Realizability of a Sketch: An Algorithmic Implementation of the Cross-Section Criterion

Sofia Kyratzi[1], Philip Azariadis[2] and Nickolas S. Sapidis[3]

[1]University of the Aegean, skiratzi@aegean.gr
[2]University of the Aegean, azar@aegean.gr
[3]University of Western Macedonia, nsapidis@uowm.gr

ABSTRACT

Humans have the ability to easily recognize 3D shape from a given sketch. On the contrary, a computer program requires mathematical criteria to establish if a sketch is realizable, i.e., if it is the projection of a 3D model. This paper presents an algorithmic method that is based on Whiteley's cross-section criterion, in order to evaluate the realizability of a single wireframe sketch (i.e., includes both visible and hidden lines) that depicts the orthographic projection of a trihedral polyhedron in general position. The proposed approach is based on an algebraic representation of the cross-section criterion and is able to evaluate the realizability of inexact wireframe sketches.

Keywords: sketch realizability, 3D reconstruction, cross-section criterion.
DOI: 10.3722/cadaps.2011.665-679

1 INTRODUCTION

Although modern Computer Graphics applications have introduced 3D models in everyday activities, sketches remain the primary tool to communicate an idea in conceptual design. A computer-aided design program encounters sketches as a 2D configuration of lines, without any geometric depth information, in opposition to humans' ability that intuitively realize the 3D geometry from a given 2D sketch. In an effort to emulate this ability, "interpretation of sketches as 3D models" forms a significant research subject in Geometric Modeling, Artificial Intelligence, and Computer Vision.

In the last forty years numerous "realizability criteria" for sketches have been introduced (see [2][11][12][14-17][20][21] and references therein) in order to provide geometric, algebraic, or computational conditions that determine whether a sketch identifies with the projection of a valid polyhedron [11]. The first attempt was introduced by Huffman [4] and Clowes [1] in 1971 and is based on a "Line Labeling" scheme and a "Junction Catalog", which includes all possible cases of junctions existing in a sketch that depicts a trihedral solid. A detailed analysis of the Line Labeling method can be found in [13][18]. Many authors refined and extended this method to include sketches of tetrahedral solids [17], of curved objects [2], or sketches with shadows [19]. The main disadvantage of this method is that a consistent labeling of a sketch is only a necessary condition for the existence of a polyhedron corresponding to a sketch. Moreover, for a sketch that admits many different consistent labeling, there is no efficient method to identify those that correspond to a valid polyhedron [8][18].

Based on the Line Labeling method, Sugihara [15] constructs a linear system of equalities and inequalities, whose solution corresponds to a polyhedron. This method is a necessary and sufficient condition for the realizability of a wireframe sketch, but it is sensitive to slight perturbations of junctions' position in the sketch. In order to overcome this problem, Sugihara eliminates the involved inequalities of the linear system. The same author introduces, in [16], the "resolvable sequence" of a wireframe sketch; this is a specific order by which all elements of a sketch can be "lifted" in space so that finally a polyhedron is produced. According to this method, a sketch is realizable if there exists at least one "lift" for the given sketch.

The works [11][14] and [20] present geometric criteria based on necessary "line-concurrence conditions" for the realizability of a sketch. On the basis of the conditions in [20] and the geometric theory of Maxwell [10] for plane frameworks with planar graphs, Whiteley [21] proposes a geometric criterion that checks the realizability of a sketch with the help of a "cross-section" that is constructed from the sketch. This criterion is a necessary and sufficient condition for a sketch to be realizable. According to this criterion, a cross-section is constructed from a given sketch so that each region/line of the sketch is represented by a line/point on the cross-section. The sketch is realizable if and only if the cross-section is compatible with the sketch (see an analytic description of this criterion in Section 3). Ros and Thomas [12] revisit Whiteley's "cross-section criterion" and succeed in developing a drastically-simplified theory and criterion without using the complex geometric theories and physical analogies employed in [21].

The cross-section criterion can be applied in a "sketch-to-solid" reconstruction process [5][6][8] in order to robustly determine and verify the geometry of a given sketch. Thus, for determining the realizability of a given wireframe sketch, the authors in [7] define the "cross-section problem" (CSP) on the basis of an algebraic model of the cross-section criterion, and a bilinear system of equations (i.e., the "cross-section system"). Moreover [7] presents a mathematical solution of the CSP that is based on expressing the unknowns of the system in terms of three independent system parameters. The limitations of this method are that: (a) it does not employ all equations of the bilinear system to solve the CSP, (b) when the number of sketch regions increases, it becomes computationally ineffective, i.e., it requires complex formulas to achieve the aforementioned expressions, and (c) it is sensitive to perturbation errors in the junctions' coordinates of a given sketch, i.e., it is not able to handle inexact sketches.

To overcome these limitations this paper proposes a new algorithmic method to solve the CSP that is based on the incremental linearization of all bilinear equations included in the "cross-section system". The advantages of this method are that: (a) it provides an algebraic model that explicitly describes the geometric cross-section criterion, (b) it utilizes all equations of the "cross-section system", and thus offering a more robust evaluation of a sketch's realizability by considering all geometric information that the sketch conveys, and thus it is more complete and accurate, (c) it can efficiently process more complex sketches, because the incremental linearization approach results in the employment of simplified formulas to produce a solution of the CSP, and (d) it can handle inexact sketches, i.e., sketches with noisy vertices. The latter is achieved by introducing a tolerance parameter that controls the accepted level of accuracy of a given sketch.

The structure of this paper is as follows. Section 2 presents the assumptions and the terminology used throughout the paper. Section 3 discusses the cross-section criterion and reviews the existing "ruler and compass" method for the construction of a cross-section from a sketch. Section 4 outlines the algebraic model of the cross-section criterion, and Section 5 proposes a new method for recognizing realizable sketches on the basis of the proposed "Incremental Linearization" algorithm. Moreover, realizability of a sketch along with the performance of the proposed algorithm are analyzed with respect to small perturbations of sketch's junctions.

2 ASSUMPTIONS AND NOTATION

This research deals with a manifold solid (polyhedron) and its orthographic projection on a plane Φ (i.e., the plane $Z=0$). In particular, it focuses on *trihedral* solids (i.e., each vertex of it belongs to exactly three faces) having planar faces, where adjacent faces lie on distinct planes. The solid

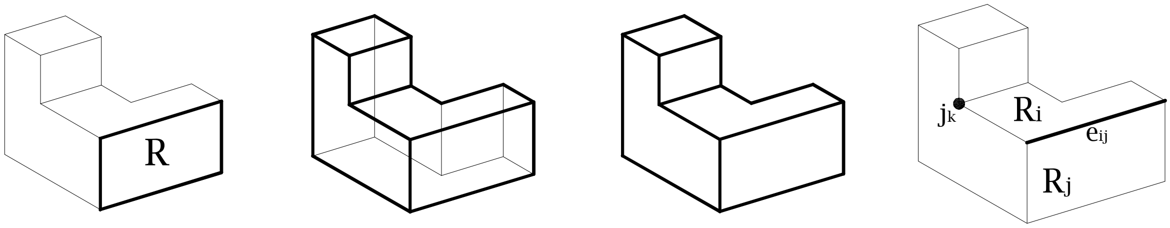


Fig.1: (a) A region of a sketch, (b) A wireframe sketch, and (c) its corresponding natural sketch, (d) Line e_{ij} belongs to regions R_i and R_j.

