# 3D Reconstruction of Solid Models from Engineering Orthographic Views using Variational Geometry and Composite Graphs

Miri Weiss-Cohen

Braude Academic College of Engineering, mweiss@ort.org.il

## ABSTRACT

Reconstruction of 3D solid models from 2D orthographic views is a topic that has received much attention in the last two decades. The 2D orthographic views in this work are engineering drawings and the proposed automatic procedure for 3D object reconstruction mimics skilled human intelligence. This is achieved by combining elements of knowledge retrieval, variational geometry, and graph theoretic methods. In particular, the main novelty in the approach is its use of understanding the nature of 2D engineering drawings. This understanding is translated into an actual algorithm by means of topological relations and dimensional scheme analysis, i.e. variational geometry, for each 2D view. The first step is creating minimal graphs representation for the dimensioning schemes of each view. These local graphs are then merged into a global dimensioning scheme of the entire object. The last step of the process is a transformation of the data from the global graphs to 3D solid models.

**Keywords:** CAD, 3D reconstruction, variational geometry, graph theoretic tools.

## 1. INTRODUCTION

Engineering drawings have been traditionally used for designing and communicating object information among designers, customers, subcontractors and quality assurance professionals. Using CAD systems, the process of design is interactive. Both geometric data and annotations are stored in the CAD system database and can be accessed and used for future design, or as input for CAM systems. Numerous mechanical engineering drawings of parts still exist in blue prints form. Significant process in scanning devices and storage technology has made reconstruction of 3D objects from paper drawing a viable research issue. Intelligent knowledge retrieval from the engineering drawing combined with graph representations and search procedures mimics the skilled human intelligence.

A number of approaches have been developed over the past two decades to interpret user-supplied orthographic views. The two main reconstruction approaches are the wireframe – B-rep bottom up approach [4] [6] [8] [9] [10][11], and the volume-solid oriented approach [1] [2][12][13][15].

This paper presents a new approach based on combining elements from knowledge retrieval, interpretation and formations of representations of 3D objects. The stage of constructing a wireframe model and translating it into a 3D Solid model in a bottom-up approach has caused many errors and research issues. This automatic procedure for 3D reconstruction mimics trained human experts, which is achieved by combining elements of knowledge retrieval, variational geometry, and graph theoretic methods. In particular, the main novelty in the approach is its use of understanding the nature of 2D engineering drawings. This understanding is translated into an actual algorithm by means of topological relations and dimensional scheme analysis,

Human experts reconstruct an object's 3D geometry by combining the object's orthographic views in their minds. But even before this procedure, human readers extract vital information from each single view, as follows:

Each view undergoes a mental 'layer separation', in which pure geometric entities (the projected object contours) are distinguished from wires (bars and arcs) belonging to the language layer. The annotation is then further analyzed. Dimension-sets are first aggregated from their primitive components, wire arrowheads and text strings containing numerical values and characteristic symbols. The resulting dimension-sets are then associated with corresponding object contours, or geometry sites. The reader of the drawing then sketches out a gross estimate of the match between each nominal dimension value, conveyed by the text, and the corresponding actual distance or angle between the pair of sites associated with each dimension- set, as seen in the drawing.

In order for useful, knowledge retrieval to be extracted from a drawing, it ought to be viewed as a dual, two-faceted entity composed of several layers [14]. The understanding process is divided into three phases: lexical or early vision, syntactic or intermediate vision, and semantic or high level vision. The lexical phase is preceded by scanning, noise removal, enhancing, thresholding and other preprocesses. Lexical analysis includes recognition of basic elements found in most engineering drawings: bars (straight line segments), arcs, text and arrowheads. At the syntactic level, the primitives are aggregated into semantically correct groups, notably dimension sets. The semantic phase involves obtaining complete 2D views of the object described in the drawing.

