

# **INTERFAZ GESTUAL PARA LA DEFINICIÓN DE CONDICIONES DE ENSAMBLAJE PARA LA GENERACIÓN DE MAQUETAS DIGITALES**

**José Luis Saorín Pérez (p), Manuel Contero González, Ferran Naya Sanchís y Julián Conesa Pastor.**

## Resumen

En el presente trabajo, se presenta un prototipo experimental denominado GEGROSS (**GE**sture & **Geometric Rec**Onstruction based **Sketch System**) que pretende facilitar al máximo el proceso de ensamblaje de las piezas necesarias para crear una maqueta digital a través de la codificación mediante un lenguaje de gestos gráficos de las condiciones de ensamblaje. Para ello los elementos utilizados han sido por una parte el motor geométrico ACIS, el gestor de restricciones de ensamblaje 3D DCM de la firma D-Cubed, y la biblioteca CALI para la definición de interfaces gestuales. En el artículo se presenta la estrategia seguida para la integración de estas complejas herramientas, y el alfabeto de gestos desarrollado para las diferentes condiciones de ensamblaje.

Palabras clave: interfaz caligráfica, bocetos, modelado de conjuntos.

## Abstract

In this paper we present an experimental prototype called GEGROSS (**GE**sture & **eometric Reconstruction based Sketch System**), that pretends to facility to the maximum the process of assembly in order to create a digital mock up, using the definition and codification of graphical gestures of assembly condition. The elements used to make this possible have been: the geometric kernel ACIS, the constraint manager 3D DCM from D-Cubed firm and the CALI library for the definition of gestural interfaces. In this paper we present the strategy for the integration of these complex tools, and the gestural alphabet developed for the different assembly conditions.

Key words: assembly modeling, gestures, sketching, calligraphic interface.

## Correspondencia

Universidad de La Laguna

Angel Guimerá Jorge s/n

38204 La Laguna

Tenerife

Tlf: 922 319448 Fax: 922 319870 E-mail: jlsaorin@ull.es

# **INTERFAZ GESTUAL PARA LA DEFINICIÓN DE CONDICIONES DE ENSAMBLAJE PARA LA GENERACIÓN DE MAQUETAS DIGITALES**

## **1. INTRODUCCIÓN.**

La utilización de maquetas digitales (digital mock-up) cada vez cobra más importancia en un entorno de desarrollo de producto basado en el empleo masivo de herramientas digitales de cálculo, análisis y simulación. En la actualidad, la aparición de nuevas clases de hardware, como son los Tablet-PC, y el desarrollo del concepto de interfaz caligráfica abre enormes posibilidades, para la creación de nuevas aplicaciones de diseño asistido por ordenador que ofrezcan un entorno más amigable de trabajo. Podemos definir interfaz caligráfica como aquel tipo de interfaz de usuario que permite la interacción entre un usuario y una aplicación informática mediante la utilización de un lápiz electrónico utilizando el dibujo a mano alzada, el texto manuscrito y símbolos o gestos gráficos como vehículos de transmisión de información.

En este contexto, en el presente trabajo, se presenta un prototipo experimental denominado GEGROSS (**GE**sture & **Geometric RecOn**struction based **Sketch System**) que pretende facilitar al máximo el proceso de ensamblaje de las piezas necesarias para crear una maqueta digital a través de la codificación mediante un lenguaje de gestos gráficos de las condiciones de ensamblaje que podemos encontrar actualmente en los sistemas CAD paramétricos/variacionales presentes en el mercado.

Para ello los elementos utilizados han sido por una parte el motor geométrico ACIS, que permite tanto recuperar la geometría de muchos sistemas CAD comerciales, como la geometría generada con nuestro propio sistema CIGRO de “modelado basado en sketching”. La utilización de ACIS permite la creación y gestión de geometrías complejas permitiendo por tanto construir sólidos de forma variada. Para dotar de capacidades paramétricas a los ensamblajes hemos utilizado el gestor de geometría paramétrica 3D DCM de la firma D-Cubed, que es la aplicación líder a nivel mundial en cuanto a la gestión de restricciones paramétricas.

En el artículo se presenta la estrategia seguida para la integración de estas dos complejas herramientas, así como el entorno gráfico de trabajo desarrollado en OpenGL y el sistema de definición de las condiciones de ensamblaje basadas en gestos.

