CORRECTION OF A 2D SKETCH FOR 3D RECONSTRUCTION

Phil Kuzo, Pierre Macé

Departement d'Informatique
Ecole des Mines de Nantes
4, rue Alfred Kastler
44307 Nantes, cedex 03 FRANCE
(kuzo@emn.fr, mace@emn.fr)

ABSTRACT

Among the different methods of 3D modeling, 3D reconstruction from sketches may be quoted. These methods have all to deal with the hard problem of errors on the sketch, leading to a minimization problem. We propose to distinguish, on the one hand, the violation of projective constraints, and on the other hand, the metrics errors, and to first correct the sketch in the way to respect only the projective and affine constraints. The methods was tested on sketches involving projective constraints.

Keywords: 3D reconstruction, sketch correction, projective space, projective geometry, known state propagation.

1 INTRODUCTION

Much significant developments are made in the field of Computer Aided Design. But despite this effort, designers still tend to prefer paper and pencil before using CAD systems. Most of these systems are maladjusted to a conception process because they break designers's focus. Designers need a quick and natural method to modelize 3D scenes. So, at the moment, much efforts are being provided to make designing more natural and CAD systems easier to use. The inputs of these systems are freehand sketches and the outputs are 3D recovery of the sketches. These systems may have many applications : input of CAD models, mechanics, architectural drawings, ...

The 3D reconstruction from two or more orthographic projections is well known but not the reconstruction from perspective projections (see [roger]). The reconstruction process from perspective projection on a plane induces under-constraint systems of equations : only one perspective view is not sufficient to the recovery of the 3D model.

In the **Gina** (Géométrie Interactive Naturelle) project, object models are defined through simple freehand sketch and textual or oral declarations which are constraints verified by the differents parts of the object. In **Desargues**, the constraint solver of **Gina,** the sketch is considered as a 2D projection of the 3D model. **Desargues** makes distinction between projective properties (properties that are preserved in projection) and metric ones, and, when it is possible, the sketch is corrected in the way to let it verify the projective constraints.

In [kuzo98] and [lhomm98], we have shown how a 3D object may be found from a correct sketch in the **Desargues** system. Next, we present **2D3P** our sketch correction algorithm. Before this, related work is reviewed.

2 RELATED WORK

Much efforts have been made in reconstructing 3D model from 2D input. Most of them analyse the 2D input and interpret the 3D geometric shapes from the sketches.

This section describes some of these researches.

Eggli, Hsu, Brüderlin and Elbert present, in [eggli97], *Quick-Sketch* a 2D and 3D modeling tool for pen-based computers. Exact shapes and geometric constraints are interpreted from the sketch by using some mode dependent preference function. Once the type of the shape is decided, the closest fit to the strokes is determined using different numerical techniques and then the system tries to interpret some relationships between them by applying some criterium like the closeness to the exact shape, the speed of the input, a user settable tolerance, the length of the stroke, and the user preference mode which represents the type of the task. A similar idea is found in [lipso96] by Shpitalni and Lipson. Their system is a non-interactive optimization-based algorithm for reconstructing a 3D model from a single, inaccurate 2D edge-vertex graph.The system analyses and interprets single 2D inaccurate line drawing input and reconstructs the 3D object. No additional information is available regarding the 3D object itself but the algorithm must achieve the most plausible reconstruction. The reconstruction algorithm starts by transforming the sketch into a 2D line and junction graph. So the graph is analysed and some image regularities are identified. Each regularity is expressed by a mathematical term and is included in a compliance function which serves as a target for minimization. The term evaluates to a value denoting how well a specific image regularity agrees with its associated 3D configuration. This 3D configuration must be the most plausible selected by a human observer.
This idea had been exploited by Leclerc and Fiscler in [lecle92] where they presented an optimization of the Marill's algorithm ([maril91]) supposedly able to duplicate human judgment in recovering the 3D wire frames geometry of objects depicted in line drawings. In their systems the depth of vertices of a body represented in a line drawing projection is optimized until the minimum-standard-deviation-of-angles (MSDA) between line pairs at junction is reached (Marill). Face planarity is considered as well (Leclerc and fiscler). For similar works, we can also note the Grimstead and Martin's ones ([grims95]). They proposed an approach using single orthographic view of a trihedral object with hidden lines removed and without any holes. Like Leclerc [lecle92] and Marill 's work [maril91], their method accept a complete wireframe drawing as input and use an optimization approach by gradually disturbing the depth of each vertex from an initially flat drawing to a 3D wireframe, minimizing the standard deviation of angles between connected lines. But unlike the others, their system accepts a hidden line removed drawing and take into account drawing errors. The techniques

