

NAYA, Fernando

CONTERO, Manuel

GOMIS, José María

Universidad Politécnica de Valencia

E.T.S. Industriales

Parámetros de trazado en la ejecución de dibujos a mano alzada en el entorno de sistemas CAD

Resumen

En la actualidad, a pesar del considerable avance experimentado por las aplicaciones de diseño asistido por ordenador, el tipo de interfases gráficas de usuario (GUI) disponibles apenas ha evolucionado, más allá del paradigma conocido por las siglas WIMP (Window, Icon, Menu, Pointing device). El grupo de investigación REGEO (<http://www.tec.uji.es/regeo>) ha desarrollado en los últimos años diferentes algoritmos en el campo de la reconstrucción geométrica. El objetivo actual del grupo es conseguir una aplicación automática (o al menos, fácil de usar) para “reconstruir modelos geométricos”, es decir, para generar automáticamente modelos virtuales 3D, partiendo de dibujos a mano alzada en 2D. La Reconstrucción Geométrica es una pequeña parte del campo de la Visión Artificial, o Visión por Computador. En la Reconstrucción Geométrica la información de entrada son figuras geométricas en 2D y un conjunto de símbolos normalizados empleados en los planos de ingeniería, y la información de salida son modelos geométricos 3D como los utilizados por las aplicaciones CAD/CAM/CAE. La reconstrucción no es tan sólo un proceso numérico basado en reglas geométricas. La información geométrica contenida en las figuras suele ser incompleta. Razón por la que las figuras 2D deben ser “percibidas” por el ordenador (en el sentido psicológico del término “percepción”). Además, la posible existencia de símbolos normalizados (tales como condiciones geométricas y dimensiones) obliga a “interpretar” ese lenguaje e incorporar la información en el correspondiente modelo. En el artículo se estudia un prototipo de interfase de usuario basado en la utilización de una tableta gráfica, en la que el usuario directamente croquiza una pseudoperspectiva de un cuerpo, y tras el correspondiente preproceso, utilizando el motor de reconstrucción geométrica realizado pasar a reconstruir dicho objeto.

1. Nuevas interfases de usuario

A pesar del enorme desarrollo experimentado por los sistemas CAD en las últimas décadas, todavía 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 [1], [2], [3].

Aunque los primeros antecedentes de los sistemas CAD actuales [4] utilizaron como elementos de entrada de datos dispositivos del tipo lápiz óptico (light pen), en la actualidad el paradigma WIMP (Window, Icon, Menu, Pointing device) es el único disponible en los sistemas CAD comerciales. Sin embargo, desde hace años existen diferentes líneas de investigación conducentes a la búsqueda de esquemas de interacción hombre-máquina alternativos

de las que podemos encontrar los primeros antecedentes en los años 70 [5], [6]. Una de estas líneas es la utilización, como dispositivo de entrada, de un “lápiz” que se utiliza sobre una tableta gráfica o una tableta LCD. Sería el equivalente de un esquema “lápiz y papel” electrónicos, donde la comunicación se realiza por medio de bocetos. Se puede afirmar con seguridad, que en la actualidad, se dispone de procesadores con la suficiente potencia de cálculo y dispositivos de entrada y salida, como tabletas sensibles a la presión y tabletas LCD, para justificar la viabilidad de tales desarrollos. A pesar de fracasos anteriores, como “Windows for Pen Computing” en la primera mitad de los años 90, debido a las prestaciones de los sistemas de la época, en la actualidad multitud de pequeños asistentes personales (PDA) utilizan algún tipo de interfase basado en este tipo de interacción. Además, está surgiendo un nuevo tipo de dispositivo denominado “Tablet

PC”, del que ya existen ejemplos comerciales, que augura un gran desarrollo de este tipo sistemas en los próximos años. Prueba de ello, es la presentación durante la Feria COMDEX, en noviembre de 2000, del prototipo de “Tablet PC” que Microsoft piensa comercializar en la segunda mitad de 2002.

La mayor parte de las aplicaciones desarrolladas bajo el concepto de “pen computing” se basan en la utilización de determinados “gestos” o “signos”, que realizados por el usuario, son interpretados [7], [8] como comandos de la aplicación. Además existen sistemas que soportan el reconocimiento de texto manuscrito y la manipulación de dibujos realizados a mano alzada.