## **2. ANTECEDENTES.**

El desarrollo de nuevos productos es una actividad esencial para la supervivencia y competitividad de la empresa. Existen diferentes estrategias de mejora del proceso de desarrollo de nuevos productos, pero la mayor parte ellas pasan por potenciar el papel del diseño y la disminución de la duración del ciclo de desarrollo del producto. De esta forma se mejora la flexibilidad de la empresa para adaptarse a las diferentes necesidades del

mercado. Por ello el disponer de herramientas de diseño adaptadas a las necesidades específicas de cada sector industrial es de especial interés

En este contexto el desarrollo de las denominadas “interfaces caligráficas” puede ser de gran interés a la hora de desarrollar herramientas innovadoras de asistencia al diseñador. El concepto de “interfaz caligráfico” es bastante reciente, y como su nombre indica se basa en la interacción del usuario a través de un lápiz electrónico

El progresivo abaratamiento de las tabletas gráficas LCD y el lanzamiento a finales de 2002 de los dispositivos denominados Tablet PC, que combinan las características de un ordenador portátil ultraligero con las de una tableta gráfica LCD, hacen viable, junto con los microprocesadores de elevada potencia de cálculo disponibles en la actualidad, el diseño de nuevos paradigmas de interacción, que permitan desarrollar aplicaciones más amigables, y adaptadas al mecanismo de expresión preferido por los diseñadores: el dibujo a mano alzada. [Jenkins, 1993], [Ullman, 1990].

La mayor parte de las aplicaciones de modelado desarrolladas bajo el concepto de “interfaz caligráfica” se basan en la utilización de determinados “gestos” o “signos”, que realizados por el usuario, son interpretados como comandos de la aplicación [Rubine, 1992]. Además existen sistemas que soportan el reconocimiento de texto manuscrito y la manipulación de dibujos realizados a mano alzada.

Dentro del campo de las interfaces caligráficas, existe una línea de investigación relativa a la utilización de dibujos o bocetos realizados a mano alzada como vía para obtener modelos geométricos 3D. Dentro de esta línea podríamos distinguir dos variantes principales. Una de ellas está basada en la interacción con el usuario mediante gestos que son reconocidos como comando generadores de sólidos a partir de secciones 2D. Como ejemplos de esta línea que denominamos “interfaz gestual” tenemos:

- SKETCH [Zelevnik, 1996]. Este sistema está orientado básicamente a formas arquitectónicas, donde el proceso de modelado de la geometría queda definido a través de un código de gestos y convencionalismos basados en el dibujo de simples líneas. El gesto más básico consiste en trazar tres líneas, alineadas con los tres ejes coordenados del dibujo, que inciden en un punto. Basado en este sistema SKETCH-N-MAKE [Bloomenthal, 1998] pretende facilitar la tarea de mecanizado por control numérico de piezas sencillas modeladas mediante una interfaz de tipo gestual.
- GIDeS [Pereira, 2000]. Este sistema permite la entrada a partir de una única proyección en perspectiva para construir diseños precisos a partir de dibujos ambiguos. El sistema dispone de un alfabeto de gestos que identifican un conjunto básico de primitivas de modelado. Además el reconocimiento dinámico de estos gestos de modelado permite ofrecer al usuario una serie de iconos contextuales, que le permiten confirmar su intención de diseño.
- QUICK-SKETCH [Eggle, 1997]. Esta herramienta está más orientada al diseño mecánico. Dispone de un entorno de dibujo 2D basado en restricciones, lo que permite obtener formas exactas y relaciones geométricas a partir del boceto

realizado sobre una tableta LCD. Permite también generar geometría 3D mediante la interpretación de una serie de gestos de modelado.

- TEDDY [Igarashi, 1999]. Este sistema permite modelar mediante una interfaz muy sencilla superficies tridimensionales de forma libre. El procedimiento consiste en trazar un boceto que represente la silueta del objeto, y el sistema propone automáticamente una superficie a través de una malla poligonal que encaje en esa silueta.

