

C_{AL}I: Uma Biblioteca de Componentes para Interfaces Caligráficas

Bem estruturada!

Manuel J. Fonseca ⁴⁺ Joaquim A. Jorge
Departamento de Engenharia Informática, IST/UTL
Av. Rovisco Pais, 1049-001 Lisboa
mjf@ist.utl.pt, jorgej@acm.org

Sumário

A utilização de diagramas e esboços é natural na comunicação entre pessoas em domínios que vão desde a engenharia mecânica e civil à música. No entanto, apesar dos diagramas serem de uso comum, as interfaces utilizador correntes, encerradas que estão na metáfora do tempo de secretária e manipulação directa, tendem a utilizar modos de interacção mais limitados e menos "naturais". Tal deve-se em grande parte à ausência de componentes de base que permitam aos programadores criarem interfaces mais sofisticadas e intuitivas. Este artigo apresenta uma modesta contribuição para a construção de interfaces caligráficas (interfaces inteligentes baseadas em esboços e interacção gestual) centrada num reconhecedor simples, ainda que poderoso de formas geométricas utilizando lógica difusa.

Palavras-chave

Reconhecimento de Gestos, Lógica Difusa, Ferramentas e Técnicas, Formas Geométricas com vários Traços

1. INTRODUÇÃO

Os fenomenais aumentos de velocidade e capacidade de processamento dos sistemas informáticos na última década vêm possibilitar estilos e técnicas de interacção de uma sofisticação crescente. Em particular, as técnicas contínuas e baseadas em reconhecimento têm o potencial de substituir com vantagem as técnicas convencionais baseadas em menus, janelas, cursor e manipulação directa (WIMP).

Em particular, retomando uma ideia antiga que se renova, as interfaces Caligráficas [Jorge 94] utilizam técnicas de desenho baseadas no reconhecimento de esboços e diagramas estruturados para facilitar inúmeras tarefas ligadas ao desenho técnico e esquemático. Tais técnicas permitem novos ganhos em usabilidade e naturalidade dado que capitalizam na utilização de aptidões adquiridas pela maioria dos utilizadores desde tenra idade, como o desenho com lápis e papel.

Se por um lado, a utilização de técnicas de reconhecimento perspectiva a simulação de comportamentos inteligentes e um carácter "mais natural" das interacções, por outro lado estas técnicas acarretam uma maior complexidade tanto do ponto de vista da engenharia de software como arquitectura e construção de diálogos. A presente comunicação descreve uma biblioteca de componentes para a construção de interfaces caligráficas. A biblioteca baseia-se num núcleo

de reconhecimento utilizando lógica difusa e atributos geométricos, que permite o reconhecimento de formas geométricas e comandos gestuais.

Começamos por descrever o algoritmo utilizado, de uma arquitectura extremamente simples, recorrendo a características geométricas elementares e de fácil cálculo que permitem reconhecer formas simples e variações de traço de um modo surpreendentemente robusto e flexível. Entre outras características o algoritmo de reconhecimento permite reconhecer figuras rodadas, independentemente da escala e tipo de traço com que foram desenhadas.

Em seguida discutimos aspectos de engenharia de software relacionados com a nossa abordagem, materializados numa biblioteca de componentes e suas implicações do ponto de vista da interface. Esta biblioteca foi utilizada na construção de um editor prototípico que apresentamos servindo de pano de fundo à discussão sobre o processamento de resultados ambíguos na interface. Este editor e a biblioteca de suporte foram utilizados numa avaliação experimental da nossa abordagem, que apresentamos e discutimos.

As interfaces baseadas em reconhecimento e as interfaces caligráficas em particular colocam um sem número de problemas interessantes cujo cerne ainda não foi sequer afluído. Na última parte desta contribuição discutimos alguns dos desafios que se nos afiguram mais interessantes e modos em que pretendemos abordá-los

em trabalhos futuros.

2. TRABALHO RELACIONADO

A ideia das interfaces caligráficas não é recente, mesmo se descontarmos o facto de a utilização de esboços e diagramas planares como forma de comunicação preceder a invenção da escrita em mais de 30 séculos [Harley 87]. Em 1963, Sutherland apresentava Sketchpad [Sutherland 63] o primeiro sistema interactivo, que utilizava uma caneta foto-sensível para introduzir diagramas directamente sobre a superfície de um ecrã. A principal limitação deste sistema residia nas capacidades, recursos limitados e custo elevado do computador utilizado.

No entanto, devido em parte aos problemas de ergonomia das canetas foto-sensíveis e à invenção do rato em 1964, as interfaces gráficas hoje em dia relegaram as interfaces baseadas em caneta para aplicações restritas em sistemas de CAD até ao aparecimento dos primeiros computadores de caneta em 1991. Mesmo estes utilizavam principalmente as canetas como instrumento de introdução de texto através de reconhecimento de escrita. O nosso presente trabalho assenta numa primeira abordagem ao reconhecimento de esquemas desenvolvida na Universidade de Washington [Apte 93] que apenas permitia reconhecer um número restrito de formas não rodadas e não permitia identificar tracejado, figuras abertas ou sobreposição de riscos. Tappert [Tappert 90], menciona também todo o trabalho desenvolvido no âmbito do reconhecimento não-interactivo de gráficos para aplicações de introdução de dados legados e diagramas em sistemas de CAD vectoriais. Rui Zhao [Zhao 93] descreve um editor interactivo de diagramas baseados no formalismo das StateCharts [Harel 87] embora o editor descrito seja baseado em rato e manipulação directa, muitas das ideias descritas por Zhao exprimem uma abordagem baseada no reconhecimento de diagramas guiado pela sintaxe.

Gross [Gross 96] descreve um sistema baseada fundamentalmente em esboços que só são parcialmente interpretados para utilização em desenhos de arquitectura, utilizando um reconhecedor substancialmente mais simples e menos robusto que o nosso. Por outro lado James Landay [Landay 96] utiliza um reconhecedor desenvolvido por Dean Rubine [Rubine 91] para esboçar o arranjo gráfico de documentos bidimensionais, embora não seja clara a utilização do reconhecimento de esboços. O reconhecedor de Rubine embora treinável não exhibe um desempenho suficientemente elevado para utilização consistente. O nosso reconhecedor embora mais complexo permite identificar um conjunto de formas geométricas de forma mais robusta embora não seja (por enquanto) possível o treino de novas primitivas, constituindo, deste ponto de vista uma abordagem complementar à de Rubine.

Nos parágrafos seguintes descrevemos sucintamente o método de reconhecimento que está na base da biblio-

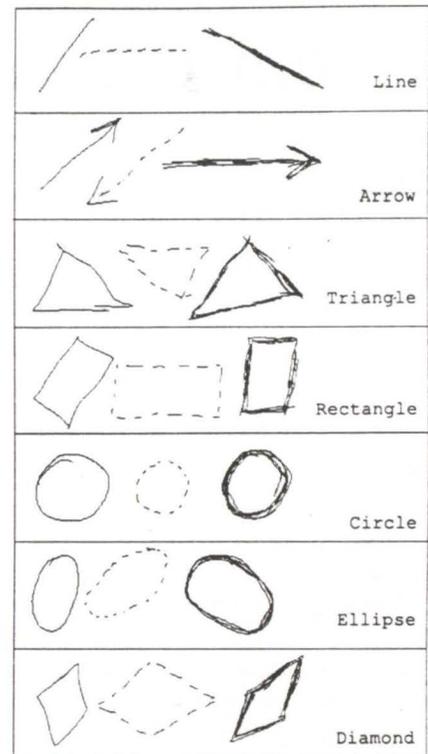


Figura 1. Formas geométricas com múltiplos traços.

teca de componentes, colocando a ênfase na descrição dos aspectos de engenharia de software e interface utilizador envolvidos na concepção e desenho de interfaces caligráficas.

3. ALGORITMO DE RECONHECIMENTO

O algoritmo de reconhecimento baseia-se em três ideias principais. Primeira, utiliza apenas propriedades geométricas globais calculadas a partir dos esboços desenhados. Uma vez que estamos prioritariamente interessados em identificar entidades geométricas, o reconhecedor baseia-se principalmente em informação geométrica. Segunda, de modo a melhorar o desempenho do reconhecimento, utilizamos um conjunto de regras tanto para identificar formas como para eliminar as não desejadas através de critérios distintos. Terceira, para lidar com incertezas e imprecisões do esboço de formas geométricas, utilizamos lógica difusa [Bezdek 92] para associar graus de certeza às formas reconhecidas, tratando de forma natural as ambiguidades.