## 2. HIGH LEVEL KNOWLEDGE RETRIEVAL (HLKR)

An engineering drawing is a combination of a number (usually three) of orthographic views of an object that show projections of the object's boundary using lines with two continuity value types – solid and dashed – as well as symbolic representations of features (e.g., radius of a hole), representations of machining requirements (e.g., surface finish), dimensions, tolerances and textual annotations. Lexical analysis of the system involves recognition of solid and dashed wires (bars and arcs), while the syntactic and semantic phases determine which wire is a geometry wire and which is an annotation wire. These classified primitives, in particular the geometric ones, serve as input to the 3D reconstruction process [4].

The process of high-level knowledge retrieval includes the following sub process:

- Lexical, syntactic and semantic analysis.
- Loop definition by finding all vertices connecting or intersecting bars and arcs in the 2D orthographic view.
- Finding all possible loops based on the definition of a loop as a minimal closed consecutive set of wires.
- Loop topology and continuity attribute definition.
- Loop content attribute definition and validation.

### 2.1 Definitions

The following definitions are used throughout this work:

- Loop: a minimal set of closed consecutive wires (bars and arcs), where closed means that the starting vertex of the first wire coincides with the ending vertex of the last wire.
- Contour: the ordered collection of the outmost closed chain of wires.

A loop is characterized by three attributes: Topology, Continuity and Contents. The possible values of each one of these attributes are as follows:

*Loop Topology Attribute:* Topology pertains to the planar position of the loop relative to other loops in the view. Possible values are contour, internal, external and intermediate, as defined below:

- Internal loop: a loop in which all wires are inside the area surrounded by the contour or a loop within it, and no wire has a common edge with any wire of another loop.
- External loop: a loop which has at least one wire that coincides with a wire of the contour.
- Intermediate loop: a loop which is neither internal nor external.

*Loop Continuity Attribute:* Any wire of the loop can be solid (continuous) or dashed (discontinuous). This gives rise to three loop continuity values: solid, dashed and mixed, as follows:

- Solid loop: a loop consisting of solid wires only.
- Dashed loop: a loop consisting of dashed wires only.
- Mixed loop: a loop consisting a mixture of at least one solid wire and one dashed wire.

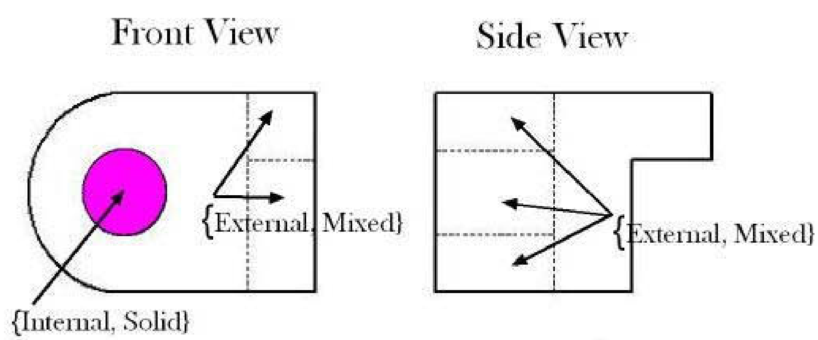


Fig. 1: High-level knowledge retrieval for the side and front view of an engineering drawing.

*Loop Content Attribute*: Loop content is an attribute that pertains to the semantics of what the loop symbolizes, i.e., the presence or lack of matter. The possible values of content are matter, depression, protrusion, and undefined, as follows:

- Matter loop: a loop enclosing solid matter of the 3D object.
- Depression loop: a loop enclosing a through (piercing) hole or lowered surface, i.e., a surface located beneath the matter surrounding loop surface level in the 3D object.
- Protrusion loop: a loop enclosing a heightened surface, i.e., located above the surrounding matter loop surface level in the 3D object.
- Undefined loop: a loop whose content is not defined.

For the Front View and Side View in Figure 1, the topology and continuity attributes are as follows are determined. In the Side view all loops are {External, Mixed}, and in the Front View two loops are {External, Mixed}, and the loop colored in yellow is {Internal, Solid}.