En cuanto a la segunda línea que podríamos denominar de “reconstrucción geométrica”, usa técnicas basadas en la visión artificial para construir formas geométricas tridimensionales a partir de representaciones bidimensionales. Se puede definir como el estudio de la información gráfica tridimensional que puede ser extraída de una representación bidimensional. Es decir, la Reconstrucción Geométrica es el nombre que normalmente se le da a la tarea de extraer la información geométrica de modelos 3D contenida en sus representaciones 2D. Para resolver este problema básicamente se han seguido dos líneas. La primera basada en el algoritmo de etiquetado de Huffman-Clowes [Huffman, 1971] [Clowes, 1971], y la segunda en la formulación del problema de reconstrucción como un problema de optimización [Wang, 1993].

La formulación como problema de optimización permite conseguir lo que desde el punto de vista de la geometría es inalcanzable: un modelo tridimensional a partir de una única proyección axonométrica, ya que esta proyección puede corresponder con la proyección de infinidad de modelos tridimensionales. Sin embargo, desde el punto de vista psicológico es también bien sabido el hecho de que los humanos no parecen tener problemas para identificar modelos 3D en imágenes 2D. Lo que es más, parece existir un gran consenso sobre cual es el modelo “correcto” y “simple” que todos los humanos ven en cada dibujo. Es por ello que la Reconstrucción Geométrica, entendida como un problema de percepción, puede ser descrita en términos de un problema de optimización matemática. Los métodos perceptuales se distinguen del resto de los métodos por intentar implementar, a través del lenguaje secuencial de los ordenadores, la forma en que percibe el ser humano. Esto es así, dada la posibilidad de establecer un símil entre los procesos iterativos característicos de los procesos de optimización y la manera de operar de la percepción humana. Algunos motores de reconstrucción han sido desarrollados por autores como Marill, Leclerc, Fischler, Lipson y Shpitalni [Marill, 1991] [Leclerc, 1992] [Lipson, 1996]. Ejemplos de este tipo de sistemas son:

- DIGITAL CLAY [Schweikardt, 2000]. Este sistema soporta objetos de tipo poliédrico, y combina por una parte una interfaz de tipo caligráfico para la entrada de información, que posteriormente es preprocesada y pasa a un motor de reconstrucción que emplea el algoritmo de Huffman-Clowes para realizar la reconstrucción de la geometría y su posterior exportación en un formato VRML. El uso de este algoritmo requiere que la forma tridimensional esté formada sólo por sólidos, y no permite reconstruir superficies sin grosor.
- Stilton [Turner, 2000], aunque está orientado al campo de la arquitectura, presenta aportaciones interesantes. En primer lugar la interfaz caligráfica se implementa

directamente en un entorno VRML. En segundo lugar el proceso de reconstrucción utiliza la formulación de optimización recurriendo en este caso a algoritmos genéticos para su resolución

- CIGRO [Contero, 2003] permite la generación de modelos poliédricos a partir de un boceto que corresponda a una axonometría ortogonal. Opera en tiempo real, permitiendo el cambio de punto de vista en cualquier momento del proceso de creación del boceto. Otra característica interesante es la simplicidad de su interfaz de usuario, que se ha reducido a un conjunto mínimo de gestos de modelado.

### 3. ARQUITECTURA DEL SISTEMA.

El prototipo que se presenta en este trabajo extiende las capacidades de nuestro anterior sistema CIGRO [Contero, 2003], para posibilitar la generación de conjuntos a partir de un alfabeto de gestos de ensamblaje. La arquitectura de dicho sistema se presenta en la Figura 1. Los principales elementos que lo constituyen son por una parte la biblioteca CALI [Fonseca, 2001] que gestiona la identificación de los gestos de diseño. El gestor de restricciones dimensionales DCM de la firma D-Cubed [D-Cubed, 2003] proporciona las herramientas necesarias para la definición de las condiciones paramétricas de ensamblaje.

