

GUSTAVO MOREIRA PIERRE

**AMBIENTE INTEGRADO
PARA POSICIONAMENTO EM OPERAÇÕES MILITARES**

DISSERTAÇÃO DE MESTRADO

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 01 de Julho de 2002.

GUSTAVO MOREIRA PIERRE

**AMBIENTE INTEGRADO
PARA POSICIONAMENTO EM OPERAÇÕES MILITARES**

Dissertação apresentada ao Departamento de
Informática da PUC-Rio como parte dos re-
quisitos para a obtenção do título de Mestre
em Ciências em Informática.

Orientadores:

Marcelo Gattass

Roberto de Beauclair Seixas

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 01 de Julho de 2002.

Para minha esposa Alexandra que amo tanto e que sempre me apoiou em tudo.

Para os meus filhos Alexandre, Carolina, Felipe e Gustavinho que, só por existirem, me dão forças para enfrentar toda e qualquer adversidade.

Para o meu grande amigo Ari, pela constante e fundamental ajuda.

Agradecimentos

Inicialmente ao meu orientador, Professor Marcelo Gattass, pela confiança depositada, fundamental para a conclusão deste trabalho.

Ao meu também orientador, Roberto de Beauclair Seixas, pelo apoio e conhecimentos transmitidos.

Aos amigos do Tecgraf/PUC-Rio e do Visgraf/IMPA pelo apoio incessante e também pelos momentos de descontração mesmo nos momentos mais difíceis. Em especial aos amigos Ari, Paula, Maurício, Wilfredo, Caco, Cláudio, Fábio Marcos e Paulinha que viveram comigo esses dois anos de muito trabalho, mas também muitas alegrias.

Aos amigos Adriano e Wagner por toda ajuda antes mesmo do início do mestrado.

A minha tia Lídia e ao Sidney que, ajudando Alexandra e as crianças, sem perceberem, me davam a tranquilidade necessária para trabalhar.

A minha mãe e aos meus irmãos que sempre me deram forças para crescer.

A Marinha do Brasil pela oportunidade inigualável e há tanto tempo esperada. E a Diretoria de Administração da Marinha pela acolhida durante este período onde apenas usufrui a infra-estrutura. Em especial a Alayde e a Comandante Ieda pelos avisos e orientações com relação à resolução de problemas administrativos que surgiram.

Resumo

Com o crescimento tecnológico temos visto uma popularização crescente de equipamentos de localização por satélite (receptores GPS) e computadores de bolso (PDA) em diversas áreas: nos transportes aéreo, marítimo e terrestre, mapeamento e geoprocessamento, esporte e lazer. Onde podemos ver, como exemplo, praticantes de caminhadas adquirindo seus próprios receptores GPS. No meio militar, esta popularização também crescente, onde as Forças Armadas de diversos países tendem a adquirir cada vez mais equipamentos de posicionamento por satélite (receptores GPS) e testam equipamentos para interação com os mesmos.

O foco deste trabalho a integração do receptor GPS e PDA visando sua utilização em operações militares, com intuito de obter respostas rápidas para um emprego mais eficiente e rápido de frações das tropas envolvidas. Este trabalho envolve a geração de arquivos legíveis aos PDA, a comunicação de receptores GPS com estes, bem como a visualização da informação fornecida por estes receptores de forma fácil e rápida. Abrange também uma comparação entre a utilização de mapas em duas diferentes representações raster e vetorial, possibilitando zoom e movimento da imagem em ambos, além da possibilidade de medir distâncias entre pontos do mapa na tela do PDA e inclusão, exclusão e alteração de observações diversas.

Verificou-se a grande vantagem da utilização de imagens raster em relação utilização de imagens vetoriais no PDA para esta finalidade, e também algumas fragilidades do uso de PDA comuns em operações militares e com receptores GPS de usos gerais.

Abstract

With technological development, we have been witnessing a growing popularization of satellite-location equipment (GPS receivers) and pocket computers (PDA) in several areas: air, sea and earth transports, mapping and geoprocessing, sports and entertainment. We can see hikers, for example, acquiring their own GPS receivers. In the military environment, this popularization is also occurring: the Armed Forces in several countries are obtaining increasing numbers of GPS receivers and are testing equipment to interact with them.

The purpose of this work is to integrate GPS receivers and PDAs, with the intention of employing them in military operations in order to obtain quick responses for a faster and more efficient use of fractions of involved troops. This work involves the generation of files readable by PDAs, the communication of GPS receivers with PDAs, and an easy and fast visualization of the information provided by the receivers. It also includes a comparison between the use of maps in two different representations, raster and vector, both allowing for zooming and movement of the image, as well as the possibility of measuring distances between points of the map in the PDA screen, and the inclusion, exclusion and alteration of several observations.

The great advantage of using raster images instead of vector images in PDAs for this purpose was verified, as well as some disadvantages in the use of PDAs, common in military operations, with general-purpose GPS receivers.

Sumário

Lista de Figuras	viii
Lista de Tabelas	x
1 Introdução	1
1.1 Visão Geral	1
1.2 Objetivos	2
1.3 Trabalhos Relacionados	2
1.4 Estrutura do Trabalho	3
2 Conceitos Básicos	4
2.1 Operações Militares	4
2.2 Formatos das Imagens	6
2.2.1 Formato Raster	6
2.2.2 Formato Vetorial	7
2.3 PDA	8
2.3.1 Diferenças da Programação	9
2.3.2 Conceitos de Programação para Palm	11
2.3.3 Ambiente de Programação	12
2.3.4 Memória	12

2.3.5	Recursos	13
2.3.6	Banco de Dados	13
2.3.7	Imagens	15
2.3.8	Comunicação Serial	18
2.4	GPS	18
2.4.1	Definição	18
2.4.2	Precisão	19
2.4.3	Segmentos do GPS	20
2.4.4	Protocolos	21
2.4.5	Outros Sistemas de Posicionamento por Satélite	21
2.5	Sistemas de Informação Geográfica	22
2.5.1	<i>Datum</i>	22
2.5.2	Escala	23
2.5.3	Sistemas de Coordenadas	23
2.5.4	Projeções	24
2.5.5	A projeção UTM	24
3	Utilização de Dispositivos Portáteis para Posicionamento em Operações Militares	26
3.1	Soluções	26
3.1.1	Hardware	26
3.1.2	Software	28
3.2	Emprego	54
4	Resultados	56
5	Conclusão e Trabalhos Futuros	61

5.1	Conclusão	61
5.2	Trabalhos Futuros	62
	Referências Bibliográficas	63

Lista de Figuras

2.1	Distribuição dos planos e calcos.	5
2.2	Mapa e calco juntos.	5
2.3	Situação atual.	6
2.4	Comparação entre utilização do PDA e do PC(fonte Palm Inc.)	9
2.5	Formato de armazenamento do BD no PC.	14
2.6	Formato lógico do BD no PC.	14
3.1	Receptor GPS conectado a um PDA.	28
3.2	Exemplo de um arquivo de projeto PRJ.	30
3.3	Geração dos arquivos PDB, a partir de um arquivo PRJ.	31
3.4	Registros em um arquivo PDB e a ordem de seus campos.	32
3.5	Divisão de uma imagem em blocos.	33
3.6	Divisão de uma imagem em blocos com faixas de imagens em comum com os blocos vizinhos.	34
3.7	Divisão de uma imagem em blocos de 160 x 160 pixels.	35
3.8	Tamanho do arquivo PRC gerado em relação ao modo de divisão da imagem.	35
3.9	Comparação entre utilização da API do PalmOS e de outra imagem para a operação de zoom	36
3.10	Exemplo de um arquivo de projeto PRB.	37
3.11	Localização relativa dos arquivos bitmaps de um projeto PRB.	37

3.12	Distribuição das informações no recurso <i>appinfostr</i>	38
3.13	Geração do arquivo PRC.	39
3.14	Indexação dos blocos de duas imagens de um mesmo PRC, onde a primeira imagem é a mais ampla e a outra é o primeiro nível de zoom.	40
3.15	As funções dos botões do Palm.	41
3.16	Tela do emulador com o mapa e a barra de status.	42
3.17	Fluxo das operações por ocasião de uma operação de zoom.	46
3.18	Proporção entre a janela possível de desenhar e a janela útil.	48
3.19	Representação da forma como é realizado o zoom com a imagem vetorial	48
3.20	Zoom com uma imagem vetorial	49
3.21	Divisão de uma imagem vetorial em blocos.	50
3.22	Representação da forma como é realizado o movimento com a imagem vetorial	50
3.23	Imagem do aplicativo no modo GPS, recebendo informações.	51
3.24	Tela do Palm no modo de observação.	54
4.1	Tempo para carregamento e tempo para desenho da imagem (vetorial x <i>raster</i>).	57
4.2	Tempo para carregamento e tempo para desenho da imagem (pixel a pixel x recursos.	58
4.3	Utilização de diversos tipos de imagens.	59
4.4	Conexão entre o Palm e o seu cabo serial.	60

Lista de Tabelas

2.1	Estrutura usada para definir um objeto	8
2.2	Estruturas do cabeçalho do recurso bitmap	17
2.3	Estruturas da tabela de cores do recurso bitmap	17
3.1	Estrutura usada para definir um objeto no arquivo PDB	30
3.2	Estrutura usada para definir o primeiro registro no arquivo PDB	31
3.3	Campos da estrutura de status	42

Capítulo 1

Introdução

1.1 Visão Geral

O emprego de equipamentos portáteis nas mais diversas áreas tem presenciado um crescimento vertiginoso, principalmente em função do grande crescimento tecnológico. Hoje, podemos observar que muitos dos praticantes de esportes como ciclismo, caminhadas, vôo livre, parapente, entre outros, têm seus próprios equipamentos portáteis para avaliação e, principalmente, localização. Podemos também observar a utilização cada vez mais comum de tais equipamentos nos transportes terrestres, marítimos e aéreos, em atividades de mapeamento e geoprocessamento e, como dito antes, lazer e esporte.

Nos últimos dez anos, vimos uma popularização crescente deste tipo de equipamento também no meio militar, onde os quartéis tendem a adquirir cada vez mais equipamentos de posicionamento por satélite (receptores GPS) e testam equipamentos para interação com os mesmos.

Em 1991, em função do receio de dificuldade de localização no deserto por parte das tropas que participariam da Guerra do Golfo, e pela escassez de equipamento para tal finalidade nos quartéis, as Forças Armadas dos Estados Unidos fizeram uma grande aquisição de receptores GPS [Cra01]. Até então estes eram muito utilizados nas operações navais e para guiar mísseis de médio e longo alcance.

Durante a ofensiva americana no Afeganistão em 2001, mais da metade dos 300 militares americanos a bordo do destróier USS McFault, ancorado no Oceano Índico para lançar mísseis contra aquele país, levaram consigo seus computadores de bolso (PDA). Estes estavam sendo usados pelos marinheiros para baixar e-mails e acessar tarefas do dia em um dos 32 dispositivos com leitores infravermelhos instalados pelos corredores do navio. E, por estarem sendo usados

cada vez mais como armas táticas e de logística em tempos de guerra moderna, as Forças Armadas americanas têm equipado seus contingentes também com modelos mais resistentes à altas temperaturas, poeira e quedas, e que trazem programas que possibilitam analisar a posição dos inimigos, seguir pessoas e realizar pesquisas sobre impacto do calor. O USS McFaul foi designado como plataforma oficial de testes para uso de dispositivos móveis em situações de guerra, onde aplicativos navais foram criados somente para aquela missão [Mes01].

O foco deste trabalho é a integração do receptor GPS e PDA visando sua utilização em operações militares, com intuito de obter respostas rápidas para um emprego mais eficiente e rápido de frações das tropas envolvidas. Este trabalho envolve a geração de arquivos legíveis aos PDA, a comunicação de receptores GPS com estes, bem como a visualização da informação fornecida por estes receptores de forma fácil e rápida. Abrange também uma comparação entre a utilização de mapas em duas diferentes representações *raster* e vetorial, possibilitando zoom e movimento da imagem em ambos, além da possibilidade de medir distâncias entre pontos do mapa na tela do PDA e inclusão, exclusão e alteração de observações diversas.

1.2 Objetivos

O objetivo principal deste trabalho é um modelo de ambiente integrado, envolvendo hardware e software, para posicionamento em operações militares. As metas deste trabalho são:

- Geração, no computador pessoal(PC), de um arquivo contendo o mapa *raster* ou vetorial e outras informações, que possa ser manuseado pelo PDA;
- Análise do formato da imagem mais adequado para manuseio pelo PDA, se *raster* ou vetorial, em relação à transferência, armazenamento e operação, levando em consideração as limitações de armazenamento e performance do PDA;
- Acompanhamento, no mapa, da posição fornecida pelo receptor GPS; e
- Possibilidade de interação do usuário com o PDA para, por exemplo, fazer observações e medir distâncias.

1.3 Trabalhos Relacionados

A tecnologia dos portáteis está em franca ascensão, o que faz surgir constantemente novos softwares. Muitos destes não têm uso militar. Os que possuem características militares

normalmente acompanham algum equipamento que é comercializado, normalmente, a preço elevado.

Existem vários softwares que implementam a interação de um mapa com um receptor GPS. Existe um software brasileiro excelente que é o GPSTrackmaker [Jún02], que opera tanto com imagens vetoriais como com imagens *raster* georeferenciadas, permitindo uma comunicação com várias marcas de receptores GPS, possibilitando a transferência de dados gerados no receptor GPS para o computador e vice-versa. O software é realmente muito bom, mas não opera com PDA, apenas computadores (desktop ou laptop).

O estudo comparativo sobre utilização de mapas vetoriais ou *raster* no palmtop tem como referência marcante o trabalho realizado por Chunyu Zhang [Zha01], mas com a limitação de não possibilitar a interação com o receptor GPS e de se limitar às pequenas áreas.

O trabalho de Steve Mann e Ray Rischpater [MR01] ignora a utilização de mapas de tamanho superior aos 400x400 pixels, que facilita consideravelmente o trabalho, mas não permite a utilização de um mapa envolvendo toda uma área de operações. Por exemplo, um mapa na escala de 1/50.000, abrangendo uma área de operações de dimensões de 20 x 20 quilômetros, apenas cerca de 10% dele pode ser visto, com o devido detalhamento, numa área de 400 x 400 pixels. Isso não diminuiu a excelência do trabalho destes no que se refere a uma série de conceitos apresentados e técnicas de zoom e movimentação de mapas no PDA.

1.4 Estrutura do Trabalho

Seguindo a proposta do trabalho, no Capítulo 2 serão fornecidas informações básicas sobre alguns temas fundamentais para o entendimento do problema. No Capítulo 3 serão abordados os métodos utilizados, tanto em relação ao hardware como ao software, utilizados para alcançar a solução apresentada. Em seguida, o Capítulo 4 apresenta alguns resultados de testes comparativos entre a utilização de imagens vetoriais e *raster* no PDA suas vantagens e dificuldades. Finalmente, no Capítulo 5, são apresentadas sugestões para trabalhos futuros e uma conclusão geral.

Capítulo 2

Conceitos Básicos

Para melhor compreensão dos termos, conceitos e assuntos tratados neste trabalho, existe a necessidade de um embasamento teórico em operações militares, imagens *raster* e vetorial, PDA, GPS e, por fim, Sistema de Informações Geográficas.

2.1 Operações Militares

A atuação de uma Força Militar numa área bem definida, chamada Área de Operações, requer um planejamento militar minucioso e flexível, que necessita de um controle das frações (também chamadas de peças de manobra) que serão projetadas nesta Área.

“Comando e Controle (C2) denota os processos e os meios utilizados pelo comandante, apoiado por seu estado-maior e outras organizações, para planejar, conduzir, coordenar e controlar forças, a fim de assegurar o cumprimento da missão. O exercício do comando e controle é desenvolvido através do emprego de pessoal, equipamento, comunicações, instalações e procedimentos que permitem a execução das funções acima assinaladas” [Har00].

O comando e controle proporcionado aos comandantes permite a manutenção da unidade de esforços para aplicar suas forças nos momentos e lugares críticos, a fim de alcançarem a vitória. Essa eficiência é buscada através de duas características constantes ao longo do tempo: o elemento humano e a necessidade de informações relevantes, precisas e oportunas. Desta forma, o comando e controle é tanto um problema de desenvolvimento de difíceis e complexas tarefas de combate quanto de gerenciamento de informações.

Durante o planejamento são gerados e distribuídos planos detalhados de operações que especificam as missões dos diversos escalões da Força envolvida. A Figura 2.1 representa grafi-

camente este fluxo. Dentro desses planos existem os calcos (Figura 2.2) que nada mais são do que desenhos baseados em símbolos gráficos padronizados que, referenciados (amarrados) a um mapa da Área de Operações, fornecem informações diversas, tais como: objetivos, obstáculos conhecidos, setores de atuação, posições e escalões conhecidos das tropas inimigas, entre outras.



Figura 2.1: Distribuição dos planos e calcos.

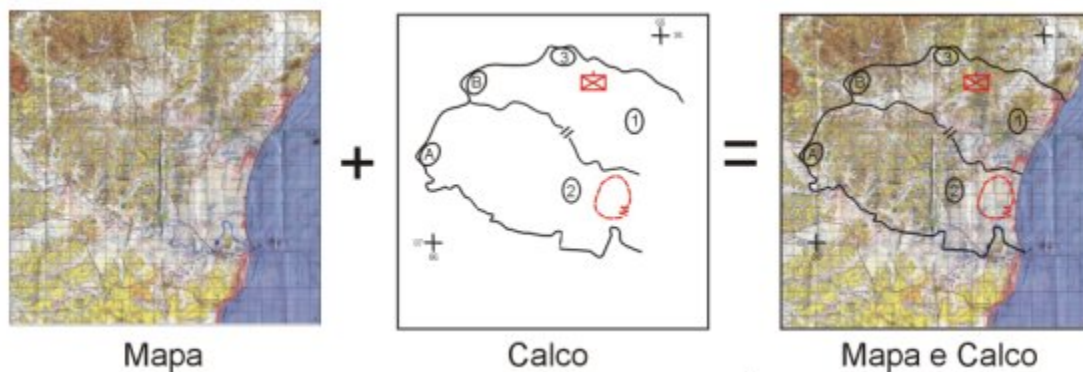


Figura 2.2: Mapa e calco juntos.

O controle das frações se faz através do acompanhamento das diversas informações, precisas e oportunas, recebidas no decorrer da manobra, tais como posição, obstáculos naturais ou artificiais, inimigo, engajamento e outras administrativas. É de extrema importância também a transmissão de informações relevantes para as peças de manobra no decorrer das operações. Essas informações, hoje, são passadas através de mensagens, padronizadas ou não, transmitidas por meios diversos (rádio, fio, mensageiro, etc). Também através dessas mensagens as peças de manobra solicitam apoio de fogo ou necessidades administrativas como, por exemplo, repletamento de pessoal. É, portanto, importante para a peça de manobra, entre outras coisas, obter seu posicionamento de forma rápida e precisa e visualizá-la no mapa da Área de Operações, comparando-a ao posicionamento do inimigo, de tropas amigas, de obstáculos conhecidos, bem como sua situação em relação ao cumprimento da missão.

Atualmente, as frações em terra fazem a verificação de seu posicionamento através de receptor GPS ou ainda pelo método tradicional no qual a posição é obtida com auxílio de uma bússola, através do cruzamento de marcações de pontos notáveis no terreno. Este posicionamento, obtido com auxílio do receptor GPS ou pelo método tradicional, é identificado

na carta da Área de Operações na escala que interessa àquela fração e comparada às outras informações existentes nos calcos. Este processo consome um tempo que precisa ser otimizado, pois reflete na utilização dessas frações em lugar e tempo oportunos, e é neste ponto que este trabalho centra. A Figura 2.3 representa graficamente este processo.



Figura 2.3: Situação atual.