(polyhedron) is considered to be in "general position" with respect to the given projection plane Φ, i.e., no face or edge of the solid is perpendicular/parallel to Φ.

A sketch *S* is a set of straight lines, on a plane Φ, that intersect at junctions. Loops of lines and junctions form the regions of the sketch (Fig.1(a)). In order for a sketch to represent the projection of a polyhedron in "general position", adjacent lines and junctions do not coincide. Thus, an "one-to-one correspondence" is considered between the vertices (V), edges (E), and faces (F) of the solid and the junctions (J), lines (L), and regions (R) of the sketch [6][9][17]. In particular, this research focuses on *wireframe sketches* (Fig.1(b)); i.e., sketches that include both visible and hidden lines/junctions/regions, in opposition to natural sketches that depict only the visible part of a solid (Fig.1(c)).

We note that the common line of two adjacent regions R_i and R_j is denoted as e_{ij} (Fig.1(d)). A line e_{ij} with terminal junctions $v_p(x_p, y_p)$ and $v_q(x_q, y_q)$ is written as $e_{ij} : k_{ij}x + m_{ij}y + n_{ij} = 0$, with $k_{ij} = y_p - y_q$, $m_{ij} = x_q - x_p$, and $n_{ij} = x_p y_q - x_q y_p$. The topological properties of wireframe sketches are a direct result of properties of trihedral polyhedra. Thus, in every wireframe sketch: (1) each junction j_k is adjacent to three lines, (2) every line is adjacent to two regions, (3) the sketch is a connected graph (Fig.1(b)), and (4) two adjacent regions of the sketch share exactly one line or two-or-more collinear lines [9].

3 THE CROSS-SECTION CRITERION

Given a trihedral polyhedron P (see example in Fig. 2(a)), consider an arbitrary plane Φ which is "in general position" with respect to P, i.e., Φ is not parallel to any face of P and also Φ does not intersect P. Intersecting the planes of the polyhedron's faces with Φ produces an arrangement of lines called cross-section of the polyhedron. If each pair of cross-section lines Lf_i and Lf_j (corresponding to the two faces F_i and F_j) intersect at a point P_{ij} on (the extension of) the common edge e_{ij} of F_i and F_j, the cross-section is called compatible. The above concurrence conditions hold true even for the wireframe sketch produced by the projection of the polyhedron onto the plane Φ, because projection preserves collinearity of points and all incidence relations; see Fig. 2(b-c). Thus, given a wireframe sketch *S* (Fig. 2(b)) with *L* lines, *J* junctions and *R* regions, a *cross-section* of *S* is an arrangement of lines $\{Lf_k; k = 1,\ldots R\}$ that represents the regions $\{R_k; k = 1,\ldots,R\}$ of *S*. In complete analogy to the above discussion, if each pair of *cross-section* lines Lf_i and Lf_j intersect at a point P_{ij} on (the extension of) the common line e_{ij} of R_i and R_j, the cross-section is called *compatible with S* (Fig. 2(c)) [12]. In this paper, we address the problem of deriving a cross-section from a given wireframe sketch *S* and we develop an algorithmic method for assessing the realizability of *S*. The plane Φ is considered to be the projection plane Z=0 (i.e., sketch plane).

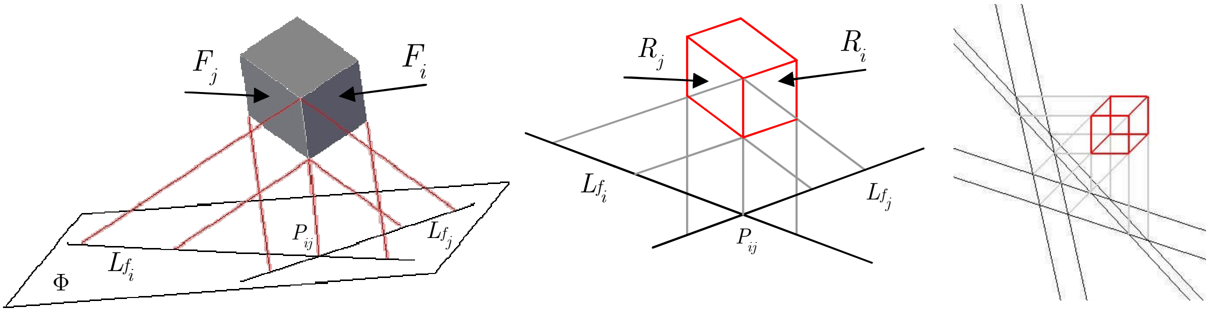


Fig. 2: (a) The cross-section of a polyhedron onto a plane Φ, (b) the cross-section of a wireframe sketch onto plane Φ, (c) compatibility check of a cross-section with the corresponding sketch.

Whiteley [21] was the first to introduce a cross-section theorem establishing the realizability of a wireframe sketch. The authors of [12] rewrote Whiteley's theorem as follows:

Theorem 1: *(Cross-Section Criterion)* A wireframe sketch is *realizable* if and only if it has a *compatible cross-section*, where the cross-section lines Lf_i and Lf_j of the adjacent regions R_i and R_j are not identical. □

According to Theorem 1, the compatibility of a cross-section forms the basis for the evaluation of a sketch in terms of realizability. Moreover, Theorem 1, establishes the existence of more than one cross-sections for a given sketch (Fig. 3(a)) [8].

The cross-section criterion implies that: (a) the cross-section lines of two adjacent regions must be different, and (b) each cross-section line must intersect with the lines of the corresponding region. These two *principal constraints* form the basis upon which the construction of a cross-section, either compatible or not, can be accomplished.

Based on this initial determination methodology, Ross & Thomas in [12] present a "ruler and compass" method for the construction of a cross-section from a sketch. This geometric method is based on an incremental construction of a cross-section from a wireframe sketch. A detailed description of this procedure can be found in [7]. According to the incremental procedure, given a wireframe sketch, all cross-section lines of the sketch can be uniquely determined, if the cross-section lines Lf_s and Lf_t of two initially chosen adjacent regions R_s and R_t are fixed (Fig. 3(b)). This is established due to the fact that (a) the cross-section criterion is applicable to both a polyhedron and its corresponding sketch, and (b) a wireframe sketch is a connected graph with each region of it adjacent to at least three lines/regions [12].