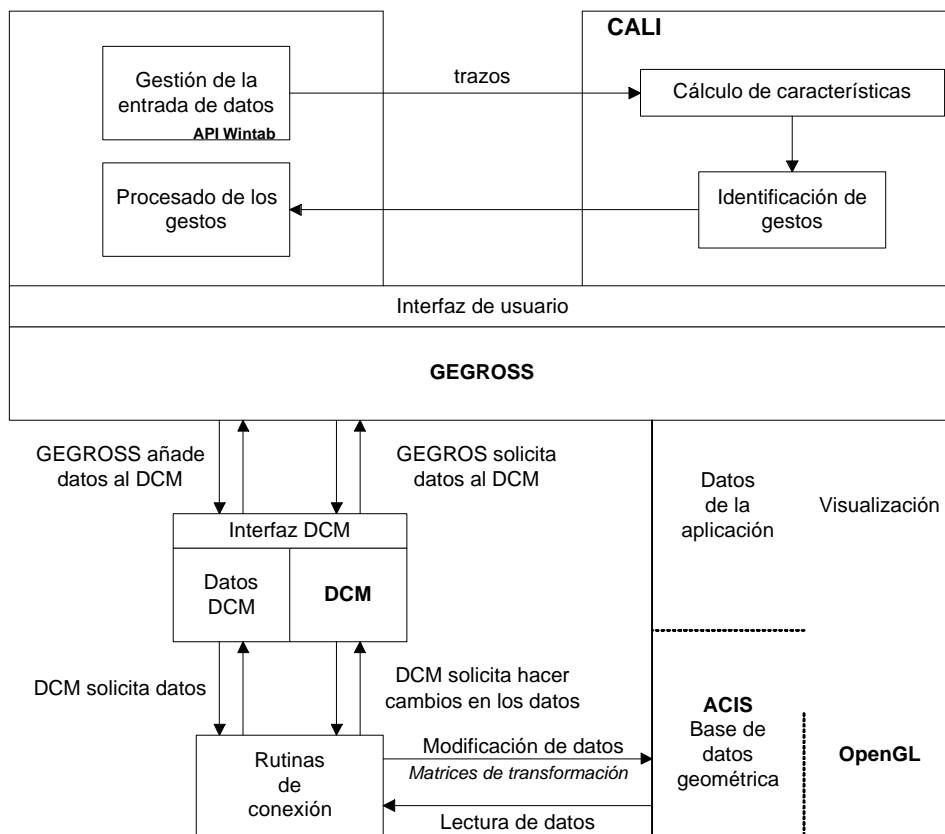


Figura 1. Arquitectura del sistema GEGROSS.

El último componente relevante es el modelador ACIS [Spatial, 2003] que proporciona los recursos necesarios para gestionar la base de datos geométrica, facilitando además la utilización de piezas modeladas con nuestro sistema CIGRO, o con cualquier sistema del mercado construido sobre esta plataforma, como los desarrollados por la firma Autodesk.

Uno de los aspectos críticos del presente trabajo ha sido la labor de integración de todas estas herramientas, que ya de forma aislada tienen una elevada complejidad de utilización. En el caso del sistema del 3D DCM (implementado en C++), un aspecto importante es que sólo mantiene de forma interna la información asociada a la conectividad del esquema de acotación empleado, siendo responsabilidad de nuestra aplicación la gestión de la base de datos geométrica (en este caso utilizamos el núcleo geométrico ACIS) y el esquema de acotación. El intercambio de información entre DCM y nuestra aplicación se realiza a través de unas “rutinas de conexión” especiales. Básicamente todo el proceso de intercambio se realiza a través de un sistema de punteros algo complejo.

#### 4. CODIFICACIÓN DE LAS CONDICIONES DE ENSAMBLAJE. GESTOS.

El ensamblaje de distintas piezas en un espacio tridimensional exige información de varios tipos. Por un lado es necesario conocer los elementos que se van a ensamblar y por otro las condiciones geométricas particulares que definen su relación en el espacio. Es por ello que para codificar una condición de ensamblaje hay que realizar varias fases. Primero de todo será necesario seleccionar los elementos de cada cuerpo sobre los que vamos a trabajar ya que podemos elegir entre las caras, aristas o vértices de cada uno de ellos.

Una vez hecha esta selección hay que indicar el movimiento que queremos realizar sobre ella. Estos movimientos, están asociados a los conceptos de apoyar, alinear, apoyar con desplazamiento, alinear con desplazamiento, insertar, orientar o tangencia.