2.2 Formatos das Imagens

2.2.1 Formato Raster

O formato da imagem *raster* utilizado é do tipo windows bitmap. Este formato pode representar imagens com 1, 4, 8, 24 ou 32 bits de profundidade de cor. A utilização de compactação RLE(*run-length encoded*) neste formato é bastante comum [Red97].

De acordo com os trabalhos de James D. Murray e William VanRyper [MV96], Paulo Marques [Mar97] e Martin Reddy [Red97], o arquivo no formato window bitmap, sem compactação e com profundidade de oito bits por pixel, é dividido em quatro partes, fundamentais à geração do bitmap no PDA:

- Cabeçalho do arquivo bitmap, que contém informações do tipo (BM), tamanho e início do *bitmapdata*.
- Cabeçalho de informação do bitmap, que define dimensões, tipo de compressão e formato da cor do bitmap.

- Tabela de cores, que associa um índice para cada combinação das cores vermelha, verde e azul, num limite de 256 cores.
- Vetor de bytes, o chamado *bitmapdata*. Cada byte representa um pixel da imagem e o vetor representa consecutivas linhas do bitmap (*scan lines*). Cada linha consiste de consecutivos pixels da esquerda para a direita. Estas linhas são representadas de baixo para cima, ou seja, a primeira linha corresponde à última do bitmap e assim por diante.

É importante ressaltar que para ser um formato independente do dispositivo que irá exibi-lo é definido que cada linha do bitmap possua um número de pixel (bytes) múltiplo de quatro, portanto, se a largura do bitmap não for de tamanho múltiplo de quatro o que falta é preenchido com bytes que serão desprezados quando for exibido o bitmap. Isso também ocorre com o formato bitmap no PDA.

2.2.2 Formato Vetorial

O formato de imagem vetorial adotado é o *shapefile* da *Environmental Systems Research Institute, Inc. (ESRI)*, utilizado atualmente no Sistema de Jogos Didáticos da Marinha do Brasil. Isso não impede que seja, futuramente, estendido a outros formatos vetoriais.

Segundo a especificação do formato *shapefile* [ESR98], este formato arquiva geometria não topológica e atributos de informações para uma feição espacial em um conjunto de dados. A geometria para uma feição é arquivada como uma figura compreendida de um conjunto de coordenadas de vetor. Os elementos gráficos suportadas pelo formato *shapefile* são: ponto, linha e área (polígonos). Os atributos são manipulados em um arquivo no formato dBASE com um relacionamento de um-para-um com o associado registro da figura.

O formato *shapefile* compreende três arquivos: o principal (.shp), o de índice (.shx) e o arquivo dBASE (.dbf), todos com mesmo nome (8 caracteres). O principal contém registros de tamanho variável onde cada registro descreve um elemento gráfico com uma lista de seus vértices. É o arquivo mais importante na geração do seu correspondente no seu PDA. No arquivo de índice, cada registro contém o *offset* do registro correspondente do arquivo principal em relação ao início dele. O arquivo dBASE contém atributos das feições com um registro por feição.

É possível representar polígonos vazados, isto é, com buracos, conforme a ordem em que seus pontos são fornecidos, mas para fins de simplificação serão utilizados somente polígonos simples, sem levar em consideração a possível ocorrência de buracos. Outra simplificação é a não utilização dos atributos existentes no arquivo dBASE relacionado. De acordo com a espe-

cificação do *shapefile* [ESR98] e também Frank Warmerdam [War99], a Tabela 2.1 apresenta a estrutura, com os campos relevantes a este trabalho, usada para definir um objeto.

Campo	Definição
nSHPType	Tipo da forma
nShapeId	Identificador do objeto
nParts	Total de partes do objeto
panPartStart[nParts]	Vetor de índice de início das partes
panPartType[nParts]	Vetor de tipos das partes (normalmente o mesmo para cada objeto)
nVertices	Total de vértices do objeto
padfX[nVertices]	Vetor de abscissas
padfY[nVertices]	Vetor de ordenadas
dfXMin	Menor valor em X da caixa envolvente(<i>bounding box</i>)
dfYMin	Menor valor em Y da caixa envolvente(<i>bounding box</i>)
dfXMax	Mair valor em X da caixa envolvente(<i>bounding box</i>)
dfYMax	Maior valor em Y da caixa envolvente(<i>bounding box</i>)

Tabela 2.1: Estrutura usada para definir um objeto

2.3 PDA

Personal Digital Assistant (PDA) é um termo cunhado por John Sculey, da Apple, para descrever o Newton, primeiro *handheld* sem teclado a ser vendido em escala. Com o tempo passou a designar todos os aparelhos similares.

Nesta seção falaremos da diferença da programação, conceitos de programação, ambiente de programação, memória, recursos, banco de dados, imagens e comunicação serial nos PDA da Palm, Inc. Os motivos de sua adoção serão apresentados no capítulo seguinte. A partir de agora utilizaremos o termo Palm no lugar de PDA.

2.3.1 Diferenças da Programação

Programar para Palm é bem diferente de programar para um desktop, pois os usuários interagem de forma diferente [BFH⁺01a]. Essas diferenças são importantes durante todo o desenvolvimento dos aplicativos para Palm. As principais diferenças são:

1. Tamanho da tela - A tela do Palm tem dimensões de 160x160 pixels, logo, a quantidade de informação que se pode exibir de uma só vez é limitada. Por esta razão se deve projetar a interface cuidadosamente com diferentes prioridades e metas do que se estivesse usando grandes telas. Deve esforçar-se para prover suficiente informação na tela sem sobrecarregá-la.
2. Padrão de Uso - Difere consideravelmente de padrão de uso dos desktops. As pessoas tendem a sentar e usar um desktop ou laptop por poucas longas sessões, usando teclado, tela grande e disco rígido para criar e editar grande quantidade de informação. Ao contrário, as pessoas usam o Palm com freqüência e em curtas sessões. Elas pegam o Palm em seus bolsos para, por exemplo, checar alguma informação ou ver um telefone e guardá-lo em seguida [BCC⁺01]. As Figuras 2.4(a) e 2.4(b) apresentam gráficos comparativos entre o uso do Palm e o uso do laptop.

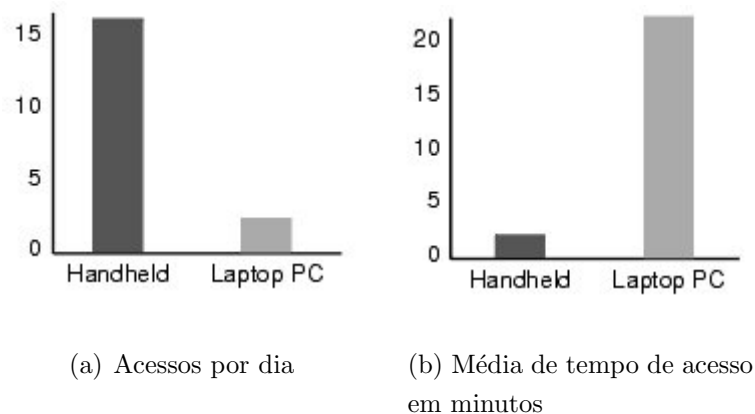


Figura 2.4: Comparação entre utilização do PDA e do PC(fonte Palm Inc.)

3. Conectividade com um PC (computador pessoal) - é um componente integral do Palm. Este vem com um berço (base) que se conecta ao PC que, com o software neste, permite realizar backup e sincronização de todos os dados do Palm, inclusive instalação de softwares compatíveis. O software de conexão é responsável por traduzir o arquivo PDB, PRC ou PQA no PC para o formato interno utilizado no Palm.
4. Métodos de Entrada de Dados - o Palm não possui teclado, nem mouse. A entrada de dados é usando normalmente a caneta, que pode ser usada com um software chamado

Graffiti que usa um padrão para cada letra a ser desenhada, ou com uma janela semelhante a um teclado na tela. Existem modelos de pequenos teclado externos para Palm, mas são vendidos separadamente. No Palm não existe o conveniente de o teclado e o mouse existente no PC, mesmo porque o usuário não deve entrar muitos dados no Palm.

5. Consumo de Energia - [Fos00] a fonte de energia para fazer funcionar um Palm são baterias, normalmente recarregáveis. O gerenciamento do consumo da energia disponível é constante. Ela pode durar semanas com operação normal, o que é impressionante, se considerar que o Palm nunca desliga realmente, pois está sempre fornecendo energia para importantes sistemas. Essa preocupação com consumo de energia não ocorre com um PC, já que ele está ligado na rede elétrica. Pressionar o botão de liga e desliga significa alternar em modos de baixo consumo e modo mais ativo. O relógio, a memória e o circuito de geração de interrupção necessitam de alguma pequena quantidade de energia constantemente para funcionar. Isto é particularmente importante no caso da memória, pois o Palm armazena os aplicativos em memória RAM, que seria perdida se deixasse de receber energia. O Palm suporta três modos de operação:
 - (a) "Dormindo" (*sleep mode*) - é quando o usuário "desliga" o Palm. Isto pode ocorrer quando o usuário pressiona o botão "on/off" ou automaticamente no caso de não estar sendo usado num período configurável de 1 a 3 minutos. Somente sistemas essenciais ficam energizados (alguns circuitos de interrupção, relógio e memória RAM). Caso ele esteja recebendo dados de um receptor GPS conectado a ele, esta conexão será perdida.
 - (b) "Cochilando" (*doze mode*) - é quando o Palm está ligado, mas não está executando instruções.
 - (c) "Executando" (*running mode*) - neste modo, o processador está executando instruções.
6. Memória - Os Palms têm limitado espaço de memória. Variam de 512k, nos Palms mais antigos, até 16M, no modelo recém-lançado. Devido a esta limitação de memória, a otimização é algo crítico. Para fazer um aplicativo tão rápido e eficiente quanto possível, é necessário otimizar primeiro a utilização da memória, em seguida preocupa-se com a rapidez e, por último preocupa-se com o tamanho do código.
7. Sistema de Arquivo - devido ao limitado espaço de armazenamento e a fim de permitir uma sincronização com o desktop mais eficiente, o Palm não usa o tradicional sistema de arquivos. Os dados são arquivados em pedaços da memória chamados registros que são agrupados em banco de dados. Este banco de dados é análogo a um arquivo. A diferença é que os dados são divididos em múltiplos registros ao invés de serem armazenados em pedaços contínuos de memória.

8. Compatibilidade com versões anteriores do Sistema Operacional - existem vários modelos de Palm atualmente, muitos deles possuem sistemas operacionais (PalmOS) diferentes. O usuários não precisam fazer atualizações de seus PalmOS, a medida que vão sendo lançadas novas versões, pois elas são desenvolvidas de modo que mantenham compatibilidade com versões anteriores, ou seja, os aplicativos que rodam em versões anteriores do PalmOS funcionam nas versões mais novas. Para que isso ocorra é importante que o programador siga as orientações constantes no trabalho de Christopher Bey *et al.* [BFH⁺01b], que sugere que muitas estruturas apresentadas não sejam acessadas diretamente.

2.3.2 Conceitos de Programação para Palm

Os aplicativos para Palm são geralmente programas não multitarefa e orientados a eventos. Apenas um programa roda de cada vez. Para construir aplicativos no Palm com sucesso é necessário conhecer como o sistema é estruturado e como estruturar os aplicativos.

- Cada aplicativo tem uma função *PilotMain*, que é equivalente à função *main* em programas em C. Para executar um programa, o sistema chama a *PilotMain* e envia a ele um código de lançamento. O código de lançamento pode especificar que o aplicativo deve tornar-se ativo e exibir uma interface (código de lançamento normal - *normal launch*), ou ele pode especificar que o aplicativo deve executar uma pequena tarefa e sair sem exibir qualquer interface. Esta é a única proposta da função *PilotMain* - receber um código de lançamento e responder a ele.
- O PalmOS é orientado a eventos, desta forma os aplicativos para Palm têm um loop de eventos, que só é iniciado em resposta a um código de lançamento normal. O aplicativo pode também executar tarefas fora deste loop, em resposta a outros códigos de lançamentos.
- A maioria dos aplicativos para Palm faz uso do elemento formulário (*form*) para construção de interfaces, como nos aplicativos para PC. Esta interface pode conter elementos de interface pré-definido e configuráveis.
- Todas os aplicativos devem utilizar o gerenciamento de dados e de memória providos pelo sistema.

O PalmOS consiste de vários gerenciadores, que são grupos de funções que trabalham juntas para implementar uma característica de um aplicativo. Como regra, todas as funções pertencentes a um mesmo gerenciador possuem o mesmo prefixo e trabalham juntas para implementar um certo aspecto da funcionalidade.

2.3.3 Ambiente de Programação

Existe uma série de ambientes de programação para criar aplicativos para Palm. Dessas muitas ferramentas disponíveis, C e C++ são as linguagens mais comuns.

Este trabalho foi implementado em C utilizando o *CodeWarrior Development Tools for Windows*, da Metrowerks, que permite construir aplicativos em C, C++ e Java. Este pacote contém o *CodeWarrior Interactive Development Environment* (IDE) de fácil utilização.

2.3.4 Memória

O PalmOS trata muito bem do gerenciamento da memória do Palm, no entanto ele não manipula sozinho todo o espaço limitado de memória. É necessário que se tenha cuidadosa atenção no uso da memória no desenvolvimento de um aplicativo. O PalmOS é construído em torno de uma estrutura de memória de 32 bits, com tipos de dados de 8, 16 e 32 bits de comprimento.

A RAM no PalmOS é dividida em duas áreas separadas: RAM dinâmica e RAM de armazenamento. A primeira é usada da mesma forma que a RAM em um PC, provendo espaço para armazenamento temporário de variáveis globais, alocações e outros dados que não requerem persistência entre execuções de um aplicativo. A RAM de armazenamento é usada de forma semelhante ao disco rígido no PC, fornecendo uma área de armazenamento para aplicativos e dados.

Todo espaço da RAM dinâmica é usada para implementar o *heap* dinâmico. Um *heap* é uma área contígua de memória que gerencia e contém pequenos pedaços de memória. Esses pedaços são também áreas contíguas de memória de tamanho entre 1 byte e um pouco menos do que 64Kbytes.

Aplicativos podem alocar, manipular e liberar memória alocada no *heap* dinâmico usando o gerenciador de memória do PalmOS. As funções deste gerenciador de memória permitem o uso seguro da memória dinâmica do Palm, sem a preocupação de como as estruturas internas estão sendo manipuladas.

A RAM de armazenamento, que corresponde a toda área restante que não está sendo usada pela *heap* dinâmico, é dividida dentro de um número de *heaps* de armazenamento. Este número e tamanho depende da versão do PalmOS e da memória disponível. A partir da versão 3.x do PalmOS toda a RAM de armazenamento é tratada, propiciando grande melhoria na solução da fragmentação da memória que ocorre.

Os pedaços de memória no *heap* de armazenamento são chamados registros. Cada registro é parte de um banco de dados implementado pelo gerenciador de dados do PalmOS. Esse banco de dados nada mais é do que uma lista de pedaços de memória mais um cabeçalho.

O gerenciador de dados do PalmOS provê funções para criar, abrir, fechar e apagar bancos de dados, bem como funções para manipular registros dentro deles. Um banco de dados no PalmOS pode ser comparado a um arquivo no PC. Dependendo do conteúdo dos registros de um banco de dados e de seu cabeçalho, um dado banco pode representar um aplicativo, uma *library* compartilhada ou simplesmente dados armazenados de um aplicativo. Em razão da quantidade limitada de memória no Palm, um aplicativo não copia dados do heap de armazenamento para o heap dinâmico para modificá-lo. O gerenciador de memória e o de dados, do PalmOS, marcam pedaços de memória, impedindo que sejam manipulados por qualquer outro processo, e os editam na posição. O PalmOS não permite a um aplicativo alterar o conteúdo de qualquer pedaço de memória sem usar as API do gerenciador de memória e do gerenciador de dados.

2.3.5 Recursos

Um aplicativo no PalmOS é composto de recursos, que são blocos que representam dados, códigos executáveis, elementos de interface e outros pedaços de um aplicativo.

2.3.6 Banco de Dados

As explicações e conceitos que serão agora apresentados descrevem o formato de banco de dados do Palm que são armazenados em arquivos no PC. Quando um desses bancos de dados é carregado no Palm, ele é arquivado na memória em um formato que é similar, mas diferente do descrito aqui. O formato do banco na memória do Palm é sujeita a mudanças que não são documentadas pela Palm Inc. [Hil01]. Os bancos de dados armazenados na ROM do Palm são arquivados em um formato da memória que não é assunto deste trabalho.

O *Palm Database*(PDB) é o formato padrão que o PalmOS utiliza para o armazenamento de dados, em forma de registro. O *Palm Resource* (PRC) também, mas armazena dados na forma de recursos. Os formatos são semelhantes. Os aplicativos do PalmOS são bancos de dados de recursos [Ale01].

De uma forma geral, um banco de dados no Palm contém um cabeçalho e uma lista seqüencial de registros ou recursos. O formato do cabeçalho é semelhante para arquivos de banco de dados PDB, PRC e PQA, diferindo apenas no tamanho e conteúdo de alguns campos.

Registros e recursos são blocos de memória que contém os dados que se quer. Os dois são usados para finalidades diferentes:

- Registros são usados para armazenar dados de aplicativos tais como memos ou entradas no catálogo de endereços.
- Registros são usados para armazenar conteúdo web em bancos PQA.
- Recursos são usados para armazenar o código e objetos da interface para um aplicativo.
- Recursos são também usados para arquivar mensagens, cadeias de caracteres e blocos de informações do aplicativo.

Cada banco é armazenado em um arquivo no PC em formato seqüencial, conforme mostra a Figura 2.5. O formato de cada arquivo de banco de dados é logicamente estruturado segundo a Figura 2.6. Cada banco contém as seguintes partes componentes:

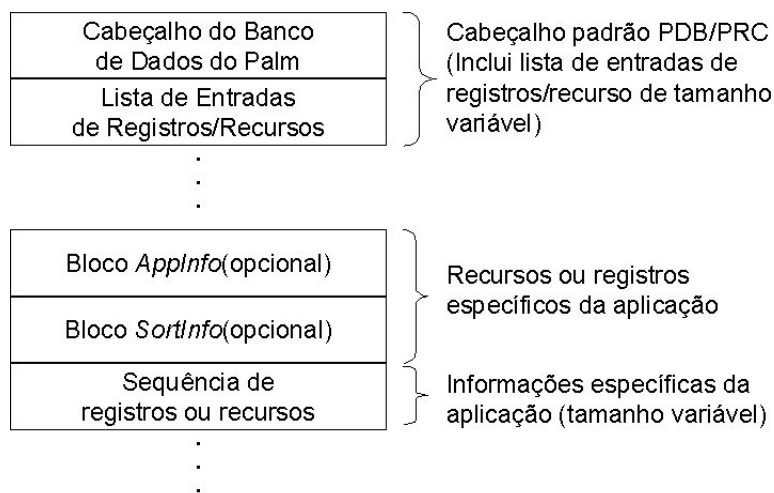


Figura 2.5: Formato de armazenamento do BD no PC.

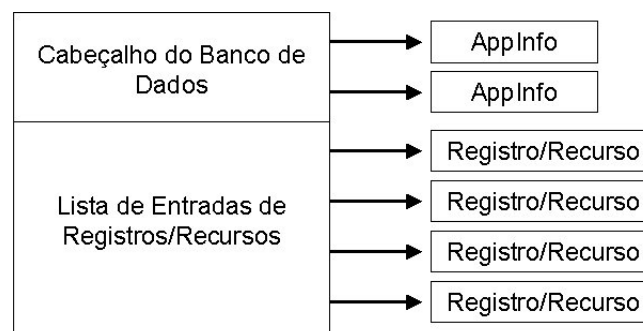


Figura 2.6: Formato lógico do BD no PC.

