

CLAUDIA ROSSI

Implementação de Interfaces para Entrada de Dados
Obtidos com um Estereorrestituído Analítico
num Sistema de Informações Geográficas

Dissertação apresentada ao Curso de Pós-
Graduação em Ciências Geodésicas do
Setor de Tecnologia da Universidade Fe-
deral do Paraná, como requisito parcial à
obtenção do grau de Mestre.

CURITIBA
1993

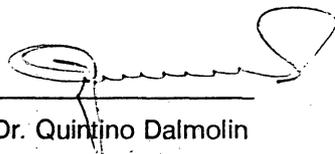
IMPLEMENTAÇÃO DE INTERFACES PARA ENTRADA DE DADOS
OBTIDOS COM UM ESTEREORESTITUIDOR ANALÍTICO NUM
SISTEMA DE INFORMAÇÕES GEOGRÁFICAS

por

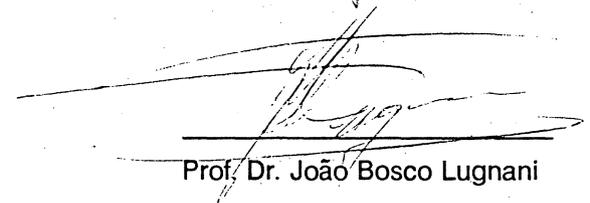
CLAUDIA ROBBI

Dissertação aprovada como requisito parcial para obtenção
do grau de Mestre no Curso de Pós-Graduação em Ciências
Geodésicas, pela Comissão formada pelos professores:

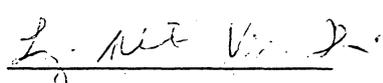
ORIENTADOR:



Prof. Dr. Quintino Dalmolin



Prof. Dr. João Bosco Lugnani



Dr. Luiz Alberto Vieira Dias

Curitiba, 12 de março de 1993.

Aos familiares e amigos sempre presentes

Prof. Dr. João Bosco Lugnani e Dr. Luiz Alberto
Vieira Dias pela confiança em mim depositada.

Empresa Esteio Engenharia e Aerolevantamentos S.A.
pela permissão de uso de equipamentos, manuais e
resultados de restituição, sem a qual este trabalho
não seria realizado.

Eng. Guaraci José Erthal pela colaboração nas
discussões sobre o SGI.

SUMÁRIO

<u>LISTA DE TABELAS</u>	vii
<u>LISTA DE FIGURAS</u>	viii
<u>RESUMO</u>	xi
<u>ABSTRACT</u>	xii
<u>1 INTRODUÇÃO</u>	1
<u>2 A IMPORTÂNCIA DA ENTRADA DE DADOS FOTOGRAMÉTRICOS PARA UM SIG</u>	3
2.1 SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS	3
2.2 ESTRUTURAS DE DADOS PARA SISTEMAS DE BANCO DE DADOS	7
2.2.1 Estruturas de dados para o nível conceitual	14
2.2.1.1 Banco de dados relacional	16
2.2.1.2 Banco de dados em rede	18
2.2.1.3 Banco de dados hierárquico	21
2.3 ESTRUTURAS DE DADOS PARA DADOS GEOGRÁFICOS	22
2.4 ENTRADA DE DADOS FOTOGRAMÉTRICOS EM SIG	29
<u>3 ARQUITETURA DO SOFTWARE E ENTRADA DE DADOS DO SGI-INPE</u>	34
3.2 ESTRUTURAS DE DADOS	38
3.2.1 Estrutura de dados para os níveis externo	38
3.2.2 Estrutura da base de dados	41
3.3 MÓDULO DE APLICATIVOS	44
3.4 MÓDULO DA BIBLIOTECA BÁSICA	47

7.2.1	Entrada de dados de mapeamento	83
7.2.2	Entrada de dados para MNT	84
8	<u>RESULTADOS</u>	86
8.1	ENTRADA NO SGI DOS DADOS DA RESTITUIÇÃO DA FOLHA 181 DO PROJETO CODEPLAN	88
8.2	ENTRADA NO SGI DA RESTITUIÇÃO DO EXEMPLO COM FEIÇÕES FICTÍCIAS	97
8.3	ENTRADA NO SGI DE ISOLINHAS PARA MNT	98
8.4	ENTRADA NO SGI DE PONTOS ISOLADOS PARA MNT	100
9	<u>CONCLUSÕES E SUGESTÕES PARA ESTUDOS FUTUROS</u>	103
	ANEXOS	106
	REFERÊNCIAS BIBLIOGRÁFICAS	129

LISTA DE TABELAS

1	Exemplo de tipo de associação e tipo de entidade	15
2	Exemplo de tipo de entidade e tipo de propriedade	16
3	Tipo e formato dos dados para arquivos gerais GEFIL	51
4	Estrutura do arquivo geral para o modo PONTO	54
5	Estrutura do arquivo geral para o modo VECTOR	56
6	Estrutura do arquivo geral para os modos SPLINE e TRACING	56
7	Estrutura do arquivo geral para o modo ARC	57
8	Estrutura do arquivo geral para o modo CIRC	58
9	Formato de dados para MNT	60

LISTA DE FIGURAS

1	Principais componentes de <u>hardware</u> de um SIG	4
2	Módulos de <u>software</u> de um SIG	6
3	Aspectos organizacionais de Sistemas de Informações Geográficas	7
4	Lista generalizada	9
5	Amostra de árvore	9
6	Grafo dirigido	10
7	Os três níveis da arquitetura de um banco de dados	12
8	Processamento de dados com Sistema de Banco de Dados	13
9	Dados na forma relacional	18
10	Representação de estrutura em rede do modelo CODASY	19
11	Dados estruturados em rede	20
12	Dados estruturados hierarquicamente	22
13	Níveis de estruturação de dados de um SIG	23
14	Ligação de dados temáticos e dados geométricos	24
15	Estrutura de dados formal para mapas vetoriais	28
16	Relações topológicas das feições	28
17	Esquema dos blocos principais do <u>software</u> SGI-INPE	36
18	Parte de um arquivo ASCII de menu	37
19	Estrutura de dados para os níveis externo e conceitual do SGI-INPE	38
20	Relações topológicas previstas no SGI	39
21	Estruturas para dados gráficos do SGI-INPE: (a)dados matriciais; (b)dados vetoriais.	40

22	Categoria de dados, entidade manipulável e formato de representação dos dados no SGI . . .	41
23	Organização da base de dados do SGI	43
24	Formato de dados de um arquivo ASCII de linhas para ser lido pelo SGI	45
25	Formato de dados de arquivos ASCII a serem lidos pelo SGI para a categoria de dados MNT: (a) isolinhas; (b) pontos isolados.	46
26	Configuração de <u>hardware</u> para medidas interativas com o programa DZ7-AS	49
27	Um registro de um arquivo geral do PLANICOMP C120	52
28	Possibilidades de interfacemento entre o PLANICOMP e o SGI. (a) Ligação física; (b) Ligação por <u>software</u> dentro do ambiente do SGI; (c) Ligação por <u>software</u> fora do ambiente do SGI.	63
29	Tela do menu principal do ambiente SGI	64
30	Tela do menu do submódulo ENTRADA	65
31	Tela do menu do submódulo ENTRADA DE ÁREA	66
32	Submódulo ENTRADA DE MNT	67
33	Submódulo ENTRADA DE MNT com menu para dados do PLANICOMP	67
34	Incremento angular entre dois pontos de uma circunferência	74
35	Sentido de rotação do incremento angular α	75
36	Algoritmo da interface para dados poligonais	77
37	Folha de controle da restituição	81
38	Folha de controle da restituição	87
39	Ativação do Projeto - nome do projeto	88
40	Ativação do projeto - projeção	89
41	Ativação do Projeto - escala	89
42	Ativação do Projeto - datum horizontal e coordenadas envolventes	90
43	Ativação do Projeto - retângulo envolvente	90
44	Ativação do plano de informação - nome do plano	91
45	Ativação do plano de informação - parâmetros do PI	91

46	Leitura do arquivo de restituição	92
47	Armazenamento da restituição das estradas pavimentadas no SGI	93
48	Armazenamento da restituição das estradas pavimentadas, vicinais e construções no SGI ..	93
49	Armazenamento da restituição da planimetria no SGI	94
50	Armazenamento da restituição das curvas de nível mestras no SGI	94
51	Armazenamento da restituição editada das estradas pavimentadas, vicinais e construções ..	95
52	Ampliação de um detalhe das estradas pavimentadas, vicinais e construções	96
53	Armazenamento da restituição editada da planimetria no SGI	96
54	Resultado do teste usando os modos de digitalização aceitos pela interface	98
55	Ativação do plano de informação - parâmetros do PI	99
56	Imagem tridimensional gerada a partir das curvas de nível mestras	100
57	Armazenamento de pontos obtidos por perfilamento de um modelo fotogramétrico no PLANICOMP	101
58	Isolinhas e imagem sintética geradas a partir da leitura de pontos de perfil	102
59	Imagem tridimensional do terreno obtida a partir das curvas de nível da figura 58.	102

RESUMO

Apesar do atual estágio de desenvolvimento da tecnologia computacional, a coleta de dados em Sistemas de Informações Geográficas - SIG permanece cara. Entre as principais razões para o uso de instrumentos fotogramétricos na digitalização de feições está o registro de posições precisas a partir de fotografias aéreas em contraste à digitalização de mapas existentes. A atual configuração do SIG desenvolvido pelo INPE - SGI-INPE permite para a entrada de dados digitalizar mapas existentes e adquirir imagens processadas com o Sistema de Processamento de Imagens do INPE - SITIM-SR. Devido a estas considerações, este trabalho desenvolveu interfaces para a entrada de dados fotogramétricos obtidos com o estereorestituídor analítico PLANICOMP C120 no SGI-INPE. Três soluções para concretizar esta proposta foram detectadas como possíveis: on-line, em batch externa ao ambiente do sistema ou em batch interna ao ambiente do SGI. Da análise das metodologias de trabalho em ambos os sistemas para o uso das interfaces em cada possível solução, as melhores justificativas direcionaram para a entrada de dados em batch com interface inserida no ambiente do SGI, considerando que esta implica em menores alterações na operacionalidade tanto do PLANICOMP como do SGI.

ABSTRACT

In spite of the present development of computational technology, data capture of Geographic Information System - GIS - remains expensive. Among many reasons for the entity digitalization with photogrammetric instruments is the register of precise positions from aerial photographs in contrast with digitalization of maps that already exists. The present GIS configuration developed by INPE, allows data input by the digitalization of existent maps and by getting the images processed by the Image Processing System of INPE -SITIM-SR. Due to the considerations above, this work developed interfaces for photogrammetric data input which were gotten from the PLANICOMP C120 analytical stereo-plotter. Three possible solutions for this proposal were: on-line, external batch in the environment system and internal batch in SGI environment. By analysing the work methodology in both systems, using the interface in each possible solution, we can say that the best justification leads to batch data entry with interface inserted in SGI environment, as less operational alteration is implied in both PLANICOMP and SGI.

1 INTRODUÇÃO

Apesar do constatado barateamento de softwares e hardwares, conseqüente do constante desenvolvimento da tecnologia computacional, o custo da coleta de dados para Sistemas de Informações Geográficas -SIG - permanece caro, se comparado ao custo total de implantação e manutenção do sistema, merecendo por isso especial atenção.

Na transição entre a tecnologia convencional nas atividades de mapeamento para a tecnologia digital, quando esta é absorvida por empresas produtoras ou usuárias de produtos cartográficos, tanto produtos convencionais como digitais são necessários na formação e manutenção das bases cartográficas dos SIG's. Isto porque enquanto os produtos digitais substituem gradativamente os convencionais, estes ainda constituem parte do acervo destas empresas ou instituições. Assim, nesta fase de transição, a digitalização de produtos convencionais é uma atividade necessária em SIG's.

No entanto, a tecnologia computacional utilizada em ciências como a Fotogrametria Analítica e o Sensoriamento Remoto viabilizou a obtenção de produtos cartográficos na forma digital. Conseqüentemente, os SIG's, além de possibilitar a digitalização de produtos convencionais, devem fornecer meios para a coleta de dados digitais na formação de suas bases cartográficas. M. M. ALLAM|1| ao descrever as principais razões para o uso de instrumentos fotogramétricos na digitalização de feições, coloca entre estas: o registro de posições precisas a partir de fotografias aéreas em contraste à digitalização de mapas existentes; a possibilidade de preparar mapas em diversas escalas; a aceleração da atualização cartográfica.

Com base nestas constatações, e considerando que a atual configuração do Sistema de Informações Geográficas desenvolvido pelo INPE, chamado SGI-INPE ou simplesmente SGI, permite para a entrada de dados, digitalizar mapas existentes e adquirir imagens processadas com o Sistema de Processamento de Imagens do INPE - SITIM-SR, porém não tem a opção de entrada de dados fotogramétricos, delineou-se como meta para este trabalho o desenvolvimento de interfaces para entrada

de dados fotogramétricos obtidos com o estereorestituidor analítico PLANICOMP C120. Três soluções para concretizar esta proposta foram detectadas como possíveis: on-line, em batch externa ao ambiente do software SGI ou em batch interna ao ambiente do SGI. Da análise das metodologias de trabalho em ambos os sistemas para o uso das interfaces em cada possível solução, as melhores justificativas direcionaram para a entrada de dados em batch com interface interna ao ambiente do SGI, considerando que esta implica em menores alterações na operacionalidade tanto do PLANICOMP como do SGI.

Por utilizar conceitos comumente não familiares aos Engenheiros Cartógrafos como SIG, estruturas de dados e estruturas de dados para dados geográficos, estes assuntos foram abordados, na descrição do embasamento teórico, com certo grau de detalhe, visando: servir como fonte de consulta rápida de tais assuntos; e apoiar o entendimento do trabalho. Além destes tópicos, a revisão bibliográfica, apresentada no capítulo 2 trata em seu último item da entrada de dados fotogramétricos em SIG's.

Na sequência do trabalho, a descrição do SGI abordando a estrutura do software, e conseqüentemente o planejamento da base de dados e estruturação destes compõe o capítulo 3. O capítulo 4 é destinado a apresentação do sistema PLANICOMP C120, enfatizando suas potencialidades para digitalização a partir de instrumentos fotogramétricos, e o armazenamento destas em arquivos ASCII.

A apresentação das possíveis soluções ao problema proposto, o detalhamento e operacionalidade da solução adotada, bem como sua justificativa, encontram-se no capítulo 5. Ao capítulo 6 é reservada a descrição da implementação das interfaces responsáveis pela entrada de dados fotogramétricos no SGI. O capítulo 7, sob o título "Metodologia de trabalho para entrada de dados adquiridos com o PLANICOMP C120 no SGI" apresenta aos usuários do SGI quais informações devem ser fornecidas com a restituição fotogramétrica, quais seus significados no SGI, e como proceder para usar as funções, neste trabalho desenvolvidas. Para testar as interfaces, três exemplos foram adotados, cujos resultados encontram-se no capítulo 8, acompanhados das conclusões observadas no decorrer dos testes. Finalizando este trabalho, o capítulo 9 descreve as conclusões e sugestões para estudos futuros.

2 A IMPORTÂNCIA DA ENTRADA DE DADOS FOTOGRAFÉTRICOS PARA UM SIG

2.1 SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS

O termo sistemas de informações geográficas - SIG - embute conceitos de sistema, de informação, de sistema de informação, e de informação geográfica. Sistema é um conjunto de ferramentas capaz de realizar um objetivo ou objetivos intencionados. Estas ferramentas podem ser outros sistemas, ou seja, subsistemas com objetivos específicos que, quando realizados, permitam que a finalidade principal seja alcançada [5]. Conceituando informação M. MOLENAAR [14] afirma que é a troca de conhecimento, sendo este todo o saber do indivíduo sobre determinado tópico ou área. A troca ou transferência de informação gera comunicação, a qual pode acontecer diretamente entre pessoas, ou através dos Sistemas de Informações. Estes sistemas devem ser capazes de armazenar informação, e responder às arguições dos usuários, a fim de informá-los sobre determinados assuntos. As informações armazenadas, ou seja, a representação destas no sistema são os dados; e o armazenamento, recuperação e atualização destes dados compete ao sistema de gerenciamento da base de dados que, por esta razão, deve ser um dos componentes do sistema de informação.

As descrições dos objetos, fenômenos ou processos que ocorrem na superfície terrestre compõem as informações geográficas, objeto da Cartografia. Estas feições, tanto naturais como artificiais, são posicionadas em relação a um sistema de coordenadas, uma vez que suas localizações devem ser conhecidas, juntamente com seus atributos. Sendo o espaço físico dinâmico, a tarefa de representar a superfície terrestre exige que este dinamismo seja contemplado. Por isso, e pelas informações geográficas serem obtidas num determinado instante, a atualização destas deve ser constante, garantindo assim um dos fatores necessários à confiabilidade das informações.

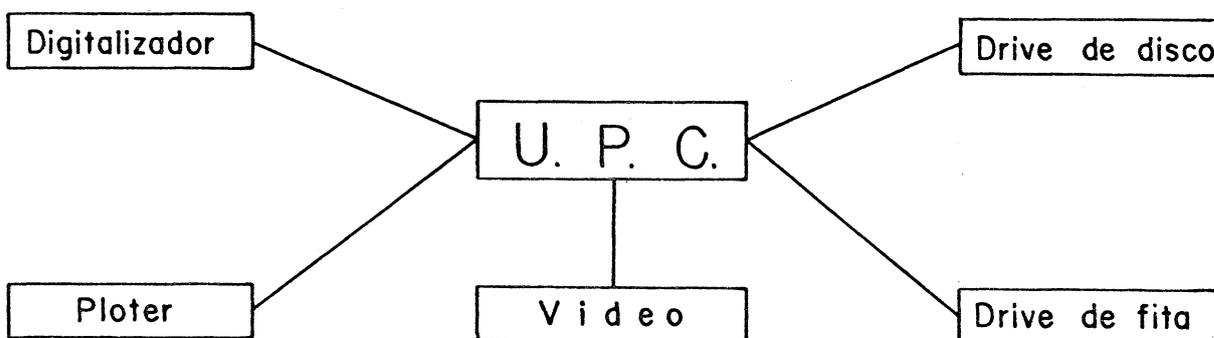
Os objetivos almejados por um determinado sistema são consequências das necessidades específicas de um conjunto de usuários. Assim, os objetivos de um SIG são definidos em função dos

usuários de informações espaciais que, como cita P. A. BURROUGH|4|, não são apenas cientistas da Terra, mas também planejadores urbanos e agências de cadastro territorial, quando utilizam informações sobre uso e recursos do terreno; engenheiros civis no planejamento de rodovias e canais; departamentos de polícia na distribuição espacial de vários tipos de crimes; organizações médicas no que diz respeito às ocorrências de doenças e epidemias; comerciantes na distribuição de mercados existentes e potenciais; concessionárias de serviços públicos, como saneamento básico, distribuição de energia elétrica e linhas telefônicas, no planejamento e manutenção destas.

Portanto SIG, como definido por P. A. BURROUGH|4|, "é um conjunto poderoso de ferramentas para coletar, armazenar, recuperar, transformar e apresentar dados espaciais do mundo real, para um conjunto particular de propósitos." Estes sistemas são compostos por três importantes componentes: hardware computacional, conjuntos de módulos de software, e um contexto organizacional apropriado.

O hardware computacional compõe-se de equipamentos capazes de permitir: a entrada de diferentes espécies de dados gráficos e não-gráficos, sendo estes basicamente, digitalizadores, drive de disco e drive de fita; a visualização das informações geográficas, tanto originalmente armazenadas como resultantes de processos de análises, na forma de produtos convencionais obtidos pelos plotters, ou numa unidade de vídeo; e o processamento dos dados armazenados através de uma Unidade de Processamento Central (fig.1)|4|.

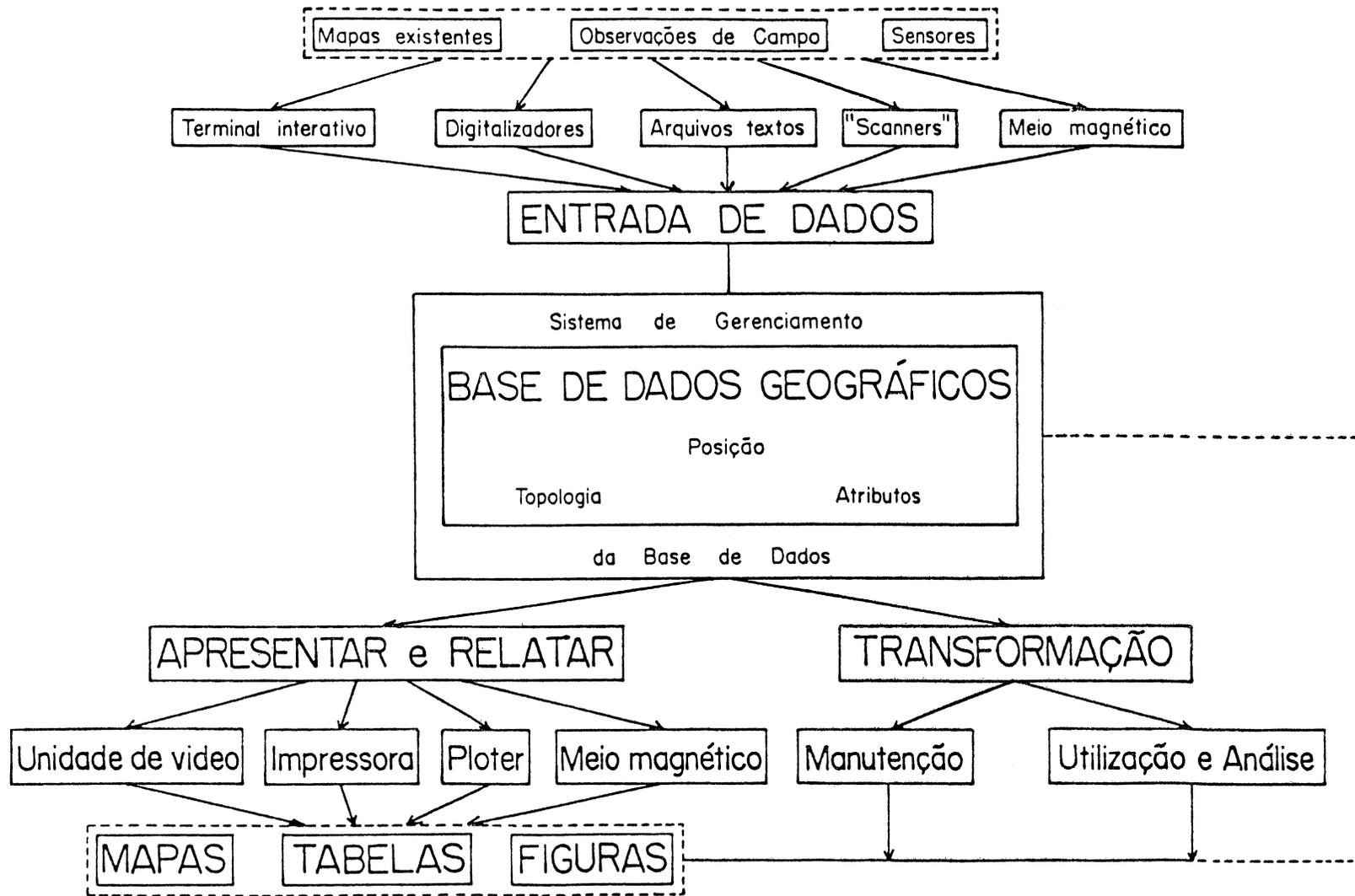
Figura 1: Principais componentes de hardware de um SIG



Fonte: P. A. BURROUGH|4| p.7

Para realizar as tarefas necessárias de um SIG, o pacote de software deve possuir cinco módulos, sendo estes:

- a) módulo de entrada e verificação dos dados, o qual permite que dados gráficos (mapas existentes, fotografias aéreas, imagens de satélites, resultados de levantamentos terretres, listas de coordenadas) e não-gráficos (atributos das feições, tais como vazão de rio, tipo de pavimento de rodovias) de diferentes fontes sejam transformados para uma forma digital compatível;
- b) módulo de armazenamento dos dados e gerenciamento da base de dados, responsável pela estruturação e organização destes, tanto para sua manipulação no computador como sua percepção pelo usuário. Portanto, este módulo deve incluir o Sistema de Gerenciamento da Base de Dados (DBMS).
- c) módulo de saída e apresentação dos dados, através do qual tanto os dados, como os resultados das análises destes, são apresentados aos usuários. Esta apresentação pode ser em forma de mapas, tabelas ou figuras;
- d) módulo de transformação dos dados, compreendendo dois conjuntos de operações, ou seja, as operações de depuração de erros, atualização e compatibilização dos dados; e as que permitem análises a partir dos dados armazenados, cujos resultados responderão às questões formuladas pelos usuários;
- e) módulo de interação com o usuário, que deve prever e conseqüentemente, permitir a formulação de questões cujas repostas atendam às necessidades dos usuários. Estas questões, estando relacionadas às informações geográficas, englobam desde informações obtidas convencionalmente de cartas, até informações resultantes de processos de análise e modelamento realizados com dados geográficos | 4 |.