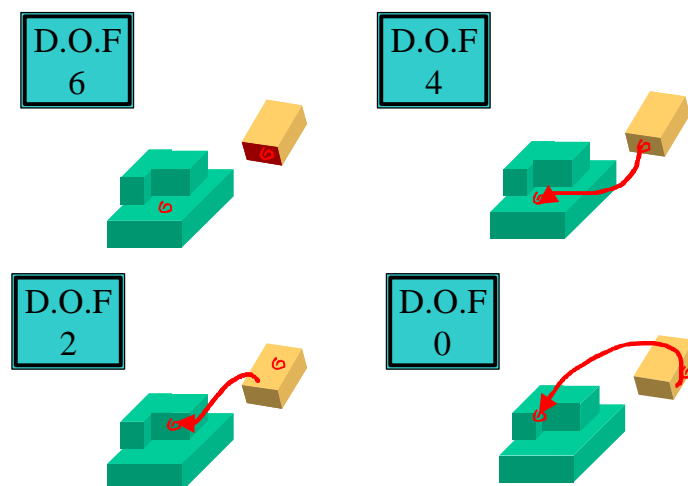


Figura 2. Definición gestual de las condiciones de ensamblaje.

El problema es que la operación de ensamblaje no se acaba con estos primeros gestos. Una vez finalizada esta etapa, tendremos definida una cierta restricción sobre los dos sólidos, pero habitualmente no habrán quedado restringidos todos los grados de libertad posibles

(D.O.F.). Esto complica las cosas, ya que la operación de ensamblaje exigirá por lo tanto varios gestos para tener definida por completo el movimiento. Por ello, existen dos caminos posibles para completar la operación con éxito. El primero consiste en que el programa se sitúe automáticamente en modo “ensamblaje” y aparezca alguna ventana adicional que suministre al usuario información sobre los grados de libertad que quedan sin restringir. De esta manera el usuario sabría que tiene que seguir indicando condiciones de ensamblaje hasta que esté perfectamente definido el movimiento (ver figura 2). El segundo método consiste en que el programa de manera inteligente ejecute las órdenes de ensamblaje, que considera más probable de manera que la pieza quede con unos ciertos grados de libertad pero en su nueva ubicación y sea el usuario mediante algún otro gesto el que ratifique la posición o la complete con un nuevo movimiento simple.

Un ejemplo de estas dos maneras de trabajar lo tenemos cuando queremos alinear dos piezas cilíndricas. Suponiendo que el usuario marque los ejes de las piezas como elementos geométricos a restringir y después haga el gesto de alinear nos encontramos con dos caminos posibles. En el primero de ellos el sistema ofrece una ventana con los grados de libertad no restringidos, donde indica al usuario que tiene que seguir realizando gestos de ensamblaje hasta ese número alcance el valor de seis. Para ello el usuario deberá indicar si las caras apoyan una sobre otra y el ángulo en el que se posicionan los cilindros. De esta manera con dos nuevos gestos se terminaría la orden de ensamblaje y el sistema la ejecutaría. En el segundo método, el sistema asumiría que por ejemplo las dos piezas además de tener los ejes alineados deberán tener una cara común y ángulo de giro nulo. Con esta suposición el sistema ejecutará el movimiento, pero dejará la pieza “libre” en los grados de libertad no indicados por el usuario para que este pueda modificar alguno de los parámetros elegidos por el sistema. De esta manera podría indicarse por ejemplo que la distancia entre las caras es diferente de cero o que la pieza tiene un cierto giro con respecto al eje común.

## 5. ALFABETO DE GESTOS.

Como hemos visto en el punto anterior, la orden de ensamblaje se compone elementos formados por la geometría elegida y el gesto propiamente dicho. Por ello cada operación de ensamblaje se compondrá de una suma de gestos básicos:

1.- *Selección de geometría.* El sistema debe permitir seleccionar la geometría a restringir independientemente de si estos elementos son vistos o no. Este gesto lo codificamos como una espiral o un punto sin cerrar que quedaría asociado a ese elemento.

2.- *Gesto básico de ensamblaje.* Es el gesto mínimo, donde el sistema interpreta la intención del diseñador y “decide” automáticamente los parámetros para conseguir restringir los grados de libertad del sistema y proponer una solución lógica. Este gesto corresponde a una flecha.

3.- *Gesto compuesto.* Se representa como una flecha con un espacio circular a la mitad donde se incluye un modificador específico de significado, a través de un gesto complementario. De esa manera es posible indicarle explícitamente al sistema, una condición específica de ensamblaje.



