

# Dimensioning Analysis

## Toward Automatic Understanding of Engineering Drawings

**A** machine drawing consists of a description of one or more principal orthogonal views (projections) of an object. These projections can be used to reconstruct the 3D structure of the object in a variety of ways. The problem of solid reconstruction from engineering drawings deals with a subclass of the class of multiview line drawings, where the object world is not restricted to a finite set of prototypes. The interpretation of multiview line drawings has been investigated quite intensively [1, 7, 8, 12, 13, 14, 15, 16, 18]. For specific wireframe projections this is known as the "fleshing out projections" problem [19]. Gigus and Malik [8] have developed an algorithm for computing the aspect graph for line drawings of polyhedral objects. They provide several references for surveys of current approaches for solving the 3D object recognition problem. Sugihara [18] presents a computational mechanism for the interpretation of line drawings by enabling a machine to reconstruct 3D object structures from their pictures drawn on a 2D plane. The objects he considers are polyhedra, and the line drawings are single-view objects bounded by planar faces. One potential application suggested by Sugihara for his mechanism is flexible human-machine communication. Since it is tedious work for designers to convert their thoughts into numerical forms, all they will have to do is draw pictures of their designed objects and give a small number of additional data, such as lengths of edges and angles between faces. The question of how this data is to be provided is not discussed.

The subclass of line drawings we are concerned with, mechanical engineering drawings, usually consists of top, front and side views. Various approaches to the reconstruction problem have been published in the literature. Broadly, they can all be viewed as applications of constraints to various 2D entities. Wesley and Markowski [19] have developed an algorithm to find all solid polyhedral objects with a given set of two dimensional projections. These projections may contain depth information in the form of dashed and solid lines, may represent cross sections, and may be overall or detailed views. Haralick and Queeney [9] apply constraints to the 2D primitive entities, which are the lines on the 2D projections. Aldefeld [1] starts from closed loops of lines and applies rules or constraints to them. This approach, although less general, may frequently be practical. Preiss [12] also casts the problem into one of constraint propagation. His control structure arrives either at a consistent constraint net or at a

conclusion that there is no consistent net, thus enabling a certain validation of the set of 2D projections. This algorithm handles also cylindrical faces, which are not treated in [19].

As noted [19], quite apart from the mathematical interest of these "fleshing out projections" algorithms, they provide a basis for automatic conversion of a set of 2D projections typically found in mechanical engineering drawings into solid volumetric representations of objects. There is, however, a major problem with these and other similar algorithms that currently limits their applicability. The algorithms assume that there exists a well-defined, noise-free set of 2D projections, readily available to be used as input. Thus they overlook the presence of *annotation* entities in practically all engineering drawings. These entities are aimed at providing the drawing reader with information about the object which cannot be expressed by its *geometry* entities. While geometry lines describe contours of facets in the object's projections, annotation adds such information as dimensions and tolerances, production specifications, administrative instructions and so forth. Reconstruction algorithms perceive this type of information as 'noise.' While obtaining a 'noise-free' set of projections may be relatively easy for models of objects whose projections were originally created using a CAD/CAM system, it is far from being trivial when the source is a paper drawing, produced either manually or by a computer driven plotter. Currently available CAD/CAM systems lack the ability to automatically incorporate manually prepared machine drawings or paper drawings, plotted as outputs of other CAD/CAM systems, into their database. A major reason for this is that it is difficult to achieve a capability of distinguishing between, and separation of, geometry and annotation entities, which is basic to any automated system aimed at understanding engineering drawings. A pre-

processing step is required in order to separate the two types of entities and to use the annotation entities to enhance the information expressed by the geometric ones before removing them. It is the scope of a Machine Drawing Understanding System (MDUS) to carry out this task [4, 5, 6].

Although a drawing usually consists of several (normally three) views, to simplify the analysis we restrict the discussion to 2½D objects (objects that have constant width), for which a drawing that consists of one view is sufficient. The view thus constitutes the top of a three-level hierarchy. The middle level is occupied by dimension-sets, while their components are at the bottom of the hierarchy. The relationships between the bottom and middle levels have been studied in [3, 6]. This work is concerned with the relationship between the middle and the top levels (i.e., how dimension-sets are arranged so as to meet the demands of drafting standards).

Problems with product information exchange are by no means limited to transfer of information from manually prepared drawings to interactive graphic CAD/CAM systems. Exchange of information among various types of such systems has been addressed by many researchers and standards organizations. The Initial Graphics Exchange Specification (IGES) [17] is a widely accepted neutral file format that establishes information structures to be used for the digital representation and communication of product definition data used by various CAD/CAM systems. Initiated in late 1979, IGES is a mature mechanism that provides a stable, standardized, vendor independent format to aid in the management and use of data from CAD/CAM systems.

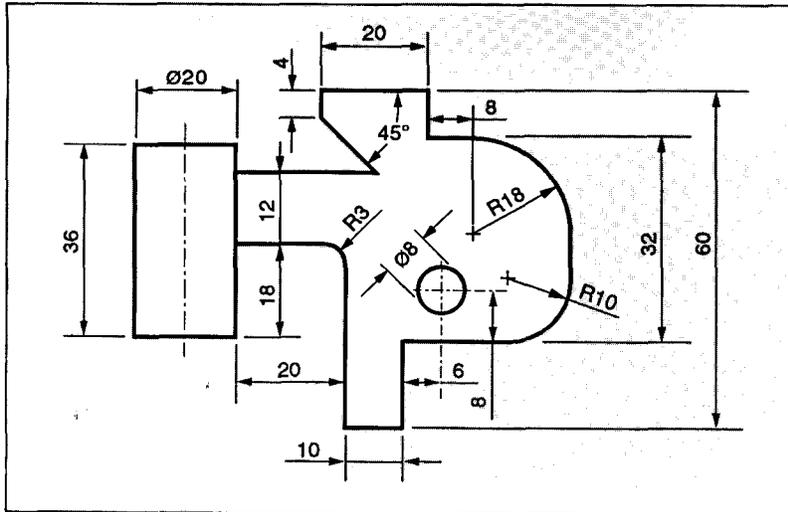
### Geometry, Annotation, and Dimensioning

Dimensioning constitutes the core of annotation entities. Adopting IGES convention, the fundamental

unit of information is the *entity*. The **DRAWING** entity (IGES no. 404) specifies a drawing as a "collection of annotation entities and views which, together, constitute a single representation of a part, in the sense that an engineering drawing constitutes a single representation of a part in standard drafting practice" [17]. It allows a set of views to be identified and arranged for human presentation. The **VIEW** entity (IGES no. 410) provides characteristics associated with individual views.

Dimensioning in engineering drawings provides an exact definition of the geometry approximated by the geometry entities. Therefore, recognition of dimensions is a key component of MDUS. This assertion is supported by the fact that of the 15 annotation entities handled by IGES, 10 are associated with dimensioning. As noted in [10], annotation in general, and dimensioning in particular, is an extremely tedious area of data exchange. The main difficulty is that geometry and annotation entities look essentially the same, as demonstrated in Figure 1, which is a view of a typical engineering drawing annotated using ISO standard. Both geometry and annotation use the same basic primitives: the **bar** (straight line segment) and the **arc**. Even the widths of these primitives in the two entity types are frequently the same. Moreover, these two types of entities are usually interleaved in a manner that makes the task of separating them even more difficult. Under such circumstances, pattern classification based on any feature space is practically impossible. It can only be used, and indeed is used [5], to detect the two object-dissimilar entities *arrowhead* and *text*. These serve as anchors that enable the initiation of parsing.