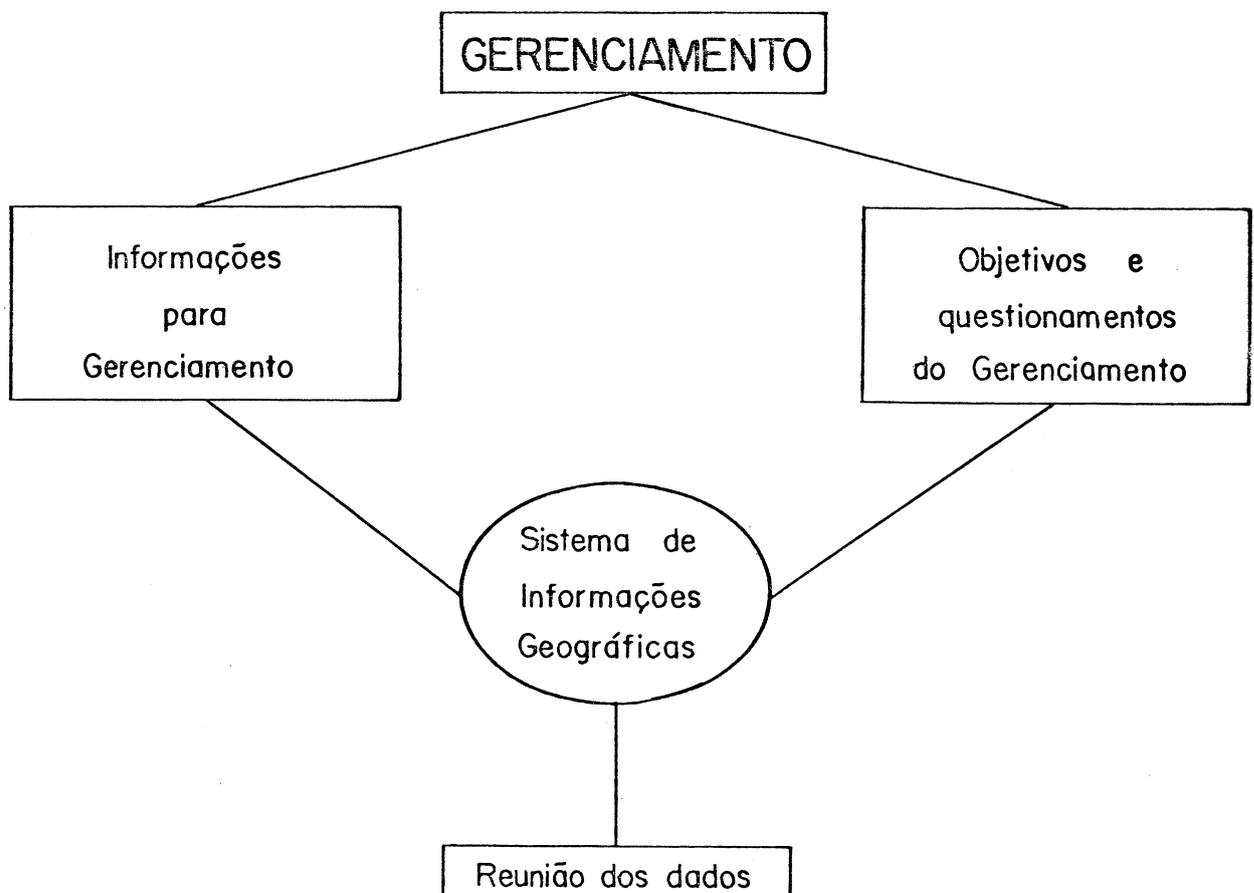


Fonte: P. A. BURROUGH | 4 | pp.8 e 9

Figura 2: Módulos de software de um SIG

Em qualquer sistema, um objetivo dominante é a eficiência que, segundo Churchman [5], no contexto do pensamento sistêmico significa o "melhor modo" de realizar uma tarefa, tendo em vista os objetivos almejados. No caso de um SIG, reportando novamente a P. A. BURROUGH [4], não é suficiente a aquisição do hardware e software adequados, mas que estes estejam inseridos num contexto organizacional apropriado; o que exige capacitação tanto do corpo técnico, como do gerenciamento atuante no sistema.

Figura 3: Aspectos organizacionais de Sistemas de Informações Geográficas



Fonte: P. A. BURROUGH [4] p.10

2.2 ESTRUTURAS DE DADOS PARA SISTEMAS DE BANCO DE DADOS

Num sistema de informações apoiado em computadores, não são definidos apenas os objetos de dados e as operações sobre estes, mas também as relações entre os conjuntos de

objetos. Isto exige a utilização de estruturas de dados, conceituadas por E. HOROWITZ & S. SAHNI | 10 | como "um conjunto de domínios D, um conjunto de funções F, e um conjunto de axiomas A. O triplo (D,F,A) caracteriza a estrutura de dados." O conjunto de axiomas descreve as semânticas das operações.

Tanto os dados, como as relações entre eles, são armazenados computacionalmente com base nos conceitos de campo, registro e arquivo. Campo é a menor unidade de dados identificável, no qual é armazenado um item de dado. Um conjunto com nome, destes campos associados é chamado de registro. Uma coleção, também com nome, de todas as ocorrências de um tipo de registro constitui um arquivo | 7 |. Ao conjunto de arquivos relacionados entre si denomina-se Base de Dados, e um conjunto de Base de Dados é chamado de Banco de Dados, ressaltando que alguns autores consideram ambos os conceitos equivalentes | 15 |.

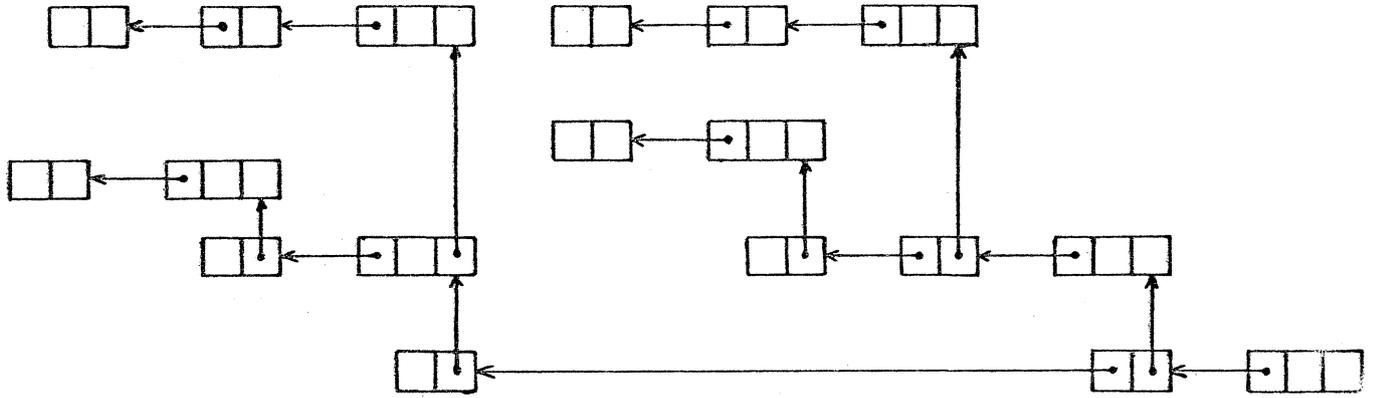
As estruturas de armazenamento dos dados, segundo o modo de armazenamento dos objetos, são classificadas em:

- a) estruturas nas quais o endereçamento da memória é implicitamente determinado, e conseqüentemente, os objetos de dados são armazenados em espaçamentos de memória fixos, definidos a partir do tipo de dado. Tais estruturas são chamadas por N. WIRTH | 18 | de estruturas básicas de dados, sendo estas vetores, matrizes, registros, conjuntos e sequências. Já E. HOROWITZ & S. SAHNI | 10 | as denominam de arranjos;
- b) estruturas que exigem espaçamentos variáveis de memória e, por isso, os objetos de dados podem ser armazenados em qualquer localização da memória, utilizando endereçamento explícito, através de ponteiros. N. WIRTH | 18 | as denomina de estruturas dinâmicas de dados, pois além dos valores das variáveis, a própria estrutura dos dados pode ser alterada durante o processamento.

As estruturas dinâmicas de dados, de acordo com suas complexidades, podem ser subdivididas em três grupos, definidos por E. HOROWITZ & S. SAHNI | 10 | como:

- a) "Lista generalizada A é uma seqüência finita de $n \geq 0$ elementos $\alpha_1, \dots, \alpha_n$, onde α_i são partículas ou listas. Os elementos que não são partículas são denominados de sublistas de A."

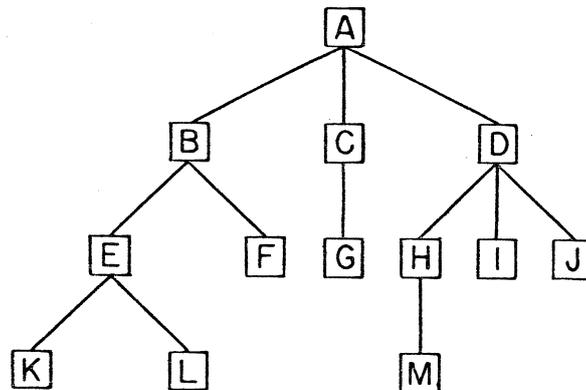
Figura 4: Lista generalizada



Fonte: E. HOROWITZ & S. SAHNI | 10 | p.146

- b) "Árvore é um conjunto finito de um ou mais nós de tal natureza que: (i) existe um nó especialmente denominado de raiz; (ii) os nós restantes estão desdobrados em $n \geq 0$ conjuntos separados T_1, \dots, T_n , em que cada um dos referidos conjuntos constituem uma árvore denominada de subárvore da raiz."

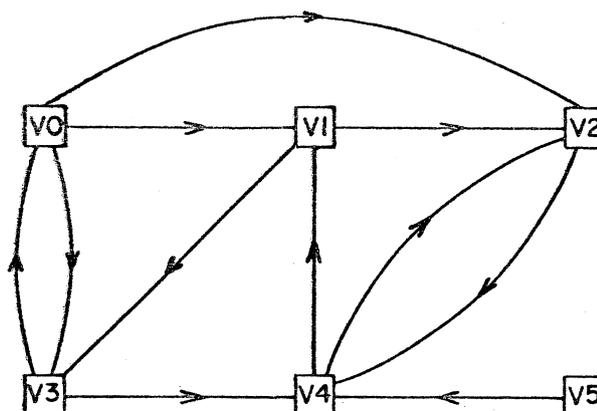
Figura 5: Amostra de árvore



Fonte: E. HOROWITZ & S. SAHNI | 10 | p.199

- c) "Grafo G consiste de dois conjuntos V e E. V é um conjunto de vértices não vazio. E é um conjunto de pares de vértices, sendo estes denominados de bordas." Se os vértices forem ordenados para individualizar as bordas, um grafo dirigido é definido. Neste caso, se uma borda é representada pelos vértices (v_1, v_2) , (v_2, v_1) será uma borda diferente desta.

Figura 6: Grafo dirigido



Fonte: E. HOROWITZ & S. SAHNI | 10 | p.269

Com a evolução dos sistemas de informações, os arquivos convencionais que serviam a aplicações específicas, e que trabalhavam com estruturas lógicas semelhantes às estruturas físicas (fig. 8a), foram substituídos por sistemas que utilizam estruturas de armazenamento mais complexas (fig. 8b). Estas permitem a integração de diversas aplicações, objetivando controlar a redundância e centralização dos dados. Com isso, as estruturas de armazenamento passaram a ficar mais distantes das estruturas lógicas. Conseqüentemente, apesar do conteúdo das informações de uma base de dados ser único, existem diferentes níveis de estruturação destes dados. Esta diferenciação preserva os usuários da necessidade de conhecimento sobre o armazenamento físico dos dados e, portanto, das diferentes arquiteturas de hardware. Assim, o usuário trabalha sobre os dados independentemente da máquina, utilizando estruturas de dados que representam abstrações da realidade definidas a partir

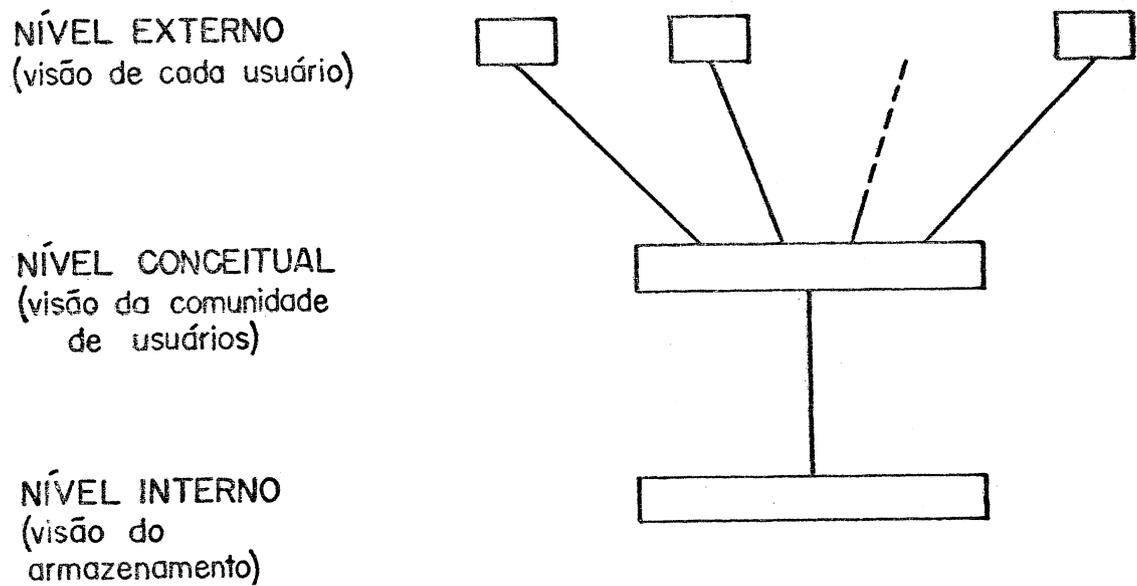
de suas necessidades. Além da possibilidade de definir representações que atendam a diferentes propósitos, estas estruturas podem ser generalizadas para usuários com finalidades comuns.

C. J. DATE [7] quando propõe uma arquitetura para sistemas de banco de dados, a divide em três níveis gerais, ou seja, o nível interno, que está mais próximo do armazenamento físico, o nível externo, representando como os dados são percebidos pelos usuários, e o nível conceitual, definido como o nível de simulação entre o externo e o interno.

Para cada nível do banco de dados há um correspondente esquema, o qual engloba as definições de cada registro daquela visão. Portanto, a visão externa é composta de registros externos, também chamados de registros lógicos, definidos no esquema externo. A visão conceitual, que representa o conteúdo total da base de dados, consiste de todas as ocorrências dos vários tipos de registros conceituais definidos no esquema conceitual. O mesmo raciocínio é realizado para a visão interna, porém como esta corresponde ao nível mais baixo da estrutura proposta, os registros internos são registros armazenados, e o esquema interno, além de defini-los, acrescenta especificações necessárias ao armazenamento físico dos dados, tais como: quais os índices existentes, como os campos armazenados estão representados, qual a sequência física dos registros armazenados. Consequentemente, a visão interna corresponde ao banco de dados armazenado, e o esquema interno à definição da estrutura de armazenamento [7].

A correspondência entre níveis consecutivos é chamada de mapeamento. Assim, as relações entre os registros e campos conceituais e seus correspondentes armazenados compõem o mapeamento conceitual/interno; e as correspondências entre a visão externa e a conceitual são definidas no mapeamento externo/conceitual. Neste ponto, cabe ressaltar que na maioria dos sistemas atuais, os níveis conceitual e externo são muito semelhantes, ou até mesmo iguais, pois os usuários possuem conhecimento completo da representação conceitual dos dados, referente à realidade por eles trabalhada [7].

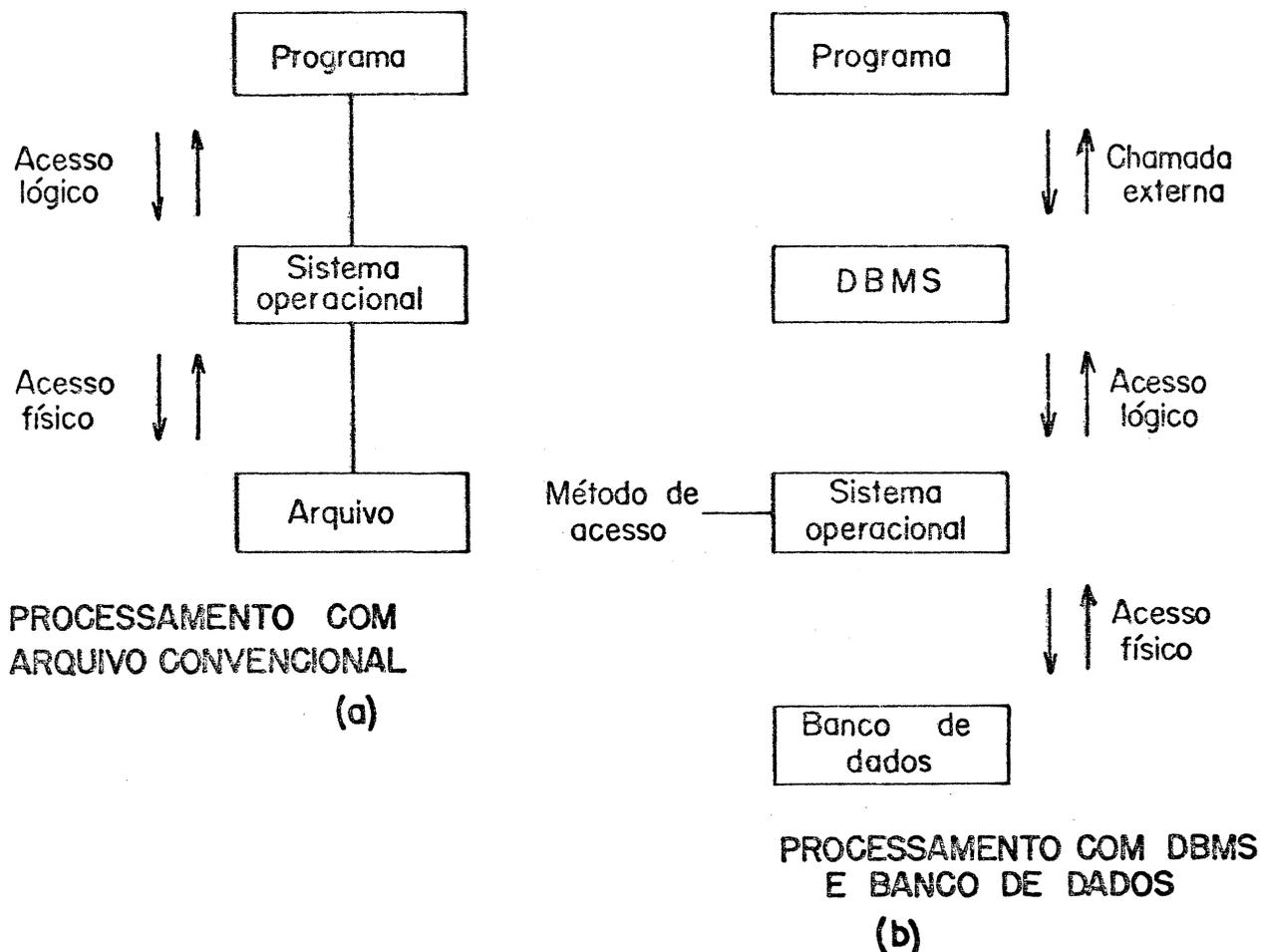
Figura 7: Os três níveis da arquitetura de um banco de dados



Fonte: C. J. DATE |7| p.38

A tarefa de isolar o usuário do nível do hardware é responsabilidade do sistema de gerenciamento da base de dados (DBMS). Além de permitir que o usuário utilize estruturas lógicas simples, o DBMS também objetiva a eficiência na utilização dos meios de armazenamento |13|. Com isso, os programas de aplicação não têm acesso direto ao armazenamento dos dados, diferenciando os processamentos com arquivos convencionais, dos processamentos com DBMS e Banco de Dados, o que pode ser visualizado na figura 8. Para isso, segundo C. J. DATE |7|, a cada solicitação do usuário, o DBMS a intercepta e interpreta, em seguida consulta o esquema externo, o mapeamento externo/conceitual e a definição da estrutura de armazenamento, e então, executa as operações necessárias na base de dados.

Figura 8: Processamento de dados com Sistema de Banco de Dados



Fonte: C. S. YONG | 15 | p.18

Porém, o DBMS não opera no nível físico do armazenamento, sendo a conversão do nível interno para o físico responsabilidade do método de acesso. Este contém um conjunto de rotinas que realizam a interface entre o registro armazenado e o DBMS. Devido ao fato do DBMS e do método de acesso exercerem diferentes funções no sistema, diferentes tipos de informações são exigidas para cada um, as quais Date | 7 | relaciona como:

- a) informações para o DBMS, que devem ser especificadas na estrutura de armazenamento:
 - i) que arquivos armazenados existem; e para cada arquivo:
 - ii) a estrutura do correspondente registro armazenado;

- iii) os campos armazenados, e quais são usados em seqüências, se estas existem;
 - iv) os campos armazenados que podem ser usados como argumentos de pesquisa para acesso direto.
- b) informações para o método de acesso:
- i) sobre blocos físicos;
 - ii) como os campos associados estão armazenados na formação dos registros armazenados;
 - iii) como a seqüência é realizada: contigüidade física, índice ou cadeia de indicadores de localização;
 - iv) como é executado o acesso direto.

Portanto, após gerada uma ocorrência de registro armazenado, esta é inserida no banco de dados através de um único endereço de registro armazenado, criado pelo método de acesso. Uma vez este endereço criado, o próprio método de acesso transmite esta informação ao DBMS, que pode com isso, estabelecer seu mecanismo de acesso aos dados.

2.2.1 Estruturas de dados para o nível conceitual

Decidir sobre a estrutura de dados do nível conceitual depende dos objetivos almejados, os quais definirão a realidade na qual o sistema estará inserido. Isto exige que duas etapas sejam cumpridas: percepção do mundo real e representação deste através do modelo conceitual. A percepção do mundo real, composto de entidades e relacionamentos entre estas, envolve segundo C. S. YONG [15], a observação de regras e comportamentos dos elementos básicos que compõem a realidade a ser modelada, e conseqüentemente a descrição destes elementos, feita através dos seguintes conceitos básicos:

- a) ENTIDADE: objeto com existência própria e comportamento constante. Do conceito de entidade define-se TIPO DE ENTIDADE, sendo este uma representação de

entidades que possuem as mesmas propriedades, portanto similares. As entidades são ocorrências de tipos de entidades.

- b) **RELACIONAMENTO**: associação entre entidades que desempenham determinadas funções. Se um dos componentes da associação for outra associação, esta é denominada associação em nesting ou ninho. Uma coleção de associações que relacionam entidades de mesmos tipos, é chamada TIPO DE ASSOCIAÇÃO, e ocorrências de tipos de associações são as próprias associações.
- c) **PROPRIEDADE**: característica qualitativa ou quantitativa de uma entidade ou associação. Ao conjunto de propriedades das entidades, ou associações, de um mesmo tipo denomina-se TIPO DE PROPRIEDADE. Neste caso, um determinado valor de propriedade é uma ocorrência de tipo de propriedade.

Para exemplificar estes conceitos, a tabela 1 mostra dois tipos de entidades: coordenador e projeto; e o tipo coordenação destes, como tipo de associação (relacionamento). Neste exemplo, um dos tipos de propriedade da entidade COORDENADOR seria a qualificação deste, como descrito na tabela 2.

Tabela 1: Exemplo de tipo de associação e tipo de entidade

tipo de associação	tipos de entidades	
tipo de COORDENAÇÃO DE PROJETO	COORDENADOR	PROJETO
coordenador financeiro	Mário	Tubarão
coordenador geral	Ivan	Álcool
coordenador geral	Pedro	Itaipu
coordenador técnico	Sérgio	Itaipu

Tabela 2: Exemplo de tipo de entidade e tipo de propriedade

tipo de entidade	tipo de propriedade
COORDENADOR Mário Ivan Pedro Sérgio	QUALIFICAÇÃO TÉCNICA Economista Administrador Engenheiro Civil Engenheiro Eletricista

Fonte: C. S. YONG |15| p.104

Os relacionamentos dependem das correspondências que deverão existir entre as entidades, definidas a partir da realidade a ser modelada pelas estruturas lógicas do sistema. Estas correspondências, ou conectividades, podem ser um-para-um, um-para-vários, chamadas de hierarquias, ou vários-para-vários. Utilizando o exemplo de um sistema para cadastro técnico, se para cada lote existe um proprietário, uma correspondência um-para-um é definida; além disso, se em cada quadra há alguns lotes, ocorrerão correspondências um-para-vários; e como vários lotes podem receber vários tipos de serviços básicos, existirão correspondências vários-para-vários.