Este algoritmo reconhece formas geométricas elementares, tais como Triangles, Rectangles, Diamonds, Circles, Ellipses, Lines e Arrows, assim como cinco comandos gestuais, Delete, Undo, WavyLine, Move e Copy, como se apresenta nas Figuras 1 e 2. As formas são reconhecidas independentemente da sua rotação, tamanho ou número de traços que a consti-

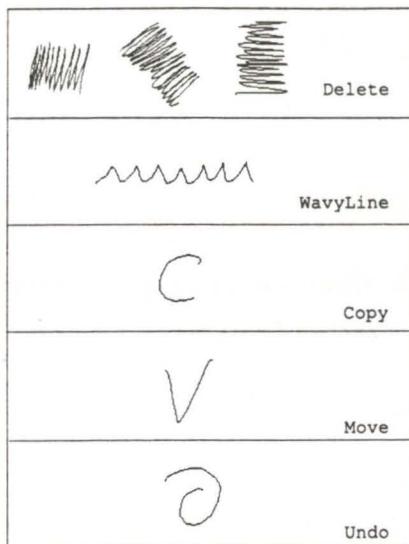


Figura 2. Comandos gestuais unitraço.

tuem. Os comandos são formas desenhadas com apenas um traço.

O reconhecedor funciona através da verificação da existência de valores para características específicas em conjuntos difusos associados a cada forma e comando. Este processo produz uma lista de formas possíveis, ordenadas por grau de certeza (dom).

O algoritmo faz a distinção entre formas Solid, Dashed ou Bold, depois de identificar a forma "base". Isto permite-nos tratar o tipo de linha de modo ortogonal relativamente à forma, tornando a concepção mais modelar, e fazendo com que seja mais fácil reconhecer novas formas no futuro.

Esta aproximação estende e melhora o trabalho desenvolvido por Kimura [Apte 93], e que não distinguia formas rodadas, a tracejado, abertas e com traço grosso. O presente trabalho é uma evolução de um anterior [Jorge 99], que reconhecia menos formas e utilizava uma árvore de decisão para eliminar as classificações incorrectas. Chegamos à conclusão que a utilização de regras de classificação permite mais flexibilidade e extensibilidade sem sacrificar a robustez. Outros autores [Ulgen 95] propuseram métodos mais complexos, envolvendo redes neuronais e procedimentos que podem ser mais robustos que o nosso, mas apesar de reivindicados não são explícitos nem substanciados por resultados experimentais.

3.1 Características Geométricas

Começamos o processo de reconhecimento pela recolha de pontos utilizando uma tablete digitalizadora, desde o primeiro evento de pousar a caneta até terminar o *timeout* lançado depois do último evento de levantar a caneta. Depois de recolhidos todos os pontos calculamos o polígono convexo correspondente, usando o algoritmo de Graham [O'Rourke 98]. Este

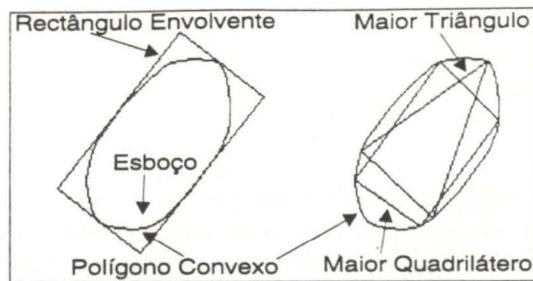


Figura 3. Polígonos usados para estimar as características.

polígono é posteriormente utilizado para calcular três polígonos especiais. Os dois primeiros são o triângulo e o quadrilátero de maior área inscritos no polígono convexo [Boyce 85], e o terceiro é o rectângulo envolvente de menor área [Freeman 75] (ver Figura 3). Finalmente calculamos a área e o perímetro para cada um dos polígonos anteriores, de modo a estimar características e graus de pertença para cada classe de formas.

Um *Traço* é definido como um conjunto de pontos desde que se pousa a caneta até a levantar. Um *Esboço* é uma sequência de traços recolhidos durante o tempo de *timeout*. Uma *Forma* é um esboço reconhecido, e relaciona os pontos desenhados com uma classe e um conjunto de atributos geométricos, tais como o ponto de início e de fim ou a caixa envolvente.

O conjunto de formas seleccionadas e apresentadas nas Figuras 1 e 2 são os elementos básicos para permitir a construção de diagramas técnicos, tais como circuitos eléctricos ou lógicos, fluxogramas ou desenhos de arquitectura. Estes tipos de diagramas exigem a distinção entre a representação a tracejado, grosso ou normal, na mesma família de formas. Tipicamente, os arquitectos usarão traços sobrepostos para engrossar as linhas nos seus esboços. Este mecanismo é frequentemente usado em alguns pacotes de desenho. Portanto, pretendemos reconhecer diferentes modos de dese-

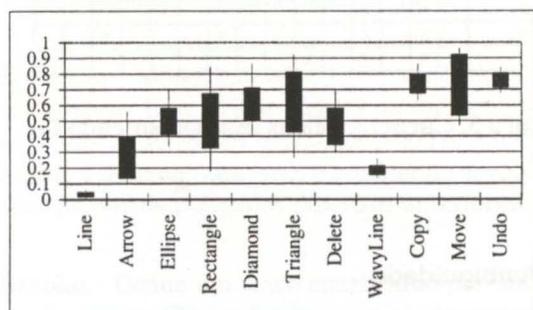


Figura 4. Percentis para a relação entre altura e largura do rectângulo envolvente.

nhar a mesma forma, tal como se mostra na Figura 1. Estamos interessados em obter diferenças qualitativas e não em obter valores precisos dos atributos, não considerando diferentes tipos de linhas e grossuras.

De modo a seleccionar as características que melhor identificam uma determinada forma, construímos gráficos de percentis para cada característica. Estes gráficos ilustram a distribuição estatística dos valores da característica pelas diferentes classes, extraídos a partir dos desenhos de treino. Para cada forma, a barra a preto vai desde o percentil 25% ao 75%, enquanto a linha vai do percentil 10% ao percentil 90% dos valores observados para uma determinada característica. A Figura 4 apresenta um gráfico de percentis para a característica que relaciona a altura com a largura do rectângulo envolvente.

A selecção inicial de características teve em atenção propriedades específicas de cada forma que se pretendia identificar. Associada a cada característica inferimos conjuntos difusos dos dados de treino, que traduziam os valores permitidos para uma determinada forma. Normalmente, uma só característica não é suficiente para distinguir as formas, levando a classificações incorrectas. Deste modo, adicionamos características extra (com os correspondentes conjuntos difusos) de modo a retirar classificações indesejadas, levando a regras mais complexas quando necessário.

As principais características seleccionadas, são relações entre perímetros e áreas dos polígonos especiais, descritos anteriormente. A Figura 5 lista todas as formas identificadas pelo reconhecedor e as características utilizadas por cada uma.

	Hollowness	Pch2/Ach	Her/Wer	Tl/Pch	Ach/Aer	Alq/Ach	Plu/Pch	Hm/Wbb	Alu/Ach	Alq/Aer	Plq/Pch	Alu/Alq	Pch*Ns/Tl	Vm/Hbb	Alu/Aer
Line			X												
Arrow	X	X			X	X					X				
Triangle	X					X		X							
Rectangle	X				X					X					
Circle	X	X													
Ellipse	X	X				X									
Diamond	X	X				X				X		X			
Delete	X		X	X	X										
WavyLine	X	X						X							
Copy		X	X				X			X				X	
Move		X	X			X		X					X		
Undo	X	X		X	X										X

Figura 5. Características usadas por cada forma.

3.2 Ambiguidade

Tendo em conta as formas identificadas pelo reconhecedor, apresentamos quatro casos especiais em que podemos obter resultados ambíguos. A ambiguidade existe entre Lines e WavyLines, WavyLines e Deletes, Circles e Ellipses; Diamonds e

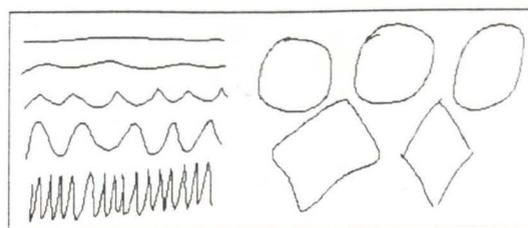


Figura 6. Casos de ambiguidade entre formas.

Rectangles (ver Figura 6).

As pessoas resolvem esta ambiguidade natural entre formas geométricas identificando mais que uma forma e fazendo a distinção final com base no contexto ou em informação adicional.

O reconhecedor descrito neste artigo lida com a ambiguidade de modo semelhante, isto é, quando não consegue identificar univocamente uma forma geométrica, devolve uma lista de possíveis candidatos. A escolha do melhor candidato será depois feita pela aplicação usando informação do contexto.

A ambiguidade entre formas é modelada naturalmente, usando lógica difusa para associar graus de certeza às formas reconhecidas. A Figura 7 mostra os conjuntos difusos correspondentes aos casos ambíguos ilustrados na Figura 6.