- Um cabeçalho do banco que descreve este, faz referência aos blocos *appinfo* e *sortinfo* e contém uma lista de entradas de registros/recursos, que define a localização de cada registro/recurso no banco.
- Um bloco opcional de informação do aplicativo(*appinfo*) que pode armazenar informações específicas para o aplicativo.
- Um bloco também opcional de informação de ordenação(*sortinfo*) no qual pode ser arquivado identificador de tabelas de referências cruzadas.
- Linhas de registros e recursos.

Entre o final da lista de entradas de registros/recursos(*record list*), no cabeçalho e o primeiro bloco de dados do banco, existe um intervalo. O primeiro bloco de dados pode ser um *appinfo*, um *sortinfo*, os recursos ou registros ou o fim do arquivo. Este intervalo tem tradicionalmente, 2 bytes de comprimento.

A lista de entradas de registros/recursos tem seu próprio cabeçalho, seguido por zero ou mais entradas de registros. Cada entrada descreve um simples registro/recurso no arquivo. A lista tem comprimento variável. Pode haver mais de uma lista de entradas de registros/recursos, mas isso raramente é usado e seu uso é desencorajado pela Palm Inc.

Cada entrada de registro/recurso refere-se a localização(*offset*) deste e contém um atributo e um identificador.

Os registros em um PDB são armazenados em blocos de registros contíguos. O *offset* do início de cada registro é armazenado na lista de registros no cabeçalho do banco de dados. O comprimento e formato do dado do registro são específicos do aplicativo. De forma similar, os recursos em um PRC é arquivado como um bloco de recursos contíguos. O *offset* do início de cada registro é também armazenado na lista de recursos no cabeçalho do banco. O comprimento e formato do dado do registro são específicos de cada registro.

Para finalizar, é importante relembrar que um registro/recurso fica contido num único pedaço de memória que não pode ultrapassar 64Kbytes. Se isso vier a ocorrer é necessário dividir o registro/recurso em quantas partes forem necessárias.

2.3.7 Imagens

No PalmOS, o gerenciador de janelas pode desenhar retângulos, pixels, linhas, caracteres e bitmaps. O desenho de uma imagem vetorial, por exemplo, se daria através do desenho de pixels (pontos), linhas e retângulos. No caso de uma imagem *raster* o PalmOS permite o

desenho pixel a pixel ou o desenho do bitmap todo. A utilização do desenho pixel a pixel é mais demorado.

A partir da versão 3.5 o PalmOS passou a suportar cores indexadas de 1, 2, 4 ou 8 bits/pixel. No PalmOS 4.0 foi introduzida as representações diretas de 16 e 24 bits pixel [BFH⁺01a].

As cores de desenho são definidas pelo chamado “estado de desenho”, que define também o padrão de desenho. A configuração do estado pode ser empilhado ou desempilhado, conforme deseja usar o novo estado ou retornar ao anterior.

No PalmOS, o bitmap é um recurso composto de um cabeçalho, uma tabela de cores e um *bitmapdata*. A Tabela 2.2 e 2.3 apresentam os campos das estruturas do cabeçalho e da tabela de cores, respectivamente. O *bitmapdata* é uma seqüência de bytes representando o índice da cor daquele pixel. Diferente do *bitmapdata* do formato windows bitmaps, a ordem do *bitmapdata* no PalmOS não é invertida, isto é, é normalmente da esquerda para a direita e de cima para baixo.

Campo	Definição
width	largura do bitmap em pixels
height	altura do bitmap em pixels
rowBytes	número de bytes para cada linha da imagem
flags	compressed(1 bit)-a compressão usada hasColorTable(1 bit) - se tem tabela de cores hasTransparency(1 bit) - se existe ou não transparência indirect(1 bit) - se falso, o bitmap é arquivado diretamente após a tabela de cores forScreen(1 bit) - Nunca marque este flag directColor(1 bit) - se verdadeiro, o bitmapa é to tipo de cores diretas(RGB), sem tabela de cores reserved(10 bits) - reservado para uso futuro
pixelSize	profundidade de cor do pixel
version	versão da codificação do bitmap usado(=2)
nextDephtOffset	no caso de família de bitmap, indica o início do próximo bitmap da família. Se não houver família de bitmap é igual a 0.
transparentIndex	se houver uma cor que representa a transparência é seu índice
compressionType	tipo da compressão utilizada
reserved	reservado(=0)

Tabela 2.2: Estruturas do cabeçalho do recurso bitmap

Campo	Definição
numEntries	número de cores na tabela
entry[]	vetor contendo: index - índice da cor r - componente vermelha g - componente verde b - componente azul

Tabela 2.3: Estruturas da tabela de cores do recurso bitmap

A compressão RLE pode ser utilizada nos recursos bitmap, o que pode diminuir o tamanho da imagem, conforme a quantidade de pixels contíguos com cores iguais.

Desenhar imagens no Palm envolve o conceito de janela. Uma janela é definida como uma região de desenho. Esta região pode estar ativa ou em buffer de memória (uma janela em *offscreen*). Janelas em *offscreen* são úteis para salvar e recuperar regiões da tela que são ocultadas por outros objetos da interface. Elas também são úteis quando se deseja realizar algum tipo de animação na imagem, como por exemplo sua movimentação (*panning*). Todos os formulários (*forms*) são janelas, mas nem todas as janelas são formulários. A janela ativa para desenho é onde o sistema desenha todas as saídas de funções gráficas. A coordenada 0,0 corresponde ao pixel no canto superior esquerdo da janela. Para desenhar em uma janela é preciso selecioná-la. A janela em *offscreen* tem um limite de dimensões em torno de 420x420 pixel.

2.3.8 Comunicação Serial

O gerenciador da serial é responsável pelo controle e entrada e saída da RS-232, *Bluetooth* ou sinais da USB. Embora o PalmOS suporte conexões *Bluetooth*, é necessário hardware e software adicionais [BFH⁺01c].

O recebimento de dados é assíncrono. Existem dois gerenciadores da porta serial, o mais antigo ainda funciona por questões de compatibilidade com determinados equipamentos. Neste trabalho foi utilizado o novo gerenciador, que difere do anterior por permitir conexão com mais de um dispositivo serial.

2.4 GPS

2.4.1 Definição

O Sistema de Posicionamento Global, na realidade denominado NAVSTAR-GPS (*Navigation Satellite with Time and Ranging*), é um sistema de radionavegação desenvolvido pelo Departamento de Defesa dos EUA, com intuito de ser o principal sistema de navegação das forças armadas norte-americanas.

A concepção do GPS permite que seu usuário, em qualquer local da superfície terrestre, ou próximo a ela, tenha a sua disposição, no mínimo, quatro satélites para serem rastreados. Isso permite que se realize um posicionamento em tempo real. Um receptor rastreia um satélite pela recepção de seu sinal. Sinais de apenas quatro satélites são necessários para obtenção de uma posição fixa tridimensional, mas é desejável um receptor que rastreie mais de quatro satélites simultaneamente. Como o usuário se desloca, o sinal de algum satélite pode ser bloqueado

repentinamente por algum obstáculo, restando satélites suficientes para orientá-lo. A maioria dos receptores rastreia de 8 a 12 satélites ao mesmo tempo. Um receptor não é melhor que outro por rastrear mais satélites. Rastrear satélites significa conhecer suas posições. Não significa que o sinal daquele satélite está sendo usado no cálculo da posição. Muitos receptores calculam a posição com quatro satélites e usam os sinais do quinto para verificar se o cálculo está correto.

O princípio básico da navegação pelo GPS consiste na medida de distâncias entre o usuário e quatro satélites. Conhecendo as coordenadas dos satélites num sistema de referência apropriado, é possível calcular as coordenadas da antena do usuário no mesmo sistema de referência dos satélites. O sistema de referência utilizado é o WGS84. O elipsóide de referência é o GRS80 [Mon00].

2.4.2 Precisão

No GPS há dois tipos de serviços, que são:

- SPS (*Standard Positioning Service* - Serviço de Posicionamento Padrão) - é um serviço de posicionamento e tempo padrão que está disponível a todos os usuários do globo, sem cobrança de qualquer taxa. A precisão deste serviço atualmente é da ordem de quinze metros.
- PPS (*Precise Positioning Service* - Serviço de Posicionamento Preciso) - proporciona melhores resultados (um metro), mas é restrito ao uso militar e a usuários autorizados.

Até maio de 2000, esta precisão do SPS era da ordem de 100 a 140m, em razão da limitação no nível de precisão determinada pelo Departamento de Defesa dos EUA e garantida pela adoção das seguintes medidas:

- AS (*Anti-Spoofing*) - processo de criptografia de um dos códigos usados pelo GPS para realizar medidas de distâncias.
- SA (*Selective Availability* - Disponibilidade Seletiva) - manipulação das mensagens de navegação e frequência dos relógios dos satélites, impedindo os receptores GPS de obterem toda a precisão capaz de ser proporcionada pelo GPS.

O contínuo desenvolvimento tecnológico passou a permitir ao Departamento de Defesa dos EUA obstruir a precisão do sistema onde e quando os interesses americanos exigissem, mesmo sem a utilização da Disponibilidade Seletiva. Recentemente essa degradação foi verificada

durante as operações militares dos EUA na região do Afeganistão e anteriormente, ainda com uso da Disponibilidade Seletiva, na Guerra do Golfo [Mon00].

Um fator que afeta a precisão é a “Geometria dos Satélites”- localização dos satélites em relação uns aos outros sob a perspectiva do receptor GPS. Se um receptor GPS estiver localizado sob quatro satélites e todos estiverem na mesma região do céu, sua geometria é pobre. Na verdade, o receptor pode não ser capaz de se localizar, pois todas as medidas de distância provém da mesma direção geral. Isto significa que a triangulação é pobre e a área comum da intersecção das medidas é muito grande, isto é, a área onde o receptor busca sua posição cobre um grande espaço. Dessa forma, mesmo que o receptor mostre uma posição, a precisão não é boa. Com os mesmos quatro satélites, se espalhados em todas as direções, a precisão melhora drasticamente. A geometria dos satélites torna-se importante quando se usa o receptor GPS próximo a edifícios ou em áreas montanhosas ou vales. Quando algum satélite é bloqueado, a posição relativa dos demais determinará a precisão, ou mesmo se a posição pode ser obtida. Um receptor de qualidade indica não apenas os satélites disponíveis, mas também onde estão no céu (azimute e elevação), permitindo ao operador saber se o sinal de um determinado satélite está sendo obstruído [Gor02].

2.4.3 Segmentos do GPS

O GPS consiste de três segmentos(componentes) principais: espacial, de controle e de usuário [JA00]. O segmento espacial consiste de 24 satélites (mais dois reservas) distribuídos em seis planos orbitais igualmente espaçados, com quatro satélites em cada plano, numa altitude de aproximadamente 20.200km. A configuração dos planos orbitais e a posição dos satélites garantem que, no mínimo, quatro satélites sejam visíveis em qualquer local da superfície terrestre, a qualquer hora.

O segmento de controle é composto por cinco estações monitoras (Hawai, Kwajalein, Ascension Island, Diego Garcia e Colorado Springs), três delas com antenas para transmitir os dados para os satélites (Ascension Island, Diego Garcia e Kwajalein) e uma estação de controle central (MCS: *Master Control Station*) localizada em Colorado Springs. Estas cinco estações, pertencentes à Força Aérea Americana, em conjunto com outras sete do NIMA (*National Imagery and Mapping Agency*), compõem as estações monitoras GPS do Departamento de Defesa norte-americano. As tarefas destas estações são:

- Monitorar e controlar continuamente o sistema de satélites;
- Determinar o sistema de tempo GPS;

- Predizer as efemérides(registros) dos satélites, calcular as correções dos relógios dos satélites; e
- Atualizar periodicamente as mensagens de navegação de cada satélites.

O segmento de usuários é constituído pelos receptores GPS, os quais devem ser apropriados para os propósitos a que se destinam.

2.4.4 Protocolos

Receptores GPS são projetados para serem compactos, não possuindo teclado alfanumérico. Todos os dados são digitados uma letra ou número ou símbolo por vez.

É possível usar o receptor associado a outro equipamento, pois a maioria dos receptores pode enviar dados para equipamentos periféricos.

O receptor GPS deve usar uma linguagem que o equipamento a ele associado possa entender. Existe uma linguagem padrão para equipamentos de navegação chamada Protocolo NMEA (*National Maritime Electronics Association*) [Ben02]. Os receptores GPS atuais operam não apenas com este protocolo, mas utilizam também protocolos proprietários de seus fabricantes. Este é o caso do receptor GPS Garmin eTrex [Gar00], adotado para o desenvolvimento deste trabalho, que opera com o Protocolo NMEA somente para envio de dados e utiliza o protocolo GARMIN [Gar99] tanto para entrada como para saída.

2.4.5 Outros Sistemas de Posicionamento por Satélite

Existem outros sistemas de posicionamento segundo João Francisco Galera Monico [Mon00]:

- GLONASS [GLO00]- é um sistema similar ao GPS, proporcionando posicionamento 3-D e velocidade, bem como informações de tempo, sob quaisquer condições climáticas, em âmbito local, regional e global. Esse sistema foi concebido no início de 1970, na antiga URSS pelo *Soviet Union's Scientific Production Association of Applied Mechanics*, e atualmente é desenvolvido e operado pela *Russian Federation Space Forces*. O segmento espacial do GLONASS era composto de uma constelação de 24 satélites mais um reserva distribuídos em 3 planos orbitais com inclinação maior que a utilizada no GPS, o que garante uma maior cobertura para altas latitudes. A manutenção do GLONASS, diferente do ocorrido com o GPS, não tem sido a esperada. Depois de ter sido declarado

operacional em janeiro de 1996, com 24 satélites em operação, o GLONASS contava com apenas 10 satélites no início de 2000.

- GALILEO - é um sistema global de navegação por satélite proposto pela Agência Espacial Européia, composta por 14 nações, concebido para garantir a sua independência estratégica face aos satélites americanos e russos. O sistema está em vias de ser abandonado. Esta ameaça se dá devido a persistência de recusa, reiterada em dezembro de 2001 pelos ministros dos Transportes da Alemanha, Reino Unido, Holanda, Áustria, Suécia e Dinamarca, de assegurar o financiamento de uma parte do projeto, considerando mais econômico recorrer ao GPS norte-americano. Os outros países, liderados pela França e Espanha, continuaram a bater-se pelo financiamento público dos 550 milhões de euros que deverão permitir o arranque do GALILEO até 2005 e a sua abertura ao setor privado. Os defensores do GALILEO argumentam que, apesar dos americanos disponibilizarem gratuitamente o seu GPS, também podem desativá-lo a todo o momento para as aplicações civis. A previsão de lançamento do primeiro satélite experimental está previsto para 2004 [ESA02].

2.5 Sistemas de Informação Geográfica

São sistemas de informação construídos especialmente para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente e indispensável para tratá-los. Dados geográficos são coletados a partir de diversas fontes e armazenados normalmente nos chamados bancos de dados geográficos.

As definições que serão agora apresentadas foram em sua grande maioria oriundas dos trabalhos de Gilberto Câmara *et al.* [CCH⁺96] [CDM⁺01].

2.5.1 *Datum*

Datum é um ponto onde a superfície do elipsóide de referência toca a Terra, sendo caracterizado a partir de uma superfície de referência (*datum* horizontal) e de uma superfície de nível (*datum* vertical). O *datum* utilizado pelo GPS é o WGS84.

2.5.2 Escala

Apesar do globo terrestre mostrar distâncias, áreas, formas e direções reais, ele não é o mais apropriado para lidar com processamentos sistemáticos. Para permitir tais processamentos, são criadas projeções num plano, ou seja, cada ponto do elipsóide ou esfera é projetado em uma superfície plana. Esta superfície - o mapa - pode ser apresentada em diferentes escalas.

Escala é a relação entre as dimensões dos elementos representados em um mapa e a grandeza correspondente, medida sobre a superfície da Terra. A escala é uma informação obrigatória para qualquer mapa e geralmente está representada de forma numérica. No nosso optou-se por exibir a escala na forma gráfica, mas de uma forma indireta, pois os valores não serão diretamente visíveis.

2.5.3 Sistemas de Coordenadas

Um objeto geográfico qualquer (como uma cidade, a foz de um rio, o cume de uma montanha) somente poderá ser localizado se puder ser descrito em relação a outros objetos cujas posições sejam previamente conhecidas, ou se tiver sua localização determinada em uma rede coerente de coordenadas. Quando se dispõe de um sistema de coordenadas fixas, pode-se definir a localização de qualquer ponto na superfície terrestre.

Os sistemas de coordenadas dividem-se em dois grandes grupos: sistemas de coordenadas geográficas ou terrestres (utilizado pelo GPS) e sistemas de coordenadas planas ou cartesianas (utilizado pelos mapas de operações militares).

No sistema de coordenadas geográficas ou terrestres, historicamente mais antigo, cada ponto da superfície terrestre é localizado na interseção de um meridiano com um paralelo. Meridianos são círculos máximos da esfera cujos planos contem o eixo dos pólos. O meridiano de origem (também conhecido como inicial ou fundamental) é aquele que passa pelo antigo observatório britânico de Greenwich. Ele é escolhido convencionalmente como a origem das longitudes sobre a superfície terrestre e como base para a contagem dos fusos horários, correspondendo ao meridiano a 0° . A leste do meridiano de origem, os meridianos são medidos por valores crescentes até $+180^\circ$. A oeste, suas medidas são decrescentes até o limite mínimo de -180° .

Paralelos são círculos da esfera cujos planos são perpendiculares ao eixo dos pólos. O Equador é o paralelo que divide a Terra em dois hemisférios: Norte e Sul. O paralelo a 0° corresponde ao Equador, 90° ao Pólo Norte e -90° ao Pólo Sul. Todos os meridianos se encontram em ambos os pólos e cruzam o equador em ângulo reto. A distância entre meridianos

diminui do Equador para os pólos. Os paralelos jamais se cruzam.

Representa-se um ponto na superfície terrestre por um valor de latitude e longitude. Longitude é a distância angular entre um ponto qualquer da superfície terrestre e o meridiano de origem. Latitude é a distância angular entre um ponto qualquer da superfície terrestre e a linha do Equador. Pontos que não correspondem à medição média dos oceanos podem ter também a altitude como terceiro parâmetro.

Como o sistema de coordenadas geográficas considera desvios angulares a partir do centro da Terra, não é um sistema conveniente para aplicações em que se buscam distâncias ou áreas. Para estes casos, utilizam-se outros sistemas de coordenadas, mais adequados, como, por exemplo, o sistema de coordenadas planas, descrito a seguir.

O sistema de coordenadas planas, também conhecido por sistema de coordenadas cartesianas, baseia-se na escolha de dois eixos perpendiculares, usualmente denominados eixos horizontal e vertical, cuja interseção é denominada origem, estabelecida como base para a localização de qualquer ponto do plano.

O protocolo NMEA0183, no qual o receptor GPS fornece as informações para o PDA, padroniza que a informação de posição é definida em coordenadas geográficas.

2.5.4 Projeções

Todos os mapas são representações aproximadas da superfície terrestre, que projetam cada ponto do globo terrestre em uma superfície plana. Para se obter essa correspondência, utiliza-se os sistemas de projeções cartográficas.

Há um grande número de projeções cartográficas, uma vez que há uma variedade de modos de projetar em um plano os objetos geográficos que caracterizam a superfície terrestre. Dependendo do que se pretende analisar no mapa, cada tipo de projeção minimiza um certo tipo de deformação, por exemplo, buscando conservação dos ângulos ou uma proporcionalidade das superfícies. A projeção utilizada nos mapas de operações terrestres é a UTM (*Universal Transverse Mercator*).

2.5.5 A projeção UTM

A projeção UTM é definida dividindo-se a Terra em 60 fusos de 6° de longitude. Para cada fuso, adota-se como superfície de projeção um cilindro transversal com eixo perpendicular ao seu meridiano central, que assume ainda o papel de longitude de origem. Apesar da característica

“universal” desta projeção, enfatiza-se que o elipsóide de referência varia em função da região da superfície terrestre.

