# Intermediate Representation for 3D Model Based Recognition from Intensity Image

Sergiu Nedevschi
Technical University of Cluj-Napoca
Cristian Goina
I.P.A. Cluj-Napoca

**Abstract**
An intermediate representation to allow the recognition of a 3-D class of objects, from an intensity image is proposed. Determination of the intermediate representation from a CAD model and from an intensity image is presented.
**Key Words**: image processing, 3-D models, feature extraction, intermediate representation.

## 1. Introduction

The problem of automatic recognition of objects viewed from an arbitrarily point of view, was defined by Besl and Jain in [1]:

- Given any collection of labeled solid objects:
  - Each object is examined using a more or less automatic method;
  - Models of the labeled objects are created using information resulted from examinations;
- Given a digital image signal corresponding to a particularly and arbitrarily point of view; all models previously defined being accessible; given a list of distinguishable objects; the following actions are taken for each object from the list:
  - It is checked whether the object is present in the digital image signal; it has to be seen whether the object can give a possible consistent interpretation for a signal's subset.
  - If it can, the number of possible interpretations will be determined.
  - For each object instance both the position and orientation in the image and the position and orientation in the scene is computed.
- If there are regions in the image signal that can't be put in a correspondence with an object from the list, the features of these regions are computed and stored, in order to be recognized in other signals.

The above specified problem is known as model based object recognition.

The model based object recognition problem can be divided in three subproblems[6]:

- features selection and extraction;
- model and scene representation;
- matching between model and scene representations.

An effective solution of this problem can be obtained only taking into account the aims of the recognition task:

- the class of the recognizable objects;
- the type of sensorial system used;
- the performances of the recognition system.

The class of the objects which has to be recognized influences both the representation dimensionality and the type of the surfaces used.

The sensorial system determines the type of the image signal. Either intensity images or depth images can be used. Depth images allow the use of 3-D features and representations.

Among the performances with a great importance in a recognition system implementation we mention: the recognition speed , the accuracy, the ability to recognize partial occluded objects, the ability to recognize an object from any viewpoint.

A high speed of recognition can be obtained only by imposing constraints to the class of objects, lightning conditions and environment configuration. These constraints make possible the reduction of the representation dimensionality and implicit they quicken the features extraction, building representation and matching process.

The improving of the recognition accuracy implies the increase of the representation dimensionality and the use of the surface descriptors which reduce the speed of the process.

To make possible the recognition of the partially occluded objects we give up the usage of the global features in the scene representation and we work with a set of local features.

For the objects viewed from an invariable viewpoint or from a small set of invariable viewpoints, a model or a set of models centered on the view direction will be used. The need to recognize objects from any view direction, requires object centered models.

According to the dimensionality used in model representation the model based recognition systems can be subdivided in three classes: 2-D, 2 1/2-D, and 3-D. For each of these classes the features extraction, models and scene representation and matching problems are solved using specific methods.

The 2-D representations are viewer centered representations and use 2-D features extracted from the image space. There is one representation for each distinct view of model object. Generally these representations are made up from shape characteristics inferred from intensity images of the model object. They are used either for 2-D object recognition or three dimensional simple objects with a reduced number of stable positions for an invariant viewpoint.

The 2 1/2-D representations are viewer centered representations and use features extracted from the surface space. These features are extracted either from the depth image or from the orientation surface map which can be inferred from the intensity image. The representation is used for 3-D complex objects with a limited number of stable positions for an invariant viewpoint.

The 3-D representations are accurate, volumetric, object centered representation which use features of the object space.

They are independent of the viewpoint and make possible computations for an arbitrarily viewpoint at an arbitrary detail level. They are built from more sensor information (more depth images or more intensity images) or by means of a CAD tool( descriptive languages or interactive visual languages). They are used when the object may have any position and orientation in space or it may be viewed from any viewpoint.

As a conclusion the general problem of automatic object recognition from any arbitrarily viewpoint require object centered 3-D models.

In order to implement an effective recognition algorithm a matching method for model and image data representation must be found.

