: \ \omega^i/\omega^4 \end{cases} \quad i=1, 2, 3 \qquad (5)$$

The orthogonal Cartesian coordinates are obtained if the edges of the C.T., which are not in Π_∞ are mutually orthogonal. Then x_k and $\omega^k, k = 1, \ldots, 4$, are called homogeneous affine coordinates and homogenous Cartesian coordinates, respectively.

Given the point coordinates (1) one can now introduce line (ray) coordinates. A line p is determined by

two different points x and x' with projective point coordinates $(x_i)_1^4$ and $(x_i')_1^4$. From the matrix

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_1' & x_2' & x_3' & x_4' \end{pmatrix} \qquad (6)$$

one defines the six homogeneous projective ray coordinates ($\delta \neq 0$, arbitrary)

$$\begin{cases} \delta p_1 = x_1' x_4 - x_4' x_1 & \delta p_4 = x_2 x_3' - x_3 x_2' \\ \delta p_2 = x_2' x_4 - x_4' x_2 & \delta p_5 = x_3 x_1' - x_1 x_3' \\ \delta p_3 = x_3' x_4 - x_4' x_3 & \delta p_6 = x_1 x_2' - x_2 x_1' \end{cases} \qquad (7)$$

See also Figure 2.

We also denote them by $(p_\rho)_1^6$ and observe that they can not all be equal to zero. The following two facts are crucial (see e.g. [16] for proofs):

Proposition 1. $(p_\rho)_1^6$ denotes a line iff

$$p_1 p_4 + p_2 p_5 + p_3 p_6 = 0 \qquad (8)$$

Proposition 2. Two lines given by $(p_\rho)_1^6$ and $(q_\rho)_1^6$ intersect iff

$$p_1 q_4 + p_2 q_5 + p_3 q_6 + p_4 q_1 + p_5 q_2 + p_6 q_3 = 0 \qquad (9)$$

The point of intersection between a line and the plane $\omega = (\omega^i)$ can be found from the following:

Proposition 3. The line $p = (p_\rho)$ and the plane $\omega = (\omega^i)$ determine the point $z = (z_i)$ with coordinates

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} 0 & -p_6 & p_5 & p_1 \\ p_6 & 0 & -p_4 & p_2 \\ -p_5 & p_4 & 0 & p_3 \\ -p_1 & -p_2 & -p_3 & 0 \end{pmatrix} \begin{pmatrix} \omega^1 \\ \omega^2 \\ \omega^3 \\ \omega^4 \end{pmatrix} \qquad (10)$$

unless the product is zero, in which case the line p is contained in the plane ω.

The proof of this statement and its dual (the relation between a line and a point) can be found e.g. in [16].

In order to treat the example of a right angle corner we also need a way of expressing orthogonality in projective terms. To do so we have to consider the space as complex — that is, allow complex coordinates. It is easy to see that the intersection of an arbitrary sphere and $\Pi_\infty (x_4 = 0)$ satisfies the equations

$$\begin{cases} x_1^2 + x_2^2 + x_3^3 = 0 \\ x_4 = 0 \end{cases} \qquad (11)$$

This is a complex circle in Π_∞, called the sphere circle, S. Now if two intersecting lines are orthogonal their points of intersection with Π_∞ are complex harmonic

conjugates with respect to S. In other words, one point lies on the polar line to the other with respect to S. If the coordinates of the two points are $(\xi_1 : \xi_2 : \xi_3 : 0)$ and $(\eta_1 : \eta_2 : \eta_3 : 0)$ this means that

$$\xi_1 \eta_1 + \xi_2 \eta_2 + \xi_3 \eta_3 = 0 \qquad (12)$$

The arguments behind this equation can be found in most standard textbooks on projective geometry.

We are now ready to give some examples of how these notions can be used.

3. Determining parallel lines using line geometry

Parallel lines in a (3–D) scene are projectively lines that intersect at the plane of infinity. Hence, finding parallel lines only concerns lines and intersections of lines. Various methods have already been proposed to determine parallelism, see e.g. Kanade, [2], and Barnard [10]. However, neither of these approaches have exploated the simple structure the problem gets when posed in projective terms.

If we use line geometric notions a straightforward formulation of what we know is sufficient. We then get expressions that are simple to solve explicitly and that also show what the degree of underdeterminedness is.

