

Abstract

An engine that automatically reconstructs a large variety of polyhedral, origami and wire-frame objects from single-view sketched drawings generated in a calligraphic interface is presented. The engine has two stages. An innovative optimisation-based line-drawing beautifier stage is introduced to convert rough sketches into tidied-up line drawings. Optimisation-based 3D reconstruction follows. Solutions are provided with which to overcome the problems associated with earlier approaches to optimisation-based 3D reconstruction. Suitable adjustments in the optimisation algorithms are proposed; simple and efficient tentative models are introduced, and current regularities are categorised in order to allow the objective function to be simplified. All three actions help to prevent local optima and improve the computational efficiency of optimisation-based 3D reconstruction. They all proved to be effective techniques to reduce the typical failure rate of optimisation approaches. A discussion of results that validate the engine is also provided.

Keywords

Graphics recognition and interpretation

Visual Perception

Sketch understanding

Line Drawings Reconstruction

Geometric Modelling

1. Introduction

The automatic reconstruction of line drawings has been an active research area since the pioneering works by Roberts [1] and Johnson [2], and good references exist that offer an introduction to the earlier advances made in this subject [3], [4], [5]. The search for robust sketch-based geometric modellers has emerged over the last decade [6], [7], [8], [9] and the field of calligraphic interfaces is also growing (see Computers & Graphics 24, 2000, for the special issue “Calligraphic Interfaces: towards a new generation of interactive systems”). Nowadays, very intuitive and easy-to-use sketch-based modelling applications can be developed by combining 3D reconstruction “engines” and calligraphic interfaces. The term “engine” is used to mean that 3D reconstruction must act like a “black box”, i.e. it produces an output every time it receives an input and the user is not required to have any knowledge whatsoever of its internal configuration. Only a few reconstruction algorithms are able to act as engines in this *modelling by sketch* approach aimed at aiding conceptual design by single-view sketches, because they must be tolerant to faults in the drawing and work incrementally [10]. Optimisation approaches have been successfully used in this context [11] [12] because models can be updated whenever the input sketch is modified and the reconstruction process is tolerant to imperfections.

Reconstruction by optimisation is supposed to be straightforward and the resulting model is assumed to fit the desired shape. However, some important problems appear and in this paper appropriate solutions are provided to overcome them:

- Line drawing beautification is performed in order to *tidy up* (i.e. repair and finish) rough sketches so as to produce clean line drawings. This process is necessary since designers tend to generate vague and imperfect sketches, while geometrically complete and consistent line drawings are needed to feed the 3D reconstruction stage. In this paper a new optimisation-based batch beautifier is proposed to deal with this problem.
- Optimisation-based 3D reconstruction is very sensitive to small variations in optimisation parameters [13]. This means that reconstruction can result in local minima, even when a complete and consistent line drawing is provided.

Appropriate adjustments can be made to reconstruction by optimising algorithm parameters, which is the approach we suggest to improve the success rate of optimisation-based reconstruction.

- Mathematical formulation of perceptual cues (also called “regularities”, or “artefacts”) has been studied by a small group of researchers within the reconstruction community ([10], [14], [15] and [16]) and has led to a poorly defined objective function, thus making a global minimum very difficult to find. A global minimum is required because, in general, just one solution is “psychologically plausible”. Psychological plausibility is itself a complex subject (see [17] for a full and updated review of the state of the art). Nevertheless, for our purposes, the psychologically plausible model is simply the one that the vast majority of humans should perceive from a drawing (human observers rarely hesitate in choosing the solution that best fits visual cues). Hence, ambiguous shapes and pathological drawings (which can hardly be considered to be images of actual 3D shapes) are outside the scope of our work here. For common shapes, whether a simple efficient objective function based on a *categorisation of regularities* is a condition to solve the problem or not is still an open question that we will consider below.
- The importance of tentative models to reduce calculation times and improve success rates is also an open question. Two tentative model approaches are described and we will argue that even the simplest optimisation algorithm with a poorly defined objective function should find the global optimum provided it starts out with a good *tentative model*.

In short, we describe a 3D reconstruction engine developed for use in a modelling-by-sketching approach obtained by linking our reconstruction engine with a calligraphic interface. The reconstruction engine joins two consecutive stages, i.e. drawing beautification followed by 3D model reconstruction. It is able to cope with non-perfect 2D geometries (by way of an innovative optimisation-based beautification stage) and then converts line drawings into 3D models (via a 3D reconstruction by optimisation stage). In this way, typical failure rates for 3D reconstruction by optimisation approaches (due to local minima) are reduced. We shall see that undesirable models,

with valid topology but wrong dimensions, are easily prevented by normalising the optimisation parameters. Finally, wrong topologies derived from local minima are avoided by careful choice of the regularities and/or by employing a good strategy to determine initial tentative models.

Our scope is limited to single-view reconstruction because 3D reconstruction from multiple orthographic views requires a radically different approach. The nature of shapes that can be reconstructed is limited too, since optimisation-based reconstruction can only deal with polyhedral blocks (manifold), origami objects and wire frame models (non-manifold). Additionally, in some implementations, it may be generalised to cope with curved surface models such as those that usually appear in sketches done during conceptual design. However, many complex shapes fall into these simple and useful categories. An example is shown in Figure 1, where the axonometric drawing of an industrial building structure (Figure 1a) is presented side-by-side with the three orthographic views (front, upper and side views) of its final 3D model (Figure 1b). Sketches of many skeletons (or “control structures”) used to model complex mechanical parts in 3D CAD modelling systems (see Figure 2) and many sketches of architectural buildings [18] belong to the “block world” subset. All of them are easily reconstructed by our application.

Insert Figure 1 about here.

Insert Figure 2 about here.

The paper is organised as follows. Section 2 includes a review of related work, where sketch-based modelling, optimisation-based 3D reconstruction and tentative models for reconstruction purposes are covered in detail. In section 3 there is a detailed explanation of the solutions provided in our engine to solve the four main problems of optimisation-based 3D reconstruction described above (i.e. the optimisation-based beautification of line drawings, the adjustment of optimisation algorithms, the categorisation of regularities and the generation of tentative models approach). In the examples and discussion section, beautification and 3D reconstruction stages are covered separately. Finally, some conclusions are outlined.

2. Related work

While the old-fashioned design-by-drawing methodology is being replaced by *design-by-digital-prototypes*, the ideal engineering design support system is still an academic challenge (see Computer-Aided Design 32, 2000, for the special issue “Conceptual design: issues and challenges”). According to Ullman [9], the first requisite for such conceptual engineering-oriented CAD tools should be 2D sketching capability, to allow fast and easy exploration of new ideas and serve as a short-term memory. In other words, sketch-based conceptual design incorporates ideas from computer graphics, user interface design and computer vision in order to facilitate the speedy creation of detailed 3D designs, usually from elementary 2D sketches, and to improve support for the flexible generation of alternative ideas at the early stages of design [19]. For details of a number of academic or prototype systems, see [20], [21], [22], [23] and [24]. A summary of the state-of-the-art in modelling by sketching, optimisation and tentative modelling follows below.

2.1 Sketch-based modelling

Although the first CAD system used the light pen as a data input device [25], the WIMP paradigm (Windows, Icon, Menu, Pointing device) is the only one available today for commercial CAD packages. However, a research area has emerged that focuses on using freehand drawings and sketches as a way to obtain 3D geometric models via input devices, like pens on graphic tablets, LCD tablets or tablet PCs. We can distinguish two main approaches to 3D modelling by sketching: gesture and reconstruction based methods.

Gestural modelling systems provide predefined gesture alphabets that encode some geometric modelling operations. For instance, in SKETCH [26], which is aimed at architectural shapes, a block-type primitive is defined by three segments starting from the same point and the positive volumes are built upwards, whereas the negative volumes are built downwards. In a similar way, SKETCH-N-MAKE [27] facilitates numerical control in the process of machining simple parts by using a gestural interface to model them. Quick-Sketch [21] is a mechanical design-oriented computer tool that consists of a 2D drawing environment based on constraints, although it also allows 3D geometry to be generated by interpreting a number of modelling gestures. Qin et al. [24]

presented a system that recognises form features by inferring knowledge from sketched gestures, but only very elemental features (like prisms, cylinders, etc.) are recognised in the first version. GIDeS [28] allows data input from a single-view projection or from several orthographic views. The system supplies a gesture alphabet to identify a basic set of modelling primitives such as a prism, a pyramid, extrusion and revolution, among others. In addition, the dynamic recognition of these modelling gestures provides the user with a number of contextual icons with which to solve ambiguities. Some sketch-based modelling applications are oriented to freeform surface modelling, such as Teddy [29], where the system automatically generates one surface using a polygonal mesh that matches the silhouette drawn by the user.

Reconstructional modelling is the alternative to gestural modelling, where geometric reconstruction techniques are applied to build the object's geometry from a sketch that represents some projection of the object. We can classify these reconstruction engines as being either projective or perceptual. Projective reconstruction is, roughly speaking, the inverse of the projection process. Projection invariants are extensively used to recover as much information about the object as possible. On the other hand, the perception approach focuses on the way humans create very rich three-dimensional scenes from retinal images and uses as many perceptual rules, or cues, as are achievable. Both categories usually coexist in *reconstructional modelling* systems. For example, Oh and Kim [30] use perceptual reasoning just to overcome the geometric uncertainty of inverse projection, and they abandon perceptual reasoning as soon as a projection matrix is determined.

In reconstructional modelling, a preliminary stage may exist where batch (off-line) [31] or interactive (on-line) beautification is performed [32] [33]. Interactive beautification provides the user with immediate feedback because it operates as the user draws the sketch and it offers better integration with a calligraphic interface. Batch beautification allows some analysis to be implemented, for example symmetry detection, which is better carried out over the whole sketch.

Some examples of reconstructional systems that implement a batch beautification stage include "Digital Clay" [18], which is oriented towards architectural design and supports polyhedral objects. This application provides sketch input, with batch beautification,

which feeds a reconstruction browser that uses Huffman-Clowes algorithms [34] [35] to reconstruct the object geometry. Stilton [36] is also oriented towards the field of architecture. Its reconstruction process uses the optimisation approach based on genetic algorithms. Oh and Kim [30] assume that edges defining the same face are drawn in consecutive order and, furthermore, they assume that the “main” faces are drawn first, following a “normal” drawing sequence.

An example of the interactive beautification approach is Cigro [37], which provides a calligraphic interface that implements an interactive beautifier and feeds a reconstruction engine operating on an axonometric projection. The system supports rectangular polyhedral objects and provides dynamic viewpoints that make it easy to implement an incremental modelling strategy.

To sum up, there are good reasons to pursue freehand sketching and system detection of designer intents. However, there are two alternative methods to capture those designer intents: automatic (reconstruction-based) and interactive (gesture-based). Important differences also appear in the beautification process. Those design intents that have a “local” impact (i.e. they affect the geometry only in the neighbourhood of some element) are primarily carried out at the same time as the sketching (interactively). Beautification that affects those design intents that require global alterations of current geometry, however, are better done after sketching (in batch mode). Corners that do not meet are good examples of local beautifications, while symmetry requires a more global consideration.

2.2 Optimisation-based 3D reconstruction

In the optimisation-based 3D reconstruction approach, a Cartesian coordinate system whose XY coordinate plane fits in the drawing plane is defined (the so-called inflation reference system). This ensures that (x_i, y_i) coordinates of every junction in the drawing will be equal to (x_i, y_i) coordinates of the corresponding vertex in the reconstructed model. Hence, the 3D model is simply defined by the set of z coordinates of all model vertices: $\mathbf{z} = (z_1, z_2, \dots, z_n)$. The infinite sets of coordinates constitute the *orthographic extension* [38]. Almost all the potential models are “*tangles*”, i.e. models that contain “obvolute” faces [13], or, in general, psychologically non-plausible models (Figure 3). Inflation is the process of choosing a set of coordinates that represent the

psychologically plausible 3D model. Inflation could be said to be the inverse of projection.

Insert Figure 3 about here.

Inflation is performed by *optimising* a figure of merit, or objective function, defined as a weighted sum of perceptual cues:

$$F(\mathbf{z}) = \sum \alpha_j R_j(\mathbf{z}) \quad (1)$$

where α_j is the j -th weighting coefficient, and $R_j(\mathbf{z})$ is the j -th cue, or regularity, expressed in terms of the independent variables \mathbf{z} . Regularities must be formulated so as to be equal to zero when complete compliance of the condition is achieved, and very different from zero for clear non-compliance. Regularities range from Minimum Standard Deviation of Angles (MSDA) [38] and face planarity [14] to the extensive set of twelve regularities detailed in [15]. In fact, equation (1) determines a multi-criteria optimisation, where the α_j coefficients serve to normalise the sensitivity of different regularities. This normalisation is necessary since the different regularities belong to very different ranges. It may also serve to increase the objective function sensitivity with respect to one or more of the regularities, thus converting those regularities into “de facto”, or “soft”, constraints.

The first authors to use mathematical minimisation employed the hill-climbing search because it is robust and easy to implement [38]. In addition, its repeatability allows results to be compared. Briefly, for an n -dimensional variable $\mathbf{z} = (z_1, z_2, \dots, z_n)$, an arbitrary step-size s_1 is defined and the $2n$ values obtained by sequentially adding and subtracting the step-size to every coordinate are checked: $\mathbf{z}_1 = (z_1 + s_1, z_2, \dots, z_n)$, $\mathbf{z}_2 = (z_1 - s_1, z_2, \dots, z_n)$, ... $\mathbf{z}_{2n} = (z_1, z_2, \dots, z_n - s_1)$. Then, the variable \mathbf{z}_i , corresponding to the smallest objective function is selected as the new current \mathbf{z} . The process is repeated in a loop until a balance is achieved (when variations in any variable improve the current value of the objective function by less than a threshold value), or when the predefined number of searches has been reached. A new loop then begins, with a smaller step-size s_2 . Typically, three to ten loops are performed with decreasing steps ($s_1 > s_2 > \dots$). The approach is simple, but its computational efficiency can be improved through, for example, a gradient descent algorithm, which is a more faithful interpretation of hill

climbing, while running much faster. However, the analytical formulations of the partial derivatives of every regularity with respect to every z coordinate are required [39], [40].

It is notable that Marill used a fixed set of three step-sizes ($s_1= 1$, $s_2= 0.5$, $s_3= 0.1$). This only worked because all their line drawings lie within the range $\Delta x= 7.79$ and $\Delta y= 8.16$, which means that the initial step is about 12% of $\max(\Delta x, \Delta y)$. Leclerc and Fischler [14] used five steps, decreasing Δz by a factor of two ($s_1= 0.125$, $s_2= 0.0625$, $s_3= 0.03125$, $s_4= 0.015$, $s_5= 0.007$). This option increases the calculation time while leading to more precise solutions. Again, their examples were in a narrow range ($\Delta x= 4.78$, $\Delta y= 3.50$). Consequently, the Δz they propose is about 2% of $\max(\Delta x, \Delta y)$. Unless they argued that larger steps force the algorithm out of the valley of attraction of the current local minimum, they did not define a true *normalisation* strategy. Similarly, Lipson and Shpitalni [15] skipped the question by limiting the sketch dimensions to half of the screen width (limiting in this way the use of scales, zooms and other similar transformations that are very helpful during sketch creation).

We can also highlight Brown and Wang's approach by assuming that "undoubtedly standard methods for minimising local minima difficulties could be employed with typical amounts of success... (included but not limited to the family of simulated annealing algorithms)" [13]. Nevertheless, our results were discouraging because we introduced a Simulated Annealing algorithm, which was supposed to be able to find the global minimum, and found that its failure rate is similar to those of local optimisation algorithms. This happens for both algorithms when the input is a "bad" tentative model (a model that tends to drive optimisation to a local optimum), and when poor or ill-defined regularities are used to generate an objective function that is unnecessarily complex and plagued with local minima [41].

2.3 Tentative models

Unfortunately, the input line drawing is a bad initial solution for the optimisation approach described above, since it is a local minimum where some regularities are trivially satisfied (face planarity, for instance). The strategies developed to avoid this trivial optimum lead to "tentative" initial models, and all of them fall into one of two categories: *iterative inflation* and *direct generation*.

Iterative Inflation techniques are based on “false regularities”. Marill’s MSDA was the first Iterative Inflation method [38]. MSDA is a *false* regularity because it does not reflect any property of the line drawing that must correspond to some properties in the searched model. In other words, it is not linked to any perceptual cue of the image. It is a heuristic rule, based on the fact that almost regular and convex polyhedra have a tendency to accept this assumption and, consequently, it seems easy to guess that this approach is acceptable only for almost regular and convex polyhedra. See [13] for an extensive analysis of the limitations of MSDA.

Leclerc and Fischler [14] presented an objective function with an “inflation” part (F_I) that included false regularities and an “optimisation” part (F_O) that contained those regularities that are trivially satisfied in the initial solution. A variable parameter λ was introduced with an initial value below or equal to one, which decreases towards zero as the optimisation advances. This allowed us to escape from trivial solutions (since the weight was initially put on those conditions that were not accomplished in the trivial solution), while guaranteeing that the final solution depends only on true regularities (since the weight was transferred to them during the process). We did generalise the approach by adding F_A , which includes those regularities that are true regularities and can always be applied since they are not trivially satisfied in the initial solution (symmetry is a good example):

$$F(\mathbf{z}) = F_A(\mathbf{z}) + \lambda F_I(\mathbf{z}) + (1-\lambda) F_O(\mathbf{z}) \quad (2)$$

In this approach, undesirable oscillations can happen, causing optimisation algorithms to work badly. In addition, small changes in optimisation parameters can mean falling into local minima during inflation. In other words, introducing false regularities generates a far more complex objective function, which contains more local minima. Hence, the algorithm escapes from the initial solution ($\mathbf{z} = 0$), but with a high risk of creating more local minima.

In the *direct generation* category, the simplest strategy to obtain a tentative 3D model is random inflation, first suggested by Brown and Wang [13]. Surprisingly, our tests show that, provided the shapes are not too complex, optimisation from a randomly chosen tentative model (where all z_i lie in the normalised interval $\max(\Delta x, \Delta y)$) gives topologically valid solutions with a low failure rate of one per fifteen attempts, and, in

case of failure (if the final values of the objective functions clearly differ from zero), subsequent random models will give the right topological result. Since the probability of four consecutive failures is so low that it never happened to us during tests, we believe this approach is an alternative in cases where other tentative model approaches fail.