## 3. TOPOLOGICAL RELATIONS AND DIMENSIONING ANALYSIS USING VARIATIONAL GEOMETRY

The input to this stage is a dimensioned 2D view, which goes through a constraint evaluation process resulting in a 2D view constraint set. Dimensions define geometric constraints, such as distance between two points, distance between a point and a bar, and an angle between two bars. Spatial relations define topological constraints such as tangency, parallelism, and perpendicularity. The constraints extracted from each 2D view represent relations among explicit and implicit characteristics. Each dimension is formulated as a constraint. There are two kinds of constraints, one, defined by a single equation, and a compound constraint, which require two or more equations.

Each constraint equation is a function of points in the geometric dimension scheme. Equation *i*, denoted $f_i$, is formulated as follows:

$$f_i = \{x_1, y_1, x_2, y_2, ..., x_n, y_n\} \quad (1)$$

Where *n* denotes the number of points constraints by the geometric entity.

For the complete 2D view, a set of constraints, denoted as F, is given as follows:

$$F = \{f_1, f_2, ..., f_m\} \quad (2)$$

As an Example, a distance from a point to a line is presented. To constrain the distance D between a point $P_a$ and line $P_bP_c$ two vectors must be defined: a unit vector $\hat{U}$ from $P_b$ to $P_c$ and a vector $\overline{V}$ from $P_b$ $to$ $P_a$/ the distance D is a cross product $\hat{U} \times \overline{V}$ .

$$\hat{U} = \frac{x_c - x_b}{|P_bP_c|}\hat{i} + \frac{y_c - y_b}{|P_bP_c|}\hat{j} = U_x\hat{i} + U_y\hat{j} \quad (3)$$

and

$$\overline{V} = (x_b - x_a)\hat{i} + (y_b - y_a)\hat{j} \quad (4)$$

where *i* and *j* are unit vectors in the x and y directions, respectively.

The point-to-line constraint if formalized in equation 5 is

$$f_1 = U_x(y_b - y_a) - U_y(x_b - x_a) - D = 0 \quad (5)$$

The system constructs a knowledge base of variational geometry rules [7] for constraining the dimensioning scheme and the relations between the geometry sites in the view. In the rule base, constraining the dimensioning scheme is done by positioning a selected point, called *anchor point*, at the origin in order to prevent solid body translation. All points are defined relative to this anchor point. To prevent solid body rotation, a bar is defined to be horizontal, i.e., parallel to the x axis. For the dimensioning and the constraint set to be valid, the Jacobian constraint matrix should meet two requirements. First, the number of constraints must be equal to twice the number of vertices, and second, the rank of the matrix must equal the number of constraints. Meeting these requirements indicates that the matrix is non-singular and hence there is neither redundancy nor lack of dimensions and definitions of the constraints. The constraint set *F*{Front}, for the Front view in Figure 1 is formalized in equation set 6.

$$
\begin{array}{lll}
f1: & (x_2 - x_1)^2 + (z_2 - z_1)^2 - c^2 = 0 & \text{Euclidian Distance} \\
f2: & (x_3 - x_2)^2 + (z_3 - z_2)^2 - d^2 = 0 & \text{Euclidian Distance} \\
f3: & (x_1 - x_8)^2 + (z_1 - z_8)^2 - a^2 = 0 & \text{Euclidian Distance} \\
f4: & (x_8 - x_6)^2 + (z_8 - z_6)^2 - b^2 = 0 & \text{Euclidian Distance} \\
f5: & (x_9 - x_{10})^2 + (z_9 - z_{10})^2 - {R_2}^2 = 0 & \text{Euclidian Distance} \\
f6: & (x_9 - x_4)^2 + (z_9 - z_4)^2 - {R_1}^2 = 0 & \text{Euclidian Distance} \\
f7: & (x_1 - x_8)(x_2 - x_1) + (z_1 - z_8)(z_2 - z_1) = 0 & \text{Perpendicularity} \\
f8: & (x_2 - x_3)(x_9 - x_3) + (z_2 - z_3)(z_0 - z_3) = 0 & \text{Perpendicularity} \\
f9: & (x_1 - x_2)(x_7 - x_2) + (z_1 - z_2)(z_7 - z_2) = 0 & \text{Perpendicularity} \\
f10: & (x_5 - x_6)(x_8 - x_6) + (z_5 - z_6)(z_8 - z_6) = 0 & \text{Perpendicularity} \\
f11: & (x_5 - x_4)(x_9 - x_4) + (z_5 - z_4)(z_9 - z_4) = 0 & \text{Perpendicularity} \\
f14: & (x_1 - x_8)(x_7 - x_8) + (z_1 - z_8)(z_7 - z_8) = 0 & \text{Perpendicularity} \\
f12: & x_5 - x_2 = 0 \quad \text{Collinear Points} & f13: \; x_6 - x_1 = 0 \quad \text{Collinear Points} \\
f15: & x_4 - x_3 = 0 \quad \text{Collinear Points} & f16: \; x_{10} - x_3 = 0 \quad \text{Collinear Points} \\
f17: & x_9 - x_3 = 0 \quad \text{Collinear Points} & \\
f18: & x_1 = 0 \quad \text{Ancor Point} & f19: \; z_1 = 0 \quad \text{Ancor Point} \\
f20: & z_2 - z_1 = 0 \quad \text{Orientation} &
\end{array}
\qquad (6)
$$