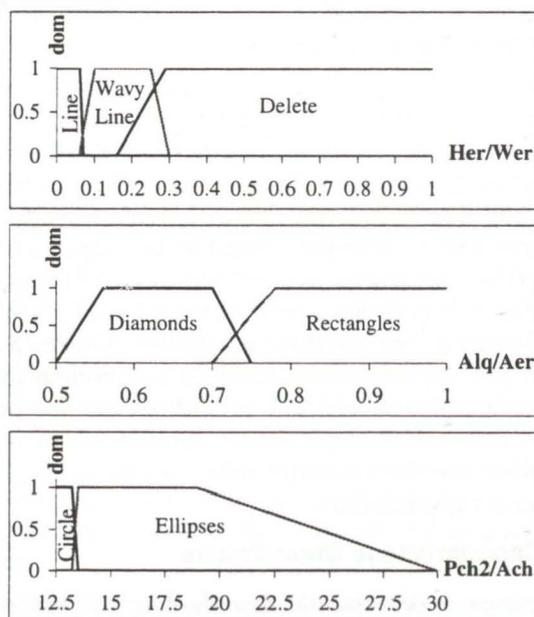


Figura 7. Conjuntos difusos representando os casos de ambiguidade suportados pelo reconhecedor.

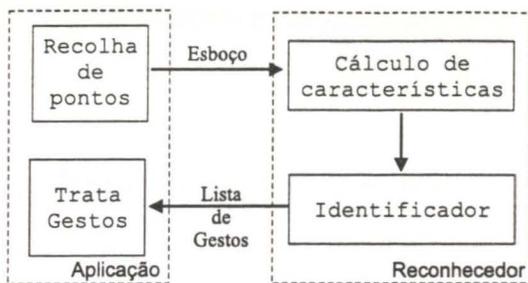


Figura 8. Diagrama de blocos.

4. ARQUITECTURA DE SOFTWARE

A biblioteca¹ de componentes, *CALI*, foi desenvolvida no ambiente Windows 95/98, utilizando o compilador MS Visual C++. Actualmente estamos a canalizar esforços para portar a biblioteca para a plataforma Linux. A biblioteca oferece mecanismo para reconhecer esboços de formas geométricas e comandos gestuais desenhados à mão livre.

A Figura 8 mostra os principais módulos do reconhecedor, assim com os módulos a desenvolver do lado da aplicação. Do lado da aplicação devem-se desenvolver dois módulos, um responsável por recolher os pontos que constituem o esboço e outro para receber e manipular os gestos devolvidos pelo reconhecedor.

O reconhecedor, por seu lado tem dois módulos principais. O primeiro recebe o esboço enviado pela aplicação e calcula o valor das características seleccionadas. O segundo, utiliza os valores das características para identificar os gestos cujos conjuntos difusos contêm os valores calculados. Os gestos identificados são colocados numa lista, ordenados decrescentemente pelo grau de certeza e devolvidos à aplicação.

4.1 Diagrama de Classes

A Figura 9 ilustra o diagrama de classes da biblioteca de componentes, que passamos a descrever.

CIRecognizer Componente principal da biblioteca, que interage directamente com as aplicações caligráficas identificando os esboços desenhados. A classe implementa um reconhecedor de formas geométricas e comandos gestuais baseado principalmente em informação geométrica extraída a partir do esboço. O reconhecedor utiliza lógica difusa e um conjunto de regras para classificar cada um dos potenciais gestos. O principal método desta classe é o `recognize()` que recebe como entrada um esboço, *CIScribble*, e devolve uma lista de possíveis gestos ordenados por ordem decrescente de grau de certeza.

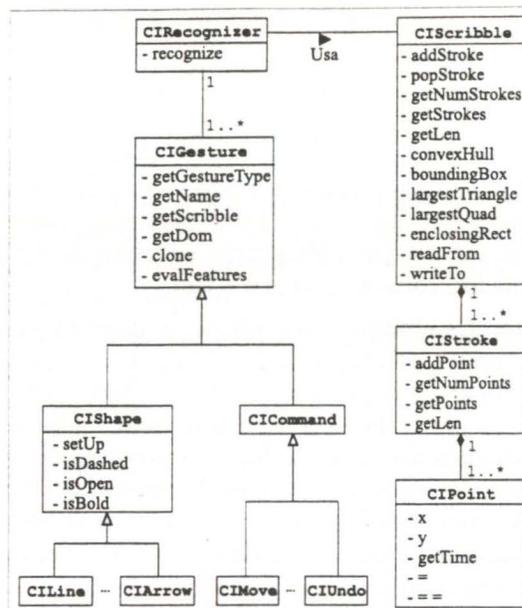


Figura 9. Hierarquia de classes da biblioteca.

CIGesture Define todas as entidades reconhecíveis, formas e comandos. Os elementos desta classe têm associados a si o esboço que lhes deu origem e o grau de certeza com que foram identificados pelo reconhecedor.

CISHape Caso particular de *CIGesture*, que modeliza as formas geométricas simples. Os elementos desta classe possuem atributos e uma definição geométrica. Os atributos são o `open`, que indica se a forma é aberta ou fechada, o `dashed`, que indica o tipo de traço utilizado, normal ou tracejado e finalmente o `bold`, que indica se a forma foi desenhado a grosso. As formas são definidas por um conjunto de pontos.

CIconmand Caso particular dos gestos, mas que ao contrário das formas, não têm atributos, nem definição geométrica. Os comandos não têm representação e tipicamente desencadeiam acções.

CIScribble Modeliza esboços constituídos por vários traços. Esta classe oferece métodos para calcular os vários polígonos utilizados pelo reconhecedor, tais como o polígono convexo, o rectângulo envolvente, e todos os outros polígonos descritos anteriormente.

CISTroke Define um traço constituído por um conjunto de pontos. Tem métodos para aceder e manipular esses pontos.

CIPoint Classe que modeliza pontos bidimensionais com uma marca temporal associada.

¹A biblioteca encontra-se disponível em <http://asterix.ist.utl.pt/~mjf/research/cali>

5. UM EXEMPLO: EDITOR CALIGRÁFICO

No passado desenvolvemos um editor multimodal de formas geométricas [Fonseca 98], baseado numa versão rudimentar do reconhecedor de gestos, que identificava menos formas geométricas e não tratava o problema da ambiguidade. O principal objectivo desse editor foi estudar a integração sinérgica de várias modalidades, como os gestos e a fala.

O Editor Caligráfico desenvolvido, e descrito nesta secção, utiliza a biblioteca de componentes descrita anteriormente, e permite o desenho de diagramas “rigorosos” a partir de esboços “imperfeitos”. Este editor possibilita a criação de formas geométricas ou a invocação de comandos, através de esboços desenhados sobre uma mesa digitalizadora. As formas podem ser desenhadas, copiadas, movidas e redimensionadas, apenas com a utilização da caneta.

Para apagar um objecto ou conjunto de objectos, podemos usar o comando gestual *Delete* (ver Figura 10). O editor fornece ainda um comando de *Undo* e outro de *Redo* para facilitar a tarefa do utilizador na criação do seu diagrama. Existe ainda uma opção que desliga o reconhecedor, permitindo o desenho de polilinhas.

O editor tem ainda outras funcionalidades adicionais, como a possibilidade de gravar e ler os desenhos efectuados, e o alinhamento das formas desenhadas segundo os eixos cartesianos.

A selecção de objectos pode ser feita através do desenho de um gesto circular em torno destes, ou pressionando a caneta durante algum tempo sobre a fronteira do objecto (ver Figura 11).

O editor apresenta uma forma simples de lidar com a ambiguidade presente no reconhecedor. Sempre que o reconhecedor identifica mais que uma forma para o esboço desenhado, o editor mostra uma régua com essas formas, como se ilustra na Figura 12. Esta solução tem a vantagem de facilitar o processo de criação de diagramas, pois elimina a necessidade de realizar novos esboços.

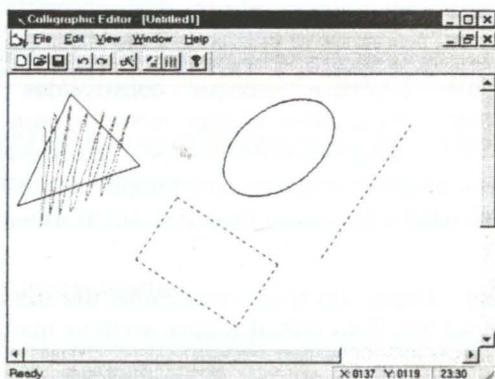


Figura 10. Remoção de formas.

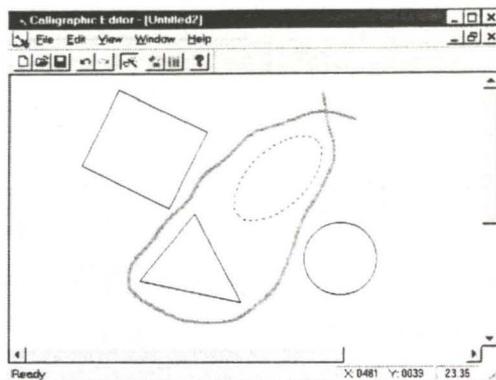


Figura 11. Selecção de várias formas.

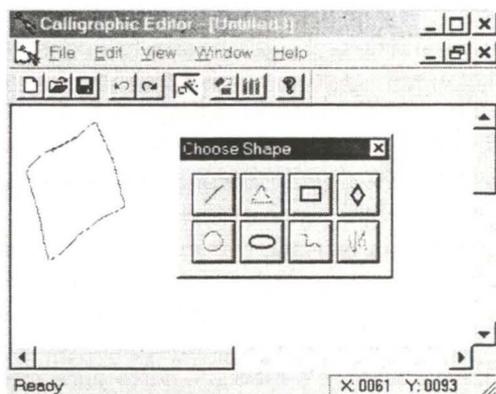


Figura 12. Régua de formas possíveis.