Uma vez definidas as entidades, relacionamentos e propriedades a serem representadas, o próximo passo é decidir qual modelo conceitual deverá ser utilizado, que descreverá a realidade através de modelos de informação. A existência de diferentes modelos de informação é consequente das distintas soluções sobre quais estruturas de dados e operadores relacionados os sistemas devem suportar, que segundo C. J. DATE [7], permite classificá-los de acordo com três abordagens: relacional, hierárquica e em rede. Cada uma destas apresenta soluções distintas para as operações básicas (inserção, remoção e atualização) sobre as informações armazenadas, consequentes das distinções na estruturação dos dados.

2.2.1.1 Banco de dados relacional

Na visão relacional, os arquivos são considerados como casos especiais de construções matemáticas conhecidas como relações, sendo estas as unidades básicas deste modelo. Matematicamente, as relações são definidas como: "se D_1, D_2, \dots, D_n são domínios, não

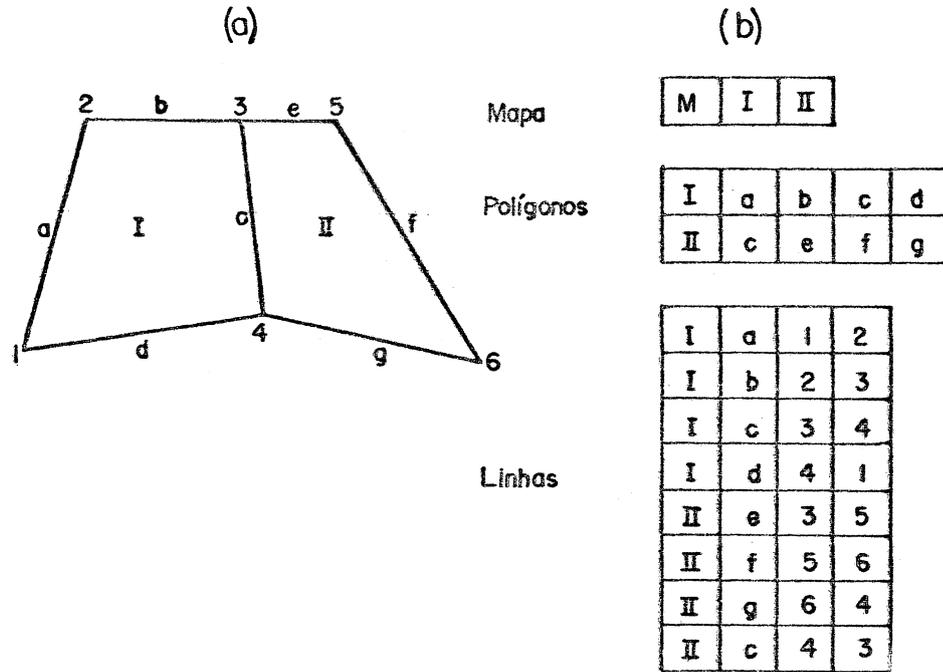
necessariamente distintos, define-se como uma relação R um subconjunto do produto cartesiano dos domínios, ou seja, $R \subseteq D_1 \times D_2 \times \dots \times D_n$." Consequentemente, a teoria dos conjuntos é a base formal para a descrição do modelo, e inclui a definição das operações a serem realizadas sobre os dados. Num banco de dados relacional, as relações são denominadas de tabelas. As linhas de cada tabela são chamadas de tuplas, e as colunas de atributos.

Se relações matemáticas fundamentam esta abordagem, os domínios dos dados devem ser definidos, sendo estes todos os possíveis valores assumidos pelos campos de cada coluna. Porém, estes domínios podem também ser relações, chamados de domínios não-simples. Isto afetaria a eficiência na utilização dos operadores da teoria dos conjuntos, o que pode ser evitado gerando-se domínios simples, ou seja, nos quais os valores são atômicos. Além disso, certas dependências funcionais entre atributos podem prejudicar, principalmente, a manutenção das relações, quando utilizadas as operações de inserção, exclusão e atualização. As operações sobre as relações que transformam domínios não-simples em domínios simples, e eliminam dependências funcionais não desejáveis, são chamadas de normalização, e as relações resultantes de normalizadas.

A necessidade de normalização das relações e manutenção da mesma ordem relativa para todas as tuplas de uma relação, acrescida da limitação do número de atributos, são restrições impostas ao modelo relacional. Por outro lado, a representação das rotas lógicas de acessos pelos valores de atributos e funções da álgebra relacional, permite que comandos simples dos operadores relacionais sejam usados para indicar pesquisas complexas de dados. Neste caso, a melhor rota física de acesso é decidida pelo DBMS, o que caracteriza o modelo relacional como o mais distante dos aspectos físicos de implementação, proporcionando maior independência ao usuário em relação ao nível interno do banco de dados [7], [9], [15].

Um exemplo apresentado por P. A. BURROUGH [4] para ilustrar tanto a abordagem relacional, como as em rede e hierárquica descritas na sequência, utiliza um mapa composto por dois polígonos como sendo a informação que deverá ser armazenada na base de dados. Na figura 9a este mapa é apresentado como visto pelo usuário, e na figura 9b, as relações para mapa, polígono e linhas da abordagem relacional.

Figura 9: Dados na forma relacional



Fonte: P. A. BURROUGH | 4 | pp.17 e 18

2.2.1.2 Banco de dados em rede

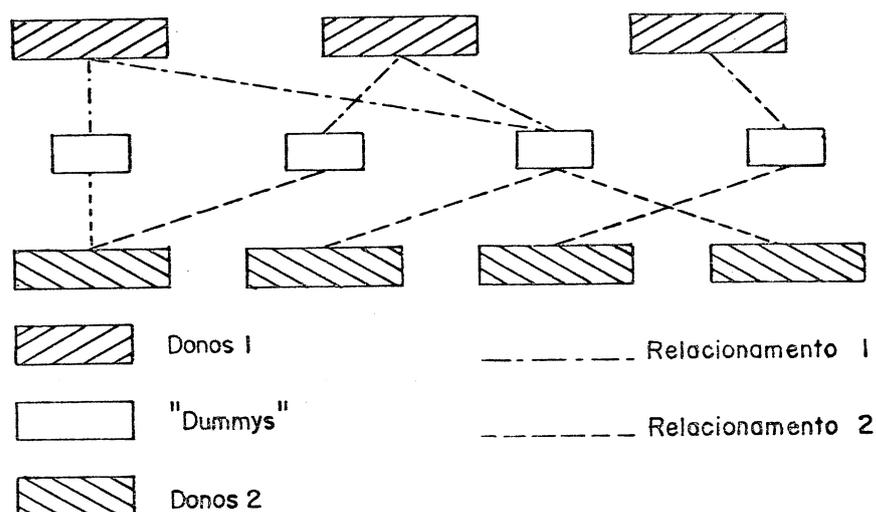
Um banco de dados que possibilita a representação de relacionamentos r elementos, tendo como unidade básica o conceito de associação, define o modelo de informação a partir da abordagem em rede para o banco de dados. Na elaboração do modelo, primeiramente determina-se as entidades básicas e independentes que o comporão, sendo estas ocorrências de registro. Na sequência, define-se os processos de nesting ou ninho, e de associações que deverão existir entre as entidades, observando com isso, que neste modelo as ligações de interesse são pré-definidas e explícitas; implementadas por ponteiros, listas de ponteiros, etc.

Na tentativa de padronizar banco de dados, a Data Base Task Group(DBTG) da Conference on Data Systems and Languages(CODASYL) propôs um modelo de rede, conhecido como modelo CODASYL. A importância deste modelo reside no fato de que, apesar de não ser adotado como padrão único, DBMS's que trabalham com modelo de dados em rede fundamentam-se em várias idéias, dentre elas os conceitos lógicos principais, contidas neste modelo.

A estruturação lógica dos dados do modelo CODASYL baseia-se no conceito de conjuntos, denominados "conjuntos-CODASYL", que representam relacionamentos 1:n entre registros, chamados de donos e membros. Cada dono pode relacionar-se com vários membros, porém cada membro está subordinado a apenas um dono. Deste modo, os relacionamentos n:m são representados indiretamente por um ou mais relacionamentos 1:n.

Apesar disto, este modelo não se limita à representação de hierarquias, uma vez que os registros podem pertencer a diferentes conjuntos, na condição de dono ou membro, cumprindo a exigência de que um registro pode estar associado a dois donos se estes forem de tipos diferentes. Além disso, esta última condição aplica-se a qualquer conjunto, ou seja, os registros dono e membro de um mesmo conjunto devem ser necessariamente de tipos diferentes. Para viabilizar o modelo, cumprindo estas exigências, o conceito de pseudo-registro ou registro dummy faz-se necessário para que os relacionamentos n:m sejam devidamente decompostos em relacionamentos 1:n. Teoricamente, não há necessidade de que estes registros contêm informação, porém na prática, utilizá-los com informações sobre os atributos dos relacionamentos pode ser útil. Estes conceitos, tanto de conjunto como de pseudo-registro, são mostrados na figura 10, na qual os registros dummy desempenham papel de membros, tanto dos donos 1 como 2.

Figura 10: Representação de estrutura em rede do modelo CODASYL

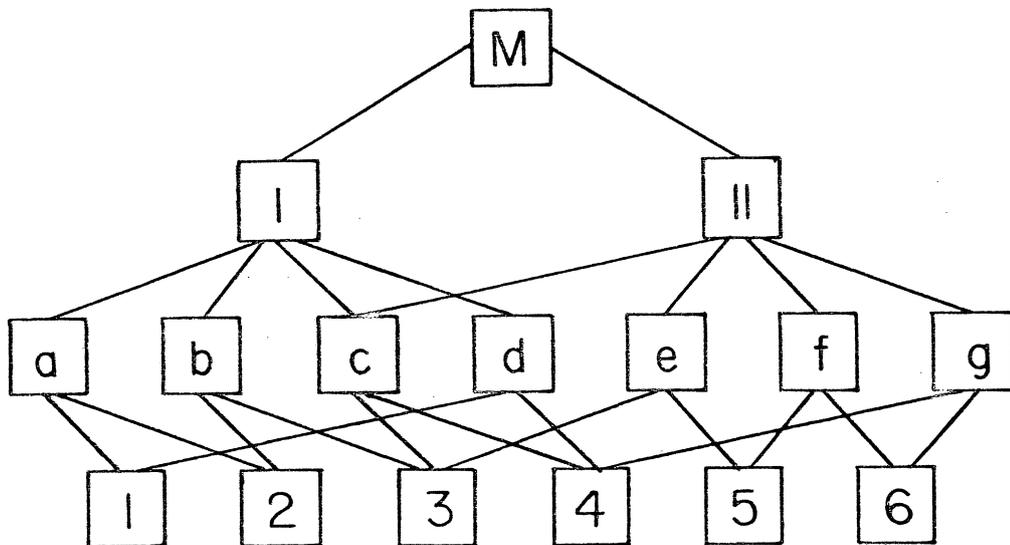


Uma vez que neste modelo a unidade básica é o conceito de associação e, conseqüentemente, todos os tipos de relacionamentos representados por ligações são explicitados, as rotas lógicas de acesso aos dados são definidas acentuadamente. Isto o coloca mais próximo do nível físico do que o modelo relacional.

Porém, a representação de relacionamentos n:m em situações onde muitos tipos de entidades estão envolvidas pode gerar esquemas complexos, sendo esta uma restrição deste modelo [7], [9], [15].

Reportando ao exemplo apresentado por P. A. BURROUGH [4], a estruturação em rede para o mapa da figura 9a é mostrada na figura 11.

Figura 11: Dados estruturados em rede



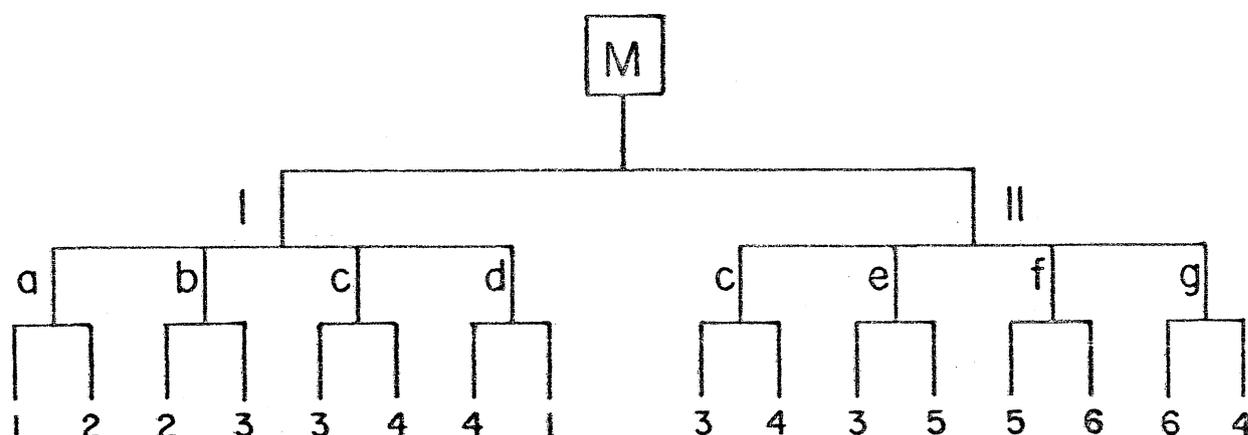
2.2.1.3 Banco de dados hierárquico

Dados estruturados hierarquicamente admitem somente relacionamentos 1:n, sendo os relacionamentos 1:1 situações particulares destes. Este tipo de banco de dados é visto pelo usuário como um conjunto de árvores, nas quais os registros são dispostos em níveis. No primeiro nível, ou o mais alto de uma hierarquia, encontra-se o registro denominado raiz, seguido por registros a ele subordinados situados no segundo nível. Registros subordinados ao segundo nível estarão no terceiro nível, e assim por diante.

Em relacionamentos 1:n, se um determinado registro está ligado a alguns registros do nível imediatamente abaixo, este é denominado ascendente ou pai de registros descendentes ou filhos. Assim, para cada ascendente poderão existir vários descendentes, por outro lado, para cada descendente haverá apenas um ascendente. Como consequência, no modelo hierárquico de dados, a inserção de um registro que não seja raiz implica na definição de seu ascendente; e a deleção de um registro acarreta a deleção de todos os seus descendentes. Com isso, na visão hierárquica, o significado completo de qualquer ocorrência de registro depende do contexto no qual este se encontra. Além disso, haverá apenas uma ligação explícita entre dois registros.

O modelo hierárquico pode ser considerado como um caso particular do modelo em rede, tendo como neste último, o conceito de associação como unidade básica. Também, da mesma forma que no modelo em rede, as rotas lógicas de acesso são definidas pela própria estrutura hierárquica. Porém, como já mencionado, este modelo admite apenas um elo explícito entre dois registros que se encontram em níveis consecutivos. Devido a isto, é adequado à representação de estruturas hierárquicas verdadeiras do mundo real. Por outro lado, para representação de relacionamentos n:m, a incompatibilidade da realidade com o modelo exige definições arbitrárias para as hierarquias. Isto pode gerar soluções complexas e inadequadas para o modelo na representação de realidades não hierárquicas [7], [9], [15]. Para um banco de dados hierárquico, os relacionamentos entre mapa, polígonos e linhas aconteceriam como na figura 12.

Figura 12: Dados estruturados hierarquicamente



Fonte: P. A. BURROUGH |4| p.17

2.3 ESTRUTURAS DE DADOS PARA DADOS GEOGRÁFICOS

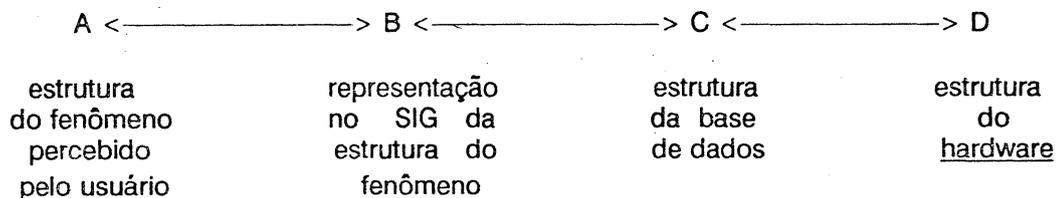
O tratamento de dados geográficos num sistema de informações exige que suas especificidades sejam contempladas em todos os níveis: externo, conceitual e interno. Estas especificidades são conseqüentes das informações geográficas representarem feições da superfície terrestre, e portanto, além dos atributos das feições, suas localizações espaciais devem ser armazenadas na base de dados. Estes dois tipos de dados são definidos como dados gráficos e não-gráficos, onde os primeiros descrevem as características geométricas das feições, e os últimos as características qualitativas e quantitativas, ou seja, temáticas.

Na cartografia convencional, a representação das feições, cujo conjunto formará a imagem cartográfica, utiliza tanto os tipos de símbolos cartográficos: pontos, linhas e áreas, como suas variáveis gráficas: forma, dimensão e cor. Para representar as feições, de modo que a transmissão da informação ao usuário seja adequada, estas devem primeiramente ser classificadas de acordo com as finalidades pretendidas à carta. Quando o usuário obtém informações da carta, as relações entre as feições são por ele percebidas e deduzidas, não havendo necessidade de, na carta, estarem explicitadas, uma vez que a imagem cartográfica criada é conseqüência destas relações. Assim,

através de observações à carta, o usuário sabe sobre adjacências de áreas, pertinência ou conectividades entre feições, além da relação espacial destas com seus atributos.

Em sistemas de informações geográficas, a estrutura do fenômeno, como percebido pelo usuário, será representada por modelos que abrangerão, além dos dados gráficos e não-gráficos, os relacionamentos entre estes e os inter-relacionamentos espaciais das feições, denominados de relações topológicas. Estes modelos, correspondentes ao nível conceitual do sistema, serão armazenados na base de dados através das estruturas de armazenamento relativas ao nível interno. Os diferentes níveis de estruturação de dados de um SIG podem ser visualizados na figura 13. A necessidade de tratamento de dados espaciais num SIG exige que, nestes sistemas, os DBMS's tenham a capacidade adicional de gerenciar este tipo de informação.

Figura 13: Níveis de estruturação de dados de um SIG

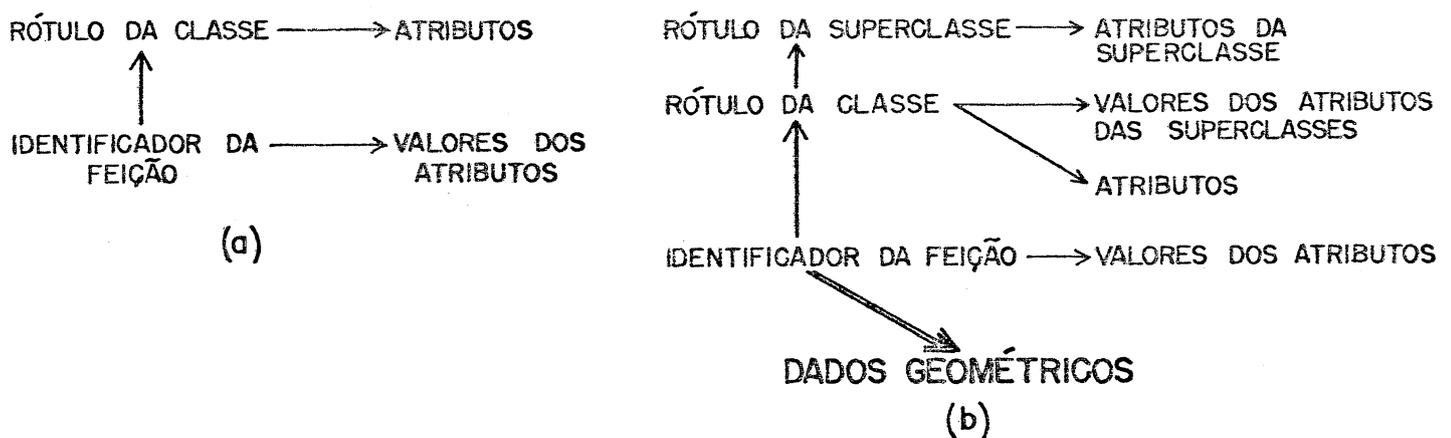


Fonte: P. A. BURROUGH | 4 | p.14

A representação no SIG da estrutura do fenômeno pode adotar estruturas de dados de nível de informação baixo, nas quais os atributos estão diretamente ligados aos dados de posição da feição. Contudo, estruturas de dados com nível de informação mais alto são possíveis se forem adotados identificadores de feições, os quais estarão ligados aos atributos temáticos e aos dados geométricos das feições. Este tipo de estruturação dos dados permite representar computacionalmente a classificação dos atributos temáticos, onde cada classe terá um rótulo e uma lista de atributos que descreverão suas características temáticas (fig. 14a). Se houver subclasses de uma mesma classe, estas serão suportadas pela estrutura acrescentando-se níveis mais altos para representar as classes.

Dessa forma, uma classificação hierárquica é gerada, na qual cada nível comporta o rótulo da subclasse, os valores dos atributos da classe imediatamente superior, e os atributos da subclasse do nível inferior. No nível mais baixo desta estrutura encontra-se a representação dos dados geométricos (fig.14b) |14|. Estes podem ser estruturados, para o nível conceitual, de dois modos: estruturas matriciais e estruturas vetoriais.

Figura 14: Ligação de dados temáticos e dados geométricos



Fonte: M. MOLENAAR |14| pp.89 e 90

Nas estruturas matriciais, o espaço geográfico é assumido como uma superfície cartesiana plana, dividida em células, onde cada célula representa uma porção do terreno. As feições lineares serão identificadas por conjuntos de células enfileiradas com atributos comuns, e as feições de área por conjuntos de células agrupadas, também de mesmos atributos |4|.

A estrutura de armazenamento de dados matriciais pode ser definida utilizando-se o identificador de feição para ligar cada célula, através de suas coordenadas, aos atributos da feição, chamada por M. MOLENAAR |14| de "raster-element-oriented-approach". A estrutura espacial das diferentes feições é obtida pela topologia matricial, construída conectando-se cada elemento a seus vizinhos que possuam o mesmo valor de atributo. Porém, se para a ligação da feição a seus atributos, for utilizado um ponteiro a todos os elementos de uma feição, linear ou de área, a geometria desta

pode ser resgatada diretamente, possibilitando otimização do acesso aos dados, e minimização das exigências de armazenamento e processamento. A este recurso de estruturação M. MOLENAAR | 14 | denomina "terrain-feature-oriented-approach". Quatro métodos de armazenamento de dados matriciais utilizam este recurso: "chain codes", "run-length codes", "block codes" e "quadrees" | 4 |, | 14 |.

Nas estruturas matriciais efetuar análises espaciais pela verificação dos elementos não conectados, definindo-se as relações espaciais existentes entre as feições, é um trabalho incômodo. Por outro lado, a comparação dos overlays de diferentes atributos, se estes basearem-se na mesma grade, é uma tarefa simples, e por isso uma vantagem deste tipo de estrutura de dados espaciais | 14 |.

Contrariamente às estruturas matriciais que discretizam o espaço geográfico, as estruturas vetoriais o concebem como contínuo, utilizando como elementos básicos os tipos de símbolos cartográficos: pontuais, lineares e de áreas. Estes tipos de símbolos correspondem aos conceitos topológicos básicos: nós, arcos e polígonos que compõem as estruturas vetoriais. Desse modo, os nós representarão os símbolos pontuais, os arcos os lineares, e os polígonos os de áreas. Apesar destas estruturas almejam representar o espaço geográfico continuamente, o armazenamento computacional não possibilita tal condição, pois o comprimento da palavra no computador limita a precisão da informação, e os equipamentos de vídeo para visualização de imagens gráficas, as mostram como matrizes bidimensionais, ou seja, um conjunto de células semelhante à estrutura matricial. Contudo, o modelo vetorial permite que estruturas complexas sejam armazenadas num mínimo espaço | 4 |.

Na estruturação dos dados, para o nível conceitual, o modelo vetorial faz uso de apenas dois elementos geométricos básicos: nós e arcos, pois os polígonos são definidos pelos arcos que os limitam. Nesta estrutura, uma feição pontual será representada por um nó, através de suas coordenadas, e ligadas a estas os dados sobre o tipo de ponto, ou seja suas variáveis gráficas, e informações sobre os atributos da feição. Para as feições lineares, as coordenadas dos pontos que a formam, e os nós iniciais e finais do arco definem sua localização. Da mesma forma que para as feições pontuais, informações sobre as variáveis gráficas e atributos da feição estarão ligadas às coordenadas dos pontos. Porém, neste caso como existem relações espaciais entre as feições, e estas

se comportam como redes, ou seja, redes de drenagem, redes de comunicação, etc., a topologia da rede deverá ser representada no modelo. Nesta as conectividades definem os relacionamentos entre as feições. Para tal, estruturas de ponteiros são utilizadas para apontar os arcos que chegam aos nós e registrar os ângulos de junção dos arcos nos nós [4].