O valor de uma coordenada horizontal avalia sua distância - em metros - do meridiano central. A coordenada vertical de uma localidade acima da Linha do Equador é sua distância - em metros - ao Equador [Gor02].

Todas as coordenadas UTM são positivas devido à incorporação no sistema de um falso Norte e um falso Leste. Isto se faz adicionando 500.000m ao meridiano central e 10.000.000m para as distâncias ao sul do Equador. Desta forma as coordenadas são fornecidas sempre em Norte e Leste [Yea98]. Como a mesma coordenada vertical pode ser associada a duas localidades distintas, uma acima e outra abaixo do Equador, é necessário indicar em qual hemisfério se localiza para identificá-la [Gor02].

Na Internet, os sites de Joe Mehaffey e Jack Yeazel [MY98] e do *National Center for Geographic Information and Analysis* [NCG00], possuem inúmeras conexões para páginas na internet sobre GPS e GIS.

Capítulo 3

Utilização de Dispositivos Portáteis para Posicionamento em Operações Militares

3.1 Soluções

Tendo em vista que o objetivo do trabalho é um modelo de ambiente integrado, envolvendo hardware e software, para posicionamento em operações militares, essa integração foi atingida com uso de soluções que serão agora descritas.

3.1.1 Hardware

O hardware envolvido neste trabalho é constituído de:

- Computador pessoal para a geração, a partir de uma imagem *raster* ou vetorial, de um arquivo de dados que possa ser transferido e, posteriormente, lido pelo PDA;
- receptor GPS GARMIN eTrex, que fornece as informações de hora e posição para o PDA;
- PDA Workpad C505 da IMB para visualizar o mapa, oriundo do computador pessoal, e obter informações do receptor GPS;
- Cabos e conectores responsáveis pela ligação entre os equipamentos.

Como, neste trabalho, a aplicação no PC que gera os arquivos PDB e PRC tem como única finalidade transformar uma imagem no formato windows bitmap(*raster*) em um arquivo PRC e um arquivo no formato *shapefile*(vetorial) em um arquivo PDB, ou seja, sem outras interações com a imagens para, por exemplo, alterá-la, o PC não precisa ter grande poder de processamento nem muita memória. Todo o trabalho foi desenvolvido e implementado alterando entre dois computadores: um Intel Pentium II - 800MHz com 512Mbytes de memória RAM e um AMD Duron 800MHz com 256Mbytes de memória RAM. Em nenhum houve algum tipo de problema de desempenho.

A opção de escolha do receptor GPS GARMIN eTrex se deu devido as suas características [Gar00], que são:

- Emissão dos dados utilizando o protocolo NMEA0183;
- Possibilidade de envio dos dados para um equipamento conectado a ele por meio de uma porta serial RS232;
- Possibilidade de inclusão de dados utilizando o protocolo proprietário GARMIN;
- 12 canais de recepção de emissão de satélites;
- A possibilidade de utilização de outros sistemas de referência (inclusive o SAD69 - utilizado nas cartas brasileiras) e outros sistemas de coordenadas (inclusive UTM - utilizado nas cartas terrestres brasileiras) na exibição da posição;
- Existência de um modo de demonstração que simula, a partir das coordenadas da última posição real obtida, o recebimento de informações de satélites e um deslocamento para o Norte, cuja direção pode ser alterada, mesmo dentro de um ambiente fechado;

Estas características aliadas ao seu baixo custo (é um dos equipamento mais baratos e completos do mercado) possibilitaram o desenvolvimento de uma ambiente que pode fazer uso de outras marcas de receptores GPS com características semelhantes, bem como a realização de testes com receptor GPS conectado ao PC sem a necessidade de transportar este para uma área que permitisse a recepção de informações dos satélites.

O custo do Workpad C505, que é igual(exceto pela cor) ao Palm M505 da Palm Inc., a facilidade de explicações, tutoriais e exemplos gratuitos principalmente na internet, além da possibilidade de conexão com a porta serial RS232 de um equipamento para recebimento e envio de dados foram fatores decisivos na escolha deste PDA. Este era o equipamento da Palm Inc. mais avançado no mercado no final do ano passado. O problema deste equipamento é sua vulnerabilidade à quedas e à água, mas essas características só se consegue com equipamentos cujo preço atinge valores de oito a dez vezes o preço de um equipamento comum.



Figura 3.1: Receptor GPS conectado a um PDA.

Outra característica positiva do Palm é a existência de um emulador para todo tipo de Palm, possibilitando testes através de chamadas diretas no ambiente de programação, sem o desconforto de ter que ficar instalando versões do programa e banco de dados todo tempo no Palm. Apenas testes finais eram feitos diretamente no Palm.

A Figura 3.1 mostra uma conexão entre um Palm e receptor GPS.

3.1.2 Software

Como parte deste trabalho foi implementado um programa que gerasse o arquivo de banco de dados que é transferido para o Palm e nele manipulado.

Geração do Banco de Dados

Esta aplicação tem duas funções básicas:

- Ler um arquivo de projeto (de extensão PRJ) contendo vários arquivos de imagens vetoriais (formato *shapefiles*) e gerar um arquivo PDB para arquivo de imagem, contendo os dados para reconstrução desta; e
- Ler um arquivo de projeto (de extensão PRB) contendo vários arquivos de imagens

raster (formato windows bitmap) e gerar um único arquivo PRC contendo os recursos tipo bitmap correspondente às imagens.

Tanto a geração do arquivo PDB como do arquivo PRC foram feitos byte a byte. A ordenação de gravação foi importante, pois eles teriam que ser gravado no esquema denominado *big-endian* [Hil01] [MV96], já citado anteriormente.

A partir de agora será especificada a criação do arquivo PDB e a criação do arquivo PRC.

Geração do PDB

O programa implementado converte os arquivos *shapefiles* (vetoriais) em arquivos PDB. Algumas simplificações foram feitas para a geração do PDB, mas que podem ser consideradas para trabalhos futuros:

- Não foram considerados os atributos constantes no arquivo dBASE, nem o arquivo de índices;
- Não foram considerados os buracos nos polígonos vazados;
- Os polígonos foram considerados como seqüências de linhas sem preenchimento;
- O programa converte, no máximo, dois arquivos *shapefiles* em PDB.

A terceira simplificação é devido a existência, no PalmOS, de apenas quatro elementos gráficos de desenho: ponto, linha, retângulo e bitmap. Não tendo sido criada nenhuma função especial para desenhar, por exemplo, polígonos. A última simplificação não tem relação com qualquer limitação do Palm, em trabalhos futuros ela pode ser estendida com facilidade.

Como um mesmo arquivo *shapefile* deve conter feições do mesmo tipo, para a representação de um mapa são necessários normalmente mais de um arquivo *shapefile*, um para cada feição. Em virtude disso houve a necessidade de se criar um arquivo de projeto, que nada mais é senão um arquivo ASCII cujo nome possui extensão PRJ e cada linha é um arquivo *shapefile*, um espaço e um número que indica a cor da feição. A Figura 3.2 mostra um exemplo de arquivo de projeto PRJ.

Tomando como exemplo o arquivo descrito na Figura 3.2, o programa lê cada arquivo *shapefile* e define uma caixa envolvente única para ambos. A partir daí, diferente do arquivo *shapefile*, ele não considerará a caixa envolvente de cada objeto de cada arquivo *shapefile*. São gerados então, para cada arquivo *shapefile* - que será lido novamente, o cabeçalho, a lista de registros e, finalmente, os registros. A Figura 3.3 representa graficamente a geração dos arquivos PDB a partir do projeto PRJ.

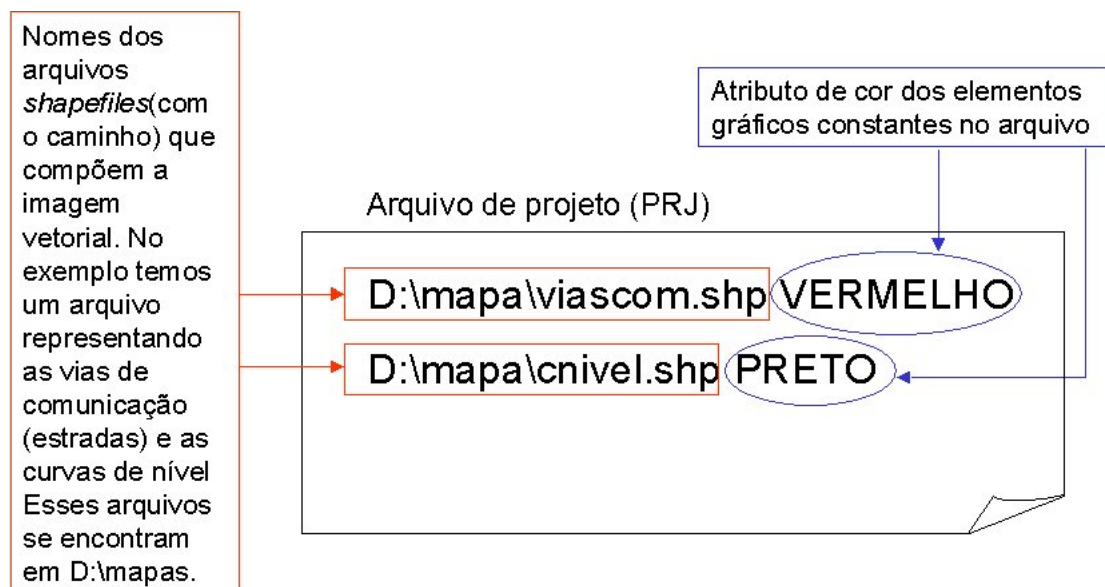


Figura 3.2: Exemplo de um arquivo de projeto PRJ.

A Tabela 3.1 define a estrutura de um objeto no arquivo PDB. Cada objeto é um registro no arquivo PDB. Existe a possibilidade de um registro ter mais do que 64kbytes de tamanho. Se isso ocorrer só será verificado quando da instalação do arquivo no Palm, pois isso acarretará um erro. Os exemplos utilizados neste trabalho não ultrapassam esse tamanho. Uma solução para evitar essa possibilidade é dividir um objeto em objetos menores, ou seja, um registro em dois ou mais registros. Cada registro representando um objeto é a formatação do segundo registro em diante. O primeiro registro contém a estrutura definida na Tabela 3.2, que é geral para todo o arquivo. Todos os campos são convertidos para caracteres antes de serem gravados e são separados por um espaço. Os registros são contíguos. A Figura 3.4 representa graficamente os registros e a ordem em que os seus campos são gravados no PDB.

Campo	Definição
nSHPType	Tipo da forma
nShapeId	Identificador do objeto
nParts	Total de partes do objeto
panPartStart[nParts]	Vetor de índice de início das partes
panPartType[nParts]	Vetor de tipos das partes (normalmente o mesmo para cada objeto)
nVertices	Total de vértices do objeto
padfX[nVertices]	Vetor de abscissas
padfY[nVertices]	Vetor de ordenadas

Tabela 3.1: Estrutura usada para definir um objeto no arquivo PDB

Campo	Definição
nent	Total de entidades (objetos)
enttype	Tipo das entidades (objetos)
bbminX	Menor valor em X da caixa envolvente(<i>bounding box</i>)
bbminY	Menor valor em Y da caixa envolvente(<i>bounding box</i>)
bbmaxX	Maior valor em X da caixa envolvente(<i>bounding box</i>)
bbmaxY	Maior valor em Y da caixa envolvente(<i>bounding box</i>)

Tabela 3.2: Estrutura usada para definir o primeiro registro no arquivo PDB

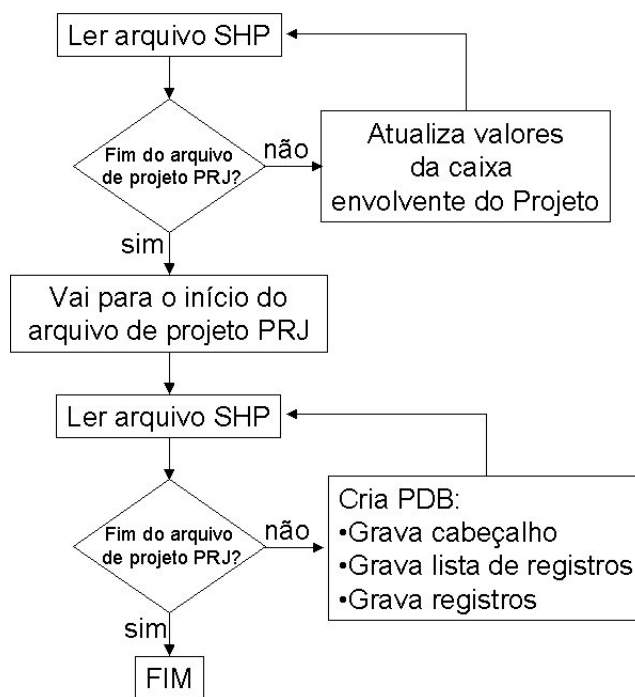


Figura 3.3: Geração dos arquivos PDB, a partir de um arquivo PRJ.

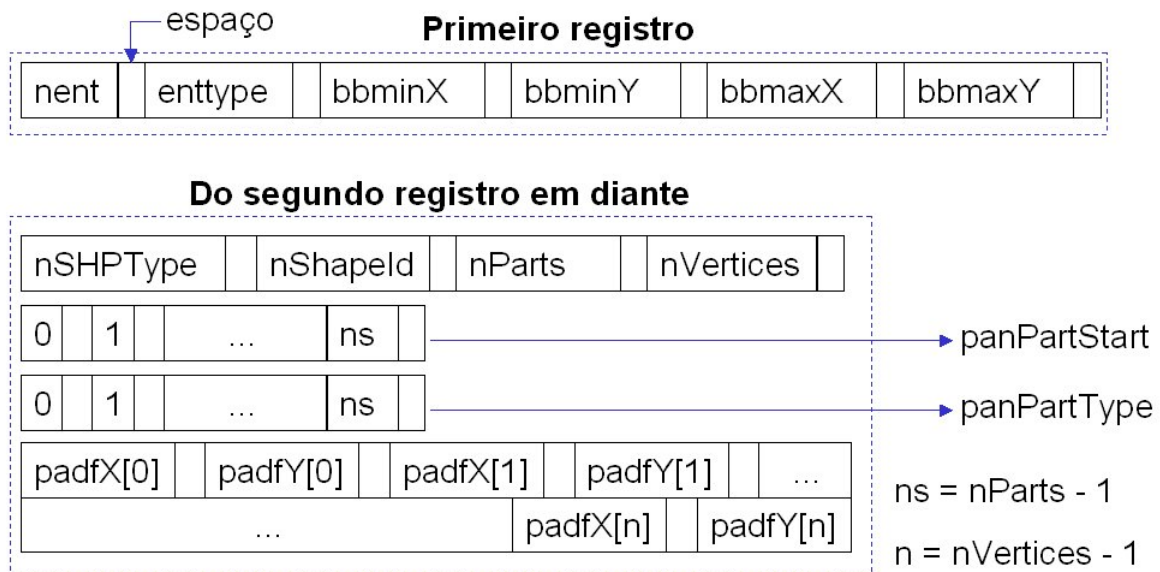


Figura 3.4: Registros em um arquivo PDB e a ordem de seus campos.

Geração do PRC

Antes de começar a explicação do processo de geração do arquivo PRC, é importante que se saiba que o tipo de arquivo bitmap utilizado foi o arquivo no formato window bitmap sem compactação e com oito bits de profundidade de cor. O motivo de não usar a compactação foi porque a compactação RLE utilizada por softwares como PaintShopPro e AdobePhotoshop, no ambiente windows, e também adotado no PalmOS para recursos bitmap, não podiam ser corretamente lidos pelo PalmOS após sua conversão para PRC. Existiam algumas particularidades geradas na compactação RLE do bitmap nestes softwares que o PalmOS não identificava corretamente. Havia a necessidade de se criar uma função para compactação RLE de forma que o PalmOS a identificasse. Outro problema é a não existência de uma documentação especificando detalhadamente a forma de compactação RLE usada pelo PalmOS. A outra simplificação é a utilização de bitmaps com oito bits de profundidade de cor, que é suficiente para representar um mapa para operações militares. Futuramente é possível, sem muita dificuldade, estendê-lo para outras profundidades de cor, inclusive 24 bits.

O uso de recursos bitmap no lugar de registros contendo as informações de cada pixel da imagem é porque o desenho pixel a pixel é bem mais lento do que o desenho do bitmap como um recurso.

Tendo em vista que as imagens utilizadas para orientação durante as operações militares requerem um certo nível de detalhamento que não permitiria colocá-las numa dimensão de cerca de 420 x 420 pixels, conforme limite de dimensões de criação de uma janela em *offscreen*

ou de um recurso bitmap no PalmOS, houve a necessidade de dividir a imagem bitmap em blocos, onde cada um destes daria origem a um recurso do tipo bitmap. A Figura 3.5 mostra um exemplo de divisão de uma imagem em blocos de 420 x 420 pixels. O tamanho máximo do bloco tem que se limitar ao tamanho máximo da janela em *offscreen*.

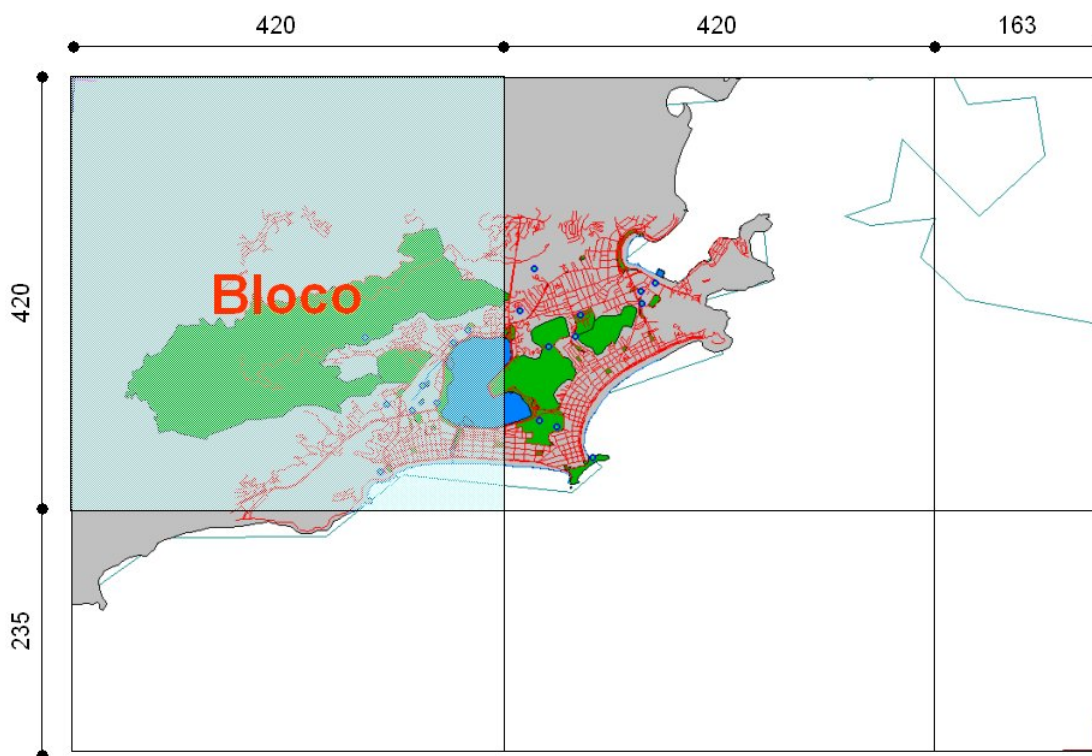


Figura 3.5: Divisão de uma imagem em blocos.