An algorithmic construction of a cross-section from a wireframe sketch that is based on the

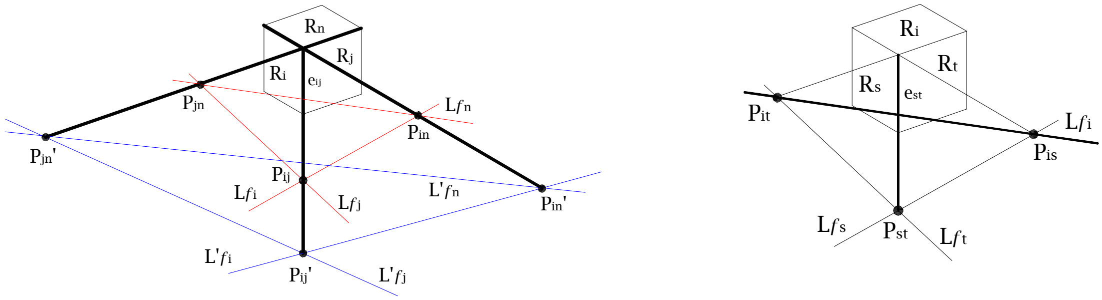


Fig. 3: (a) Two cross-sections of the same sketch, (b) Incremental construction of a cross-section: the cross-section lines Lf_s and Lf_t of two initially chosen regions are fixed and the rest of the cross-section lines $\{Lf_i\}$ are defined with respect to the initial fixed lines.

involved iterative process of the "ruler and compass" method [12] and employs an algebraic model of the cross-section criterion is presented in Section 5.

4 ALGEBRAIC MODEL OF THE CROSS-SECTION CRITERION

This section describes the construction of a cross-section from a given wireframe sketch along with an analysis of the corresponding constraints. This analysis is outlined in parallel with the proposed algebraic model of the cross-section criterion and in complete analogy to the concept of incremental construction. On the basis of Theorem 1, an algebraic representation of the cross-section criterion can be defined as follows [7].

Definition 1: *(Compatible Cross-section of a wireframe sketch)* Let *S* be a wireframe sketch with *R* regions, *L* lines and *J* junctions (see e.g., Fig. 3(a)). A compatible with *S* cross-section is a set of lines $\{Lf_i\}$ such that:

(A) Each cross-section line Lf_i that corresponds to region R_i, is written in the form: $b_i x + a_i y + 1 = 0$, $i = 1,\ldots,R$.

(B) The cross-section lines Lf_i and Lf_j of two adjacent regions R_i and R_j are not identical.

(C) For each R_i, its adjacent lines e_{ij} intersect the cross-section line Lf_i.

(D) Each line e_{ij} of *S* that is adjacent to regions R_i and R_j, and the corresponding to these regions cross-section lines Lf_i and Lf_j intersect at a point P_{ij}. □

Concerning property (A), we consider cross-section lines that do not pass through the origin in order to minimize the set of unknown parameters required for the definition of a cross-section (see Section 4.1 for details). In Section 5.1.1 we prove that this consideration does not pose any limitation to the proposed method. Properties (B) and (C) ensure that the generating cross-section lines will follow the principal construction rules, while property (D) is related to the compatibility of the cross-section with a sketch.

The following Theorems 2 and 3 are introduced in order for the lines $\{Lf_i\}$ of a cross-section to be in accordance with properties (A), (B), and (C) of Definition 1. Theorem 2 along with its proof can be found as "Theorem 3" in [7].

Theorem 2: A set of lines $\{Lf_i\}$ satisfies the properties (B), (C) of Definition 1 *if and only if* the following hold true:

(C1) $a_i b_j \neq a_j b_i$,

(C2) $k_{ij} a_i - m_{ij} b_i \neq 0$,

for each pair of variables (a_i, b_i), with $i = 1,\ldots,R$, where: $j \in \{1,\ldots,R\}$ corresponds to a region R_j that is adjacent to R_i. □

The next theorem includes a new added (C3) constraint, which ensures that none of the cross-section lines $\{Lf_i\}$ passes through the origin, according to property (A).

Theorem 3: Given two cross-section lines Lf_i, Lf_j that correspond to two adjacent regions R_i, R_j and satisfy properties (A)-(C) of Definition 1, distinct points P_{in} and P_{jn} define a line Lf_n that does not pass through the origin *if and only if*

(C3) $(a_i n_{in} - m_{in})(k_{jn} - b_j n_{jn}) - (a_j n_{jn} - m_{jn})(k_{in} - b_i n_{in}) \neq 0$

where P_{in} (resp. P_{jn}) is the intersection point of cross-section line Lf_i (resp. Lf_j) with sketch line e_{in} (resp. e_{jn}), and $n \in \{1,\ldots,R\}$ corresponds to a region R_n that is adjacent to both R_i and R_j.

Proof: Intersection point P_{in} (Fig. 3(a)), is derived through the following system:

$$\left.\begin{array}{l} b_i x + a_i y + 1 = 0 \\ k_{in} x + m_{in} y + n_{in} = 0 \end{array}\right\} \Rightarrow P_{in} = \left(\frac{a_i n_{in} - m_{in}}{D_{in}}, \frac{k_{in} - b_i n_{in}}{D_{in}}\right), \tag{4.1}$$

where $D_{in} = m_{in} b_i - k_{in} a_i \neq 0$, according to constraint (C2) of Theorem 2. In complete analogy to P_{in}, point P_{jn} is also defined as:

$$P_{jn} = \left(\frac{a_j n_{jn} - m_{jn}}{D_{jn}}, \frac{k_{jn} - b_j n_{jn}}{D_{jn}}\right) \tag{4.2}$$

with $D_{jn} \neq 0$.

($\Rightarrow$) The line equation of Lf_n can be written in terms of points P_{in} and P_{jn}:

$$(y_{in} - y_{jn})x + (x_{jn} - x_{in})y + x_{in} y_{jn} - x_{jn} y_{in} = 0 \tag{4.3}$$

According to Eqns. (4.1), (4.2) & (4.3), line Lf_n does not pass through the origin if

$$\begin{aligned} x_{in} y_{jn} - x_{jn} y_{in} \neq 0 \Leftrightarrow \frac{(a_i n_{in} - m_{in})(k_{jn} - b_j n_{jn}) - (a_j n_{jn} - m_{jn})(k_{in} - b_i n_{in})}{D_{in} D_{jn}} \neq 0 \Leftrightarrow \\ \Leftrightarrow (a_i n_{in} - m_{in})(k_{jn} - b_j n_{jn}) - (a_j n_{jn} - m_{jn})(k_{in} - b_i n_{in}) \neq 0 \end{aligned} \tag{4.4}$$