used by Grimstead and Martin are line labelling, least squares systems solving, skewed symmetry, completion of hidden faces, tidying process...

2.1 Discussion

The key idea of these systems is that they interpret the 3D shapes of the objects from the 2D input. They deal with psychological assumptions to produce the 3D models. So they produce models the most plausible to a human observer. Contrarily to these works, we decided to add the 3D constraints to our sketch. So the 3D models found by our system verify, strictly, these constraints. Another difference beetween the two approaches is that our system corrects in first, when it is possible, the sketch before the 3D recovery. This allow us to avoid the global minimization problem like the least square methods, and to use known state propagation algorithm.
Most of above work use orthographic views, and the methods used in that case was generalized and completed for perspective views. For example, see [ulipi91]. Although this type of projection is difficult to sketch, and too errors prone in quick freehand sketching, it contains more information than an orthographic view and architects prefer using of perspective views rather than orthographic ones (because of a better understanding of the depth). So our sketches are, in general, perspective views of 3D models.

3 THE DESARGUES SYSTEM

The input to **Desargues**, our geometric constraint solver, is a sketch - which represents a projection of a 3D form - and constraints verified by the form (oral or textual expressions of properties given by the user). Now, only projective and affine constraints are assumed (colinearity, coplanarity, parallelism...).
Desargues accepts a hidden line removed drawing and reconstruct the visible points and the hidden ones, provided that the constraints relying the latter to the other points are written.
The elementary objects : points, lines, planes are represented in projective space by the use of homogeneous coordinates. The Grassmann-Cayley Algebra gives a formalism very suited to projective geometry. Its three operators *join, meet,* and *dual* (($\vee, \wedge$, and $*$)) allow classical operations on the elementary objects like generation, intersection ... That allows us to perform geometrical reasonning as well as numerical calculus [mace97]. In **Desargues**, the affine and projective constraints are decomposed in a set of incidence constraints. The reasonning performed is facilitated by the fact that every 3D point is related to its projection by the use of a specific data structure. So this relation is maintained all over the process. The 3D constraints are solved by a known data propagation method inferring the projective properties. For instance, a point is known if it verifies a set of incidence constraints to objects already known by **Desargues**: a point may be determined by:

- intersection of two lines,
- intersection of a line and a plane,
- intersection of 3 planes.

In the way to have no contradiction between redundant data, we use numerical calculus with infinite precision. **Desargues** accepts any drawing of polyhedrons (convex or not, with or without holes) on condition that the drawing is strictly correct (with an infinite accuracy). So, whatever the care taken by the user, it is necessary to compute again the correct coordinates. On the other hand, our purpose is not to force the user to respect the graphical constraints. He has to think about the “qualitative” aspect and the computer must add the “quantitative” one.

4 SKETCH CORRECTION

Before the correction algorithm, let’s see an example of correct sketch, an example of a false sketch (corrected by our system) and an example of a sketch refused by the system.