Devido à possibilidade de deslocamento do mapa (*panning*) e a fim de permitir uma transição suave de um bloco para o outro, deveria haver uma repetição de uma faixa da imagem em cada bloco na dimensão da tela do Palm em relação a cada bloco vizinho, ou seja, cada bloco deveria ter pelo menos 160 pixels em comum com os blocos que lhe fazem fronteira. Conforme cálculos realizados a utilização de blocos de 420 x 420 pixels com faixas de imagens em comum aumentaria o tamanho do arquivo em até cerca de 140%. A Figura 3.6 mostra um exemplo de divisão de uma imagem em blocos de 420 x 420 pixels com faixas de imagens em comum com os blocos vizinhos de 160 pixel.

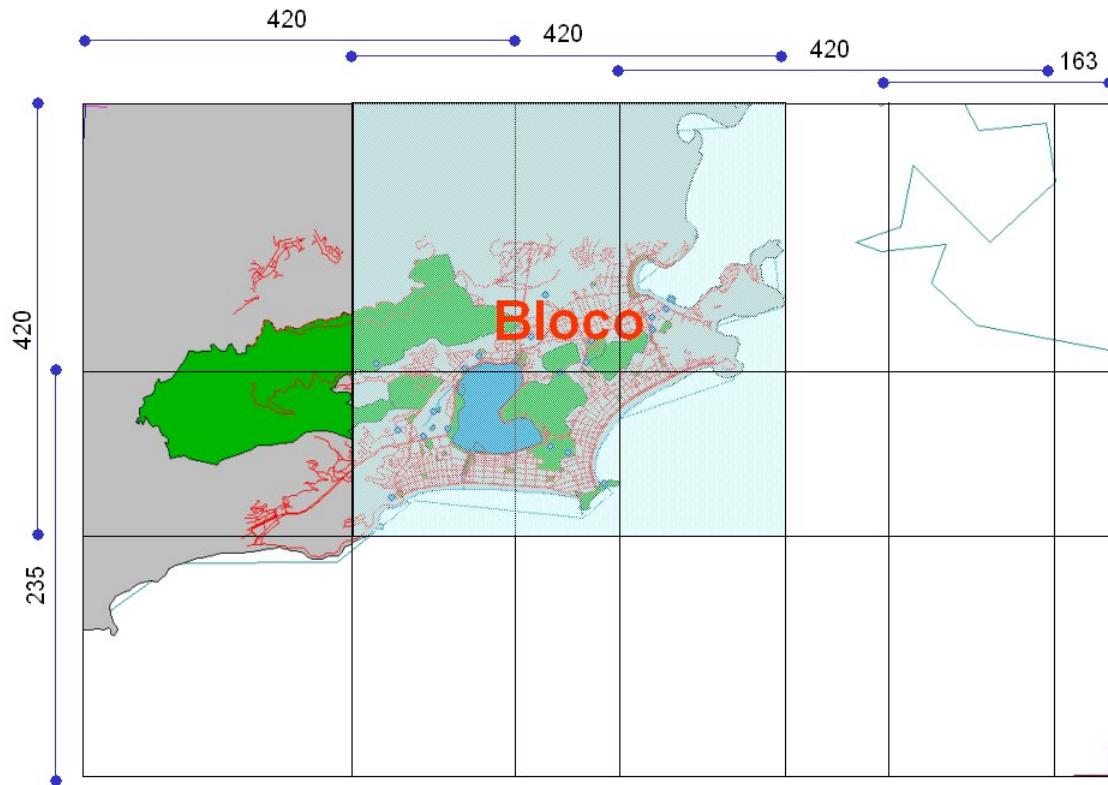


Figura 3.6: Divisão de uma imagem em blocos com faixas de imagens em comum com os blocos vizinhos.

Outra divisão da imagem, e que foi adotada, é em blocos do tamanho da tela, que não geraria repetição de imagens em arquivo. Desta forma cada janela de fundo seria composta de quatro blocos. A Figura 3.7 representa a divisão de uma imagem em blocos de 160 x 160 pixels e uma área de janela em *offscreen* selecionada. Esse método, conforme os cálculos aumentaria o tamanho do arquivo apenas em até cerca de 6% do seu tamanho original, em virtude do maior número de cabeçalhos (cada recurso bitmap tem um) e da repetição da tabela de cores para cada recurso. O gráfico da Figura 3.8 representa estes cálculos realizados e permite observar facilmente a diferença entre o tamanho do arquivo gerado utilizando a divisão em blocos de 420 x 420 pixels com faixas de imagem em comum e utilizando a divisão em blocos de 160 x 160 sem faixas de imagens em comum com seus vizinhos em relação ao arquivo original no formato bitmap.

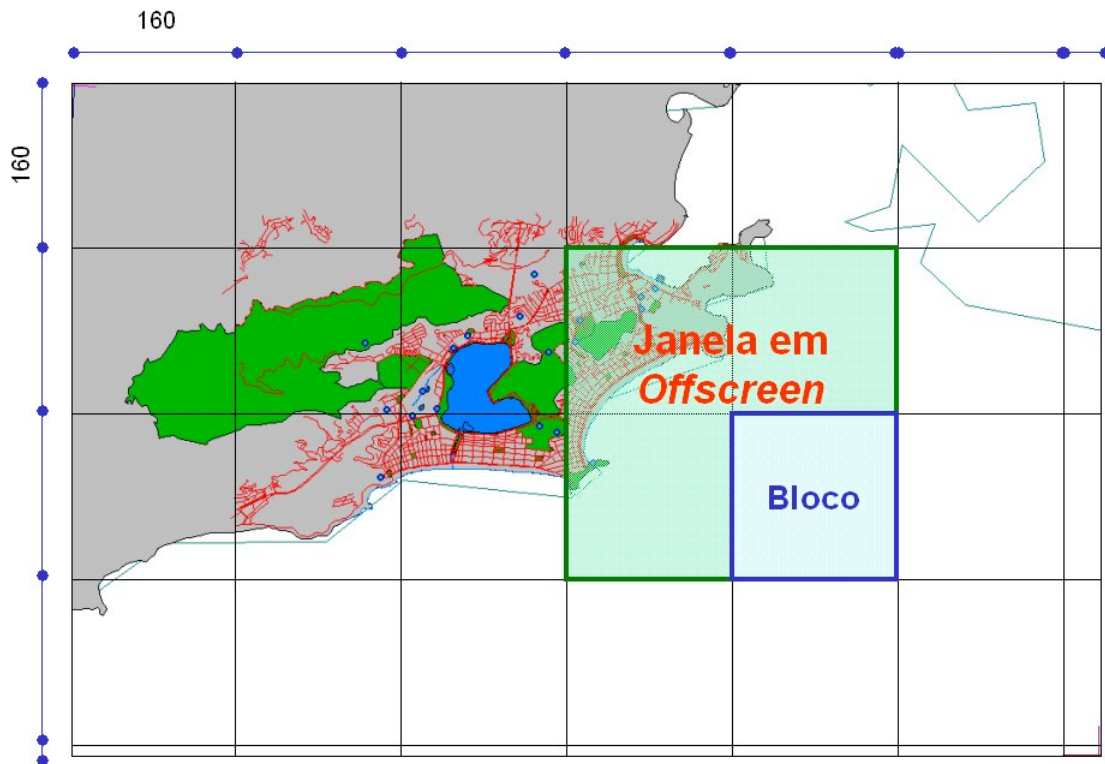


Figura 3.7: Divisão de uma imagem em blocos de 160 x 160 pixels.

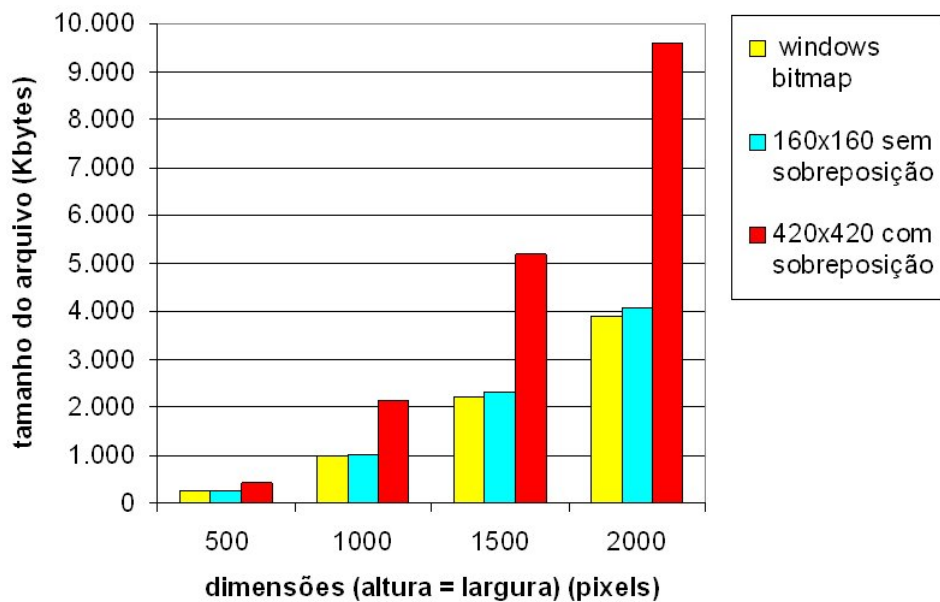


Figura 3.8: Tamanho do arquivo PRC gerado em relação ao modo de divisão da imagem.

No decorrer do trabalho foi feita uma modificação para aceitar outras imagens no mesmo PRC para indicar os níveis de zoom, uma vez que o zoom conseguido através das API do PalmOS deixam a imagem truncada, facilmente verificado nas Figuras 3.9(a), 3.9(b), e 3.9(c). Em seus trabalhos, Steve Mann e Ray Rischpater [MR01] também desaconselham, pelo mesmo motivo, a utilização da API do PalmOS diretamente na imagem para realizar o zoom e sugerem a utilização de outras imagens com mais detalhes para isso, conforme foi feito.

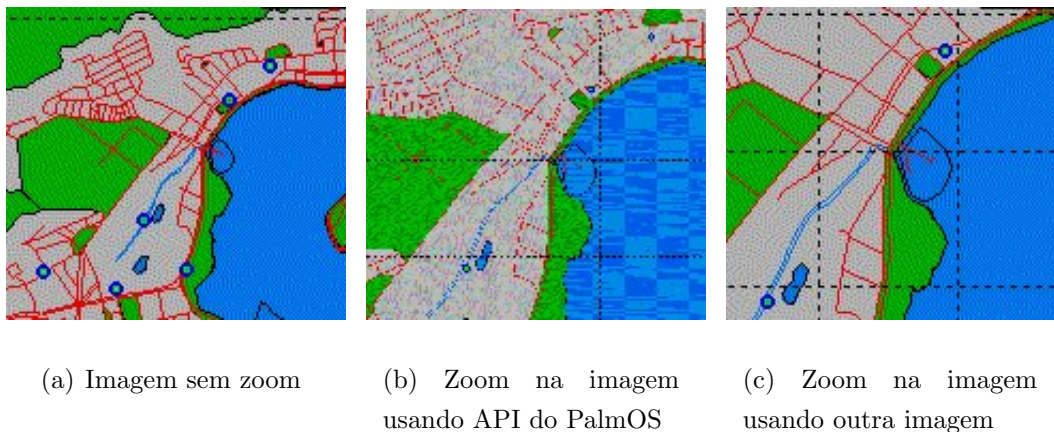


Figura 3.9: Comparação entre utilização da API do PalmOS e de outra imagem para a operação de zoom

Para a geração do arquivo PRC foi implementado um programa que lê um arquivo ASCII, com extensão PRB, contendo na primeira linha a quantidade de arquivos bitmap a serem convertidos e nas demais os dados desses arquivos (um em cada linha) separados sempre por um espaço, na seguinte ordem:

- Índice, iniciando em zero;
- Localização(*path*) e nome do arquivo bitmap; e
- Limites mínimo e máximo de longitude e de latitude em UTM.

O terceiro item é fundamental para o georeferenciamento da imagem. A Figura 3.10 apresenta um exemplo de um arquivo de projeto PRB. Todo o arquivo de projeto PRB dará origem a um único arquivo PRC (com mesmo nome, mas com extensão PRC), ou seja, para cada bitmap será feita uma divisão da imagem em blocos de 160 x 160 pixels e gravados como recursos num único arquivo PRC. Os arquivos bitmap constantes no arquivo de projeto PRB devem estar contidos na área abrangida pelo bitmap de índice zero, ou seja, do arquivo bitmap de índice um em diante deve ser um *zoom in* ou algum detalhamento maior de uma área dentro do bitmap de índice zero. Isso porque quando se executa o *zoom in*, procura-se pela primeira

imagem cujos limites estejam dentro dos limites da imagem atual. Se não houver, não será executado o zoom, portanto, a imagem que não estiver dentro dos limites da primeira nunca será usada, estará apenas aumentando o tamanho do arquivo PRC. Futuramente isso pode ser tratado na geração do arquivo PRC. A Figura 3.11 ilustra a localização dos bitmaps em relação ao primeiro.

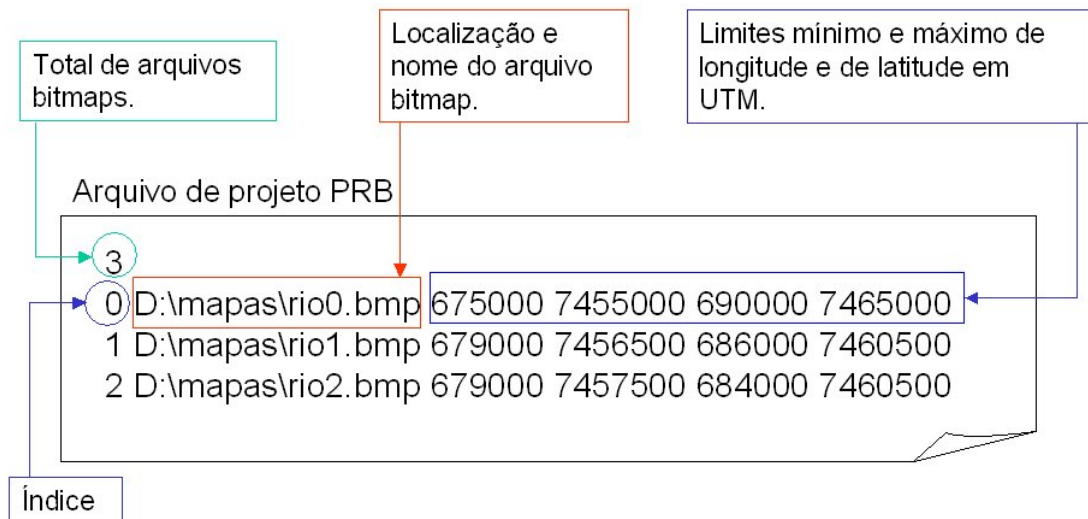


Figura 3.10: Exemplo de um arquivo de projeto PRB.

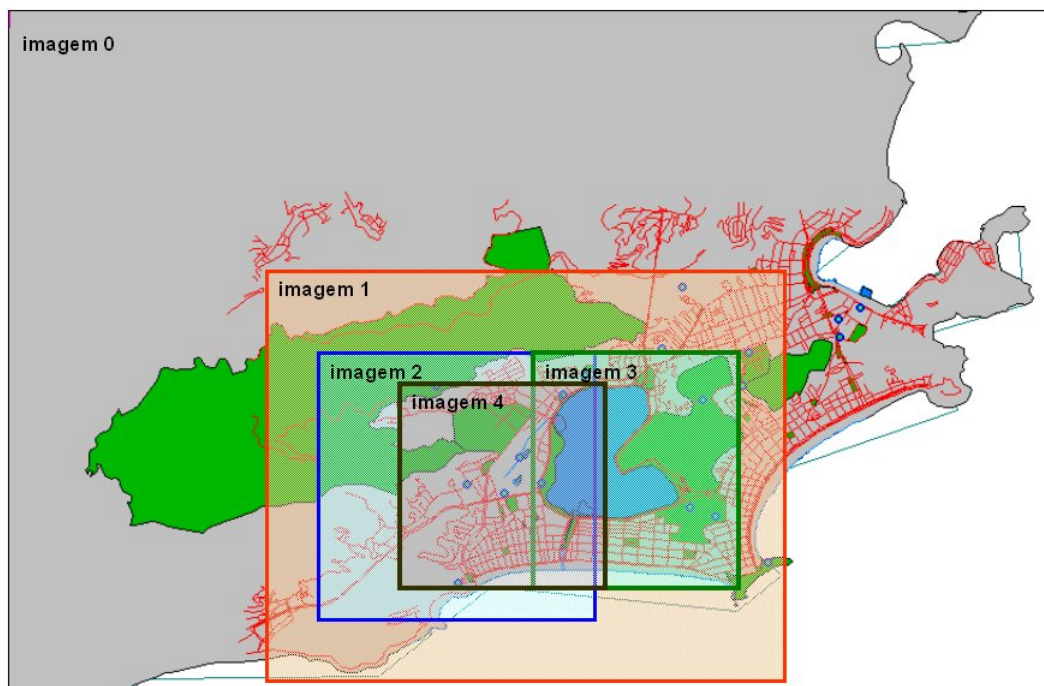


Figura 3.11: Localização relativa dos arquivos bitmaps de um projeto PRB.

Para guardar informações de número de blocos na horizontal e na vertical de cada bitmap, visando a sua reconstrução, bem como os seus limites mínimos e máximos, visando o georeferenciamento da imagem, foi criado um recurso extra, do tipo texto de informação da aplicação (*appinfostr*), que é basicamente uma cadeia de caracteres contendo o número total de blocos na horizontal, o número total de blocos na vertical, os limites mínimo e máximo de longitude e de latitude em UTM, todos separados por um espaço. A Figura 3.12 mostra a distribuição das informações no recurso *appinfostr*.

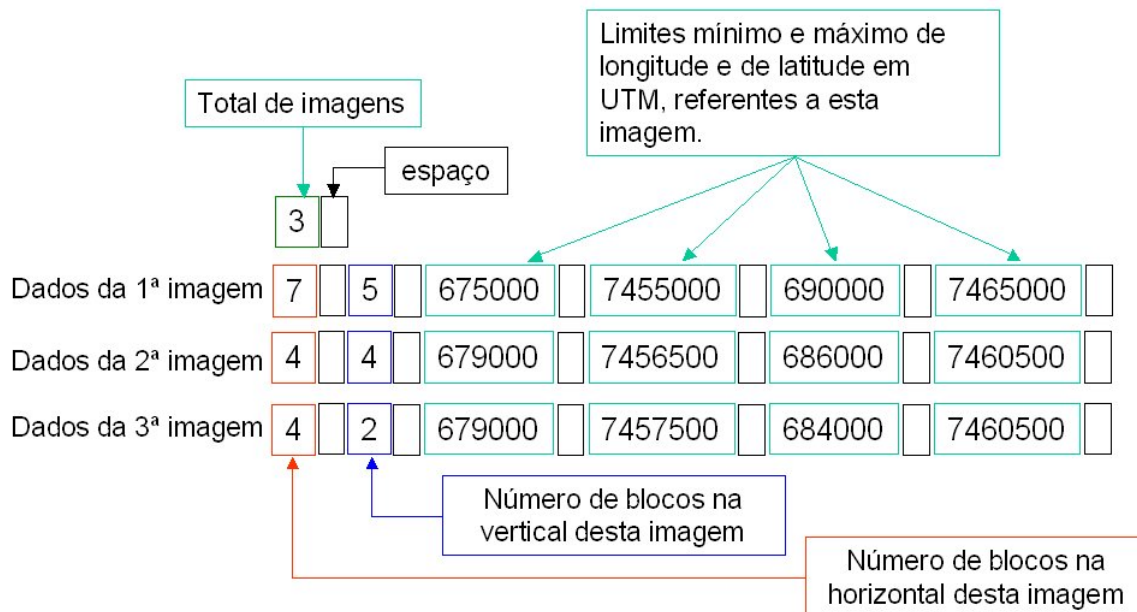


Figura 3.12: Distribuição das informações no recurso *appinfostr*.