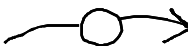



OPERACIÓN		GESTO
Gesto básico de seleccionar geometría		
Gesto básico de ensamblaje (ensamblaje automático por el sistema)		
Gesto compuesto (Necesita gestos complementarios)		
Gestos complementarios de definición		
	Paralelismo	
	Perpendicularidad	
	Tangencia	

Tabla 1. Gestos de ensamblaje.

Como podemos ver en la Tabla 1, incluimos en una serie mínima de gestos que permitan realizar algunas operaciones de ensamblaje. En un futuro se ampliará la lista de gestos complementarios para poder cubrir todas las situaciones que se presentan en la realidad industrial del proceso de ensamblaje de un producto. Para ello, está previsto utilizar en la medida de lo posible convencionalismo gráficos como los que se utilizan en la representación de las tolerancias geométricas.

## 6. INTERCONEXIÓN ACIS – 3D DCM

Como hemos visto la aplicación prototipo de ensamblaje por gestos se apoya en un motor geométrico (ACIS) y un gestor de restricciones (3D DCM), debido a ello tendremos que resolver los problemas asociados a la conectividad de estos dos sistemas.

En ACIS, cada cuerpo tiene asociada una matriz de transformación que permite aplicarles giros o traslaciones. Esta filosofía coincide con la del 3D DCM, ya que después de solucionar el sistema de restricciones de ensamblaje, se asocia a cada elemento de la geometría una matriz que indica la transformación que tendría que tener el cuerpo para cumplir con las ecuaciones. El único problema por lo tanto es conseguir que las transformaciones que se aplican en los cuerpos sean las mismas.

3D DCM devuelve una matriz de giro 4x4 que está formada por la matriz de rotación y la de traslación conjunta. Se puede aplicar directamente a las coordenadas homogéneas de la geometría implicada y de esta forma se obtiene la posición solución del sistema de ecuaciones que representan las restricciones.

$$\begin{bmatrix} R(0,0) & R(0,1) & R(0,2) & dx \\ R(1,0) & R(1,1) & R(1,2) & dy \\ R(2,0) & R(2,1) & R(2,2) & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} coord\_x \\ coord\_y \\ coord\_z \\ 1 \end{bmatrix} = \begin{bmatrix} nueva\_coord\_x \\ nueva\_coord\_y \\ nueva\_coord\_z \\ 1 \end{bmatrix} \quad (1)$$



ACIS no tiene ningún problema para aplicar las traslaciones sobre un cuerpo. Para ello se selecciona la columna correspondiente en 3D-DCM a la traslación y se aplica directamente. El problema surge con los giros ya que ACIS no permite utilizar la matriz de transformación, sino las coordenadas del vector alrededor del cual gira el cuerpo y el ángulo de giro. Por ello hay que lograr descomponer la matriz obtenida en 3D-DCM en una sucesión de giros a lo largo de los tres ejes principales. Para ello hay que resolver un sistema de ecuaciones cuyas incógnitas son los ángulos de giro en el eje X (A), eje Y (B) y eje Z (C).

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & \sin A \\ 0 & -\sin A & \cos A \end{pmatrix} \times \begin{bmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{bmatrix} \times \begin{bmatrix} \cos C & \sin C & 0 \\ -\sin C & \cos C & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R(0,0) & R(0,1) & R(0,2) \\ R(1,0) & R(1,1) & R(1,2) \\ R(2,0) & R(2,1) & R(2,2) \end{bmatrix}$$

Una vez obtenidos los ángulos tenemos resuelta la comunicación entre nuestra aplicación y el gestor de restricciones 3D DCM.

## 6. CONCLUSIONES.

En el presente trabajo hemos presentado los resultados preliminares del prototipo GEGROSS, que permiten validar la viabilidad de desarrollar un módulo completo de gestión de condiciones de ensamblaje a través de una codificación gestual de las mismas. Aunque nos encontramos en una fase preliminar de desarrollo, las experiencias realizadas con algunos usuarios que anteriormente no habían tenido experiencia con un sistema de modelado basado en bocetos son muy prometedoras, destacando la rapidez de aprendizaje del mismo, y su carácter intuitivo. Hasta el momento los principales problemas en el desarrollo del sistema, han sido causados por la complejidad de las herramientas utilizadas, pero la arquitectura desarrollada, consideramos que es lo suficientemente flexible, para experimentar con diferentes códigos gestuales de ensamblaje en un futuro.