As estruturas de dados para as feições de área, deverão descrever, além dos polígonos através de seus arcos limitantes e as conectividades destes, as relações topológicas das áreas, impostas pelos dados geográficos. Estas são tamanhos, significando que cada polígono possui forma, perímetro e área únicos; vizinhanças, ou seja a definição das adjacências entre polígonos; e hierarquias, uma vez que os polígonos não encontram-se todos no mesmo nível, quando alguns polígonos estão totalmente internos a outros polígonos, como as ilhas. Não são todas as construções de estruturas de dados de polígonos que comportam totalmente esta topologia, sendo isto alcançado apenas pelas mais complexas. P. A. BURROUGH [4] descreve três tipos de construções, e as relaciona de acordo com o grau de capacidade, e conseqüentemente complexidade, para suportar relações topológicas, sendo estas:

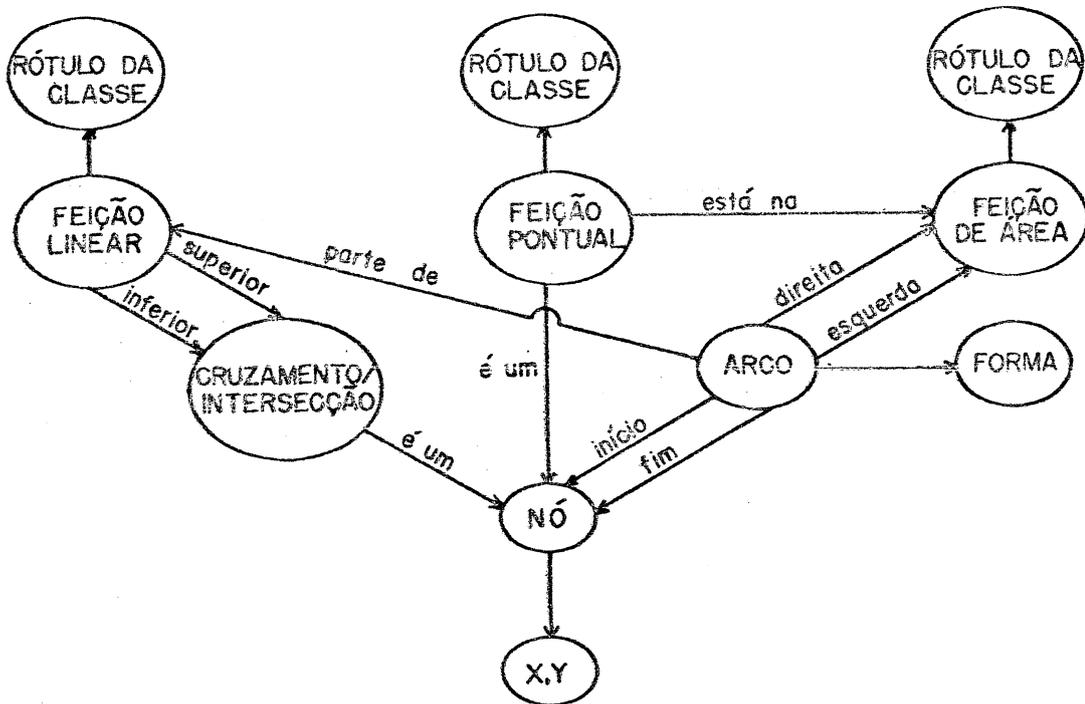
- a) Polígonos simples: cada polígono é representado como um conjunto de coordenadas das linhas limitantes, sendo os nomes ou símbolos gráficos ligados à estas coordenadas. Apesar da simplicidade de concepção, esta estrutura apresenta as seguintes desvantagens:
 - i) Por não conter informações de vizinhança, as linhas comuns a dois polígonos são digitalizadas duas vezes, podendo ocorrer com isto sérios erros de junção entre elas;
 - ii) A estrutura não comporta hierarquias, sendo estas construções puramente gráficas;
 - iii) A verificação de que as vizinhanças estão corretas ou completas não é fácil.
- b) Polígonos com dicionário de pontos: os pontos, armazenados por suas coordenadas e numerados sequencialmente, são associados aos polígonos através de um

dicionário, no qual tais associações estão armazenadas. Nesta estrutura, sendo as adjacências de fronteiras únicas, uma das desvantagens da estrutura anterior é eliminada, porém as demais ainda permanecem (itens b e c). Além disso, dificulta a supressão de fronteiras entre polígonos adjacentes;

- c) Polígonos com estruturas topológicas explícitas: neste caso a topologia é criada, ou durante a entrada de dados, ou por meios de softwares a partir das coordenadas armazenadas dos arcos que formam os polígonos. Para a construção da topologia, primeiramente define-se as conectividades dos arcos a partir dos nós comuns aos mesmos. Em seguida, identificadores de polígonos são estabelecidos, pela definição de um ponto interno a estes, utilizados tanto para ligar os polígonos aos seus atributos e polígonos vizinhos, como também reconhecer e identificar os arcos que os formam. Estes identificadores permitem que a hierarquia entre polígonos seja estabelecida, ou seja, ilhas sejam reconhecidas. Dessa forma, ilhas e vizinhanças podem ser manuseadas através da incorporação das relações topológicas explícitas na estrutura de dados.

Para visualizar a estrutura de dados vetorial, M. MOLENAAR | 14 | propõe o que ele chama de estrutura de dados formal, porque não define a estrutura de armazenamento, servindo como um modelo matemático para o projeto da base de dados, que poderá utilizar a abordagem relacional, de rede ou hierárquica em sua representação. Esta estrutura é apresentada na figura 15, e o diagrama da figura 16 mostra as relações topológicas que podem ser derivadas desta estrutura formal.

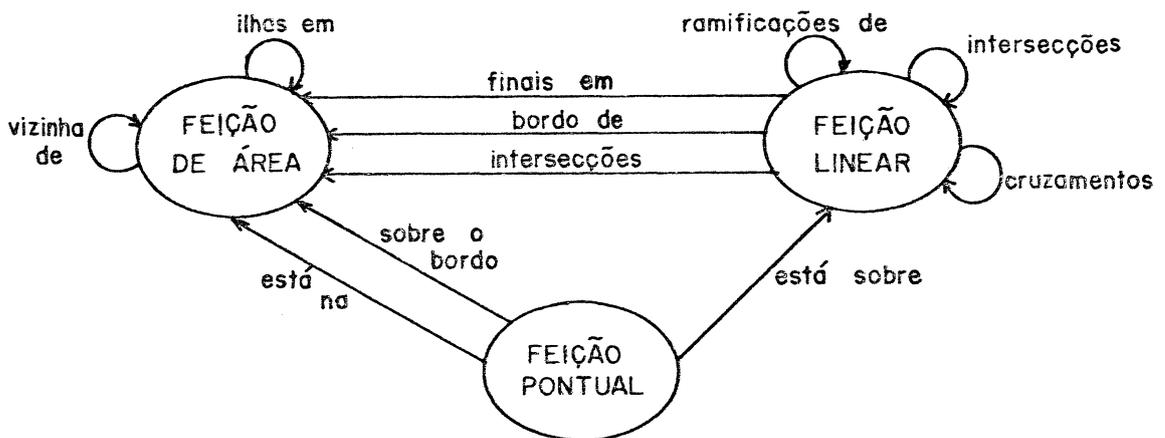
Figura 15: Estrutura de dados formal para mapas vetoriais



Fonte: M. MOLENAAR | 14 | p.94

Considerando que as estruturas vetoriais para dados gráficos facilitam as análises espaciais, porém dificultam as análises de overlays, sendo esta situação oposta para as estruturas matriciais, estas são complementares num SIG | 14 |.

Figura 16: Relações topológicas das feições



Fonte: M. MOLENAAR | 14 | p.94

2.4 ENTRADA DE DADOS FOTOGRAMÉTRICOS EM SIG

Apesar das diferenças entre os objetivos existentes entre SIG's e sistemas de mapeamento digitais, os quais exigem para estes últimos uma base de dados estruturada de forma mais simples, o custo da coleta de dados afeta ambos os sistemas, exigindo por isso especial atenção. Contrariamente ao relativo barateamento dos softwares e hardwares, a captura de dados permanece cara, podendo chegar, como cita P. F. DALE [6] numa estimativa grosseira, até 80% do custo total do estabelecimento de um SIG. W. S. WARNER et alli [17], ao relatar a experiência do Norwegian Institute for Land Inventory - NIJOS- afirma que dos orçamentos projetados para implantação de um SIG, mais de 50% seria para a aquisição de dados.

A capacidade de um SIG de receber dados gráficos de diferentes fontes depende dos dispositivos de hardware disponíveis, e de softwares que possibilitem interfaceamento entre os dados de entrada e a base de dados. A base de dados, por sua vez, deve ser capaz de armazenar dados estruturados tanto na forma matricial como vetorial. Os dados obtidos de produtos cartográficos convencionais são introduzidos num SIG, ou manualmente por meio de mesas digitalizadoras, ou automaticamente por digitalizadores de varredura automática. Devido às agências de mapeamento, ou empresas produtoras ou usuárias de produtos cartográficos, possuírem um acervo de produtos convencionais, a vasta utilização destes equipamentos se faz necessária nesta fase de transição entre a Cartografia convencional e a totalmente digital. Em situações onde os produtos cartográficos são resultantes de dados provenientes de levantamentos fotogramétricos ou sensoriamento remoto, restituidores analíticos, digitalizadores de vídeo ou pacotes de processamento de imagens fornecem tais produtos diretamente na forma digital.

Digitalizar mapa existentes em mesas digitalizadoras, é uma tarefa mais fácil do que digitalizar modelos fotogramétricos, uma vez que nos primeiros as feições já estão interpretadas, exigindo do operador apenas o cuidado de acompanhá-las com o cursor. Contudo, as feições representadas nos mapas sofrem processos de generalização, os quais incluem simplificações, seleções, exageros e deslocamentos, afetando a qualidade dos dados armazenados. Já o segundo caso exige que as feições sejam interpretadas além de digitalizadas. Porém, entre os principais

objetivos da digitalização por instrumentos fotogramétricos, relacionados por M. M. ALLAM [2], estão: o registro de posições precisas a partir de fotografias aéreas em contraste à digitalização de mapas existentes; a possibilidade de preparar mapas em várias escalas; e a aceleração da atualização cartográfica. Esta qualidade da informação, como comenta P. F. DALE [6], não depende apenas da escala, exatidão e precisão dos dados, mas também da acessibilidade, conveniência em qualquer circunstância, clareza, compatibilidade, consistência e atualidade.

Na evolução dos sistemas de estereorestituição, conseqüente do desenvolvimento da tecnologia computacional, diferentes equipamentos surgiram, os quais foram classificados por PETRIE e ADAM, citados por A. S. WALKER [16], em quatro categorias de acordo com suas capacidades:

- a) Unidades baseadas em hardware: nas quais os componentes eletrônicos são fabricados para contagem de pulsos, mostrador de coordenadas, restabelecimento de coordenadas, e saída gráfica. Estas têm bom desempenho, porém falham em flexibilidade. Os instrumentos incluídos nesta categoria são: Wild EK8 e EK22, e Kern ER2;
- b) Unidades baseadas em firmware: que substituem os componentes eletrônicos por microprocessadores. Estes equipamentos, na prática, não demonstram aumento de flexibilidade e redução de custos, sendo eles: Kern ER34, DC_2B, e Zeiss Coordímetro G;
- c) Unidades baseadas em software: nestas um microcomputador, que pode ser programado com linguagem de alto nível, é interfaceado aos instrumentos para medições no restituidor. Isto fornece flexibilidade suficiente para, além de registrar e mostrar dados coletados, executar tarefas tais como: calibração de instrumentos, orientação interna e externa, compilação de mapas, observação e conexão de modelos independentes. Os instrumentos desta classe são: Kern MAPS200 no DSR11, e Zeiss PLANIMAP nas séries PLANICOMP;
- d) Unidades de multiestação em tempo compartilhado: nas quais um único computador recebe dados de vários restituidores.

Nos restituidores analíticos, categoria (c), instrumentos capacitados à digitalização de modelos fotogramétricos, as tarefas desempenhadas pelos pacotes computacionais são hierarquizadas, permitindo a descentralização das atividades nos seguintes subsistemas: manutenção do modelo estereoscópico pelos servo-motores das placas; superposição das imagens; correlação automática; interfaces existentes com o restituidor | 2 |.

As estruturas de dados mais complexas para SIG's, que comportam relacionamentos e armazenamento de dados não-gráficos capazes de permitir análises e manipulação de dados, exigem mudanças nos procedimentos fotogramétricos. A digitalização dos modelos deve ser acompanhada da estruturação e rotulação dos dados na base de dados, pois estes sistemas possuem gerenciadores da base de dados (DBMS) que operam com modelos relacionais e hierárquicos. Esta nova situação fez com que fabricantes de instrumentos fotogramétricos desenvolvessem produtos para SIG'S como o Kern INFOCAM, Wild System 9, e Zeiss PHOCUS; e software houses aumentassem seu interesse para entrada de dados fotogramétricos. Esta evolução proporcionou, como descreve A. S. WALKER | 16 | citando MCLAREN e BERRIL, uma nova classificação para sistemas de aquisição de dados fotogramétricos (SAF):

- a) SAF's cujos dados adquiridos não são mantidos na forma digital, possibilitando apenas a plotagem destes: Kern AT, Wild Aviotab TA e Zeiss DZ7;
- b) SAF's que armazenam dados não-estruturados digitalmente, e não constroem uma base de dados em tempo real. Ainda o objetivo é a plotagem, encontrando-se neste grupo: Wild EK22 e Zeiss(Jena) Coordímetro G, sistemas de estereorestituição assistidos por computador como Kern MAPS200, Wild RAP2 e Zeiss(Oberkochen) PLANIMAP, a maioria dos restituidores analíticos, e a maioria dos sistemas, tais como os criados pela Hunting Surveys;
- c) SAF's que armazenam dados estruturados numa base de dados em tempo real durante a aquisição destes, sendo subdivididos em:
 - i) estações autônomas sem interconexão de hardware. A comunicação de dados é realizada por armazenamento portátil;

- ii) SAF's apoiadas em estações de trabalho de propósitos gerais conectadas a um microcomputador central. Nestas a estrutura da base de dados pode não ser topologicamente rigorosa. Estas são: Wildmap e Intergraph Interact Stations com IGDS/DMRS;
- iii) SAF's autônomas com rede e segunda geração de gerenciamento da base de dados, sendo estas: estações de trabalho fotogramétricas da Kern INFOCAM, Wild System 9 e Zeiss SIG's PHOCUS; e o Intergraph Intermap Analytic Analytical Stereoplotter.

Como já mencionado, a amplitude de capacidade dos SIG's exigindo armazenamento de dados estruturados traz exigências adicionais ao operador do restituidor. A construção on-line, já possível, da base de dados durante o processo de restituição faz com que o operador fotogrametrista constantemente saia da ocular do instrumento para executar tarefas para as quais não foi treinado. Isto onera o trabalho, e como comenta A. S. WALKER | 16 |, os fins podem não justificar os meios, o que faz com que muitas organizações prefiram utilizar os métodos convencionais de aquisição de dados, e posteriormente transferi-los para a base de dados do SIG, executando o processo em batch. Na experiência do NIJOS, descrita por W. S. WARNER | 17 |, para o estabelecimento de um SIG, dois procedimentos chamados bundle e modular foram estudados. Neste estudo, as conclusões sobre os problemas da solução bundle, baseiam-se na percepção do operador e no custo, pois a diferença de rotinas de trabalho pode tornar o sistema de entrada de dados ineficiente. A solução modular foi decidida como mais apropriada, a qual apresentou as seguintes vantagens:

- a) o sistema de base de dados anteriormente utilizado permaneceria internamente nesta agência, e com outras agências de mapeamento;
- b) o treinamento de pessoal seria minimizado, pois os usuários destes sistemas operariam a entrada fotogramétrica no SIG;
- c) o capital investido amortizaria rapidamente pelo uso de instrumentos fotogramétricos disponíveis; e

- d) a captura de dados permaneceria uma entidade separada, e sua integração interessaria somente ao pessoal alocado no módulo de entrada fotogramétrica.

3 ARQUITETURA DO SOFTWARE E ENTRADA DE DADOS DO SGI-INPE

Concebido para apoiar projetos específicos que necessitam de informações de uma região geográfica limitada, como planejamento de uma rodovia, estudos sobre impactos ambientais, o SGI-INPE baseia-se em tecnologia de custo relativamente baixo. A configuração básica do hardware consiste de:

- a) microcomputador compatível com IBM-PC 80x86;
- b) co-processador 80x87;
- c) memória principal mínima de 640Kbytes;
- d) disco rígido de no mínimo 40Mbytes;
- e) disco flexível;
- f) terminal de vídeo alfanumérico;
- g) unidade visualizadora de imagens, com 1 ou 4 planos de 8 bits por pixel (do inglês picture element) e um monitor colorido de alta resolução;
- h) mesa digitalizadora;
- i) plotter;

Além disso, possui uma interface com o software SITIM-SR, sistema para realce e classificação de imagens, podendo estas serem de dois tipos: imagens de satélite: LANDSAT, SPOT, GOES e NOAA, e imagens temáticas geradas por classificação de imagens de satélites ou pelos softwares de manipulação de dados do SGI.

Nesta configuração de hardware percebe-se que a captura de dados gráficos é possível por mesas digitalizadoras e pelo SITIM-SR, ou seja, dados de cartas já existentes e de imagens de satélites. Além disso, coordenadas de feições lineares gravadas em arquivos, em discos flexíveis, podem ser introduzidas na base de dados, se o formato destas for compatível com o software responsável por este interfaceamento.

Em relação ao ambiente de processamento, o SGI-INPE foi desenvolvido para o sistema operacional DOS ou compatível, na linguagem C. Sua estrutura de software compreende cinco módulos: interação/ativação; aplicativos; biblioteca básica; base de dados; e aplicativos do usuário, esquematizada na figura 17.

3.1 MÓDULO INTERAÇÃO/ATIVAÇÃO

No SGI, o usuário troca informações com o sistema através do módulo interação/ativação. Ativando um aplicativo pelos menus ou códigos de chamada, o usuário interage com o sistema por meio dos diálogos padronizados do aplicativo ativado.

Os menus são definidos hierarquicamente de acordo com seu gerenciador, concebido para o sistema DOS ou MS-DOS e monitores com placa CGA, EGA ou VGA, centralizado num arquivo de texto ASCII e subdividido em três blocos principais: bloco de configuração, bloco de ajuda e bloco dos programas executáveis.

O título do menu principal e como este será apresentado graficamente ao usuário são responsabilidades do bloco de configuração, tendo para isto o seguinte formato no arquivo ASCII:

```
@título (máximo 80 caracteres)
&cor do título
&cor das bordas
&cor da área de auxílio
&cor da função a ser executada
&cor das funções não executadas
```

O caracter @ identifica o início do arquivo de menu, e & o bloco de configuração de cores, definidas obviamente em monitores policromáticos, sendo nos monitores monocromáticos substituídas por tons de cinza.

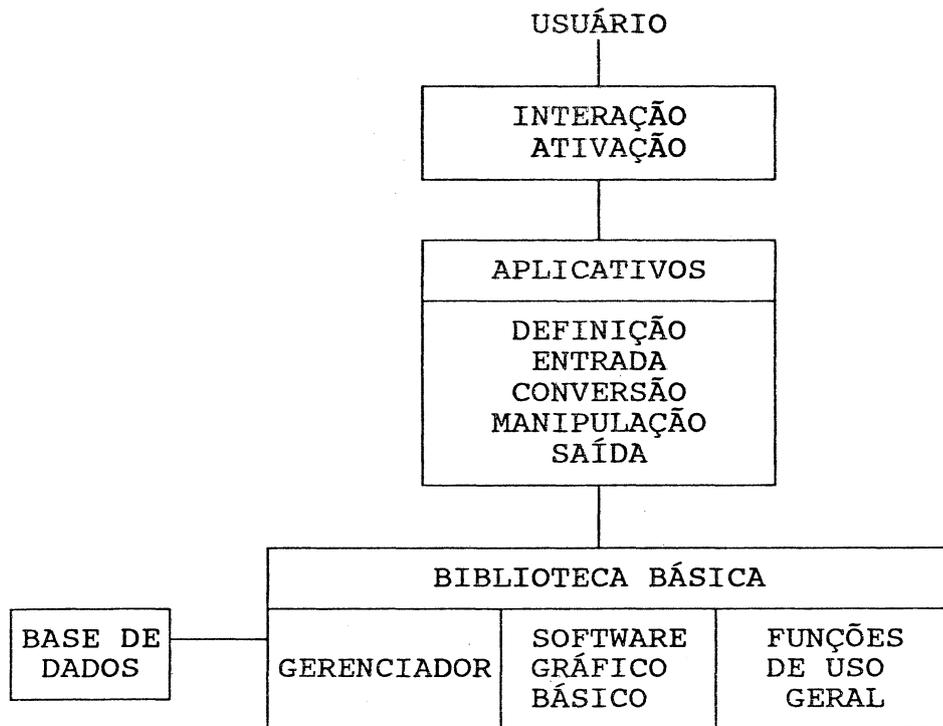
As funções de rolamento de tela, acionadas pelas setas: para esquerda, para direita, para cima e para baixo; e as mais frequentes no sistema, pela teclas F1 a F10, compõem o bloco de ajuda. O formato do arquivo ASCII para este bloco contém duas linhas para cada uma das dez teclas de função, onde a primeira linha define a tecla a ser acionada e o título da ação a ser realizada, e a

segunda linha o caminho e nome da função a ser executada. As linhas para as teclas não utilizadas deverão apresentar-se da seguinte forma:

1ª linha: NADA

2ª linha: ZERO

Figura 17: Esquema dos blocos principais do software SGI-INPE



Fonte: Manual de programação do SGI | 12 |

O bloco de programas executáveis contém na sequência de linhas: o número total de menus do menu principal; coluna, linha de visualização na tela; número de funções do menu e título deste. Após estas informações, para cada menu são listados:

- a) nome da função na tela do menu, onde:

- i) # identifica uma função que gerará outro menu, sendo esta uma função não executável,
- ii) * indica uma função executável, as quais são seguidas por:
 - 1) abreviatura da função;
 - 2) caminho e nome da função executável;
 - 3) nome do arquivo que contém a documentação da função;

Um exemplo mostrando parte de um arquivo ASCII de menus é apresentado na figura 18.

Figura 18: Parte de um arquivo ASCII de menu

```
@ SISTEMA DE INFORMACOES GEOGRAFICAS
& 3
& 14
& 3
& 14
& 3
F1
ZERO
F2 JAN CENTR
ATIVAPL.EXE c
F3 JAN 2PTS
ATIVAPL.EXE s
F4 JAN ESC
ATIVAPL.EXE e
F5 JAN ENQUAD
ATIVAPL.EXE q
F6 JAN ATUAL
ATIVAPL.EXE a
F7 MONITOR
ATIVAPL.EXE v
F8 MESA
ATIVAPL.EXE m
F9 CALCULAR
ATIVAPL.EXE u
F10 DOS
ZERO
```

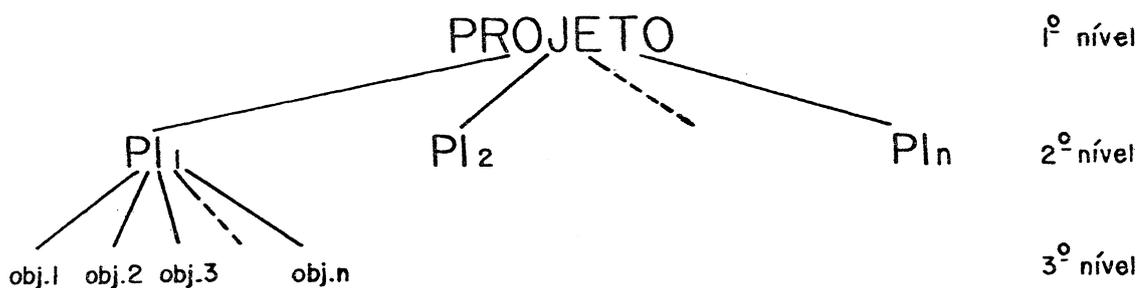
Fonte: Manual de programação do SGI | 12 |

3.2 ESTRUTURAS DE DADOS

3.2.1 Estrutura de dados para os níveis externo e conceitual

Para o usuário, os dados no SGI são estruturados hierarquicamente, encontrando-se no primeiro nível desta estrutura o projeto, definido pela região de trabalho limitada geograficamente. A cada projeto estão subordinados alguns planos de informação (PI's), estando estes no segundo nível da hierarquia. Em cada PI algumas classes de feições são representadas, consistindo estas de um conjunto de objetos geográficos, os quais situam-se no terceiro nível da estrutura (fig. 19). Os dados gráficos dos objetos podem representar feições do terreno por estruturas vetoriais ou matriciais.