Following the constraint definitions and proper dimensioning check for each 2D orthographic view, we combine the constraints into a minimal graph. The motivation for this operation is to find the minimal set of relations that fully represent the 2D view.

## 4. GRAPH REPRESENTATION AND DEPENDENCIES MATCHING 2D VIEW CONSTRAINT SET

The graph representation of a constraint expresses the relationship and connections among parameters. Moreover, a graph representation of a constraint-set is a declarative structure which expresses the existence of relations among the parameters of more than one constraint. The motivation of this conversion is to find the minimal set of relations that fully represent the 2D view. The 2D pair representation provides a compact quantitative and qualitative mapping of the relations among the parameters for each view. The terminology of graph representations is found in detail in [5].

The process of converting the constraint-set into a 2D pair graph has the following stages:

- Representing the constraint set as a full undirected graph. The nodes represent the parameters and the arcs represent the existence of the constraint between the parameters. The arcs are labeled according to the constraint.
- Obtaining a bipartite graph from the full undirected graph and finding a maximum matching for the bipartite graph. This is done by applying a matching process yielding a mach set. The result maximum matching for the bipartite graph for the Front view is represented in Figure 2.
- Transforming the full undirected graph into a minimal graph using dependencies found in the previous step. Using the results of the matching process, the undirected graph is transformed into a minimal graph by the following rules: For a matching pair (v,fi) the undirected graph is modified such that all arcs labeled other than fi and incident on v are removed.
- Converting the minimal graph into a pair graph. Using the minimal a pair graph is constructed by combining two single nodes to one pair node. For nodes labeled xi and yi, for example, i=j, a new pair node labeled {xi,yi}, is defined. All other arcs are removed. The result 2D pair graph for the Front view is represented in Figure 2.