Dentro de este contexto, 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 basado en la interacción con el usuario mediante gestos que son reconocidos como comando generadores de sólidos a partir de secciones 2D. Una segunda alternativa, es el empleo de algoritmos, que a partir de un boceto que represente una proyección del objeto, procedan a su reconstrucción geométrica. Por último, existe una línea híbrida que combina ambos enfoques.

Como ejemplos de sistemas del primer tipo, y que denominaremos de “interfase gestual” de modelado tenemos:

- SKETCH [9] orientado básicamente a formas arquitectónicas, donde el proceso de modelado de la geometría queda definido a través de un código de gestos y convencionalismos. Por ejemplo, una primitiva tipo bloque queda definida a partir de tres trazos realizados desde el mismo punto. Los volúmenes positivos se construyen hacia arriba, mientras que los negativos (volumen a restar) se realizan hacia abajo. Basado en este sistema SKETCH-N-MAKE [10] pretende facilitar la tarea de mecanizado por control numérico de piezas sencillas modeladas mediante un interfase de tipo gestual.
- Quick-Sketch [11] es una herramienta más orientada al diseño mecánico. Dispone de un entorno de dibujo 2D basado en restricciones. Permite también generar geometría 3D mediante la interpretación de una serie de gestos de modelado.
- Teddy [12], permite modelar mediante un interfase muy sencillo superficies 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. Un aspecto interesante es su implementación en Java y su ejecución a través de un simple navegador de Internet.

- GIDes [13] tiene la particularidad que permite la entrada a partir de una única proyección en perspectiva o a partir de varias vistas diédricas. En el caso de la reconstrucción a partir de una única proyección, el sistema dispone de un alfabeto de gestos que identifican un conjunto básico de primitivas de modelado como son: prisma, pirámide, extrusión y revolución entre otras. Además el reconocimiento dinámico de estos gestos de modelado permite ofrecer al usuario una serie de iconos contextuales, que le permiten confirman su intención de diseño.

En cuanto a la segunda línea que podríamos denominar de “reconstrucción geométrica”, la podríamos definir como el estudio de la información gráfica tridimensional que puede ser extraída de una imagen bidimensional, proyección axonométrica del modelo. Para resolver este problema básicamente se han seguido dos líneas. La primera basada en el algoritmo de etiquetado de Huffman-Clowes [14], [15], [16], y la segunda en la formulación del problema de reconstrucción como un problema de optimización. Esta segunda aproximación permite conseguir lo que desde el punto de vista de la geometría es inalcanzable: un modelo tridimensional a partir de una única proyección axonométrica. 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. Los procesos de optimización para la reconstrucción geométrica presentan una característica fundamental que les hace diferir del resto de los problemas de optimización: los mínimos locales representan soluciones erróneas en tanto que ofrecen modelos tridimensionales no acordes con la tendencia visual humana, es decir, mientras en otros análisis de procesos de optimización, la consecución de mínimos locales podría ser considerada como válida en tanto que ofrece una reducción del coste de la función objetivo, dicho mínimo local no podría ser entendido como solución válida en la reconstrucción geométrica, en tanto que representaría modelos tridimensionales no acordes con nuestra percepción visual [17], [18], [19], [20].

Por último podríamos hablar de una vía que combina tanto el enfoque “gestual” como el enfoque de “recons-

trucción”, y que denominaríamos “híbrida”. El desarrollo de esta línea es bastante reciente como lo muestra la bibliografía disponible. Los sistemas más interesantes en este enfoque son:

- Digital Clay [21], [22] soporta objetos de tipo poliédrico, combina por una parte un interfase 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.
- Stilton [23], aunque está orientado al campo de la arquitectura, presenta aportaciones interesantes. En primer lugar el interfase caligráfico 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.