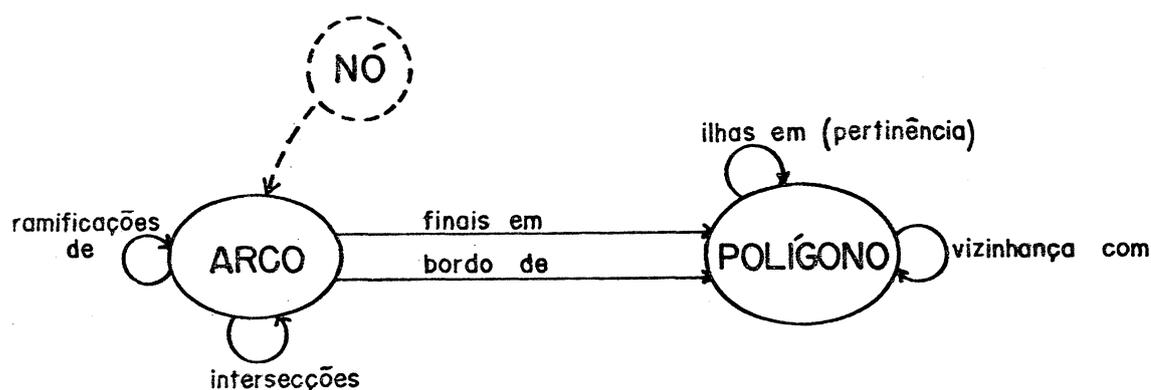
Figura 19: Estrutura de dados para os níveis externo e conceitual do SGI-INPE



Para os dados vetoriais, a topologia é construída por polígonos com estruturas topológicas explícitas. Esta é criada a partir das coordenadas das feições armazenadas, por aplicativos que desempenham cada passo desta construção. Para tal, exigem que os nós sejam definidos pelo usuário durante a digitalização, ou pelos recursos de edição. Além disso, polígonos abertos e arcos, devem ser identificados, ou após o armazenamento das coordenadas das feições ou digitalizando-os separadamente. Tanto os polígonos fechados como as ilhas (polígonos com apenas um nó), além de digitalizados separadamente, são identificados pela inserção de pontos em seu interior chamados de centróides, que os conectam à classe a que pertencem, sendo esta a ligação possível no SGI entre os dados gráficos e seus atributos (fig. 21b).

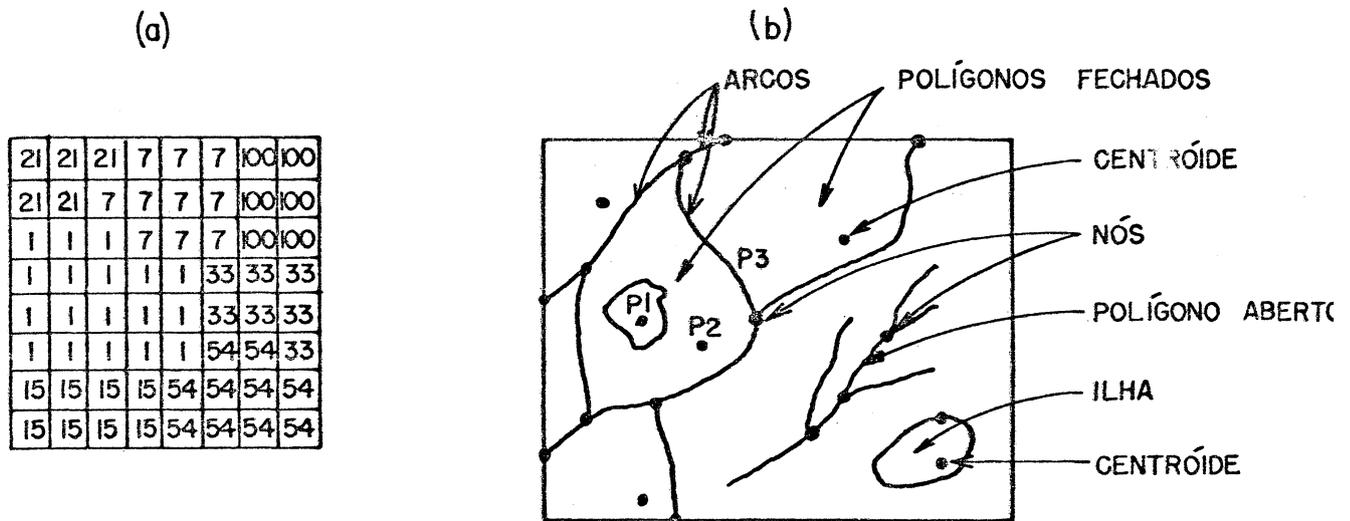
Na construção da topologia das estruturas vetoriais no SGI, os nós existem apenas temporariamente para realizar o ajustamento das terminações das linhas e a definição dos polígonos a partir destas. Ajustar as terminações das linhas significa definir um único ponto, no qual as linhas que se interceptam iniciam ou terminam. Os arcos compõem as redes, chamadas de polígonos abertos, ou contornos de polígonos fechados e ilhas. Além da topologia de rede, as relações topológicas previstas para polígonos são pertinência e vizinhança mostradas na figura 21b. A figura 20 apresenta um diagrama, baseado no proposto por M. MOLENAAR |14|, para as relações topológicas definidas no SGI.

Figura 20: Relações topológicas previstas no SGI



Nas estruturas matriciais, uma construção semelhante à citada por M. MOLENAAR |14| como "raster-element-orient-approach" é adotada, neste caso tendo para cada pixel apenas um atributo definido. Nesta, para o SGI, cada célula é localizada espacialmente pela linha e coluna da matriz que representa a região do terreno. Em cada elemento da matriz há um número, no intervalo de 0 a 255, que define o atributo da célula. Este é o formato de dados no qual as imagens de satélite são transferidas ao SGI (fig. 21a). Já, a digitalização de cartas exige estrutura vetorial (fig. 21b).

Figura 21: Estruturas para dados gráficos do SGI-INPE: (a) dados matriciais; (b) dados vetoriais.



Além destas estruturas de dados matriciais e vetoriais, no SGI é previsto representar os dados gráficos por coordenadas de pontos isolados (x,y,z) nos quais a coordenada z pode representar tanto altitude como também qualquer grandeza relacionada à fenômenos geográficos. Para tanto, dois formatos são previstos: para dados vetoriais, amostras 3D nas quais cada ponto da superfície terrestre é representado por suas três coordenadas; e para dados matriciais, grade isolada, cujas coordenadas planas são regularmente espaçadas, e a elas está associada a terceira coordenada.

Além das estruturas de representação das características gráficas das feições geográficas, o SGI utiliza o conceito de categoria de dados que define o tipo de entidade a ser manipulada pelo sistema. Assim, para manipulação de áreas, ou seja, mapeamentos como uso do solo e culturas, os dados pertencerão a categoria chamada de dados poligonais. Na manipulação das entidades chamadas de elemento do terreno, como altimetria, geofísica, geoquímica, os dados serão enquadrados na categoria de modelo numérico do terreno. Na terceira categoria de dados prevista no SGI, chamada imagem multiespectral, a entidade manipulável é a banda.

A diferença entre categoria de dados e estrutura de representação destes, reside no fato de que, numa mesma categoria de dados pode haver mais de um tipo de estrutura de representação. Os relacionamentos entre categoria de dados, entidade manipulável e formato de representação são mostrados na figura 22.

Figura 22: Categoria de dados, entidade manipulável e formato de representação dos dados no SGI

CATEGORIA	ENTIDADE MANIPULÁVEL	FORMATO DE REPRESENTAÇÃO
dados poligonais	área	vetor varredura
modelo numérico do terreno	elemento do terreno	amostras grade varredura vetor
imagem multiespectral	banda	varredura

Fonte: Manual do usuário do SGI | 11 |

3.2.2 Estrutura da base de dados

Estruturada de acordo com o modelo hierárquico, a base de dados do SGI é dividida em projetos, compostos por planos de informação, e estes pelos objetos que o formam. A organização da base de dados do SGI baseia-se em arquivos, gerados e manipulados durante o processamento dos aplicativos, e armazenados na memória de massa ou memória principal. O nome do projeto de trabalho do último processamento é armazenado no arquivo contexto, e para cada projeto os seguintes arquivos são utilizados para defini-los:

- a) diretório: com os atributos do projeto e dos PI's,
- b) amarração-mesa: contendo as coordenadas, de mesa e de projeção, dos pontos utilizados para a determinação dos parâmetros de transformação;
- c) cartas: no qual estão armazenados os dados necessários para definir a plotagem destas;

e pelos planos de informação do projeto.

No nível dos planos de informação, os objetos são definidos de acordo com a estrutura de dados a qual pertencem, e pelos arquivos de textos, com todos os textos e atributos de um PI; e de símbolos com os quais as feições de cada PI serão representadas na plotagem.

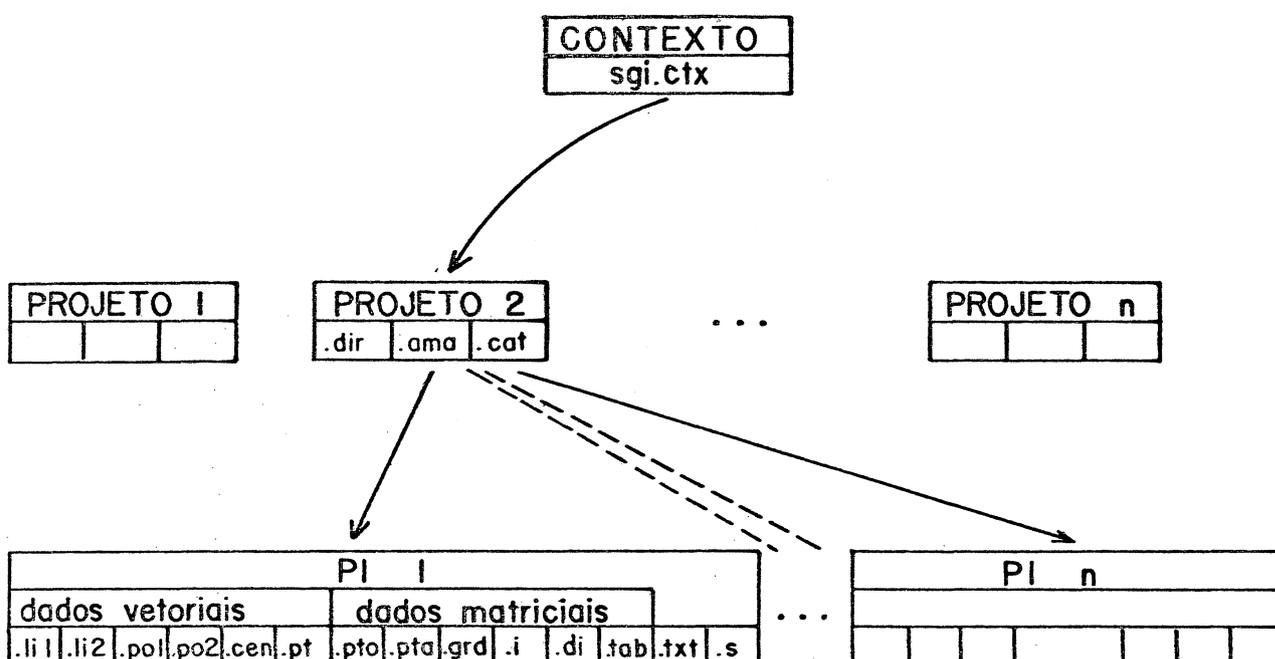
O armazenamento de estruturas de dados matriciais na base de dados compreende os arquivos definidores de imagem e de grade (fig. 23):

- a) arquivo imagem: o qual armazena informações sobre os valores dos pixels, que para as imagens de satélite são valores de reflectância (0 a 255) detectados pelos sensores;
- b) arquivo descritor: contendo informações gerais sobre o arquivo imagem, como nome do projeto, nome do PI, resoluções horizontal e vertical, coordenadas matriciais e vetoriais da janela em uso, etc;
- c) arquivo tabela: mantém informações sobre os atributos da imagem, como o número de classes do PI, e sobre cada classe, nome da classe, cor, tipo de preenchimento gráfico, área da classe e texto.
- d) árvore(2D) e pontos organizados: gerados no aplicativo ORGANIZAR PONTOS, para armazenar os nós de uma estrutura de árvore, e os pontos de cada partição desta, respectivamente. Ambos os arquivos são criados a partir de arquivos de pontos 3D, ou de arquivos de linhas ou polígonos. Estes últimos são estruturados vetorialmente, porém pertencem à categoria de dados MNT.
- e) grade regular: criado no aplicativo GERAR GRADE REGULAR, onde para ponto desta grade regularmente espaçada, os valores da terceira coordenada (z) são interpolados dos vizinhos mais próximos. A geração desta grade é realizada a partir dos arquivos de árvore(2D) e pontos organizados.

Para objetos estruturados vetorialmente, além de arquivos para armazenamento das linhas, poderá existir arquivos para armazenamento de polígonos e de centróides, compostos durante o processo de construção da topologia destes dados. Como mostra a figura 23, esta estruturação de dados ocupa cinco arquivos na base de dados:

- a) arquivo de cabeçalho das linhas: dividido em duas partes, armazena na primeira informações gerais sobre as linhas de um PI, entre estas o número de linhas, índice de compactação, coordenadas mínimas e máximas da área de trabalho. Na segunda parte encontram-se as informações sobre cada linha, como número de pontos da linha, coordenada z se houver, indicação sobre a realização ou não do ajuste de pontas, coordenadas mínimas e máximas de cada linha, quais linhas estão ligadas aos primeiro e último pontos desta, etc.;

Figura 23: Organização da base de dados do SGI



Fonte: Manual de programação do SGI | 12 |

- b) arquivo de dados das linhas: contendo as coordenadas planas dos pontos de cada linha;
- c) arquivo de cabeçalho dos polígonos: semelhante ao de linhas, armazena na primeira parte as informações que correspondem a todos os polígonos do PI, tais como número de polígonos; e na segunda parte sobre cada polígono, sendo estas índice

do polígono, coordenada z, área, rótulo do polígono, número de linhas, retângulo envolvente, tipo de polígono (indefinido, aberto ou fechado) e listas das linhas que os formam;

- d) arquivo de centróides: que contém o número de centróides, e para cada centróide, índice deste e suas coordenadas planas na projeção utilizada na representação do PI.

Como já mencionado, ainda em relação às estruturas vetoriais há no SGI a possibilidade de armazenar pontos isolados da superfície terrestre representados por suas coordenadas, em arquivos próprios para tal, sendo estes denominados de arquivos de pontos(3D).

3.3 MÓDULO DE APLICATIVOS

Os aplicativos desenvolvidos para possibilitar aos usuários desempenhar as tarefas necessárias em cada projeto específico, encontram-se neste módulo. Sua organização o divide em cinco submódulos, de acordo com o tipo de tarefa que realizam: definição, entrada, conversão, manipulação e saída. A concepção do SGI exige que as operações sejam executadas sobre um mesmo conjunto de dados. Assim, sobre o PI ativo serão realizadas as operações de entrada de dados, e para conversão, manipulação e saída é solicitado que se informe os PI's envolvidos. Tanto para entrada de dados, como as demais operações citadas, é necessário que se defina em qual PROJETO estas serão efetuadas, o que é feito ativando-se este.

Portanto, a primeira operação no SGI, após o sistema estar ativado, é definir qual será o PROJETO ATIVO, e em seguida pode-se criar um PI, ou ativar um já existente. Estas tarefas são possíveis no submódulo DEFINIÇÃO. Na definição do PROJETO ATIVO, além do nome, devem ser informados os seguintes atributos deste: projeção cartográfica, escala, unidade métrica das coordenadas, datum horizontal e coordenadas envolventes da região de trabalho. Para a criação de um PI ativo, os seguintes parâmetros são requisitados: categoria dos dados, escala, número de classes e nome de cada classe.

No submódulo ENTRADA estão as funções de entrada de dados para as diferentes categorias destes, alocadas em alguns submódulos, sendo estes:

- a) ENTRADA DE ÁREAS: através do qual pode-se realizar a entrada de dados vetoriais tanto por mesa digitalizadora como por leitura de arquivos ASCII de linhas gravados em meio magnético. O formato de armazenamento das linhas no arquivo a ser lido é mostrado na figura 24. Acompanhando as funções de digitalização, encontram-se neste submódulo as funções de edição, que permitem ajustar manualmente os nós e, suprimir, separar e concatenar linhas. Além das funções para digitalização, contém funções para identificação dos diferentes elementos topológicos, que no caso de áreas é feita pela inserção de centróides, podendo estes serem editados pelas funções de supressão de centróide ou modificação da classe atribuída. Juntamente com as funções de identificação de áreas e edição, as funções para ajustar linhas de, e poligonalizar, um PI, criam a topologia da estrutura vetorial do PI.

Figura 24: Formato de dados de um arquivo ASCII de linhas para ser lido pelo SGI

```
# pontos
x y
x y
.
.
# pontos
x y
x y
.
.
```

Fonte: Manual do usuário do SGI | 11 |

- b) ENTRADA DE MNT: permite a entrada de dados, via mesa digitalizadora ou arquivos gravados em meio magnético, de pontos isolados ou curvas de isovalores. Da mesma forma que para entrada de áreas, estas são acompanhadas de funções para edição tanto de linhas, como de pontos. Para as linhas, as funções para edição existentes permitem alterar a cota de, ou suprimir, uma isolinha. No caso de pontos, possibilitam

listá-los, suprimí-los ou alterar suas cotas. O formato dos arquivos de pontos e isolinhas é mostrado na figura 25.

- c) **ENTRADA DE IMAGENS:** realiza a transferência de imagens do SITIM-SR para o SGI, criando automaticamente um PI com mesmo nome da banda da imagem de entrada.

Figura 25: Formato de dados de arquivos ASCII a serem lidos pelo SGI para a categoria de dados MNT: (a)isolinhas; (b) pontos isolados.

# pontos	cota z	x y z
	x y	x y z
	x y	x y z
	.	x y z
	.	x y z
# pontos	cota z	x y z
	x y	.
	x y	.
	.	.
	.	.
(a)		(b)

Fonte: Manual do usuário do SGI | 11 |

As transformações entre estruturas de dados, ou seja, vetorial para varredura, varredura para vetorial e grade regular para vetorial gerando curvas de isovalores, são efetuadas por funções pertencentes ao submódulo CONVERSÃO. Além disso, pode-se converter resoluções e projeções de PI's ativos. Neste submódulo os PI's são preparados para as operações de manipulação, uma vez que, exceto para reclassificação de polígonos e geração de perfis, todas as demais operações sobre PI's são realizadas em dados estruturados matricialmente.

Através do submódulo MANIPULAÇÃO novas informações são geradas através do inter-relacionamento entre os dados de um ou mais PI's. Para tal, contém funções que executam união de classes de um PI; operações lógicas entre classes de dois PI's; operações lógicas com regras definidas entre classes de até dez PI's; cruzamento de até dez PI's através da média ponderada das

classes de cada PI; operações aritméticas entre duas grades; cálculo de área para todas as classes de um PI, e de corte e aterro referente a uma cota base; geração de cartas de declividade e de isolinhas em dados da categoria MNT; e geração de cartas de distâncias mínimas a uma classe ou a um conjunto de classes.

Para visualizar ou listar as informações da base de dados, as funções do submódulo SAÍDA devem ser acionadas. As listagens são relatórios sobre o projeto ativo e seus PI's. A visualização dos resultados pelo monitor não se restringe, como aos plotters, a mapas de linhas, mas inclui vistas perspectivas de MNT com relevo sombreado, ou combinação de imagens de satélites com MNT.

3.4 MÓDULO DA BIBLIOTECA BÁSICA

As funções comuns a todos os aplicativos pertencem ao módulo da biblioteca básica, dividido em três submódulos de acordo com as diferentes finalidades que lhes cabem:

- a) Gerenciador da base de dados;
- b) Pacote gráfico básico - GKS;
- c) Funções de uso geral.

Todas as funções que permitem o intercâmbio de informações do sistema com o usuário dentro dos aplicativos, encontram-se no submódulo das funções de uso geral. Devido a isso, são denominadas de funções de diálogo, pois algumas destas emitem mensagens aos usuários, e outras solicitam aos usuários informações necessárias à execução do aplicativo ativado. Como exemplo do primeiro caso, a função DIA_ERRO(), cuja entrada é um string de caracteres, o escreve como uma mensagem na cor vermelha e piscante, e aguarda até o usuário pressionar a tecla ENTER. Para exemplificar o segundo caso, a função DIA_PEGA_INT() permite adquirir um valor inteiro digitado pelo usuário.

4 SISTEMA DE DIGITALIZAÇÃO DO PLANICOMP C120

4.1 CARACTERÍSTICAS GERAIS DO PLANICOMP C120

No sistema de estereorestituição digital PLANICOMP C120, um estereorestituidor analítico, construído com encoders responsáveis pela medição ou contagem dos movimentos das manivelas (X,Y) e do pedal (Z), é suportado por um computador HP 1000 com unidade de disco e terminal CRT, podendo ter periféricos opcionais como impressoras e outros terminais. Uma unidade eletrônica de aquisição e transferência de coordenadas DIREC 2 permite que estas, através de uma interface HP IB 12009 A, sejam armazenadas em arquivos digitais, ou diretamente plotadas pela mesa traçadora digital DZ-7. Esta configuração de hardware é representada na figura 26 | 19 |.

Sendo um sistema analítico, possui características que lhe proporcionam algumas flexibilidades, entre elas:

- a) obtenção de dados de fotografias de até 240 x 240mm com qualquer distância focal;
- b) oculares do sistema de visualização com lentes-padrão que ampliam 8 vezes a imagem, com "zoom" variando de 7,5 a 30 vezes. Estas podem ser substituídas por binoculares que ampliam 16 vezes, com "zoom" de 15 a 60 vezes;
- c) manivelas e pedal com velocidade de 1000 incrementos por revolução, podendo os movimentos destas serem selecionados para:
 - i) foto esquerda, foto direita e foto esquerda e direita para medidas monoscópicas, com incremento de $SCH \times SCHK = 0,001\text{mm} \times 1$ para X e Y e $SCF = 0,001\text{mm}$ para Z;
 - ii) modelo normal para medidas estereoscópicas, e modelo terrestre para mapeamento on-line, com incremento $SCH = 0,001\text{mm}$;

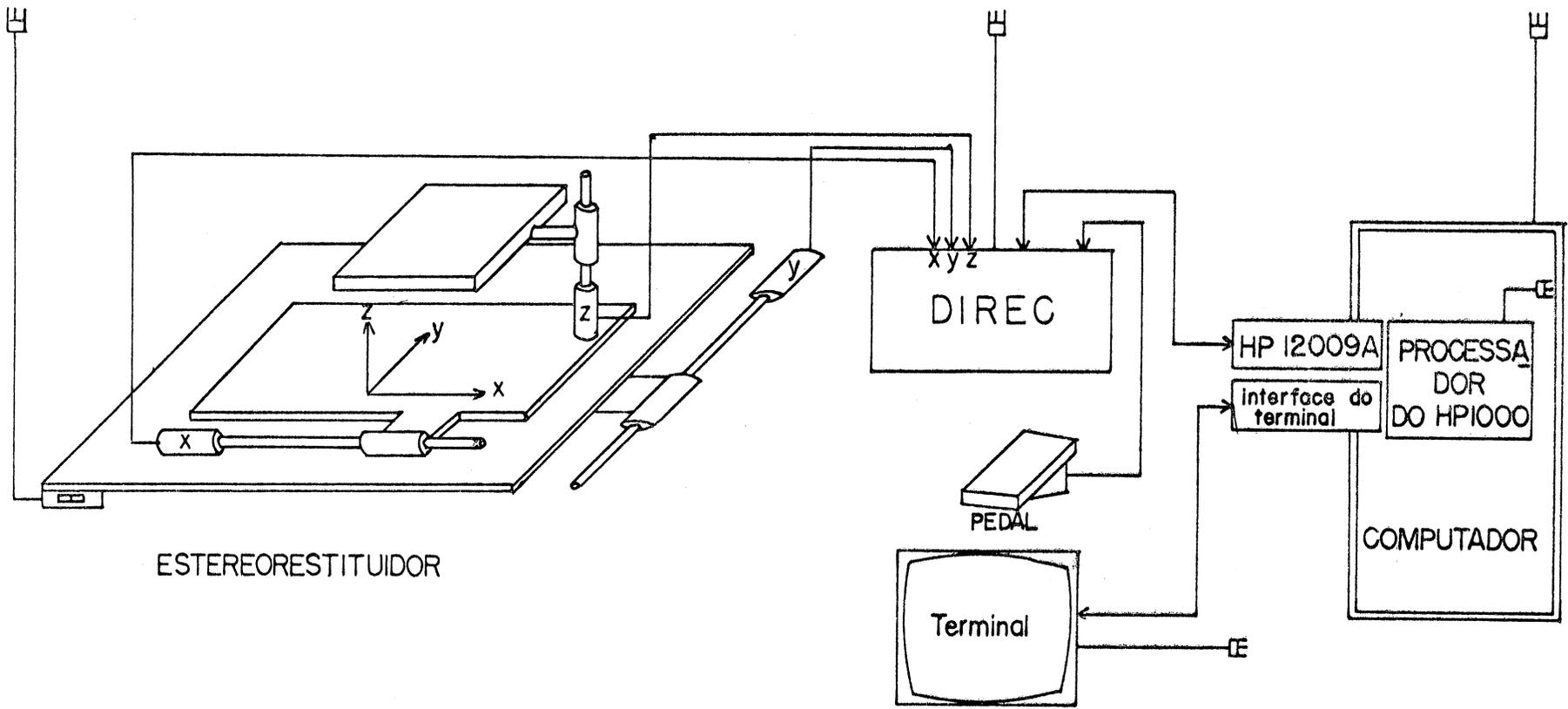


Figura 26: Configuração de hardware para medidas interativas com o programa DZ7-AS

- iii) perfilar modelo para obtenção de perfis do terreno utilizados na produção de ortofotos, modelos numéricos do terreno, registros de seções do terreno para trabalhos de locação de rotas, e registro de perfis gráficos, com velocidade de movimentos iguais aos dos modos normal e terrestre.