Para a geração do arquivo PRC, será necessário ler os bitmaps do arquivo de projeto PRB três vezes para, em cada uma, realizar as seguintes tarefas:

1. São definidos e gravados os dados do cabeçalho do arquivo PRC e definido os dados do recurso *appinfostr* (calculado o número de blocos na horizontal e vertical de cada bitmap e lido os limites mínimos e máximos de longitude e latitude de cada bitmap);
2. É gerada a lista de entradas de recursos;
3. Os bitmaps são divididos em blocos e gravados como recursos e é gravado o recurso *appinfostr*.

A Figura 3.13 mostra um fluxograma representando a geração do arquivo PRC, a partir de um arquivo de projeto PRB.

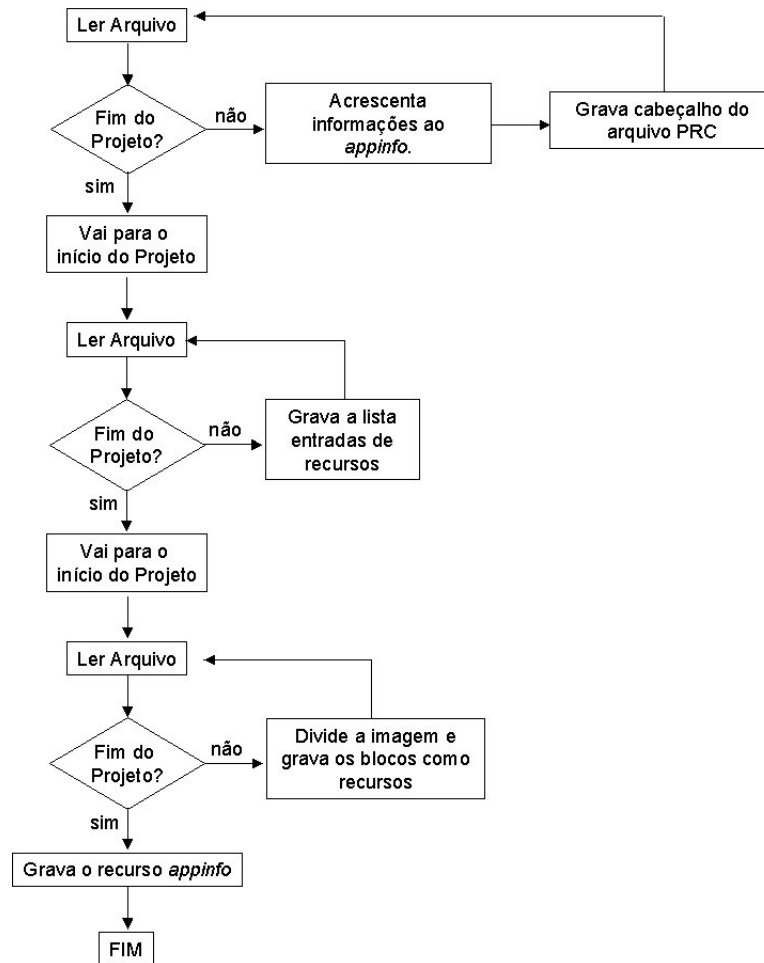


Figura 3.13: Geração do arquivo PRC.

No cabeçalho de cada recurso tem o índice, que é fundamental para identificar o bloco na imagem como um todo, sendo assim necessário para a montagem da imagem e também para a obtenção de uma imagem quando for executada uma operação de zoom. O índice tem que ser único para recursos do mesmo tipo. A padronização adotada é a seguinte: na primeira imagem o índice do primeiro bloco do canto superior esquerdo é 5.000, à medida que se anda na horizontal vai acrescentando 1 no índice de cada bloco e a medida que se anda na vertical acrescenta 100 no índice de cada bloco; o primeiro bloco da imagem seguinte começa em 10.000, a seguinte 15.000 e assim por diante, ou seja, a cada imagem o seu índice inicial é acrescido de 5.000. A Figura 3.14 representa um exemplo desta indexação. O índice utilizado para o recurso *appinfostr* é 4.000.

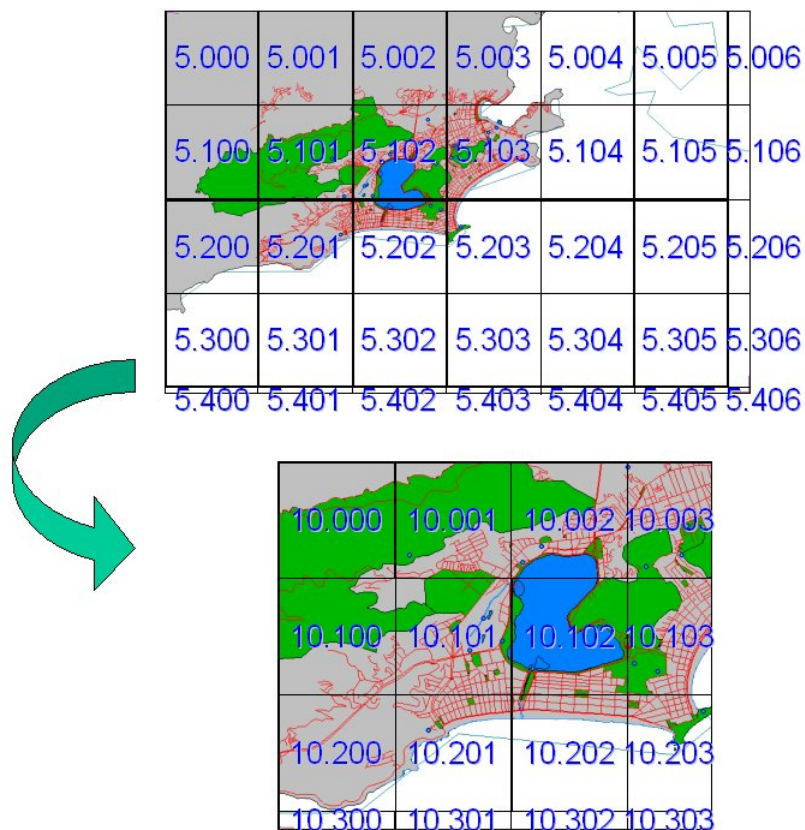


Figura 3.14: Indexação dos blocos de duas imagens de um mesmo PRC, onde a primeira imagem é a mais ampla e a outra é o primeiro nível de zoom.

Transmissão para o Palmtop

Para a transmissão dos arquivos PRC/PDB, gerados no PC, para o Palm, foi utilizado o software de sincronismo que acompanha o Palm (*Palm Desktop*).

Os Comandos no Palm

A utilização de um Palm em operações militares pode dificultar a utilização da caneta na tela, pois o ambiente pode ser hostil não só pela presença de tropas inimigas como pela própria situação do terreno e clima, sem mencionar o desgaste do militar durante a operação. Devido a isso quase todas as operações no Palm podem ser feitas utilizando os seus botões e, sempre que possível, a caneta também. No entanto a medição de distâncias e entrada de dados só são possíveis com toques da caneta na tela. Procedimentos alternativos utilizando os botões podem ser pensados futuramente. A Figura 3.15 mostra os botões do Palm e suas funções.

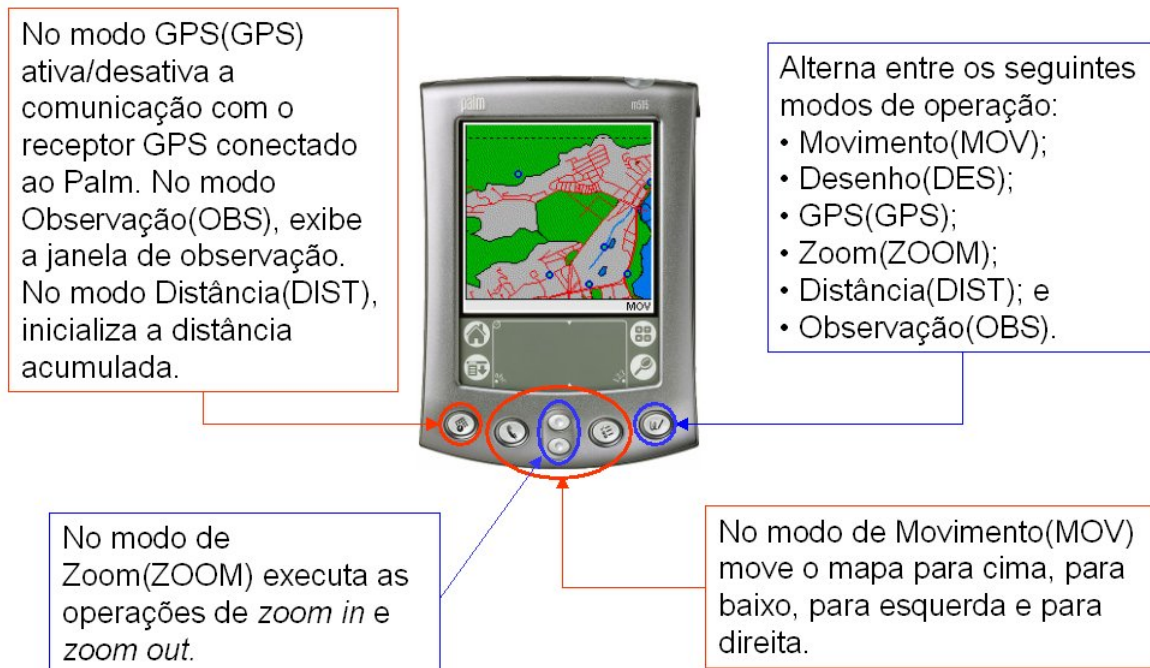


Figura 3.15: As funções dos botões do Palm.

Para possibilitar a utilização dos botões do Palm, foi necessário primeiro desabilitar as funções padrões dos mesmos no *loop* principal (função *EventLoop*) através de um comando que verifica se o botão foi pressionado e, caso verdadeiro, continua o *loop*. Outra medida tomada foi criar uma função para tratar o que deve ocorrer quando cada botão for pressionado (função *ConsoleGetInput*) e colocá-la dentro do *loop* principal.

Visualização da Imagem Raster

A imagem não é visualizada em toda tela do Palm, pois uma faixa de 10 pixels de altura na parte inferior da tela foi reservada como barra de status. Nesta serão exibidas informações, conforme o modo, de: hora, latitude e longitude em UTM, distâncias, modo de operação e mensagens. Ficando, portanto, uma área de 160 x 150 pixels para a imagem. A Figura 3.16 mostra o emulador do Palm exibindo o mapa e a barra de *status* indicando que está no modo de operação de movimento da imagem(*panning*).



Figura 3.16: Tela do emulador com o mapa e a barra de status.

O programa no Palm inicia lendo os dados do recurso *appinfo* referente à primeira imagem, isto é, a mais ampla. Esses dados (quantidade de imagens, número de blocos na horizontal e vertical e os limites de longitude e latitude), juntamente com outros que serão gerados no Palm, são guardados em uma variável global (*status*) cuja estrutura é mostrada na Tabela 3.3. Cada um dos campos desta variável guarda informações que serão importantes na visualização e manuseio da imagem. Esta variável é fruto das sugestões constantes em um documento técnico da Palm Inc. sobre Gráficos, Animação e Jogos no Palm [Pal01]. A idéia é guardar em uma estrutura as informações importantes para aquela situação, gravá-la como “preferências” quando o aplicativo é fechado, e depois poder retomá-lo naquele ponto quando for executado novamente.

Tabela 3.3: Campos da estrutura de status

Campo	Definição
buffer_h	altura do buffer, isto é, da janela em <i>offset</i>
buffer_w	largura do buffer, isto é, da janela em <i>offset</i>
numblockbuffer_h	número de blocos, na vertical, no buffer (pode ser 1 ou 2 blocos, dependendo da altura da imagem)

continua na próxima página

Tabela 3.3: Campos da estrutura de *status*(continuação)

Campo	Definição
numblockbuffer_w	número de blocos, na horizontal, no buffer(pode ser 1 ou 2 blocos, dependendo da largura da imagem)
indexblock_h	índice vertical do bloco, não confundir com a indexação do recurso, vai de 0 a (numblocks_h-1). Esse índice representa o bloco do buffer onde y_view se encontra.
indexblock_w	índice horizontal do bloco, não confundir com a indexação do recurso, vai de 0 a (numblocks_w-1). Esse índice representa o bloco do buffer onde x_view se encontra.
blocksbuffer_h	soma das alturas dos blocos no buffer
blocksbuffer_w	soma das larguras dos blocos no buffer
posblock_y	posição em y no bloco
posblock_x	posição em x no bloco
x_view	valor X, no buffer, do canto superior esquerdo da viewport, isto é, da área do buffer que será copiada para a tela
y_view	valor Y, no buffer, do canto superior esquerdo da viewport, isto é, da área do buffer que será copiada para a tela
numblocks_h	número total de blocos existentes na vertical
numblocks_w	número total de blocos existentes na horizontal
width	largura da imagem
height	altura da imagem
gps	indica se a recepção do gps está ativa ou não
bound	limites do mapa em UTM
modo	modo de operação
totzoom	total de zoom
zoom	zoom atual
extent	distância na carta entre as marcações da escala
deltaX	distância em X na tela entre as marcações da escala
deltaY	distância em Y na tela entre as marcações da escala
numinfo	total de recurso tipo <i>string</i> existentes

Após lidos os dados constantes no recurso *appinfo* referentes à primeira imagem (*bound*,

numblocks_w e *numblocks_h*), são seguidos os seguintes passos para a definição dos valores dos campos de status nesta inicialização:

1. São calculados os valores de *width* e *height* somando-se as larguras e alturas, respectivamente, de cada recurso bitmap referente àquela imagem. Esses dados constam do cabeçalho de cada recurso.
2. Se o número de blocos na horizontal for maior do que 1 o valor de *buffer_w* será 320 pixel e *numblockbuffer_w* será igual a 2, caso seja igual a 1 o valor de *buffer_w* será 160 pixel e *numblockbuffer_w* será 1. O mesmo é feito para *buffer_h* e *numblockbuffer_h* em relação ao número de blocos na vertical.
3. Os campos *indexblock_h* e *indexblock_w* iniciam como zero, ou seja, a primeira janela em *offscreen* que será criada será composta pelos blocos [0,0], [0,1], [1,0] e [1,1], caso o número de blocos na horizontal e na vertical sejam maior do que 1.
4. Sabendo os blocos que existem define-se a largura (*blocksbuffer_w*) e a altura (*blocksbuffer_h*) da imagem no buffer (podem ser menores do que as dimensões do *buffer*, nunca serão maiores).
5. Os valores de *posblock_x*, *posblock_y*, *x_view* e *y_view* são inicializados como zero, ou seja, a abscissa e ordenada da posição inicial em relação a janela em *offscreen* será o canto superior esquerdo do bloco[0,0]. E a abscissa e ordenada da área que será copiada para a tela são as do canto superior esquerdo do mesmo bloco.
6. O modo inicial de operação é Movimento(MOV).
7. Como o aplicativo inicia sem comunicação com o receptor GPS conectado ao Palm, a variável *gps* é falsa.
8. O total de zoom corresponde a o total de figuras no arquivo PRC, mesmo sabendo que as imagens mais detalhadas podem corresponder a áreas diferentes de uma mesma área maior.
9. O valor da variável *zoom* inicia como zero e vai sendo incrementada a cada vez que se realiza uma operação de *zoom in* e diminuído com *zoom out*.
10. A variável *numinfo* corresponde ao total de informações que existem, no caso iniciando com zero.
11. A escala é calculada (função *DefScale*) em função dos limites do mapa e da largura e altura da imagem. Através das equações 3.1 e 3.2 obtém-se os valores de *deltaX* e *deltaY* em relação a uma distância(*extent*) que inicia em 4.000m e vai diminuindo até que o

δX ou δY fiquem com valores entre 10 e 80 pixels, fora dessa faixa só aparecerá a linha da escala sem divisão.

$$\delta X = \frac{\text{extent} \cdot \text{width}}{(\text{MaxX} - \text{MinX})} \quad (3.1)$$

$$\delta Y = \frac{\text{extent} \cdot \text{height}}{(\text{MaxY} - \text{MinY})} \quad (3.2)$$

Sabendo que:

- extent: é a distância cujas divisões(δX e δY) representarão;
- width: largura da imagem em pixels;
- height: altura da imagem em pixels;
- MaxX e MinX: são os limites máximo e mínimo das longitudes do mapa;
- MaxY e MinY: são os limites máximo e mínimo das latitudes do mapa;

Feita essa inicialização dos valores em relação ao primeiro mapa, os passos seguintes são:

1. apagar a janela em *offscreen* (*buffer*) existente(se houver) e criar uma nova e desenhar os blocos correspondentes nela(função *DrawBuffer*);
2. com base em *x_view* e *y_view*, copiar um retângulo de dimensões 160 x 150 pixels do *buffer* para as coordenadas [0,0] da janela ativa, ou seja, exatamente a área de mapa da tela (função *DrawBMP*);
3. desenhar as linhas horizontal e vertical da escala com (ou sem) as divisões (função *DrawScale*);
4. Limpar a barra de status e a preencher com o modo de operação ativo (funções *ClearStatusBar* e *DrawModo*);

Quando é selecionado o modo de zoom(ZOOM) e pressiona o botão para realizar um *zoom in* verifica-se se o centro da tela está dentro dos limites da imagem seguinte e, se estiver, faz quase todo este processo de inicialização para essa nova imagem, conforme descrito anteriormente desde leitura do recurso *appinfo* para esta nova imagem até o desenho de uma área da janela em *offscreen* para a tela. A diferença é que os valores de *x_view*, *y_view*, *indexblock_w* e *indexblock_h* serão calculados em função das coordenadas do centro da tela. O campo *zoom* da estrutura *status* é incrementado. Para o *zoom out* o processo é o mesmo e o campo *zoom* é diminuído de 1.

Por ocasião do zoom, o centro da tela atual é o centro da nova tela, se possível, isto é, se não ultrapassar as fronteiras da nova imagem. Para que isso ocorra guarda-se as coordenadas do centro da tela em coordenadas do mundo e depois calcula-se x_view , y_view , $indexblock_w$ e $indexblock_h$ na nova imagem em função destas coordenadas. A Figura 3.17 representa graficamente este processo.

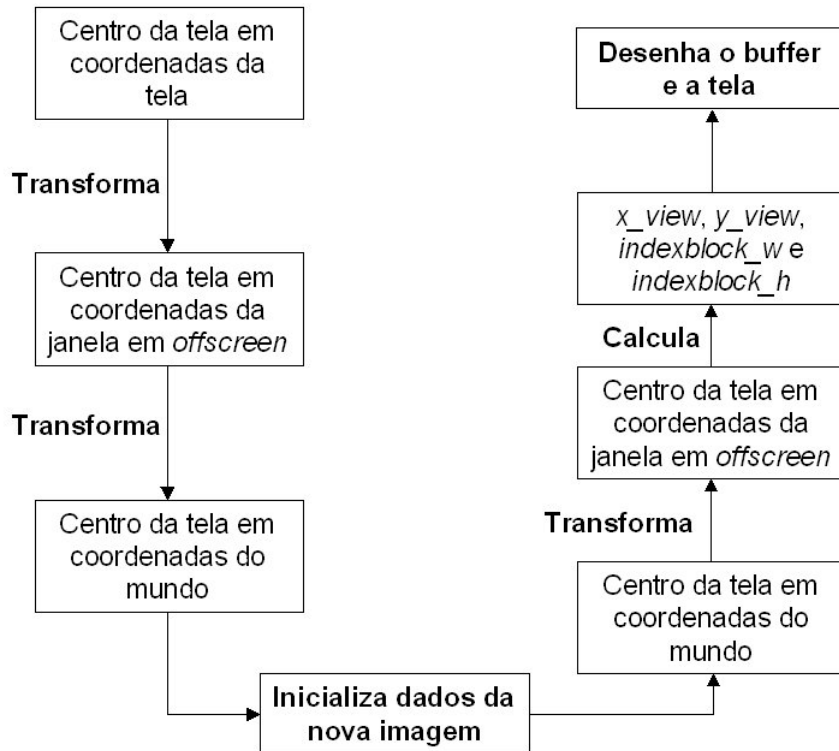


Figura 3.17: Fluxo das operações por ocasião de uma operação de zoom.