Preliminary reconstruction [11] belongs to the direct-generation category for obtaining tentative 3D models. Its clearest antecedents were the approaches by Kanatani [42] and Lamb and Bandopadhyay [43]. The process begins by obtaining the prevailing angles from the angular distribution graph of lines, and hence the main directions. A model is then built by assigning a coordinate $z_i=0$ to an arbitrary node (this action fixes the position of the model regarding the plane of the image, but does not entail loss of generality in the 3D reconstruction). The process advances by assigning coordinates z_i to nodes connected to those already known. If the shape has an angular distribution with exactly three main directions, the assignment of coordinates can spread to all the nodes, giving rise to a good tentative model.

3. Description of our reconstruction engine

In our reconstruction engine, called REFER, C++ is used to implement the calculations and management of data. Graphical User Interaction is achieved by a calligraphic interface implemented on the Wintab API (<http://www.pointing.com/FTP.HTM>), under the Windows 2000 operating system, and 3D visualisation is implemented in OpenGL.

The process begins with an interactive session where the designer draws 2D sketches on the tablet and REFER generates rough interactive graph-like drawings that are converted first into beautified line drawings and then into 3D shapes (Figure 4). The output is in DXF format containing a wireframe, surface or solid model according to the sketched geometry.

Insert Figure 4 about here.

A detailed description of our calligraphic interface is not necessary here, since similar systems are described in the literature (see, for instance, the “editor block” in [32]). Its main tasks are the dynamic conversion of strokes into straight segments (every stroke in the sketch of Figure 4 is converted into the corresponding line segment in the line-drawing beautification), searching for connectivity relations (endpoint snapping),

trimming, breaking or joining lines when intermediate vertices are added, and deleting wrong lines.

Below, we explain the solutions provided in our engine to solve the optimisation-based beautification of line drawings, the adjustment of optimisation algorithms, the categorisation of regularities, and the generation of tentative models approach. Both the categorisation of regularities and the generation of tentative models benefit from prior automatic detection of the classification of models. This subject is briefly considered in section 3.4. The generation of tentative models is separated into two subsections: section 3.5 outlines the *axometric inflation* approach, while section 3.6 considers a second approach called *level inflation*.

3.1. Line-drawing beautification

Inflation approaches give good results in 3D reconstruction whenever the input drawing is “geometrically correct”, i.e. it effectively fits an orthogonal projection of a real shape. However, if the input drawing is a rough sketch, with imprecise coordinates, the inflation will give distorted or even tangled shapes. Adding more conditions to the objective function during 3D reconstruction can prevent those tangled shapes, but only with difficulty, while feeding correct line drawings is the best alternative. Calligraphic interfaces are typically configured to manage an interactive conversion of sketched lines into straight segments, to join segments whose ends are close to each other, to ensure parallelism, and so on (see the “analyser block” in [32]). This interactive beautification is efficient in some cases. However, if too many restrictions are active, sketching becomes a difficult task and so free sketching followed by a batch beautification phase becomes an interesting alternative. Additionally, interactive beautification can only just deal with global regularities. In summary, interactive capabilities can be complemented by batch beautification to tidy up the complete sketch after the user finishes the drawing, thus providing a more detailed analysis of the user’s sketch and discovering those design intents that are difficult to detect interactively.

An optimisation-based *batch beautification* process was implemented in REFER to repair the drawing and capture some designer intents in the geometric characteristics of the line drawings produced by the calligraphic interface. First, a refinement is carried out to correct some errors that occurred during interactive beautification. Zero length

lines are removed and missing junctions are properly connected. The final beautified drawing is then obtained by *optimising* one objective function defined as a weighted sum of perceptual cues. A hill-climbing algorithm is used, where the optimisation variables are the coordinates of all the n vertices $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$.

$$F(\mathbf{x}, \mathbf{y}) = \sum \alpha_j R_j(\mathbf{x}, \mathbf{y}) \quad (3)$$

At present, our batch beautification searches for parallelism among the sketched lines that represent “intended” parallel edges. A penalty that is inversely proportional to the convergence angle between each pair of lines is added to the objective function, in order to measure this intention. The weight is:

$$\omega_{ij} = \mu_{0^\circ, 10^\circ}(\cos^{-1}(\underline{\mathbf{u}}_i \cdot \underline{\mathbf{u}}_j)) \quad (4)$$

where $\underline{\mathbf{u}}_i(\mathbf{x}, \mathbf{y})$ and $\underline{\mathbf{u}}_j(\mathbf{x}, \mathbf{y})$ are the unit vectors of i -th and j -th edges, respectively, in the input line drawing, and the practical formulation of the compliance factor is [15]:

$$\mu_{a,b}(x) = \max [0, 1.1 e^{-((x-a)/b)^2} - 0.1] \quad (5)$$

where “ a ” is the target value and “ b ” is the maximum deviation allowable. Hence, a null weight (i.e. an angle greater than b) means that this pair of lines is not to be parallelised.

The analytical formulation to evaluate parallelism is:

$$R_{\text{PARALLELISM}} = \sum_{i=1}^m \sum_{j=i+1}^m \omega_{ij} \cdot [\cos^{-1}(\underline{\mathbf{u}}_i \cdot \underline{\mathbf{u}}_j)]^2 \quad (6)$$

where m is the number of edges in the input line drawing; $\underline{\mathbf{u}}_i(\mathbf{x}, \mathbf{y})$ and $\underline{\mathbf{u}}_j(\mathbf{x}, \mathbf{y})$ are the unit vectors of the current i -th and j -th edges, respectively.

Figure 5 shows a typical sketch and its corrected line drawing. Since the errors cannot be estimated in the drawing, we will highlight the fact that, for example, the initial angle between lines d_1 and d_2 is 3.7° , and the angle between e_1 and e_2 is 2.2° . After running parallelisation-based 2D reconstruction, the greatest divergence between any pair of supposed parallel lines becomes less than 0.01° . The precision of the final model is also below this threshold.

Insert Figure 5 about here.

The second regularity introduced in our beautification stage is collinearity, which is formulated in the following terms:

$$R_{\text{COLLINEARITY}} = \sum_{i=1}^m \sum_{j=i+1}^m \omega_{ij} \cdot [(\cos^{-1}(\underline{u}_i \cdot \underline{u}_j))^2 + \max [(\cos^{-1}(\underline{u}_i \cdot \underline{v}_{i1j1}))^2, (\cos^{-1}(\underline{u}_i \cdot \underline{v}_{i1j2}))^2, (\cos^{-1}(\underline{u}_i \cdot \underline{v}_{i2j1}))^2, (\cos^{-1}(\underline{u}_i \cdot \underline{v}_{i2j2}))^2]] \quad (7)$$

where:

- m is the number of edges in the input line drawing.
- $\underline{u}_i(\mathbf{x}, \mathbf{y})$ and $\underline{u}_j(\mathbf{x}, \mathbf{y})$ are the unit vectors of the current i -th and j -th edges, respectively.
- v_{i1} is the current position of the initial vertex of the i -th edge, v_{j1} is the current position of the initial vertex of the j -th edge, and \underline{v}_{i1j1} is the unit vector from v_{i1} to v_{j1} .
- The weight ω_{ij} is:

$$\omega_{ij} = \mu_{0^\circ, 10^\circ} [(\cos^{-1}(\underline{u}'_i \cdot \underline{u}'_j))^2 + \max [(\cos^{-1}(\underline{u}'_i \cdot \underline{v}'_{i1j1}))^2, (\cos^{-1}(\underline{u}'_i \cdot \underline{v}'_{i1j2}))^2, (\cos^{-1}(\underline{u}'_i \cdot \underline{v}'_{i2j1}))^2, (\cos^{-1}(\underline{u}'_i \cdot \underline{v}'_{i2j2}))^2]] \quad (8)$$

where $\underline{u}'_i(\mathbf{x}, \mathbf{y})$ and $\underline{u}'_j(\mathbf{x}, \mathbf{y})$ are the unit vectors of the i -th and j -th edges, respectively, in the input line drawing; v'_{i1} is the initial position of the initial vertex of the i -th edge, v'_{j1} is the initial position of the initial vertex of the j -th edge, and \underline{v}'_{i1j1} is the unit vector from v'_{i1} to v'_{j1} .

The modules for 3D reconstruction are covered in detail below.

3.2. Adjustment of optimisation algorithms

Reconstruction is highly sensitive to the optimisation parameters. It can easily be verified that when the initial step s_1 of the unidirectional search of the optimisation algorithm is too large, the solutions do not satisfy the psychological assumption of proportionality (Figure 6). Of course, “exact” dimensions cannot be induced from rough sketches or from tidied line drawings. However, the perceptual requisite of keeping proportions is affordable and convenient, since human observers assume that the object

represented in the line drawing must have regular dimensions in all directions [17] [44]. This is because they deem it has been represented by choosing a “general” point of view that clearly enhances the actual proportions.

Insert Figure 6 about here.

We solved the problem by normalising step s_1 with respect to $\max(\Delta x, \Delta y)$. This makes optimisation completely independent of the dimensions in the input line drawing. We suggest the range 10-30% because this is an indirect way of imposing the psychological assumption of proportionality ($s_1 = p \cdot \max(\Delta x, \Delta y)$ / $p \in [0.1, 0.3]$). In addition, we concluded that between five (to get high speed and low precision) and ten (to get low speed and high precision) iteration loops is a good range to obtain good solutions in a reasonable amount of time. In previous works, the decrement in s_i was constant [14], but in our experience this leads to no significant improvement in computational efficiency because the stopping criterion in the algorithm cuts off those levels where no significant improvement is achieved. Nevertheless, avoiding large differences in consecutive steps is a guarantee against failures in the algorithm and, after testing, the best behaviour in the objective function seems to be obtained when decrements fit exponential rather than linear curves: for instance, if the next step is 30-40% of the current step ($s_{i+1} = d \cdot s_i$ / $d \in [0.3, 0.4]$). However, a slower decrement (60-80%) is better, since this improves the probability of escaping from local optima that becomes vital if the tentative model is tangled. Finally, the value adopted by the smallest s_i is clearly related to the final precision, and we recommend it be made directly proportional to the greatest deviation we will accept in order to accomplish the regularities.

3.3. Classification of models

It is widely accepted that there is no general strategy that is always valid to reconstruct *all* objects and that the more general the approaches are, the less successful they will be. Two alternative ways to increase the success rates involve using specific approaches or improving the adjustments of general approaches. Both methods can be applied only after detecting the “nature” of the object to be reconstructed (“identifying the special cases”, in the terminology of Varley and Martin [16]), and then selecting the best fitting particular approach or automatically adjusting the reconstruction configuration. The purpose of detecting the nature of the object is to find the taxonomy of shapes by

grouping the ones with similar perceptual cues. The main difficulty comes from the fact that only two-dimensional information contained in the input line drawing can be used since detection must be done before reconstruction.

At present, our detection system is useful for our purpose, but is not as sophisticated as the categorisation proposed by Varley and Martin [16]. It is based on some simple perceptual rules and, hence, we established a model classification for polyhedral objects, distinguishing between *pyramidal* and *prismatic* polyhedral objects, and “*normalons*”. (Here we define a “normalon” as a 3D extension of Dori’s definition [45] of a normalon, i.e. “a polygon having the property that the angle between any two of its adjacent sides is 90° .” In other words, normalons are what Kanatani [42] identified as rectangular trihedral polyhedra. We use the term “normalons” for brevity, since such polyhedra play a fundamental role in our approach.)

We adopted angular distribution graphs [11] to detect normalons and similar criteria were developed to detect prisms and pyramids. In short, a figure is considered to represent a prism whenever one single direction is dominant, i.e. when the number of edges parallel to one direction is at least 10% greater than the number of edges parallel to any other direction. A figure is considered to represent a pyramid whenever one single vertex is dominant. In other words, when the number of edges connected to one vertex is more than 10% greater than the number of edges connected to any other vertex.

3.4. Categorisation of regularities

Regularities currently in use in optimisation approaches were extensively described and formulated in [15], but determining a *categorisation of regularities* aimed at avoiding ill-conditioned objective functions is still an open problem (some ideas can be found in [16]). Such categorisation is desirable since our goal is to reduce the complexity of the objective function in order to improve the success rate of optimisation-based reconstruction. To this end, the distinction between true and false regularities, already mentioned above, is pertinent. False regularities do not reflect any perceptual cue of the image, while true regularities reflect some property of the image that *is likely* to correspond to some properties in the searched model. In other words, a false regularity is an artefact for obtaining models with some special features that we consider to be

desirable characteristics in the 3D model while they are not tied to any cue identified in the two-dimensional drawing. Conversely, a true regularity is a feature detected in the 2D drawing and is very likely to correspond to some feature in the 3D model.

We will also distinguish between direct and indirect regularities. Regularities are direct when they are related to cues that are similar in nature. For instance, two lines being *parallel* to each other in a 2D drawing are supposed to correspond to two *parallel* edges in the 3D model (because of the parallelism invariance of axonometric projections). Indirect regularities, however, are linked to a property that is different in nature to the property perceived in the image. For instance, two lines perceived as being *parallel* to two main directions in the 2D drawing are likely to represent *perpendicular* edges in the 3D model (since main directions are supposed to be orthogonal to each other). The difference is meaningful since indirect cues can barely be detected while sketching. Regularities are also indirect when they work as elements of a set of cues that provide information about a complex feature (i.e. when a set of symmetry lines determines a symmetry plane). In this sense, the benefits of indirect cues have not yet been fully exploited in the problem of line drawing reconstruction.

The advisability of reducing the number of regularities included in the objective function becomes apparent if the actual relations among them are observed. For instance, some irregularities appear in the upper face of the parallelepiped represented in Figure 7: 1) A face (a,b,c,d), or “flatness” condition; 2) Two parallelism conditions (a//c and b//d); 3) Two perpendicularity conditions ($a \perp b$ and $a \perp d$) detectable by line orthogonality; 4) Four skewed symmetries (with axis e_1 , e_2 , e_3 and e_4); and 5) Four orthogonal corners (ab, bc, cd, da) detectable by line orthogonality. Several dependences among those regularities are obvious: 1) if a and c are parallel, then a and c are coplanar, and 2) if a is parallel to c and if a is perpendicular to b, then b and c are perpendicular.

Insert Figure 7 about here.

We will now go on to analyse the regularities detailed in [15] except “prismatic face”, which is not considered because it is only useful in the detection of a particular kind of curved surfaces (ruled ones) and, hence, falls outside the scope of this paper. New formulations are presented that we consider better describe the cues.

Face planarity. This is a true regularity for polyhedral shapes and probably the most important one, since it virtually eliminates tangles, as predicted by Brown and Wang [13]. Its main drawback is that it becomes a direct regularity only after faces have been detected *in the line drawing*, and this is a complex task when hidden edges are drawn (see [46], [47], [48] and [49]). The alternative of sketching only frontal geometry (without hidden edges) and inferring hidden parts later, when 3D geometry has been given a preliminary interpretation, was explored by [10] and [16].

Skewed facial symmetry. This is a true and interesting regularity, which was extensively explored by Varley and Martin [16]. Nevertheless, we believe that facial symmetries are better considered as indirect cues of a more general model symmetry. More work is needed in this area, although some interesting advances may be found in [50], [51], [52], [53] and [54].

Line parallelism. This is a true and very powerful regularity. Its main disadvantage is that parallelism is invariant only in pictorial (non-perspective) views. Hence, it is a direct cue in pictorial views, and an indirect one in perspective images (see [55]). A more difficult problem is its unacceptable computational cost (for instance, if all the possible parallel lines of a shape like the one in Figure 1 are considered). The use of bundles and thresholds was proposed for such situations [16].

Line collinearity. This is a particular case of line parallelism, unless it is invariant in perspective views too, and may serve to detect alignment constraints among edges and faces. In other words, it is a cue of a true and direct model constraint: alignment. Further improvements in its formulation are required. In the formulation that we consider to be correct (similar to the one described in the beautification section) the computational cost is very high.

Line orthogonality. This is not a projective invariant and, hence, it is indirectly related to a parallelism cue: two edges must be orthogonal if their images are parallel to two directions perceived as being orthogonal to each other. Consequently, it is linked to detection of the main orthogonal directions and some limitations about perspective projections appear as for line parallelism.

In our implementation, lines are considered as being intended to be perpendicular if they are parallel to the main directions. Those main directions are determined by following

the isometric directions criteria (automatically assigned when more than 60% of the edges are parallel to three lines running $120^\circ \pm 5^\circ$ to each other). A more sophisticated criterion is also implemented, which considers the rule of projective consistency for trihedral orthogonal corners, topological constraint (three directions cannot be main directions if they are not simultaneously present in at least one vertex), and so on [56].

In order to impose such parallelism among edges and main directions in the 3D shape, the reconstruction of the axonometric system (see $O_M X_M Y_M Z_M$ in Figure 8) must be performed before optimisation. The axonometric inflation described below is used for this purpose.

Skewed facial orthogonality. This is a particular case of line orthogonality extended to all edges of the same face. It is true for the faces of normalons (rectangular polyhedra).

Line verticality. This is a true regularity. However, it is more concerned with the position of the model than with its shape. In our implementation, we considered that edges that are vertical in the input drawing (parallel to the vertical axis) are “pitched” in the final model. In fact, the *inflation coordinate system* (see $O_I X_I Y_I Z_I$ in Figure 8) is used to reconstruct, while line verticality of the model would have to be referred to the axonometric system (see $O_M X_M Y_M Z_M$ in Figure 8), and would be vertical lines in the sense of being *parallel* to the three dimensional vertical axis of the axonometric system.

Insert Figure 8 about here.

Isometry. This is a variant of the “standard deviation of segment lengths” [13]. Of course, it may be a true regularity, since a human observer assumes that the object represented in the line drawing must have regular dimensions in all directions, but according to projective laws, this isometry cannot usually be measured directly in the drawing. Hence, it is a false, or at best indirect, cue. In addition, mathematical formulation of this regularity would have to be limited to lines that run parallel to the main directions. Finally, normalisation carried out in the regulation of optimisation parameters guarantees a high degree of isometry, making it unnecessary to introduce it explicitly.