A **dimension-set** is a set of entities that denote the measure (length or angle) between two geometry entities. A set of dimension-sets is used to accurately and completely define the geometry of the view. A **type** of a dimension-set defines its



type name	symbol	function (IGES entity no.)	example
linear	L	Distance between two geometry entities (216)	
angular	A	Angle between two lines or planes (202)	
diameter	D	Diameter of a top-viewed body of revolution (206)	
ordinate	O	Distance from datum line to reference line (218)	
radius	R	Radius of a circle or an arc (222)	
point	P	General note referring to a specified point (220)	

function. Several types of dimension-sets appear in Figure 1. The group of types of dimension-sets that denote length is termed **longitudinal**, while the type of those denoting angle is termed **angular** (IGES entity type number 202). The longitudinal group includes the **linear** (type number 216), **diameter** (206), **ordinate** (218), **radius** (222), and **point** (220). Figure 2 lists and exemplifies the six various types handled by IGES. The primitives that comprise the dimension-set are termed **components**. To resemble the IGES terminology, they are called leader (IGES entity number 214), witness (IGES entity number 106—Copious Data, Form Number 40), text (IGES entity number 212—General Note), and geometry (IGES entity numbers 100—Circular Arc, 110—Line, and 116—Point). A syntactic approach has been proposed [3, 6] that makes use of the spatial relationship among dimensioning annotation entities and formulates them as a 2D web grammar. The grammar can be used [11] to parse dimensioning entities (components) which constitute the bulk of annotation entities, and assemble dimension-sets from them. Thus it is possible to separate the geometry entities from the annotation entities to which they refer into two *layers*: the geometry layer and the annotation layer. The declarative information represented by the dimension-sets in the annotation layer enhances the spatial information represented by the geometry entities in the geometry level.

### Implicit Conventions and Dimensions

Drafting standards (mainly ISO and ANSI) are the primary source of drafting rules and conventions. Some conventions in engineering

**Figure 1.** A view of an engineering drawing with ISO annotation

**Figure 2.** The six types of dimension-sets handled by IGES

drawings are so inherent that no standard even bothers to indicate them, and they are therefore referred to as **implicit conventions**. Following is a list of the most prominent implicit conventions and their implications.

- **Bar parallelism:** A dimension between two bars can be specified if and only if the bars are parallel. Conversely, if there exists a dimension-set between two bars, then they are parallel.

- **Bar normality:** If an angle between two adjacent bars is not specified, then the bars are reciprocally normal. Conversely, if the angle between two adjacent bars is a right angle it need not be specified. This convention stems from the fact that right angles are so frequent in machine drawings, that  $90^\circ$  is implicitly accepted as the default angle, and is very seldom specified explicitly.

- **Arc normality:** If the angle of an arc is not specified, then it is a right angle. Conversely, if an angle of an arc is  $90^\circ$ , it need not be specified. Rounded corners and fillets make use of this implicit convention.

- **Bar colinearity:** If two bars or more are colinear (i.e., lie on the same line), then it is not necessary to denote the distance from a bar which is not colinear to more than one of the colinear bars. Conversely, if there is no dimension-set that relates two bars which seem colinear, then these two bars are indeed colinear. The implicit assumption here is that between two colinear lines there should be a dimension-set which is not depicted, whose text value is zero. However, it is not forbidden to denote the distance from any bar to more than one colinear bar, so when ambiguity may arise (e.g., big gap between the two colinear bars) it does make sense to ignore this implicit convention.

- **Bar-arc tangentiality:** If a bar is tangent to an arc to which it is connected, there is no need to denote the distance between the bar and the center of the arc. Conversely, if the distance between these two ele-

ments is not denoted, then the bar is tangent to the arc.

Figure 3 demonstrates the difference between dimensioning that applies these implicit conventions to a 2D object (left) vs. theoretic dimensioning of the same object that ignores these conventions (right). The former dimensioning is evidently superior to the latter, as it is not cluttered by details that seem to be obvious, leaving more blank area to be filled with relevant annotation.

### Proper Dimensioning and Normalons

Based on both explicit and implicit conventions of the major drafting standards, a view of an object is **properly dimensioned** if and only if its dimensioning completely and nonredundantly defines its geometry. The main motivation for the proper dimensioning requirement by the standards is that it compels the designer to specify those product dimensions that are the most important from a functional point of view. Proper dimensioning thus minimizes the number of specified dimensions, which, in turn, reduces the possible occurrence of contradictions, especially when tolerances are involved. Two exceptions permitted by the main standards occur when either the dimension is considered a 'reference' dimension, in which case the text is accompanied by the letters REF, or the dimension is considered an 'auxiliary' dimension (i.e., it facilitates the computation of other dimensions), in which case it should be enclosed with parentheses. These exceptions can be taken care of if the content of the text (presence of parenthesis and REF, respectively) is considered. To keep the discussion from getting complicated, though, these exceptions are assumed not be applied in drawings discussed in the following analysis.

Two properties that emerge directly from the definition of proper dimensioning are:

- (1) **Completeness:** The location

and orientation of each geometry entity is defined relative to all the other geometry entities. Completeness ensures that the view is not underdimensioned.

- (2) **Nonredundancy:** No dimension of the object's projection can be determined in more than one way. Nonredundancy ensures that the view is not overdimensioned.

We define a **normalon** as a polygon having the property that the angle between any two of its adjacent sides is  $90^\circ$ . The definition of normalon is motivated by the bar normality implicit convention and the fact that right angles are so widespread in drawings. Many contours of object projections have the shape of a normalon. Many other contours, which are not strictly normalons, can be decomposed into a normalon and one or more other segments, which contain arcs and nonhorizontal or nonvertical sides (e.g., Figure 3).

A **redundant** normalon is a normalon which has at least one pair of colinear bars. To reduce the complexity that arises because of the bar colinearity implicit convention, we restrict the discussion to nonredundant normalons. Unless otherwise specified, the term 'normalon' will refer to a nonredundant normalon. The polygon in Figure 4 is an example of a nonredundant normalon. The completeness property of proper dimensioning implies that it should be possible to tell the distance from any bar to each of the other bars of a properly dimensioned normalon.

Since by the bar-parallelism implicit convention we should only consider distances among parallel bars, we divide the set of bars in a normalon into two subsets, called **bar-sets:** the **horizontal** bar-set and the **vertical** bar-set. The choice of names for the two bar-sets is arbitrary and therefore interchangeable; in any case, all the bars within each bar-set are parallel, and each is perpendicular to each bar of the other bar-set.

**Sweeping** is an operation that changes the geometry of a nor-

malon. A **decremental sweep** is defined as sweeping one of the horizontal bars parallel to itself such that the swept area is a rectangle, until the bar touches any other horizontal bar for the first time. An **incremental sweep** consists of the following two steps:

- (1) Divide a horizontal bar into two bars.
- (2) Sweep one of the newly formed bars parallel to itself such that the swept area is a rectangle until it touches another horizontal bar for the first time.

Using decremental sweep we can

tessellate any normalon into rectangles by applying the following algorithm.