($\Leftarrow$) We first prove that (C3) results in two distinct points P_{in} and P_{jn} that define Lf_n. According to Eqn. (4.4), if $\{x_{in} y_{jn} = 0 \text{ or } x_{jn} y_{in} = 0\} \Rightarrow P_{in} \neq P_{jn}$. Otherwise $\{x_{in} y_{jn} \neq 0 \text{ and } x_{jn} y_{in} \neq 0\}$, and according to Eqns. (4.1), (4.2) & (4.4)

$$\frac{(k_{in} - b_i n_{in})}{(a_i n_{in} - m_{in})} \neq \frac{(k_{jn} - b_j n_{jn})}{(a_j n_{jn} - m_{jn})} \Rightarrow \frac{\frac{(k_{in} - b_i n_{in})}{D_{in}}}{\frac{(a_i n_{in} - m_{in})}{D_{in}}} \neq \frac{\frac{(k_{jn} - b_j n_{jn})}{D_{jn}}}{\frac{(a_j n_{jn} - m_{jn})}{D_{jn}}} \Rightarrow \frac{y_{in}}{x_{in}} \neq \frac{y_{jn}}{x_{jn}} \Rightarrow P_{in} \neq P_{jn}.$$

Thus, points P_{in} and P_{jn} are distinct and define the line Lf_n according to Eqn. (4.3). According to Eqn. (4.4), the constant factor of line (4.3) is non-zero. □

In the following sections, the term "cross-section line" refers to a line Lf_i, with $i = 1,\ldots,R$, that satisfies the constraints of Theorems 2 and 3. Theorems 2 and 3 are employed in the construction process of a cross-section from a given wireframe sketch. It is noted that none of them include property (D) as a constraint, because it is exactly the compatibility of a generated cross-section with a sketch that evaluates the realizability of the latter. This property will be encompassed in the formulation of the corresponding cross-section system, which is presented below.

4.1 The Cross-Section System

For any wireframe sketch *S* with *R* regions, *L* lines and *J* junctions (Fig. 3(a)), the cross-section lines Lf_i and Lf_j corresponding to the adjacent regions R_i and R_j, intersect at the point

$$P_{ij} = \left(\frac{a_j - a_i}{a_i b_j - a_j b_i}, \frac{b_i - b_j}{a_i b_j - a_j b_i}\right) \tag{4.5}$$

In order for the cross-section to be compatible, the point P_{ij} must be on the (extension of) line e_{ij}, where

$$e_{ij} : k_{ij} x + m_{ij} y + n_{ij} = 0 \tag{4.6}$$

Thus, Eqn. (4.5) is substituted in Eqn. (4.6) and produces:

$$(a_j - a_i)k_{ij} + (b_i - b_j)m_{ij} + n_{ij}(a_i b_j - a_j b_i) = 0 \tag{4.7}$$

Relation (4.7) is the equation of the *"Cross-Section system"* that corresponds to the line e_{ij}. For each line of the wireframe sketch *S*, a similar equation is obtained by following the same process. The coefficients (a_i, b_i), with $i = 1,\ldots,R$ are the unknown variables of the system:

$$\begin{array}{ll}
e_{st} \leftrightarrow eq_{st} : & (a_s - a_t)k_{st} + (b_t - b_s)m_{st} + n_{st}(a_t b_s - a_s b_t) = 0 \\
 & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
e_{ij} \leftrightarrow eq_{ij} : & (a_j - a_i)k_{ij} + (b_i - b_j)m_{ij} + n_{ij}(a_i b_j - a_j b_i) = 0 \\
e_{pq} \leftrightarrow eq_{pq} : & (a_q - a_p)k_{pq} + (b_p - b_q)m_{pq} + n_{pq}(a_p b_q - a_q b_p) = 0 \\
e_{pn} \leftrightarrow eq_{pn} : & (a_p - a_n)k_{pn} + (b_n - b_p)m_{pn} + n_{pn}(a_n b_p - a_p b_n) = 0 \\
e_{qn} \leftrightarrow eq_{qn} : & (a_q - a_n)k_{qn} + (b_n - b_q)m_{qn} + n_{qn}(a_n b_q - a_q b_n) = 0 \\
 & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
e_{kp} \leftrightarrow eq_{kp} : & (a_k - a_p)k_{kp} + (b_p - b_k)m_{kp} + n_{kp}(a_p b_k - a_k b_p) = 0
\end{array} \tag{4.8}$$

The number of the unknowns is $2R$ and the total number of equations is L. Each equation in (4.8) includes four unknowns and each pair of unknowns (a_i, b_i) (with $i = 1,\ldots,R$) appears in $\deg(R_i) \geq 3$ equations, where $\deg(R_i)$ equals to the number of lines adjacent to region R_i. A detailed analysis of the system can be found in [7].

5 SOLUTION OF THE CROSS-SECTION PROBLEM

System (4.8) combined with the constraints of Theorems 2 and 3 defines the "*Cross-Section Problem*" (CSP), whose solution determines the realizability of a sketch. The basic assumptions concerning the desirable properties of the two initial cross-section lines of Theorem 3 are taken into account in the proposed method for solving the CSP (see "IDVA" below).

This section presents a new method that first constructs and then evaluates the compatibility of a cross-section with a given sketch. In [7], CSP is solved by utilizing a partial set of $2L/3 + 1$ equations of the cross-section system in order to express $2R - 3$ unknowns with respect to three independent parameters α_s, b_s, b_t. By requesting that all unknowns comply with the algebraic-model constrains, valid values for α_s, b_s, b_t can be determined. The unused equations are employed to evaluate the compatibility of the produced cross-section with the sketch.

In this paper, the CSP is solved on the basis of an incremental linearization of system (4.8). The proposed "*Incremental Linearization Algorithm*" (ILA) proceeds iteratively to calculate all unknown pairs (a_i, b_i), $i = 1,\ldots,R$. When the unknowns of the system are incrementally determined, equations of system (4.8) become gradually linear, enabling a new pair (a_n, b_n) to be calculated. Suppose that in the $r-th$ iteration the values of both pairs (a_p, b_p) and (a_q, b_q) have been calculated. Then, according to system (4.8), the equations eq_{pn} and eq_{qn} become linear with respect to (a_n, b_n) producing a linear sub-system, which is solved in terms of the latter.

ILA requires an "initial values" determination, in order to proceed, but exploits all equations of system (4.8) in order to produce a feasible solution. In complete analogy to the geometric construction of a cross-section, the initial values are selected to be three unknowns (α_s, b_s) and b_t that correspond to the cross-section lines Lf_s and Lf_t of two adjacent regions R_s and R_t (note that eq_{st} directly determines α_t).

5.1 The Incremental Linearization Algorithm

ILA employs iteratively the following "*Initial Value Determination Algorithm*" (IVDA) and "*Cross-Section Calculation Algorithm*" (CSCA) in order to calculate all the unknowns of system (4.8). In order for the computed solution to correspond to a cross-section, all system's unknowns are calculated with respect to Theorems 2 and 3. Henceforth, all unknown values that satisfy both Theorems 2 and 3 are called as "valid values", while the produced cross-section is called as "valid cross-section".

Incremental Linearization Algorithm (ILA)

input system (4.8); output valid cross-section

Step 1: Select three initial system variables α_s, b_s, b_t that correspond to two adjacent regions R_s, R_t having a common sketch line e_{st} other than $y = 0$ or $x = 0$.

