

Uma Ferramenta para Vetorização Baseada em X Window

ROGÉRIO ALVES DE BARROS
ARNALDO DE ALBUQUERQUE ARAÚJO

UFMG—Universidade Federal de Minas Gerais
DCC—Departamento de Ciência da Computação
Caixa Postal 702
30161-970 Belo Horizonte, MG, Brasil
rogerio@dcc.ufmg.br

Abstract. This work describes a vectorization process of maps of the state of Minas Gerais, Brazil, which will be used in the graphical user interface of an integrated supervision system (SIS) for telecommunication networks.

1 Introdução

Nesta comunicação é descrito o processo de captura e geração dos mapas a serem utilizados no projeto SIS¹ (Sistema Integrado de Supervisão) [Nogueira(1991)], [Meira(1991)].

O projeto SIS é um sistema em desenvolvimento pelo Departamento de Ciência da Computação da Universidade Federal de Minas Gerais em conjunto com a TELEMIG (Telecomunicações de Minas Gerais S/A), com o objetivo de executar tele-supervisão de redes de telecomunicações. O sistema está sendo desenvolvido para estações de trabalho (*workstations*) gráficas operando sob o sistema *X Window*.

Foi desenvolvido um aplicativo para fazer a vetorização dos mapas gerando segmentos de retas que delimitam os contornos das regiões de atuação da TELEMIG.

Este aplicativo foi desenvolvido para o ambiente *X Window* utilizando o conjunto de *widgets* da *OSF/Motif*.

2 Sistema X Window

X Window é um sistema gráfico multitarefa baseado em janelas, com as características de ser um sistema distribuído e independente do hardware.

No mundo das estações de trabalho, interfaces baseadas em janelas têm estado presente há alguns anos e com o uso, quase que universal, do sistema operacional Unix combinado com a necessidade de compartilhar dados em redes, têm criado fortes pressões para a sua padronização. O objetivo tem sido obter redes transparentes e dispositivos independentes, permitindo que um programa sendo executado em uma rede de estações de trabalho, crie janelas na

tela de outras estações, podendo estas ser de outros fabricantes.

Widgets são componentes utilizados na tela de uma aplicação, tais como: *labels*, *pushbuttons*, *scroll bars* etc. Neste trabalho está sendo utilizado o conjunto de *widgets* da *Open Software Foundation* — *OSF/Motif*. *Motif* pode ser definido como:

- interface gráfica
- guia de estilo para criar aplicações consistentes
- gerenciador de janelas
- uma biblioteca com um conjunto de ferramentas de auxílio ao programador

Hoje coexistem, internacionalmente, duas linhas de interfaces gráficas que operam sobre o sistema *X Window*: *Open Look* e *Motif*. Dentre os fatos que motivaram a escolha do *Motif*, podem ser citados:

- um grande número de fabricantes de hardware fornecem o *Motif*, tais como: *IBM*, *DEC* e *HP*, enquanto que o único fornecedor do *Open Look* é a *SUN*.
- *Motif* provê interfaces padronizadas e um consistente *look and feel*. Portanto o usuário terá mais facilidade para utilizar novas aplicações desenvolvidas em *Motif*.
- *Motif* possui uma biblioteca de alto nível orientada a objetos, permitindo o desenvolvimento de programas gráficos extremamente complexos com uma pequena quantidade de código fonte.
- Pesquisa realizada [Burgardi(1991)] mostra que entre usuários, projetistas de *hardware* e *software* e analistas, *Motif* está predominando. Apesar disso, todos concordam com uma coexistência dos dois sistemas por vários anos.

¹Trabalho financiado pelo convênio TELEMIG - DCC/UFMG

3 Ferramenta para Manipulação de Objetos

FIG (*Facility for Interactive Generation of figures*) é uma ferramenta que permite ao usuário desenhar e manipular objetos interativamente. Existem várias versões que rodam sob os ambientes de janela *Suntools/Sunview* e *X11* (versão 11 do *X Window*).

FIG foi desenvolvido por Supoj Sutanthavibul da Universidade do Texas, Austin e modificado e expandido por Frank Shmuck da Universidade de Cornell.

Para o projeto SIS, foi determinado que os mapas seriam representados através de segmentos de retas, os quais definiriam os contornos do mapa.