O movimento (*panning*) da imagem da tela é feito através dos botões assinalados na Figura 3.15 e também com a caneta. Com a caneta é feito através dos eventos de *penDown*, *penMove* e *penUp*, descritos abaixo:

- No evento *PenDown* as coordenadas da tela que a caneta tocou são guardadas em duas variáveis globais (gX e gY) e uma terceira variável global indica que a caneta encostou na tela ($gPenDown = \text{verdadeiro}$).
- No evento *PenMove* as novas coordenadas da caneta, junto com as coordenadas iniciais gX e gY são passadas a função *DrawNewLocation*, que calcula o deslocamento, aplica-o a x_view e y_view , determinando se deve ou não redesenhar o *buffer*, e redesenha a tela e a escala. No final os valores de gX e gY são atualizados.

- No evento *PenUp* a terceira variável global *gPenDown* é definida como falso, indicando que a caneta não está mais em contato com a tela.

Para suavizar o movimento e não haver pulos de imagens quando houver redesenho do *buffer* o deslocamento em qualquer direção se dá de, no máximo, 5 em 5 pixels. O movimento utilizando os botões define as variáveis globais *gX* e *gY* como zero e define um deslocamento de 5 pixels em x ou em y (positivo ou negativo), conforme o botão pressionado, fornecendo esses dados à mesma função *DrawNewLocation*.

Os eventos *PenDown* e *PenMove* em modo de Movimento também executam a função *CoordScreenToCoordWorld*, para transformar as coordenadas do ponto da tela em coordenadas do mundo e exibi-la na barra de *status*.

No modo de medição de distâncias (DIST), as distâncias que podem ser medidas tem que estar na tela atual. A cada vez que a caneta é encostada à tela, a distância é calculada em relação ao ponto anterior e vai acumulando, em uma variável global, o valor e exibindo-o na barra de status. Para inicializar a medição de distância basta pressionar o botão mais à esquerda do Palm, conforme mostra a Figura 3.15.

Visualização da Imagem Vetorial

Para o desenho da imagem vetorial no PalmOS, uma imagem é processada de cada vez. O acesso aos arquivos é necessário sempre que tem que redesenhar a imagem. Alguns conceitos existentes para imagem *raster* são usado na imagem vetorial, como por exemplo os blocos e a estrutura *status*, mas com algumas alterações. O bloco no caso de desenho de imagem vetorial é do tamanho da janela em *offscreen*, que neste caso foi possível ter dimensões de 460 x 460 pixels. Ele é necessário para definir a localização de um ponto da tela em relação em relação às coordenadas no mundo real.

Como o PalmOS só tem como elementos gráficos de desenho o ponto, a linha e o retângulo, os objetos do tipo polígonos foram desenhados ligando seus pontos linha a linha.

Houve a necessidade de fazer um *clipping* dos elementos desenhados para que só fosse desenhada a parte correspondente às dimensões da janela em *offscreen*. Na realidade não foi usado nenhum algoritmo específico. As funções de desenho do PalmOS permite que se desenhe em coordenadas muito além das dimensões da janela em *offscreen*. Só que a parte fora desta será descartada. Esta área possível de desenhar é cerca de 76 vezes maior do que o tamanho da janela em *offscreen*, conforme mostra a Figura 3.18. O zoom é feito ajustando as coordenadas da imagem dentro destes limites de desenho possível, conforme as Figuras 3.19(a), 3.19(b)

e 3.19(c) representam. As imagens constantes na Figura 3.20 mostram a definição da imagem mesmo após três operações de zoom.

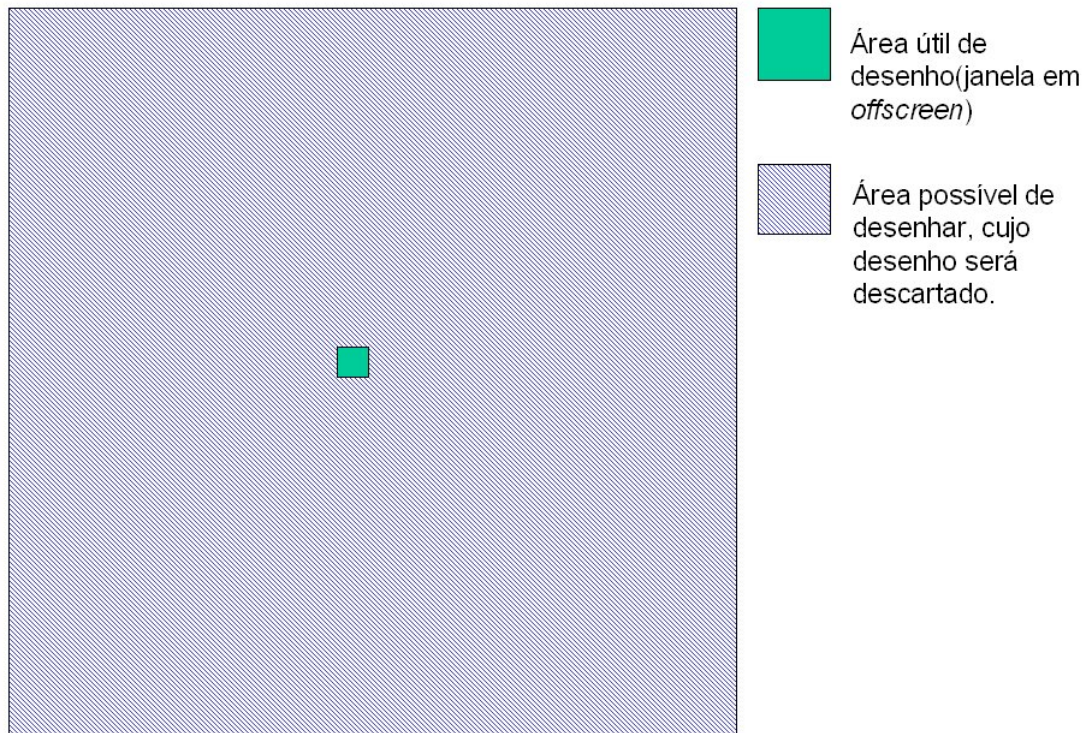


Figura 3.18: Proporção entre a janela possível de desenhar e a janela útil.

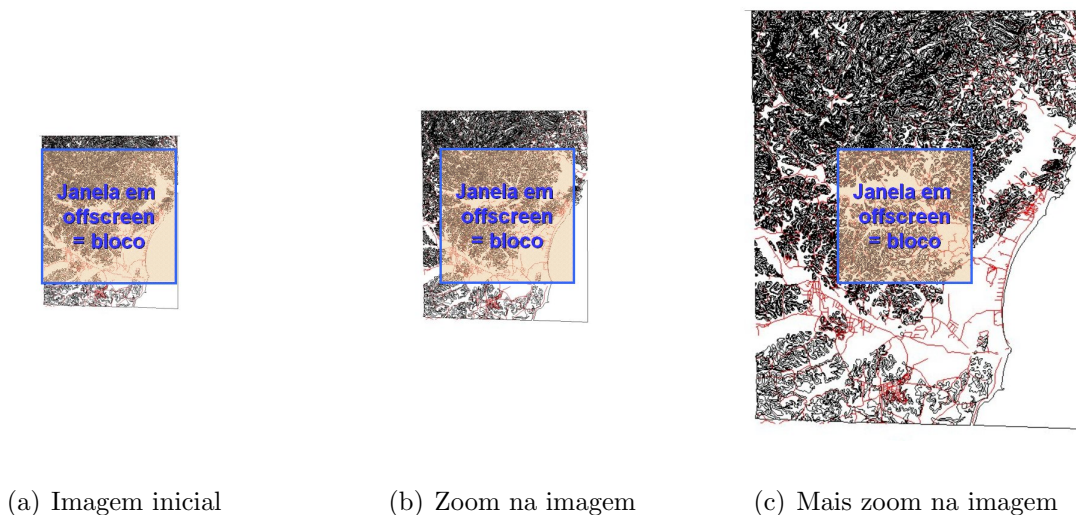


Figura 3.19: Representação da forma como é realizado o zoom com a imagem vetorial

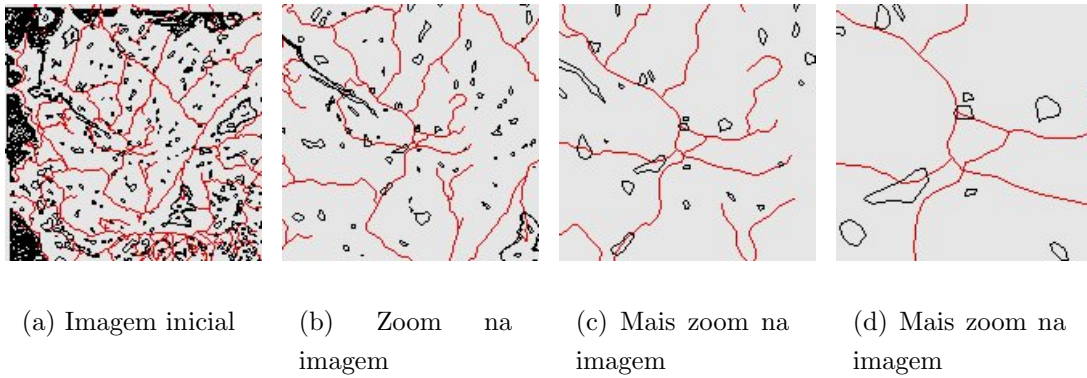


Figura 3.20: Zoom com uma imagem vetorial

Cada vez que a imagem é lida com um nível diferente de zoom ela é inicializada e dividida em blocos com faixas em comum de 160 pixel de largura ou altura para evitar pulos quando cruza as suas fronteiras. Este pulo ocorrerá devido ao tempo gasto no redesenho da imagem. A Figura 3.21 mostra uma imagem desenhada e seus possíveis blocos. Os blocos servem para acompanhar quando redesenhar a imagem novamente, definindo em que coordenada se fará o redesenho. Esta divisão só é possível com a imagem com estas dimensões, se mudá-la(zoom) terá que ser feita nova divisão. Diferente da imagem *raster*, essa divisão não gera aumento do banco de dados porque é feita no Palm e em tempo de execução, sem alteração do banco de dados. A mudança de bloco gera um novo desenho da imagem toda em uma nova posição, como mostram as Figuras 3.22(a), 3.22(b) 3.22(c).

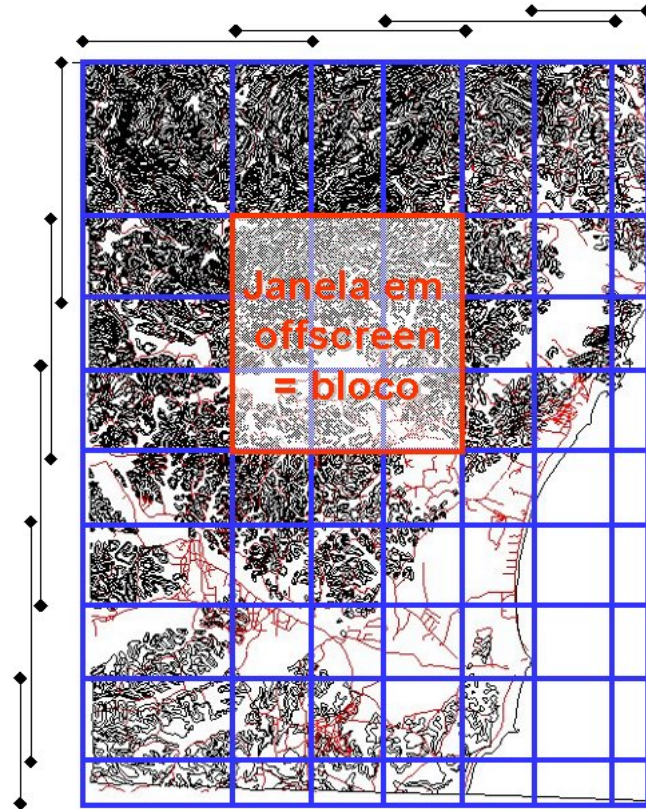
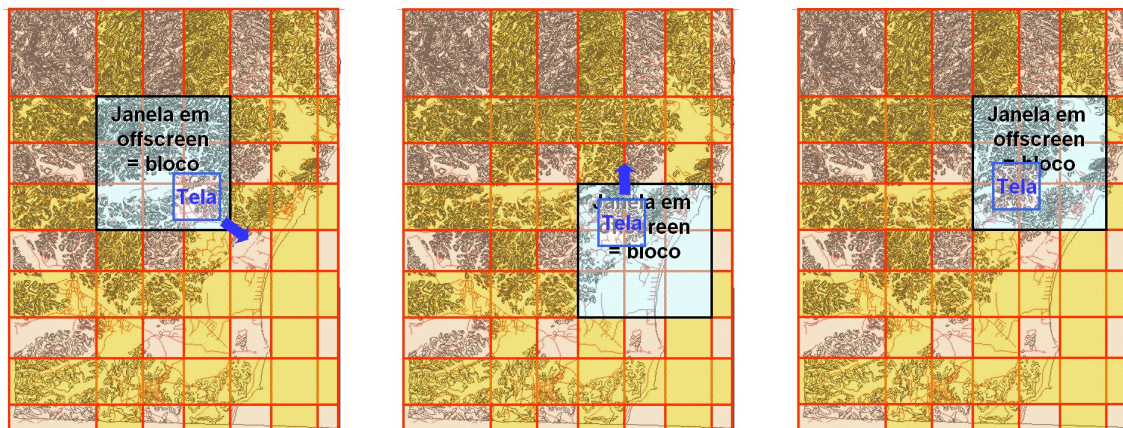


Figura 3.21: Divisão de uma imagem vetorial em blocos.



(a) Posição da tela e da janela em *offscreen*

(b) Nova posição da tela e da janela em *offscreen*

(c) Mais uma nova posição da tela e da janela em *offscreen*

Figura 3.22: Representação da forma como é realizado o movimento com a imagem vetorial

O tempo de redesenho torna inviável a utilização das imagens vetoriais de grandes áreas, como as utilizadas em operações militares, devido a isso o trabalho se voltou unicamente à

imagens *raster*. No Capítulo seguinte serão feitas comparações sobre a utilização dos dois formatos.

Aquisição dos Dados do Receptor GPS

A Figura 3.23 mostra a tela quando em modo GPS e recebendo dados do receptor GPS.



Figura 3.23: Imagem do aplicativo no modo GPS, recebendo informações.

Para a utilização do receptor GPS conectado ao Palm alguns detalhes dos dois equipamentos devem ser ressaltados:

1. Durante deslocamentos em operações militares utilizando o Palm conectado ao receptor GPS, o Palm não deve entrar no chamado *sleep mode*, pois além de fechar a conexão com o receptor GPS, a tela se apagará. Esse tempo para entrar em *sleep mode* é configurável de 1 a 3 minutos no Palm que está sendo utilizado no trabalho. Para evitar que ele entre neste modo, faz-se necessário zerar o controlador responsável por esta contagem

de tempo antes de completar o tempo mínimo, ou seja, um minuto. Isso é feito através da função do PalmOS *EvtResetAutoOffTimer*, caso a conexão com o receptor GPS seja ativada. A cada 50 segundos a função é executada.

2. O receptor GPS fornece os dados captados dos satélites no formato do protocolo NMEA0183. Neste protocolo as coordenadas são fornecidas em graus, minutos e frações de minutos e no sistema WGS84. Os mapas de operações militares estão normalmente em UTM e no sistema SAD69. Os mapas utilizados neste trabalho estão em UTM e no sistema WGS84. Houve a necessidade de fazer uma transformação de graus, minutos e frações de minutos para UTM. Como trabalho futuro será sugerida a utilização também de outras projeções e outros sistemas de referência.

Neste trabalho, como as informações que nos interessava eram de hora e coordenadas, nos interessava apenas uma sentença do protocolo NMEA0183. Este protocolo é emitido constantemente pelo receptor GPS Garmin eTrex, a partir do momento que ele é ligado, mesmo que não esteja recebendo sinal de nenhum satélite. O protocolo é emitido no formato ASCII a uma taxa de transmissão de 4800 bps constituído de 10 linhas a cada 2 segundos, onde cada linha é uma sentença, todas iniciadas por um código e com as informações separadas por vírgula. A sentença que nos interessa é a iniciada por \$GPRMC [RM99]. A cadeia de caracteres abaixo representa um exemplo desta sentença:

```
$GPRMC,204700,A,3403.868,N,11709.432,W,001.9,336.9,170698,013.6,E*6E
```

As informações que nos interessam no exemplo acima são:

- 204700: hora no meridiano de Greenwich;
- A: se a letra nesta posição for A é porque a informação é correta e obtida do satélite, se for V é porque tem algum problema (não deve estar conseguindo captar sinal de satélite) ou está no modo de demonstração (receptor Garmin eTrex);
- 3403.868: latitude em GGMM.MMM, onde G é grau e M é minuto, utilizando o sistema de referência padrão do GPS.
- N: direção da latitude, N - Norte (neste caso) ou S - Sul;
- 11709.432: longitude em GGGMM.MMM, do mesmo modo que a latitude; e
- W: direção da longitude, W - Oeste (neste caso) ou E - Leste;

Para transformação das coordenadas de graus, minutos e frações de minutos para UTM foram utilizadas funções para PC disponíveis na internet [Wag02] e adaptadas para o PalmOS. Essas funções fazem uso de algumas poucas funções matemáticas. Seguindo a sugestão dada por Steve Mann e Ray Rischpater [MR01], com a finalidade de evitar um gasto desnecessário de espaço que seria ocupado por *libraries* matemáticas para o PalmOS, criei as minhas próprias funções matemáticas apenas com as operações que interessavam.

O aplicativo, ao receber os dados do receptor GPS, localiza as coordenadas fornecidas e o coloca como centro da tela, desde que isso seja possível, pois se estiver muito perto das extremidades da imagem o ponto localizado não ficará no centro.

O ponto correspondente às coordenadas recebidas naquele momento é marcado na tela do Palm. À medida que se caminha, a nova posição fornecida é colocada novamente como centro da imagem, redesenhando a tela com uma imagem com este ponto central, o que dá uma idéia de movimento do mapa mantendo o ponto central marcado e fixo. Quando chega próximo a uma das bordas da imagem, o mapa para de se deslocar e quem passa a se deslocar é o ponto.

As Observações

A possibilidade de escrever observações no Palm também foi implementada. Para tal foram utilizados recursos do tipo cadeia de caracteres (*string*). Onde ao gravar uma observação cria-se um novo recurso com aquela observação, incrementando o índice do recurso. Como a navegação entre as observações é feita em função dos índices dos recursos, ao apagar uma observação que se encontra no meio de outras é necessário refazer os índices, diminuindo de um todos os índices dos recursos que vem após o que foi apagado. A Figura 3.24 mostra a tela do modo Observação quando incluindo uma. As observações são escritas com a caneta do Palm usando um formulário em forma de teclado ou o *Graffiti*, ambos já estão previamente instalados no Palm.



Figura 3.24: Tela do Palm no modo de observação.