Minimum standard deviation of angles (MSDA). This was used very successfully by Marill to introduce an optimisation approach, but it is a false regularity (unless applied to regular polyhedra).

Corner orthogonality. This is a particular case of MSDA, where all angles defined by connected edges are equal to 90° . Consequently, it is a false regularity unless applied to rectangular polyhedra.

Face perpendicularity. In manifolds, this is a particular case of corner orthogonality. Its interest is limited to normalons, where it is a true regularity.

In summary, in the fields of polyhedral blocks, origami objects and wire frame models, face planarity, symmetry and line parallelism are the best regularities, despite the difficulty involved in detecting faces and symmetry in the line drawing. The geometry of the resulting shape may be improved by reinforcing parallelism with line collinearity, line orthogonality and skewed facial orthogonality, unless those regularities make it more difficult to deal with perspective-like and/or sketched line drawings.

Line verticality, isometry, MSDA, corner orthogonality and face perpendicularity can be ignored, since their effects on the model are highly redundant compared to the one provided by planarity, symmetry and parallelism and, in addition, the constraints they add are better introduced indirectly during beautification and tentative model generation. Skewed facial orthogonality, corner orthogonality and face perpendicularity are true in normalons but, in such cases, better strategies for direct reconstruction exist that make them unnecessary.

3.5. Tentative models by axonometric inflation

We adopted the preliminary reconstruction of the direct generation approach described in section 2.3 and improved it by developing an algorithm and expanding the number of types of models to which it applies. Our own version of this approach is called “Axonometric Inflation”.

Normalons (rectangular trihedral polyhedra, whose corners consist of three mutually perpendicular edges) benefit from a rectangularity constraint that greatly simplifies the 3D reconstruction process [42]. A detailed study of axonometric inflation can be found

in [56]. Below, we will summarise it and highlight the strategy proposed to extend the approach to a more general class of “quasi-normalon” shapes.

Let us suppose an orthogonal projection of an orthogonal corner such as A'B'C'D' in Figure 9. Let us define vertex D (the one connected to all three edges in the corner) as the *central vertex* and let us define the rest of the vertices as *lateral vertices*. To reconstruct the model (i.e. to determine the x, y and z coordinates of vertices A, B, C and D) the Cartesian coordinate system associated to inflation is used to trivially obtain the x and y coordinates of all four vertices ($x_A = x_{A'}$, $y_A = y_{A'}$, ...).

Insert Figure 9 about here.

The Z coordinate of the central vertex D can be arbitrarily fixed without loss of generality. Axonometric inflation thus benefits from formulations that determine the angle between each edge (CD) and its own line-segment (C'D') when three orthogonal edges are connected to the same central vertex (D). Relative z coordinates of the *lateral* vertices (A, B, C) are then obtained from the z coordinate of the central vertex (D) in the following way:

$$z_C = z_D \pm L_{C'D'} \cdot \tan(\text{asin}(\sqrt{\cotg(\alpha) \cdot \cotg(\beta)})) \quad (9)$$

where $L_{C'D'}$ is the length of line segment C'D' (CD edge projection) and z_C and z_D are the respective z coordinates of the lateral and the central vertex that determine the reconstructed edge.

To determine the z coordinates of all the vertices in a normalon polyhedron, a specific spanning tree is required that converts already-determined lateral vertices into new central vertices, and in this way does not miss the relative position among vertices and constructs a topologically consistent 3D model. We use a specific version of the Kruskal algorithm [57] to obtain a spanning tree made up of all the edges that connect the successive central vertices. The lateral vertex connected through the longest edge is defined as the central vertex for the new corner, because it is assumed that longer edges are less prone to error than shorter edges (i.e. the dimensional relative errors are smaller, and long lines are more accurately drawn in hand drawing). The process is repeated until all the vertices in the model have been determined. Should converting any of the present lateral vertices into central ones generate a circuit (i.e. if all lateral vertices have

already been visited), the branch is abandoned and the longest as yet unexplored lateral edge is used to begin a new branch. Moreover, particular cases must be studied to prevent numerical inconsistencies during the process (for instance, those that may arise due to trigonometric operations).

The algorithm requires the concurrence of three orthogonal edges in every central vertex. Nevertheless, information about faces is not required and the restriction applies to central vertices *only*. The *valence* (the number of edges concurring at a vertex) of lateral vertices is irrelevant. Consequently, the approach will work if a spanning tree can be obtained where all central vertices have a valence of three and, thus, all vertices of other valences can be determined as laterals. In other words, Axonometric Inflation can be applied to the “extended” class of polyhedra whose spanning tree is a normalon. In those quasi-normalon models (like the one in Figure 10), the temporary elimination of all line-segments that are non-parallel to any of the three main directions determines an *equivalent* normalon if: a) no vertex disappears during the conversion, and b) the graph is still connected. In Figure 10, the spanning tree begins at vertex 10 and the coordinates of vertices 5, 7 and 9 can be determined. The next central vertex should be 9 and this will therefore allow 8 and 11 to be calculated. When continuing the process, however, neither 11 nor 8 can be central since their valences are two. Evaluation continues by opening the branch connecting 10 to 5, which allows vertices 3 and 6 to be obtained. We then move to central vertex 6 to determine vertex 4. A new move to 4 allows the calculation of 2. Finally, the evaluation of vertex 1 is carried out through vertex 3, concluding the definition of the model.

Insert Figure 10 about here.

Sometimes, in spite of achieving an equivalent normalon, axonometric inflation cannot be applied if certain vertices are not accessible through some valid spanning tree. This is the case when some junctions are only connected to other junctions of valence below three. For instance, in Figure 11, vertex 3 is only connected to vertex 4, which cannot be central because its valence is two. The same happens to vertices 6 and 7.

Insert Figure 11 about here.

When the valence of a central vertex is less than three, equation (9) cannot be used. Nevertheless, the assumption that the model is a normalon has already been made. Hence, adding fictitious line-segments is consistent with the assumption and solves the problem. These fictitious line-segments are defined to be of unit length and oriented in accordance with the main directions not yet present in the vertex. In Figure 12, fictitious edges 4-F₁, 12-F₂ and 6-F₃ allow vertices 3, 6 and 7 to be determined.

Insert Figure 12 about here.

When vertices of a valence greater than three appear, the approach will still work if a “valid” spanning tree can be obtained, i.e. a tree where those vertices are not central. When this is not possible, we have confined ourselves to obtaining one of the potential models by randomly choosing three of the edges that converge at the vertex.

3.6. Tentative models by level inflation

Reconstruction by level inflation of vertices is a new perceptual method of *direct* (non-iterative) inflation, where z coordinates are determined depending on the shapes of the junctions. The shapes of the junctions have been extensively used in labelling approaches to validate the figures to reconstruct [58], [34], [35], [59], [60], [61]. Our goal is very different, since labelling requires a complete line drawing, while in modelling-by-sketches incomplete drawings must be processed. The shapes of the junctions are not the key of the proposed method, which instead consists in accepting the fact that there is a high degree of correspondence among the shape of each junction and the z coordinate of its corresponding vertex in the model. Therefore, it is supposed that the shapes of the junctions can be ordered in increasing “levels”, which are matched to growing z coordinates. The correspondence between shapes and levels can be intuitively justified by examining Figure 13, where the input image (contained in the XY plane) and the psychologically valid model of a cube are shown. It can be seen how the six levels have been labelled from 0 up to 5.

Insert Figure 13 about here.

The assignment of z coordinates to levels is sensitive to the “pitch” angle of the model (clearly visible in the lateral view YZ in Figure 14), because edges that are parallel to the vertical axis in the input image are inclined in the final model. As the pitch angle

varies, different projections of the same object are obtained. Bowing the object in Figure 13 a little more (increasing its pitch) would make the z coordinates of vertices labelled 3 greater than the z coordinates of level 4 vertices, although their shapes (their appearance in the image) will remain the same until a very large change in pitch occurs.

The “yaw” and “roll” parameters modify the views of an object in the same way pitch does. The yaw angle is measured from the Z axis to the horizontal projection of line BD. We observe that the “symmetric” vertices (for instance, C and E) have different z coordinates when this angle clearly differs from zero.

Our psychologically rooted assumption was that the observer tends to make the smallest pitch and yaw movements as a compromise to get a general point of view while maintaining symmetry. In addition, roll is left at zero to maintain model verticality. The six defined shapes of junctions that are obtained as a result of these assumptions are shown in Figure 14 in ascending order of z values.

Insert Figure 14 about here.

One additional consideration is that both positive and negative pitch angles give psychologically plausible solutions, known as Necker reversals. We opted for a positive pitch (i.e. the upper face is visible) without loss of generality in the reconstruction, because the reversal is obtained by changing all z coordinates from positive to negative (Figure 15).

Insert Figure 15 about here.

We assume that, as long as the designer deliberately chooses the appropriate point of view to show the real object proportions, the range of depth ($\Delta z = Z_{\max} - Z_{\min}$) must be similar to width ($\Delta x = X_{\max} - X_{\min}$) and height ($\Delta y = Y_{\max} - Y_{\min}$). Moreover, uniform gradation is the simplest strategy to assign z coordinates to intermediate levels:

$$z_i = i * (\Delta z/n) = i * (\max (\Delta x, \Delta y) / n) \quad (10)$$

where n is the number of level jumps (total number of levels minus one), the value of which is 5 according to the classification in Figure 14.

Clearly distorted models appear in cases where the same shapes (i.e. an inverted Y) appear in two or more vertices with obviously different depths (like A, D and E in Figure 16). This was solved by a new *incremental* level inflation approach where z coordinates of intermediate levels do not only depend on their absolute level, but on the relative one (the difference of levels between the calculated vertex and another vertex connected to it), and the distance to the vertices to which it is connected. This proposal is again justified assuming the psychological tendency to proportion.

Insert Figure 16 about here.

Consequently, the Z_B coordinate of a vertex B is obtained incrementally, relative to the z coordinate of another vertex A, to which B is connected by an edge:

$$Z_B = Z_A + AB * (\text{Level}_B - \text{Level}_A) * \max(\Delta x, \Delta y) / n \quad (11)$$

where AB is the length of the projection of the edge connecting A to B. $\text{Level}_B - \text{Level}_A$ is the increment (number of jumps) from one level to the other.

The *incremental* level inflation formulated in (11) can be easily extended to determine all vertices by successively calculating as many vertices as necessary. A spanning tree is used both to go from general (big) to particular (small) details and to ensure repeatability in the tentative inflation. We use a specific version of the Kruskal algorithm, starting out from the most central level 0 vertex (the one closest to the barycentre of the figure) and continuing to the vertex connected with the current one by the longest as yet unvisited edge.

A detailed study of level inflation can be found in [62]. We will conclude this section by simply underlining their main extensions.

Predominant edges. The classification of levels shown in Figure 14 includes the corners of many drawings, when their predominant orientation is vertical. To cope with other orientations, our algorithm detects the predominant direction of every input drawing and performs the vertex matching after the rotation, i.e. the angle measured from this predominant direction with respect to the vertical has been neutralised. The most frequent direction is selected as the predominant one. Should two or more orientations appear with the same frequency, the one closest to 90° (which corresponds to lines drawn vertically and bottom-up) or 270° (lines drawn vertically and top-down)

acts as the predominant edge. The term “vertical” refers to the input drawing, and depends on the calligraphic interface, but it is normally configured to match the vertical direction on the screen where the sketch is made. In Figure 17a, both d_A and d_B appear three times. However, d_B is closer to the vertical and is therefore selected as the predominant edge.

Insert Figure 17 about here.

Vertices with more than three edges.

When vertices of a valence above three appear, the approach will still work if a “valid” spanning tree can be obtained, i.e. a tree where all central vertices have a valence of 3. When this is not possible, we have confined our approach to obtaining one of the potential models by randomly choosing three of the edges that converge at the vertex. In fact, the choice is not random but heuristic. When four or more edges meet at a vertex, the two edges encompassing the rest are selected (edges 1 and 5 in Figure 18) and the third edge will be the one closest to the bisecting line of the angle determined by the two edges chosen in the previous step (edge 2 in Figure 18).

Insert Figure 18 about here.

Vertices with two edges. When vertices occur in which two edges meet, we implement a particular version of the creation of 3-edge vertices strategy [10]. The level is then determined according to the shape of the “extended” junction and the tentative model is obtained. In our approach, however, only two-dimensional information about vertices and edges is available. Hence, our criterion is to assume that the third edge is occluded by a face determined by the two visible edges, but the face itself is not determined unless it becomes necessary. Initially, the third edge is simply assumed to be coincident with the bisector of the angle defined by the two edges (see vertices V_1 , V_2 and V_3 in Figure 19). Bisecting the smallest angle between the two lines is a simple but not always valid rule. A more sophisticated rule determines when the two-dimensional vertex belongs to the apparent contour (see vertex V_1 in Figure 20). In this case, the third edge is the bisector that goes *inside* the apparent contour. The faces must be determined only if vertices with two edges do not belong to the apparent contour.

Selection of the bisector that goes inside the face the other two edges belong to then takes place.

Insert Figure 19 about here.

Insert Figure 20 about here.

4. Examples and Discussion

During tests, our modelling-by-sketch system proved to be user-friendly and useful for practical purposes. The system is able to reconstruct a large variety of shapes in an almost automatic manner (see Figures 1 to 6). Nevertheless, the user's decisions are sometimes required and, under extreme situations, the system fails and the user must delete the part of the sketch that caused the failure before the system can find the appropriate solution. For instance, in Figure 21, edge A has a clearly distorted parallelism relative to the intended parallel edge B and, in addition, it is unlikely to be parallel to edge C; the system was unable to solve this conflict (the intended solution, in Figure 21e was so tangled that OpenGL was unable to draw the front face!). By simply deleting edge A and sketching it again, the correct shape was obtained (Figure 21f).

Insert Figure 21 about here.

To prevent erroneous cue detections and, hence, ill-functioning such as that described in Figure 21, the “bundling method” proposed by Varley and Martin [16] seems an interesting alternative to the weighting approach we used.

One limitation that was clearly apparent during tests was the absence of global symmetry regularity. It is more or less masked in complex shapes, where the redundant effect of indirect cues imposes some degree of symmetry, but it becomes apparent when such indirect cues are not present. Figure 22 illustrates how both interactive and batch beautification reinforce connectivity, parallelism and collinearity, while 3D reconstruction is concerned with face planarity. The output model is not good, however, since the clearly perceivable cue of symmetry was never present during reconstruction.

Insert Figure 22 about here.

The system works in real time and our tests show that users feel quite comfortable during sketching and direct reconstruction. However, optimisation processes are bottlenecks because the user must wait (typically from some seconds up to a few minutes, e.g. on a Pentium III based personal computer fifteen seconds were required for parallelism and collinearity tidying in the example in Figure 21). Implementation of efficient optimisation algorithms and elimination of screen updates to monitor the process would significantly decrease these times. However, direct reconstruction of particular typologies through specific approaches, like axonometric inflation, seems more promising. In fact, axonometric inflation gives final models when the inputs are geometrically valid line drawings. A subsequent optimisation process is only required to “refine” imperfect models obtained when the input drawing is an imperfect sketch which could not be completely tidied during beautification. In short, optimisation-based beautification is our present bottleneck, since calculation times are clearly unacceptable for an interactive session. However, optimisation-based 3D reconstruction performance is not a difficult problem, since it is only required in a small set of complex shapes. Nevertheless, discussions on beautification and 3D reconstruction performances are detailed separately below.

4.1. Testing and evaluation of the beautification phase

Typically, during sketching, interactive beautification allows lines to be trimmed and rotated if it is necessary to get appropriate junctions, and almost parallel lines can be interactively parallelised (Figures 23a and 23b). However, interactive beautification only propagates in a “local” way. For instance, Figure 23b shows that the pairs of neighbouring edges ($E_1 - E_3$, $E_3 - E_5$, $E_6 - E_7, \dots$) have been parallelised, but the pair $E_3 - E_5$ was not made parallel to the pair $E_6 - E_7$, since connectivity was the dominant constraint, and both connectivity and parallelism were only achievable through a global change in the geometry. In contrast, batch beautification outputs a globally corrected line drawing, where perceived cues are accomplished in a geometrically consistent way (i.e. a global parallelisation is achieved in Figure 23c). Moreover, this parallelisation is achieved even if interactive parallelisation was not performed previously (Figure 23f).

Additionally, interactive beautification sometimes forces a different final model. If parallelism detection is done in both an interactive and a batch system, the final models

(Figures 23c and 23f) will differ. As shown in Figure 23a, if lines are drawn in the sequence $E_1, E_2, E_3, E_4, \dots$, the interactive approach will force E_3 to be parallel to E_1 as it is being drawn (this implies a dynamic correction of vertex A). Subsequently, when the sketching continues and line E_4 is drawn, its position is determined by the already-moved vertex A. In other words, the designer's gesture has been dynamically modified. The result is a final narrow upper step (Figure 23c), which is different from the one intended by the designer although easily obtained through a similar sketching session where interactive beautification of parallel lines was disabled and finally batch parallelisation was performed (Figures 23d, 22e and 23f).

Insert Figure 23 about here.

We conclude that interactive beautification with small thresholds, followed by batch beautification, outputs line drawings geometrically correct while allowing more freedom during sketching than interactive beautification modules do when used alone.

Shapes where parallelism is dominant can be reconstructed in two dimensions so as to obtain well-conditioned input line drawings. The main advantage is that a corrected drawing, where parallelism is present, may be easily converted into good tentative models that guarantee parallelism by axonometric inflation.

As a final consideration, it must be pointed that, due to the regularities (i.e. collinearity and parallelism), perceptual cues are limited in optimisation-based beautification: they are just *probable* properties of the final model. Accidental points of view can cause erroneous detections of false parallelisms if two non-parallel edges are accidentally projected as parallel lines in the drawing. Figure 24 shows an example where edge 5-6 would be considered parallel to edges 1-4, 2-3 and 7-8, although it should really be interpreted as a crossing edge.

Insert Figure 24 about here.