Foi escolhido o formato FIG para representar estes segmentos de reta, por se tratar de um *software* de domínio público e disponível para os ambientes *Sunview* e *X11*. Além do aplicativo FIG, existe uma série de utilitários que permite, a partir de um arquivo no formato FIG, gerar um arquivo nos formatos *L^AT_EX*, *bitmap* para *X11*, *PostScript*, entre outros.

O aplicativo FIG poderá também ser utilizado para fazer alterações no mapa após os processos de digitalização e vetorização, que serão discutidos nas próximas seções.

A Figura 1 mostra a tela do *software* FIG, que é composta de duas colunas verticais com opções e uma área de desenho.

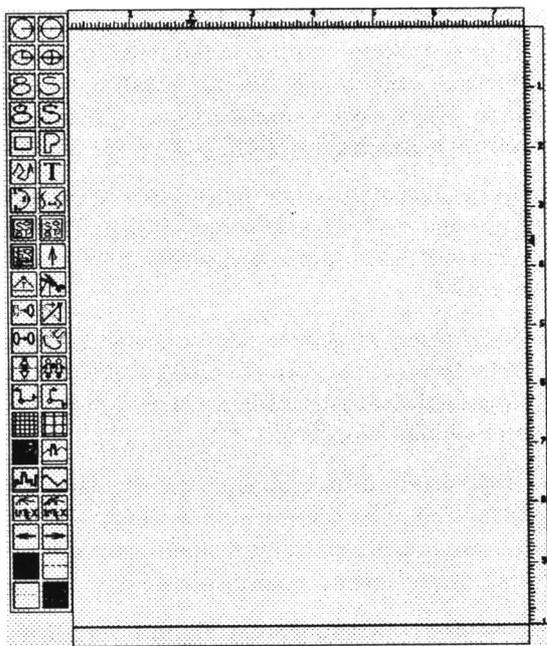


Figura 1: FIG - *Facility for Interactive Generation of figures*

4 Digitalização

O processo de digitalização consiste em fazer uma amostragem do nível de cinza ou cor de uma imagem. Desde que a imagem contém valores contínuos de intensidade, deve-se em seguida fazer uma quantização dos pontos da imagem, obtendo-se uma matriz com valores discretos que representam a cor ou o nível de cinza destes pontos, que são chamados de *pixels*.

Para o projeto SIS, o processo de digitalização tem como objetivo obter imagens digitais do mapa do estado de Minas Gerais, de suas regiões e sub-regiões, para em seguida serem submetidas ao processo de vetorização.

Foi utilizado um *scanner* da Howtek (*Personal Color Scanner*) [Howtek(1990)a] para o processo de digitalização dos mapas. O *scanner* trabalha em arquiteturas IBM-PC e acompanha um *software* para a captura de imagens [Howtek(1990)b]. As características deste *scanner* são:

- resolução variável de 50 a 600 dpi
- digitalização em nível de cinza, cor ou espalhamento (*dithering*)
- transmissão de dados assíncrona
- realce da qualidade da imagem
- área de digitalização: 8,5x14 polegadas
- captura da imagem usando 24, 8 ou 1 bit por pixel
- possibilidade de salvar a imagem em vários formatos

O *scanner* é de tamanho A4, o que limita o tamanho dos mapas a serem digitalizados. No caso do projeto SIS foi feita uma redução xerográfica dos mapas para a adaptação ao tamanho A4. Esta redução não trouxe perdas significativas da qualidade dos mapas a serem digitalizados já que, para o processo de vetorização, todo o interesse está nos contornos das regiões.

5 Vetorização

Com o processo de digitalização, são obtidas imagens digitais dos mapas do estado. Foi desenvolvido um aplicativo para fazer a vetorização destes mapas, com o objetivo de obter os contornos no formato de segmentos de retas definido pelo FIG.

O aplicativo denominado *VECTOR* faz de forma semi-automática esta vetorização já que, o usuário terá que definir os segmentos que irão compor os contornos dos mapas das regiões de atuação da TELEMIG.

Esta vetorização não pode ser feita de uma forma totalmente automática, pois nem sempre estes contornos são limites de município, não havendo, nestes casos, bordas para serem detectadas numa vetorização automática. Estes problemas decorrem do fato de não existirem mapas com as divisões utilizadas pelo SIS, mas apenas mapas políticos com divisões de municípios.