No futuro pretendemos experimentar outras alternativas à régua, como seja a utilização de diferentes cores para indicar se a forma reconhecida é única ou não. No caso de não ser única, teremos um mecanismo que permita circular pelas várias formas devolvidas pelo reconhecedor. Em [Mankoff 99] descreve-se um mecanismo algo semelhante para lidar com os erros de reconhecimento na escrita à mão.

6. AVALIAÇÃO EXPERIMENTAL *9.4.1.2*

Para avaliar o algoritmo de reconhecimento, pedimos a vários utilizadores para desenharem cada forma repetidamente usando linhas normais, a tracejado ou com traço grosso. Os utilizadores foram informados que a experiência pretendia testar o reconhecimento, prevenindo assim a realização de formas “não naturais”.

Utilizamos uma tablete digitalizadora Wacom PL-300, com ecrã LCD incluído, para desenharem as formas. Três utilizadores eram experientes no uso da caneta e tablete, enquanto os restantes nunca tinham usado uma tablete digitalizadora. Demos uma breve descrição do reconhecedor aos utilizadores, incluindo o conjunto de formas e comandos identificados por este. Informamos ainda da possibilidade de utilizar formas com múltiplos traços, da independência relativamente

Tipo Gesto	Primeira Escolha	Três Primeiras
Individual	92%	93%
Grupos	93%	94%

Tabela 1. Taxas de reconhecimento.

à rotação ou tamanho e sobre o *timeout*. Os utilizadores menos experientes tiveram uma pequena sessão prática, no sentido de se acostumarem ao conjunto caneta e tablete digitalizadora. Durante a sessão de desenho o reconhecedor foi desligado, para não interferir com a recolha dos dados.

O reconhecedor identificou com sucesso 92% dos esboços desenhados. Os resultados globais das taxas de reconhecimento estão presentes na Tabela 1. Uma rápida análise à matriz de confusão (não apresentada por falta de espaço) mostra que *Circles* são facilmente confundidos com *Ellipses*, o que é um comportamento aceitável e *intuitivo*. Se considerarmos *Circles* e *Ellipses* "gordas" como pertencentes à mesma classe de formas, então a taxa de reconhecimento aumenta para 93%. Uma segunda análise, mostra que *Diamonds* são normalmente confundidos com *Rectangles*, e têm uma taxa de reconhecimento baixa. Também neste caso, a taxa aumenta se considerarmos as duas classes de formas como uma só (quadriláteros). Mais uma vez, a lógica difusa joga a nosso favor permitindo vários graus de certeza para distinguir estas classificações ambíguas, *Diamonds* são aqueles que têm uma classificação mais restritiva, fazendo com que tenham uma taxa de reconhecimento baixa.

Se considerarmos como classificação bem sucedida aquela em que a resposta correcta se encontra nas três primeiras, então a taxa de reconhecimento aumenta de 92% para 93%. Esta taxa aumenta para 94%, se agruparmos *Circles* com *Ellipses* "gordas" e *Diamonds* com *Rectangles* (última linha da Tabela 1).

7. CONCLUSÕES

Neste artigo descrevemos uma biblioteca de componentes para a construção de interfaces caligráficas. De notar que os problemas colocados por estas interfaces são fundamentalmente diferentes dos inerentes às interfaces convencionais, ou de tempo de secretária. Enquanto que nestas as interacções são discretas (selecção de objectos, opções em menus, introdução de dados em formulários) e os erros se devem fundamentalmente a acções incorrectas do utilizador, as interfaces baseadas em reconhecimento de formas e gestos, a par de uma interacção contínua e potencialmente mais natural, exibem ainda modos de erro e imprecisão próprios e de difícil solução. A correcta solução destes problemas requer disciplinas de programação distintas, não se esgotando na criação de bibliotecas de suporte em conjunto com normas e práticas de de-

senho.

As implicações deste processamento da ambiguidade de forma vantajosa estendem-se desde a concepção de novos estilos de diálogo até a arquitecturas diferentes das baseadas em acontecimentos mais adequadas a este tipo de interfaces. Embora algum trabalho recente [Mankoff 98] aborde alguns destes aspectos, estamos em crer que o tópico se encontra muito longe de uma resposta definitiva.

É nossa crença que estes traços distintivos além de constituírem desafios de investigação profundamente motivadores consistem em vantagens fundamentais de uma nova família de aplicações interactivas, se abordadas de forma correcta.

Referências

- [Apte 93] Ajay Apte, Van Vo, e Takayuki Dan Kimura. Recognizing Multistroke Geometric Shapes: An Experimental Evaluation. Em *Proceedings of the ACM (UIST'93)*, páginas 121-128, Atlanta, GA, 1993.
- [Bezdek 92] James C. Bezdek e Sankar K. Pal. *Fuzzy Models for Pattern Recognition*. IEEE Press, 1992.
- [Boyce 85] J. E. Boyce e D. P. Dobkin. Finding Extremal Polygons. *SIAM Journal on Computing*, 14(1):134-147, Fevereiro 1985.
- [Fonseca 98] Manuel J. Fonseca e Joaquim A. Jorge. Eddy: Um Editor Multimodal com Reconhecimento de Fala e Gestos. Em *Actas do 8º Encontro Português de Computação Gráfica*, páginas 49-64, Coimbra, Portugal, Fevereiro 1998.
- [Freeman 75] H. Freeman e R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409-413, July 1975.
- [Gross 96] Mark D. Gross. The Electronic Cocktail Napkin - A computational environment for working with design diagrams. *Design Studies*, 17(1):53-69, 1996.
- [Harel 87] David Harel. StateCharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231-274, 1987.
- [Harley 87] J. B. Harley e D. Woodward, editores. *The History of Cartography*, volume 1. University of Chicago Press, Chicago, IL, 1987.

- [Jorge 94] Joaquim A. Jorge. *Parsing Adjacency Grammars for Calligraphic Interfaces*. Tese de Doutorado, Rensselaer Polytechnic Institute, Troy, New York - USA, Dezembro 1994.
- [Jorge 99] Joaquim A. Jorge e Manuel J. Fonseca. A Simple Approach to Recognise Geometric Shapes Interactively. Em *Proceedings of the Third Int. Workshop on Graphics Recognition (GREC'99)*, Jaipur, India, Setembro 1999.
- [Landay 96] James A. Landay. *Interactive Sketching for the Early Stages of User Interface Design*. Tese de Doutorado, Carnegie Mellon University, Computer Science, Pittsburgh - USA, Dezembro 1996.
- [Mankoff 98] Jennifer Mankoff e Gregory D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. Em *Proceedings of the ACM (UIST'98)*, Technical note, páginas 213-214, 1998.
- [Mankoff 99] Jennifer Mankoff e Gregory D. Abowd. Error Correction Techniques for Handwriting, Speech, and other ambiguous or error prone systems. Relatório técnico, GVVU TechReport GIT-GVVU-99-18, Junho 1999.
- [O'Rourke 98] Joseph O'Rourke. *Computational geometry in C*. Cambridge University Press, 2nd edição, 1998.
- [Rubine 91] Dean Harris Rubine. *The Automatic Recognition of Gestures*. Tese de Doutorado, Carnegie Mellon University, Computer Science, Pittsburgh - USA, Dezembro 1991.
- [Sutherland 63] Ivan E. Sutherland. Sketchpad: A Man-Machine Graphical Communication System. Em *Spring Joint Computer Conference*, páginas 2-19. AFIPS Press, 1963.
- [Tappert 90] Charles C. Tappert, Ching Y. Suen, e Toru Wakahara. The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):787-807, August 1990.
- [Ulgen 95] F. Ulgen, A. Flavell, e N. Akamatsu. Geometric shape recognition with fuzzy filtered input to a backpropagation neural network. *IEEE Trans.*
- [Zhao 93] Rui Zhao. Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors. Em *Proceedings of the INTERCHI'93*, páginas 95-100, Amsterdam, 1993.
- Inf. & Syst.*, E788-D(2):174-183, February 1995.

GIDeS: Uma Abordagem Caligráfica à Edição 3D

João P. Pereira ~~3.5~~
Dep. de Eng.ª Informática
ISEP/INESC
R. de São Tomé, Porto
jpp@dei.isep.ipp.pt

Joaquim A. Jorge ~~3.5~~
Dep. de Engenharia Informática
IST/UTL
Av. Rovisco Pais, Lisboa
jorge@acm.org

Vasco Branco
Dep. de Comunicação e Arte
Univ. de Aveiro
Aveiro
vab@ca.ua.pt

F. Nunes Ferreira
Dep. de Eng.ª Electrotécnica e de Computadores
FEUP
R. dos Bragas, Porto
fnf@fe.up.pt

Sumário

GIDeS (Gesture-based Intuitive Design System) é um sistema gestual de modelação que visa ultrapassar as conhecidas insuficiências ergonómicas dos sistemas de CAD actuais no design de formas de objectos em fase de concepção. O seu desenvolvimento teve em vista a utilização de um computador munido de um ecrã táctil e de um estilete, na tentativa de estabelecer uma aproximação dos interfaces de CAD aos tradicionais papel e lápis, normalmente preferidos pelos designers nas fases iniciais do seu trabalho. O sistema GIDeS reage às acções gestuais do utilizador com informação dependente do contexto, apresentada sob a forma icónica, de modo a minimizar a carga cognitiva imposta ao utilizador e focar a sua atenção na tarefa de desenho.