Reinforcing such false parallelism during beautification was a mistaken goal, but caused no further problems since the output line drawing was perfectly processed during the subsequent 3D reconstruction (i.e. appropriate checks were applied to detect such singularities during axonometric inflation). In the actual case of Figure 24, vertex 2 has

an equivalent valence of 2 and is avoided in the spanning tree. Thus, the “correct” reconstruction is obtained.

4.2. 3D reconstruction test and evaluation

Normalisation of optimisation parameters is effective. A large set of more than one hundred and fifty examples with dimensional ranges (Δx , Δy) varying from (1.17, 1.12) to (418.85, 435.29) were successfully solved.

Optimisation is very sensitive to step size and the number of steps. Adjusting these parameters to the general values proposed in section 3 results in a reasonably robust optimisation process. Nevertheless, “erratic” behaviours have sometimes been detected. For instance, in model J by Leclerc and Fischler (see Figure 3) a step s_1 in the ranges 20-37 % and 65-100 % leads to a model with the “right” geometrical form (that is, a hexagonal prism), while different ranges convey up to three different “tangled” models corresponding to different local minima. Another example (example A by Marill) is presented in Figure 25 (their input drawing was shown in Figure 6a).

Insert Figure 25 about here.

Figure 26 shows the influence exerted by perturbations of the z coordinate of vertex 4 (lower rear central corner in Figure 25) on the objective function (where only face planarity applies), as evaluated in the neighbourhood of the optimum shape. Figure 26 shows how close two local optima and the global optimum are and how a slight distortion of one single variable drives the solution to local optima. This suggests that, unless adjusting the parameters that control the behaviour of the algorithm is really necessary, it is not sufficient to prevent the solution from falling into local minima. In our opinion, good tentative models are required to prevent this difficulty.

Insert Figure 26 about here.

The tentative models obtained by the axonometric inflation algorithm have proved to be valid for all normalons of valence 3 and almost every type of quasi-normalons; some representative examples are shown in Figure 27. The automatic trimming of lines is also apparent in Figure 27a1. The gesture made to delete a line is visible in Figure 27b1 and the effects of the batch beautification of vertices and line collinearity are marked in the tentative model in Figure 27c3.

Insert Figure 27 about here.

Strictly speaking, axonometric inflation is only valid when the image corresponds to an orthogonal axonometric projection. Nevertheless, for certain oblique axonometric views where the projection does not deviate excessively from the orthogonal, the proposed method does succeed in reconstructing approximate models.

In Cavalier's axonometric views (Figure 27d1), Axonometric Inflation is unable to obtain a model. Nevertheless, an artifice consisting in arbitrarily adding a threshold to conflicting angles is employed. The values proposed are 91° instead of 90° and 134° instead of 135° . This is a good compromise, because a threshold below 1° would enlarge oblique faces, i.e. the L/H proportion in Figure 27d3 is clearly greater than the perceived proportions in the Cavalier drawing. Furthermore, a greater threshold would rapidly increase local distortions (such as the one observed in vertex A in Figure 27d3).

In short, Axonometric Inflation gives exact results when applied to normalon models that are projected according to orthogonal and parallel projections. Under such conditions, "perfect" models are obtained and further 3D reconstruction is not required. In this context, "perfect" refers to a 3D model that is perceived to be equivalent to the input drawing. Its geometry must be the same as the majority of observers would perceive from the input drawing, and the dimensions and other metric properties that can be perceived in an approximate fashion (such as parallelism and proportion) must be achieved qualitatively. Prior face detection is not required, and an extended class of "quasi-normalons" can also be successfully reconstructed.

Fixed level inflation gives good results for convex polyhedra with a high degree of regularity, particularly in prismatic and pyramidal shapes, which are successfully detected by our rudimentary classification system. However, results of the *incremental* level inflation are better when more complex shapes appear. For instance, the incremental inflation of the sheet metal part in Figure 28a1 (shown in Figure 28a3) is clearly less distorted than in the fixed inflation of Figure 28a2. Typically, level inflation reduces the iterative optimisation loop by an order of magnitude. Nevertheless, the final models suffer the same problems as before, i.e. complex and indirect regularities (like global symmetry) are clearly absent (Figure 28a4). The model in Figure 28b1 was reconstructed through incremental level inflation (Figure 28b2) followed by

optimisation, where parallelism, planarity, collinearity and line orthogonality were active regularities. Figure 28b3 also illustrates the effect of a lack of both local and global symmetry regularities.

Insert Figure 28 about here.

The use of two consecutive optimisation processes is another alternative, where the first optimisation involves only regularities that are not trivially accomplished in the input drawing and outputs a tentative model. For instance, corner orthogonality was employed to reconstruct the sketch in Figure 28c1. The tentative model obtained in this way (Figure 28c2) was then improved by including a number of other regularities (line parallelism, collinearity and face planarity in the case of Figure 28c3). This approach is an alternative to the balanced optimisation function described in (2), which does not include false regularities (with their associated local minima) and prevents undesirable oscillations in the numerical calculations, since the objective function remains the same throughout each optimisation process. This alternative gives good results when the model has some main directions (see comparison between direct optimisation in Figure 28d3) and an initial optimisation that includes line orthogonality (Figures 28d2), followed by an optimisation of face planarity and line orthogonality (Figure 28d4). In fact, it is a sort of “handmade” multicriteria optimisation. Perhaps a more formal multicriteria optimisation approach should be considered. Nevertheless, our early (though incomplete) attempts to formalise a multicriterion algorithm were discouraging [63].

5. Conclusions

Optimisation is a good approach, but still too academic to be used as an engine for a sketch-based modelling system, since it is too sensitive to the imperfections inherent in sketches. Improvements are required to take the method towards full automation (diminishing the need for user interaction) and greater robustness (increasing the success rate). An adjustment of the optimisation parameters improves the efficiency even for very simple numerical methods. We used a simple descent algorithm to demonstrate the benefits of correct adjustment and good tentative models even with simple algorithms. Better optimisation algorithms exist and they improve computational

efficiency but, to the contrary to what is claimed in some previous works, they cannot solve the main problem on their own; they are faster than simple descent algorithms, but they get the same tangled shapes.

We apply two complementary adjustment strategies: normalisation of all optimisation parameters according to the range of the dimensions of the line drawing, and definition of sets of optimisation parameters that better fit the “nature” of models detected by our own automatic procedure.

A reduced set of the best regularities, in conjunction with some tentative inflation methods, have been proposed in order to avoid the classical problems of local minima in reconstruction by optimisation. This solves a difficult problem, because the original line drawing is a trivial local minimum for many regularities (such as face planarity) and many local minima are present in the feasible region, since objective functions are ill-conditioned. What we achieve with tentative inflations is to escape from trivial optima, whilst working with a better objective function, thus increasing the success rate and reducing calculation time.

The efficiency of direct inflation approaches clearly depends on the model topology but, in our experience, automatic detection of the more generic typologies is easy to implement and tentative models improve the optimisation process with simple descent algorithms.

To sum up, tentative inflation approaches allow an initial model to be defined by applying some simple rules. These approaches proved to be efficient for a wide range of polyhedral models and are quite easy to implement and run fast, even for complex models.

The strategies we employed to obtain tentative models were presented. They were implemented and tested and, according to our experience, axonometric inflation gives excellent results for normalon and quasi-normalon models, and level inflation seems to be a good approach for obtaining tentative models from prismatic models. Hence, good quality final models can be obtained at a reduced cost for such typologies.

Future work must start out from the fact that optimisation processes are bottlenecks and thus there is a need to explore alternative strategies, for instance, parallel processing of 3D optimisation in the background during the sketching session. Another approach to be

explored should be the substitution of numerical optimisation algorithms by an “algebraic constraint manager”, configured to solve the set of “soft” constraints embedded in the regularities derived from image cues (i.e. 2D and 3D DCM from D-Cubed, see <http://www.d-cubed.co.uk/index.htm>).

Acknowledgements

This work was partially supported by the authors' regional government ("Generalitat Valenciana"), under its grant programme for Technological Research and Development (Project GV97-TI-04-39, entitled "3D Object Reconstruction from 2D Representations").

It was also partially supported by Universitat Jaume I ("Plan 2002 de promoció de la investigació a l'UJI", Project P1-1B2002-08, entitled "From sketch to model: new user interfaces for CAD systems").

REFERENCES

- [1] Roberts L.G. *Machine Perception of Three-Dimensional Solids*. PhD Thesis. Massachusetts Institute of Technology. 1963.
- [2] Johnson T.E. *Sketchpad III. Three Dimensional Graphical Communication with a Digital Computer*. PhD Thesis. Massachusetts Institute of Technology. 1963.
- [3] Sugihara K. *Machine interpretation of Line Drawings*. The MIT Press. 1986.
- [4] Nagendra I.V. and Gujar U.G. 3-D Objects From 2-D Orthographic Views – A Survey. *Computers & Graphics*. Vol. 12, No. 1, pp. 111-114. 1988.
- [5] Wang W. and Grinstein G.A. Survey of 3D Solid Reconstruction from 2D Projection Line Drawings. *Computer Graphics Forum*. Vol. 12, No 2, pp. 137-158. 1993.
- [6] Ullman D.G, Wood S., Craig D. The Importance of Drawing in the Mechanical Design Process. *Computers & Graphics* Vol 14, No 2, pp263-274. 1990.
- [7] Cugini U., The problem of user interface in geometric modelling, *Computers in Industry* 17, 335-339, 1991.
- [8] Dori D. From engineering drawings to 3D CAD models: are we ready now? *Computer Aided Design*, vol. 27 No. 4, pp. 243-254. 1995.
- [9] Ullman D.G. Toward the ideal mechanical engineering design support system. *Research in Engineering Design*. No 13. pp 55-64. 2002.
- [10] Grimstead I.J. and Martin R.R. Creating solid models from single 2D sketches. *Proc 3rd Symposium on Solid Modeling and Applications, ACM Siggraph*. pp. 323-337, 1995.
- [11] Lipson H. and Shpitalni M.A. New Interface of Conceptual Design Based on Object Reconstruction From a Single Freehand Sketch. *Ann CIRP*. Vol. 44, No. 1, pp. 133-136. 1995.
- [12] Lipson H.; Shpitalni M. Correlation-Based Reconstruction of a 3D Object From a Single Freehand Sketch" *AAAI Spring Symposium Series- Sketch Understanding* 2002.

- [13] Brown E.W. and Wang P.S. Three-dimensional object recovery from two-dimensional images: A new approach. *Proc. SPIE. Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, vol 2904, pp. 138-147, 1996.
- [14] Leclerc Y. and Fischler M. An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames. *International Journal of Computer Vision*. Vol. 9, No. 2, pp. 113-136. 1992.
- [15] Lipson H. and Shpitalni M. Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing. *Computer-Aided Design*. Vol. 28, No. 8, pp. 651-663. 1996.
- [16] Varley P.A.C. and Martin R.R. Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch - Frontal Geometry and Sketch Categorisation. *Proc. First UK-Korea Workshop on Geometric Modeling and Computer Graphics, Kyung Moon Publishers*. pp.113-128. 2000.
- [17] Palmer S.E. *Vision Science. Photons to Phenomenology*. The MIT Press. 1999.
- [18] Schweikardt E. and Gross M.D. Digital Clay: deriving digital models from freehand sketches. *ACADIA '98*, Seebom T. and Wyk S. V. eds., Quebec City, Canada, pp. 202-211, 1998.
- [19] Cera C.D., Regli W.C., Braude I., Shapirstein Y. and Foster C.V. A Collaborative 3D Environment for Authoring Design Semantics. *IEEE Computer Graphics and Applications*, Vol. 22 No. 3, pp.43-55. 2002.
- [20] Sturgill M., Cohen E., and Riesenfel R.F. Feature-Based 3-D Sketching for Early Stage Design. *Proc. Am. Soc. Mechanical Engineers (ASME) Computers in Eng. Conf.*, ASME Press, New York, pp. 545-552, 1995.
- [21] Eggli L., Hsu C.Y., Bruederlin B.D. and Elber G. Inferring 3D Models from Freehand Sketches and Constraints. *Computer-Aided Design*, vol. 29, no. 2, pp. 101-112. 1997.
- [22] Hearst M.A. Sketching Intelligent Systems. *IEEE Intelligent Systems*, vol. 13, no. 3, May/June, pp. 10-19. 1998.

- [23] Lipson H. and Shpitalni M. Conceptual Design and Analysis by Sketching. *Artificial Intelligence for Eng. Design, Analysis, and Manufacturing (AIEDAM)*, vol. 14, no. 5, pp. 391-402. 2000.
- [24] Qin S.F., Wright D.K., and Jordano I.N. From On-line Sketching to 2D and 3D Geometry: A System Based on Fuzzy Knowledge. *Computer-Aided Design*, vol. 32, no. 14, pp. 851-866. 2000.
- [25] Sutherland I.E. Sketchpad: a man-machine graphical communication system. *Proc. Spring Joint Computer Conference. AFIPS*. pp. 329-346, 1963.
- [26] Zeleznik R.C., Herndon K.P. and Hughes J.F. SKETCH: An interface for sketching 3D scenes. *SIGGRAPH'96 Conference Proceedings*, pp. 163-170, 1996.
- [27] Bloomenthal K., Zeleznik R.C. et al. SKETCH-N-MAKE: Automated machining of CAD sketches. *Proceedings of ASME DETC'98*, pp. 1-11, 1998.
- [28] Pereira J., Jorge J., Branco V. and Nunes F. Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons. *WSCG'2000. Conference Proceedings*, Skala V. Ed., 2000.
- [29] Igarashi T., Matsuoka S. and Tanaka H. Teddy: a sketching interface for 3D freeform design. *ACM SIGGRAPH 99 Conference Proceedings*, pp. 409-416, 1999.
- [30] Oh B.S. and Kim C.H.. Progressive 3D reconstruction from a sketch drawing. *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, pp. 108 -117, 2001
- [31] Pavlidis T. and Van Wyk C.J.. An Automatic Beautifier for Drawings & Illustrations. *Computer Graphics*, 19 (3). pp. 225-234. 1997.
- [32] Jenkins D.L. and Martin R.R. Applying constraints to enforce users' intentions in free-hand 2-D sketches. *Intelligent Systems Engineering*, 1 (1). pp 31-49, 1992.
- [33] Igarashi T., Matsuoka S., Kawachiya S. and Tanaka H. Interactive Beautification: A Technique for Rapid Geometric Design. *Proceedings of the ACM Symposium on User Interface Software and Technology UIST'97*, pp. 105-114", 1997.

- [34] Huffman, D. A. Impossible objects as nonsense sentences. *Machine Intelligence* Edinburgh University Press, pp. 295-323. 1971.
- [35] Clowes, M.B. 1971. On seeing things. *Artificial Intelligence*, vol 2, pp. 19-116.
- [36] Turner A, Chapmann D. and Penn A. Sketching space. *Computers & Graphics* No 24, pp 869-879. 2000.
- [37] Naya, F; Conesa, J; Contero, M; Company, P. and Jorge J. Smart Sketch System for 3D Reconstruction Based Modeling. *Lecture Notes in Computer Science*. Vol. 2733. pp 58-68. 2003.
- [38] Marill T. Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects. *International Journal of Computer Vision*. Vol. 6, No. 2, pp. 147-161. 1991.
- [39] Baird L. and Wang P. 3D object recognition using Gradient descent and the Universal 3-D array grammar. *Proc. of SPIE. Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, vol. 1607, pp. 711-718. 1991.
- [40] Baird L., and Wang P. 3D Object Perception Using Gradient Descent. *Int. Journal of Mathematical Imaging and Vision (IJMIV)*, 5, 111-117, 1995.
- [41] Company P., Gomis J.M. and Contero M. Geometrical Reconstruction from Single Line Drawings Using Optimization-Based Approaches. *WSCG'99. Conference proceedings*, Volume II, pp. 361-368. 1999.
- [42] Kanatani K. The Constraints on Images of Rectangular Polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, N° 4, pp.456-463. 1986.
- [43] Lamb D. and Bandopadhyay A. Interpreting a 3D object from a rough 2D line drawing. *Proceedings of Visualization'90*, pp.59-66. 1990.
- [44] Hoffmann D. *Visual Intelligence. How we create what we see*. Norton Publishing. 2000.
- [45] Dori D. Dimensioning Analysis. Toward Automatic Understanding of Engineering Drawings. *Comm. of the ACM*, vol. 35 No. 10, pp. 92-103. 1992.

- [46] Courter S.M. and Brewer J.A. Automated Conversion of Curvilinear Wire-Frame Models to Surface Boundary Models: A Topological Approach. *Comm. of ACM*, Vol. 20 (4). pp. 171-178. 1986.
- [47] Shpitalni M. and Lipson H. Identification of Faces in a 2D Line Drawing Projection of a Wireframe Object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 18, No. 10, pp. 1000-1012. 1996.
- [48] Liu J.Z., and Lee Y.T., "A Graph-based method for Face Identification from a Single 2D Line Drawing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 10, pp 1106-1119, 2001.
- [49] Liu J.Z., Lee Y.T., and Cham W.K., "Identifying Faces in a 2D Line Drawing Representing a Manifold Object", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 12, pp 1579-1593, 2002.
- [50] Langbein F.C., Mills B.I., Marshall A.D. and Martin R.R. Approximate Symmetry Detection for Reverse Engineering. *Proc. Sixth ACM Symposium on solid Modelling and Applications*, pp 206-215, 2001.
- [51] Tate S.J. and Jared G.E.M. Recognising symmetry in solid models. *Computer-Aided Design*. Vol. 35, pp 673-692, 2003.
- [52] Piquer A., Company P and Martin R.R. Skewed mirror symmetry in the 3D reconstruction of polyhedral models. *International Conference in Central Europe on Computer Graphics, (WSCG'2003)*, 2003, *Journal of WSCG. Volume 11, Number 3*, pp. 504-511. 2003.
- [53] Piquer, A., Martin, R.R. and Company, P. Using skewed mirror symmetry for optimisation-based 3D line-drawing recognition. *GREC'2003, Proceedings of the Fifth IAPR International Workshop on Graphics Recognition*, pp. 182-193, 2003.
- [54] Piquer A. Percepción artificial de dibujos lineales. PhD. Thesis. Univ. Jaume I, Castellon, Spain, 2003. (in Spanish only)
- [55] Kuzo P. and Macé P. "Correction of a 2D Sketch for 3D Reconstruction". *WSCG'99. Conference proceedings*, Vol 1, pp. 148-155. 1999.