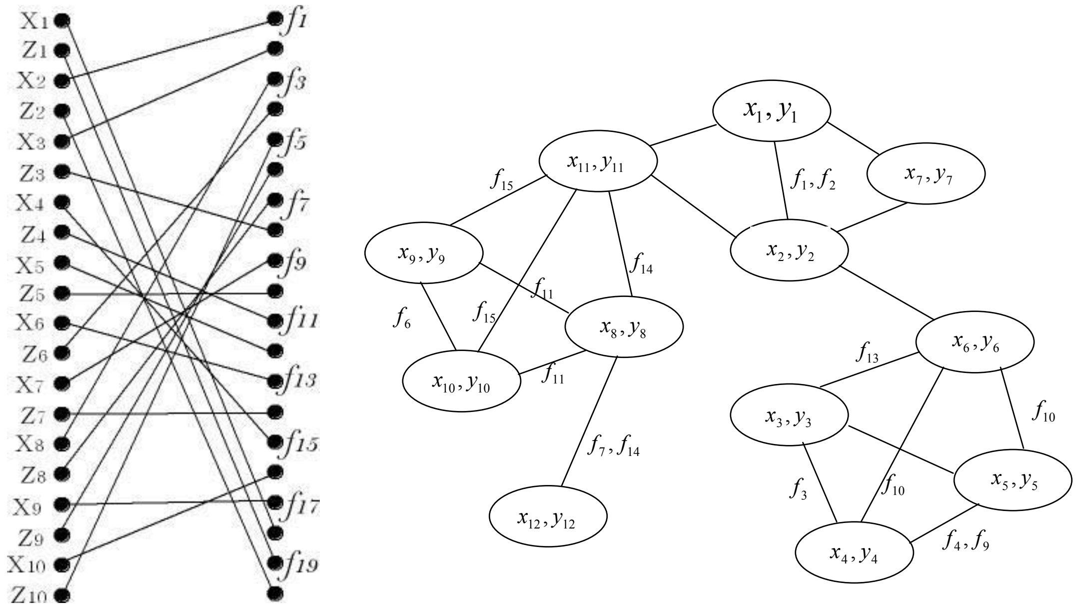


Fig. 2: Bipartite graph (left) and 2D pair graph for the front view of Fig. 1.

## 5. 3D RECONSTRUCTION APPROACH

Once the 2D pair graph for each orthographic view is obtained, a 3D object can be reconstructed. This process consists of three major procedures:

1. Initial matching,
2. Complete matching
3. Graph-to-object conversion.

The processes of initial matching and complete matching use the set of separate 2D pair graphs for each 2D view. These graphs are analyzed for matching and merging conditions. Figure 3 schematically illustrates the starting point, in which each 2D view has an independent pair graph representation (left). The right scheme illustrates a complete graph generated by adding new edges (the dashed lines) which connect vertices in the various 2D pair graphs.

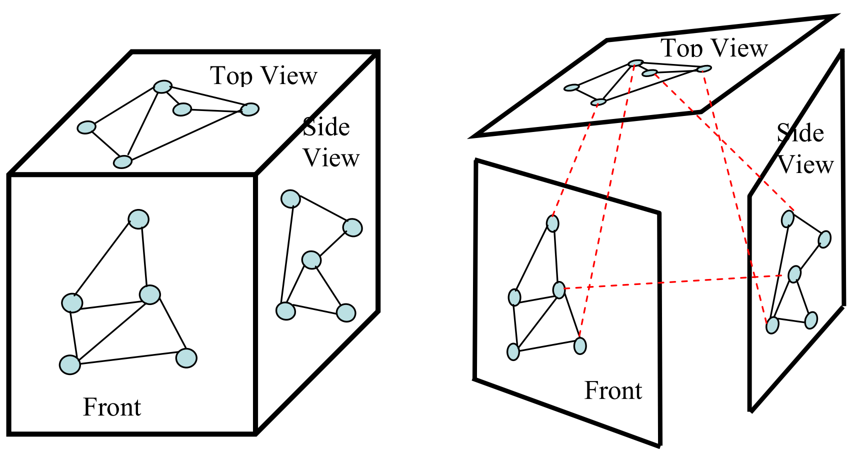


Fig. 3: A schematic illustration of the composite 3D graph representation.

### 5.1 Initial Matching

Matching the first 3D vertex is based on searching each one of the 2D projections for different plane representations of the vertex. These vertices from the projections are matched simultaneously in the representation graphs. The matching rules are described in Table 1

| Plane | View | Each vertex has different values | Two or more vertices have the same value |
|---|---|---|---|
| (X,Y) | Top | Largest value of xi | Largest value of xi and smallest value of yi |
| (X,Z) | Front | Largest value of xi | Largest value of xi and smallest value of zi |
| (Y,Z) | Side | Smallest value of zi | Smallest value of zi and smallest value of yi |

Table 1: Matching rules for initial matching.

### 5.2 Complete Matching

The initial 3D vertex serves as a starting point for the matching completion. Each link represents both the connecting related parameters and the nature of the link (e.g., parallelism, dimensions, etc.). The procedure of complete matching starts from the initial point on the initial composite graph and proceeds by searching the connected edges of the initial 3D vertex, while checking two criteria: connectivity and constraint type. The process of complete matching has the following steps:

Step1: Starting from the initial matching of the 3D vertex, an arbitrary pair graph related to a view is chosen. The chosen pair graph is the basis for the complete matching.