Palavras-chave

Técnicas de Interação, Modelação 3D, Interfaces Gestuais, Desenho de Esboços, Interfaces Caligráficos.

1. INTRODUÇÃO

Apesar da grande evolução verificada nos sistemas de CAD nos últimos 30 anos, a sua utilização como ferramenta de desenho das formas que traduzem os objectos idealizados na mente dos designers, continua a ser muito complexa e pouco natural, quando comparada com a simplicidade do papel e lápis. Isto deve-se, sobretudo, ao facto de mesmo os mais "amigáveis" sistemas de computação imporem ao utilizador diálogos rígidos e muito estruturados, perturbando o fluxo criativo das ideias. Desta forma não surpreende que o computador seja preferido face aos tradicionais esboços a lápis, na representação rápida de formas de objectos em fase de concepção. Não obstante, se se conseguir que os computadores sejam utilizados logo nas fases iniciais do design, as vantagens poderão ser enormes, dado o impacto que as alterações iniciais do produto têm nos últimos estágios do processo de desenvolvimento deste.

Além disso, se uma imagem vale mil palavras, os modelos físicos valem um número ilimitado de imagens [Potter94]. Actualmente, poucas ideias acabam por se concretizar em modelos físicos, devido à grande dificuldade que a criação dos mesmos acarreta, quando comparada com a facilidade com que podem ser esboçados numa

folha de papel. Isto poderia alterar-se significativamente, caso se conseguisse simplificar o processo de produção de modelos físicos a partir de esboços.

Em 1994 apresentámos IDeS [Branco94], um sistema vectorial, manobrado por menus, que tentou explorar a ideia de criar modelos poliédricos aproximados através da combinação de desenho e de operações de construção. GIDeS é mais um passo no sentido de criar uma nova geração de interfaces com o utilizador, com o intuito de ajudar e não de entrar o mesmo na tarefa de produzir modelos a partir de esboços. Chamamos a estes interfaces, baseados no desenho de esboços, interfaces caligráficas [Jorge94], na medida em que estes lidam explicitamente com a ambiguidade e a imprecisão características dos seres humanos, transformando-as em mais valias que levam a uma maior aproximação dos sistemas computacionais de apoio ao design aos tradicionais papel e lápis.

O sistema GIDeS foi projectado para funcionar num computador munido de um ecrã táctil e de um estilete. Substituímos o estilo de interação baseado em menus por um estilo gestual, de modo a aumentar a usabilidade do sistema. Incluímos, também, um mecanismo de ícones dependentes do contexto, o que permitiu não só



Figura 1: Ícones dependentes do contexto

reduzir a carga cognitiva do utilizador, como também lidar com ambiguidades.

O designer pode desenhar sem ter que se preocupar com a memorização dos gestos de modelação, pois sempre que os seus gestos são reconhecidos, a aplicação apresenta uma barra composta por um ícone (Fig. 1a), ou conjunto de ícones (Fig. 1b), relacionados com o contexto do desenho, numa tentativa de antecipar o que quer que seja que o utilizador tenha em mente. Este poderá aceitar a sugestão do sistema ou, em alternativa, prosseguir com o desenho.

Deste modo, os ícones dependentes do contexto providenciam uma maneira intuitiva de acelerar o processo de design, sem pôr em causa a liberdade de desenho do utilizador. Também constituem uma maneira ergonómica e tecnicamente interessante de lidar com eventuais ambiguidades do processo de reconhecimento.

Neste artigo procedemos à descrição do funcionamento do interface caligráfico do sistema GIDeS, comparando-o com outros trabalhos e apresentando a nossa abordagem ao processo de lidar de uma forma natural com a interacção ambígua. Certos pormenores do funcionamento do sistema são complementados com exemplos, de forma a facilitar a sua compreensão. Por último, fazemos uma breve descrição da investigação em curso e do trabalho a realizar futuramente.

2. TRABALHO RELACIONADO

Apesar do muito trabalho que tem sido feito no campo das metodologias de design [Jones92], o desenho continua a ser o principal método de concepção de objectos 3D. Isto leva-nos a concluir que o paradigma fundamental do desenvolvimento de interfaces para a modelação 3D deve apoiar-se em esboços e desenhos.

A aceitação deste paradigma tem como consequência que a nossa abordagem ao processo de modelação com base em esboços recorra à utilização de dispositivos de interacção 2D, não obstante os dispositivos 3D poderem ser mais adequados para as fases posteriores do processo de design [Galyean91] [Sachs91].

Zelevnik et. al. [Zelevnik96] [Forsberg97] trouxeram o reconhecimento de gestos para a área da modelação 3D. Os gestos do utilizador fornecem a informação suficiente para determinar a primitiva a criar, bem como as suas dimensões e posicionamento na cena.

No sistema SKETCH [Zelevnik96], toda a interacção é feita com base num rato de três botões, ocasionalmente conjugado com uma tecla modificadora. Há dois tipos de elementos gestuais - cinco classes de traços (feitos com o primeiro botão do rato) e duas classes de agentes de in-

teracção (feitos com o segundo botão). A manipulação da câmara recorre ao terceiro botão do rato.

Outro aspecto do sistema SKETCH reside no recurso ao sentido de desenho dos traços para inferir as operações CSG.

A nossa abordagem é diferente. Dado o estilete não ter botões, os comandos e primitivas têm de ser inferidos apenas com base na informação disponível no desenho feito pelo utilizador. Por outro lado, o sentido de desenho dos traços que compõem os gestos é irrelevante. Isto tem a vantagem de não perturbar os processos normais de pensamento e de desenho do designer e de estabelecer diálogos mais "naturais" com o mesmo. A liberdade de desenho dos designers deve ser respeitada, de modo a que estes possam focar a sua atenção no desenvolvimento das suas ideias, sem terem que pensar constantemente nas peculiaridades do processo de interacção com o sistema.

Isto implica que, por exemplo, o reconhecimento de um gesto não conduza directamente à criação de uma primitiva 3D mas, em vez disso, a uma sugestão da sua utilização.

Encarnação et. al. [Encarnação99] apresentam um sistema que combina as metáforas tradicionais de secretária com um interface de realidade virtual. A inclusão neste sistema de tecnologias de interacção 3D, tais como sensores com seis graus de liberdade e óculos estereoscópicos, permite ao utilizador a criação directa de objectos simples num espaço 3D virtual, mediante o uso de gestos icónicos que evocam os contornos da projecção de cima para baixo da geometria dos objectos. O sistema requer equipamento muito sofisticado, tais como uma caneta e um painel transparentes, os já referidos óculos estereoscópicos, localizadores magnéticos e um dispositivo de visualização do tipo mesa virtual.

No caso do sistema GIDeS, optámos por uma abordagem diferente e minimalista. A criação de objectos é feita de uma forma "natural" e construtiva, com base no desenho de um esboço 2D da geometria do objecto. Além disso, a metáfora do papel e lápis evita a necessidade de utilização de hardware sofisticado e dispendioso.

Igarashi et. al. [Igarashi99] desenvolveram um sistema (Teddy) que permite criar modelos 3D de formas livres, tais como animais recheados e outros objectos rotundos. O utilizador começa por esboçar em 2D a silhueta do objecto que pretende criar, após o que o sistema procede automaticamente à construção de uma superfície poligonal tridimensional. O objecto pode ser posteriormente modificado, com base num conjunto disponível de operações de edição.

Uma destas operações, no entanto, não é puramente gestual, pois requer uma mudança do modo de interacção, a qual é efectuada carregando num dos botões auxiliares que estão dispostos na parte inferior da janela de trabalho.

Além disso, as operações de manipulação da câmara requerem, tal como no sistema SKETCH, o uso de um botão secundário do rato.

Por último, o sistema Teddy não permite, ao contrário do nosso, criar vários objectos simultaneamente, nem dispõe de operações booleanas que permitam combinar dois objectos num só.

Igarashi et. al. descrevem uma técnica de design geométrico 2D, designada por embelezamento interactivo [Igarashi97], que transforma os traços desenhados à mão livre pelo utilizador em segmentos de recta que satisfazem determinadas restrições geométricas, tais como perpendicularidade, congruência e simetria, entre outras. Esta técnica foi concretizada num protótipo designado por Pegasus, o qual lida com as ambiguidades próprias dos traços desenhados à mão livre, gerando múltiplos candidatos, após o que o utilizador selecciona o candidato pretendido com um toque da caneta sobre o mesmo.