- [56] Company P., Conesa J and Aleixos N. *Axonometric Inflation in Line Drawings Reconstruction*. Regeo Technical Report 01/2001 (www.tec.uji.es/d/regeo). 2001.
- [57] Gibbons A. *Algorithmic Graph Theory*. Cambridge University Press. 1985.
- [58] Guzmán A. Decomposition of a visual scene into three-dimensional bodies. *AFIPS American Federation of Information Proceedings Fall Joint Computer Conference*, Vol. 33, pp. 291-304. 1968.
- [59] Thorpe C. and Shafer S. Correspondence in line drawings of multiple views of objects. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Vol. 2. 1983.
- [60] Malik J. Interpreting line drawing of curved objects. *International Journal of Computer Vision*. vol 1, pp.73-103. 1987.
- [61] Martí E., Regincós J., López_Krahe J. and Villanueva J.J. Hand line drawing interpretation as three-dimensional objects. *Signal Processing*, Vol. 32, pp. 91-110. 1993.
- [62] Company P., Conesa J and Contero M. *Tentative Level-Inflation in Line Drawings Reconstruction*. Regeo Technical Report 02/2001 (www.tec.uji.es/d/regeo). 2001.
- [63] Conesa J. Reconstrucción geométrica de sólidos tridimensionales utilizando técnicas de optimización. PhD. Thesis. Univ. Politécnica de Cartagena, 2001. (in Spanish only)

VITAE

Pedro Company is a full professor of Engineering Graphics and the head of the Technology Department at Universitat Jaume I in Castellón, Spain. His research interests include technical drawing, computer-aided design, computer graphics, geometrical reconstruction and calligraphic interfaces. He received an MSc and a PhD in Mechanical Engineering from the Polytechnic University of Valencia, Spain.

Manuel Contero is an associate professor of Engineering Graphics at the Polytechnic University of Valencia, Spain. His research interests include calligraphic interfaces and modelling methodologies for CAD applications, concurrent engineering, and product data quality models. He received an MSc and a PhD in Electrical Engineering from the Polytechnic University of Valencia, Spain.

Julian Conesa is an associate professor of Engineering Graphics at the Polytechnic University of Cartagena, Spain. His research interests include geometrical reconstruction, calligraphic interfaces, computer-aided design and computer graphics. He received an MSc and a PhD in Mechanical Engineering from the Polytechnic University of Cartagena, Spain.

Ana Piquer is an assistant professor of Engineering Graphics at Universitat Jaume I in Castellón, Spain. Her research interest focuses on three-dimensional reconstruction, computer graphics, computer-aided design and technical drawing. She received an MSc in Manufacturing Engineering from the Polytechnic University of Valencia, and a PhD in Mechanical Engineering from the Universitat Jaume I in Castellón, Spain.

FIGURE CAPTIONS

Figure 1. Wire-frame building structure.

Figure 2. Polyhedral skeleton of a plotter support.

Figure 3. Inflation process. Some intermediate “tangled” models, together with the good one, are shown in this exploded view of example J by Leclerc and Fischler.

Figure 4. Four stages during a modelling-by-sketch session: input sketch, intermediate interactive generated line drawing, batch-beautified line drawing and 3D model.

Figure 5. Input sketch with intermediate, interactively generated line drawing, batch beautified line drawing showing parallelised lines and 3D model.

Figure 6. Big steps in Hill-Climbing lead to non-proportional models. (a) the original line drawing, (b) psychologically plausible model (obtained with $s_1 = 0.2 \max(\Delta x, \Delta y)$), and (c) non-proportional model (obtained with $s_1 = 2.0 \max(\Delta x, \Delta y)$).

Figure 7. Regularities in the line drawing of a parallelepiped.

Figure 8. Inflation versus model reference systems.

Figure 9. Axonometric Inflation of an orthogonal corner.

Figure 10. Transformation from (a) quasi-normalon to (b) equivalent normalon, by deleting dotted lines, and defining a spanning tree such that all central vertices have a valence of 3.

Figure 11. First transformation from (a) quasi-normalon to (b) non-solvable normalon.

Figure 12. Solvable equivalent normalon.

Figure 13. Basic Model for the definition of the shapes of junctions.

Figure 14. Classification of levels of vertices.

Figure 15. Two Necker-reverse inflated modes.

Figure 16. Polyhedron with fissure.

Figure 17. Detection of dominant edge in an indefinite object: a) Original orientation, b) Rotation to see the effect of neutralisation.

Figure 18. Edges selected to determine the shape of a junction.

Figure 19. Inflation of vertices determined by two edges.

Figure 20. Inflation of vertices determined by two edges.

Figure 21. Limitations in the automatic detection of perceptual cues.

Figure 22. Limitations in the detection of symmetry.

Figure 23. Two alternative reconstructions showing three stages: Input sketch, intermediate interactive line drawing and batch-beautified line drawing.

Figure 24. (a) Line drawing with false edge parallelism; (b) most plausible perceptual solution.

Figure 25. Global minimum and tangled local minimum.

Figure 26. Two local minima in the neighbourhood of the global minimum in the design sub-space z_1 - z_4 .

Figure 27. Some normalons and quasi-normalons reconstructed by REFER.

Figure 28. Some polyhedral shapes reconstructed by REFER.

FIGURES

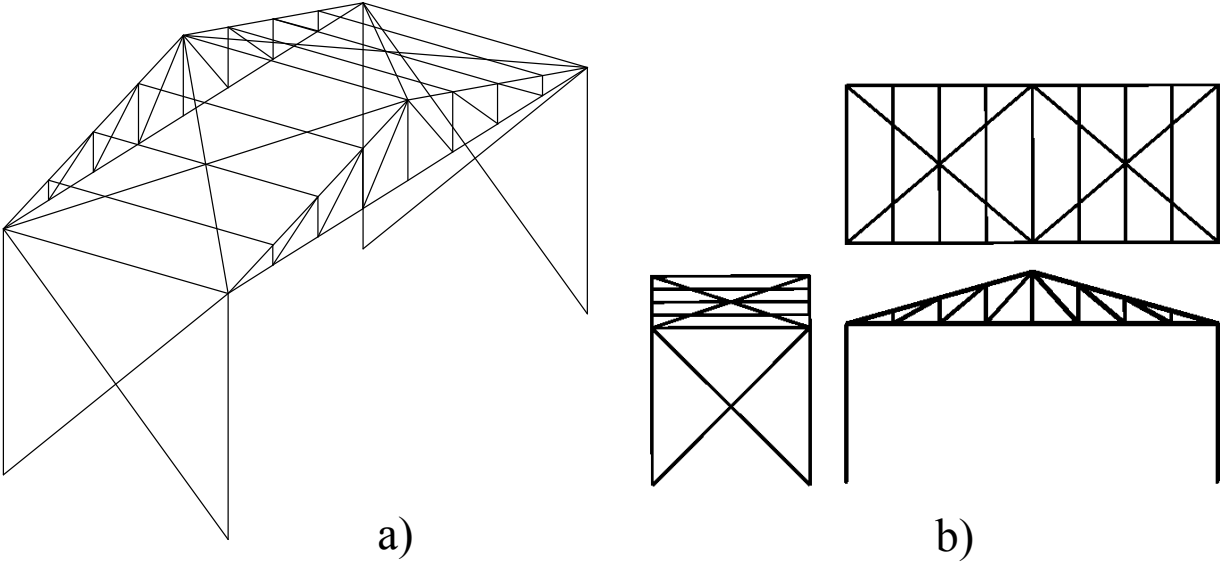


Figure 1

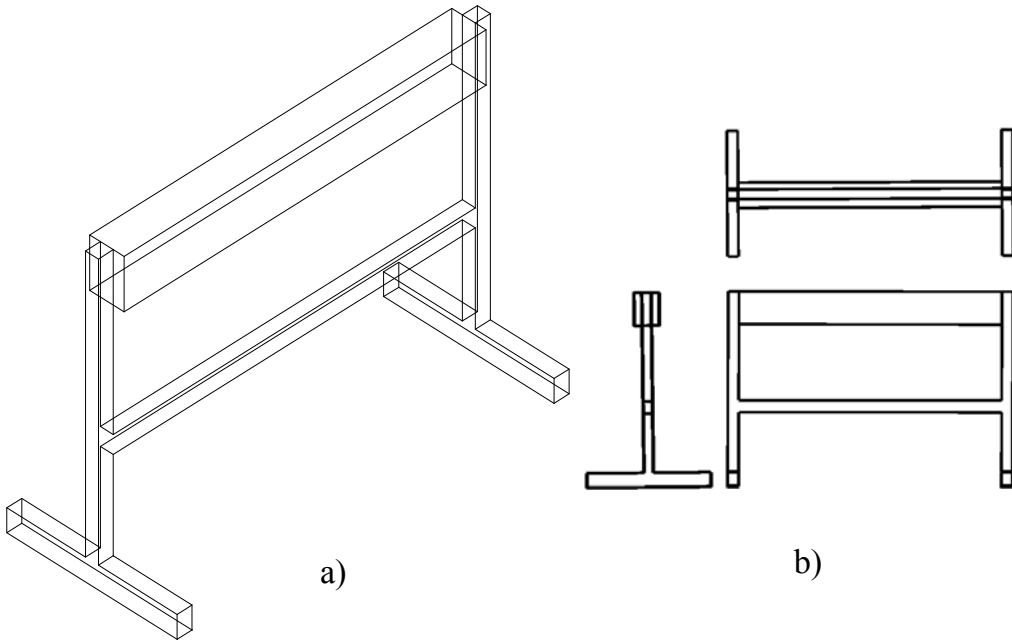


Figure 2

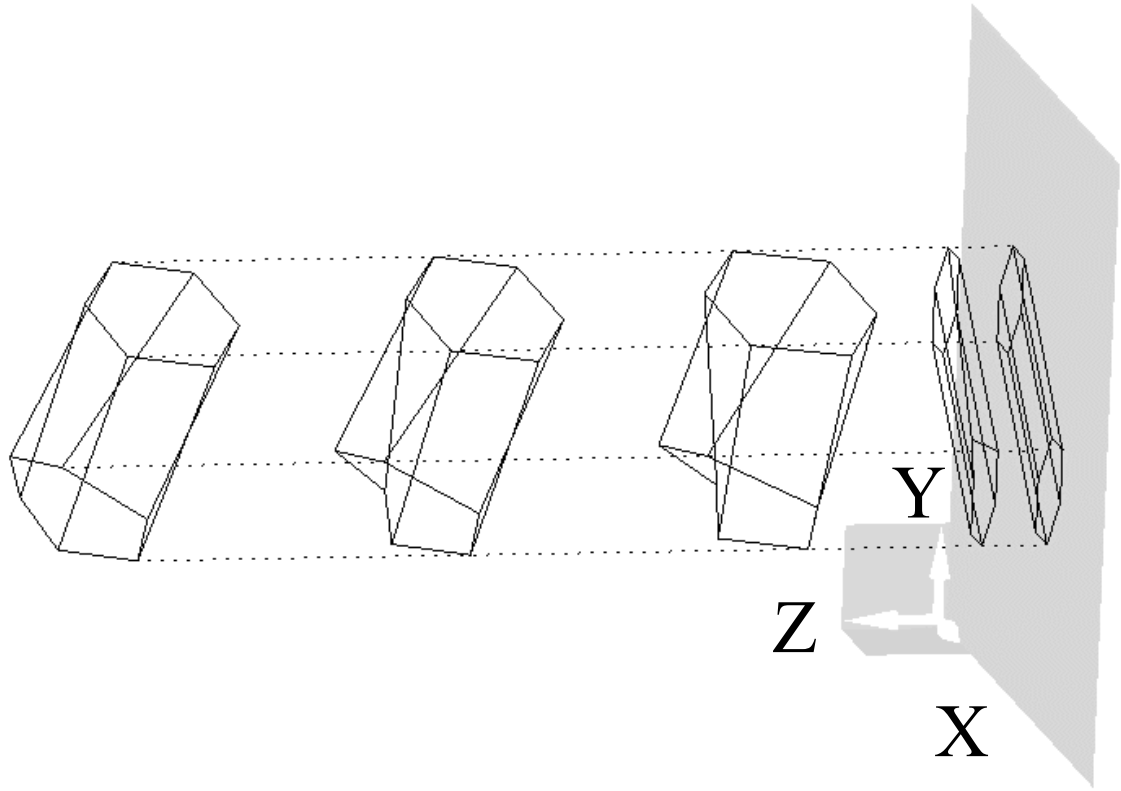


Figure 3

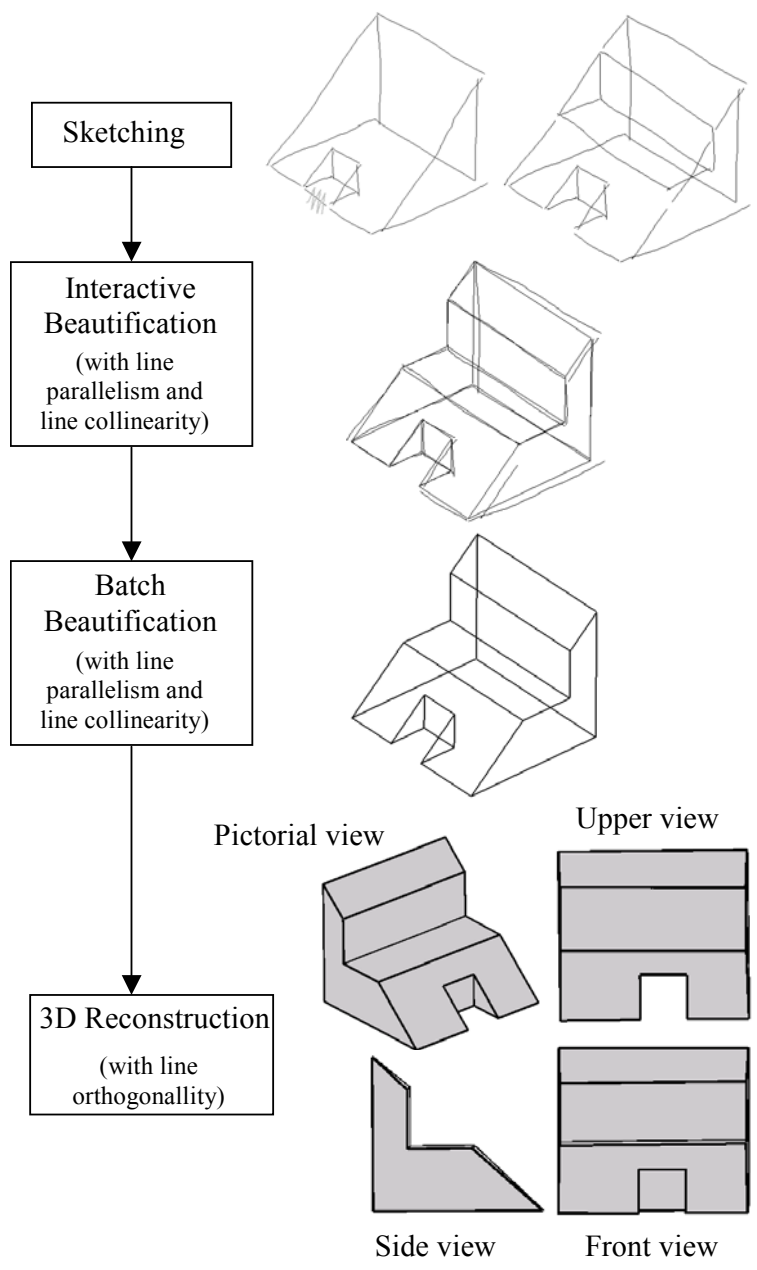


Figure 4

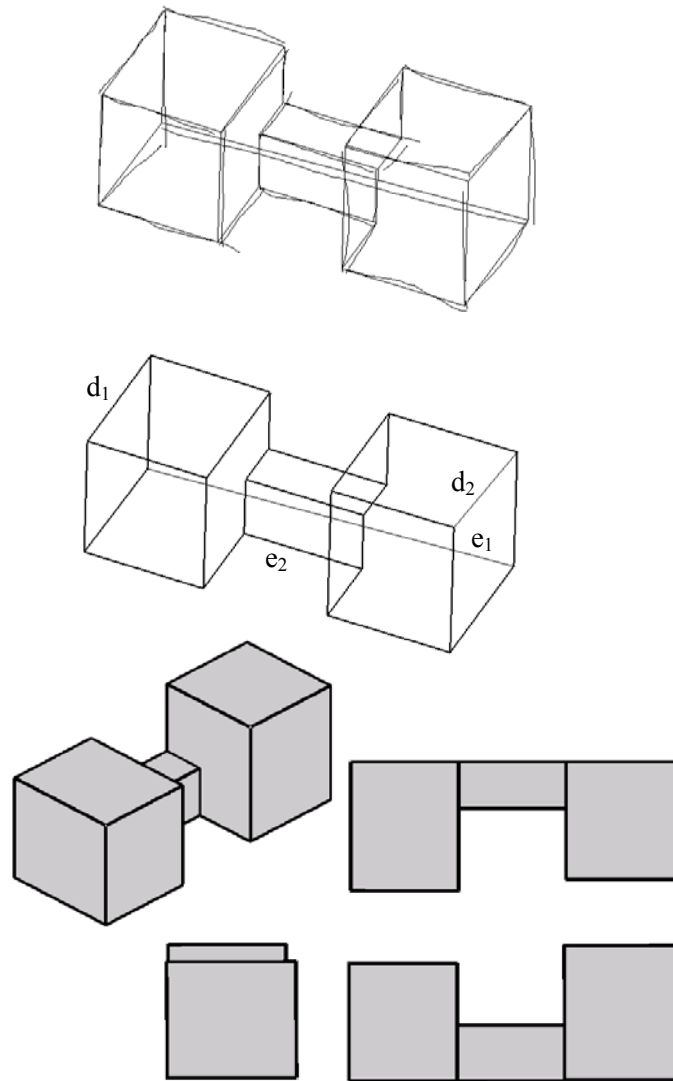
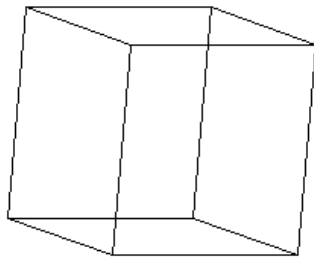
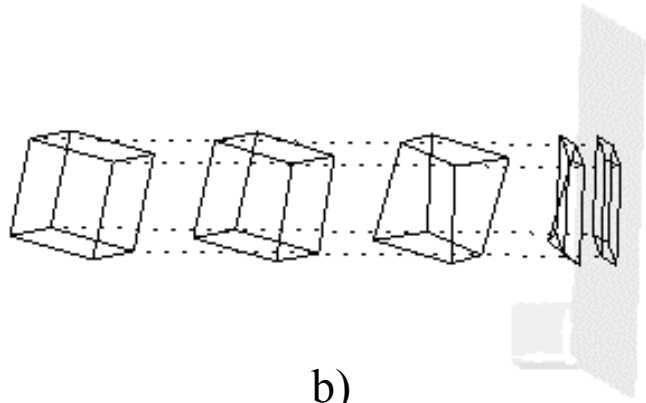