Step 2: On the chosen pair graph the initial vertex node is the starting point of the search and linking process. All edges connected to the initial vertex node are checked by constraint type. Edges having the constraint type of Euclidian distance between two points are chosen first, followed by a matching criterion of topological connections represented in the graph or equality of coordinate values in the relevant planes. The complete matching process yields a composite graph, where all 3D vertices in the 3D object are represented by the triplet of two nodes and an edge linking them. Figure 4 is a partial illustration of the composite graph

.

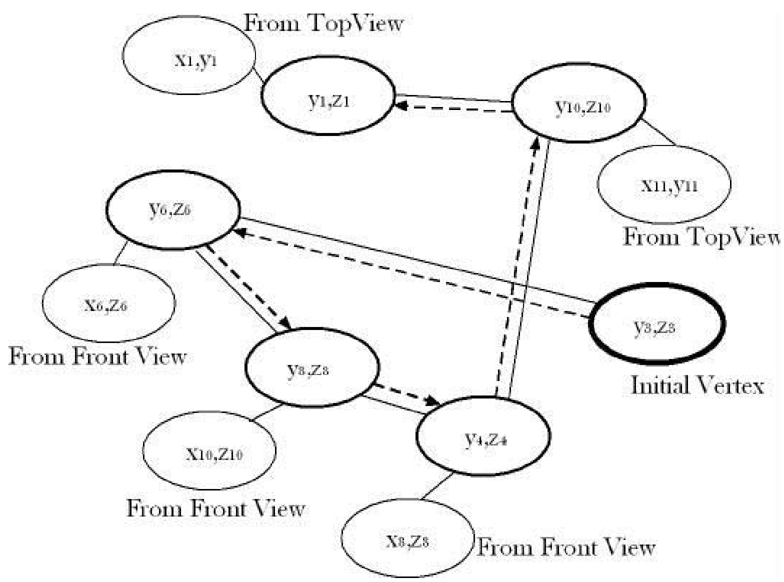


Fig. 4: Illustration of the partial composite graph.

### 5.3 Graph to Object Conversion

The complete composite graph is converted into a 3D object by translating the links, connections and constraint types into geometry and topology. The 3D object is represented by a boundary representation (B-rep). This process is based on using graph theory tools, search models and heuristics for retrieving information from the composite graph and then using the high level information understanding of the 2D views. The Chinese Postman Problem (CPP) algorithm is used for arc routing of all vertices in the graph and traversing all edges by the criterion of minimal total distance. The CPP algorithm is described in detail in Evans [5].

Implementation of the CPP assists in defining tours in which each triplet (two nodes and a connecting edge) is translated into the geometry of the 3D object, depending on the constraint type and value. This procedure is comprised of the following steps:

*Step 1:* The first tour is defined by starting from the initial matching, a 3D vertex represented in the composite graph and the contour loop defined in the high level understanding. This step has the following sub-processes:

- Choose the initial 3D vertex representation for the selected view and label it 'selected node'.
- Check the edges connected to the 'selected node' for a tuple (edge or node) so that the node represents a vertex on the contour loop. If more than one node is found, select one randomly.
- Traversing to the node found in the selected view, check the composite graph for connecting edges in the remaining views.
- Translate the vertex node representation and relevant constraint type represented on the edge to the geometry and topology. In cases where more than one constraint is represented on the edge, each of them is checked for feasibility by checking the loop content value found in the high level understanding stage.
- Implement sub-processes 2-4 until the first 'selected node' is reached for the second time.

*Step 2:* Starting from the last node representation found in Step 1 and the resulting loops from the HLKR process, define a new tour.

*Stopping condition:* Translation from the 3D composite graph is completed when all loops found in the HLKR process have been covered.

Figure 5 is a partial illustration of the conversion process.