**Algorithm 1: Tessellation of a normalon into rectangles**

Initially the original normalon is the only one in *stack*.

While *stack* is not empty do

begin

Let  $N$  be the normalon on top of *stack*.

Pop  $N$  off of *stack*.

Let  $H$  be the topmost and leftmost horizontal bar of  $N$ .

Perform a decremental sweep of  $H$ .

Append the swept rectangle  $R$  to *rectangle-list*.

Push all the normalons created by subtracting  $R$  from  $N$  onto *stack*.

end

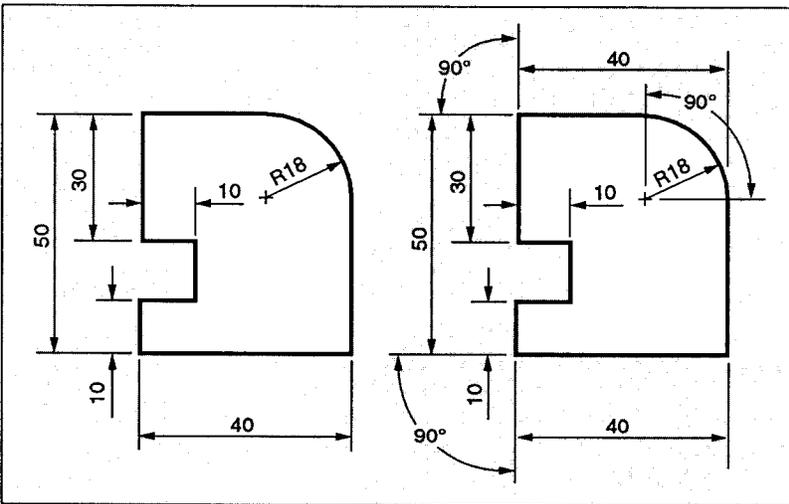
At the end of the execution, *rectangle-list* holds the complete list of rectangles into which the normalon was tessellated. Using this algorithm we prove the following theorem.

**THEOREM 1. (Tessellation Theorem):** Any normalon with  $r$  horizontal bars can be tessellated into  $r - 1$  rectangles by a series of  $r - 1$  decremental sweeps.

**Proof:** Any normalon can be tessellated into rectangles by a series of decremental sweeps by applying algorithm 1. We now show that the number of rectangles in *rectangle-list* at the end of the tessellation is  $r - 1$ .

Let  $p$  be the number of horizontal bars whose two edges lie along the sweeping bar  $H$  when the decremental sweeping is completed, such that no edge coincides with any edge of  $H$ . In other words,  $p$  is the number of bars that 'float' along  $H$ .  $p$  such bars cause  $p + 1$  normalons to be added to *stack* at any iteration. (If the normalon is redundant then  $p$  can be at most 1). Let  $k$  be the number of horizontal bars whose two edges coincide with the two edges of the sweeping bar  $H$ .  $k$  can be either 0 or 1.  $k = 1$  occurs exactly once for each normalon, in the last iteration, when the swept normalon is a rectangle, and just before it 'disappears'. The two events  $k = 1$  and  $p > 0$  are mutually exclusive by definition. Hence, at least one of either  $p$  or  $k$  must be 0 at any iteration.

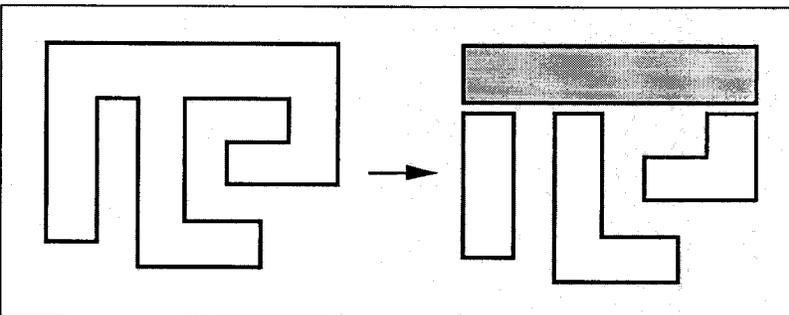
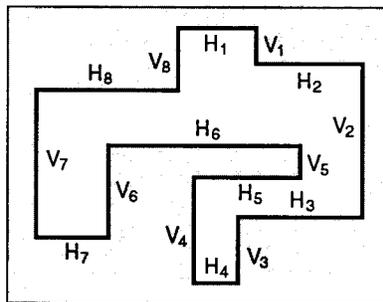
The number of normalons  $m$  pushed onto *stack* at any single iteration is  $m = p + 1 - k$ . Figure 5 demonstrates a situation where  $p = 2$  and  $k = 0$ , such that  $m = 3$ . The right hand side of Figure 5 shows the 3 normalons that were pushed onto *stack* after one iteration. The rectangle added to *rectangle-list* is shaded.



**Figure 3.** The effect of implicit conventions on dimensioning. Left: implicit conventions applied. Right: implicit conventions not applied.

**Figure 4.** A nonredundant normalon of rank 8,  $N_8$

**Figure 5.** An example of one iteration of the tessellation algorithm



Let  $n_i$  be the total number of horizontal bars in all the normalons in the stack at the end of iteration  $i$ , and let  $\Delta n_i = n_{i-1} - n_i$ , i.e. the difference between the number of horizontal bars in the previous and the current iteration. We show that  $\Delta n_i \in \{0, 1, 2\}$  as follows:

$\Delta n_i = 2$  if  $k = 1$  (and hence  $p = 0$ ) because a rectangle disappears from *stack* but no new normalon is pushed onto *stack*, causing a net decrease of 2 horizontal bars.

$\Delta n_i = 1$  if  $p = k = 0$ , that is, when one normalon is pushed onto *stack*. The sweeping horizontal bar  $H$  merges with the first horizontal bar it meets, reducing  $n_i$  by 1.

$\Delta n_i = 0$  if  $p \geq 1$  (and hence  $k = 0$ ) because  $p + 1$  horizontal bars disappear: the one being swept,  $H$ , and the  $p$  which 'floated' along  $H$  when  $H$  coincided with these  $p$  horizontal bars. But at the same iteration  $p + 1$  new horizontal bars were also created, each becoming the topmost bar of one of the  $p + 1$  new normalons pushed onto *stack*.

For each iteration in which  $\Delta n_i = 0$  (called " $\Delta n_i = 0$  type iteration"), there is exactly one  $\Delta n_i = 2$  type iteration, because each new normalon pushed onto *stack* eventually becomes a rectangle and then disappears in the iteration that follows next. Therefore, each iteration, except for the very last one, eliminates on the average exactly one horizontal bar. The last iteration eliminates the two horizontal bars of the rectangle which is part of the original normalon and hence did not require a  $\Delta n_i = 0$  type iteration to be pushed onto *stack*. Thus, if the original normalon has  $r$  horizontal bars, then exactly  $r - 1$  iterations are required to complete its tessellation into rectangles. Since each iteration adds exactly one rectangle to *rectangle-list*, the normalon is tessellated into  $r - 1$  rectangles.

**THEOREM 2.** (Equality of number of horizontal and vertical bar-sets): *The number of horizontal bar-sets of a normalon is equal to the number of its vertical bar-sets.*

**Proof:** We show that in each itera-

tion the number of horizontal and vertical bars that are eliminated is equal.