Os fatores de escala SCH e SCF significam milímetro por pulso de manivela ou pedal, respectivamente; e o fator SCHK usado no modo foto é multiplicativo. Todos estes podem ser alterados através do programa que define os parâmetros utilizados na restituição.

Os dados de coordenadas de máquina, coordenadas de terreno e orientação coletados, são armazenados em arquivos pertencentes a uma área de memória comum, sendo estes:

- a) GROUND FILE para coordenadas de pontos do terreno;
- b) ORIENTATION FILE para dados de orientação;
- c) GENERAL FILE para dados gerais medidos ou calculados.

Esta área comum é utilizada por programas que transferem dados entre memória principal, arquivos em disco e equipamentos de entrada/saída. Neste sistema, entre as tarefas fotogramétricas suportadas por software encontram-se: medida digital de pontos, registro de modelo digital do terreno, registro de perfis e mapeamento on-line. Com exceção do programa denominado RECORD e acionado pelo código A10, o qual registra as coordenadas de pontos individuais pela posição da marca flutuante num periférico de saída, a estes programas é permitido utilizar os chamados arquivos gerais, criados pelos programas A56 ou GEFIL, se para saída é definido o endereço destes arquivos |19|.

4.2 PROGRAMA GEFIL

Inserido no pacote GEFIO, o programa GEFIL permite criar e editar o conteúdo de um arquivo geral. Além do programa GEFIL, o pacote GEFIO inclui dois outros programas, GREAD e GWRIT, para entrada de dados de fita magnética e saída de dados para a fita magnética ou arquivo ASCII, respectivamente. Duas versões do GEFIO são disponíveis no sistema PLANICOMP C120, ou seja, tanto para o sistema operacional RTE IV A como para o RTE A.

Na criação de um novo arquivo geral, além da denominação deste, a qual deve iniciar com as letras FIL seguidas de três números que podem variar entre 200 a 999, o formato dos dados deve ser definido, baseado na linguagem de programação FORTRAN, através do código RFORM. A estrutura de dados destes arquivos é função da definição do código RFORM, que pode combinar tipos de dado e quantidade destes, de acordo com as possibilidades mostradas na tabela 3. Qual código definir dependerá da tarefa específica a ser realizada, cujos resultados serão armazenados no arquivo indicado.

Tabela 3. Tipo e formato dos dados para arquivos gerais GEFIL

Tipo	Formato	Precisão	Comprimento da palavra
1	I4	inteiro	1
2	I6	inteiro	1
3	F8.3	real	2
4	F10.3	real	2
5	F12.3	dupla precisão	3
6	I13	dupla precisão	3

Assim, se o código RFORM para um determinado arquivo é estabelecido como 162514, significa que cada registro consiste de um tipo de valor 6, 2 tipos de valor 5 e um tipo de valor 4. Como os tamanhos destes arquivos são determinados quando criados, deve-se ter claro que eles podem conter até 32767 registros.

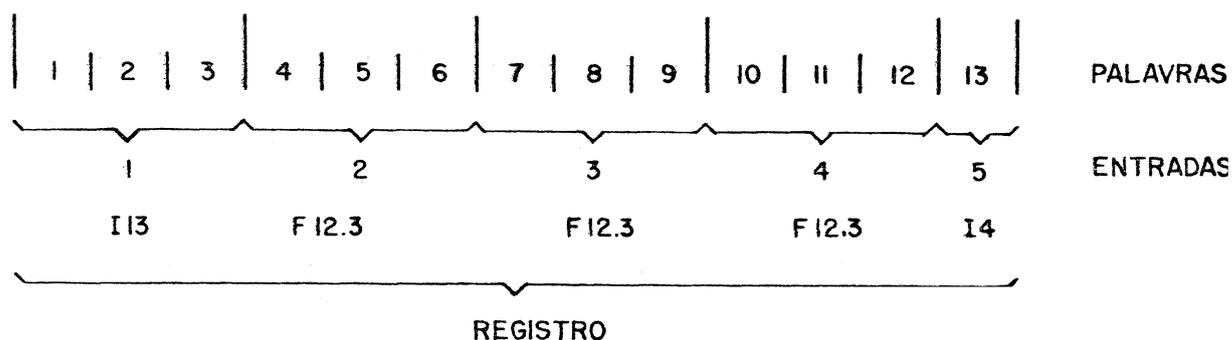
Para editar um arquivo geral, uma série de comandos disponíveis no programa GEFIL, que atuam sobre os registros criados permitem: pulá-los; mostrá-los; suprimí-los; inserí-los; gravá-los em qualquer periférico ou fita magnética; introduzí-los no arquivo; ou encontrá-los. Além disso, janelas

podem ser criadas, estabelecidas ou mostradas; e registros de uma determinada janela apresentados. Após a edição de um arquivo, se o comando de supressão foi usado, o comando de empacotamento do arquivo deve ser acionado. O trabalho de edição deve apoiar-se em terminais CRT ou na mesa traçadora, uma vez que este é realizado sobre registros definidos em cada comando [22].

4.3 PROGRAMA B83 - RECORD PLOT DATA

Para medir e codificar elementos gráficos, no sistema PLANICOMP C120, aciona-se o programa B83, elaborado para gerar arquivos estruturados de tal forma que programas do pacote DZ7-AS possam executar as operações necessárias à plotagem das feições restituídas. No programa B83, em cada registro composto de 13 palavras, do arquivo resultante da digitalização das feições, há 5 entradas de dados, cujos tipos são definidos pelo parâmetro RFORM. Para este programa, tal parâmetro deve ser 163511, ou seja, para cada registro há um tipo de dado de valor 6, três tipos de valor 5 e um tipo de valor um, mostrado na figura 27.

Figura 27: Um registro de um arquivo geral do PLANICOMP C120



Após a chamada do programa, os parâmetros necessários à sua execução, entre eles o dispositivo lógico de saída, são definidos. Neste caso, são previstas duas situações: mapeamento on-line, no qual as feições medidas são diretamente plotadas na DZ-7; ou o armazenamento dos registros da restituição num arquivo geral, para plotagem posterior. Antes de iniciar a restituição de qualquer feição, o operador estabelece o modo de operação, para os quais as seguintes possibilidades são

oferecidas: modo ponto, modo linha aberta, modo linha fechada, modo declividade. Importante ressaltar que, antes de executar qualquer digitalização em arquivos ainda vazios, os primeiros dados destes são os pontos medidos para orientação do plotter. Para tal, este programa aceita até 24 pontos.

Sendo o PLANICOMP C120 um sistema concebido para mapeamento digital, os arquivos resultantes da digitalização armazenam, além das entidades gráficas que formarão as feições, os códigos que definem a aparência gráfica, segundo a qual estas feições serão representadas. Assim, para cada entidade gráfica, e simbologia utilizada em sua representação, há um código específico, relacionado ao modo de digitalização. Esta concepção, incluindo as potencialidades deste sistema para restituição numérica, pode ser entendida pela descrição dos comandos definidores de cada ação. Na sequência de comandos, primeiramente seleciona-se o modo de operação; em seguida, os parâmetros que definem a aparência gráfica da feição; e por último o modo de digitalização relacionado ao tipo e forma da feição | 19 |.

4.3.1 Modo de operação PONTO

Selecionado para digitalização de feições pontuais, é acompanhado de códigos para tipo, tamanho e ângulo de rotação do símbolo para a plotagem; e tamanho e rotação do texto que acompanhará o símbolo, podendo este ser altitude ou número do ponto.

As possibilidades de digitalização compreendem:

- a) plotagem de símbolos de pontos medidos, não acompanhados de texto;
- b) plotagem de símbolos com texto em posição pré-estabelecida;
- c) plotagem de símbolos com texto posicionado em localização a ser medida.

Cada modo de digitalização gera alguns códigos armazenados nos registros do arquivo geral, cuja finalidade é orientar o programa de plotagem na execução da ação pretendida. A estrutura de dados resultante de cada modo de digitalização é mostrada na tabela 4, tendo cada código uma especificação própria. Assim:

- a) 100: o número, tamanho e posição relativa, ao eixo X do sistema terrestre, do símbolo da feição pontual;

- b) 101: a caneta traçadora, o número ou altitude da feição, e rotação destes;
- c) 111: posição da feição onde será desenhado o símbolo desta;
- d) 112: número da feição, e posição do início deste número;
- e) 113: altitude da feição, localização do início deste texto |20|.

Tabela 4: Estrutura do arquivo geral para o modo PONTO

PONTO

Fonte: DZ7 - AS Reference Manual 51-1129 e |20|

tipo de informação	primeira entrada	segunda entrada	terceira entrada	quarta entrada	código
início do ponto	código do símbolo	tamanho do símbolo	rotação do símbolo	0	100
	caneta	tamanho do texto	rotação do texto	0	101
posição do símbolo	casas decimais	X	Y	Z	111
posição do número	número do ponto	X	Y	Z	112
posição da altitude	altitude	X	Y	Z	113

4.3.2 Modo de operação LINHA ABERTA

Utilizado para restituição de feições lineares, ou de área, com contornos ou forma irregulares. Para a representação destas, os seguintes parâmetros são definidos: código, tamanho e rotação dos símbolos pontuais desenhados ao longo da feição linear; espaçamento entre os símbolos pontuais, em milímetros na escala da plotagem; distância entre o símbolo pontual e o início do símbolo linear; tipo de linha, podendo ser contínua, tracejada ou traço-ponto em diversas combinações.

Para este modo de operação, o sistema oferece os seguintes modos de digitalização:

- a) vector, neste caso o operador armazena ponto a ponto da linha, sendo esta formada por segmentos de reta que unem estes pontos;
- b) tracing, no qual o armazenamento dos pontos é incremental, exigindo que o operador apenas acompanhe a feição. Os pontos a serem armazenados, ou seja, o incremento depende de um tubo imaginário que envolve dois pontos digitalizados. Enquanto a marca flutuante não sai deste tubo, pontos não são armazenados. No momento em que a marca deixa o tubo, o programa armazena o último ponto medido e estabelece um novo tubo a partir dos dois últimos pontos. A largura deste tubo é arbitrada, sendo por default 0,5mm.
- c) spline, que armazena pontos de uma feição curva. O operador acompanha a feição armazenando pontos desta. A diferença deste modo para o tracing repousa em, se por um lado no modo tracing o registro dos pontos de uma feição curva é incremental, por outro neste modo spline todos os pontos da feição são armazenados pelo operador. Se acionado este modo ou tracing, e apenas dois pontos forem armazenados, um vetor os unirá;
- d) arc, utilizado para armazenar três pontos de um arco. O acionamento de outro modo de digitalização será possível apenas após o registro do terceiro ponto.

Os registros gerados no arquivo geral quando estes modos são acionados estão descritos nas tabelas 5, 6 e 7. Os códigos relativos a cada modo têm os seguintes significados:

- a) modo vector:
 - i) 212: a caneta plotadora segue abaixada até a posição dada;
 - ii) 213: a caneta plotadora segue levantada até a posição especificada | 20 |.

Tabela 5: Estrutura do arquivo geral para o modo VECTOR

VECTOR

Fonte: DZ7 - AS Reference Manual 51-1129 e | 20 |

tipo de informação	primeira entrada	segunda entrada	terceira entrada	quarta entrada	código
caneta abaixada	-1	X	Y	Z	212
caneta levantada	-1	X	Y	Z	213

b) modos spline e tracing:

- i) 230: pontos inicial e intermediários da curva;
- ii) 231: último ponto da curva;

Tabela 6: Estrutura do arquivo geral para os modos SPLINE e TRACING

SPLINE E TRACING

Fonte: DZ7 - AS Reference Manual 51-1129 e | 20 |

tipo de informação	primeira entrada	segunda entrada	terceira entrada	quarta entrada	código
pontos	-1	X	Y	Z	230
último pt.	-1	X	Y	Z	231

c) modo arc:

- i) 241: ponto inicial do arco;
- ii) 242: ponto intermediário do arco;
- iii) 243: ponto final do arco.

Tabela 7: Estrutura do arquivo geral para o modo ARC

ARC

Fonte: DZ7 - AS Reference Manual 51-1129 e

tipo de informação	primeira entrada	segunda entrada	terceira entrada	quarta entrada	código
início	-1	X	Y	Z	241
intermed.	-1	X	Y	Z	242
fim	-1	X	Y	Z	243

4.3.3 Modo de operação LINHA FECHADA

Para feições de área com formas regulares, este modo oferece algumas opções de primitivas gráficas que podem agilizar o trabalho de restituição. Os modos de digitalização, neste caso, possíveis são:

- a) closed vector, para a digitalização de polígonos a partir do armazenamento dos pontos de seus vértices. O primeiro vértice do polígono não é reocupado, uma vez que o fechamento deste é automático;
- b) squared 4 corner house, no qual o armazenamento de um retângulo ou paralelogramo é efetuado por três pontos, sendo automático o fechamento da feição;
- c) circle, definido a partir do armazenamento de três de seus pontos. Semelhantemente ao modo "arc", a mudança de modo de digitalização é possível somente após a digitalização do terceiro ponto.

No arquivo geral, os arquivos resultantes dos modos closed vector e squared 4 corner house são iguais ao do modo vector, uma vez que os vértices dos polígonos são conectados com vetores. O armazenamento do modo circle é apresentado na tabela 8, tendo os códigos 341, 342 e 343 a função de definir as coordenadas dos pontos de um círculo.

Tabela 8: Estrutura do arquivo geral para o modo CIRC

CIRC Fonte: DZ7 - AS Reference Manual 51-1129 e |20|

tipo de informação	primeira entrada	segunda entrada	terceira entrada	quarta entrada	código
1° ponto	-1	X	Y	Z	341
2° ponto	-1	X	Y	Z	342
3° ponto	-1	X	Y	Z	343

4.4 PACOTE DZ7 - AS

Desenvolvido para operações da mesa traçadora digital DZ7, quando esta é conectada a um computador HP 1000 através da interface HP IB, contém programas de aplicação que desempenham diferentes tarefas na obtenção do produto final da restituição, sendo estes:

- a) SHEET - SHEET PREPARATION: preparação da folha do mapa, incluindo plotagem de quadriculado ou reticulado, podendo este ser representado por cruces ou linhas cheias, e rotulação dos cantos do reticulado;
- b) C089 - DIGITAL PLOTTING: plotagem do conteúdo gráfico armazenado pelo programa B83. Através deste os arquivos gerais são acessados e verificados, e as entidades gráficas construídas.
- c) SEAR - SEACH POINT: recuperação dos elementos gráficos. Permite a obtenção da listagem de todos os pontos contidos num círculo fechado definido pelo usuário. Com estas informações, através do programa GEFIL os dados podem ser editados e armazenados |20|.

Estes programas utilizam a interface GRAPH F1, responsável por todas as operações e ações na DZ-7. Consequentemente as subrotinas que constroem as entidades gráficas, a partir do arquivo geral gerado durante a restituição, encontram-se nesta interface. Assim, subrotinas específicas permitem que o círculo ou arco de círculo sejam completamente traçados a partir de 3 pontos

digitalizados; uma curva spline seja devidamente reconstruída, e neste caso o método de H. AKIMA | 1 | é utilizado. Além da construção das entidades gráficas (arco de círculo, círculo, spline), subrotinas que controlam os movimentos (X,Y) da mesa, e das canetas traçadoras (viajar abaixada ou viajar levantada) geram o produto final, ou seja, a carta | 21 |.

4.5 MEDIDAS DE INFORMAÇÕES PARA MNT

Para gerar modelos numéricos do terreno, o sistema PLANICOMP C120 oferece três possibilidades de captura de dados, as quais diferem pelas localizações planas dos pontos, cujas coordenadas serão armazenadas. Estas diferentes possibilidades encontram-se em três programas:

- a) B70 - "RECORD TERRAIN MODEL";
- b) B71 - "RECORD TERRAIN PROFILE";
- c) B72 - "RECORD TERRAIN GRID".

No programa B70, o registro de pontos é feito percorrendo-se feições lineares, podendo estas serem curvas de nível, rios, contornos, e assim por diante. Este oferece dois modos de digitalização, permitindo registrar coordenadas individualmente, ou seja, ponto a ponto, ou incrementalmente. Neste segundo caso, o incremento pode ser arbitrado para tempo, altura ou distância horizontal, chamada viagem.

Se varredura ao longo de perfis padronizados é pretendida, o programa B71 a possibilita. A determinação dos perfis depende de parâmetros definidos anteriormente à digitalização, sendo estes:

- a) número total de perfis;
- b) comprimento do perfil em unidade do terreno;
- c) espaçamento entre perfis em unidade do terreno;
- d) ângulo horário em relação à direção positiva do eixo Y do modelo.

Estes parâmetros são dependentes da posição da marca flutuante no momento em que a digitalização é iniciada. Neste programa, as possibilidades de registro incremental de pontos são idênticas ao programa B70.

Na terceira possibilidade, ou seja, no programa B72 uma grade é previamente definida a partir dos parâmetros:

- a) comprimento dos perfis;
- b) número de perfis;
- c) intervalos em X e Y da grade.

Com a grade determinada, dois modos de digitalização são disponíveis:

- a) modo "CROSS": o qual posiciona automaticamente a marca flutuante nas intersecções da grade, e as coordenadas destes pontos são registradas após o operador posicionar a marca flutuante sobre o terreno;
- b) modo LINHA: que permite o traçado livre sobre feições, e no momento que a marca flutuante cruza uma linha da grade, este ponto é automaticamente registrado.

Como pode-se, através destes programas, armazenar as coordenadas de pontos nos arquivos gerais, o formato dos dados deve ser definido a partir do parâmetro RFORM. Para estes, o sistema aceita os formatos apresentados na tabela 9, quando a intenção é armazenar coordenadas no sistema terrestre | 19 |.

Tabela 9: Formato de dados para MNT

Fonte: PLANICOMP C120 - Manual de manejo 51-1088 s | 19 |

FORMATO DOS DADOS		RFORM
I13, 3F12.3	para NO, XG, YG, ZG	1635
I13, 2F12.3	NO, XG, YG	1625
I13, F10.3	NO, ZG	1614
2F12.3, F10.3	XG, YG, ZG	2514

5 ENTRADA DE DADOS FOTOGRAMÉTRICOS OBTIDOS COM PLANICOMP C120 NO SGI

Atualmente, se o desenvolvimento de um projeto tendo o SGI como ferramenta necessitar de dados obtidos com Fotogrametria, estes deverão estar na forma de produtos convencionais, ou seja, em cópia rígida. Isto porque, tais produtos de restituição fotogramétrica são, no SGI, transformados para a forma digital por meio de mesas digitalizadoras. Considerando que, desde meados da década de 80, o mercado cartográfico brasileiro dispõe de sistemas fotogramétricos para mapeamento digital, apesar destes não objetivarem qualquer análise ou manipulação de dados, permitem que as restituições sejam armazenadas em arquivos digitais. Portanto, para introduzir tais dados no SGI, estes são transformados da forma digital para a analógica, e esta última, através de mesas digitalizadoras, novamente transformadas para a forma digital.

Este processo digital-analógico-digital pode comprometer a qualidade dos dados e a eficiência na captura destes. Assim, se os arquivos digitais resultantes do processo de restituição forem transferidos nesta forma ao SGI, além do ganho na qualidade dos dados, o trabalho de digitalização através de mesas digitalizadoras é eliminado e, conseqüentemente, horas de trabalho dispensadas para tal.

5.1 SOLUÇÕES POSSÍVEIS

A transferência de dados adquiridos com o PLANICOMP C120 ao SGI exige o desenvolvimento de uma interface que transforme a estrutura de dados dos arquivos gerados na restituição, para uma forma digital compatível com a estrutura da base de dados do SGI. Para desenvolver esta interface, o processo de transferência destes dados deve ser definido, podendo este ser on-line ou em batch.

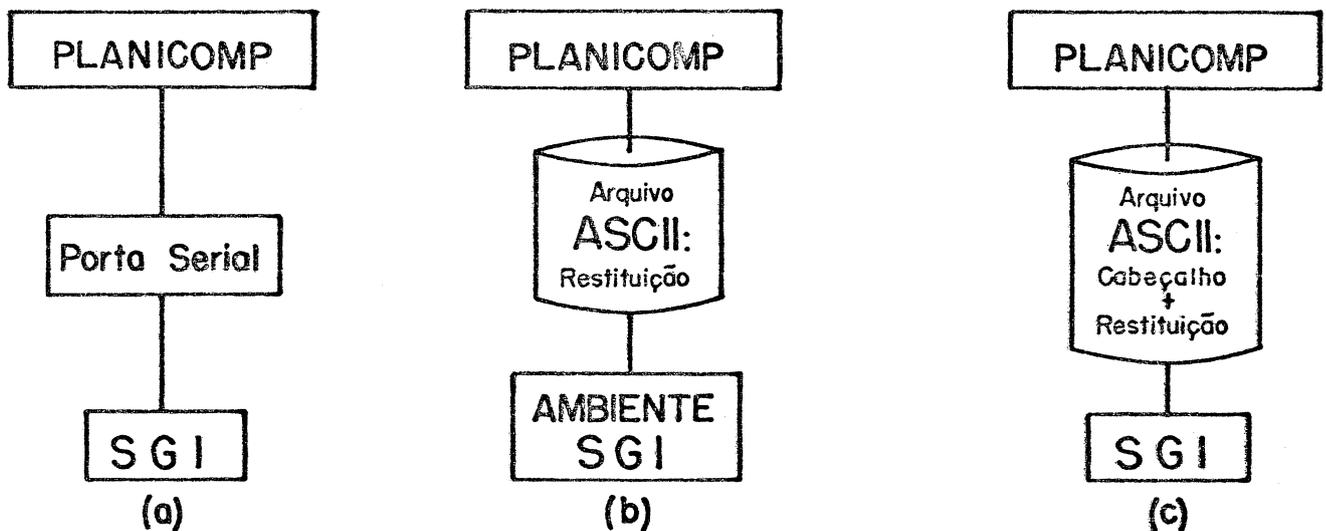
O primeiro (fig.28a) pressupõe a ligação física entre os dois sistemas. A existência de uma porta serial, entre o estereorestituidor e a mesa DZ-7 do PLANICOMP, viabiliza a conexão com o SGI. Contudo a adoção desta solução exige que ambos os sistemas estejam no mesmo espaço físico. Dessa forma, a empresa capacitada a gerar produtos fotogramétricos deve possuir tanto o PLANICOMP como o SGI.

O processo em batch é viável para situações nas quais os sistemas estão alocados em diferentes lugares. Os dados gravados em meios magnéticos são, por meio destes, transferidos de um sistema a outro. Neste caso, um software responsável pelo interfaceamento entre os sistemas desempenha a compatibilização entre as estruturas de dados destes.

A estrutura hierárquica de projetos, planos de informação e objetos do SGI conduz a dois caminhos para a solução em batch. Estes dependem da maneira na qual os dados sobre os projetos e planos de informações, relativos à região restituída, serão armazenados na base de dados. Um dos caminhos (fig.28b) assume que estes dados estejam gravados no início do arquivo de restituição, num cabeçalho deste. Neste caso, o pacote responsável pelo interfaceamento não estará inserido no ambiente do SGI, e este, por sua vez não precisará ser acessado para que a entrada de dados seja realizada. Porém, um programa deve ser desenvolvido para o PLANICOMP, com a finalidade de permitir o devido preparo dos arquivos da restituição pelo operador fotogrametrista.

Um outro caminho (fig.28c) pode ser seguido se esta interface estiver inserida no ambiente do SGI. Uma vez este ativado, os dados sobre o projeto e planos de informação poderão ser introduzidos na base de dados pelo aplicativo DEFINIÇÃO, que contém softwares específicos para esta tarefa. Os arquivos resultantes da restituição, neste caso, não necessitam de complementação, podendo ser utilizados com as informações geradas unicamente pelo PLANICOMP.

Figura 28: Possibilidades de interfacimento entre o PLANICOMP e o SGI. (a) Ligação física; (b) Ligação por software dentro do ambiente do SGI; (c) Ligação por software fora do ambiente do SGI.



5.2 DISCUSSÃO SOBRE AS POSSÍVEIS SOLUÇÕES

As diferentes tarefas desempenhadas por SIG's e sistemas fotogramétricos, conseqüentes das distintas finalidades que lhes cabem, fazem com que a maioria das instituições ou empresas usuárias de SIG's, quando necessitam em seus projetos de produtos fotogramétricos, os adquiram de empresas especializadas em produzi-los. Em tais situações, a coleta de dados fotogramétricos e a introdução destes em SIG's são procedimentos realizados em locais e épocas distintas. Nestas condições a solução em batch é a única viável.

Em vista desta consideração, qualquer solução exigindo conexão física entre os sistemas foi desconsiderada neste trabalho, no qual a integração de dados fotogramétricos adquiridos com o PLANICOMP no SGI é realizada em batch. Nesta solução, a coleta de dados fotogramétricos pelo PLANICOMP e entrada destes no SGI, são tarefas realizadas separadamente.

