

# Estrategias de Reconstrucción 2D en Interfaces Caligráficas

Ferran Naya, Manuel Contero, José Luis Saorín<sup>(1)</sup>, Julián Conesa<sup>(2)</sup>

*Departamento Expresión Gráfica en la Ingeniería. Universidad Politécnica de Valencia  
Camino de Vera, 46022 Valencia  
Tlfn: 963879516 Fax: 963879513 E-mail: fernasan@degi.upv.es*

<sup>(1)</sup> *Dpto. Expresión Gráfica en Arquitectura e Ingeniería. Universidad de La Laguna  
Avda. Ángel Guimerá Jorge, 38204 La Laguna Tenerife*

<sup>(2)</sup> *Departamento Expresión Gráfica. Universidad Politécnica de Cartagena  
Dr. Fleming, 30202 Cartagena*

## Resumen

Los sistemas CAD actuales presentan una excesiva rigidez en las primeras etapas del proceso de diseño, donde los bocetos realizados con lápiz y papel constituyen todavía la herramienta fundamental que permite expresar la creatividad del ingeniero. En la actualidad el grupo de investigación REGEO, trabaja en una aplicación prototipo que permite la generación automática de modelos sólidos de objetos 3D poliédricos, a partir del esbozo a mano alzada de una vista axonométrica de los mismos. Esta aplicación integra Interfaces Caligráficas y Reconstrucción Geométrica. Para poder generar el modelo 3D, se requiere una etapa de preproceso, denominada Reconstrucción 2D, que realice las tareas necesarias de limpieza y ajuste de los datos de entrada. El objetivo de esta etapa es convertir los bocetos realizados por el usuario, en figuras 2D geoméricamente consistentes, que sirvan para generar los modelos tridimensionales. La descripción de este proceso de reconstrucción 2D es el objetivo de la presente comunicación.

**Palabras Clave:** Interfaces Caligráficas, Reconstrucción Geométrica.

## Abstract

CAD systems still show an excessive stiffness at the first stages of the design process, in which pen-and-paper sketches are the basic tools to express the engineer's creativity. The research team REGEO has developed a prototype application to automatically generate 3D solid models, from freehand 2D drawings. This application integrates geometric reconstruction and calligraphic interfaces (graphical user interfaces based on freehand sketches and gestures). A pre-processing module to cleanup input data and adjust edges, is required to generate the 3D model. This stage, 2D Reconstruction, consists of transforming the sketches into geometrically robust figures which could be used for generating 3D models. This paper presents that 2D reconstruction phase.

**Keywords:** Calligraphic Interfaces, Geometric Reconstruction.

## 1. Introducción

El grupo de investigación REGEO ha desarrollado en los últimos años una intensa actividad investigadora alrededor del tema de la reconstrucción geométrica,

desarrollando diferentes algoritmos en este campo [1]. En la actualidad, el grupo trabaja en una nueva línea de trabajo orientada a la creación de nuevos paradigmas de uso en la interacción hombre-máquina. Por ello, complementando el trabajo realizado en el campo de la reconstrucción 3D, la implementación de interfaces caligráficas [2] puede facilitar un nuevo tipo de aplicaciones, que permita generar de forma interactiva modelos virtuales 3D, partiendo de dibujos a mano alzada en 2D.

Actualmente se ha desarrollado una aplicación prototipo que permite la generación automática de modelos sólidos tridimensionales del tipo normalón y cuasi-normalón, a partir del boceto de una perspectiva axonométrica del mismo. Se considera que una figura representa un normalón cuando hay tres direcciones principales, es decir, cuando los enlaces de la figura están orientados según tres y sólo tres direcciones. Un modelo se considera cuasi-normalón si cumple la condición de que la eliminación de todas las aristas no paralelas a tres direcciones principales no supone la pérdida de vértices. Esta aplicación integra Reconstrucción Geométrica [3] con Interfaces Caligráficas (interfaces de usuario basadas en esbozos y gestos). Algunos sistemas desarrollados dentro del contexto de las interfaces caligráficas son GIDeS [4] y CALI [5].

## **2. Descripción de la aplicación prototipo**

La aplicación prototipo existente genera automáticamente el objeto correspondiente, actualizando el modelo 3D reconstruido cada vez que se produce algún cambio en el esbozo de entrada (Figura 1), sin necesidad de información adicional por parte del usuario. Para ello, se ejecutan secuencialmente las siguientes etapas:

- Adquisición de Datos: etapa encargada de interactuar con el usuario y extraer las entidades necesarias del tipo línea o comando borrar, desestimando aquella información que no sea necesaria.
- Reconstrucción Geométrica 2D: etapa encargada de transformar la información de la etapa de Adquisición de Datos, en una base de datos a partir de la cual reconstruir el modelo en la siguiente etapa.
- Reconstrucción Geométrica 3D: etapa encargada de obtener el modelo geométrico tridimensional a partir de la figura bidimensional generada en la etapa anterior.