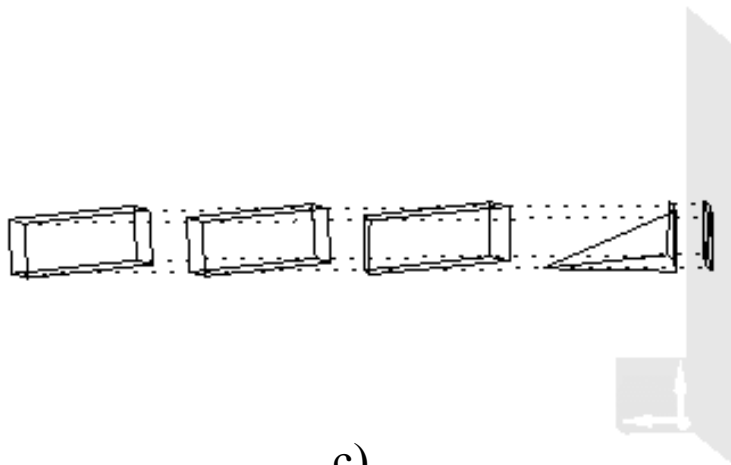
Figure 5



a)



b)



c)

Figure 6

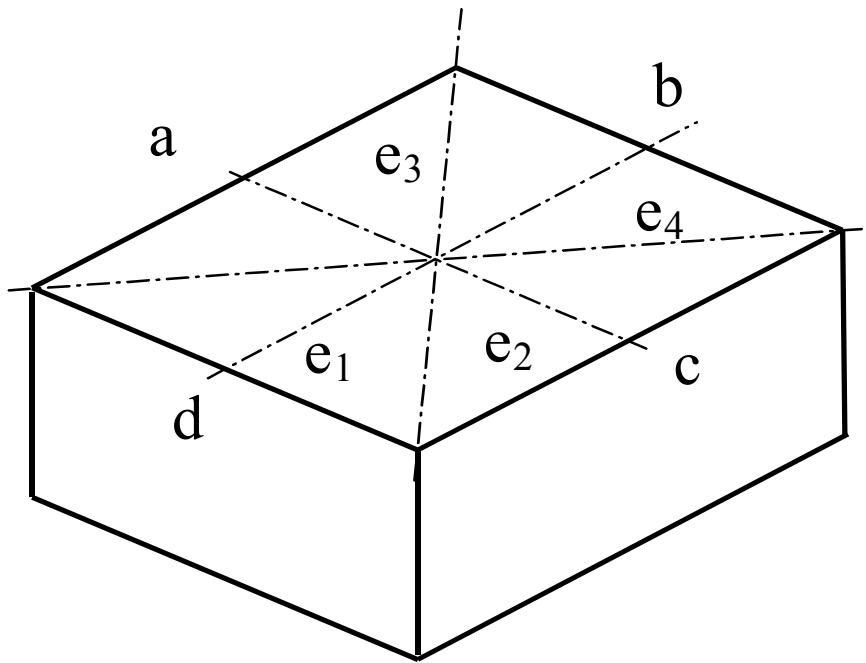


Figure 7

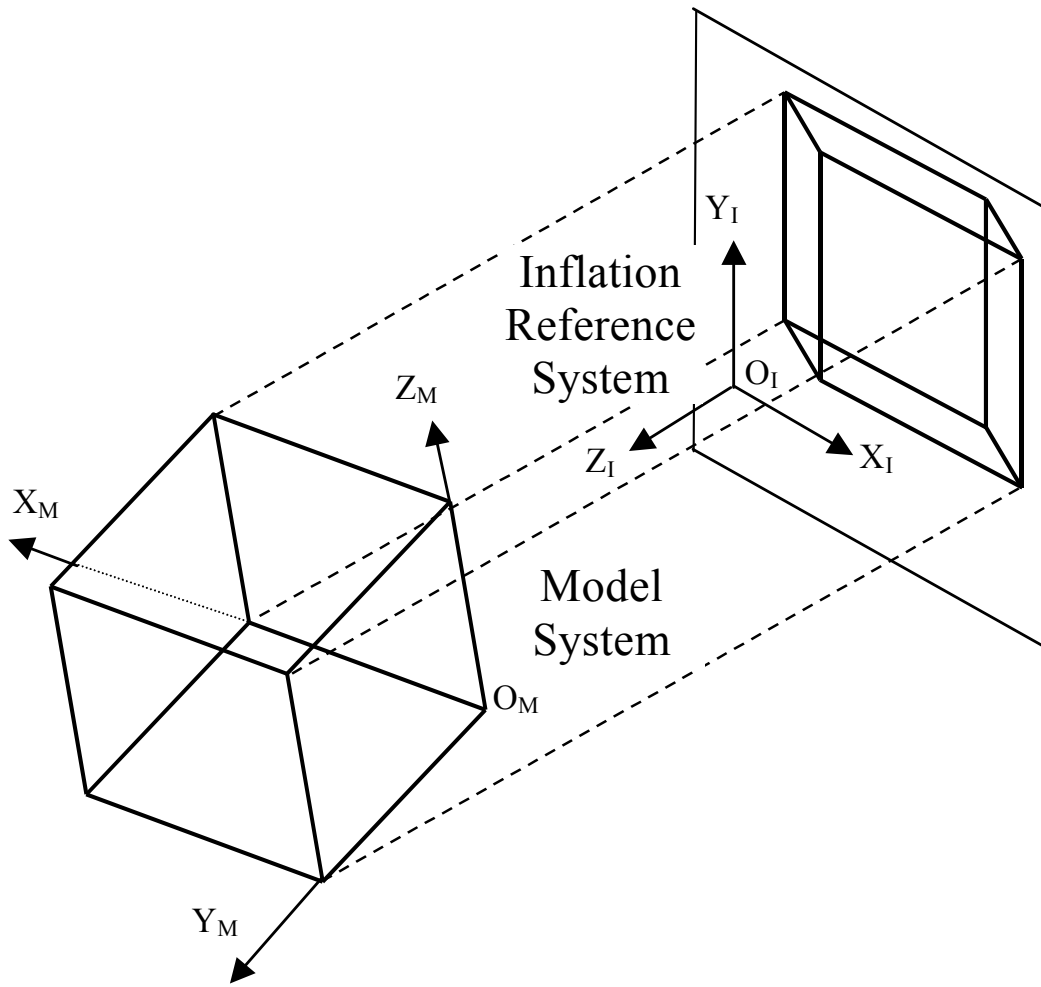


Figure 8

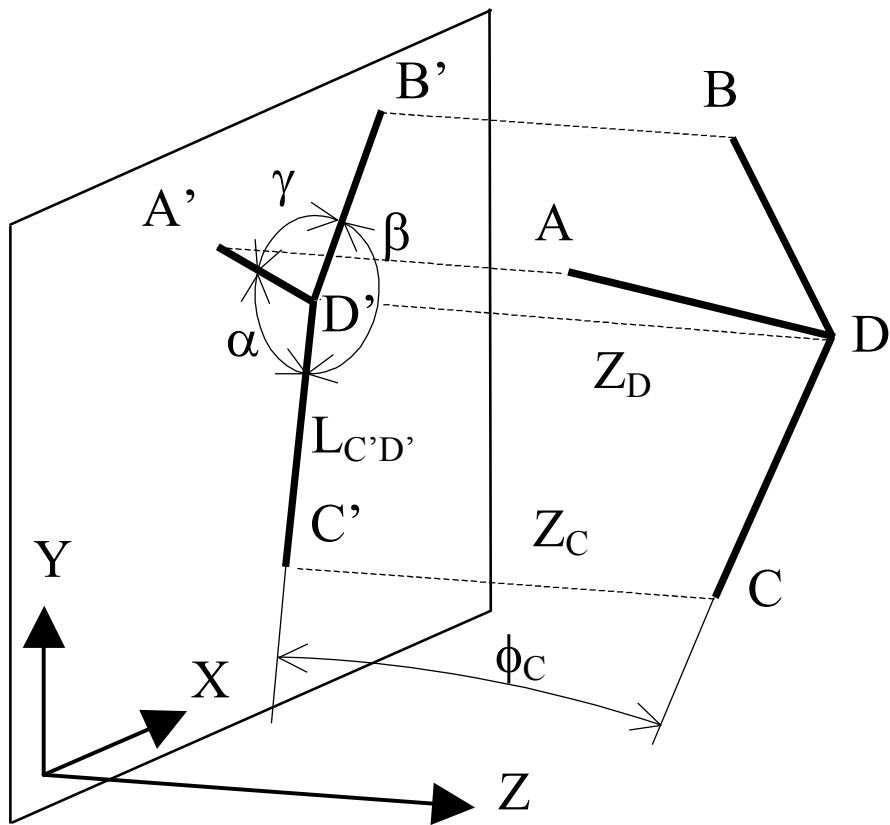


Figure 9

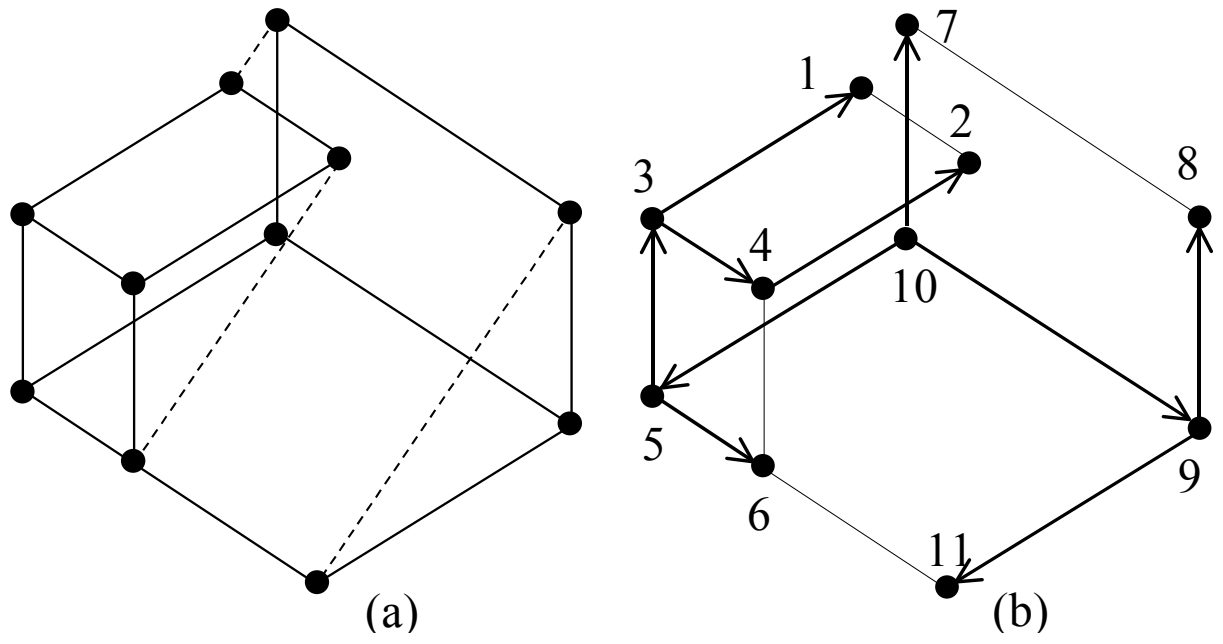


Figure 10

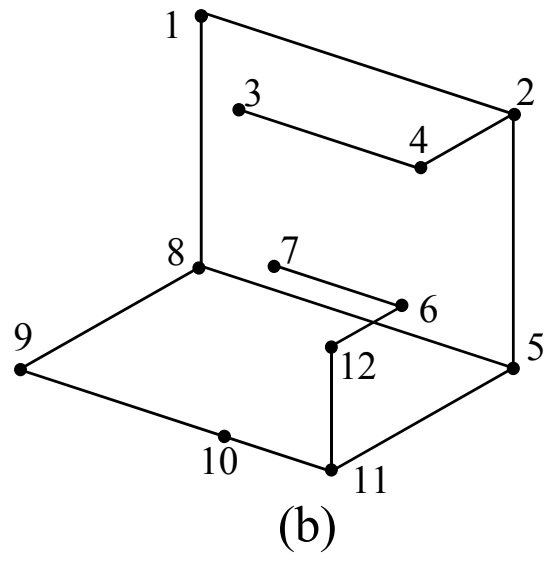
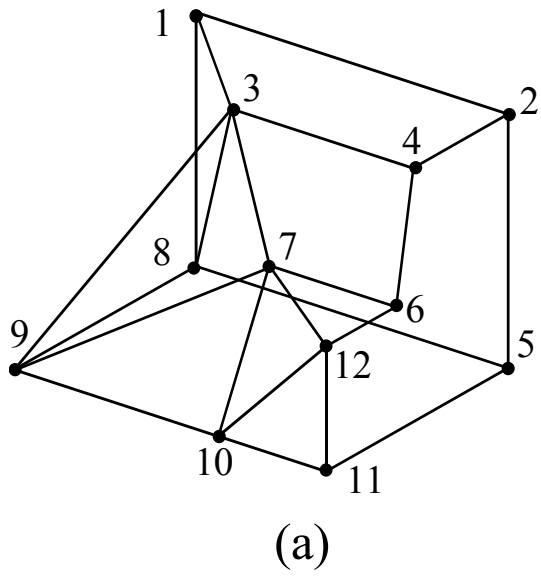


Figure 11

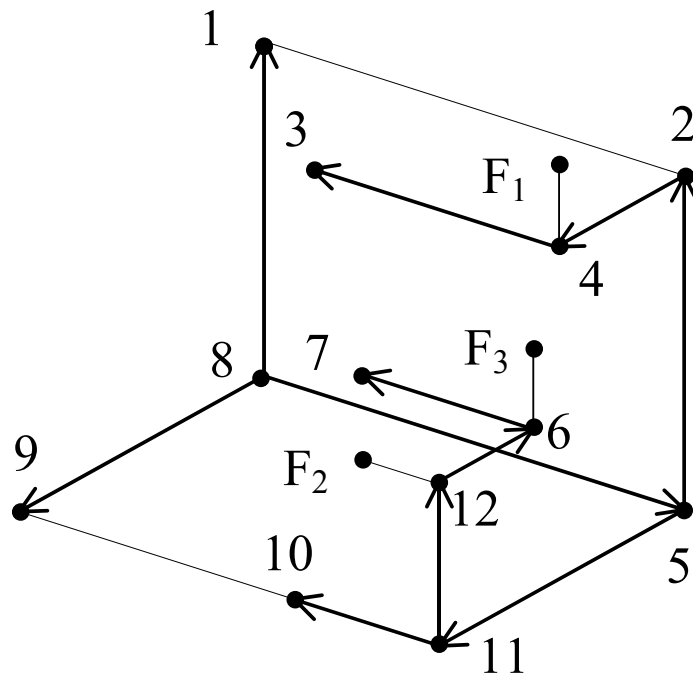


Figure 12

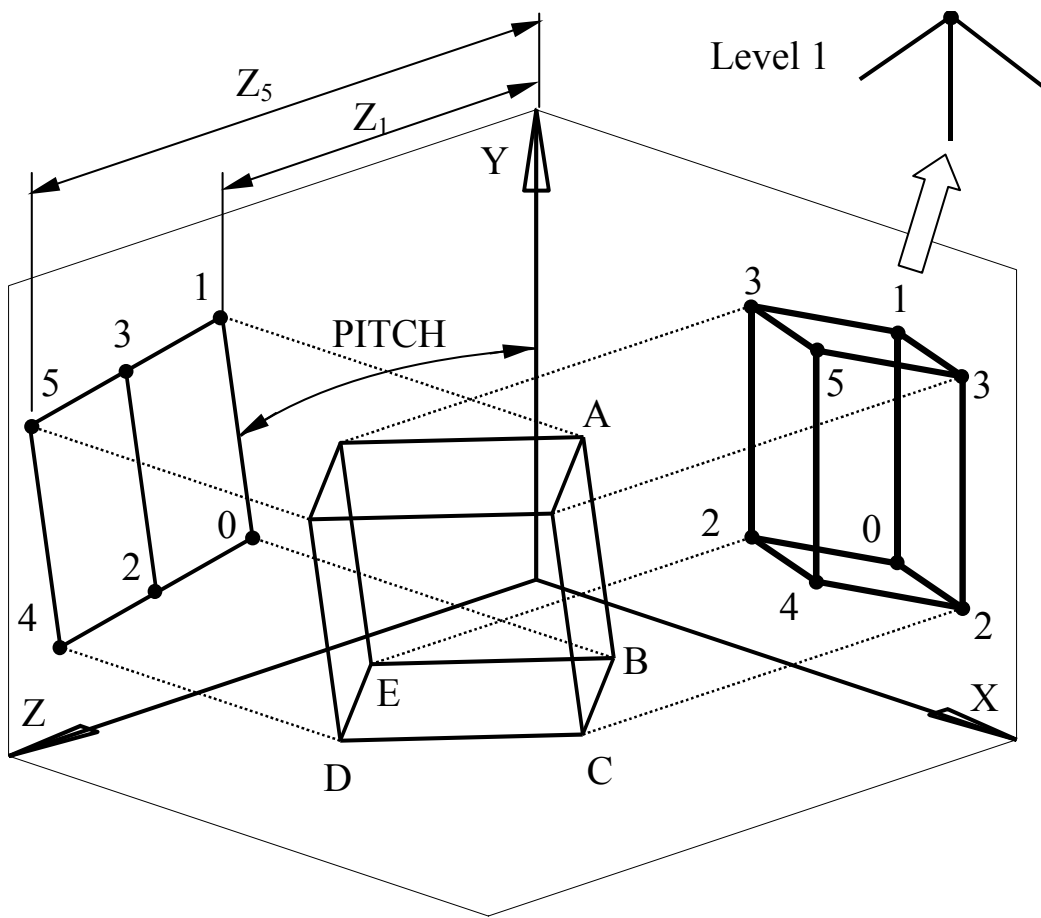


Figure 13




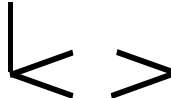

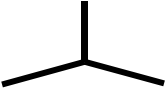
					
Level 5	Level 4	Level 3	Level 2	Level 1	Level 0
Y	W	Lateral inverted W	Lateral W	inverted W	inverted Y