$\Delta n_i = 2$  type iteration involves the disappearance of a rectangle from *stack*, causing a net decrease of two vertical bars along with the two horizontal ones.

$\Delta n_i = 1$  type iteration, in which one horizontal bar disappears, involves sliding the edge of a horizontal bar along a vertical bar  $V$  until it coincides for the first time with an edge of another horizontal bar, at which point  $V$  disappears because its length is 0. Thus, if one horizontal bar disappears in an iteration, so does one vertical bar.

$\Delta n_i = 0$  type iteration, in which no horizontal bar disappears, involves sweeping a horizontal bar  $H$  parallel to itself until it coincides for the first time with another horizontal bar  $H'$  such that none of the edges of  $H$  coincide with any edge of  $H'$ , so no vertical bar is eliminated either.

Since in each iteration the number of horizontal and vertical bars that are eliminated is equal, and the last iteration leaves us with zero bars of both kinds, the number of vertical bars in the original normalon must be  $r$ , the number of horizontal bars.

$r$  is called the **rank** of the normalon.  $N_r$  denotes a normalon of rank  $r$ . Thus,  $N_2$  is a rectangle—the simplest normalon and the only one which is not concave.  $N_3$  may be considered one of three possible rectangles that underwent an incremental sweep. The total number of sides in a normalon  $N_r$  is  $2r$ . As shown in Figure 4, the notational convention is that  $H_1$  is the topmost horizontal side,  $V_1$  is the adjacent (vertical) side in the clockwise direction, and it is followed by  $H_2, V_2, H_3, V_3, \dots, H_r, V_r$ .

All the dimension-sets of a normalon are of longitudinal type. A dimension-set of a normalon is called horizontal (or vertical) if it describes the distance between two horizontal (or two vertical) sides. Note that the leaders of a horizon-

tal dimension-set are vertical and vice versa. Figure 6 illustrates one of the many possible ISO-standard based proper dimensionings of the normalon of Figure 4. Note that both completeness and nonredundancy are concurrently maintained; the position of each bar is defined with respect to all the others, while at the same time there is exactly one way to calculate this position.

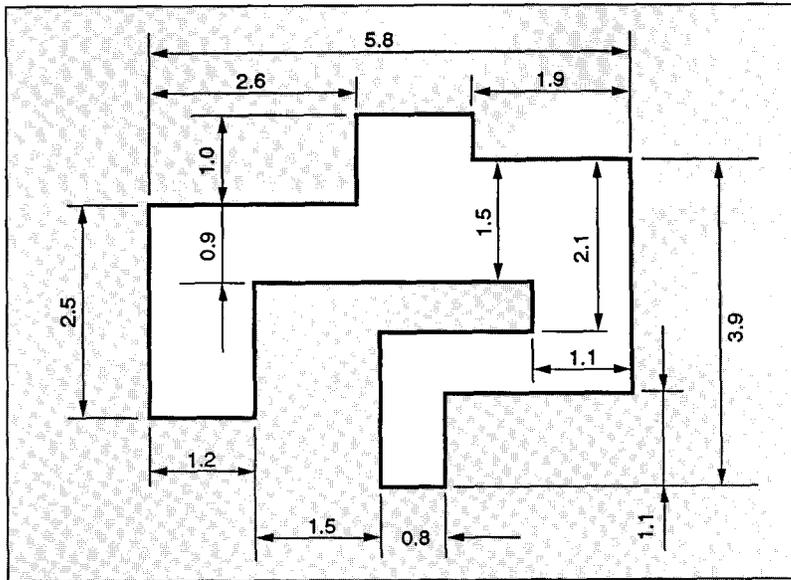
## Dimensioning Graphs

Since the terms 'horizontal' and 'vertical' for the two bar-sets  $\mathbf{H}$  and  $\mathbf{V}$  are interchangeable, we shall henceforth refer to the horizontal bar-set only, with the understanding that everything defined and proved for  $\mathbf{H}$  is applicable to  $\mathbf{V}$  just as well. Adopting a graph-theoretic approach, we define for each of the two bar-sets the following three alternative descriptions of a dimensioning graph. Each description stresses another aspect of a normalon's dimensioning.

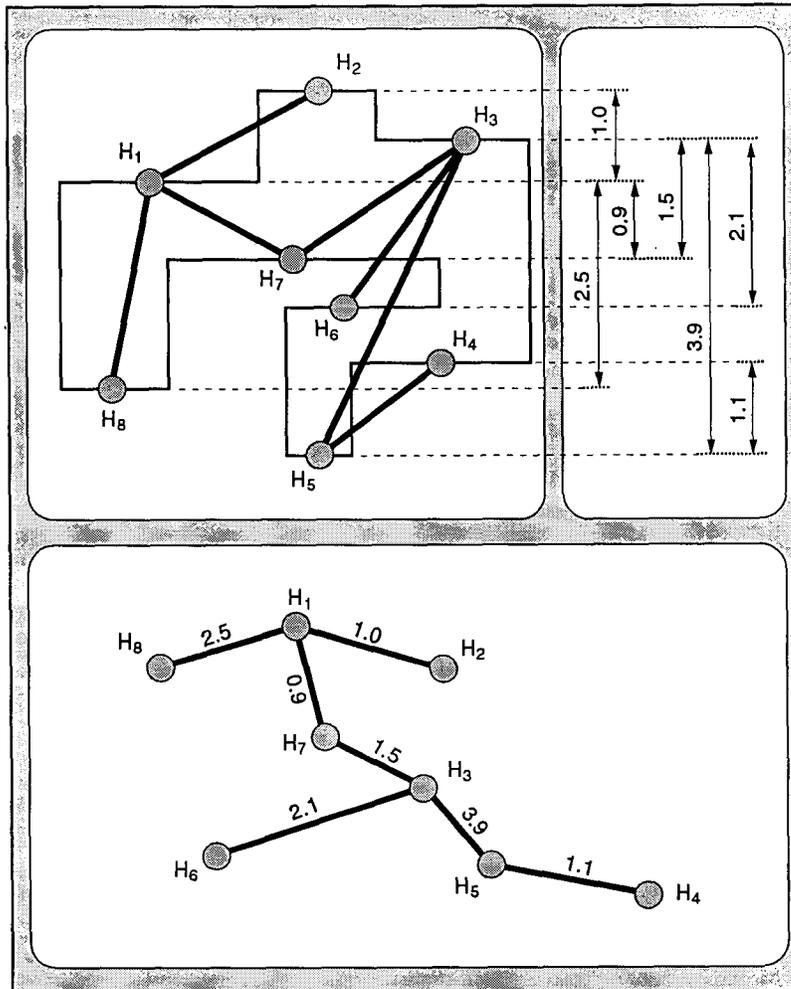
(1) A **spatial graph description**  $G_s(N_r)$  of a normalon  $N_r$  is an  $r$  node graph in which each bar  $H_k$  in the bar-set  $\mathbf{H}$  is a node  $k$  spatially located in the mid-point of  $H_k$  and each dimension-set  $D_{k,l}$  is an edge connecting nodes  $k$  and  $l$ .