Sendo o SGI um software para sistemas de informações geográficas, distintamente de sistemas para mapeamento digital, objetiva não somente a produção de mapas temáticos a partir dos dados coletados, mas também análises e manipulações dos dados armazenados. Assim, qualquer

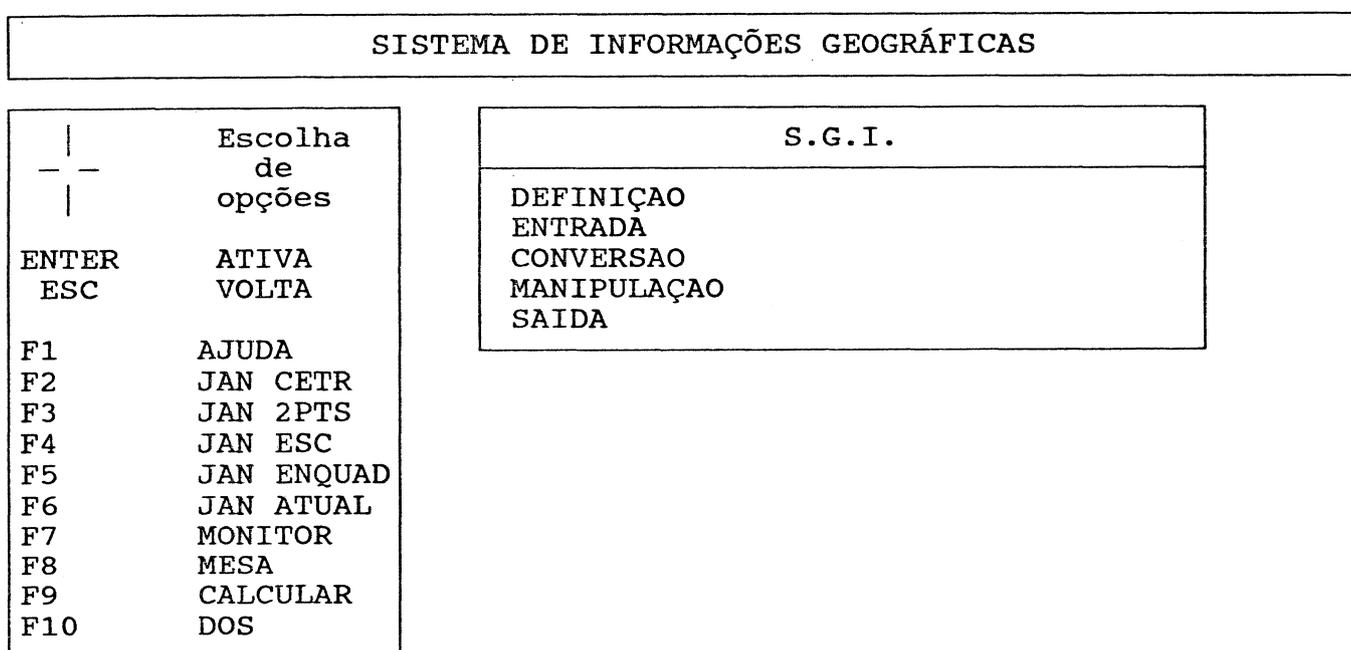
trabalho de coleta de dados no SGI não estará desvinculado de seus objetivos, os quais podem ser alcançados através dos aplicativos disponíveis no sistema.

Diante disso, o usuário do SGI ao utilizar dados fotogramétricos em seus projetos deverá, para rodar qualquer aplicativo neste sistema, ativar seu ambiente bem como introduzir os parâmetros definidores dos demais planos de informação necessários aos projetos. Portanto, mesmo que a entrada de dados fotogramétricos seja realizada externamente ao ambiente do SGI, este será sempre acessado pelo usuário. Isto direciona as melhores justificativas para a solução em batch através de interface inserida no ambiente do SGI.

5.3 INTERFACES DE ENTRADA DE DADOS FOTOGRAMÉTRICOS INSERIDAS NO AMBIENTE DO SGI

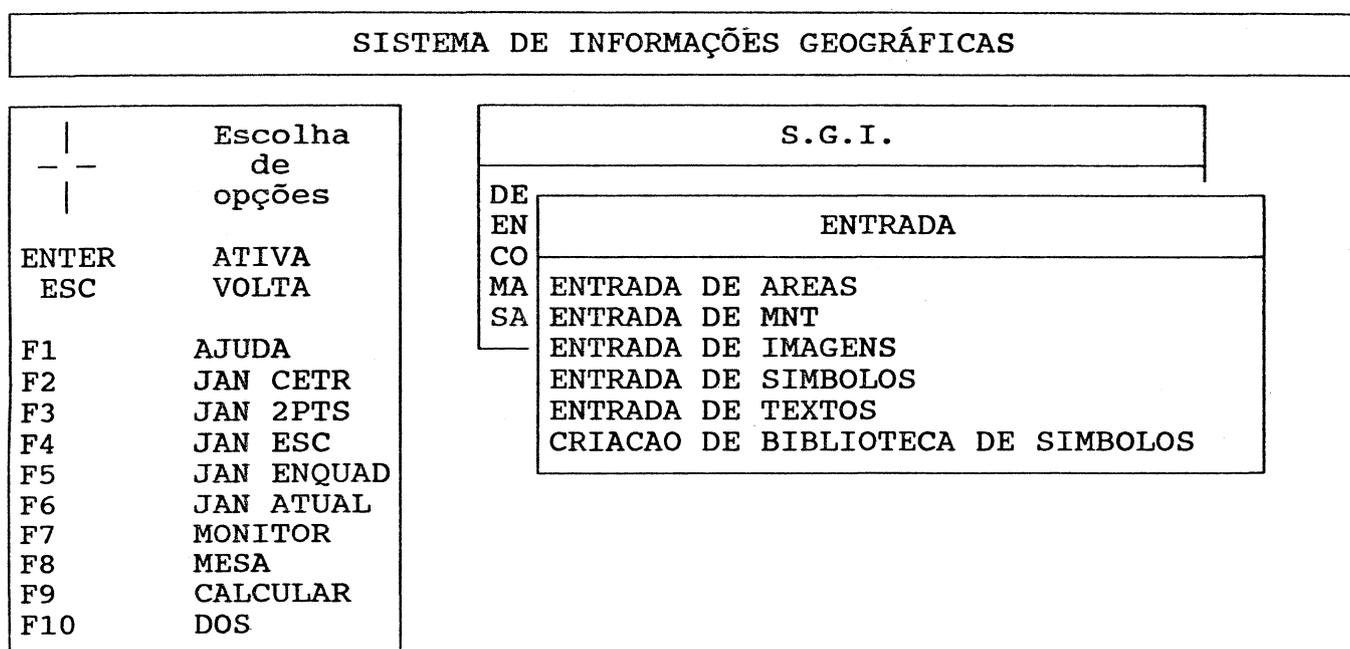
Para o ambiente do SGI conter uma interface de entrada de dados fotogramétricos, esta deve ser alocada no submódulo ENTRADA. No momento em que o usuário ativa o ambiente do sistema, o menu principal deste, contendo os cinco submódulos que o compõe incluindo o de ENTRADA, lhe é apresentado como mostra a figura 29.

Figura 29: Tela do menu principal do ambiente SGI



Ao acionar o submódulo ENTRADA, uma nova tela é apresentada ao usuário, na qual são mostradas todas as possibilidades de entrada de dados atualmente presentes no sistema (fig.30). Pela concepção do SGI, entre as opções de tipos ou categorias de dados possíveis de serem armazenados na base de dados, os dados adquiridos com o PLANICOMP enquadram-se, tanto na categoria de dados poligonais como para MNT. Assim, para a entrada de dados de tais categorias, as interfaces estarão alocados em ambos os submódulos, ou seja: Entrada de área e Entrada de MNT.

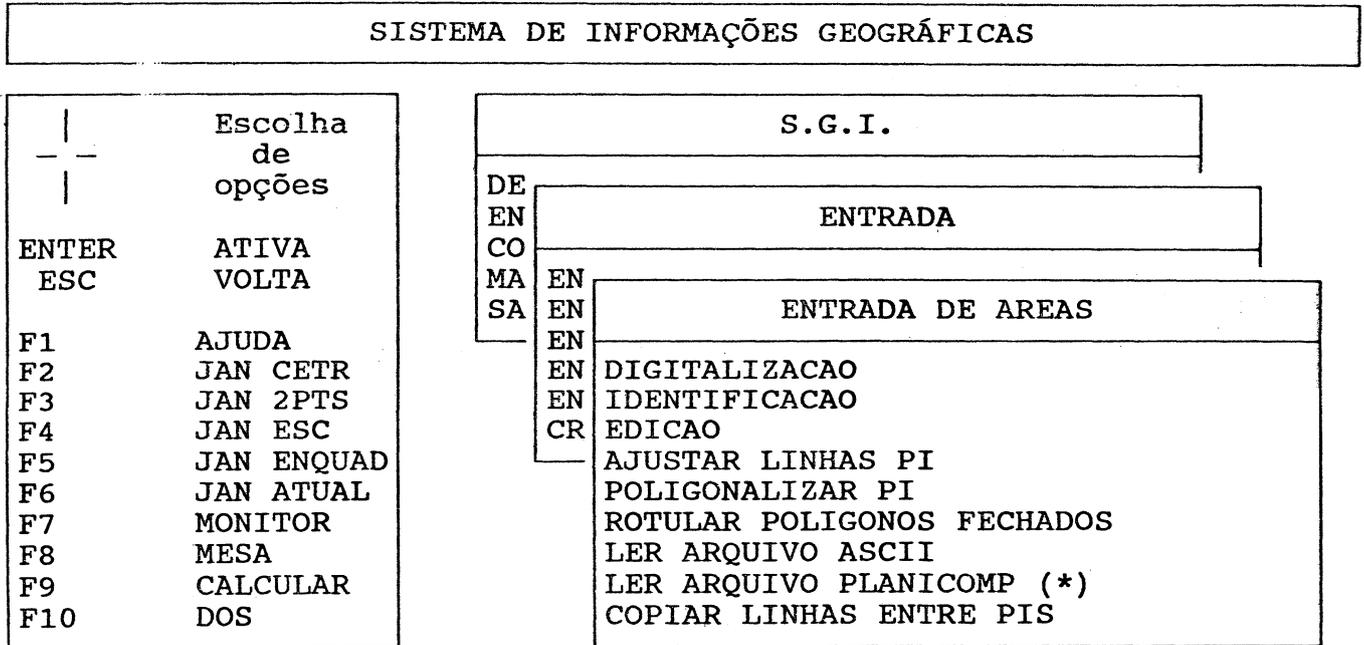
Figura 30: Tela do menu do submódulo ENTRADA



Composto por seis aplicativos, o submódulo ENTRADA DE AREAS permite, além da digitalização de feições por mesa digitalizadora incluindo a edição destas, a construção da topologia de estruturas vetoriais e a leitura de arquivos ASCII de feições lineares. Considerando que a digitalização de um modelo fotogramétrico, ou seja a restituição deste pelo PLANICOMP, resulta na representação de feições de parte da superfície terrestre por dados da categoria poligonal, a leitura dos arquivos ASCII gerados neste processo deve ser efetuada por meio de um aplicativo deste

submódulo. A inclusão deste aplicativo no menu é mostrada na figura 31, estando este indicado por um asterisco.

Figura 31: Tela do menu do submódulo ENTRADA DE ÁREA



Por se tratar de um sistema fotogramétrico apoiado computacionalmente, o PLANICOMP apresenta recursos que extrapolam a restituição de modelos fotogramétricos para mapeamento convencional. Entre as potencialidades deste sistema, encontram-se os programas que permitem a restituição de pontos para MNT. As coordenadas de pontos assim obtidas enquadram-se, no SGI, na categoria de dados MNT. Consequentemente, os aplicativos responsáveis pela coleta e armazenamento destes dados na base de dados do SGI, devem ser parte do submódulo ENTRADA DE MNT.

Como na categoria de dados MNT do SGI, tanto isolinhas como pontos isolados podem gerar modelos numéricos do terreno, os aplicativos capazes de coletar tais dados provenientes do PLANICOMP devem prever ambas as situações. Por isso, no submódulo ENTRADA DE MNT um menu, contendo os aplicativos para ler arquivos de isolinhas e pontos de perfil obtidos fotogrametricamente, é adicionado no ambiente SGI, como pode ser observado nas figuras 32 e 33.

Figura 32: Submódulo ENTRADA DE MNT

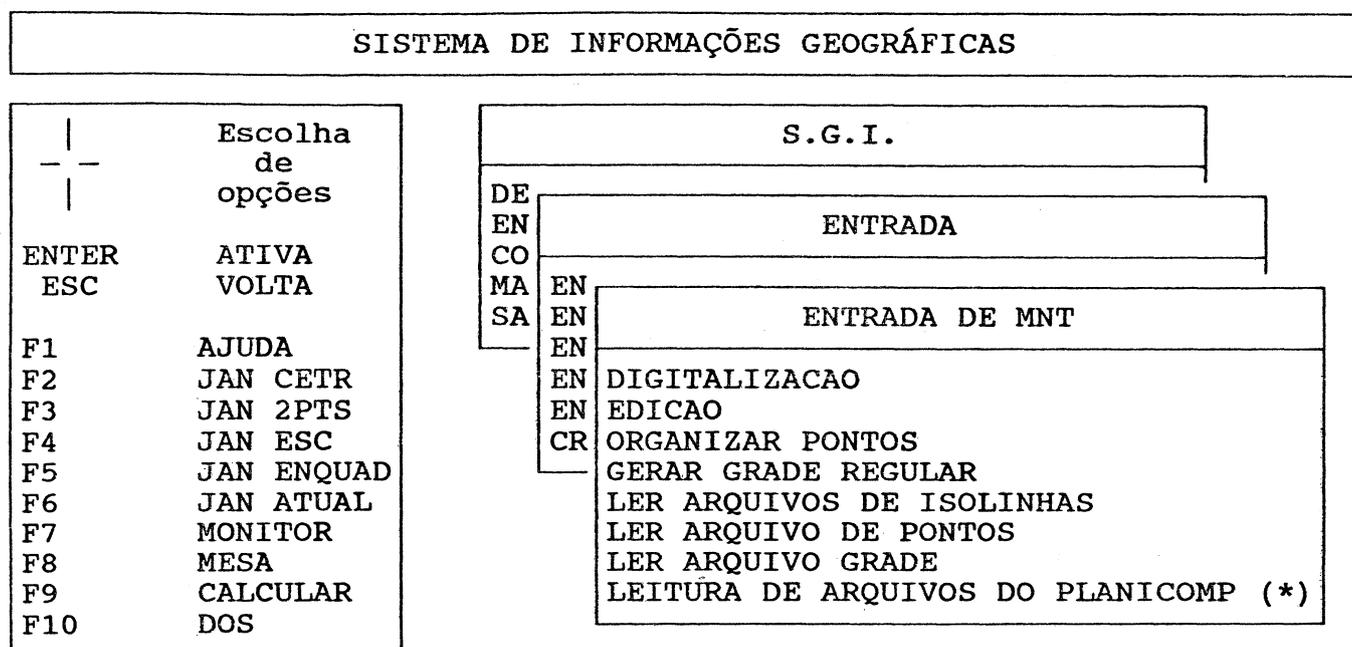
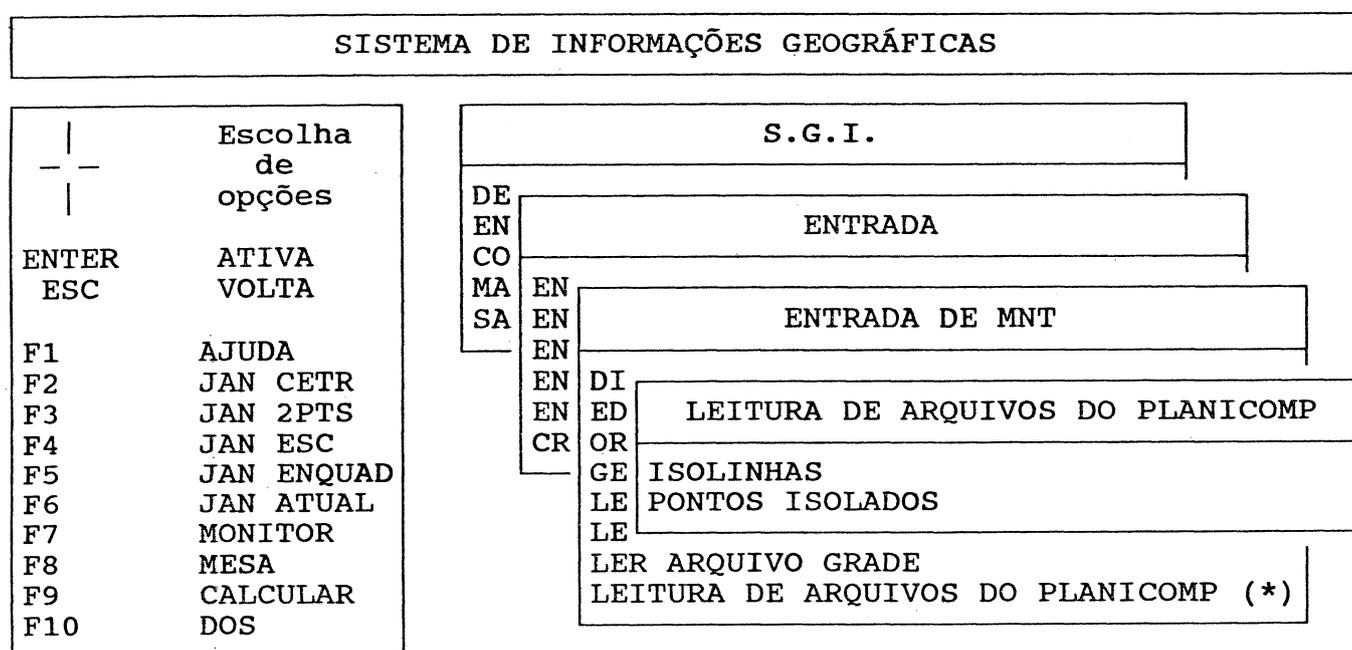


Figura 33: Submódulo ENTRADA DE MNT com menu para dados do PLANICOMP



Estas considerações conduzem ao desenvolvimento de três interfaces responsáveis por realizar a conexão do PLANICOMP C120 com o SGI, capazes de abranger as potencialidades do primeiro úteis ao segundo.

6 IMPLEMENTAÇÃO

6.1 IMPLEMENTAÇÃO DA INTERFACE PARA DADOS POLIGONAIS

Nos arquivos ASCII resultantes da digitalização de feições, para mapeamento destas, de um modelo fotogramétrico pelo PLANICOMP C120, as primitivas gráficas definidas por cada modo de digitalização são identificadas por códigos específicos. Estes são gravados em cada registro de pontos digitalizados da feição, juntamente com as coordenadas destes. A partir destes códigos e das coordenadas dos pontos, o traçado completo das feições restituídas pela mesa traçadora DZ-7 é controlado pela interface GRAPH F1.

Por outro lado, uma feição linear é sempre armazenadas na base de dados do SGI pelas coordenadas dos pontos que a compõe e a quantidade destes. Como, neste sistema não há o uso de primitivas gráficas para o armazenamento de dados gráficos, o traçado destas feições é realizado por segmentos de reta que unem cada par de pontos de uma linha, consecutivamente armazenados. Assim, além da necessidade da definição dos pontos que caracterizam uma feição, para feições curvas quanto maior o número de pontos melhor suas definições.

Conseqüentemente, uma interface responsável por compatibilizar os dados gráficos das feições restituídas pelo PLANICOMP para armazená-las no SGI, deve definir todos os pontos das linhas identificadas por primitivas como spline, arc e circ, uma vez que apenas alguns pontos destas são gravados nos arquivos gerados no PLANICOMP.

6.1.1 Modos de digitalização: VECTOR, CLOSED VECTOR e HOUSE

O uso destes modos de digitalização para restituir feições de modelos fotogramétricos, significa que estas feições serão construídas pela união dos pontos registrados através de segmentos

de reta. Por isso, a interface do SGI deve, quando reconhecer os códigos de VECTOR, contar o número de pontos da feição e armazenar as coordenadas destes.

Pelos significados dos códigos 212 e 213 que caracterizam estes modos, o primeiro ponto de uma linha será sempre registrado com o código 213, e os demais, incluindo o último, com 212. Porém, a interface do SGI deve prever algumas situações possíveis de ocorrer durante o processo de restituição, e que podem ocasionar erros na definição das feições.

Uma destas situações acontece quando o operador fotogrametrista inicializa repetidamente este modo de digitalização, sem mudar a marca flutuante de lugar. Neste caso, um mesmo ponto é armazenado repetida e consecutivamente com o código 213. A interface do SGI deve, detectando este fato, armazenar apenas um destes registros na base de dados. Da mesma forma que para o código 213, quando pontos intermediários ou finais da feição, ou seja código 212, forem registrados repetidamente, esta repetição deve ser eliminada pela interface do SGI, que neste caso deve armazenar apenas um ponto.

Se o operador fotogrametrista inicializa este modo de digitalização porém, muda de modo sem registrar qualquer feição, apenas um ponto com código 213 será registrado no arquivo ASCII da restituição. Neste caso, este ponto não será armazenado no SGI, ou seja deverá ser eliminado.

Como, para o armazenamento das feições na base de dados do SGI é limitado o número máximo de pontos de uma linha, este fato deve ser controlado pela interface para evitar a impossibilidade da leitura do arquivo ASCII. Assim, se uma linha possui mais pontos que o máximo permitido, a interface a dividirá em duas ou mais linhas conforme necessário.

6.1.2 Modo de digitalização: SPLINE e TRACING

Quando estes modos de digitalização são acionados, sobre os pontos registrados da feição será traçada uma spline pela mesa traçadora. Nesta situação, uma subrotina da interface GRAPH F1 reconstitue a spline a ser traçada, pelo método de AKIMA $|1|, |21|$. Devido a isto, este mesmo método é também utilizado para determinar os pontos a serem armazenados no SGI. Porém, se apenas dois pontos forem digitalizados no PLANICOMP para definir a feição, estes deverão ser conectados por um

segmento de reta. Conseqüentemente, nesta situação a spline não será desenvolvida, e apenas os dois pontos armazenados.

No método de AKIMA, para n pontos conhecidos, uma curva é interpolada por uma spline cúbica, ou seja, entre cada dois pontos é determinado um polinômio de terceiro grau. Das condições que garantem a continuidade da curva nos nós (pontos conhecidos), são determinados os coeficientes dos polinômios. Neste método, para a condição de continuidade das tangentes, estas são definidas no terceiro de cinco pontos consecutivos, em função das inclinações das retas que passam a cada dois deste pontos. Conseqüentemente, no início e fim da curva, dois pontos devem ser extrapolados. Nesta extrapolação, é assumido que os três primeiros ou últimos pontos da curva pertencem a uma parábola $|1|$.

Para não ultrapassar o limite máximo de pontos de uma linha permitido pelo SGI, o número de pontos interpolados deve ser fixo e o número máximo de nós, de cada linha, estabelecido. Esta última condição exige que algumas linhas sejam divididas. Portanto, na determinação das tangentes dos pontos iniciais e finais de cada linha, cinco situações possíveis são previstas:

- a) curvas abertas:
 - i) se o número de nós não ultrapassa o limite, a extrapolação nos nós inicial e final é efetuada;
 - ii) se o número de nós ultrapassa o limite:
 - 1) na primeira divisão da linha, para o primeiro ponto desta são definidos dois pontos adicionais por extrapolação; e para o último ponto, os dois primeiros pontos da próxima divisão são utilizados para definir a tangente neste;
 - 2) nas divisões intermediárias, os dois últimos pontos da divisão anterior e os dois primeiros pontos da divisão posterior são usados na determinação das tangentes no primeiro e último ponto respectivamente;

- 3) na última divisão da spline, para o cálculo da tangente no primeiro ponto desta utiliza-se os dois últimos pontos do segmento anterior; e para a tangente ao último ponto, extrapola-se dois pontos à linha;
- b) curvas fechadas: como o ponto de fechamento da curva é registrado duas vezes, no primeiro registro deste, ou seja no nó inicial, os dois últimos pontos anteriores ao fechamento da curva serão adotados para a determinação da tangente a este; já para o último registro do ponto de fechamento, nó final, são utilizados os dois pontos posteriores ao início da curva.