2. El proyecto CIGRO (Calligraphic Interce for Geometric ReconstrucciOn)

En este contexto, el grupo REGEO (<http://www.tec.uji.es/regeo>) se encuentra en la línea de obtener una aplicación que genere automáticamente modelos virtuales 3D en un formato estándar susceptible de ser importado en una aplicación CAD comercial, partiendo de dibujos a mano alzada en 2D. En la actualidad se dispone de un “reconstructor” geométrico [24], [25] razonablemente robusto que permite realizar el proceso de reconstrucción a partir de una perspectiva axonométrica. El reconstructor opera con un dibujo perfectamente delineado mediante un programa CAD 2D, en que la pseudoperspectiva de entrada al sistema queda definida con primitivas del tipo línea, que inciden en vértices correctamente definidos, exportándose en un formato DXF a nuestra aplicación de reconstrucción. Esta aplicación a la que hemos denominado REFER, implementa algoritmos

basados en la transformación del problema de reconstrucción en un problema de optimización, y proporciona un modelo B-rep del sólido reconstruido. En la actualidad se trabaja en la línea de integrar el reconstructor REFER en un entorno más amigable de trabajo, en la línea de lo que se denominan “interfases caligráficas”. Para ello, se pretende desarrollar una interfase que permita al usuario interactuar con la aplicación mediante una tableta gráfica o tableta LCD donde éste esbozará un croquis en pseudoperspectiva que posteriormente servirá de entrada al módulo de reconstrucción. Para ello se está desarrollando una biblioteca C++ que permita interactuar con tabletas y otros dispositivos gráficos similares mediante el API Wintab (<http://www.pointing.com>).

Como parte del desarrollo de la interfase caligráfica nos planteamos cómo explotar la información que la tableta gráfica proporciona a través del API Wintab, que es vehículo de comunicación estándar entre una aplicación y ese tipo de dispositivos.

3. El API Wintab y la interacción con tabletas gráficas

La necesidad de un interfase de programación estándar para las tabletas gráficas y equipos digitalizadores, era requerido tanto por los fabricantes de dispositivos como por los desarrolladores de aplicaciones. Era ineficiente requerir a los desarrolladores de aplicaciones que escribieran un controlador diferente para cada uno de los dispositivos con los que podían trabajar, y a los fabricantes de dispositivos que desarrollaran controladores para las distintas aplicaciones que usaban. Esta fue la principal razón para la creación del Interfase Wintab, ya que la existencia de controladores simplificaría el proceso de desarrollar aplicaciones Windows © por Microsoft Corporation.

Wintab es un interfase abierto que directamente recoge información de entrada de los equipos digitalizadores y la transmite de un modo estándar a las aplicaciones. Las

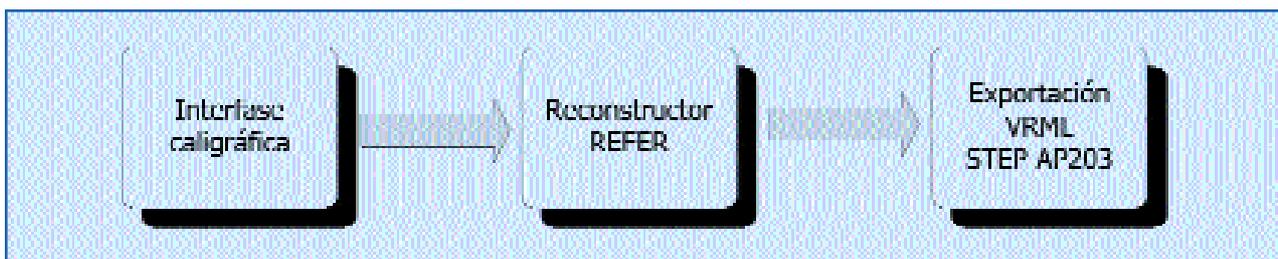


Figura 1. Estructura general de la aplicación.

ventajas son claras, se permite una fácil y consistente integración entre sofisticados equipos digitalizadores y las aplicaciones, y el usuario es capaz de controlar el equipo de una forma fácil y natural a través de simples llamadas a unas funciones ya existentes.

Para poder obtener los datos esbozados en una tarjeta gráfica, es necesario desarrollar una aplicación que se encargue de gestionar la solicitud y la recepción de los mismos. La tableta gráfica aporta información a través de mensajes (packets) enviados a la aplicación. Es esta última la encargada de filtrarlos y procesarlos adecuadamente. Existe la posibilidad de filtrar la información recibida en cada uno de los mensajes, lo que evita la transacción de información no útil para la aplicación.