Step 2: Assign to α_s an arbitrary non-zero value, with $\alpha_s \in \mathbb{R}^*$.

Step3:

do

while (IVDA not equal *true*)

while (CSCA not equal *true*). □

Initial Value Determination Algorithm (IVDA)

input system (4.8); input initial value of α_s; output valid values for b_s, b_t, a_t; returns *true* if successful.

Step 1: Assign to b_s an arbitrary non-zero value $b_s \neq 0$, with $b_s \in \mathbb{R}^*$.

Step 2*: If* values (α_s, b_s) do not satisfy (C2) and $k_{st} - n_{st} b_s \neq 0$ return *false*.

Step 3: Assign an arbitrary non-zero value to b_t, with $b_t \in \mathbb{R}^*$.

Step 4: Find equation eq_{st} (see system (4.8)) and calculate a_t.

Step 5: *If* values (α_t, b_t) and (α_s, b_s) satisfy (C1) and (C2): Mark eq_{st} as "used" and return *true*, *else*: return *false*. □

Cross-Section Calculation Algorithm (CSCA)

input system (4.8); input values α_s, b_s, b_t, a_t; output system solution; returns *true* if successful.

<u>Step 1</u>: Repeat until all unknown pairs (a_i, b_i), with $i \in \{1,\ldots,R\} - \{s,t\}$, are calculated

<u>Step 1.1</u>: Find all equations that include one calculated pair (a_i, b_i) and mark them as "linear".

<u>Step 1.2</u>: Find all equations that include two calculated pairs and mark them as "used".

<u>Step 1.3</u>: For ($n = 1,\ldots,R$) do

<u>Step 1.3.1</u>: Find one non-calculated pair (a_n, b_n).

<u>Step 1.3.2</u>: For ($k = 1,\ldots,L$) find all "linear" equations $\{eq_{ni}\}$, with $i \in \{1,\ldots,R\}$ that include (a_n, b_n).

<u>Step 1.3.3</u>: *If* the number of equations in $\{eq_{ni}\}$ is greater than or equal to two: construct a system of equations S_n and *break*.

<u>Step 1.4</u>: If no S_n exists return *true*.

<u>Step 1.5</u>: If S_n is a $m \times 2$ system, with $m > 2$, transform it to a 2×2 system (see below).

<u>Step 1.6</u>: Calculate the determinant $\det[S_n]$ of S_n.

<u>Step 1.7</u>: *If* $\det[S_n] \neq 0$: solve S_n in terms of (a_n, b_n), else return *false*.

<u>Step 1.8</u>: *If* the calculated values (a_n, b_n) violate (C1) or (C2): return *false*.

<u>Step 2</u>: Return *true*. □

More specifically, CSCA employs, in each iteration, one linear sub-system S_n for the determination of each unknown pair (a_n, b_n). While each linear subsystem S_n always includes two unknowns, the number of equations in S_n varies depending on how many equations $eq_{ni}, i \in \{1,\ldots,R\}$ include the above unknown parameters and an already calculated pair (a_i, b_i).

If the system S_n, with

$$S_n : \begin{cases} a_n(k_{pn} - b_p n_{pn}) + b_n(n_{pn} a_p - m_{pn}) + b_p m_{pn} - a_p k_{pn} = 0 \\ a_n(k_{qn} - b_q n_{qn}) + b_n(n_{qn} a_q - m_{qn}) + b_q m_{qn} - a_q k_{qn} = 0 \end{cases}, \tag{5.1}$$

includes two equations, then it is solved using Cramer's rule [3]. Thus, when CSCA in Step 1.7 evaluates the determinant of S_n, it directly examines whether constraint (C3) of Theorem 3 is true.

In the opposite case that S_n, with

$$S_n : \begin{cases} a_n(k_{pn} - b_p n_{pn}) + b_n(n_{pn} a_p - m_{pn}) + b_p m_{pn} - a_p k_{pn} = 0 \\ a_n(k_{qn} - b_q n_{qn}) + b_n(n_{qn} a_q - m_{qn}) + b_q m_{qn} - a_q k_{qn} = 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_n(k_{kn} - b_k n_{kn}) + b_n(n_{kn} a_k - m_{kn}) + b_k m_{kn} - a_k k_{kn} = 0 \end{cases}, \tag{5.2}$$

includes more than two equations, a least squares method [3] is employed to estimate its solution. The $m \times 2$ system S_n is converted to a 2×2 system S_n': $CX = D$, with $C = A^T A$ and $D = A^T B$. Matrices A and B are respectively the coefficient and constant matrices of system (5.2). It is well-known that the determinant of S_n' is given by

$$\det(C) = {D_{pq}}^2 + {D_{pk}}^2 + \ldots + {D_{kq}}^2 , \tag{5.3}$$

where $D_{pq}, D_{pk}, \ldots, D_{kq}$ correspond to the determinants of all the 2×2 sub-systems that exist within Eqn. (5.2). Consequently, at least one determinant $D_{pq}, D_{pk}, \ldots, D_{kq}$ complies with constraint (C3) of Theorem 3.

5.1.1 Validity and Convergence of Incremental Linearization Algorithm

IVDA produces a set of valid initial values for three unknowns of system (4.8), on the basis of which a solution of the cross-section problem is derived. Consequently, convergence of IVDA guaranties the convergence of ILA. The following theorem establishes the convergence of IVDA.

Theorem 4: Let $V = \{a_s, b_s, b_t \in \mathbb{R}^* : \mathit{Theorem}\ 2 \ \wedge\ \mathit{Theorem}\ 3 \ \wedge\ \text{System (4.8)}\}$ be the set of initial valid values determined by IVDA. For a given value $a_s \in \mathbb{R}^*$, there exists at least one value for b_s and b_t, with $a_s, b_s, b_t \in V$.

Proof: With respect to Theorems 2 and 3, and system (4.8), equality in the constraints of these theorems designate all the non-valid values of b_s and b_t, when a_s is fixed. These non-valid values form a set V', with $V' \subset \mathbb{R}$ and $V' \cap V = \varnothing$. Thus, $V \neq \varnothing$, which establishes the convergence of IVDA. □

Bellow we prove that the produced solution is also a “non-trivial one”, i.e., (a) the cross-section lines are not collinear, and (b) not all unknowns (a_i, b_i) equal to zero.

Theorem 5: Given the values of the three variables $a_s, b_s, b_t \in V$, Incremental Linearization Algorithm always produces a non-trivial cross-section of a sketch.