VECTOR foi desenvolvido para o ambiente *X Window* [Jhonson(1990)] e [Nye(1990)a], utilizando os *widgets* do *Motif* [Nye(1990)b] e [Jhonson(1991)]. VECTOR tem como parâmetros de entrada o nome de um arquivo contendo a imagem digitalizada no formato BM (*bitmap*) definido em [Davis(1992)] e o nome do arquivo FIG que será gerado.

A Figura 2 mostra o *layout* dos *widgets* utilizados pelo VECTOR. A Figura 3 mostra a tela do utilitário VECTOR.

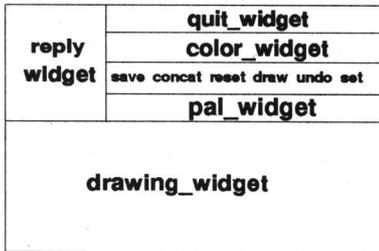


Figura 2: VECTOR - *Layout* dos *Widgets*

Para o processo de vetorização, o usuário deve definir manualmente os contornos das regiões e sub-regiões. A definição de uma região é feita pelos contornos das suas sub-regiões. Na Figura 4 é mostrado um exemplo de uma região com duas sub-regiões. A sub-região norte é formada pelos conjuntos de segmentos 1 e 2, sendo que o conjunto 1 possui três segmentos e o 2 possui cinco segmentos. A segunda sub-região (sul) é composta também, por dois conjuntos de segmentos: 2 e 3, sendo que o conjunto 2 possui cinco segmentos e o 3 possui sete segmentos.

Neste exemplo o conjunto de segmentos 2 será utilizado para formar as duas sub-regiões. O usuário deverá então, desenhar estes três conjuntos de segmentos e depois fazer a concatenação do conjunto 1 com 2 e salvar como sendo a primeira sub-região e depois fazer a concatenação do conjunto 2 com o 3 e salvar como sendo a segunda sub-região.

A hierarquia de *widgets* do programa VECTOR é formada pelos seguintes *widgets*:

- *form_widget* - Pai dos *widgets*
- *quit_widget* - *Widget* do tipo *PushButton*, utiliza-

comunicações

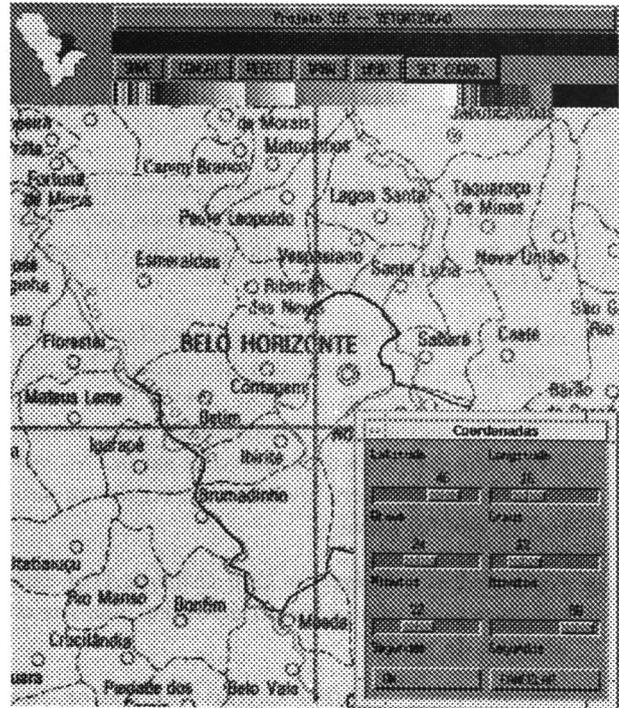


Figura 3: VECTOR - Aplicativo para vetorização dos mapas

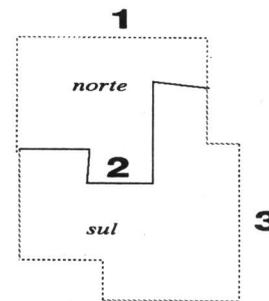


Figura 4: Exemplo de vetorização de uma região com duas sub-regiões

zado para encerrar o programa. Neste *widget* foi colocado o nome do projeto e o objetivo do aplicativo (Projeto SIS - Vetorização).