In fact, let us assume that we have the parallel lines $(a_i)_1^6, (b_i)_1^6$ and $(c_i)_1^6$ given in the coordinate system of Figure 3. They will then intersect at a point $(0 : z_1 : z_2 : z_3)$ in the plane $(1 : 0 : 0 : 0)$. The point of intersection is given by (10). Since all the lines give the same point we find

$$\begin{pmatrix} a_5 \\ a_6 \\ a_1 \end{pmatrix} = \alpha \begin{pmatrix} c_5 \\ c_6 \\ c_1 \end{pmatrix} \quad , \quad \begin{pmatrix} b_5 \\ b_6 \\ b_1 \end{pmatrix} = \alpha \begin{pmatrix} c_5 \\ c_6 \\ c_1 \end{pmatrix} \qquad (13)$$

Moreover, we know that $(a_i)_1^6, (b_i)_1^6$ and $(c_i)_1^6$ define lines, that is by (8)

$$\begin{cases} a_1 a_4 + a_2 a_5 + a_3 a_6 = 0 \\ b_1 b_4 + b_2 b_5 + b_3 b_6 = 0 \\ c_1 c_4 + c_2 c_5 + c_3 c_6 = 0 \end{cases} \qquad (14)$$

In these equations a_2, a_3, a_4 etc. are observable in the image plane. To see what the solution of (11) and (12) are we can define the vectors $a, \hat{a}, b, \hat{b}, c, \hat{c}$ in analogy with

$$a = \begin{pmatrix} a_5 \\ a_6 \\ a_1 \end{pmatrix} \quad , \quad \hat{a} = \begin{pmatrix} a_2 \\ a_3 \\ a_4 \end{pmatrix} \qquad (15)$$

Using formal definitions on inner and outer products on these vectors we can write (13) and (14)

$$\begin{cases} a \cdot \widehat{a} = 0 \\ b \cdot \widehat{b} = 0 \\ c \cdot \widehat{c} = 0 \qquad (16) \\ a = \alpha c \\ b = \beta c \end{cases}$$

that is

$$\begin{cases} c \cdot \widehat{a} = 0 \\ c \cdot \widehat{b} = 0 \qquad (17) \\ c \cdot \widehat{c} = 0 \end{cases}$$

a system which is obviously solved by

$$c = \gamma(\widehat{a} \times \widehat{b}) \qquad (18)$$

We note that $\widehat{a}, \widehat{b}$ and $\widehat{c}$ are linearly dependent, since they form a pencil in the plane $(1 : 0 : 0 : 0)$. Moreover, α, β and γ are indeterminate. In fact, the projection operator onto $x_1 = 0$ in the chosen coordinate system has matrix (see [16])

$$\widehat{C}_1^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad (19)$$

Hence

$$\widehat{C}_1^1 \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \widehat{C}_1^1 \left[\begin{pmatrix} 0 \\ a_2 \\ a_3 \\ a_4 \\ 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} a_1 \\ 0 \\ 0 \\ 0 \\ a_5 \\ a_6 \end{pmatrix} \right] = \begin{pmatrix} 0 \\ a_2 \\ a_3 \\ a_4 \\ 0 \\ 0 \end{pmatrix}$$
$$(20)$$

that is

$$\begin{pmatrix} a_1 \\ 0 \\ 0 \\ 0 \\ a_5 \\ a_6 \end{pmatrix} \epsilon \ker \widehat{C}_1^1 \qquad (21)$$

4. Application of line geometry to the corner problem

Assume that the planes in 3-space intersect at right angles and that the scene is projected onto an image plane, Π, by a sensor (camera or eye) at position P. The introduction of projective coordinates described

above suggests a coordinate system which is illustrated in Figure 1 and Figure 2. Let the three lines be (see Figure 4)

$$\begin{cases} p^a \underline{\Delta}(a_i)_1^6 \\ p^b \underline{\Delta}(b_i)_1^6 \qquad (22) \\ p^c \underline{\Delta}(c_i)_1^6 \end{cases}$$

If the image plane is $x_1 = 0$ then the projections of the lines into the image plane have components 1, 5 and 6 equal to 0. If we (like most other authors who have treated the problem) make the simplifying assumption that the eye is at infinity and set $P = (1 : 0 : 0 : 0)$ then the projected lines are

$$\begin{cases} (0 : a_2 : a_3 : a_4 : 0 : 0) \\ (0 : b_2 : b_3 : b_4 : 0 : 0) \qquad (23) \\ (0 : c_2 : c_3 : c_4 : 0 : 0) \end{cases}$$