Proof: System (4.8) includes R unknowns and L equations. Given the values of three variables α_s, b_s, b_t, corresponding to the cross-section lines of two adjacent regions, one can obtain the value of a_t from equation eq_{st}. Subsequently, the number of unknowns in system (4.8) reduces to $2R-4$, and the number of equations to $L-1$. In [7], we proved that $L-1 \geq 2R-4 \Leftrightarrow L \geq 3$. This, combined with the fact that each unknown pair (a_i, b_i) appears in $\deg(R_i) \geq 3$ equations of system (4.8), establishes that for the calculation of each pair (a_i, b_i) there exists at least one linear system. It remains to prove that the produced cross-section is not trivial. Constraint (C1) of Theorem 2 asserts that the cross-section lines are not identical. Then, consider a generating sub-system S_n (see Eqn. (5.2)). This sub-system has a non-trivial solution (i.e., not all unknowns equal to zero) when it is non-homogeneous and its determinant is non-zero. Indeed, constraint (C2) of Theorem 2 and (C3) of Theorem 3, respectively, establish that constants of system (5.2) and the determinant of S_n (see Eqn. (5.3)) do not equal to zero. □

5.2 Compatibility Evaluation of the Cross-Section

In this section, the cross-section produced by the Incremental Linearization Algorithm is evaluated in terms of compatibility. A direct evaluation method is achieved by substituting the produced solution back into the equations of system (4.8). If the solution satisfies system (4.8) at some acceptable level of accuracy, the cross-section is compatible with the given sketch. On the basis of this, a sensitivity analysis of ILA is performed with respect to the accuracy level of the input sketch, i.e., the numerical precision of the coordinates of its junctions.

Regarding the cross-section problem as a geometric problem, the cross-section is compatible with a sketch if and only if the cross-section point P_{ij} (Eqn. (4.5)) is on (the extension of) the sketch line e_{ij}, or equivalently if the distance

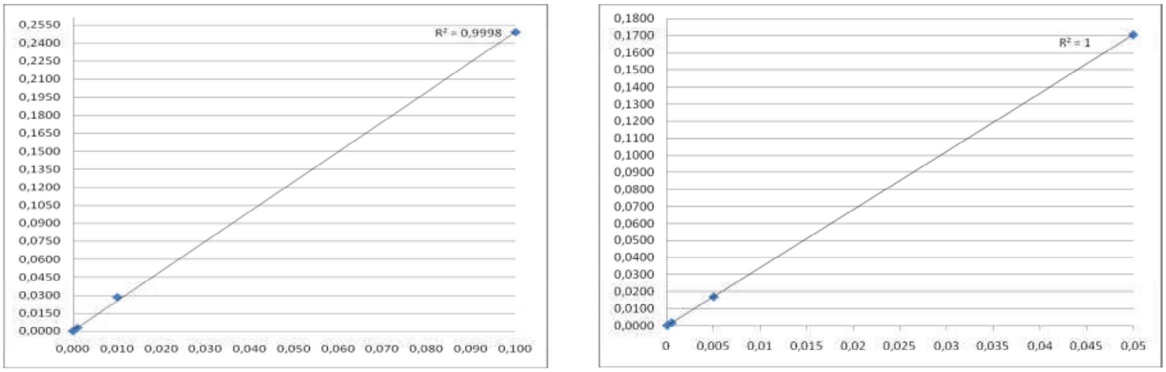


Fig. 4: The maximum distance error $d_{\max}$ with respect to (a) $E = 1 * 10^{-r}, r = 1,\ldots,6$, and (b) $E = 5 * 10^{-r}, r = 1,\ldots 6$.

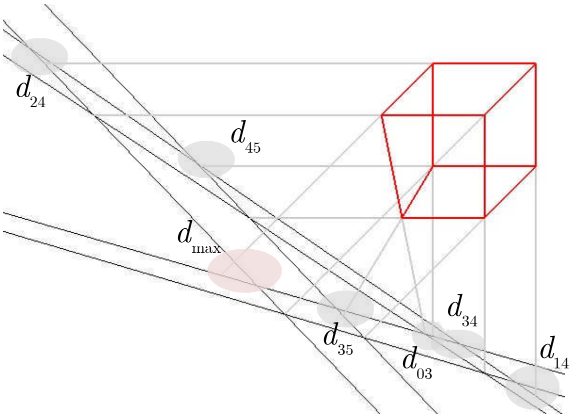
$$d_{ij} = \frac{\left| \frac{(a_j - a_i)}{(a_i b_j - a_j b_i)} k_{ij} + \frac{(b_i - b_j)}{(a_i b_j - a_j b_i)} m_{ij} + n_{ij} \right|}{\sqrt{{k_{ij}}^2 + {m_{ij}}^2}} \tag{5.4}$$

of point P_{ij} from e_{ij} equals to zero. Considering system (4.8), one can prove that each equation eq_{ij} is equivalent to $d_{ij} = 0$. Thus, compatibility of a cross-section with a given sketch can be evaluated with respect to the accuracy level for all d_{ij}.

The sensitivity analysis is performed in terms of a given error E associated with the coordinates of the sketch's junctions; i.e., each junction's "perturbed" coordinates are in the form $(x+E, y+E)$. Two case studies are performed using ILA. In the first case study, for a given set of initial variables, the maximum distance value $d_{\max} = \max(d_{ij})$ is calculated for a variety of error values according to $E = n * 10^{-r}$, with $n = 1,\ldots,9$ and $r = 1,\ldots,6$. The results indicate a linear correlation between E and $d_{\max}$ as it is shown in (Fig. 4) for $n = 1,5$ and $r = 1,\ldots,6$.

In the second case study, for a typical sketch with error $E = n * 10^{-r}$, $n = 1,\ldots,9$, different cross-sections are considered and the corresponding $d_{\max}$ values are obtained. The obtained results designate that for a given sketch with $E = n * 10^{-r}$, $1 \le n < 5$, the 85% of the calculated distance errors are $d_{\max} < 10^{-(r-1)}$. A standard deviation analysis for $d_{\max}$ showed that the order of the standard deviation error σ is 10^{-r}. The above analysis allows an evaluation of sketch realizability with respect to a preferred level of accuracy $E = n * 10^{-r}$, $1 \le n < 5$. Indeed, following the standard deviation analysis method, if μ stands for the mean value of $d_{\max}$ then setting a threshold $d_{\max} \le 2\sigma + \mu$ will allow for the successful evaluation of at least 95% of the cross-sections produced by the proposed ILA. After a thorough study of the experimental results we concluded that by setting an upper limit for $\mu = 10^{-(r-1)}$, the produced threshold $d_{\max} \le 2 * 10^{-(r-1)}$ provides cross-sections which comply with the anticipated rate.