(2) A **conceptual graph description**  $G_c(N_r)$  of a normalon  $N_r$  is an  $r$  node graph in which each bar  $H_k$  in the bar-set  $\mathbf{H}$  is a node  $k$  and each dimension-set  $D_{k,l}$  is an edge connecting nodes  $k$  and  $l$ . A conceptual graph is thus a spatial graph in which the requirement for spatial location of nodes has been relaxed.

(3) A **projected graph description**  $G_p(N_r)$  of a normalon  $N_r$  is an  $r$  node graph in which each bar  $H_k$  in the bar-set  $\mathbf{H}$  is a node  $k$  represented by a dashed line colinear with  $H_k$ , and each dimension-set  $D_{k,l}$  is an edge connecting nodes  $k$  and  $l$  and perpendicular to both. A projected graph may be considered as a projection of the corresponding spatial graph in the horizontal direction, but since a total projection would yield just a straight line along which the nodes are represented by



**Figure 6.** One possible proper dimensioning of the normalon of Figure 4.



points, the projection is only partial so as to allow recognition of individual dimension-sets.

$G_s(N_r)$ ,  $G_c(N_r)$ , and  $G_p(N_r)$  are different representations of the same dimensioning graph of  $N_r$ , denoted  $G(N_r)$ . To distinguish between horizontal and vertical dimensioning graphs of  $N_r$ , they are denoted  $G_h(N_r)$  and  $G_v(N_r)$ , respectively. Figure 7 shows the spatial, conceptual, and projected graph descriptions of the horizontal bar-set of the normalon of Figure 6. In the earlier definition we referred only to  $\mathbf{H}$ , but, as noted, all three definitions hold for  $\mathbf{V}$  just as well. Figure 8 shows the same three representations of  $G(N_r)$  for the vertical bar-set of the same normalon.

### Proper Dimensioning of Normalons

The necessary and sufficient conditions that a normalon should satisfy in order for it to be properly dimensioned are stated in the normalon proper dimensioning theorem. We first prove two lemmas that are later used in the proof of the theorem.

**LEMMA 1:** Let  $D_h(N_r)$  be the set of horizontal dimension-sets in a normalon  $N_r$ . If the normalon is properly dimensioned, then the number of horizontal dimension-sets in  $N_r$  is  $|D_h(N_r)| = r - 1$ .

**Proof:** By induction; for a basis, the number of horizontal dimension-sets in a rectangle, for which  $r = 2$ , is  $|D_h(N_2)| = 2 - 1 = 1$ . Suppose now that  $|D_h(N_r)| = r - 1$  for  $N_r$ . We have to prove that  $|D_h(N_{r+1})| = r$ . To convert  $N_r$  into  $N_{r+1}$  we perform an incremental sweep: one of the horizontal bars, say  $H_i$ , has to be divided into two bars,  $H'_i$  and  $H''_i$ , such that  $H'_i$  remains where  $H_i$  was, and  $H''_i$  is swept in a direction perpendicular to  $H_i$ . In order for the new normalon,  $N_{r+1}$ , to remain properly dimensioned, exactly one horizontal dimension-set has to be added to denote the distance by

**Figure 7.** Spatial (left), conceptual (bottom), and projected (right) horizontal dimensioning graph descriptions of the normalon of Figure 6

which  $H_i''$  was swept with respect to  $H_i'$ , so  $|D_h(N_{r+1})| = |D_h(N_r)| + 1 = r$ .

**LEMMA 2:** Let  $G(N_r)$  be the graph of a normalon  $N_r$ . If  $N_r$  is properly dimensioned, then  $G_c(N_r)$  is connected.

**Proof:** By contradiction; suppose  $G_c(N_r)$  is disconnected, then it contains at least two nodes  $k$  and  $l$ , between which there is no path. This implies that there is no way to compute the distance between  $H_k$  and  $H_l$  in  $N_r$ , but since  $N_r$  is properly dimensioned we get a contradiction.

**THEOREM 3.** (Normalon Proper Dimensioning Theorem): Let  $N_r$  be a dimensioned normalon of order  $r$ , let  $G_h(N_r)$  and  $G_v(N_r)$  be its horizontal and vertical dimensioning graphs, respectively. Let  $H(D_{k,l}) = \{H_k, H_l\}$  be the set of two horizontal bars,  $H_k$  and  $H_l$ , between which a dimension-set  $D_{k,l}$  exists, and let  $V(D_{k,l}) = \{V_k, V_l\}$  be the analogous set of two vertical bars.  $N_r$  is properly dimensioned if and only if the following two conditions hold:

(1) Both  $G_h(N_r)$  and  $G_v(N_r)$  are trees, and

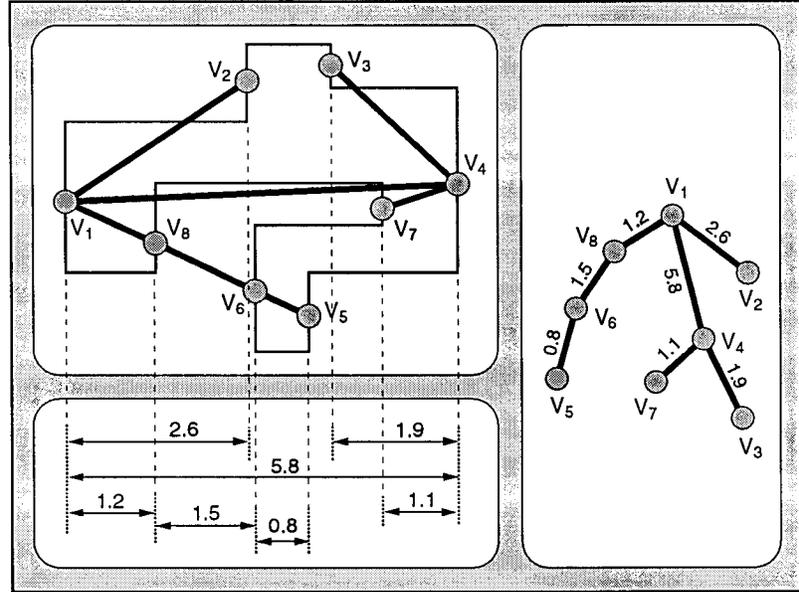
$$(2) \bigcup_{\substack{k,l=1 \\ k \neq l}}^r H(D_{k,l}) = \mathbf{H} \text{ and}$$

$$\bigcup_{\substack{k,l=1 \\ k \neq l}}^r V(D_{k,l}) = \mathbf{V},$$

i.e., the union of all pairs of horizontal and vertical bars between which an explicit dimension exists is the entire horizontal and vertical bar-set, respectively.

**Proof:** Due to the symmetry of  $\mathbf{H}$  and  $\mathbf{V}$  we refer only to  $\mathbf{H}$ . For each of the two conditions stated in the theorem we need to prove two things. First, given that the dimensioning is proper, we have to show that the condition is met. Second, if condition (1) is met, then we show that the dimensioning is nonredundant (i.e., not overdetermined), and if condition (2) is met, then we show that the dimensioning is complete (i.e., not underdetermined). The coexistence of these two properties implies that the dimensioning is proper by definition.

To prove the first part of the



**Figure 8.** Spatial (up left), conceptual (right), and projected vertical dimensioning graphs of the normalon of Figure 6

two-way statement for condition (1), we assume that the normalon is properly dimensioned and need to prove that the dimensioning graph is a tree. We note that a graph is a tree if and only if it satisfies the following two requirements: (a) if  $n$  and  $e$  are the numbers of its nodes and edges, respectively, then  $e = n - 1$ , and (b) the graph is connected.