## AGRADECIMIENTOS.

Este trabajo ha sido financiado por la Universidad de La Laguna a través del “Programa de Ayudas a la Investigación para la Formación y Promoción del Profesorado. Ayudas para Estancias de Investigadores Invitados” y por la Generalidad Valenciana, a través del proyecto de referencia CTIDIB/2002/51 de la convocatoria 2002 de Proyectos de I+D.

## REFERENCIAS.

Bloomenthal, K.; Zeleznik, R.C. et al. *SKETCH-N-MAKE. Automated Machining of CAD Sketches*. Proceedings of ASME DETC'98, 1998, pp 1-11.

Clowes, M.B. *On Seeing Things*. Artificial Intelligence, 2, 1971, pp. 79-116.

- Contero, M.; Naya, F.; Jorge, J. and Conesa, J. *CIGRO: A Minimal Instruction Set Calligraphic Interface for Sketch-Based Modeling*. Lecture Notes in Computer Science Vol. 2669, 2003, pp 549-558.
- D-Cubed, *3D Dimensional Constraint Manager*, <http://www.d-cubed.co.uk>, 2003
- Eggl, L.; Hsu, C.; Brüderlin, B.D.; Elber, G. *Inferring 3D Models from Freehand Sketches and Constraints*. Computer-Aided Design, 29 (2), 1997, pp 101-112.
- Fonseca, M.; Jorge, J. *Experimental Evaluation of an On-Line Scribble Recognizer*. Pattern Recognition Letters, 22 (12), 2001, pp 1311-1319.
- Huffman, D.A. *Impossible objects as nonsense sentences*. In Meltzer B., Michie D. eds. Machine Intelligence No 6, Edimburgo UK. Edinburgh University Press, 1971, pp 295-323.
- Igarashi, T.; Matsuoka, S.; Tanaka, H. *Teddy: A Sketching Interface for 3D Freeform Design*. ACM SIGGRAPH 99 Conference Proceedings, 1999, pp 409-416.
- Jenkins, D.L.; Martin, R.R. *The importance of free-hand sketching in conceptual design: automatic sketch input*. Design Theory & Methodology (DTM 93), Hight T.K. y Stauffer L.A. Eds., ASME Vol. DE-53, 1993, pp 115-128
- Leclerc, Y.; Fischler, M. *An Optimization-Based Approach to the Interpretation of Single Line Drawing as 3D Wire Frames*. Int. Journal of Comp. Vision, 9 (2), 1992, pp 113-136.
- Lipson, H., Shpitalni, M. *Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing*. Computer Aided Design, 28 (8), 1996, pp 651-663.
- Marill, T. *Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects*. International Journal of Computer Vision, 6 (2), 1991, pp 147-161.
- Pereira, J.; Jorge, J.; Branco, V.; Nunes, F. *Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons*. WSCG'2000. Conference Proceedings, Skala V. Ed., 2000.
- Rubine, D. *Combining gestures and direct manipulation*. Proceedings ACM CHI'92 Conference Human Factors in Computing Systems, 1992, pp 659-660.
- Schweikardt, E.; Gross, M.D. *Digital Clay: deriving digital models from freehand sketches*. Automation in Construction, Vol. 9, 2000, pp 107-115.
- Spatial Corp. *3D ACIS Modeller*, <http://www.spatial.com/products/3D/modeling>, 2003
- Turner, A.; Chapman, D.; Penn, A. *Sketching space*. Computers and Graphics, 24 (6), 2000, pp 869-879.
- Ullman, D.G.; Wood, S.; Craig, D. *The importance of drawing in the mechanical design process*. Computers and Graphics, 14 (2), 1990, pp 263-274.
- Wang, W.; Grinstein, G. *A Survey of 3D Solid Reconstruction from 2D Projection Line Drawing*. Computer Graphics Forum, 12 (2), 1993, pp 137-158.
- Zelevnik, R.C.; Herndon, K.P.; Hughes, J.F. *SKETCH: An interface for sketching 3D scenes*. SIGGRAPH'96 Conference Proceedings, 1996, pp 163-170.