The above sensitivity analysis allows us to employee ILA within a "sketch-to-solid" algorithm in order to determine the hidden geometry of an input natural "imperfect" sketch and to construct a 3D solid whose projection is identical with this sketch. If a generated by ILA cross-section is found not compatible with the input sketch, according to the above tolerance criterion, then two cases are



Tolerance parameter: $d_{\max} \leq 2*10^{-4}$

Acceptable error: 10^{-5}

Sketch error: 10^{-1}

$d_{\max}$ = *0.646493307* (d_{23})

d_{24} = *0.295916161*	d_{03} = *0.000449064*
d_{45} = *0.061231349*	d_{34} = *0.531433071*
d_{35} = *0.188047439*	d_{14} = *0.386387151*

Fig. 5: The threshold value for the tolerance parameter d_{ij} assesses the compatibility of the generated cross-section with the sketch. In this example, $d_{\max} > 2*10^{-4}$ and the sketch is evaluated as non-realizable.

examined: (a) if $\frac{d_{\max}}{10^{-(r-1)}} < 10$ then we re-run ILA in order to produce another valid cross-section (5% of criterion failure). If this case is repeated for the same sketch for more than 100 times then the sketch is not realizable. (b) if $\frac{d_{\max}}{10^{-(r-1)}} \geq 10$ the given sketch is not realizable.

An example illustrating a non-compatible cross-section generated by an inexact sketch with a junction error of order 10^{-1} is given in Fig. 5: all lines within highlighted circles should intersect at a “unique” point under the accuracy level $d_{\max} \leq 2*10^{-4}$. In this case, the final maximum error is found d_{23}=0.646493307 which renders the given sketch non-realizable according to the above accuracy criterion.

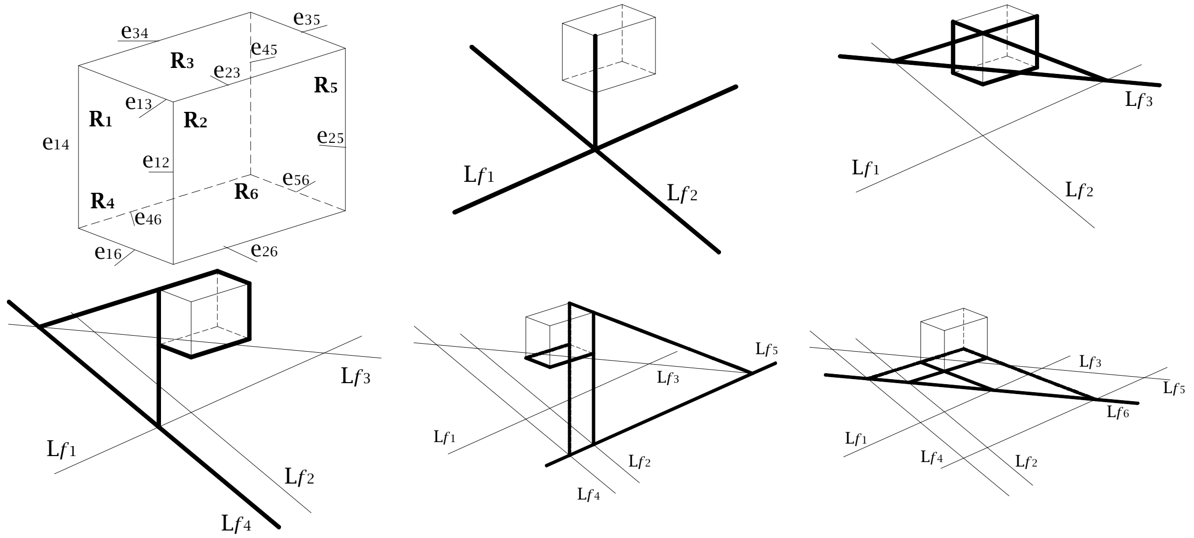


Fig. 6: From left to right, (a) a wireframe sketch, (b) the result of the “Initial Value Determination” algorithm, (c)-(f) the four iterations of the “Cross-Section Calculation” algorithm produce a cross-section for the wireframe sketch in (a).

Sketch	A	B	C	D	E	F	G	H
$d_{\max}$	$142.4\cdot10^{-6}$	$19.9\cdot10^{-6}$	$33\cdot10^{-6}$	$95.4\cdot10^{-6}$	$53\cdot10^{-6}$	$178.6\cdot10^{-6}$	$36.7\cdot10^{-6}$	$56.4\cdot10^{-6}$

Tab. 1: The $d_{\max}$ value of the sketches shown in Fig. 7 (from left to right and top to bottom).

6 EXAMPLES

In order to analyze the proposed algorithm, we first demonstrate the application of ILA along with a graphical representation of the solution for a "trivial" case of a wireframe sketch corresponding to an orthogonal parallelepiped (Fig. 6(a)). IVDA, for $\alpha_1,\ b_1,\ b_2$, produces cross-section lines L_{f_1} and L_{f_2} (Fig. 6(b)). In the first iteration (Fig. 6(c)) of CSCA, the equations $\{eq_{13},eq_{14},eq_{16},eq_{23},eq_{25},eq_{26}\}$ become linear, and during Step 1.7 the unknown pair (a_3,b_3) is calculated using the linear sub-system $S_3=\{eq_{13},eq_{23}\}$. Accordingly, in the second iteration (Fig. 6(d)) the set of "not used" linear equations is $\{eq_{14},eq_{16},eq_{25},eq_{26},eq_{34},eq_{35}\}$, and the unknowns (a_4,b_4) are calculated using the system $S_4=\{eq_{14},eq_{34}\}$. The third iteration (Fig. 6(e)) involves the equation set $\{eq_{16},eq_{25},eq_{26},eq_{35},eq_{45},eq_{46}\}$, and employs $S_5=\{eq_{25},eq_{35},eq_{45}\}$ to calculate (a_5,b_5). Finally, in the fourth iteration (Fig. 6(f)), all linear and "not used" equations $\{eq_{16},eq_{26},eq_{46},eq_{56}\}$ form a linear system for calculating (a_6,b_6).

More complicated sketches which include noisy junctions (i.e., error order 10^{-5}) have been tested with success. Examples are given in Fig. 7, which illustrates the cross-sections that are produced by ILA for the given wireframe sketches. All produced cross-sections are found compatible according to the tolerance parameter $d_{\max}\le 2*10^{-4}$. The resulting $d_{\max}$ value of each sketch is demonstrated in Tab. 1.

Test results showed that for a typical sketch that includes 20 junctions, 30 lines, and 12 regions, ILA involves, on the average, 70 iterations of IVDA, 6 iterations of CSCA, and produces a cross-section in less than 0.1 seconds using a modern PC equipped with a Core i7 CPU.