Let  $N_r$  be a properly dimensioned normalon. In  $G_h(N_r)$  the nodes are the  $r$  horizontal bars of the bar-set  $\mathbf{H}$ , and the edges are the dimension-sets connecting selected pairs of elements of  $\mathbf{H}$ . By Lemma 1,  $|D_h(N_r)| = r - 1$ , which satisfies requirement (a) above, and by Lemma 2,  $G_h(N_r)$  is connected, which satisfies requirement (b) above. Thus  $N_r$  is a tree.

We now prove the second part of this two-way statement for condition (1). Since the assumption is that the dimensioning graph is a tree, it has no cycles, so the distance between any two bars can be calculated by one way at the most, which is following the only path between the two nodes in the tree representing the two bars between which the distance is sought. This satisfies the nonredundancy requirement of

proper dimensioning.

To prove the first part of condition (2), we assume that the dimensioning is proper and we need to show that

$$\forall p \exists \{ (H_p \in \mathbf{H}) \wedge (H_p \in \bigcup_{\substack{k,l=1 \\ k \neq l}}^r H(D_{k,l})) \}.$$

Since proper dimensioning implies completeness, each bar  $H_k$  is related by a dimension-set to at least one other parallel bar  $H_l$ , so

$$\bigcup_{\substack{k,l=1 \\ k \neq l}}^r H(D_{k,l}) \text{ is } \mathbf{H}, \text{ the set of all the horizontal bars.}$$

For the second part of condition (2), we are given that

$$\bigcup_{\substack{k,l=1 \\ k \neq l}}^r H(D_{k,l}) = \mathbf{H},$$

and we need to show that the dimensioning is complete. If it is not complete, then there exists at least one bar  $H_p$  such that there is no dimension-set between it and any other bar parallel to it. This implies

$$\bigcup_{\substack{k,l=1 \\ k \neq l}}^r H(D_{k,l}) \notin \mathbf{H},$$

contrary to the original assumption. This completes the proof.

The normalon proper dimensioning theorem can be rephrased

more compactly as the following corollary:

*A normalon of rank  $r$  is dimensioned properly if and only if each of its horizontal and vertical dimensioning graphs is an  $r$  node tree.*

The requirement that the graph be a tree takes care of the *nonredundancy* of the dimensioning, while

the requirement that the tree has exactly  $r$  nodes guarantees the dimensioning's completeness.

### Applications of the Theorem

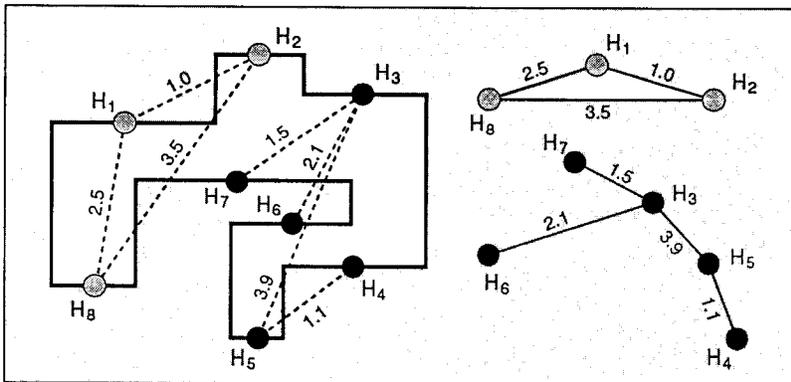
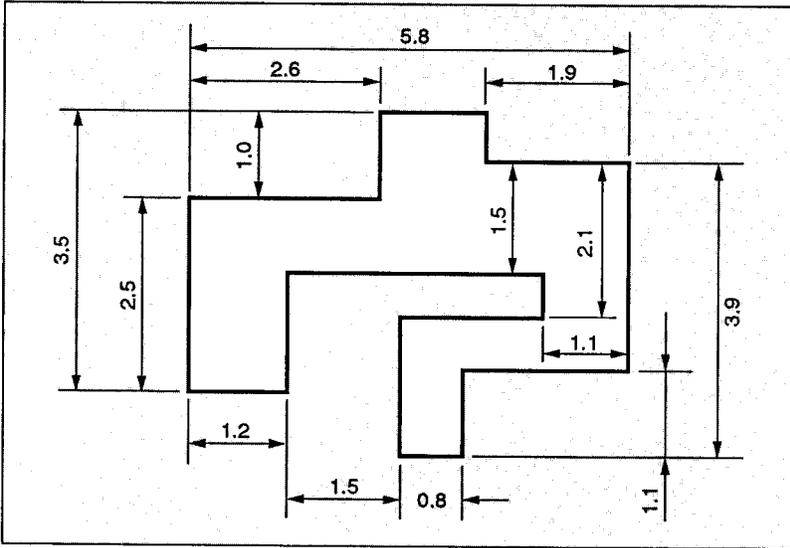
The proper dimensioning theorem provides a sound theoretical basis for a variety of operations in the view-level analysis phase of engineering drawing understanding.

(1) *Annotation removal as a preprocessing stage for 3D reconstruction.* By constructing the 2D graphs  $G_h(N_r)$  and  $G_v(N_r)$  of a normalon  $N_r$ , it is possible to determine if the set  $\mathbf{D} = \mathbf{H} \cup \mathbf{V}$  of dimension-sets detected by the syntactic phase of MDUS is both complete and nonredundant. To perform this we first verify that

$$|\mathbf{H}| = |\mathbf{V}| = r - 1,$$

that is, the number of each of the horizontal and vertical detected dimension-sets is  $r - 1$ . If so—each of the two sets  $\mathbf{H}$  and  $\mathbf{V}$  of  $r - 1$  dimension-sets is checked as to whether or not both  $G_h(N_r)$  and  $G_v(N_r)$  are trees.

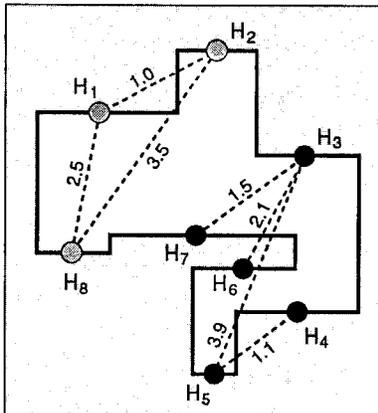
Figure 9 is an example of an improper dimensioning of the normalon of Figure 4, in which only the first condition is met. In spite of the fact that  $|\mathbf{H}| = |\mathbf{V}| = 8 - 1 = 7$ , the dimensioning is incomplete and therefore improper. This is verified by observing that on one hand, one of the three dimension-sets whose values are 1.0, 2.5, and 3.5, is redundant, while on the other hand, one dimension-set is missing, causing disconnectedness of the corresponding graph. Thus, even though the number of dimension-sets is 7, as it should be, the corresponding dimensioning graph in Figure 10 is disconnected—it consists of one tree and one graph which has a loop (triangle), rather than a tree, and is therefore improper. Figure 11 demonstrates that the dimensioning in Figure 9 is underdetermined by showing one member of an infinitely large family of normalons, all of which are possible due to the improper dimensioning of the normalon in Figure 9. If the dimensioning is



**Figure 9.** Improper dimensioning of the normalon of Figure 4

**Figure 10.** Spatial (left) and conceptual dimensioning graphs of the normalon of Figure 9.

**Figure 11.** Another possible underdimensioned normalon



proper, then since  $\mathbf{D} = \mathbf{H} \cup \mathbf{V}$ ,  $|\mathbf{H}| = |\mathbf{V}| = r - 1$ , and  $\mathbf{H} \cap \mathbf{V} = \emptyset$ , we get  $|\mathbf{D}| = |\mathbf{H}| + |\mathbf{V}| = 2(r - 1)$ . If all the annotation in the drawing is just dimensioning, then removing all  $2(r - 1)$  dimension-sets is highly likely to produce a clean, noise-free normalon to be used as input to any 3D reconstruction algorithm.