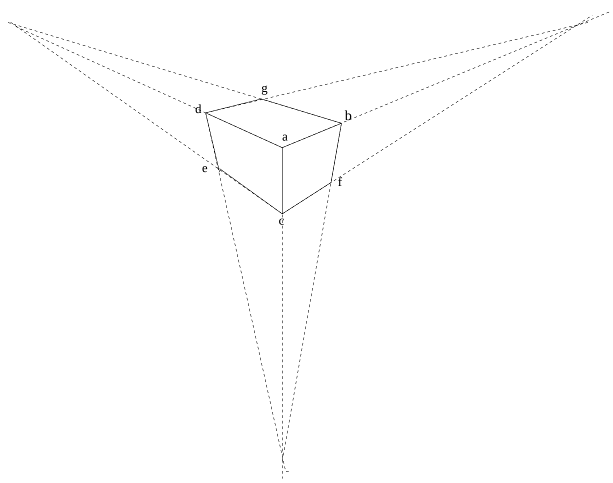

Figure 1: a correct parallelepiped

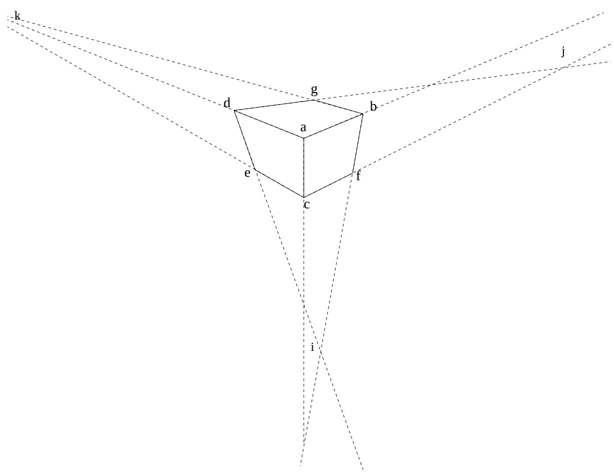

Figure 2: an incorrect parallelepiped

The first parallelepiped is correct because all the parallel lines are secant in an unique point. This point is the projection of the infinitely distant point of the

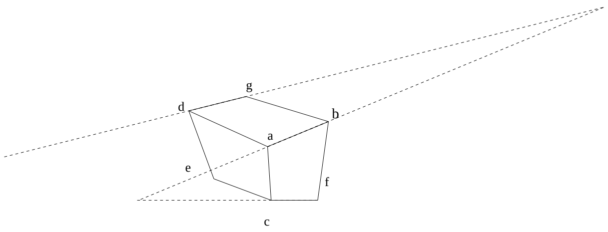

Figure 3: an impossible parallelepiped

parallel lines. That is not the case in the second parallelepiped: the projection of the parallel lines are secant in three differents points. This parallelepiped can be corrected in the way to have one vanishing point for a set of three parallel lines. The third sketch is refused because the intersection points of the three couples of lines are not in the same side of the sketch. We say it is “qualitatively incorrect”.

The correction process is based on the non-violation of the projective and affine constraints. For example, three aligned points may be replaced by their projections on their best line fit and a set of parallel lines are incident to their infinitely distant point. Their projections must be secant on one point. We say that a point is *related* to another point if there is a line passing through the two points.

Here is the algorithm of the correction process (for the starting points, see section4.2):

algorithm

```
V = {corrected vanishing points}
L = {all the other points}
M = V  /* the set of correct points*/
while L non-empty do
  N = {starting points}
  add N to M
  retrieve N from L
  for p in L
  if p  related to 2 correct couples
    of points (P1, P2), and (P3,P4) then
```

```
      D1 = line(P1,P2)
      D2 = line(P3,P4)
      p = intersection(D1,D2)
      add p to M
      retrieve p from L
   if p  incident to one couple
   of correct points (P1,P2)
      then
      D = line(P1,P2)
      p = projection_on(D)
      add p to M
      retrieve p from L
  end for
end while
```

The system starts by computing the different vanishing points according to the parallelism constraints. Then it classifies the points by labelling them *correct* (the beginning points) or *incorrect*. After that, the points correction are processed by a simple known state propagation algorithm.

4.1 Vanishing points

Different sets of parallel lines have different vanishing points. This is illustrated in the figure 4.