Since the object model has more information then input data, it is not possible to convert the input data into complete model data, and do matching in the model format. In order to reduce the dimension of the problem, it is profitably to work in an intermediate space, which can be computed both from the input data and model.

## 2. Intermediate representation

The input data have to be compressed by an abstraction process in order to make information supplied by the sensorial system usable. This is achieved by the low level processing and is materialized by features extraction from the image. The features yield a symbolic description which points out the geometrical properties and local relationships of the image.

On the other hand, on object model, is a collection of both 3-D primitives and relations between them. This representation has to be converted in such a way that, the observable object features to be explicitated.

In order to improve the matching process it is necessary to define a representation easy obtainable both from the image and the model (Fig. 1). Such a representation is called "intermediate representation".

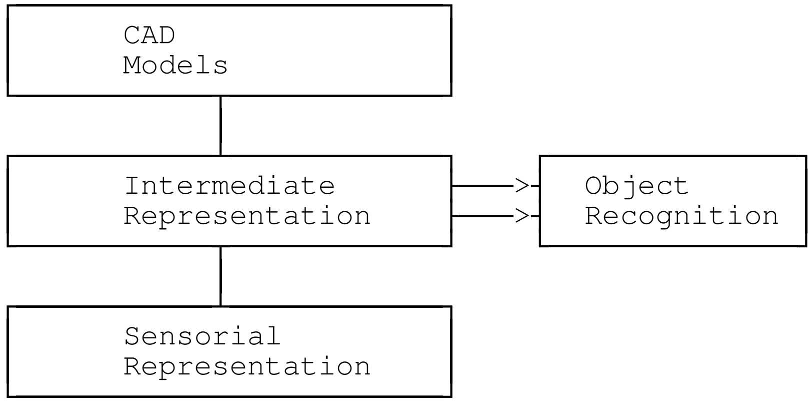


Fig. 1

The intermediate representations category include the symbolic scene description [6], sensor-tuned representation [12], structural hierarchical description [11].

The symbolic scene description representation is based on features which can be immediately detected from the image. The features by means of their attributes and relations between them depict the objects from the scene.

The sensor-tuned representation can also include more other significant features of the object or scene, especially the geometrical ones, which aren't immediately detectable from the sensorial data. Let's try to recognize a box in a scene where only the topmost surface is completely visible. Assume that there are several object models which have identical surfaces with the top surface of the box, but different heights. A matching scheme which uses only features extracted from the sensorial data (surfaces, edges, corners) will not be able to solve the ambiguity. On the other hand if the location of the base plane is known, and information about the surface height is available then this information can be used to augment the visual information and identify the object correctly. This approach is very useful in complex scenes where each object imposes geometric constraints on the objects around it.

The structural hierarchic description is based on both the features which can be immediately detected from the image and those resulted from a feature grouping process. While the features detection is a bottom-up process, grouping them together is a top down one and makes assumptions about the scene geometry. Through the grouping process we can get more abstract feature hierarchies which quicken the matching process.

Although a good representation has to fulfil the validity criteria completeness and uniqueness, in case of intermediate representation the fulfillment of these conditions is not compulsory, laying the stress on helping the recognition process.

The intermediate representation depends on the type of the image signal supplied by the sensorial system.

In the case of depth images the intermediate representation uses information about surfaces, which can be easily obtained from these. Generally, surface features like shape, location, orientation, relationships between surfaces like adjacency, or relative orientation, information concerning shape of the edges which separate surfaces, are extracted.The above specified features are invariant to rotation, translation, scaling, thus improving the matching.

In the case of intensity images, the intermediate representation is based on 2-D features and relationships, which can be associated with 3-D objects. From 3-D object properties the ones which are invariant to projection have to be chosen. The junctions defined as the intersection of 2 or more edges always denote a vertex projection. The contour symmetries from intensity images may represent some symmetric object projections.