If a normalon is found by the automatic system to be improperly dimensioned, then either the original drawing is improperly dimensioned or the syntactic phase of MDUS has failed to extract the correct set of dimension-sets. Since the second possibility is more likely, this test serves as a good indication as to the global success of dimension-set parsing.

If the number of detected dimension-sets is smaller by  $k$  than the expected  $r - 1$ , then probably  $k$  dimension-sets were overlooked by the system, and should be searched using different approaches, such as relaxation of parameters used to detect arrowheads in the drawing [5]. Furthermore, if  $k = 1$ , then the dimensioning graph consists of two disconnected components, and the search can be guided by the fact that the missing dimension-set should denote the distance between a bar in one component of the graph and a bar in the other component.

If, on the other hand, the number of detected dimension-sets is greater than the expected  $r - 1$  by  $k$ , then probably  $k$  dimension-sets have been misidentified and should be excluded from the set. If there are  $k$  loops in the graph, then  $k$  dimension-sets should be deleted from the graph, one from each loop, such that the graph becomes a tree. The decision as to which dimension-set should be deleted is based on the relative likelihood of each dimension-set to be misdetected and on whether or not its removal changes the connectivity of the dimensioning graph.

(2) *Determine implicit dimensions:* An implicit dimension is the distance between two parallel bars that are not components of the same di-

mension-set. An important property of a tree is that there exists exactly one path from each node  $i$  to any other node  $j$  in the tree. This implies that there is exactly one way to calculate the implicit dimension between any pair of elements of  $\mathbf{H}$  which do not share a common dimension-set. Using the dimensioning tree of a properly dimensioned normalon, it is possible to determine the distance  $d(H_i, H_j)$  between any two horizontal sides  $H_i$  and  $H_j$  of the normalon. To do so we have to find the (unique) path in the tree between  $H_i$  and  $H_j$ .

Let the length of the path be  $k$ ;  $2 \leq k \leq r - 1$ , and let the path pass through  $H_i = H_{m_0}, H_{m_1}, H_{m_2}, \dots, H_{m_k}, H_j$ , where for each pair  $H_{m_{p-1}}, H_{m_p}$ ;  $1 \leq p \leq k$ , there exists an explicit dimension-set  $D_{m_{p-1}m_p}$ , whose value is  $d(H_{m_{p-1}}, H_{m_p}) = D_{m_{p-1}m_p}$ . The implicit dimension between  $H_i$  and  $H_j$  is then:

$$\begin{aligned} d(H_i, H_j) &= |d(H_{m_0}, H_{m_1}) + \\ &\quad d(H_{m_1}, H_{m_2}) + \dots + d(H_{m_{k-1}}, H_{m_k})| \\ &= s_1 D_{m_0 m_1} + s_2 D_{m_1 m_2} + \\ &\quad \dots + s_k D_{m_{k-1} m_k} \\ &= \sum_{p=1}^k s_p D_{m_{p-1} m_p} \end{aligned}$$

where  $s_p$ , the sign symbol is,

$$s_p = \begin{cases} +1, & \text{if the direction from} \\ & H_{m_{p-1}} \text{ to } H_{m_p} \\ & \text{is the same as the} \\ & \text{direction from } H_i \text{ to } H_j; \\ -1, & \text{if the direction from} \\ & H_{m_{p-1}} \text{ to } H_{m_p} \\ & \text{is opposite to the} \\ & \text{direction from } H_i \text{ to } H_j. \end{cases}$$

The definition of  $s_p$  is such that  $d(H_i, H_j)$  is always positive, eliminating the need for absolute value. The direction is most easily determined by examining the projected horizontal tree. For example, to find the distance between  $H_4$  and  $H_6$  in the normalon of Figure 8, using the dimensioning graph we find that the path is  $(H_4, H_5, H_3, H_6)$ . Therefore,  $(m_0, m_1, m_2, m_3) = (4, 5, 3, 6)$ , and  $(s_1, s_2, s_3) = (-1, +1, -1)$  For example,  $s_1 = -1$  because the direction from  $H_4$  to  $H_5$  is op-

posite to the direction from  $H_4$  to  $H_6$ . Thus,

$$\begin{aligned} D(H_4, H_6) &= \\ d(H_4, H_5) + d(H_5, H_3) + \\ d(H_3, H_6) &= \\ \sum_{p=1}^3 s_p D_{m_{p-1} m_p} &= \\ (-1)D_{4,5} + (+1)D_{5,3} + \\ (-1)D_{3,6} &= -1.1 + 3.9 - 2.1 = 0.7. \end{aligned}$$

(3) *Provide for automatic dimensioning of normalons:* CAD/CAM systems are usually not equipped with an intelligent mechanism for automatic dimensioning of designed objects due to the variety and complexity of issues that need to be addressed: engineering considerations, tolerance accumulation, and physical location.

The proper dimensioning theorem calls for a search for two orthogonal spanning trees, one for the horizontal bar-set and one for the vertical bar-set, while taking into account the effects of the following points.

• **Engineering considerations:** Frequently, the human designer has in mind requirements as to which dimensions should be explicitly dimensioned. These preferences are influenced by the following two factors:

1) *Functionality:* The importance of the accuracy of distance between two geometry entities of the object to the proper operation of the product, of which the object is a part.

2) *Measurability:* The ability to carry out direct quality assurance measurements of the explicit dimension. The designer usually assigns a value for general (default) tolerance demand which the manufactured part has to meet in order to function, usually as a part of an assembly. In many instances, the choice of a particular distance to be explicitly dimensioned is accompanied by a demand for tolerance which is more tight than the general tolerance.

Frequently, the original dimensioning assigned by the designer is

altered once by the manufacturing engineer for producibility and yet another time by the quality assurance engineer to enable quality checks. The undesired cumulative effect of these redimensioning operations is that the end product may deviate from the original intent of the designer. The decisions as to what are the most important dimensions for both functionality and measurability require a high level understanding of the product and its manufacturing process, and are therefore left to the design, manufacturing, and testing personnel at this stage. Once these crucial dimensions are determined, though, the proper dimensioning theorem can be utilized to automate the rest of the dimensioning process. Intelligent automatic dimensioning is discussed in more detail in [4]. The problem of redimensioning can be approached by incorporating global, high-level producibility and measurability considerations into the system.

• **Tolerance accumulation:** As noted in ANSI Y14.5M standard [2], to avoid accumulation of deviations from the nominal dimension, the number of explicit dimensions that have to be added and/or subtracted in order to calculate any implicit dimension should be minimized. In terms of the dimensioning graph, this requirement implies that the depth of the spanning tree of the graph,  $d_{max}$ , (i.e., the length of the longest path in the tree) should be as small as possible.