We now simply state what we know about the configuration:

$$(a_i)_1^6, (b_i)_1^6, (c_i)_1^6 \quad \text{represent lines} \qquad (i)$$

$$p^a, p^b, p^c \quad \text{intersect} \qquad (ii)$$

$$p^a, p^b \text{and } p^c \quad \text{form right angles} \qquad (iii)$$

Propositions 1 and 2 give equations corresponding to *(i)* and *(ii)*. To use *(iii)* we note that (11) implies that the points of intersection with Π_∞ are $(a_1 : a_2 : a_3 : 0), (b_1 : b_2 : b_3 : 0), (c_1 : c_2 : c_3 : 0)$. We then use (12) and get

$$\begin{cases} a_4 a_1 + a_2 a_5 + a_3 a_6 = 0 \\ b_4 b_1 + b_2 b_5 + b_3 b_6 = 0 \\ c_4 c_1 + c_2 c_5 + c_3 c_6 = 0 \\ b_4 a_1 + a_2 b_5 + a_3 b_6 + a_4 b_1 + b_2 a_5 + b_3 a_6 = 0 \\ c_4 a_1 + a_2 c_5 + a_3 c_6 + a_4 c_1 + c_2 a_5 + c_3 a_6 = 0 \quad (24) \\ b_4 c_1 + c_2 b_5 + c_3 b_6 + c_4 b_1 + b_2 c_5 + b_3 c_6 = 0 \\ a_1 b_1 + a_2 b_2 + a_3 b_3 = 0 \\ a_1 c_1 + a_2 c_2 + a_3 c_3 = 0 \\ b_1 c_1 + b_2 c_2 + b_3 c_3 = 0 \end{cases}$$

The first six equations are linear in the unknowns $a_1, a_5, a_6, b_1, b_5, b_6, c_1, c_5, c_6$. If the corner is in a general position, this linear system has rank five. The last three equations are quadratic. Choosing one of the unknowns as a parameter and solving the combined linear-quadratic system one obtains exactly two solutions for each parameter value. This explicitly tells us that

- the distance to the corner is undetermined

- for each value of the distance there are exactly two solutions, corresponding to depth reversals.

We summarize the two examples by noting that

- the solution was found by using exactly the information that is given, (i)–(iii);

- the degrees of freedom in the solution come out explicitly.

These were the goals we wanted to achieve and we have shown that we attained them in these, albeit simple, examples.

5. Using projective line geometry — a computational approach

We have introduced some basic notions of projective line geometry which can be used for deriving scene structure from image structure. The framework is presented in more detail by Naeve in [16]. In order to apply this framework to real image data it is of paramount importance to find the primitives (like lines and conics) with high precision at their exact positions and with correctly determined directions. Due to various types of noise and irrelevant details in the image data this is difficult. There is a need to smooth the data without destroying the positional accuracy. Recent work on edge detection (see, e.g. [17] and [18]) and on scale–space description of image structure (see, e.g., [19] – [21]) have considered these conflicting goals. We propose an approach to finding the geometric primitives which embodies these principles. In principle the approach works as follows (see also [22]):

- First, significant edges are detected by a method that focuses in on the edges found by an operator that blurs the image considerably. Scale is treated continously, and the step size in scale is determined analytically based on a model of possible shifts in edge position as scale varies. A detailed account can be found in Bergholm [23].

- Secondly, contours are traced in a non–committing manner. This means that simple rules of good continuation and multiple thresholds are used. However, the contour follower is not forced to find semantically meaningful boundaries, e.g., by filling gaps. The necessary information is not always available at this processing level.

- Finally, the traced contours are piecewise approximated with first and second order polynomials. This is also done in scale–space, scale now corresponding

to the error tolerance. In this manner the contours are smoothed but position, direction and incidence structure is preserved for the important features. For details see Bengtsson et. al., [24].

Early results show that this approach works well and that it can give primitives to be used in the mathematical framework we have presented. Ongoing work aims at showing this for real imagery.