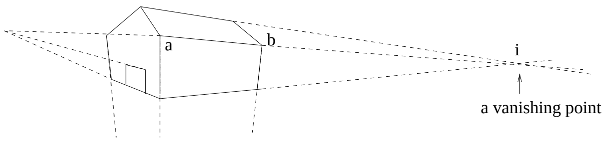

Figure 4: three vanishing points

The vanishing point must be unique for every set of parallel lines. So we decide to take the closest point to the set of lines. That involves the minimization of a distance. This problem is the dual of the known least square line fit problem. It is a metric problem and we choose the euclidian metric with cartesian coordinates. It may be stated like this : Let

$$l_i x + m_i y + n_i = 0$$

with $l_i^2 + m_i^2 = 1$, the equation of the line L_i. We have to minimize the sum $2e$ of the square of the quadratic distances between the concerned point and the lines L_i, $1 \leq i \leq n$.

$$e = \frac{1}{2} \sum (l_i x + m_i y + n_i)^2$$

By zeroing the partial derivative regard to x and y, this problem leads to the resolution of a system of two linear equations:

$$\begin{cases} (\sum l_i^2)x & + & (\sum l_i m_i)y & = & -\sum l_i n_i \\ (\sum l_i m_i)x & + & (\sum m_i^2)y & = & -\sum m_i n_i \end{cases}$$

Discussion

If the set of the projections of the parallel lines is already secant in one point, the algorithm finds again this point. So it is not necessary to verify the correctness of the sketch before the propagation. If the projected lines are parallel (f. i. orthographic projection), then our algorithm would fail since distance between two infinitely distant points has no sense. But this case is detected (by a null determinant) and do not require preliminary test. In that case, we take the infinitely distant point of the lines to be their vanishing point.

4.2 Starting points

According to the scene's degree of freedom, we can assume some points to be correct. For example, the house in the figure 4 has one degree of freedom in the 3D space (there is a set of homothetic houses which have the same perspective projection). So we can fix one coordinate of one point. In the same way, we suppose that there is one correct point on the 2D sketch. We choose the projection of the nearest point to the eye position. This point may be found by studying the relative position of every couple of points regard to their vanishing point : this nearest point is that one which is related to the all vanishing points and is in the opposite side of the vanishing points regard to the second point. For example in the figure 4, a is in the opposite side of i regard to b, it is the nearest point to the eye position.

4.3 Propagation

We create a set M containing the correct points. At the beginning, it contains the given point(s) by the above process and the corrected vanishing points. The incidence constraints are propagated. A line is labelled *correct* once it passes through two points labelled *correct*. A point is recomputed if it is incident to at least one correct line :

- by projection, if it is incident to one correct line
- by intersection, if it is incident to two correct lines.

And then, it is labelled *correct*. For hidden lines removed drawings, there exists a propagation order enabling the entire correction process.

For example, the figure 2 is corrected in this way :
The vanishing points i, j, k of the set of lines $\{dg, ab, cf\}$, $\{de\ ,\ ac,\ bf\}$and $\{ad, bg,\ ec\}$ are computed. The system finds the point a to be the nearest point to the eye, and the set M contains points a, i, j, k. The point b must be on the line ak, so the correct b is found by projecting the given b onto the line ak. So on for c and d : d is projected onto aj, and c onto ai.

At this stage, a, b, c, d, i, j, k are correct. The point e related to (c, j) and (d, i). f incident to (b, i) and (c, k), g is incident to (b, j) and (d, k). So :

- the correct e is found by intersecting cj by di.
- f is found by the intersection of bi and ck.
- And g is the intersection of bj and dk.

5 LIMITATIONS OF THE METHOD

We have presented a system which takes a freehand sketch and constraints among differents parts of the objects, corrects the sketch and produces a 3D model. The method is simple and fast, since it is possible to avoid global least square methods, but our correction greatly depends on the propagation order. The correction may be very different from a point to another (see figures 5 and 8), and may change according to the order of the data. So it is possible that the user want a new correction: he may take care of the proposed corrections to draw a new sketch.

Our system is suited to a limited, but not less interesting domain : architecture.

6 CONCLUSION

The method currently works for some objects involving projective and affine constraints. It has been tested successfully on polyhedrons with only three vanishing points. Now, **Desargues** can be used for several problems of architecture and it may constitute a kernel of an interactive system for conception and modeling.

We are studying orthogonality constraints and a planification algoritm in order to correct the drawing of any polyhedron objects. A multimodal interface for the sketch and the oral declaration capture is being studying. These works and the introduction of metric constraints may lead to a new generation of 3D modelers.