Ideally, for  $N_r$  with  $r \geq 3$   $\min(d_{max}) = 2$ , in which case it is possible to calculate the distance between any two bars of a bar-set by adding only two numbers. The spanning tree in this case has the shape of a star. In practice, however, such a dimensioning is frequently impossible because it contradicts engineering and/or physical location considerations.

• **Physical location:** The layout of dimension-sets has to be designed so that they will not interfere with

one another and will enable the human drawing reader to distinguish between geometry and annotation entities. Leaders should not cross any other line, be it geometry or annotation, while perpendicular witnesses of different dimension-sets may cross each other, although such crossing should be avoided as much as possible. Annotation inside the contour drawn by the geometry entities should also be avoided as much as possible. Even if we consider only normalons that just use dimension-sets of longitudinal type, for each pair of bars between which a dimension-set should exist, decisions have to be made as to the number of witnesses that ought to be used (zero, one, or two; if one – which of the two?) and as to the kind of dimension-set that should be used (symmetric or asymmetric, regular or irregular? [3])

### Discussion and Open Problems

This article has presented a pre-processing step which should precede the application of any 3D reconstruction algorithm from a set of views provided by any manually or automatically prepared engineering machine drawing. This is one task of a Machine Drawing Understanding System (MDUS).

Open problems that ought to be considered are:

• **Managing the combinatorial explosion.** The number of possible proper dimensionings of a normalon  $N_r$ ,  $Z(N_r)$ , is extremely big even for normalons of modest order  $r$ , and even if we ignore the different physical positions of a dimension-set between two given bars. Since horizontal and vertical dimensionings are independent of each other, if we denote by  $S(r)$  the number of spanning trees of a graph with  $r$  nodes, then  $Z(N_r) = (S(r))^2$ . The situation gets much more complex if we consider non-normalons and multiple views. Some good heuristics ought to be found in order to efficiently reduce the search space and still come up with a reasonable, if not optimal,

dimensioning scheme.

• **Generalization within the single 2D view.** The class of nonredundant normalons should be augmented to any single 2D shape by gradually allowing for redundant normalons, presence of bars tilted at any angle, arcs, and splines. How should these various elements be incorporated into the current data structure, i.e., the dimensioning graph?

• **Generalization to multiple orthogonal views.** The single 2D view considered in this research should be generalized to multiple orthogonal views (usually three in engineering drawings), axonometric views and cross sections. The challenge to proper dimensioning here is that the set of dimension-sets is distributed among the orthogonal views, calling for a revision in the definition of proper dimensioning of a single view and in the derived data structure.

• **Exploitation of dimension-set information.** Rather than treating dimension-sets as noise that needs to be removed from the drawing, recognize the content of their text and use it to enhance the accuracy of the 2D views and the 3D model derived thereof. G

### References

1. Aldefeld, B. On automatic recognition of 3D structures from 2D representations. *Comput. Aided Design* 15, 2 (1983), 59–64.
2. ANSI Y14.5M. *Dimensioning and Tolerancing*. American National Standard, Engineering Drawings and Related Documentation Practices. The American Society for Mechanical Engineers, New York, 1982.
3. Dori, D. A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. *Comput. Vision, Graph. Image Process.* 47 (1989), 271–291.
4. Dori, D. Intelligent automatic dimensioning of CAD engineering machine drawings. In *Proceedings of The International Society for Mini and Microcomputers Conference on Com-*

- puter Applications in Design, Simulation, and Analysis, (Reno, Nev., Feb. 22-24). ISMM/MIMI, Anaheim, Calif.: ACTA Press, 1989, pp. 137-140.
5. Dori, D. Syntax enhanced parameter learning for recognition of dimensions in engineering machine drawings. *Int. J. Robotics and Automation* 5, 2 (May 1990).
  6. Dori, D. and Pnueli, A. The grammar of dimensions in machine drawings. *Comput. Vision, Graph. Image Process.* 42 (1988), 1-18.
  7. Freeman, H. Analysis of line drawings. In *Digital Processing and Analysis*, J.C. Simon and A. Rosenfeld Eds., Noordhoff, Leyden, 1977.
  8. Gigus, Z. and Malik, J. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Trans. Pattern Analysis and Machine Intell. PAMI-12*, 2 (Feb. 1990), 113-122.
  9. Haralick, R.M. and Queeney, D. Understanding engineering drawings. *Comput. Graph. Image Process.* 20, 3 (1982), 242-258.
  10. Kelly, J.C., Parks, R.E. and Saylor, D.B. CAD/CAM-005: An introduction to the data exchange process using IGES. SANDIA Rep. SAND86-2564. UC-32, 1987.
  11. King, A.K. An expert system facilitates understanding the paper engineering drawings. In *Proceedings, IASTED International Symposium Expert Systems Theory and Their Applications* (Los Angeles, Calif. Dec. 1988), ACTA Press, pp. 169-172.
  12. Preiss, K. Constructing the solid representation from engineering projections. *Comput. Graphics* 8, 4 (1984), 381-389.
  13. Shapira, R. A technique for the reconstruction of a straight-edge, wire-frame object from two or more central projections. *Comp. Graph. Image Process.*, 3 (1974), 318-326.
  14. Shapira, R. The use of objects' faces in interpreting line drawings, *IEEE Trans. Pattern Analysis and Machine Intell. PAMI-6* (1984), 789-798.
  15. Shapira, R. More about polyhedra—interpretation through construction in the image plane. *IEEE Trans. Pattern Analysis and Machine Intell. PAMI-7* (1985), 1-16.
  16. Shapira, R. and Freeman, H. The Cyclic Order Property of Vertices as an Aid in Scene Analysis. *Commun. ACM* 22 (1976), 368-375.
  17. Smith, B. and Wellington, J. *Initial Graphics Exchange Specification (IGES), Version 3.0*, U.S. Department of Commerce, National Institution of Standards, NBSIR 86-3359, Apr. 1986.
  18. Sugihara, K. *Machine Interpretation of Line Drawings*. The MIT Press, Cambridge, Mass., 1986.
  19. Wesley, M.A. and Markowski, G. Fleshing out projections. *IBM J. Res. Develop.* 25 6, (1981) 934-953.

**CR Categories and Subject Descriptors:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—curve and object representations, geometric algorithms; I.4.5 [Image Processing]: Reconstruction—curve and object representations, geometric algorithms

**General Terms:** Documentation Standardization

**Additional Keywords and Phrases:** Annotation, CAD/CAM, document analysis, documentation automation,

engineering drawings, graph representation, proper dimensioning.

**About the Author:**

**DOV DORI** is a Senior Lecturer of industrial engineering and management at the Technion, Israel Institute of Technology. His current research interests include image processing, pattern recognition, image understanding, geometric modeling, and object-oriented analysis. **Author's Present Address:** Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Technion City, Haifa 32000, Israel; email: ierdori@technion.bitnet.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© ACM 0002-0782/92/1000-092 \$1.50



## The Gift of Life

Since St. Jude Children's Research Hospital opened in 1962, it has forged new treatments for childhood cancer and has helped save thousands of children. But the battle has just begun. You can join the fight. To find out how, call 1-800-877-5833.