Figure 14

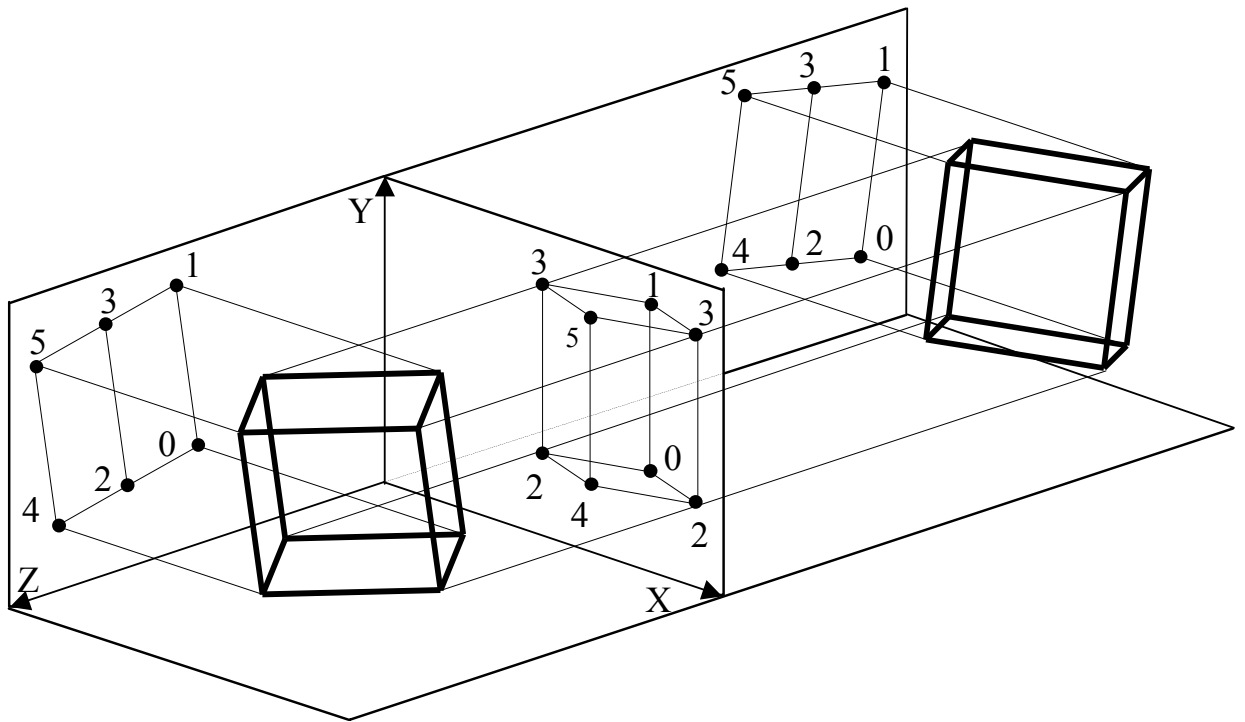


Figure 15

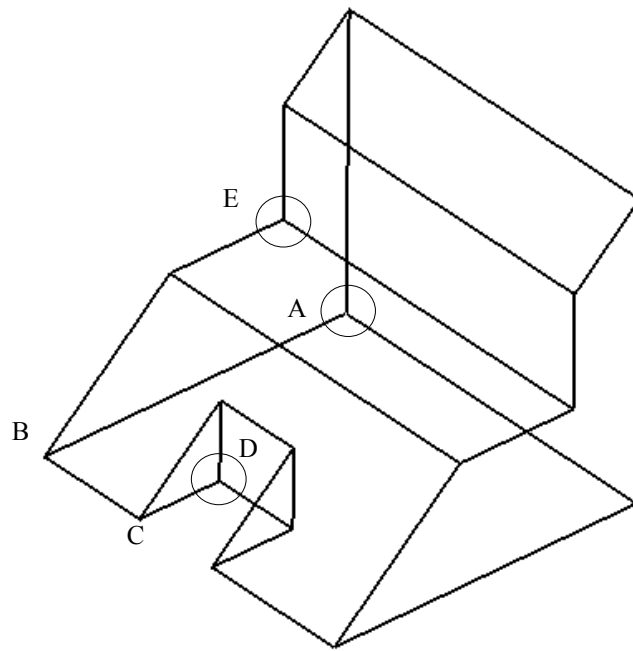


Figure 16

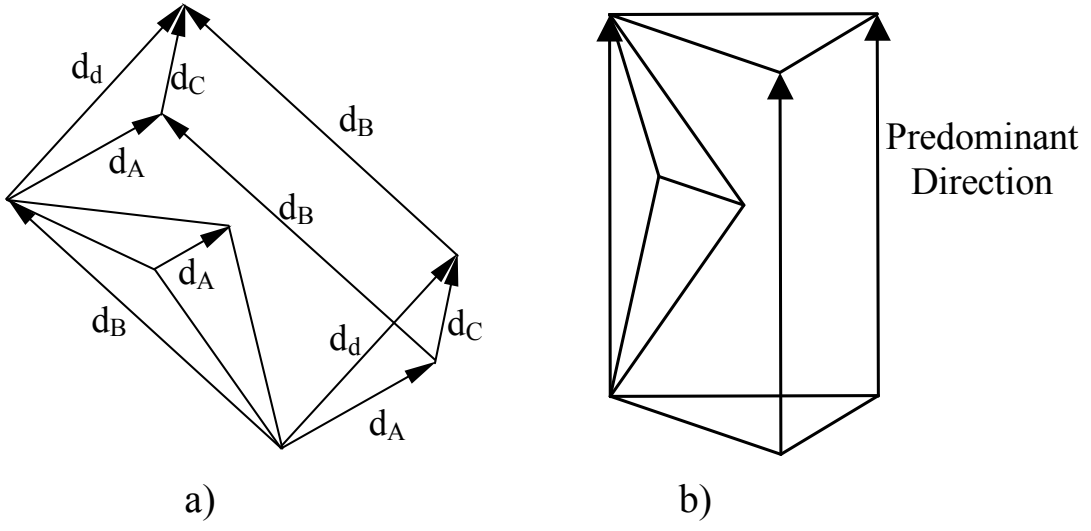


Figure 17

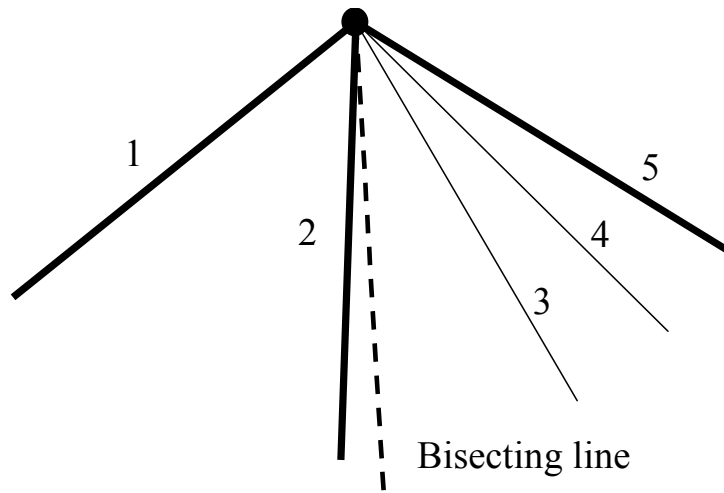


Figure 18

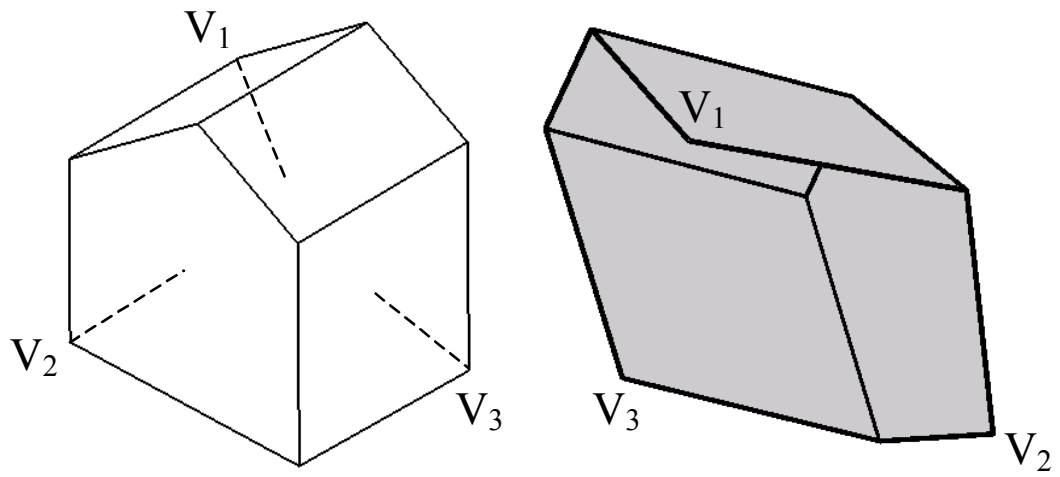


Figure 19

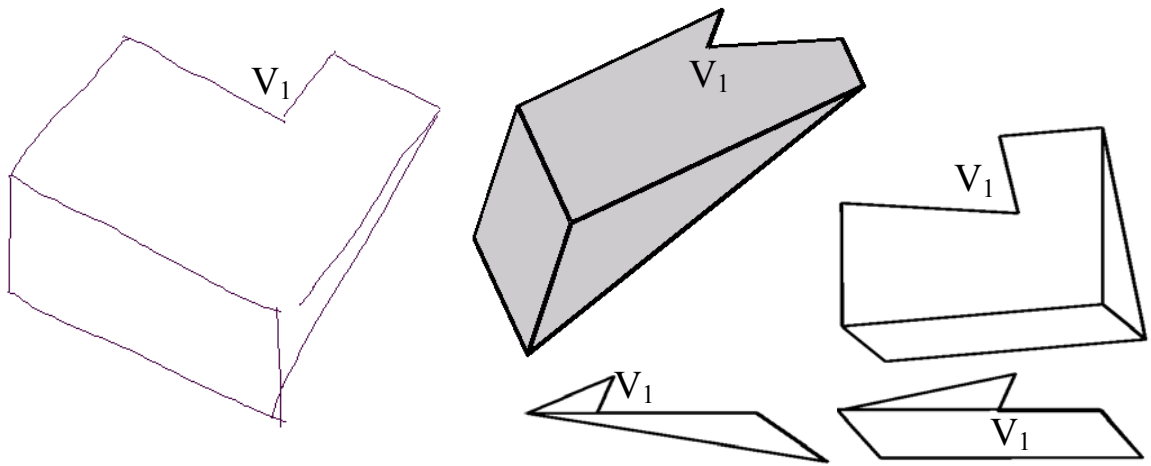


Figure 20

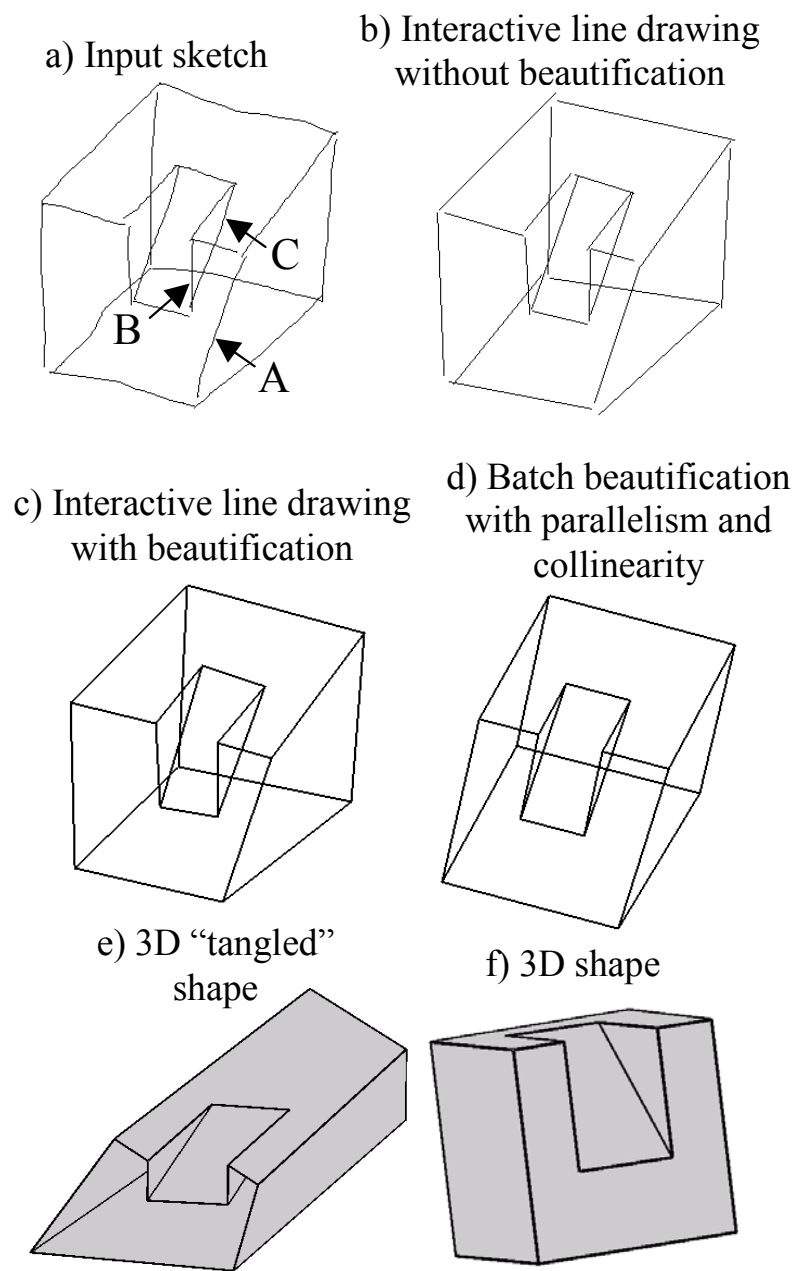


Figure 21

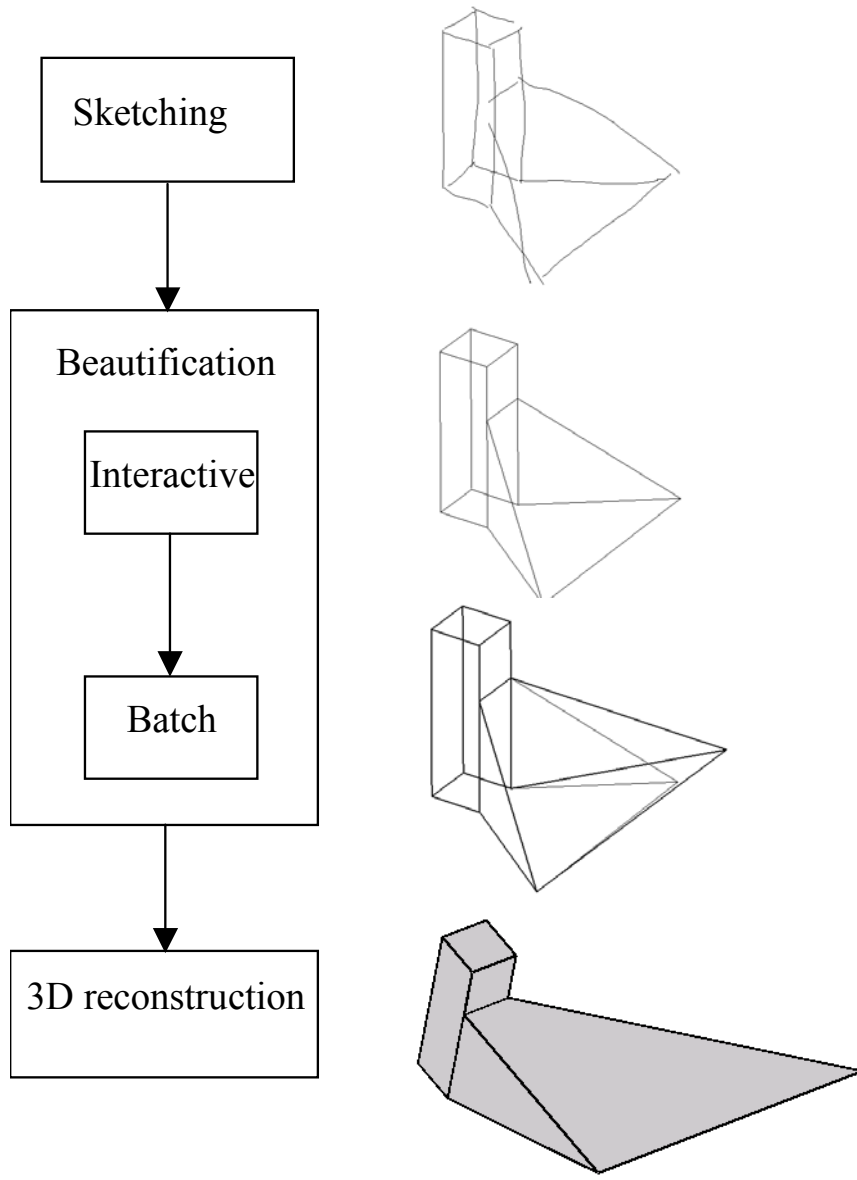


Figure 22

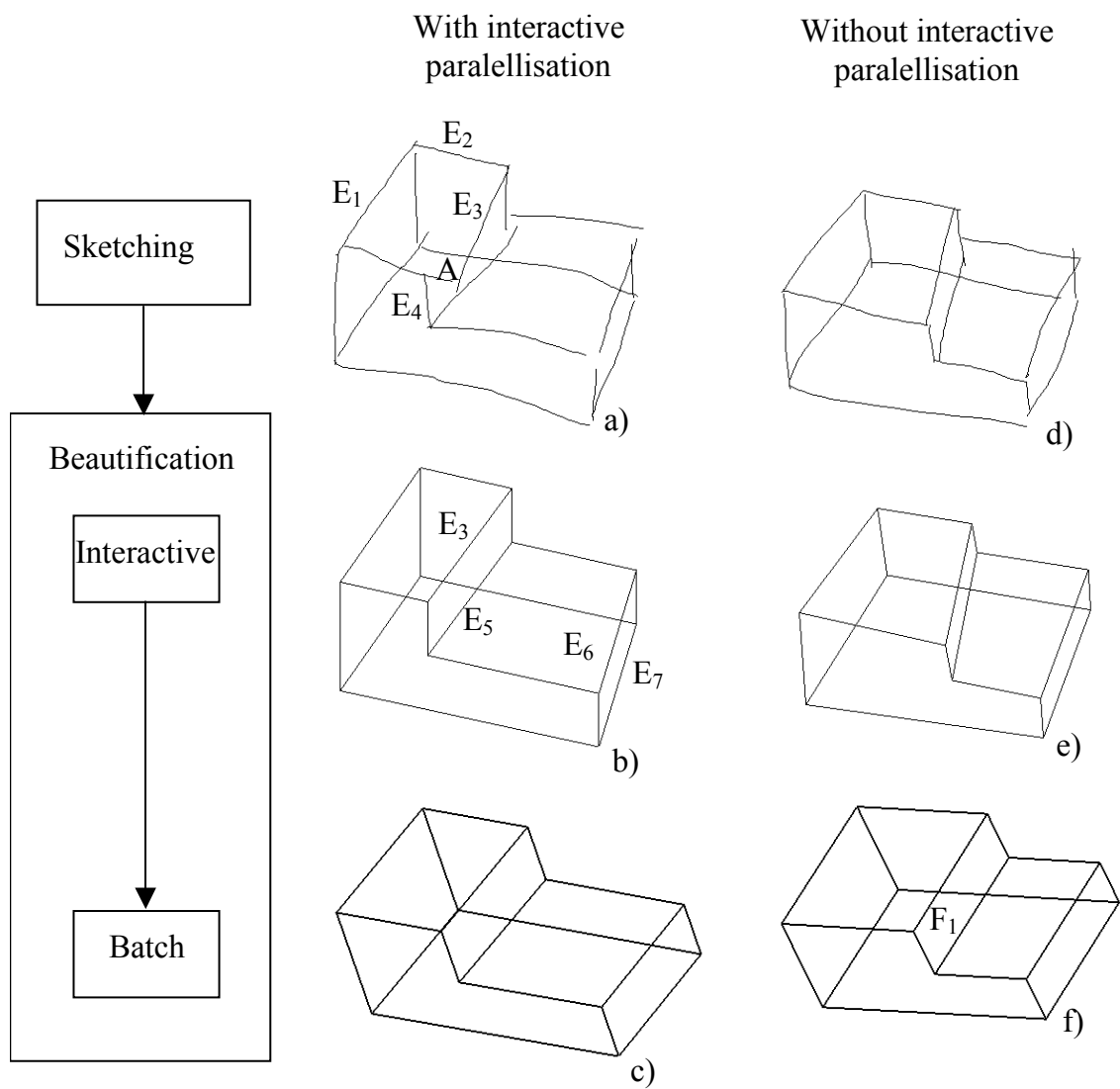
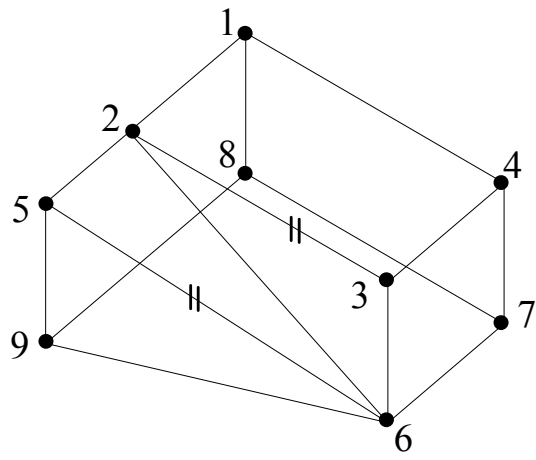
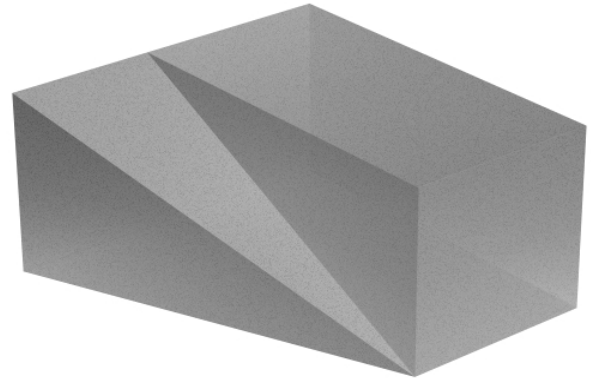


Figure 23

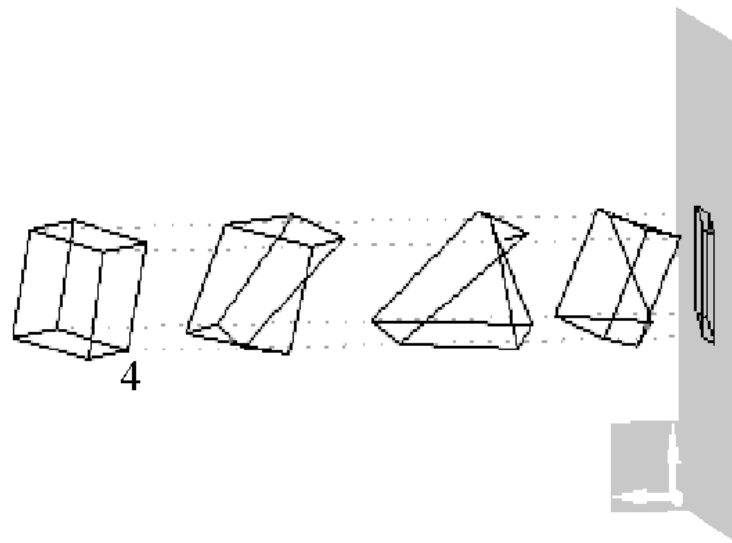


a)

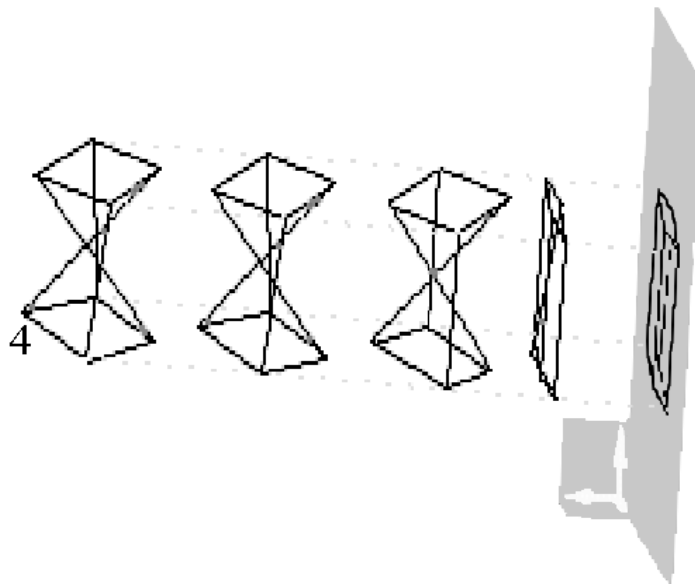


b)

Figure 24



Global minimum
 Initial step length (2,5-7) % and (15-80) %



Tangled local minimum
 Initial step length (9-14) %

Figure 25

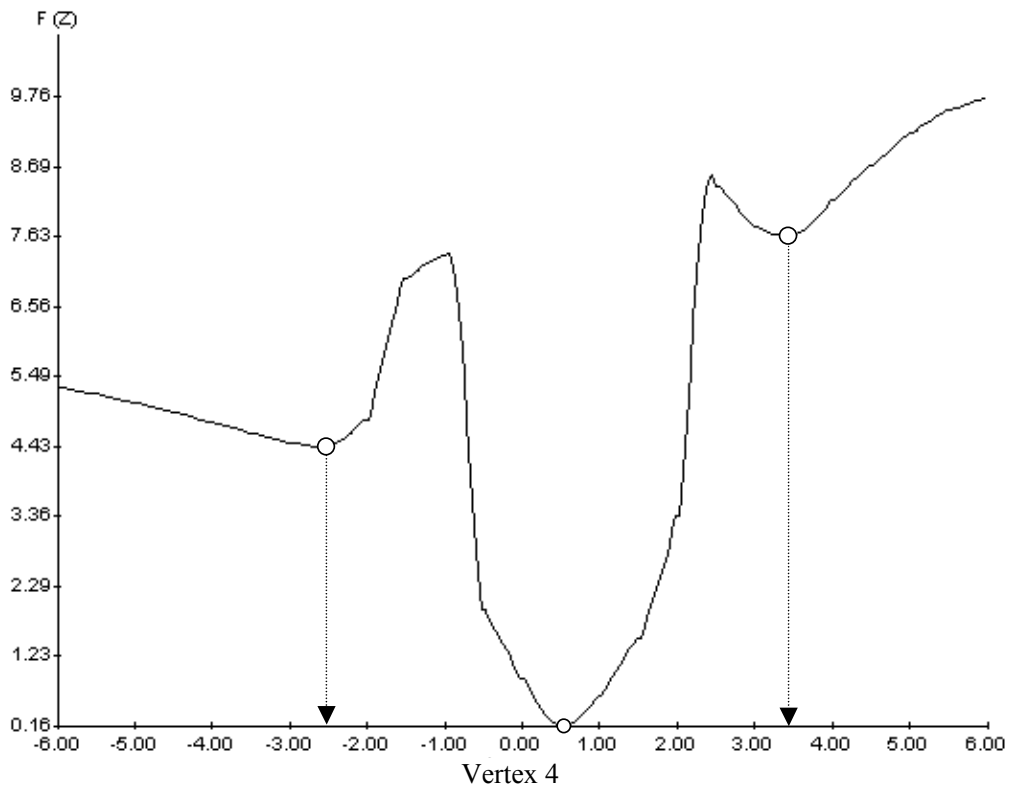


Figure 26

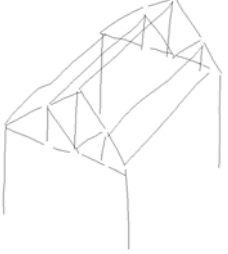
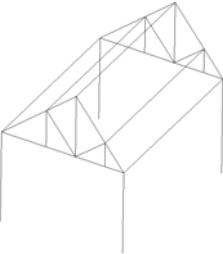
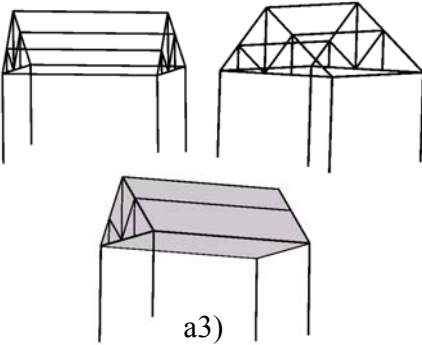
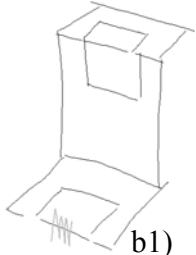
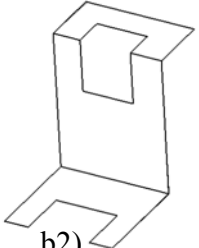
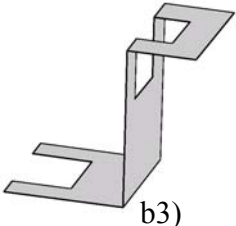