Existen en la actualidad tabletas gráficas que soportan el API Wintab, con las que es posible obtener la siguiente información:

- La posición del trazador en el área activa de la tableta, a lo largo del eje X y del eje Y (figura 2).
- El estado de los botones del trazador.
- El tiempo en el que tiene lugar cada uno de los eventos recibidos.
- Los parámetros enviados que han sufrido cambios desde el último mensaje.
- El tipo de trazador que ha generado el mensaje.
- El valor de la presión ejercida por el trazador sobre el área activa de la tableta.
- La rotación dextrógira de la proyección ortogonal del trazador sobre el plano XY, respecto al eje Z del área activa de la tableta (ángulo de la figura 2). El valor obtenido en este caso oscila desde los 0° a los 360° .

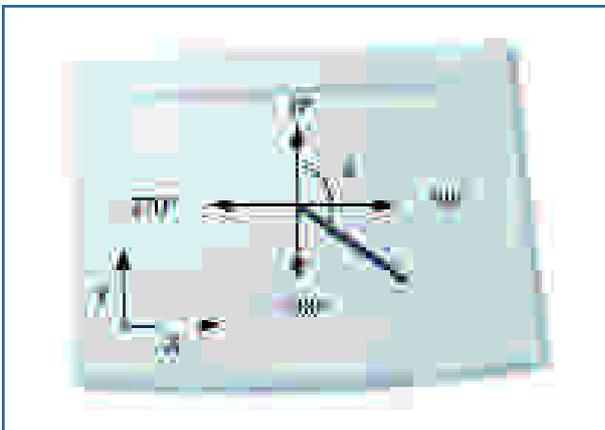


Figura 2. Ángulo del trazador y origen de coordenadas.

- El ángulo que forma el trazador con el plano XY del área activa de la tableta (ángulo de la figura 3). Los valores obtenidos en este caso están comprendidos entre 25° y 90° (perpendicular a la superficie de la tableta).
- Información estática de la tableta: dimensiones, rango de presiones, etc.

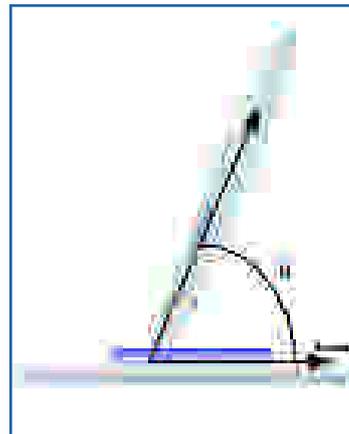


Figura 3. Ángulo del trazador.

4. Estudio experimental de los parámetros de trazados

Para la realización del presente trabajo se ha utilizado como equipo digitalizador, una tableta gráfica que soporta el API Wintab.

Como la finalidad de la Interfase Caligráfica es obtener un fichero de entrada para el módulo de reconstrucción, a partir de un croquis en pseudoperspectiva esbozado a mano alzada, se ha utilizado un trazador que permite visualizar sobre un papel lo que se está representando. Este trazador no tiene cables ni batería, y tiene en su extremo una mina de tinta. De este modo, si colocamos un papel sobre la tableta gráfica, parece que estemos esbozando de un modo real, pero al mismo tiempo los datos del trazado están siendo procesados. Es decir, para el usuario es como si estuviese realizando un boceto con lápiz y papel.

Los resultados que aparecen en la presente comunicación, han sido obtenidos a partir de un estudio realizado a 25 personas, entre las cuales existen tanto gente diestra co-

mo gente zurda. Cada uno de ellos ha esbozado varias circunferencias de distinto radio y varias líneas rectas con distinta inclinación (horizontales, verticales y oblicuas).

Por lo tanto la metodología seguida para la obtención de los parámetros de trazado de cada uno de los bocetos ha sido la siguiente:

1. Colocar un papel sobre la tableta gráfica.
2. Esbozar las líneas curvas y las líneas rectas.
3. Almacenar los parámetros de trazado, mientras se realiza el esbozo.
4. Procesar los parámetros obtenidos, para obtener el fichero de entrada para el módulo de reconstrucción.