## 3. The intermediate representation for 3-D object recognition from an intensity image.

An intermediate representation based on local features in order to allow the recognition of a 3-D class of objects, bordered by planar, conic or cylindrical surfaces, from an intensity image is proposed.

The primitive of this representation is the line segment. The line segments are obtained either from the geometric model by projection or from the intensity image through edge extraction. The primitives themselves and those associated by relations of adjacency, parallelism, coliniarity, symmetry, generate basic features.

Analyzing basic features properties, projection invariant 2-D features are identified , which can be easily associated with some 3-D elementary objects. These invariants will be referred as index features.

The basic objects which compose most of the industrial parts are: the parallelipiped, the cylinder, the cone or sections of these objects.

Table I shows the basic objects with their index feature generator elements, and table II shows the index features generated by index feature generator elements.

**Table I**

| Basic object | Index feature generator element |
|---|---|
| Parallelipiped: | 3-D parallelogram, vertex |
| Cylinder: | 3-D circle, cylinder side, vertex |
| Cone: | 3-D circle, cone side, vertex |
| Polyeder: | 3-D polygon, vertex |

**Table II**

| Index feature generator element | 2-D index feature |
|---|---|
| 3-D circle: | circle, ellipse, circular arc, ellipsoidal arc |
| Cylinder side: | parallel lines |
| Cone side: | 2 adjacent lines |
| 3-D parallelogram: | 2-D parallelogram |
| 3-D polygon: | 2-D polygon |
| Vertex: | junction, angle |

The quantitative measure of the basic and index features is given by a set of specific attributes (Tables III and IV).

**Table III**

| Basic feature | Attribute |
|---|---|
| Line segment: | centre coord., length, orientation |
| Collinear lines: | distance |
| Polygonal line: | lines number, lengths of lines, angles between lines |
| Polygonal contour: | lines number, lengths of lines, angles between lines |

**Table IV**

| Index feature | Attribute |
|---|---|
| Parallel lines: | parallels centre coord., orientation, median orientation, medium length, distance |
| Angle: | angle, lengths of lines, intersection coord. |
| Junction: | type, angles between lines, lengths of lines, junction coord. |
| Parallelogram: | parallelogram centre coord., orientation, lengths of lines, angles between lines, distances between facing lines |
| Circular arc: | centre coord., radius, end points coordinates in the local coord. system |
| Circle: | centre coord., radius |
| Ellipsoidal arc: | centre coord., axes lengths, major axis orientation, end points coord. in the local coord. system |
| Ellipse: | center coord., axes lengths, major axis orientation |

A local coordinate system is associated with each index feature. The position of the index feature can be computed from the relation between the local coordinate system and the global one.

In order to allow subsequent refinements, both index and elementary features are kept in the intermediate representation.

## 4. Determination of the intermediate representation from the model

The 3-D models are generated using the Boundary Representation scheme(BR). Models are depicted in the data base by relations as:

*surface(MId, SId, FirstSegId, SType, ConturType, (SAttrib))*
*segment(MId, SId, EId, SegId, NextSegId, Type,(SegAttrib))*
*edge(MId, EId, Seg1Id, Seg2Id, Type, (EdgeAttrib))*
*adjacency(MId, S1Id, S2Id, AdjId, (AdjAttrib)),*

MId - model identifier;
SId - surface identifier;
SegId - segment identifier;
EId - edge identifier;
AdjId - adjacency identifier;
SType - planar, cylindrical, conic;
Type - linear, circular, elliptic.

The generation of the intermediate representation of an object has two stages:
- the computation of the orthogonal or perspective projection along a given direction;
- features extraction and attributes computation.

The projection computation algorithm consists of:
- rotation of the coordinate system so that the Oz axis to become the projection direction;
- the projection along the Oz axis;
- the elimination of the hidden edges and surfaces.

The curved edges and surfaces are piecewise approximated by line segments and planar surfaces respectively. The false edges that correspond to conic and cylindric generator axes, visible due to approximation, have to be eliminated.