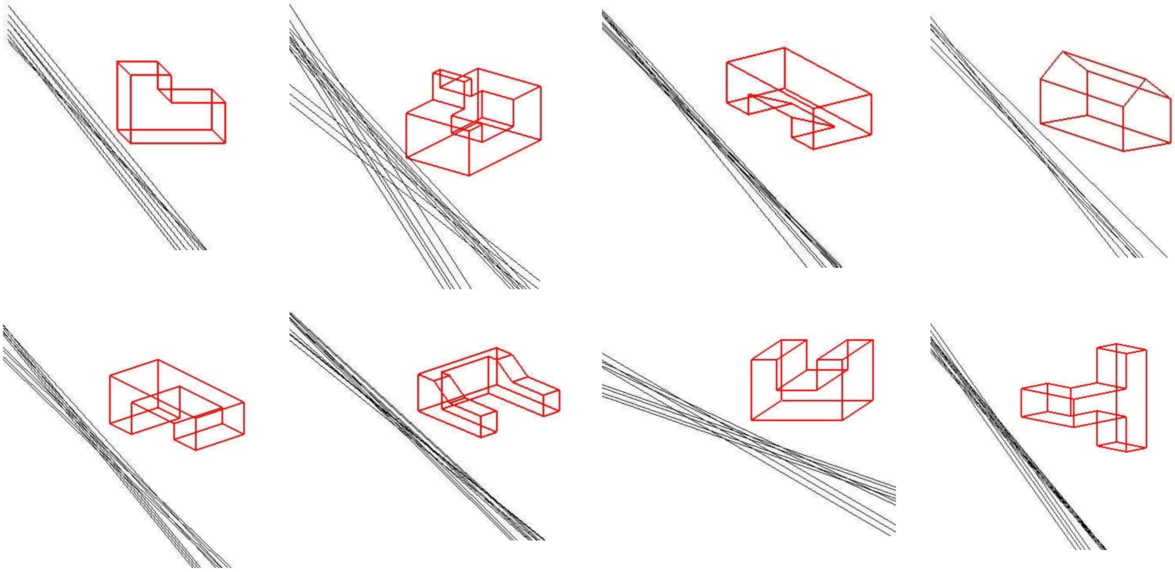


Fig. 7: Compatible cross-sections of the given wireframe sketches produced by the Incremental Linearization Algorithm.

7 CONCLUSIONS

The cross-section criterion forms a robust geometric tool to check realizability of a given sketch. The main advantage of this criterion is that it provides a necessary and sufficient condition to assert the existence of a polyhedron from a given sketch, without using any heuristic rules, such as the “Line Labeling” scheme. This paper surpasses the classical geometric “ruler-and-compass” framework for the generation of a cross-section from a sketch by employing an algebraic description of the cross-section problem and providing an algorithmic solution to the underlying bilinear system. The presented approach is based on the “Incremental Linearization Algorithm” (ILA) to effectively evaluate the realizability of a given (imperfect) sketch. Test results show that the proposed algorithm is able to evaluate an imperfect wireframe sketch with a 95% success ratio. In addition, the evaluation criterion can be adjusted according to the accuracy level of the input sketch. Thus, ILA can be effectively employed within a “sketch-to-solid” algorithm in order to (a) facilitate the process of the “hidden geometry determination” that is required for natural sketches, (b) correct the geometry of a given wireframe sketch, and (c) generate a compatible cross-section for constructing the trihedral polyhedron from a given sketch. Our future research will focus on all three aforementioned issues.

8 ACKNOWLEDGEMENTS

The authors express their deep appreciation to the two referees of this paper and to the CAD’11 Chair for their detailed criticism of the manuscript that has led to significant improvements. This research was partially funded by the European Union and the Greek Ministry of National Education and Religious Affairs through the “Heraclitus Research Grant: Preliminary Industrial Design: Geometric/Information Models and Methods for Interactive Product-Design”.

9 REFERENCES

[1] Clowes, M.B.: On seeing things, Artificial Intelligence, 2(1), 1971, 79-116, doi:10.1016/0004-3702(71)90005-1.

[2] Cooper, M.C.: Linear-time algorithms for testing the realisability of line drawings of curved objects, Artificial Intelligence, 108, 1999, 31-67, doi:10.1016/S0004-3702(98)00118-0.

[3] Strang, G.: Linear Algebra and its Applications, Harcourt Brace Javanovich Publishers, 1988.

[4] Huffman, D. A.: Impossible objects as nonsense sentences, Machine Intelligence, 6, 1971, 295-323.

[5] Kyratzi, S.; Sapidis, N.: An Interactive sketching method for 3D object modeling, In Proceedings of the 3rd ACM Conference on Digital Interactive Media in Entertainment and Art DIMEA’08, 2008, 335-342.

[6] Kyratzi, S.; Sapidis, N.: Extracting a polyhedron from a single view sketch: Topological construction of a wireframe sketch with minimal hidden elements, Computers and Graphics, 33(3), 2009, 270-279, doi:10.1016/j.cag.2009.03.001.

[7] Kyratzi, S.; Sapidis, N.S.: From sketch to solid: An algebraic cross-section criterion for the realizability of a wireframe sketch, Computing, 86(2-3), 2009, 219-234, doi:10.1007/s00607-009-0058-3.

[8] Kyratzi, S.; Sapidis, N.S.: 3D object modeling using sketches, to appear in International Resource Management Journal.

[9] Lipson, H.: Computer Aided 3D Sketching for Conceptual Design, Ph.D. Thesis, Israel Institute of Technology, 1998.

[10] Maxwell, J. C.: On reciprocal figures and diagrams of forces, Philosophical Mag. Ser., 4 27, pp 1864, 250-261.

[11] Ros, L.; Thomas, F.: Analysing spatial realizability of line drawings through edge-concurrence tests, In: Proc. of IEEE Int. Conf. on Robotics and Automation, 1998, 3559-3566.

[12] Ros, L.; Thomas, F.: Geometric methods for shape recovery from line drawings of polyhedra, Journal of Mathematical Imaging and Vision, 22(1), 2005, 5-18, doi:10.1007/s10851-005-4779-4.

[13] Sapidis, N.; Kyratzi, S; Azariadis, F.: Improved computational tools for concept development based on sketches and advanced CAD technologies, Computer Aided Design and Applications, 2(6), 2005, 707 – 716.

[14] Stolfi, J.: Oriented Projective Geometry, ACM Press New York, USA, 1987.

[15] Sugihara, K.: Machine Interpretation of Line Drawings, The MIT Press, 1986.

[16] Sugihara, K.: Resolvable representation of polyhedral, Discrete and Computational Geometry, 21(2), 1999, 243-255, doi:10.1007/PL00009419.

[17] Varley, P.: Automatic Creation of Boundary-Representation Models from Single Line Drawings, Ph.D. Thesis, Cardiff University, 2002.

[18] Varley, P.; Martin, R.; Suzuki, H.: Frontal geometry from sketches of engineering objects: Is Line Labelling necessary?, Computer-Aided Design, 37(12), 2005, 1285-1307, doi:10.1016/j.cad.2005.01.002.

[19] Waltz, D.: Understanding line drawings of scenes with shadows, The Psychology of Computer Vision (P. H. Winston cd.), McGraw Hill, 1975, 19-91.

[20] Whiteley, W.: Realizability of polyhedral, Structural Topology, 1, 1979, 46-58.

[21] Whiteley, W.: Weavings, sections and projections of spherical polyhedral, Discrete Applied Mathematics, 32(3), 1991, 275-294, doi:10.1016/0166-218X(91)90004-G.