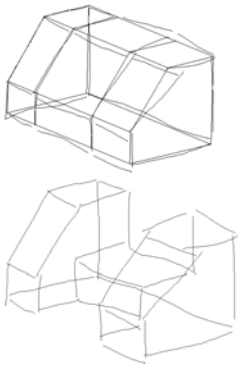
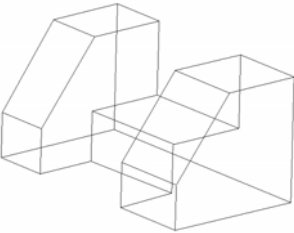
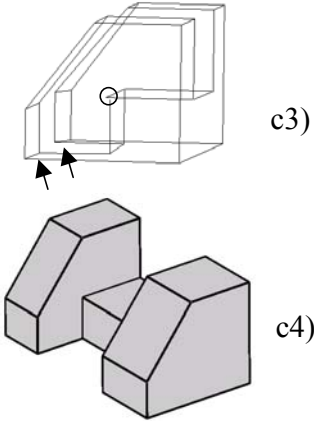
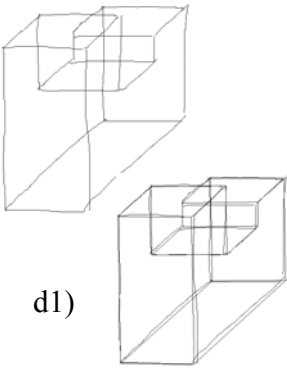
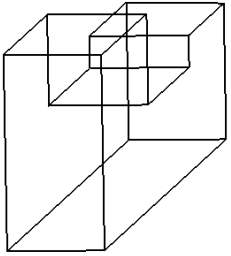
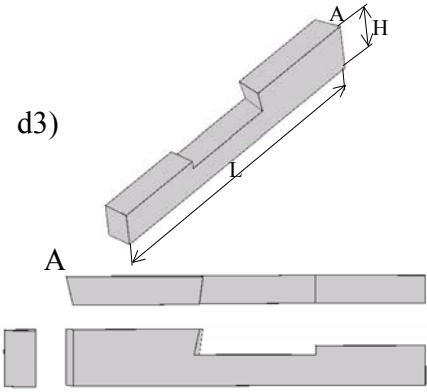
Input sketch	2D line drawing	3D shape
 <p>a1)</p>	 <p>a2)</p>	 <p>a3)</p>
 <p>b1)</p>	 <p>b2)</p>	 <p>b3)</p>
 <p>c1)</p>	 <p>c2)</p>	 <p>c3)</p> <p>c4)</p>
 <p>d1)</p>	 <p>d2)</p>	 <p>d3)</p>

Figure 27

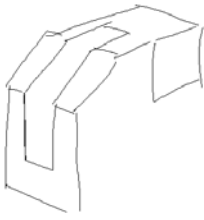
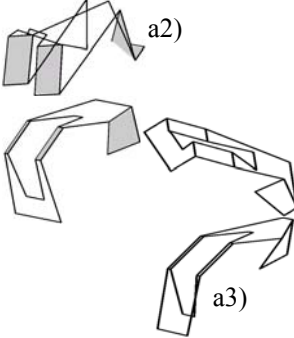
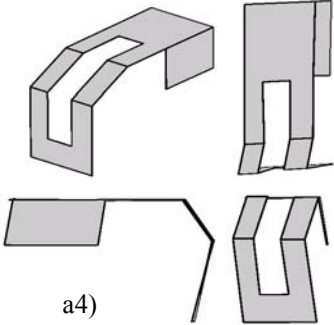
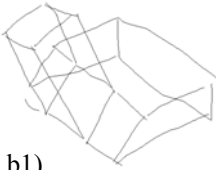
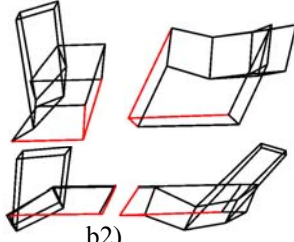
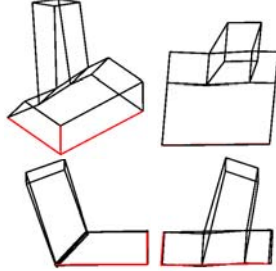

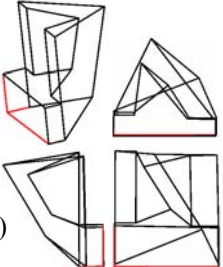
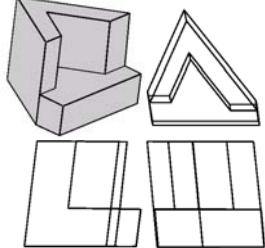

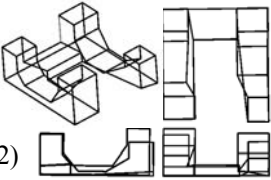
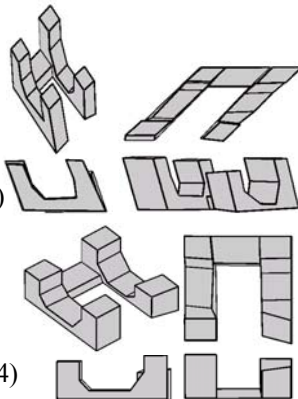
Input sketch	3D tentative model	3D shape
<p>a1)</p> 	<p>a2)</p>  <p>a3)</p>	 <p>a4)</p>
<p>b1)</p> 	 <p>b2)</p>	 <p>b3)</p>
<p>c1)</p> 	 <p>c2)</p>	 <p>c3)</p>
<p>d1)</p> 	 <p>d2)</p>	 <p>d3)</p> <p>d4)</p>

Figure 28