As a result a set of segments S is obtained, representing visible edge projections, depicted by the following relation:

*sline(MId, EId, SegId, LineId, (LineAttrib)),*
where:

*LineId* - projection line identifier;
*EId,SegId* - the edge and segment identifiers which the projection line is generated from;

This manner of projection representation allows an easy identification of the index features.

If there is a subset Ck in the set of projections S, such that:

$$C_k = \{ s \mid \forall s_i, s_j \in C_k, s_i \neq s_j, s_i.IdSeg = s_j.IdSeg \} \quad (1)$$

and card(Ck) >= 4, then all line segments from Ck come from the projection of a curve segment and have to be converted into circular or elliptic features.

By associating the projections that come from the segments which belong to the same face, polygonal contours or polygonal lines Pk are obtained, where:

$$P_k = \{ s \mid \forall s_i, s_j \in P_k, s_i \neq s_j, s_i.IdSeg.IdS = s_j.IdSeg.IdS \} \quad (2)$$

From the polygonal contours which correspond to planar faces parallelogram features are extracted and from polygonal contours which correspond to cylindrical or conic faces index features (parallels, ellipses, circles, arcs or angles) are extracted.

For all extracted features the proper attributes are computed.

The intermediate representation for a 3-D object viewed from a given viewpoint consists of the following relations:

*line(lineId, $x_c$, $y_c$, length, orientation)*
*polygonalContur(poligContId, (list of lines id.), numberOfLines,*
*(list of lines lengths), (list of angles between adjacent lines))*

From these basic features the following index features are determined:

*angle( angleId, ( firstLineId, secondLineId ), $\Theta$, $x_a$, $y_a$,*
*( firstLineLengths, secondLineLengths))*
*junction( junctionId, (list of lines id.), type, (list of lines lengths),*
*(list of angles), $x_j$, $y_j$)*
*parallels( parallelId, ( firstLineId, secondLineId ), $x_c$, $y_c$, $\Theta$, $\Theta_l$, length, distance)*
*parallelogram( parallelogramId, (firstParallelsId, secondParallelsId),*
*$x_c$, $y_c$, $length_1$, $length_2$, $distance_1$, $distance_2$, $\Theta$, $\Theta'$)*
*elipsoidalArc( ellArcId, (list of lines id.), $x_c$, $y_c$, (lengths of axes),*
*majorAxisOrientation, $\Theta_1$, $\Theta_2$)*
*circularArc( circArcId, (list of lines id.), $x_c$, $y_c$, radius, $\Theta_1$, $\Theta_2$)*

## 5. Determination of the intermediate representation from the intensity image

The determination of the intermediate representation from the intensity image involves edge detection followed by feature extraction and attribute computation.

In order to accomplish edge detection, the algorithm presented in [15] was implemented. It has the following steps:

- filtering with a median and a Gaussian filter in order to eliminate the noise and to smooth the surfaces;
- the computation of the second order directional derivative along the gradient direction;
- determination of the zero crossings of the second order directional derivative with subpixel

accuracy;

- edge extraction as a list of points;
- contour closing;
- interpolation of the contour lines and closed contours by line segments.

As a result of the contour detection stage basic features (line segments, polygonal lines and polygonal contours) are obtained.

The next stages of determination of the other features and the computation of their attributes as well are exclusively based on the results obtained in the first stage.

The extraction of the features like angle, junction can be done according to the methods presented in [12].

In the case of the parallels (Fig. 1) the extraction of the index features is done by selecting those segments which fulfil the condition:

$\Theta i - \Theta j < \varepsilon$, where $\Theta i$ and $\Theta j$ represent the directions of the segments.