La aplicación desarrollada en este caso se encarga de realizar los dos últimos pasos. Debido a que la tableta gráfica utilizada soporta el Interfase Wintab, el desarrollo de la aplicación para la obtención de sus datos se ha podido programar de un modo económico y rápido, usando la MFC de Visual C++.

El objetivo inicial del estudio realizado, es analizar aquellos parámetros atribuibles al trazado de una entidad simple (líneas curvas o líneas rectas) esbozada a mano alzada, que permitan diferenciar el orden de la misma (1 ó -1) y su tipología (línea continua, discontinua, etc.). Es decir, llegar a diferenciar si se traza una línea recta o una línea curva, y si es continua o discontinua, simplemente procesando los datos referentes al trazado de la misma.

Los parámetros elegidos para realizar el análisis han sido los siguientes:

- La Presión del trazador sobre el área activa de la tableta gráfica. Los valores de presión obtenidos oscilan entre 0 y 1.023 (presión máxima). Es decir, es posible obtener 1.024 niveles de presión.
- El ángulo que forma el trazador con el plano XY (ángulo α).
- El ángulo que forma el trazador con el plano YZ (ángulo β).
- La posición del trazador sobre el área activa de la tableta, a lo largo del eje X y del eje Y.

Los parámetros arriba indicados vienen representados de un modo gráfico en la figura 4. Los resultados obtenidos han sido procesados con la finalidad de encontrar posibles relaciones entre los valores de los parámetros obtenidos, y el orden y la tipología de cada una de las líneas esbozadas con el trazador sobre el papel.

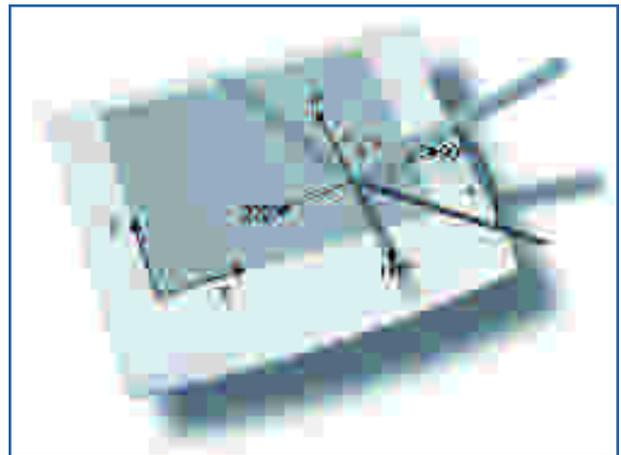


Figura 4. Parámetros de trazado utilizados.

Del estudio experimental realizado, es posible extraer las siguientes generalidades:

1. Existe en general, para casi todos los usuarios, un valor límite que separa los valores medios del ángulo α obtenidos para cada uno de los tramos rectos (líneas rectas), con relación a los valores medios obtenidos para cada uno de los tramos curvos (líneas curvas).

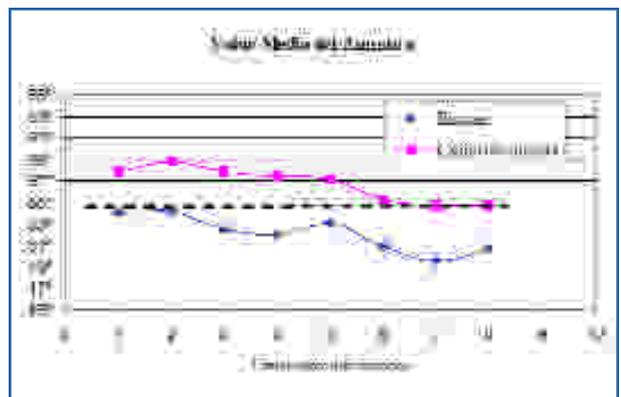


Figura 5. Límite de separación de los valores medios del ángulo α .

2. Del mismo modo, aparece un valor límite para cada usuario que separa las desviaciones típicas del ángulo α obtenidas para tramos rectos (líneas rectas), de las desviaciones típicas obtenidas para tramos curvos (líneas curvas).