O principal problema desta técnica tem a ver com o facto, referido pelos autores, de ser difícil ao utilizador seleccionar o segmento pretendido de entre os restantes candidatos, dado estes, normalmente, se sobreporem mutuamente.

No caso do sistema GIDeS, e não obstante o enquadramento ser diferente, a solução já abordada de lidar com as ambiguidades mediante o recurso a uma barra de ícones dependentes do contexto, não sofre deste problema.

3. GIDES: SITUAÇÃO ACTUAL

Embora baseado em trabalho anterior na área da modelação (sistema IDeS [Branco94]), GIDeS é um sistema novo, que se distingue do anterior em vários aspectos, dos quais se destaca a utilização de um interface caligráfico em substituição do interface baseado em menus do sistema IDeS. O referido interface é complementado por um mecanismo de ícones dependentes do contexto, tal como já foi referido acima.

3.1 Processamento dos Traços do Utilizador

Uma vez que o estilete usado na interacção com o sistema GIDeS não tem quaisquer botões, não é possível, à partida, determinar se os traços desenhados pelo utilizador constituem simples elementos de desenho ou, pelo contrário, comandos ou primitivas 3D. Isto tem como consequência que o processamento dos referidos traços seja mais complexo do que é habitual noutros interfaces gestuais. A Fig. 2 ilustra a estrutura básica deste processo.

Sempre que o utilizador desenha um traço, este começa por sofrer um pré-processamento que tem como objectivo

a eliminação do ruído do mesmo. O traço é, em seguida, passado ao reconhecedor de gestos de comando o qual, em caso de sucesso, invoca a execução do comando correspondente. Caso não tenha ocorrido nenhum reconhecimento, o sistema assume que o traço é um elemento de desenho e procede à inclusão do mesmo no grafo de desenho.

Em seguida, o sub-grafo constituído pelo traço em questão e por todos os traços que lhe estão ligados é objecto de uma análise topológica e geométrica, levada a cabo pelo reconhecedor de primitivas 3D. Caso ocorra o reconhecimento de uma ou mais destas primitivas, o sistema procede à construção de uma barra de ícones representativas das referidas primitivas, a qual aparece no ecrã, junto ao cursor. Se o utilizador optar por seleccionar um dos ícones, procede-se à eliminação do grafo de todos os traços que constam do sub-grafo em análise, e à construção e colocação da primitiva seleccionada na cena 3D.

Na realidade, a complexidade do processo de reconhecimento dos gestos de comando é acrescida pelo facto de serem permitidos gestos compostos por mais do que um traço. Isto obriga à existência no sistema de um *buffer* de armazenamento de traços pendentes, de uma acção por omissão e de uma lista de expectativas.

A necessidade do *buffer* de armazenamento de traços pendentes advém do facto de, por vezes, a reacção do sistema estar condicionada pela acção futura do utilizador, isto é, o sistema só poder tomar uma decisão depois de o utilizador desenhar mais um traço.

A acção por omissão é executada pelo sistema sempre que a acção do utilizador não corresponde à expectativa do sistema. Inicialmente, a acção por omissão consiste em interpretar o traço do utilizador como um elemento de desenho e não como um traço constituinte de um gesto de comando. Esta situação poderá ser alterada, como se pode ver mais adiante.

A lista de expectativas reflecte, em cada instante, a expectativa do sistema face à acção futura do utilizador, na assunção de que este está a efectuar um gesto de comando. Inicialmente, a lista de expectativas é composta por todos os traços constituintes de todos os gestos de comando definidos no sistema. À medida que vão chegando mais traços, a lista de expectativas vai sendo actualizada em conformidade, até que ocorra uma de três situações possíveis:

- A lista fique vazia: neste caso, o sistema opta por efectuar a acção por omissão.
- A lista fique reduzida a um único traço, o qual é parte integrante de um único gesto de comando: neste caso, o sistema opta por executar o comando associado ao referido gesto.

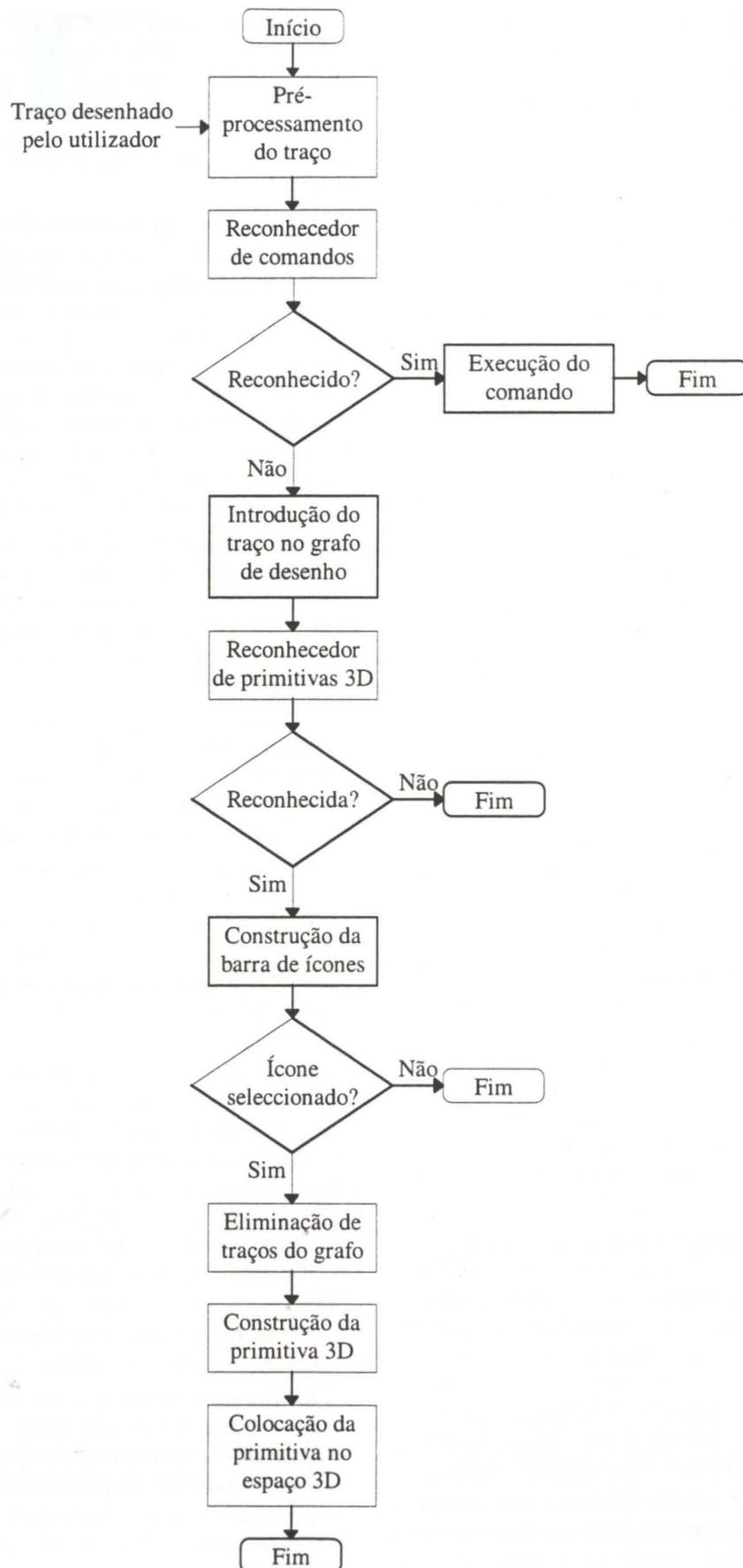


Figura 2: Processamento dos traços desenhados pelo utilizador

- A lista fique reduzida a um único traço, o qual é parte integrante de dois ou mais gestos de comando: neste caso existe uma situação de ambiguidade e, à semelhança do que acontece no reconhecimento de primitivas 3D, é criada uma barra de ícones representativos dos comandos em causa. Se o utilizador seleccionar um dos ícones, o comando que lhe está associado é executado; caso contrário, é executada a acção por omissão.

Sempre que o sistema tomar uma decisão, a acção por omissão e a lista de expectativas retomam os seus valores iniciais e o ciclo repete-se.

Para melhor ilustrar o funcionamento de todo este processo, vamos supor que existiam apenas quatro gestos de comando:

- Um gesto composto por dois traços, designados por A e B.
- Um gesto composto por um único traço (B).
- Um gesto composto por um único traço (C).
- Um gesto composto por um único traço (D).

Note-se que o traço B é parte integrante de dois gestos de comando (AB e B).