6.1 Figures

Here is some examples of sketches correction and their reconstruction by **Desargues**: a house and a staircase.
In each drawing there are hidden lines (respectively 2 and 1 hidden points). We have given the properties of these lines (incidence, parallelism), so it is possible to determine them. The two systems were over-constrained but consistent (after the correction process). As **Desargues** works with an infinite accuracy, it finds the only solution. **Desargues** fails on non-consistent systems. When the system is under-constrained, it is possible to get formal expressions for the class of solutions.

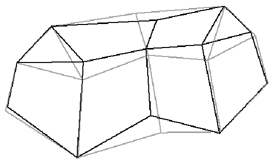

Figure 5: a house (dashed) and its correction

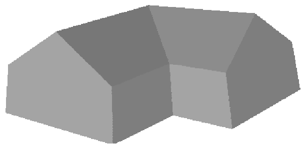

Figure 6: a view of its 3D reconstruction

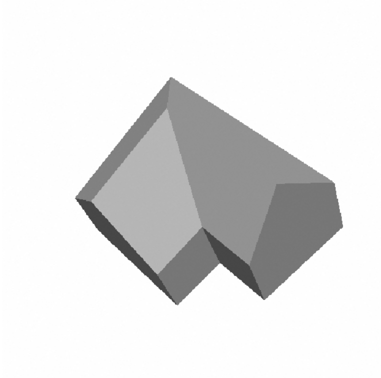

Figure 7: another view

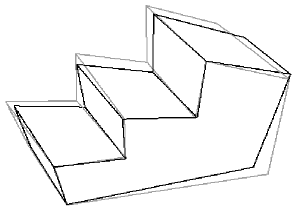

Figure 8: a staircase and its correction

REFERENCES

[eggli97] Eggli, L.,Hsu, C- Y., Brüderlin, B., Elbert, G.:*Inferring 3D models from freehand sketches and constraints,* Computer-Aided Design, Vol. 29, No. 2, pp. 101–112, 1997

[grims95] Grimstead,I.J., Martin,R. R.:*Creating Solid Models From Single 2D Sketches,*ACM SIGGRAPH, Proc. third Symp. on Solid Modeling Applications, pp. 323–337, 1995

[kuzo98] Kuzo,P., Macé,P., Lhomme,O., *A constraint-based system for 3D geometric modeling,* 3'IA Limoges, conference proceedings, pp. 135–143, may 1998

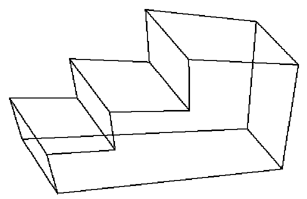

Figure 9: a view of its 3D reconstruction

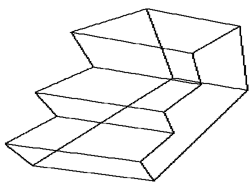

Figure 10: another view

[lecle92] Leclerc, Y. G., Fiscler, M. A.:*An optimization-based approach to the interpretation of single line drawings as 3D Wire frames,* Int. J. Compt. Vision Vol. 9 No. 2, pp. 113–136, 1992

[lhomm98] Lhomme, O., Kuzo, P., Macé, P.:*Desargues : a constraint-based system for 3D projective geometry,* Geometric Constraint Solving and Applications, eds : B. Brüderlin, D. Roller, Springer-Verlag, 1998

[lipso96] Lipson,H., 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

[mace97] Macé, P.:*Tensorial Calculus of Line and Planes in Homogeneous Coordinates,* Computer Networks and ISDN Systems 29 (1997) pp. 1695–1704, Elsevier-Science B.V.

[maril91] 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.

[roger]
Rogers, D.,Adams, J. A.:*Mathematical Elements for Computer Graphics, ed: Mc-Graw-Hill International Editions*

[ulipi91]
Ulupinar, F.,Nevatia, R.:*Constraints for interpretation of line drawings under perspective projection,* CVGIP; Image Understanding, Vol. 53, No. 1, pp. 88–96, jan 1991