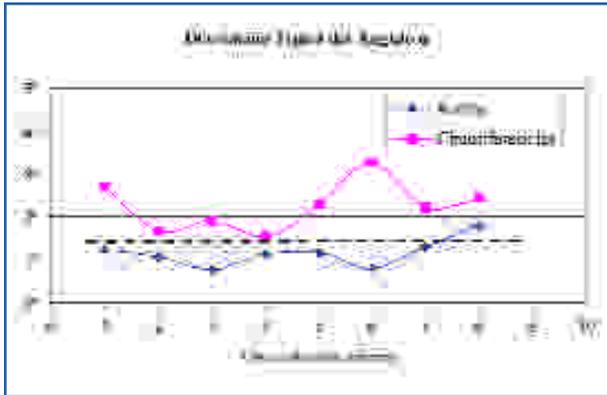


Figura 6. Límite de separación de las desviaciones típicas del ángulo.

3. El valor del ángulo del trazador al esbozar una recta, presenta un comportamiento lineal como se aprecia en la figura 7.

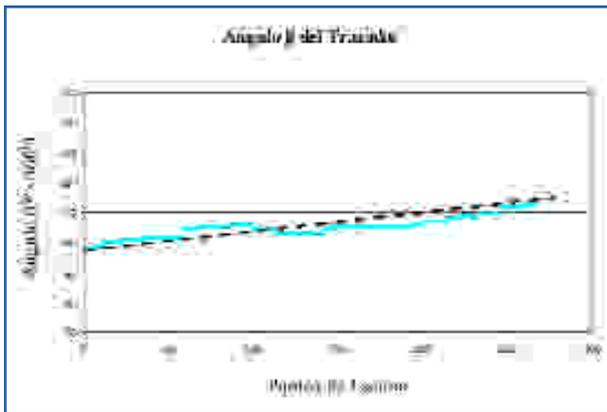


Figura 7. Ángulo del trazador al esbozar una línea recta.

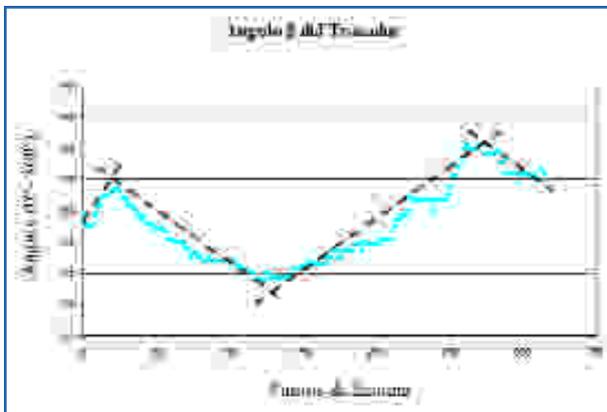


Figura 8. Ángulo del trazador al esbozar una línea curva.

4. El valor del ángulo del trazador al esbozar una línea curva, forma tramos ascendentes y descendentes como se aprecia en la figura 8. Por lo tanto, se aprecia una diferencia en cuanto a la evolución del valor del ángulo del trazador en función de si se esboza un tramo recto o un tramo curvo.
5. El valor del ángulo del trazador al esbozar una recta, presenta un comportamiento lineal como se aprecia en la figura 9.

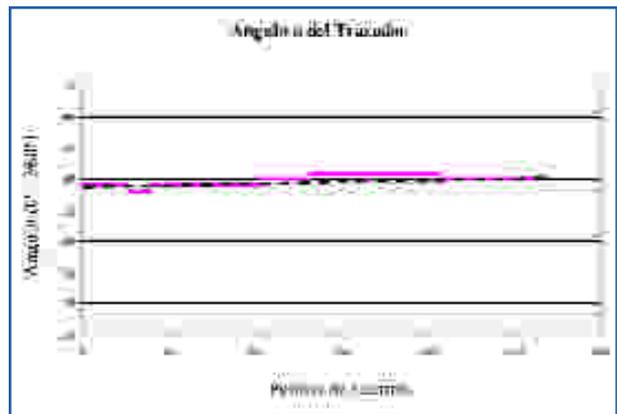


Figura 9. Ángulo del trazador al esbozar una línea recta.