3.2 Emprego

A utilização do ambiente proposto neste trabalho propicia:

1. Uma diminuição no tempo de obtenção da posição, marcação no mapa e comparação com informações existentes em calcos e extratos de planos, já que as coordenadas da posição são obtidas pelo receptor GPS e mostrada na tela do Palm juntamente com o mapa com calcos desenhados nele, bem como de dados escritos sobre a forma de registros de observações. Essa rapidez e precisão das informações será fundamenta à Operação uma vez que permitirá o emprego das frações certas no momentos e lugares certos.
2. Uma atualização das informações importantes como as referentes ao mapa(não existência de um rio que foi represado, por exemplo) ou à operação(inexistência de um campo minado previsto, por exemplo) através do registro de observações.
3. Um menor desgaste físico dos militares que fazem grandes deslocamentos e, na grande

maioria das vezes, não motorizados, pelo emprego de um hardware de peso e dimensões reduzidos que dispensa o transporte de extratos de planos, tabelas, calcos e mapas em papel. Facilitando ainda os deslocamentos em terrenos acidentados e com vegetação abundante ou outros obstáculos.

4. A possibilidade de utilização dos receptores GPS já existentes nas Organizações Militares, desde que permitam conexão com outros equipamentos via porta serial e que utilizem o protocolo NMEA0183 para transmissão dos dados.
5. A facilidade de utilização do hardware, permitindo uma rápida prontificação de militares para sua operação.
6. A aquisição de uma quantidade considerável de hardware devido ao seu baixo custo, propiciando o emprego por um número maior de frações.

Capítulo 4

Resultados

A utilização de imagens vetoriais mostrou-se bastante vantajosa no que se refere a operações de zoom, devido à precisão da nova imagem gerada (Figura 3.20), mas perde muito em relação ao tempo de redesenho da janela em *offscreen*. A transferência para o Palm também foi mais lenta. Mas o que foi mais marcante e que praticamente inviabilizou sua utilização no Palm, foi o tempo de redesenho. A possibilidade de usar imagens vetoriais mais simples prejudica sua utilização em operações militares onde esses detalhes são importantes. A Figura 4.1 mostra um gráfico comparativo dos tempos de transmissão da imagem para o Palm e o tempo de desenho para diversas imagens vetoriais e *raster*.

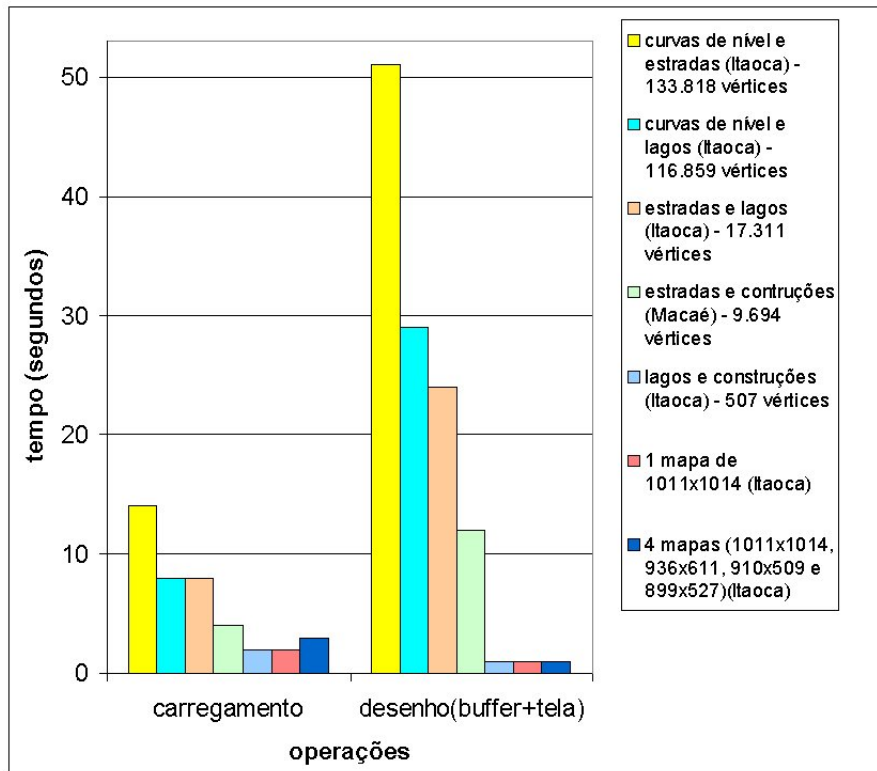


Figura 4.1: Tempo para carregamento e tempo para desenho da imagem (vetorial x *raster*).

No início das implementações, foi gerado um arquivo PDB, semelhante ao arquivo contendo a imagem vetorial, só que mais simples, para desenhar o bitmap pixel a pixel, mas foi verificado que a utilização de recursos bitmap em um arquivo PRC permite um desenho mais rápido da imagem do que se for desenhá-la pixel a pixel. A Figura 4.2 mostra a grande diferença no tempo gasto para o carregamento e o desenho para imagens geradas para desenho pixel a pixel e para desenho do recurso bitmap como um todo.

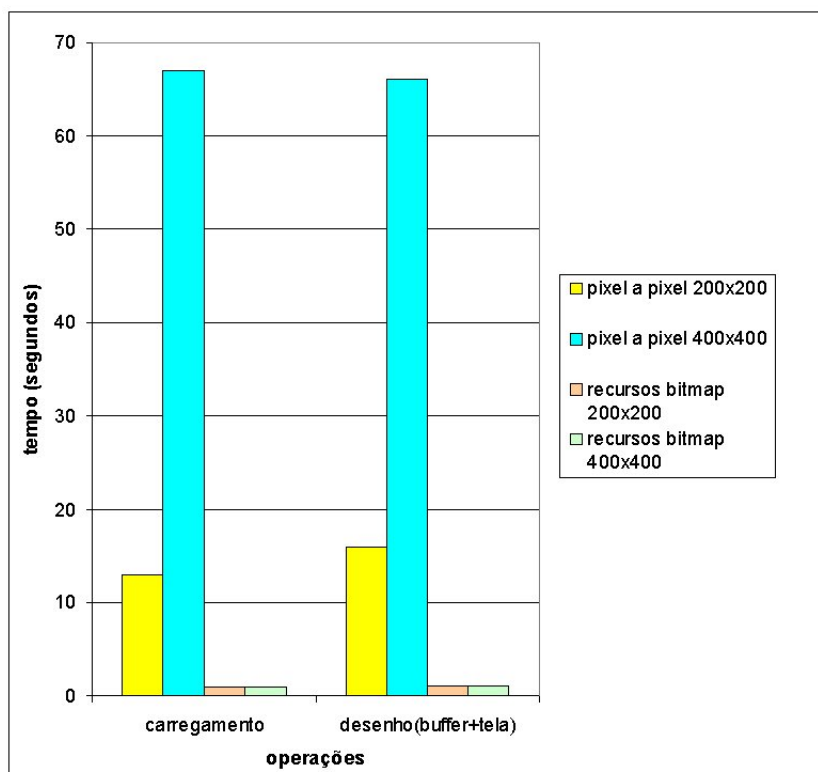


Figura 4.2: Tempo para carregamento e tempo para desenho da imagem (pixel a pixel x recursos).

Outro problema inicial de utilização de imagens *raster* é o zoom, que se for realizado através das API do PalmOS ficam truncadas, ou seja, distorcidas, conforme mostram as Figuras 3.9(a), 3.9(b), e 3.9(c). Mas a utilização de outras imagens *raster* para representar o zoom de uma determinada área permitiu uma excelente definição da imagem após o zoom e a possibilidade de utilização também de imagens satélites ou fotografias aéreas (desde que estejam georeferenciadas). Isso sem perda de performance do Palm. A Figura 4.3 mostra esta utilização de tipos diferentes de imagens (satélite, fotografias aéreas e mapas).

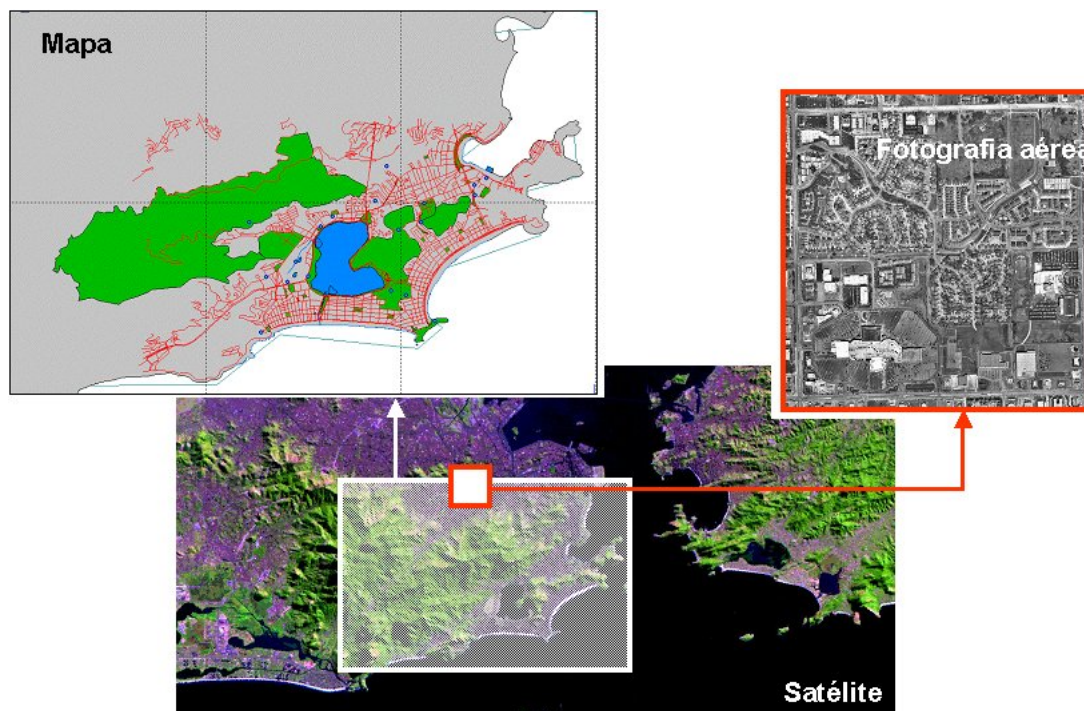


Figura 4.3: Utilização de diversos tipos de imagens.

Durante os testes foi observada a fragilidade do conector para porta serial do Palm no que se refere a sua fixação para manter esta conexão. Muitos dos testes tiveram que ser feito com o Palm sobre uma mesa para evitar movimentos do conector e a consequente perda de conexão com o receptor GPS. Este, no entanto, tem um conector muito bom, com uma fixação excelente. A Figura 4.4 mostra o cabo de conexão e o local de encaixe no Palm. A ativação da conexão com o receptor GPS, o recebimento dos dados e seu tratamento não apresentaram problemas.



Figura 4.4: Conexão entre o Palm e o seu cabo serial.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

Englobar todas as possibilidades possíveis para atingir as funcionalidades necessárias ao ambiente da forma mais adequada é um trabalho que necessita um tempo bem mais longo do que o que foi utilizado neste. Mas foi possível tirar algumas conclusões, tais como:

- a utilização da imagem *raster* é mais vantajoso do que imagens vetoriais.
- a utilização de outras imagens *raster* para representar o zoom de determinadas áreas supre a limitação das API do PalmOS e acrescenta a possibilidade de usar também outros tipos de imagens, tais como imagens satélites e fotografias aéreas.
- a possibilidade de utilização de receptores GPS disponíveis nas Organizações Militares.
- a fragilidade da conexão do Palm com o receptor GPS, em virtude da fixação não robusta do conector do seu cabo serial, necessitando, provavelmente, de um conector semelhante aos existentes em receptores GPS desenvolvidos especificamente para Palm.
- a fragilidade do Palm que não possui qualquer resistência a quedas e a água, a despeito do receptor GPS ter baixo custo e, mesmo assim, ser resistente a impactos e cuja resistência à água permite permanecer a um metro de profundidade por trinta minutos. Isso não permite seu emprego incondicional, é necessário ter cuidados que podem tornar seu uso um fardo. A utilização de Palm militarizado é uma saída para este problema, mas eleva seu custo.
- a utilização deste equipamento permite uma aquisição mais rápida da posição da fração em relação às informações existentes no mapa, permitindo uma maior rapidez no emprego desta no momento e locais adequados.

- o transporte deste equipamento no lugar de extratos de planos, calcos e mapas, devido a seu baixo peso e pequenas dimensões possibilita um menor desgaste das frações em grandes deslocamentos não motorizados, bem como em terrenos de difícil transposição.

5.2 Trabalhos Futuros

Durante o trabalho surgiram algumas idéias para trabalhos futuros, abaixo relacionados:

- Implementação de desenho no mapa para a inclusão de dados gráficos, como uma nova estrada ou uma possível área de campo minado, bem como inclusão direto no mapa de símbolos militares padronizados. Usando, por exemplo, bitmap com cor de fundo transparente ou imagens vetoriais;
- Compressão dos dados na geração dos arquivos de Banco de Dados e durante a manuseio dos mesmos no PDA;
- Estudo sobre desempenho de outras possibilidades de divisão da imagem *raster*, inclusive, utilizando blocos de tamanhos variáveis dependendo da existência de detalhamentos no blocos;
- Aplicativo no PC que leia as informações alteradas no Palm para que elas possam ser utilizadas em operações futuras, como as informações referentes a alterações no terreno;

Referências Bibliográficas

- [Ale01] Márcio Alexandroni. Entenda o formato PDB (palm database), 2001. Disponível em <http://www.clubepalm.com.br/FormatoPDB.htm>. Acesso em 27 junho 2002.
- [BCC⁺01] Monty Boyer, John Cardozo, Gina Clark, Bob Ebert, David Fedor, Roger Flores, Rob Haitani, Jeff Hawkins, Michael Lunsford, Michael Mace, Jean Ostrem, Lon Poole, Chris Raff, Maurice Sharp, Phillip Shoemaker, Joe Sipher, and Carl Stone. *Zen of Palm*. Palm Inc., 2001.
- [Ben02] Peter Bennett. NMEA-0183 and GPS information, 2002. Disponível em <http://vancouver-webpages.com/pub/peter/index.html>. Acesso em 27 junho 2002.
- [BFH⁺01a] Christopher Bey, Elly Freeman, Gary Hillerson, Jean Ostrem, Ruben Rodriguez, Greg Wilson, and Mark Dugger. *Palm OS Programmer's Companion*. Palm, 2001.
- [BFH⁺01b] Christopher Bey, Elly Freeman, Gary Hillerson, Jean Ostrem, Ruben Rodriguez, and Greg Wilson. *Palm OS Programmer's API Reference*, 2001.
- [BFH⁺01c] Christopher Bey, Elly Freeman, Gary Hillerson, Jean Ostrem, Ruben Rodriguez, and Greg Wilson. *Palm OS Programmer Companion 2 Communications*. Palm Inc, 2001.
- [CCH⁺96] Gilberto Câmara, Marco A. Casanova, Andrea S. Hemerly, Geovane Cayres Magalhães, and Claudia Maria Bauzer Medeiros. *Anatomia de Sistemas de Informação Geográfica*. Escola de Computação, Campinas, 1996.
- [CDM⁺01] Gilberto Câmara, Clodoveu Davis, Antônio Miguel Vieira Monteiro, João Arge-miro Paiva, Júlio César Lima D'Alge, Suzana Fuks, Marco A. Casanova, and Ricardo Cartaxo. Geoprocessamento: Teoria e aplicações, 2001. Disponível em <http://www.dpi.inpe.br/gilberto/livro/>. Acesso em 27 junho 2002.
- [Cra01] Paulo Crawford. A teoria da relatividade e o GPS. *O Observatório*, 7(4), 2001. Disponível em <http://astro.cc.fc.ul.pt/oobservatorio/vol7/n4>. Acesso em 27 junho 2002.

- [ESA02] ESA. What is galileo?, 2002. Disponível em http://www.esa.int/export/esaSA/GGGMX650NDC_navigation_0.html. Acesso em 27 junho 2002.
- [ESR98] ESRI. ESRI Shapefile Technical description. Technical report, Environmental Systems Research Institute, Inc., 1998.
- [Fos00] Lonnon R. Foster. *Palm OS Programming Bible*. IDG Books Worldwide, Inc., California, 2000.
- [Gar99] Garmin. GARMIN GPS interface specification, 1999. Disponível em <http://www.garmin.com/products/etrex/spec.html>. Acesso em 27 junho 2002.
- [Gar00] Garmin. eTrex personal navigation: Owner's manual, 2000. Disponível em <http://www.garmin.com/manuals/etrex.pdf>. Acesso em 27 junho 2002.
- [GLO00] GLONASS. Global navigation satellite system, 2000. Disponível em <http://mx.iki.rssi.ru/SFCSIC/english.html>. Acesso em 27 junho 2002.
- [Gor02] Miguel Gorgulho. G.p.s. - o sistema de posicionamento global. *GPS Global*, 2002. Disponível em <http://www.gpsglobal.com.br/>. Acesso em 27 junho 2002.
- [Har00] Tenente-Coronel James E. Harris. Combater digital ou analogicamente. *Military Review*, 2000.
- [Hil01] Gary Hillerson. *Palm File Format Specification*. Palm Inc., 2001.
- [JA00] Nelson R. Gasparin Júnior and Fábio Durante Pereira Alves. Global positioning sistem (GPS), 2000.
- [Jún02] Odilon Ferreira Júnior. GPSTrackMaker, 2002. Disponível em <http://www.gpstm.com.br>. Acesso em 27 junho 2002.
- [Mar97] Paulo Marques. Estrutura do formato de ficheiro bitmap, 1997. Disponível em <http://www.deetc.isel.ipl.pt/analisedesainai/Pessoais/PauloMarques/utills/bitmap.pdf>. Acesso em 27 junho 2002.
- [Mes01] Renata Mesquita. Palm entra na guerra do afeganistão. *InfoExame*, 2001.
- [Mon00] João Francisco Galera Monico. *Posicionamento Pelo NAVSTAR-GPS: Descrição, Fundamentos e Aplicações*. Editora UNESP, Praça da Sé, 108, 01001-900 - São Paulo - SP, 2000.
- [MR01] Steve Mann and Ray Rischpater. *Advanced Palm Programming, an Expert's Guide to Creating Killer Application*. Wiley Computer Publishing, 2001.

- [MV96] James D. Murray and William VanRyper. *Encyclopedia of Graphics File Formats, Second Edition*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, 1996.
- [MY98] Joe Mehaffey and Jack Yeazel. Joe mehaffey and jack yeazel's GPS information, 1998. Disponível em <http://joe.mehaffey.com/>. Acesso em 27 junho 2002.
- [NCG00] NCGIA. The NCGIA core curriculum in GIScience, 2000. Disponível em <http://www.ncgia.ucsb.edu/education/curricula/giscc/>. Acesso em 27 junho 2002.
- [Pal01] Palm. Graphics, animation, and games, 2001. Disponível em <http://oasis.palm.com/dev/kb/papers/1148.cfm>. Acesso em 27 junho 2002.
- [Red97] Martin Reddy. The graphics file formats page, 2d bitmap specifications, 1997. Disponível em <http://www.dcs.ed.ac.uk/home/mxr/gfx/2d-hi.html>. Acesso em 27 junho 2002.
- [RM99] Neil Rhodes and Julie McKeehan. *Palm Programming: The Developer's Guide*. O'Reilly and Associates, Inc., 1999.
- [Wag02] Jan-Oliver Wagner. FreeGIS, 2002. Disponível em <http://www.freegis.org/>. Acesso em 27 junho 2002.
- [War99] Frank Warmerdam. Shapefile c library v1.2, 1999. Disponível em <http://gdal.velocet.ca/projects/shapelib/shapelib.html>. Acesso em 27 junho 2002.
- [Yea98] Jack Yeazel. FAQ: WHAT IS UTM?, 1998. Disponível em <http://joe.mehaffey.com/utm-faq.html>. Acesso em 27 junho 2002.
- [Zha01] Chunyu Zhang. UMap - a raster-based campus map application on palm. Master's thesis, University of Minnesota, 2001.