Tal como foi dito, a acção por omissão consiste, inicialmente, em interpretar os traços do utilizador como elementos de desenho, e a lista de expectativas é composta por todos os traços constituintes de todos os gestos de comando (neste exemplo, os traços A, B, C e D). Sempre que o utilizador desenha um traço, este vai ser comparado com os traços que constam da lista de expectativas. Consideremos três cenários distintos:

1. O utilizador começa por desenha o traço A. Neste caso, o sistema fica na expectativa de o utilizador prosseguir com o desenho do traço B (isto é, a lista de expectativas passa a ser composta apenas pelo traço B). Se o utilizador corresponder à expectativa do sistema (isto é, se desenha o traço B), o comando associado ao gesto AB é executado. Caso contrário, o sistema opta pela acção por omissão, isto é, interpreta o traço A como um elemento de desenho.
2. O utilizador começa por desenha o traço B. Neste caso, a acção por omissão passa a ser a execução do comando associado ao gesto B e a lista de expectativas passa a ser composta apenas pelo traço A. Se o utilizador prosseguir, então, com o desenho do traço A, o comando associado ao gesto AB é executado. Caso contrário, o sistema opta pela acção por omissão, isto é, executa o comando associado ao gesto B.
3. O utilizador desenha um traço ambíguo, o qual foi reconhecido simultaneamente como sendo o traço C e o traço D. Neste caso, o sistema opta por apresentar ao utilizador uma barra composta pelos ícones representativos dos comandos associados aos gestos C e D. Se o utilizador optar por seleccionar um destes ícones, o comando correspondente é executado. Caso

contrário, o traço é considerado um elemento de desenho (acção por omissão).

3.2 Reconhecedor de Gestos de Comando

O módulo de reconhecimento de gestos de comando é, na realidade, composto por dois reconhecedores distintos, que se complementam mutuamente.

O primeiro é uma versão aperfeiçoada do reconhecedor de Rubine [Rubine91]. As alterações introduzidas visam três aspectos que consideramos importantes:

- Permitir o reconhecimento de gestos compostos por mais do que um traço. A sequência pela qual o utilizador desenha os diversos traços é irrelevante.
- Tornar o reconhecimento insensível ao sentido dos traços desenhados pelo utilizador.
- Lidar com ambiguidades.

Rubine propõe um conjunto de 11 características estáticas (de cariz geométrico) e 2 características dinâmicas (de cariz temporal) opcionais, usadas na classificação dos gestos do utilizador. No sistema GIDeS, tomamos em linha de conta apenas as características geométricas, de modo a que o reconhecimento não seja influenciado pela velocidade de desenho do utilizador. Além disso, para que o reconhecedor possa lidar com gestos compostos por mais do que um traço, acrescentamos um segundo nível de características geométricas, as quais consistem, para cada gesto, em pares de grandezas que representam as distâncias (na horizontal e na vertical) entre os pontos médios dos rectângulos envolventes dos traços que compõem o gesto, tomados dois a dois (Fig. 3).

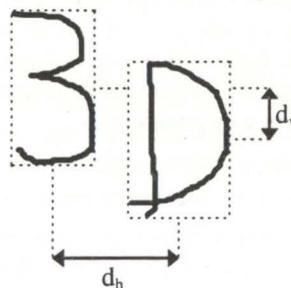


Figura 3: Distâncias entre pares de traços

Foram, também, feitas pequenas modificações nas equações estatísticas propostas por Rubine. As alterações prendem-se com a necessidade de tomar em consideração todos os traços que compõem cada classe gestual.

Seja C o número de classes gestuais, S_c o número de traços que compõem a classe gestual c , E_c o número de amostras de treino da classe gestual c , e F o número de características usadas na identificação dos traços.

Seja f_{cesi} a i -ésima característica do s -ésimo traço da e -ésima amostra de treino da classe gestual c .

O vector médio de características para cada traço de cada classe é dado pela equação:

$$\bar{f}_{csi} = \frac{1}{E_c} \sum_{e=0}^{E_c-1} f_{cesi}$$

$$0 \leq c < C, 0 \leq s < S_c, 1 \leq i \leq F$$

A matriz de covariância para cada traço de cada classe é dada pela equação:

$$\Sigma_{csij} = \sum_{e=0}^{E_c-1} (f_{cesi} - \bar{f}_{csi})(f_{cesj} - \bar{f}_{csj})$$

$$0 \leq c < C, 0 \leq s < S_c, 1 \leq i, j \leq F$$

A matriz de covariância comum é dada pela equação:

$$\Sigma_{ij} = \frac{\sum_{c=0}^{C-1} \sum_{s=0}^{S_c-1} \Sigma_{csij}}{\sum_{c=0}^{C-1} (E_c - 1) S_c}$$

$$1 \leq i, j \leq F$$

Os pesos de cada traço de cada classe são dados pelas equações:

$$w_{csj} = \sum_{i=1}^F (\Sigma^{-1})_{ij} \bar{f}_{csi}$$

$$w_{cs0} = -\frac{1}{2} \sum_{i=1}^F w_{csi} \bar{f}_{csi}$$

$$0 \leq c < C, 0 \leq s < S_c, 1 \leq j \leq F$$

Para tornar o reconhecimento insensível ao sentido de desenho dos traços tenta-se, em primeiro lugar, classificar cada traço no sentido desenhado pelo utilizador. Se a classificação falhar, inverte-se o sentido do traço (isto é, o primeiro ponto do traço passa a ser o último, o segundo passa a ser o penúltimo, e assim sucessivamente), após o que se tenta novamente a classificação.

Para lidar com ambiguidades, o processo de Rubine de classificação de gestos teve de sofrer algumas alterações.

Rubine começa por usar uma equação que permite calcular, para cada classe gestual c , uma grandeza designada por v_c . A classificação do gesto g do utilizador consiste simplesmente na determinação do c para o qual v_c é máximo. Em seguida, calcula uma estimativa $P(c|g)$ da probabilidade de g ter sido classificado correctamente, bem como uma grandeza δ^2 , designada por distância de Mahalanobis [Duda73]. O gesto será rejeitado se $P(c|g) < 0.95$ ou se $\delta^2 > F^2/2$.

No nosso caso, começamos por calcular para cada traço s de cada classe c a distância de Mahalanobis, dada pela equação:

$$\delta_{cs}^2 = \sum_{i=1}^F \sum_{j=1}^F (\Sigma^{-1})_{ij} (f_i - \bar{f}_{csi})(f_j - \bar{f}_{csj})$$

$$0 \leq c < C, 0 \leq s < S_c$$

eliminando como candidatos todos os traços para os quais $\delta_{cs}^2 > F^2/2$. Em seguida, calculamos para todos os candidatos que restam a grandeza v_{cs} dada pela equação:

$$v_{cs} = w_{cs0} + \sum_{i=1}^F w_{csi} f_i$$

$$0 \leq c < C, 0 \leq s < S_c$$

Com base nestas grandezas, calculamos para cada um dos candidatos uma estimativa $P(cs|g)$ da probabilidade de o referido candidato corresponder ao gesto g do utilizador:

$$P(cs|g) = \frac{1}{\sum_{i=0}^{C-1} \sum_{j=0}^{S_i-1} e^{(v_{ij}-v_{cs})}}$$

$$0 \leq c < C, 0 \leq s < S_c$$

Com base nestas estimativas de probabilidades, a aceitação ou rejeição dos traços candidatos é feita de acordo com o seguinte critério:

- Se existir um candidato com probabilidade $P \geq 0.95$, o candidato é aceite e todos os outros são rejeitados (neste caso, não há lugar a qualquer ambiguidade).
- Se tal não acontecer, e se existirem dois candidatos, cada um dos quais com probabilidade $P \geq 0.95/2$, os dois candidatos são aceites (ambiguidade) e os restantes rejeitados.
- Se tal não acontecer, e se existirem três candidatos, cada um dos quais com probabilidade $P \geq 0.95/3$, os três candidatos são aceites (ambiguidade, mais uma vez) e os restantes são rejeitados.
- E assim sucessivamente.

O segundo reconhecedor serve de complemento à versão melhorada do reconhecedor de Rubine. Cada gesto tem um algoritmo próprio de reconhecimento, o que permite, para cada caso, tornar o reconhecedor insensível ou não às transformações geométricas, bem como classificar gestos dotados de características geométricas variáveis.

Por exemplo, o gesto de apagar consiste num traço em ziguezague, composto por três ou mais segmentos de recta (o número pode variar consoante a vontade e/ou a necessidade do utilizador), com qualquer orientação e quaisquer dimensões. Estas peculiaridades implicam que o gesto não seja susceptível de ser classificado pela versão aperfeiçoada do reconhecedor de Rubine.