6. El valor del ángulo del trazador al esbozar una línea curva, forma tramos ascendentes y descendentes como se aprecia en la figura 10. Como sucedía con los valores del ángulo, existe también cierta diferencia entre la evolución del valor del ángulo del trazador en función de si se esboza un tramo recto o un tramo curvo.

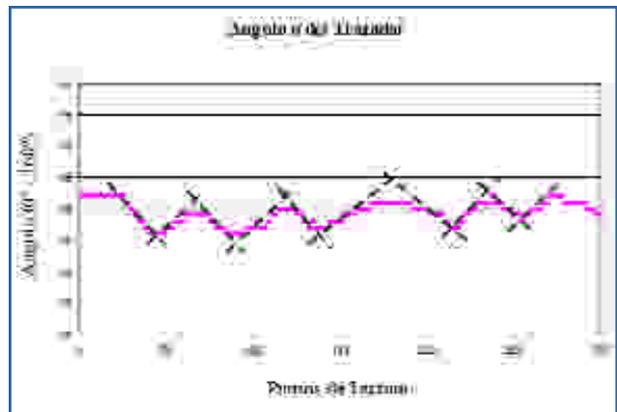


Figura 10. Ángulo del trazador al esbozar una línea curva.

7. Otro factor interesante a tener en cuenta, es que el contacto del trazador con la superficie de la tableta gráfica viene en todo momento controlado por el valor que toma el parámetro de la presión. Es decir, cuando la presión es distinta de cero existe contacto, y cuando toma el valor cero no existe contacto.

5. Conclusiones

A partir de las generalidades expuestas en el punto anterior, se aprecia que procesando adecuadamente los parámetros de trazado, es posible complementar los métodos analíticos existentes para atribuir a cada una de las líneas continuas esbozadas el atributo de línea curva o de línea recta.

Además, de cara a mejorar la capacidad de discriminación de esta interfase, parece prometedor estudiar las posibles relaciones entre los parámetros utilizados en este trabajo y la curvatura de las líneas bocetadas.

Existe la posibilidad de determinar el tipo de línea (continua, de trazos, de trazos y puntos, etc.) a partir de los parámetros obtenidos, evitando de este modo la dificultad existente para conseguir esto utilizando el proceso de vectorización de un esbozo escaneado. Para obtener el tipo de línea es suficiente fijarse en el parámetro de la presión, ya que éste indica la existencia o no de contacto con el área activa de la tableta. La longitud de cada uno de los tramos, y el punto de inicio y final de los mismos se puede obtener de los parámetros de las coordenadas obtenidos en el momento de inicio y fin del contacto del trazado con la tableta gráfica. Es decir, en el momento en que el valor de la presión deja de valer cero, se toman las coordenadas del punto de inicio, y cuando la presión vuelve a valer cero se toman las coordenadas correspondientes a ese evento como punto final del trazo. La obtención de los espacios es posible realizarla del mismo modo.

Una vez clasificado el orden (recta o curva) y la tipología de cada uno de los tramos, procesando adecuadamente las coordenadas de los puntos que los forman (Reconstrucción 2D) se obtiene la información necesaria para crear el archivo que requiere el motor de reconstrucción geométrica.

6. Referencias bibliográficas

[1] JENKINS, D.L.; MARTIN R.R.; *"The importance of free-hand sketching in conceptual design: automatic sketch input"*. Hight T.K. y Stauffer L.A. Eds. ASME, 1993. Design Theory & Methodology, pp. 115-128.

[2] ULLMAN, D.G.; WOOD, S.; CRAIG D.; *"The importance of drawing in the mechanical design process"*. Computers and Graphics. vol. 14, núm. 2, pp. 263-274, 1990.

[3] GOEL V.; *"Sketches of thought"*. Cambridge: MA: MIT Press, 1995.

[4] SUTHERLAND, I.E.; *"Sketchpad: a man-machine graphical communication system"*. Proc. Spring Joint Computer Conference. AFIPS. pp. 329-346. 1963.

[5] NEGROPONTE, N.; *"Recent advances in sketch recognition"*. Proceedings of the AFIPS 1973 National Computer Conference, pp. 663-675, 1973.