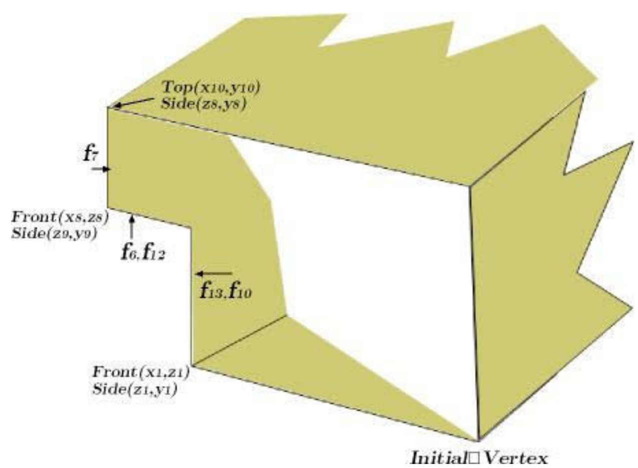


Fig. 5: Partial illustration of the graph to object conversion process.

The 3D boundary models represent a solid indirectly through a representation of its bounding surface. Boundary models as a representation of a solid object by dividing its surface into a collection of faces, the 3D object reconstruction from three engineering drawing representing orthographic views yields a Boundary representation of a solid.

Figures 6 and 7 are examples of the results of the reconstruction process. Figure 6 shows the front, side and top views of an engineering drawing with the results of the HLKR.

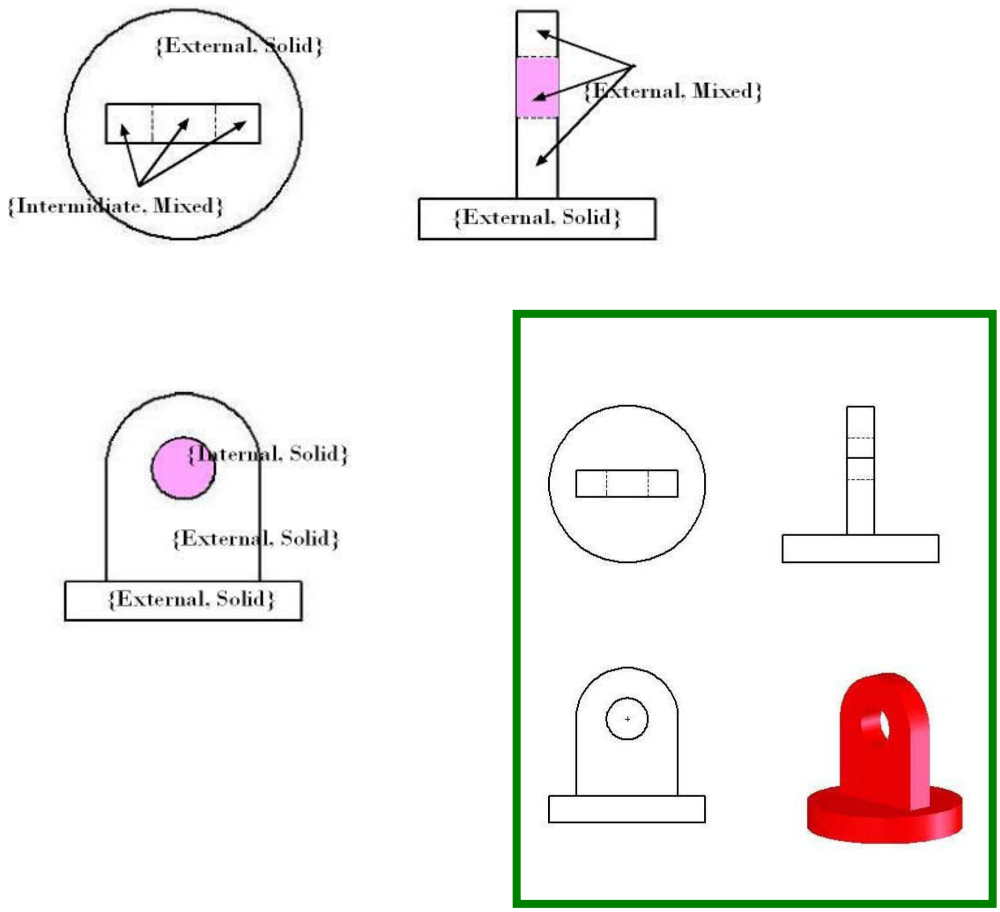


Fig. 6: Orthographic 2D views {Front, Side and Top} with the HLKR and the 3D solid after the automatic reconstruction.