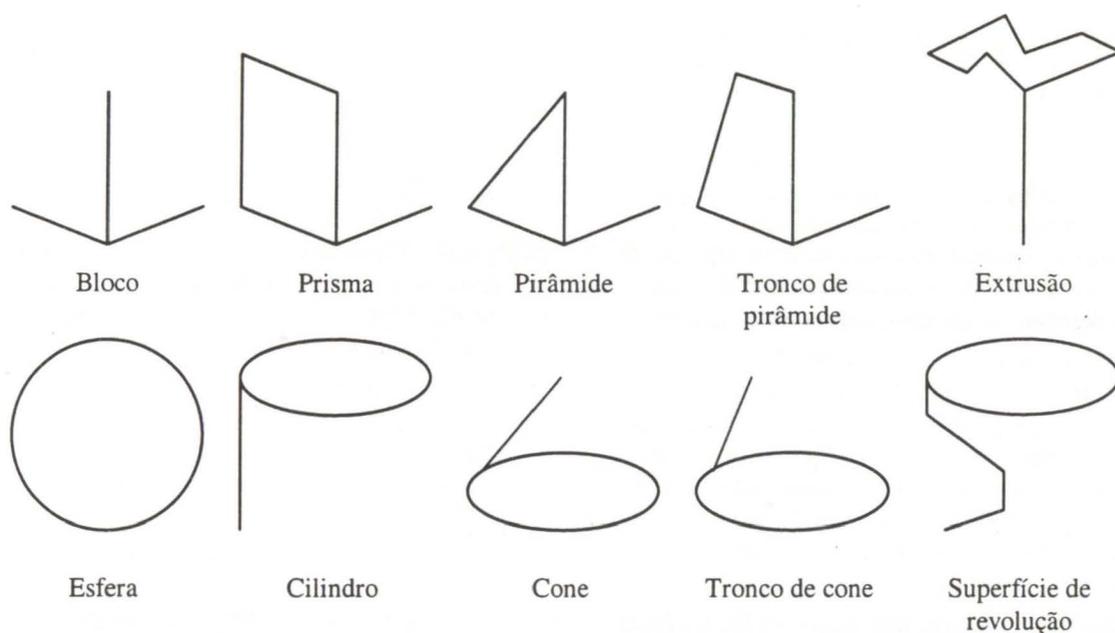


Figura 4: Primitivas 3D

3.3 Reconhecedor de Primitivas 3D

O módulo de reconhecimento de primitivas 3D funciona a dois níveis.

O primeiro nível consiste numa análise topológica de uma parte do grafo de desenho do sistema. Sempre que o utilizador desenha um novo traço, o sub-grafo composto por este traço e por todos os que lhe estão ligados é objecto de uma análise topológica, a qual tenta classificar o desenho numa de três categorias (Fig. 4):

- Junção trivalente (caso do bloco).
- Linha fechada (caso da esfera).
- Linha fechada ligada a uma linha aberta composta por um ou mais segmentos de recta (caso da superfície de revolução), ou por um e um só segmento de recta (todos os restantes casos).

Uma vez encontrada a categoria a que pertence o desenho, entra em acção o segundo nível de reconhecimento. Este procede a uma análise geométrica do desenho, de modo a refinar a classificação e a encontrar a primitiva (ou primitivas, em caso de ambiguidade) que se enquadram no mesmo.

Por exemplo, para que o cilindro, o cone e o tronco de cone sejam reconhecidos, é necessário que a linha fechada acima referida seja uma elipse. No caso do cilindro, o segmento de recta ligado à elipse terá de ser perpendicular (no espaço 3D) ao plano da mesma. Se tal não acontecer, a extremidade livre do segmento terá (no caso do cone) ou não (no caso do tronco de cone) de estar próxima do eixo perpendicular ao plano da elipse que passa pelo centro da mesma.

4. CONCLUSÕES E TRABALHO FUTURO

A avaliação preliminar da usabilidade do sistema GIDeS, nomeadamente do seu interface baseado no reconhecimento de gestos e em ícones dependentes do contexto, foi realizada junto de uma pequena amostra de utilizadores

potenciais, composta por seis designers e arquitectos, os quais utilizam, normalmente, sistemas CAD 3D nas suas actividades profissionais. A avaliação centrou-se em dois atributos de usabilidade: desempenho inicial e primeira impressão. A preferência por estes dois atributos em detrimento de outros, prende-se, sobretudo, com o facto de estes parecerem ser os mais adequados no sentido de nos fornecerem pistas que conduzam ao aperfeiçoamento do sistema.

A avaliação do desempenho inicial consistiu em medir o tempo que os diversos participantes levaram a realizar um conjunto de sete tarefas de modelação, recorrendo quer a um sistema CAD comercial, quer ao protótipo do sistema GIDeS. Os resultados obtidos são bastante animadores, na medida em que o tempo médio gasto na execução das diversas tarefas foi, nalguns casos, sensivelmente idêntico para ambos os sistemas e, noutros, claramente favorável ao sistema GIDeS.

A avaliação da primeira impressão consistiu num inquérito feito por duas vezes aos participantes: a primeira vez após uma demonstração do sistema (antes de qualquer contacto com o mesmo) e a segunda após o primeiro contacto. As respostas a esse inquérito podiam variar entre -3 (caso mais desfavorável) e +3 (caso mais favorável). O valor médio obtido após o primeiro contacto (+2) revela que a opinião geral do sistema por parte dos utilizadores é bastante favorável.

Com base nas sugestões feitas pelos elementos do grupo de avaliação, estamos a considerar algumas modificações ao nosso trabalho, designadamente ao nível do reconhecimento de elipses projectadas noutros planos que não o horizontal, o que permitirá a construção de primitivas 3D já orientadas de acordo com a vontade do utilizador, dispensando, deste modo, quaisquer operações subsequentes de rotação dos objectos.

Também estamos a investigar uma maneira alternativa de o sistema apresentar sugestões de primitivas ao utilizador: em vez de uma barra de ícones, o sistema constrói imediatamente a primitiva mais provável, com base num registo das opções anteriores do utilizador. Este poderá aceitar a primitiva ou, tocando simplesmente com a ponta do estilete na mesma, percorrer uma sequência de primitivas alternativas. Quando esta sequência se esgotar, um toque adicional com a ponta do estilete leva ao reaparecimento do desenho inicial, após o que o ciclo se repete.

Em 1990, Jim Blinn escreveu [Blinn90]: “*There is a tool that works perfectly fine for the ideation phase of creation. I know it might be heretical to say this, but the ultimate creative design tool is: Paper and Pencil*”. Após descrever as vantagens do papel e lápis, Blinn conclui: “*The combination of paper and pencil works ... and I don't see computer graphics replacing it. AND THAT'S OK. I'm not being funny here*”.

Consideramos estas palavras não como um facto inelutável, mas como um desafio e um guia que nos ajuda a alcançar o objectivo de encontrar uma ferramenta computacional que seja adequada às fases criativas do design geométrico.

5. REFERÊNCIAS

- [Blinn90] Blinn J F: Jim Blinn's Corner - The Ultimate Design Tool, *Computer Graphics & Applications, IEEE*, Vol. 10, No. 11, pp. 90 - 92, 1990.
- [Branco94] Branco V, Ferreira F N, Costa A: Sketching 3D models with 2D interaction devices, *EUROGRAPHICS '94* (Conference Proceedings), M Daehlen, L Kjeldahl (eds.), Oslo, Blackwell Pub., pp. 489 - 502, 1994.
- [Duda73] Duda R, Hart P: Pattern Classification and Scene Analysis, Wiley Interscience, 1973.
- [Encarnação99] Encarnação L M, Bimber O, Schmalstieg D, Chandler S D: A Translucent Sketchpad for the Virtual Table Exploring Motion-based Gesture Recognition, *Computer Graphics Forum*, Vol. 18, No. 3, pp. C-277 - C-285, 1999.
- [Forsberg97] Forsberg A S, LaViola Jr. J J, Markosian L, Zeleznik R C: Seamless Interaction in Virtual Reality, *Computer Graphics & Applications, IEEE*, Vol. 17, No. 6, pp. 6 - 9, 1997.
- [Galyean91] Galyean T A, Hughes J F: Sculpting: An Interactive Volumetric Modeling Technique, *SIGGRAPH '91* (Conference Proceedings), ACM, Vol. 25, No. 4, pp. 267 - 274, 1991.
- [Igarashi97] Igarashi T, Matsuoka S, Kawachiya S, Tanaka H: Interactive Beautification: A Technique for Rapid Geometric Design, Proceedings of the ACM Symposium on User Interface Software Technology (UIST), 1997.
- [Igarashi99] Igarashi T, Matsuoka S, Tanaka H: Teddy: A Sketching Interface for 3D Freeform Design, *SIGGRAPH '99* (Conference Proceedings), ACM, 1999.
- [Jones92] Jones J C: Design Methods, 2nd edition, New York, Van Nostrand Reinhold, 1992.
- [Jorge94] Jorge J A: Parsing Adjacency Grammars for Calligraphic Interfaces, PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York, 1994.
- [Potter94] Potter C D: A Select Few Push the Edge of Design, *Computer Graphics World*, PennWell Publishing, Vol. 17, No. 4, pp. 20 - 28, 1994.
- [Rubine91] Rubine D: Specifying Gestures by Example, *SIGGRAPH '91* (Conference Proceedings), ACM, Vol. 25, No. 4, pp. 329 - 337, 1991.
- [Sachs91] Sachs E, Roberts A, Stoops D: 3-Draw: A Tool for Designing 3D Shapes, *Computer Graphics & Applications, IEEE*, Vol. 11, No. 6, pp. 18 - 26, 1991.
- [Zeleznik96] Zeleznik R C, Herndon K P, Hughes J F: SKETCH: An Interface for Sketching 3D Scenes, *SIGGRAPH '96* (Conference Proceedings), ACM, Vol. 30, No. 4, pp. 163 - 170, 1996.