[6] HEROT, C.; *"Graphical Input through Machine recognition of Sketches"*. ACM SIGGRAPH Computer Graphics, vol. 10, núm. 2, pp. 97-102, 1976.

[7] Rubine, D.; *"Combining gestures and direct manipulation"*. Proceedings ACM CHI'92 Conference Human Factors in Computing Systems, pp. 659-660, 1992.

[8] LONG, A.C.; LANDAY, J.A.; ROWE, L.A.; MICHIELS, J.; *"Visual Similarity of Pen Gestures"*. Proceedings of Human Factors in Computer Systems (SIGCHI), pp. 360-367, 2000.

[9] ZELEZNIK, R.C.; RENDÓN, K.P.; HUGHES, J.F.; *"SKETCH: An interface for sketching 3D scenes"*. SIGGRAPH'96 Conference Proceedings, pp. 163-170, 1996.

[10] BLOOMENTHAL, K.; ZELEZNIK, R.C.; otros; *"SKETCH-N-MAKE: Automated machining of CAD sketches"*. Proceedings of ASME DETC'98, pp. 1-11, 1998.

[11] EGLI, L.; HSU, C.; BRÜDERLIN, B.D.; ELBER, G.; *"Inferring 3D models from freehand sketches and constraints"*. Computer-Aided Design, vol. 29, núm. 2, pp. 101-112, 1997.

[12] IGARASHI, T.; MATSUOKA, S.; TANAKA, H.; *"Teddy: a sketching interface for 3D freeform design"*. ACM SIGGRAPH 99 Conference Proceedings, pp. 409-416, 1999.

[13] 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.

[14] HUFFMAN, D.A.; *"Impossible objects as nonsense sentences en Meltzer B., Michie D."*, Eds. Machine intelligence, Núm. 6, Edimburgo UK. Edinburgh University Press, pp. 295-323. 1971.

[15] CLOWES, M.B.; *"On seeing things"*. Artificial Intelligence, 2, pp. 79-116. 1971.

- [16] MARTÍ, E.; REGINCÓS, J.; LÓPEZ-KRAHE, J.; VILLANUEVA, J.J.; "Hand line drawing interpretation as three-dimensional objects". *Signal Processing*, vol. 32, núms. 1-2, pp. 91-110, 1993.
- [17] MARILL, T.; "Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects". *International Journal of Computer Vision*, vol. 6, núm. 2, pp. 147-161, 1991.
- [18] LECLERC, Y.; FISCHLER, M.; "An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames". *International Journal of Computer Vision*, vol. 9, núm. 2, pp. 113-136, 1992.
- [19] WANG, W.; GRINSTEIN, G.; "A Survey of 3D Solid Reconstruction from 2D Projection Line Drawings". *Computer Graphics Forum*, vol. 12, núm. 2, pp. 137-158, 1993.
- [20] LIPSON, H.; SHPITALNI, M.; "Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing". *Computer Aided Design*, vol. 28, núm. 8, pp. 651-663, 1996.
- [21] SCHWEIKARDT, E.; GROSS, M.D.; "Digital Clay: Deriving digital models from freehand sketches". *ACADIA '98*, Sebohm T. y Wyk S. V. eds., Quebec City, Canada, pp. 202-211, 1998.
- [22] GROSS, M.D.; DO, E.Y.; "Drawing on the Back of an Envelope: a framework for interacting with application programs by freehand drawing". *Computers and Graphics*, vol. 24, núm. 12, pp. 835-849, 2000.
- [23] TURNER, A.; CHAPMAN, D.; PENN, A.; "Sketching space". *Computers and Graphics*, vol. 24, núm. 12, pp. 869-879, 2000.
- [24] COMPANY, P.; GOMIS, J.M.; CONTERO, M.; "Geometrical reconstruction from single line drawings using optimization-based approaches". Plzen (República Checa). Conference Proceedings of the 7th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media. 1999. ISBN: 80-7082-490-5, vol. II, pp. 361-368.
- [25] CONESA, J.; CONTERO, M.; COMPANY, P. "Comportamiento de los algoritmos de optimización en la reconstrucción geométrica de sólidos". *Anales de Ingeniería Mecánica*. Año 13, vol. 1, pp. 425-430, 1999. ISSN: 0212-5072.