*Figura 1. Interacción del usuario con la aplicación*

### **3. Reconstrucción Geométrica 2D**

El modelo 3D reconstruido se obtiene a partir de una estructura de datos adecuada, aplicando posteriormente los algoritmos desarrollados para la reconstrucción de normalones del grupo REGEO [1]. Con la finalidad de generar esta estructura de datos, se requiere un módulo de preproceso que realice las tareas necesarias de limpieza y ajuste de las entidades que forman el esbozo de entrada. Esta etapa la denominamos Reconstrucción Geométrica 2D.

Es en esta etapa donde se aplican técnicas de filtrado de los croquis o bocetos generados por el diseñador, adquiridos mediante la delineación con la interfaz caligráfica (tableta gráfica LCD), en la etapa de Adquisición de Datos. Se trata de convertir los bocetos en figuras geoméricamente consistentes, que sirvan a la etapa de Reconstrucción 3D para generar modelos tridimensionales.

La Reconstrucción Geométrica 2D no sólo debe eliminar las imperfecciones provocadas por los errores de la delineación por parte del usuario. Su tarea principal es filtrar todas las imperfecciones y errores que contienen los bocetos de partida, y que son consustanciales a su propia naturaleza de dibujos imprecisos e incompletos. El problema se puede ilustrar con el caso de las líneas que se “perciben” paralelas, aunque geoméricamente se constate que son convergentes; o las líneas cuyos extremos no definen un vértice, porque determinan puntos muy próximos pero no completamente coincidentes.

Por lo tanto, para cada una de las líneas del boceto introducidas por el usuario, es necesario modificar la inclinación de aquellas líneas que según criterios de la psicología de la percepción, deben ser paralelas a alguna de las aristas existentes del modelo.

Además, es necesario ajustar los puntos iniciales y finales, que definen cada una de las nuevas líneas, para que coincidan exactamente con los vértices que forman el modelo.

Con la finalidad de realizar estas tareas de un modo eficiente, se realiza una clasificación de las entidades que el usuario esboza. Para realizar esta clasificación de la nueva línea, se realiza el análisis de la misma, siguiendo los siguientes pasos:

1. Detección de si la nueva línea es paralela a alguna de las aristas existentes del modelo, teniendo en cuenta una tolerancia de paralelismo. En el caso de que la recta sea paralela, es necesario redefinir el punto inicial y el punto final.
2. Búsqueda de vértices existentes próximos a los puntos de definición de la nueva línea, teniendo en cuenta una tolerancia de proximidad. En el caso de que alguno de los puntos tenga más de un vértice próximo, se toma como Vértice Próximo de éste aquel cuya distancia entre el punto y el vértice sea mínima.
3. Búsqueda en aquellos puntos de definición de la nueva línea, que exista Vértice Próximo, de aristas existentes que tienen a este Vértice Próximo como uno de sus puntos de definición.
4. En aquellos puntos de definición de la nueva línea, que no exista Vértice Próximo, se analiza si están próximos a alguna arista existente, teniendo en cuenta una tolerancia de proximidad. En el caso de que alguno de los puntos de definición de la nueva arista, tenga más de una arista próxima, se toma como Arista Próxima aquella cuya distancia entre el punto y la arista sea mínima.

Todo el trabajo de clasificación previo, está en función de unos valores de tolerancia. Estos valores varían en función de la longitud de la nueva línea. Las tolerancias utilizadas en la aplicación son las siguientes:

- Tolerancia de paralelismo de líneas: valor que define la desviación máxima entre las pendientes de dos rectas para considerarlas paralelas.
- Tolerancia de proximidad de vértices: valor que define la distancia máxima entre los puntos de definición de la nueva línea y los vértices existentes, para considerarlos próximos. Este valor se utiliza para buscar el Vértice Próximo.
- Tolerancia de proximidad de aristas: valor que define la distancia máxima entre los puntos de definición de la nueva línea y las aristas existentes, para considerarlos próximos. Este valor se utiliza para buscar la Arista Próxima.

De este modo la nueva línea queda clasificada en función del valor que toman los siguientes parámetros.