Knowing the centres of the line segments their lengths and directions, the following attributes are computed:

$$
\begin{aligned}
l &= \frac{l_1 + l_2}{2} \\
x_c &= \frac{x_{m1} + x_{m2}}{2} \\
y_c &= \frac{y_{m1} + y_{m2}}{2} \\
m &= \sqrt{(x_{m1} - x_{m2})^2 + (y_{m1} - y_{m2})^2} \\
\theta_1 &= arctg \frac{y_{m1} - y_{m2}}{x_{m1} - x_{m2}} \\
d &= m \bullet \sin(\theta_1 - \theta)
\end{aligned}
\tag{3}
$$

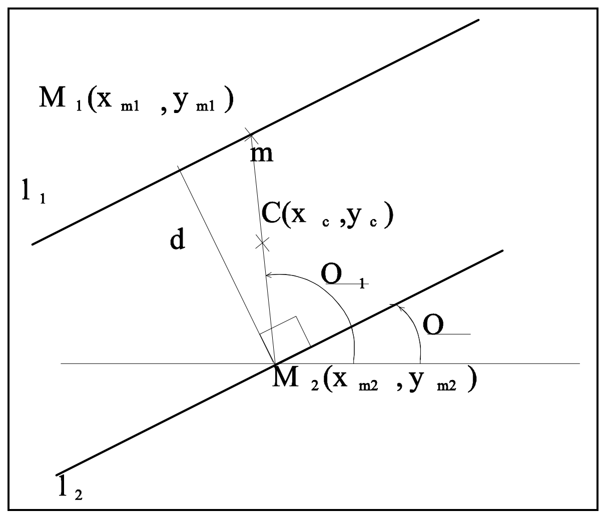

Fig. 2

The parallelogram index features are determined by choosing the pair of parallel segments P(l1,l2) and Q(l3,l4), which fulfill the condition:

1. *$l1 \approx l2$ and $l3 \approx l4$*
2. *Centre(P)≈Centre(Q)*
3. *$\Theta_1(P) \approx \Theta(Q)$ and $\Theta_1(Q) \approx \Theta(P)$*
4. *$l(P) \approx m(Q)$ and $l(Q) \approx m(P)$*

In the case of ellipses, circles and arcs features, there has to be, within the polygonal contour or polygonal line, sequences of segments which fulfil the following conditions:

1. $\varepsilon_1 < |\Theta_i| < \pi/2 - \varepsilon_1$ ;
2. $\Theta_i * \Theta_{i+1} > 0$ ,

where: $\Theta i$ is the angle between two consecutive segments.

The condition (1) guarantees the smoothness of the curve and condition (2) guarantees its convexity. If the segment sequence has more than 4 segments, the following condition will be checked:

3. $|\Theta_i - \Theta_{i+1}| < \varepsilon_2$ .

If the condition 3 is fulfilled throghout the whole sequence then the line sequence approximates a circle or a circle arc.

If the condition is partially fulfilled then we can say that we have an ellipse or an ellipse arc.

In order to determine the curve parameters the following equation is used:

$$f(x,y)= a_{11}x^2 + 2a_{12}xy +a_{22}y^2 + 2a_{10}x + 2a_{20}y +1 = 0 \qquad (4)$$

The computation of the parameters is done by choosing 5 points from the curve and solving the resulted system of equations.

By reducing the conic to canonic form, the curve attributes are determined:

$x_c$, $y_c$ - conic centre

a,b - axes lengths of the ellipse or,

r - circle radius

$\Theta$ - major axis direction.

## 6. Results

Some results of the intermediate representation generation process from an intensity image of a cube (Fig. 3) are presented below:

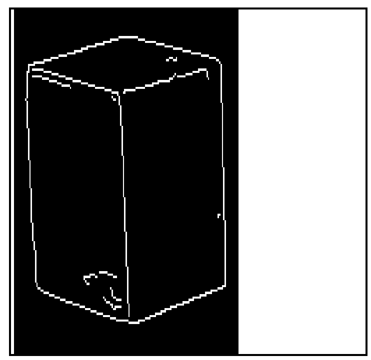

Fig. 3

*line( 8 98.1 176.7 65.5 -13.56 )*
*line( 7 63.5 122.0 94.2 -86.64 )*
*line( 6 95.9 68.1 71.6 -168.83 )*
*line( 5 164.1 67.1 67.4 169.85 )*
*line( 4 199.4 120.5 95.1 92.53 )*
*line( 3 168.2 176.9 68.8 14.91 )*
*line( 2 129.8 136.4 99.2 -84.01 )*
*line( 1 94.8 81.4 60.7 -10.64 )*
*angle( 1, (2, 1), 106.7688 124.648, 87.000, (99.2, 60.7))*
*angle( 2, (3, 2), -98.9268 135.000, 185.711, (68.8, 92.2))*
*angle( 3, (4, 3), 102.4152 201.500, 168.000, (95.1, 68.8))*
*angle( 4, (5, 4), 102.7175 197.305, 73.000, (67.4, 95.1))*
*angle( 5, (2, 8), 70.5175 134.953, 185.562, (99.2, 65.5))*
*junction( 1, (8, 2, 3), Y, (99.2, 68.8, 65.5), (70.5, 98.92), 134.977, 185.633)*
*parallels( 1, (7, 2), 96.5, 129.2, 85.3, 12.3, 96.7, 50.8)*
*parallels( 2, (8, 1), 96.5, 128.6, 12.3, 85.3, 63.5, 77.5)*
*parallelogram( 1, (1, 2), 96.5, 129, 96.7, 63.5, 85.3, 12.3)*

The results format was presented in section 4.

The intermediate representation generator from intensity image is still in development.

## References


1. P.J Besl, R.C. Jain, Three-Dimensional Object Recognition, *Computing Surveys*, Vol. 17, No. 1, March 1985, pp. 75-145.
2. P.J. Besl, R.C. Jain, Invariant Surface Characteristics for 3D Object Recognition in Range Images, *Computer Vision Graphics , and Image Processing*, Vol. 33, No. 1, 1986, pp. 33-80.
3. P.J. Besl, Geometric Modeling and Computer Vision, *General Motors Research Laboratories*, GMR-6248, April 1988, pp. 1-41.

4. R.A. Brooks, Symbolic Reasoning Among 3-D Models and 2-D Images, *Artificial Intelligence*, No. 17, 1981, pp. 285-348.
5. J.B. Burns, A.R. Hanson, E.M. Riseman, Extracting Straight Lines, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 4, Jul 1986, pp. 425-455.
6. R.T.Chin, C.R. Dyer, Model-Based Recognition in Robot Vision, *Computing Surveys*, Vol. 18, No. 1, March, 1986, pp. 67-108.
7. B.A. Draper, R.T. Collins, J. Brolio, A.H. Hanson, E.M. Riseman, The Schema System, *International Journal of Computer Vision*, No. 2, 1989, pp. 209-250.
8. M.A. Eshera, K.S. Fu, An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph-Matching, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 5, September 1986, pp. 604-618.
9. M.A. Fischler, R.C. Bolles, Perceptual Organisation and Curve Partitioning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, January, 1986, pp. 100-105.
10. R. Horaud, "New Methods for Matching 3-D Objects with Single Perspective Views", *IEEE Transactions on Pattern analysis and Machine Intelligence*, Vol. 9,No. 3, May 1987, pp. 401-412.
11. R. Horaud, T. Skordas, "Finding Geometric and Relational Structures in an Image", *Proc. First European Conference on Computer Vision*, Antibes, France, April, 1990, pp. 374-384.
12. A.C. Kak, A.J. Vayda, R.L. Cromwell, W.Y. Kim, C.H. Chen, Knowledge-based robotics, *Int. J. Prod. Res.*, Vol. 26, No. 5, 1988, pp. 707-734.
13. Y. Kuno, Y. Okamoto, Robot Vision Using a Feature Search Strategy Generated from a 3-D Object Model, *IEEE Transactions on Pattern analysis and Machine Intelligence*, Vol. 13, No. 10, October 1991, pp. 1085-1097.
14. H.C. Lie, K.S. Fu, Generating Object Descriptions for Model Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 5, September, 1983, pp. 462-471.
15. S. Nedevschi, C. Goina, Edge Extraction and Contour Closing with Subpixel Accuracy, ACAM, No. 2, 1992.