- *drawing_widget* - Área na qual o mapa digitalizado será mostrado. O usuário irá *clicar* um dos botões do *mouse* sobre os contornos do mapa e serão desenhados, sobre o mapa, os segmentos de reta na cor definida pelo usuário através do *widget pal_widget*. Esta cor está mostrada no *color_widget*.
- *color_widget* - *Widget* do tipo *DrawingArea*, mostra a cor corrente que será utilizada para desenhar os segmentos de reta que irão compor os contornos dos mapas.
- *save_widget* - *Widget* do tipo *PushButton*, salva segmentos de reta em um arquivo no formato FIG. Cada vez que o botão *SAVE* for pressionado, os segmentos concatenados pelo usuário serão armazenados no arquivo de saída. No *widget reply_widget* será mostrado a região ou sub-região que foi salva.
- *reply_widget* - *Widget* do tipo *DrawingArea*, mostra as regiões que já foram definidas e salvas no arquivo de saída.
- *concat_widget* - *Widget* do tipo *PushButton*, que ao ser pressionado o cursor do *mouse* muda de forma, devendo o usuário *clicar* o botão do *mouse* sobre os segmentos anteriormente definidos por ele, estabelecendo os conjuntos de segmentos que formarão os contornos das regiões ou sub-regiões.
- *reset_widget* - *Widget* do tipo *PushButton*, reinicia o conjunto de segmentos concatenados. Este botão deve ser utilizado quando o usuário fizer a concatenação errada de algum segmento
- *draw_widget* - *Widget* do tipo *PushButton*, reinicia as variáveis de contexto para que o usuário possa definir um novo conjunto de segmentos no mapa. O cursor do *mouse* passará a ter forma de um lápis.
- *undo_widget* - *Widget* do tipo *PushButton*, remove o último conjunto de segmentos de reta definido pelo usuário.
- *pal_widget* - Mostra a tabela de cores do sistema (paleta). Se o botão do *mouse* for pressionado sobre uma cor, variáveis de contexto serão iniciadas para que o usuário possa definir um novo conjunto de segmentos de reta. Estes próximos segmentos serão exibidos nesta cor e esta cor será mostrada no *widget color_widget*.

- *set_widget* - *Widget* do tipo *PushButton*, que ao ser pressionado a forma do cursor irá ser alterada, passando a ser uma cruz. O usuário deverá pressionar o botão sobre uma posição do mapa e em seguida, será mostrada uma janela para que sejam especificadas as coordenadas deste ponto. Estas coordenadas devem ser em graus, minutos e segundos para a latitude e longitude. Esta informação servirá para que o sistema possa fazer uma correlação entre coordenadas de tela com coordenadas do mundo.

6 Conclusão

Uma próxima etapa será a modificação do software de vetorização adaptando-o para aceitar imagens *bit-maps* em formatos mais universais tais como GIF, PCX ou TIFF.

Uma versão final do utilitário VECTOR será responsável pelo processo de reconfiguração dos mapas das regiões e sub-regiões do sistema SIS.

7 Referências

- M. Burgard, "Who's Winning The GUI Race?", Unix-World, August, 1991.
- C. A. Jr. Davis, "PixelWare - Um Sistema de Tratamento de Imagens", Tese de Mestrado, Departamento de Ciência da Computação - Universidade Federal de Minas Gerais, Fevereiro 1992.
- E. F. Jhonson and K. Reichard, "Advanced X Window Applications Programming", MIS:Press, Portland, 1990.
- E. F. Jhonson and K. Reichard, "Power programming ... MOTIF", MIS:Press, Portland, 1991.
- A. Nye, "Xlib Programming Manual - Version 11, Release 4", O'Reilly & Associates, 1990.
- A. Nye and T. O'Reilly, "X Toolkit Intrinsic Programming Manual - OSF/Motif Edition", O'Reilly & Associates, 1990.
- Howtek Inc, "Personal Color Scanner - User's Guide", Hudson, 1990.
- Howtek Inc, "Scan-It for the Personal Color Scanner - User's Guide", Hudson, 1990.
- D. M. Meira, J. E. R. Dantas e R. S. Bigonha, "Especificação Funcional do Sistema Integrado de Supervisão", RT SIS 1101, UFMG/TELEMIG, Belo Horizonte, 1991.
- J. M. S. Nogueira e D. M. Meira, "Definição da Arquitetura Básica do SIS", RT SIS 2102, UFMG - TELEMIG, Belo Horizonte, 1991.