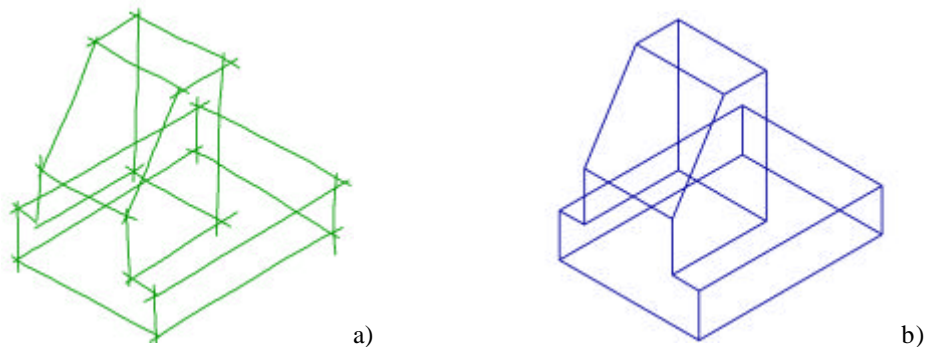
Parámetros booleanos:

- Línea paralela a alguna de las aristas existentes.
- Vértice Próximo en el punto de definición inicial.
- Vértice Próximo en el punto de definición final.
- Arista Próxima en el punto de definición inicial.
- Arista Próxima en el punto de definición final.

Parámetros numéricos:

- Número de aristas en el Vértice Próximo del punto de definición inicial.
- Número de aristas en el Vértice Próximo del punto de definición final.

A partir de esta clasificación, la aplicación está en condiciones de realizar el ajuste necesario de los puntos de definición de la nueva línea y de los Vértices Próximos y Aristas Próximas implicadas, para poder incorporarla a la base de datos de entrada de la etapa de Reconstrucción 3D. En la Figura 2 se puede apreciar como quedan las entidades del boceto del objeto modelado, tras la etapa de Reconstrucción 2D.



**Figura 2.** Boceto (a) Reconstrucción 2D (b)

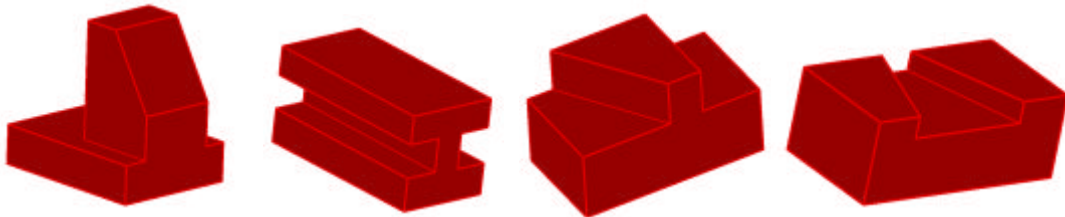
Toda la información referente a la imagen bidimensional tras la etapa de Reconstrucción 2D, se almacena en una base de datos. La estructura de esta base de datos es la siguiente:

- Lista de Vértices: lista de entidades del tipo vértice. Cada una de estas entidades contiene las coordenadas 2D (x,y) de los vértices que forman el modelo tras la reconstrucción 2D.

- Lista de Aristas: lista de entidades del tipo arista. Cada una de estas entidades contiene dos referencias a entidades de la Lista de Vértices. Estas referencias definen el punto inicial y final de la línea.

#### 4. Ejemplos

En la Figura 3 se presentan algunos ejemplos de modelos obtenidos con la aplicación.



*Figura 3. Modelos 3D obtenidos con la aplicación prototipo*

#### 5. Conclusiones

Partiendo de la experiencia adquirida por el grupo de investigación REGEO durante los últimos años, en lo referente al campo de la reconstrucción geométrica, se ha desarrollado una aplicación prototipo capaz de modelar objetos del tipo normalón y cuasi-normalón, a partir de un boceto generado a mano alzada. Esta aplicación combina una interfaz caligráfica con un algoritmo de reconstrucción geométrica. Debido a que el proceso de reconstrucción requiere un dibujo perfecto, en el que las primitivas del tipo línea inciden en vértices correctamente definidos, es necesario una etapa de Reconstrucción 2D que ajuste las entidades esbozadas por el usuario.

#### 6. Referencias

1. J. Conesa, P. Company, J.M. Gomis, *11th ADM International Conf. of Design Tools and Methods in Industrial Engineering*, Initial Modeling Strategies for Geometrical Reconstruction Optimization-Based Approaches, Palermo, (1999).
2. D.G. Ullman, S. Wood, D. Craig, *Computer and Graphics*, 14(2) (1990) 263.
3. H. Lipson, M. Shpitalni, *Computer Aided Design*, 28(8) (1996) 651.
4. J. Pereira, J. Jorge, V. Branco, F. Nunes, *WSCG 2000*, Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons, Plzen, (2000).
5. M. Fonseca, J. Jorge, *Pattern Recognition Letters*, 22(12) (2001) 1311.