6. Conclusion

We have presented a framework for using projective geometry in the recovery of 3–D scene structure. We have also presented solutions to a few simple but important examples. These examples can all be treated in Euclidean terms. However, the projective approach implies that we get simple and explicit formulations of what is known and what is not known. The required search for structure can be replaced by explicit solutions to systems of equations, since $\mathcal{P}^n$ is compact. Hence the presented framework can be considered as appropriate for finding the scene structure one is looking for.

References

[1] Barrow, H.G., Tenenbaum, J.M.,
Interpreting Line Drawings as Three–Dimensional Surfaces,
AI, Vol. 17, 1981, pp. 75–116.

[2] Kanade, T.,
Geometical Aspects of Interpreting Images as a Three–Dimensional Scene,
Proc. IEEE, Vol. 71, No. 7, 1983, pp. 789–802.

[3] Baker, H.,
Depth from Edge and Intensity Based Stereo,
AIM 347, Stanford AI Laboratory, Sept. 1982.

[4] Medioni, G. G.,
Matching Images Using Linear Features,
ISG Report 103, Department of EE and Computer Science, Univ. Southern California, Aug. 1983.

[5] Aggarwal, J.K., Mitiche, A.,
Structure and Motion from Images,
Proc. DARPA IUW, 1985, pp. 89–97.

[6] Duda, R. O., Hart, P. E.,
Use of the Hough Transform to Detect Lines and Curves in Pictures,
CACM, Vol. 15, No. 1, 1972, pp. 11–15.

[7] Mackworth, A. K.,
Model–Driven Interpretation in Intelligent Vision Systems,
Perception, Vol. 5, 1976, pp. 349–370.

[8] Barrow, H. G.,
Another Look at Polyhedal, Scene 1 — Incidence,
Memo DAI–WP–6, Dept. of Artificial Intelligence, Univ. of Edinburgh, Dec. 1974.

[9] Barnard, S. T.,
Choosing a Basis for Perceptual Space,
CVGIP, Vol. 29, 1985, pp. 87–99.

[10] Barnard, S. T.,
Interpreting Perspective Images,
AI, Vol. 21, 1983, pp. 435–462.

[11] Kanade, T.,
Recovery of the Three–Dimensional Shape of an Object from a Single View,
AI, Vol. 17, 1981, pp. 409–460.

[12] Marr, D.,
Analysis of the Occluding Contour,
Proc. Roy. Soc. of London, Ser. B, Vol. 197, 1977, pp. 441–475.

[13] Stevens, K. A.,
The Visual Interpretation of Surface Contours,
AI, Vol. 17, 1981, pp. 47–74.

[14] Klein, F.,
Höhere Geometrie
Chelsea Publishing Company, New York, 1949.

[15] Brooks, R. A.,
Symbolic Reasoning Among 3–D Models and 2–D Images,
AI, Vol. 17, 1981, pp. 285–348.

[16] Naeve, A.,
Projective Line Geometry of the Visual Operator,
TRITA–NA–8606, Department of Computer Science, Royal Institute of Tecnology, Stockholm, Sweden, 1986.

[17] Marr, D., Hildreth, E. C.,
Theory of Edge Detection,
Proc. Royal Society of London, Ser. B, Vol. 207, 1980, pp. 187–217.

[18] Canny, J. F.,
A Computational Approach to Edge Detection,
IEEE, Vol. 8, 1986, pp. 679–698.

[19] Witkin, A. P.,
Scale–Space Filtering,
Proc. IJCCAI-83, 1983, pp. 1019–1022.

[20] Koenderink, J. J.,
The Structure of Images,
Biological Cybernetics, Vol. 50, No. 5, 1984, pp. 363–370.

[21] Mokhtarian, F., Mackworth, A.,
Scale–Based Description and Recognition of Planar Curves and Two–Dimensional Shapes,
IEEE–PAMI, Vol. 8, 1986, pp. 34–43.

[22] Bergholm, F., Eklundh, J. O., Howako, J.,
Finding Geometric Structure,
Proc. 4th SCIA, Trondheim, Norway, June 1985.

[23] Bergholm, F.,
Edge Focusing,
TRITA–NA–8518, Department of Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1985. To appear in IEEE–PAMI.

[24] Bengtsson, A., Eklundh, J.O., Howako, J.,
Shape Preserving Contour Approximation Using First and Second Order Polynomials,
TRITA–NA–8607, Department of Computer Science, Royal Institute of Technology, Stockholm, Sweden.