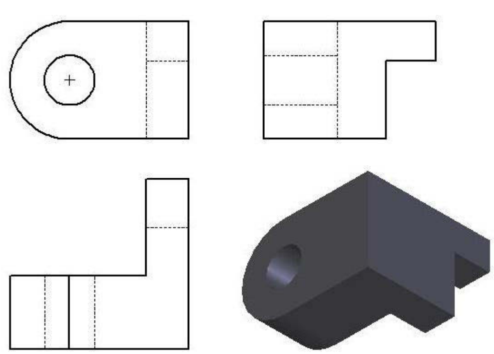

Fig. 7: Orthographic 2D views {Top, Side and Front} and the 3D Solid after the automatic reconstruction.

## 6. CONCLUSIONS

This paper proposes an approach for solving the problem of 3D object reconstruction from 2D orthographic views that combines elements from the experienced human mind. The approach begins with high level intelligent knowledge retrieval and goes on to use the relationships among bars, arcs, facets etc. It concludes by combining these elements into a visualized 3D object. The process is implemented by using the HLKR process followed by formulation by means of variational geometry and representation and translation of composite graphs. These elements are amenable to automation, and the complex procedure described in this work therefore serves as a means for reliable and accurate 3D reconstruction of solids.

## 7. REFERENCES


[1] Aldefeld, B.,: On automatic reconstruction of 3D structures from 2D representations, CAD, 15, 1983.

[2] Chen, K.-Z.; Feng, X.-A.: Holo-extraction of information from paper drawing for 3D reconstruction, Computer Aided Design, 34, 2002, 665-677.

[3] Dori, D.; Tombre, K.: From engineering drawing to 3D CAD models: are we ready now?, CAD, 24, 1995, 243-254.

[4] Dori, D.; Weiss, M.: A scheme for 3D object reconstruction from dimensioned orthographic views, Engineering Applications of Artificial Intelligence, 9, 1996, 52-64.

[5] Evans, R. J.; Mineika, E.: Optimization Algorithms for Network and graphs. Marcel Dekker, N.Y., 1992.

[6] Idesawa, M.: A system to generate a solid figure from three view, Bulletin of the JSME, 16(92), 1973, 216-224.

[7] Lin, V. C.; Light, R. A.; Gossard, D. C.: Variational Geometry in Computer Aided Design, Computer Graphics, 14, 1981, 171–177.

[8] Liu, S.; Hu, S.; Chen, Y.; Sun, J.: Reconstruction of curved solids from engineering drawing, Computer-Aided Design, 33, 2001, 1059-1072.

[9] Markovsky, G.; Wesley, M. A: Fleshing out wireframes, IBM J. of Research and Development, 24(5), 1980, 582-597.

[10] Martson, R. E.; Kuo, M. H.: Reconstruction of 3D object from three orthographic projections using decision chaining method, IAPR MVA'94, Workshop on machine vision applications, Japan, Dec. 1994.

[11] Nagasami, V.; Noshir, A. L.: Reconstructing of three dimensional object using a knowledge base environment, Engineering with Computers, 7, 1991, 23-35.

[12] Liu, S.-X.; Hu, S.-M.: Reconstruction of curved solids from engineering drawings, Computer Aided Design, 33, 2001, 1059-1072.

[13] Shum, S. P.; Lau, W. S.; Yuen, M. F.; Yu, K. M.: Solid reconstruction from orthographic opaque views using incremental extrusion, Computers & Graphics, 21, 1997, 787-900.

[14] Tombre, K.: Technical drawing recognition and understanding: from pixels to semantics, MVA'92 IAPR Workshop on Machine Vision Applications, Tokyo, 1992.

[15] Weidong Geng; Jingbin Wang: Embedding visual cognition in 3D reconstruction from multi-view engineering drawing, Computer Aided Design, 34, 2002, 321-336.