Na definição matemática da spline, a função polinomial que determina y em função de x ,

$$y = p_0 + p_1 \cdot x + p_2 \cdot x^2 + p_3 \cdot x^3 \quad (5.1)$$

é parametrizada para permitir o procedimento computacional, assim:

$$\begin{aligned} x &= p_0 + p_1 \cdot t + p_2 \cdot t^2 + p_3 \cdot t^3 \\ y &= q_0 + q_1 \cdot t + q_2 \cdot t^2 + q_3 \cdot t^3 \end{aligned} \quad (5.2)$$

onde: $p_i, q_i \ i = 0,1,2,3$ são coeficientes do polinômio
 t parâmetro variando entre 0 e 1

Os coeficientes do polinômio, entre os pontos i e $i+1$, são determinados como dado abaixo:

$$\begin{aligned} p_0 &= x_i \\ p_1 &= r \cdot \cos \theta_i \\ p_2 &= 3(x_{i+1} - x_i) - r(\cos \theta_{i+1} + 2 \cdot \cos \theta_i) \\ p_3 &= -2(x_{i+1} - x_i) + r(\cos \theta_{i+1} + \cos \theta_i) \\ q_0 &= y_i \\ q_1 &= r \cdot \sin \theta_i \\ q_2 &= 3(y_{i+1} - y_i) - r(\sin \theta_{i+1} + 2 \cdot \sin \theta_i) \\ q_3 &= -2(y_{i+1} - y_i) + r(\sin \theta_{i+1} + \sin \theta_i) \end{aligned} \quad (5.4)$$

onde:

$$r = \sqrt{[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]} \quad (5.5)$$

$$\begin{aligned}\cos\theta_i &= A / \sqrt{(A^2 + B^2)} \\ \text{sen}\theta_i &= B / \sqrt{(A^2 + B^2)}\end{aligned}\quad (5.6)$$

$$\begin{aligned}A &= \omega_{i-1} \cdot a_{i-1} + \omega_i \cdot a_{i+1} \\ B &= \omega_{i-1} \cdot b_{i-1} + \omega_i \cdot b_{i+1}\end{aligned}\quad (5.7)$$

$$\begin{aligned}\omega_{i-1} &= |a_i \cdot b_{i+1} - a_{i+1} \cdot b_i| \\ \omega_i &= |a_{i-2} \cdot b_{i-1} - a_{i-1} \cdot b_{i-2}|\end{aligned}\quad (5.8)$$

$$\begin{aligned}a_i &= x_{i+1} - x_i \\ b_i &= y_{i+1} - y_i \quad i = 1, n\end{aligned}\quad (5.9)$$

Para definir dois pontos adicionais no início e final da curva, extrapolando-a por uma parábola, tem-se no primeiro e último pontos, considerando suas coordenadas (x_3, y_3) e (x_n, y_n) , respectivamente:

$$\begin{aligned}x_2 &= 3 \cdot x_3 - 3 \cdot x_4 + x_5 \\ y_2 &= 3 \cdot y_3 - 3 \cdot y_4 + y_5\end{aligned}\quad (5.10)$$

$$\begin{aligned}x_1 &= 3 \cdot x_2 - 3 \cdot x_3 + x_4 \\ y_1 &= 3 \cdot y_2 - 3 \cdot y_3 + y_4\end{aligned}$$

$$\begin{aligned}x_{n+1} &= 3 \cdot x_n - 3 \cdot x_{n-1} + x_{n-2} \\ y_{n+1} &= 3 \cdot y_n - 3 \cdot y_{n-1} + y_{n-2}\end{aligned}\quad (5.11)$$

$$\begin{aligned}x_{n+2} &= 3 \cdot x_{n+1} - 3 \cdot x_n + x_{n-1} \\ y_{n+2} &= 3 \cdot y_{n+1} - 3 \cdot y_n + y_{n-1}\end{aligned}$$

6.1.3 Modo de digitalização: ARC e CIRC

Quando estas primitivas gráficas são usadas para a digitalização de feições circulares, o operador do restituidor registra três pontos da feição. A partir destes pontos a interface GRAPH F1 reconstituirá a feição a ser traçada pela DZ-7. Portanto, a interface responsável pelo armazenamento destas feições no SGI deverá desempenhar esta mesma tarefa, ou seja, calcular os demais pontos da feição.

Se estes modos de digitalização forem acionados durante o trabalho de restituição, o programa B83 permite que o operador mude de modo de digitalização somente após o registro do

terceiro ponto da feição. Conseqüentemente, no arquivo ASCII da restituição, estes pontos estarão sempre em seqüência, tendo a interface do SGI que armazená-los, e a partir deles determinar os demais pontos da feição.

O cálculo dos demais pontos da feição segue a definição matemática da circunferência num plano. Se três pontos de uma circunferência (x_1, y_1) , (x_2, y_2) e (x_3, y_3) são conhecidos pode-se determinar o raio e centro desta, pelas equações (5.12) e (5.13).

$$x_0 = \frac{b_1 \cdot c_2 - b_2 \cdot c_1}{a_1 \cdot b_2 - a_2 \cdot b_1} \quad (5.12)$$

$$y_0 = \frac{a_1 \cdot c_2 - a_2 \cdot c_1}{a_2 \cdot b_1 - a_1 \cdot b_2}$$

sendo: $a_1 = x_2 - x_1$ $a_2 = x_3 - x_1$

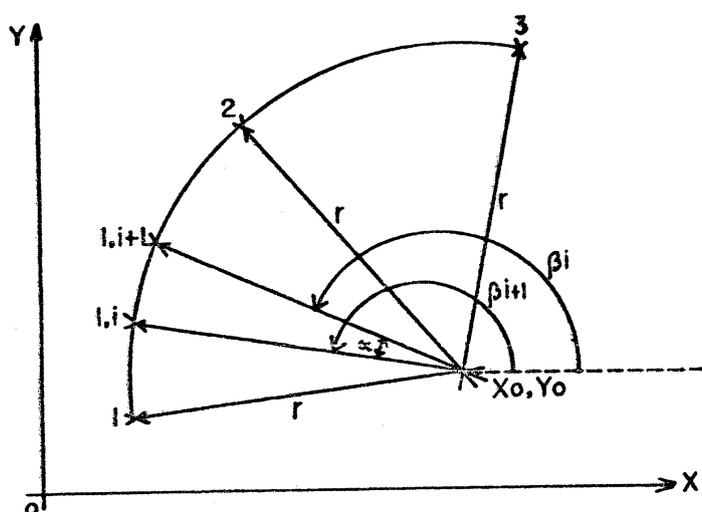
$b_1 = y_2 - y_1$ $b_2 = y_3 - y_1$

$$c_1 = \frac{x_1^2 + y_1^2 - x_2^2 - y_2^2}{2} \quad c_2 = \frac{x_1^2 + y_1^2 - x_3^2 - y_3^2}{2}$$

$$r^2 = \frac{[(x_1 - x_0)^2 + (y_1 - y_0)^2 + (x_2 - x_0)^2 + (y_2 - y_0)^2 + (x_3 - x_0)^2 + (y_3 - y_0)^2]}{3} \quad (5.13)$$

Com estes elementos, centro e raio, pode-se calcular (x, y) de todos os demais pontos da feição em função de um incremento angular (α) , como dado pelas equações (5.14).

Figura 34: Incremento angular entre dois pontos de uma circunferência



$$\begin{aligned}x_{i,1} &= x_0 + r \cdot \cos \beta \\y_{i,1} &= y_0 + r \cdot \sin \beta \quad i = 1, 2, 3, \dots\end{aligned} \quad (5.14)$$

Nas equações (5.14) β é o ângulo formado pelo raio vetor do ponto a ser calculado com a direção positiva do eixo X. Este ângulo β , dado pela equação (5.15), é incrementado pelo ângulo α , que define a distância angular entre dois pontos consecutivos da circunferência, assim:

$$\beta_{i+1} = \beta_i \pm \alpha \quad (5.15)$$

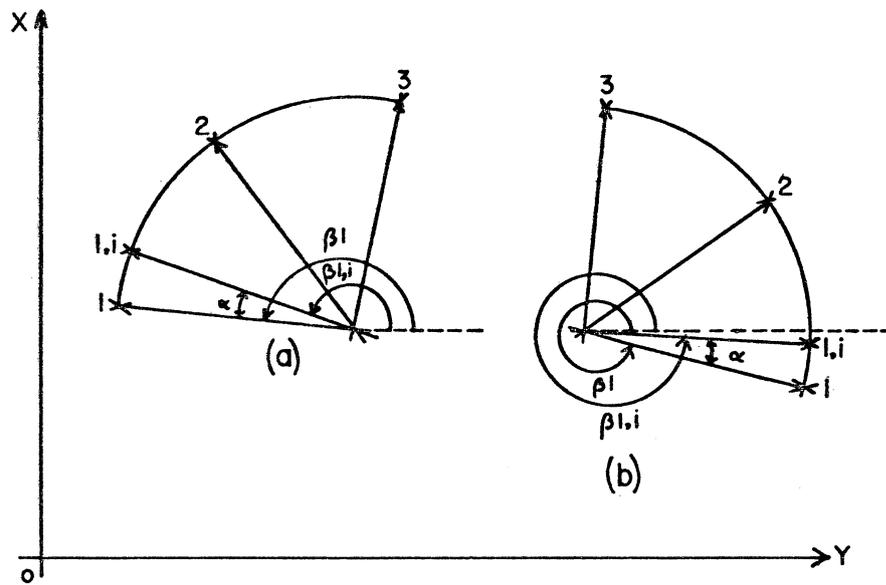
O incremento ou decremento de β , ou seja adição ou subtração de β por α , depende da localização dos pontos, primeiro, intermediário e último, armazenados da feição. A posição destes define se a feição deve ser varrida no sentido horário ou anti-horário, como mostra a figura 35.

A definição da amplitude do incremento α considera duas condições:

- que a distância entre dois pontos consecutivos da feição seja suficientemente pequena para definí-la o mais completamente possível, porém;
- que o número de pontos armazenados não seja excessivo acarretando desperdício de memória.

Com base nestas condições, a acuidade visual do operador foi adotada como critério, uma vez que se este operador percorresse a feição digitalizando todos os seus pontos, a máxima resolução obtida condiziria com sua percepção visual. O valor da acuidade visual adotado equivale à distância de 0,2mm na escala do modelo. Outro fato a ser observado diz respeito à ampliação do sistema de lentes do estereorestituidor. Este podendo ser de 8 ou 16 vezes, diminui a distância no terreno relativa a acuidade visual, uma vez que a imagem é ampliada.

Figura 35: Sentido de rotação do incremento angular α



(a) decremento

(b) incremento

Sob estas considerações, o incremento α é definido matematicamente como descrito na equação (5.16).

$$\alpha = \frac{x.E}{r.A} \quad (5.16)$$

onde: x é a acuidade visual do operador
 E é o denominador da escala do modelo
 r é o raio da circunferência
 A é o fator de ampliação do sistema de lentes

6.1.4 Algoritmo

Como as interfaces pertencem ao ambiente do SGI, a primeira parte do software recupera informações armazenadas na base de dados quando da definição do projeto e PI ativos. Além disso, apresenta diálogos padronizados na tela do vídeo, em função dos quais o usuário fornecerá as informações que, juntamente com as do projeto e PI, serão necessárias para rodar este aplicativo. Para tal são utilizadas as funções de diálogo existentes no módulo da BIBLIOTECA BÁSICA. Com isso, o software está devidamente preparado para, lendo os registros do arquivo ASCII, armazenar os dados de interesse na base de dados do SGI.

A leitura do arquivo ASCII e as operações necessárias para compatibilizar as informações provenientes do PLANICOMP com o SGI são executadas na segunda parte do software. O raciocínio no qual se baseia o algoritmo desta etapa, considerou que primeiramente os diferentes modos de digitalização devem ser detectados. Em seguida, todas as coordenadas (x,y) de cada linha devidamente armazenadas. A partir destas coordenadas, funções específicas para arc,circ e spline calculam os demais pontos das linhas. Este algoritmo é esquematizado na figura 36 e apresentado no Anexo 1.

Os resultados deste são vetores das coordenadas x e y dos pontos das linhas e a quantidade destes. Estas informações são, na última parte do software, armazenadas na base de dados, tanto no arquivo de cabeçalho como no arquivo de dados das linhas.

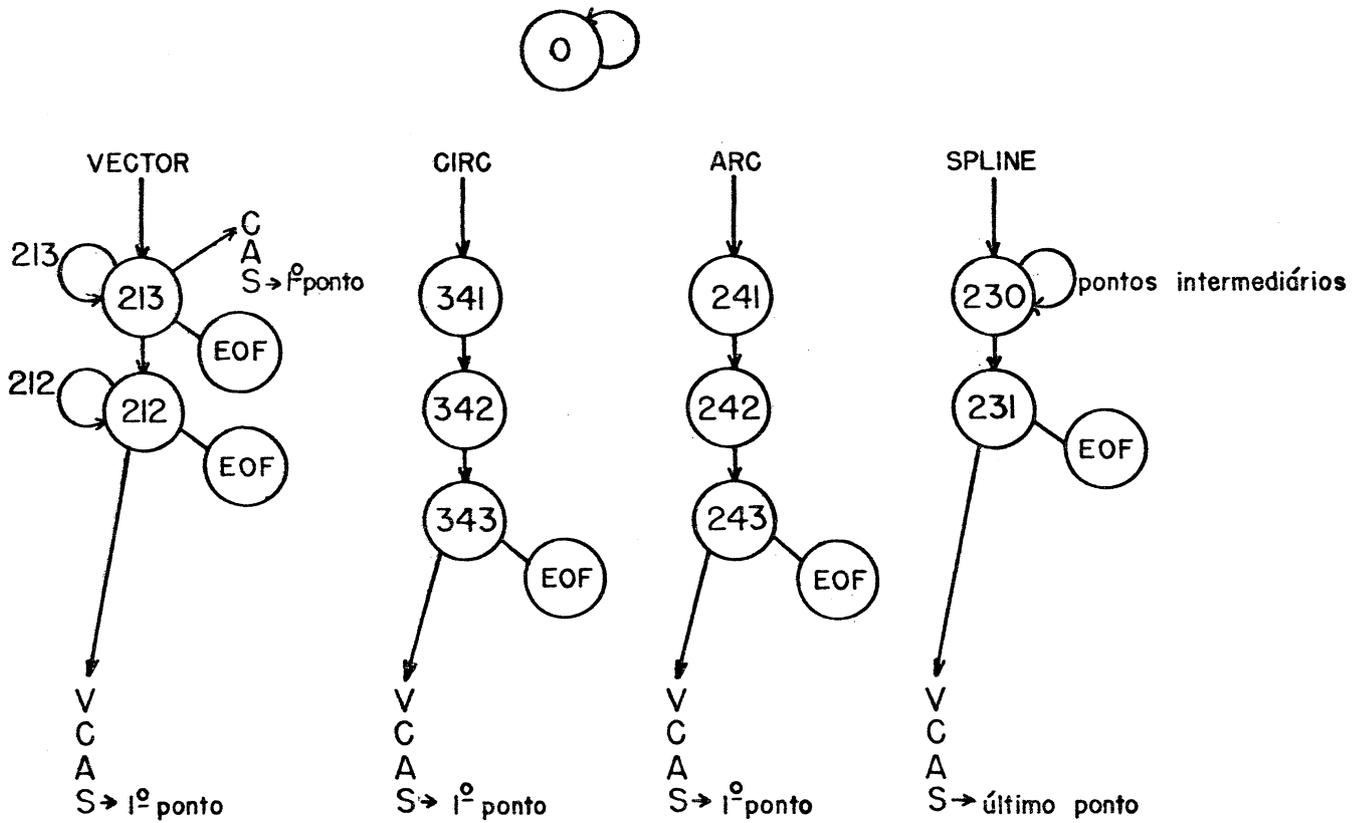
Na construção deste software foram desenvolvidas as seguintes funções:

- a) pega_registro: cujo objetivo é ler os registros dos arquivos ASCII e enviar ao programa principal as coordenadas (x,y) e o código de cada ponto;
- b) ARC_CIRC: que a partir das coordenadas (x,y) dos três pontos da linha digitalizados, escala da restituição e fator de ampliação do sistema de lentes, gera os demais pontos do arco de círculo;
- c) CIRCULO: que com os mesmos dados da função ARC_CIRC, gera os demais pontos de um círculo;

- d) BETA_ARC: que determina o ângulo formado pelo raio vetor do primeiro ponto armazenado do círculo, ou arco de círculo, com a direção positiva do eixo X do sistema de coordenadas terrestres;
- e) SPLI_DZ7: que define os pontos de uma curva interpolada entre n pontos dados, pelo método de AKIMA |1|.

Estas funções são apresentadas no Anexo 2.

Figura 36: Algoritmo da interface para dados poligonais



6.2 IMPLEMENTAÇÃO DAS INTERFACES PARA MNT

6.2.1 Leitura de arquivos de isolinhas

Sendo as isolinhas, neste caso curvas de nível, representações de feições curvas da superfície terrestre, a digitalização destas no PLANICOMP utiliza as primitivas gráficas spline ou tracing. Assim, no arquivo ASCII resultante desta digitalização, os registros das coordenadas dos pontos armazenados terão formatos relativos a estes modos.

No SGI as isolinhas pertencentes a categoria de dados MNT são armazenadas na base de dados pelas coordenadas dos pontos das feições, quantidade dos pontos definidores de cada linha e a coordenada z, cota ou altitude, de cada isolinha. Conseqüentemente, a interface capacitada a ler e compatibilizar os registros gravados no arquivo ASCII da restituição, está baseada no mesmo algoritmo da interface que armazena isolinhas na categoria de dados poligonais.

Devido à necessidade de armazenamento da coordenada z de cada linha, a este algoritmo é acrescentada a leitura das coordenadas z de cada ponto. Porém, isto exige a consideração de dois fatores. O primeiro constata o fato de que no SGI apenas uma coordenada z para cada isolinha é armazenada. O segundo, diz respeito à sensibilidade dos encoders do PLANICOMP, isto é, por melhor que seja o trabalho do operador fotogrametrista em acompanhar uma mesma isolinha, diferenças nos dígitos decimais em coordenada z serão registradas. Considerando estes fatores, a interface assume a média aritmética, das coordenadas z, de todos os pontos de uma mesma isolinha, para o armazenamento do valor da cota desta. Por ser uma interface para dados da categoria MNT, a interpolação de uma curva entre os pontos conhecidos da linha não é realizada. Isto porque, na geração do modelo numérico do terreno, uma grade regular é criada, sobre a qual são interpolados os pontos que definem a superfície do terreno. (Anexo 3)

6.2.2 Leitura de pontos isolados

Se por um lado o PLANICOMP permite o registro de pontos para MNT, e por outro o SGI pode a partir de pontos de coordenadas (x,y,z) conhecidas gerar MNT, a digitalização de pontos no PLANICOMP com os programas B70, B71 e B72 podem ser úteis a projetos desenvolvidos com o SGI.

Neste caso, o mesmo software que executa a transferência de pontos de amostras (x,y,z), da categoria MNT, de um arquivo ASCII para a base de dados do SGI pode ser utilizado, se o formato de registro dos dados no PLANICOMP for RFORM = 2514, no qual apenas as coordenadas (x,y,z) de cada ponto serão armazenadas.

Com base neste raciocínio, uma interface para este propósito não precisa ser desenvolvida, e a solução para esta situação consiste em acionar o programa asc_pt2.exe já existente no ambiente do SGI, quando for chamada a função PONTOS ISOLADOS do menu LEITURA DE ARQUIVOS DO PLANICOMP do submódulo ENTRADA DE MNT.

7 METODOLOGIA DE TRABALHO PARA ENTRADA DE DADOS ADQUIRIDOS COM O PLANICOMP C120 NO SGI

Os procedimentos tanto no PLANICOMP como no SGI dependem da solução adotada neste trabalho, ou seja, as interfaces entre os sistemas estão contidas no ambiente do SGI. Conseqüentemente, as informações necessárias para definir o projeto e os planos de informação são armazenadas na base de dados do SGI através do módulo DEFINIÇÃO, que o prepara devidamente para receber os dados obtidos com o PLANICOMP.

Com isso, a metodologia de trabalho do operador fotogrametrista no PLANICOMP permanece inalterada. Apenas informações adicionais às restituições necessárias para definir projeto e PI's serão anotadas na folha de controle mostrada na figura 37. Esta é uma das vantagens desta solução, para situações nas quais usuários do SGI necessitam de dados da superfície terrestre levantados fotogrametricamente.

7.1 DIGITALIZAÇÃO DO MODELO FOTOGRAMÉTRICO

Antes de iniciar a digitalização das feições o trabalho deve ser planejado, estando incluídas nesta etapa as definições dos overlays e dos modos de digitalização, com os quais as diferentes feições serão restituídas.

Na restituição numérica, as feições existentes na parte da superfície terrestre a ser mapeada são digitalizadas em diferentes níveis de informação, correspondentes às diferentes classes e subclasses que compõem a base cartográfica. Assim, cada classe ou subclasse definirá um overlay, e cada overlay será restituído num determinado arquivo. Estes arquivos são criados pelo programa GEFIL, o qual exige a definição prévia dos tamanhos destes. Se o tamanho máximo permitido para

Ao serem gerados, os arquivos recebem denominações que os identificam, as quais iniciam com as letras FIL seguidas de três dígitos arbitrados pelo operador. Estas denominações são anotadas na folha de controle, bem como o tamanho do arquivo e a classe ou subclasse referente àquele overlay.

Definir previamente os modos de digitalização correspondentes às diferentes feições otimiza o trabalho de restituição para o operador do restituidor. Qual modo de digitalização usar depende da forma da feição linear. Assim, se a feição é uma curva irregular, como no caso de um rio, o modo spline é o mais adequado. Para feições contínuas, porém compostas por segmentos de retas, como alguns trechos de estradas, o modo vector é usado. No caso de feições circulares, como cantos de quadras ou poços, há a possibilidade de arc ou circ, respectivamente. Se a feição possuir contornos regulares porém for fechada, como divisas de propriedades ou construções, o modo closed vector é apropriado.

Após concluídas as restituições dos overlays, as informações adicionais constantes na folha de controle devem ser fornecidas. As coordenadas dos cantos da folha ou modelo restituído, necessárias ao programa SHEET, no SGI definirão a região geográfica do projeto. Além destas coordenadas, as informações constantes do lado direito destas na folha de controle (fig. 37), significam:

- a) sistema de coordenadas: ou projeção cartográfica utilizada na determinação dos pontos aerotriangulados, podendo ser UTM, topográfico local ou GAUSS;
- b) escala da restituição: definida pelo seu denominador;
- c) unidade métrica: dos pontos aerotriangulados, podendo ser metros ou quilômetros;
- d) datum horizontal e meridiano central: fornecidos se para os pontos aerotriangulados forem usados os sistemas UTM ou GAUSS.

7.2 ENTRADA DE DADOS FOTOGRAFÉTRICOS NO SGI

Após recebidos os arquivos ASCII das restituições fotogramétricas e as folhas de controle destas, o usuário do SGI, para proceder à entrada destes dados no sistema, deve primeiramente ativá-lo. No primeiro menu do sistema, o módulo DEFINIÇÃO deve ser acionado. Neste módulo, a primeira

providência é chamar a função DEFINIR PROJETO ATIVO. Para que esta possa ser executada, as seguintes informações devem ser fornecidas:

- a) nome do projeto: de no máximo três caracteres;
- b) atributos do novo projeto, sendo estes:
 - i) projeção: correspondendo ao sistema de coordenadas usado para referenciar os pontos definidos na aerotriangulação. Se o sistema for topográfico local, a opção NENHUMA do SGI deve ser escolhida;
 - ii) escala: denominador da escala de restituição;
 - iii) unidade: unidade métrica dos pontos aerotriangulados;
 - iv) datum horizontal: podendo ser SAD, para o SAD 69; CORREGO, para Córrego Alegre; ou OUTROS, para um datum diferente dos previstos pelo sistema. Se os pontos aerotriangulados foram determinados num sistema topográfico local, a opção NDA o contempla.
- c) coordenadas envolventes do projeto: coincidentes com as coordenadas dos cantos da folha da restituição.

7.2.1 Entrada de dados de mapeamento

Após a definição do projeto, segue a criação dos planos de informação, os quais corresponderão aos níveis de informação restituídos. A criação dos PI's é possibilitada pela função DEFINIR/CRIAR PI ATIVO do módulo DEFINIÇÃO. Quando chamada, esta função solicita as seguintes informações:

- a) nome do PI: com no máximo quatro caracteres;
- b) parâmetros do novo PI, sendo:
 - i) categoria: neste caso polígonos;
 - ii) escala do mapa: denominador da escala do modelo;
 - iii) número de classes: se cada subclasse ou classe for restituída num determinado nível de informação, cada PI terá apenas uma classe;

- iv) rótulo da classe: nome da subclasse ou classe restituída, como rios, estradas pavimentadas.

Este mesmo procedimento deve ser repetido para todos os planos de informação restituídos.

Com os PI's criados, pode ser realizada a leitura dos arquivos ASCII gerados com o PLANICOMP. Como a interface entre os sistemas está contida no módulo ENTRADA, este deve ser acionado, e neste, a função LER ARQUIVO PLANICOMP chamada. Nesta função, as seguintes informações são solicitadas ao usuário para que as leituras dos arquivos possam ser executadas:

- a) nome do PI: necessariamente já criado;
- b) ampliação do sistema de lentes: do PLANICOMP, podendo ser de 8 ou 16 vezes;
- c) nome do arquivo ASCII: nome do arquivo de restituição do nível de informação correspondente a este PI.

Desta forma, todos os níveis de informação restituídos podem ser armazenados na base de dados do SGI. Após executada esta função, é realizada a edição dos PI's. Naturalmente, quanto melhor a qualidade da restituição, menor a necessidade de edição no SGI. A primeira tarefa, neste caso, é corrigir os erros de restituição, como linhas deslocadas, problemas nas finalizações de linhas fechadas, podendo para isto serem usadas as funções SEPARAR e SUPRIMIR linhas. A segunda tarefa consiste em definir os nós, através da função SEPARAR linhas.

Com a edição realizada, as estruturas topológicas podem ser criadas, através das funções AJUSTAR LINHAS PI, INSERIR CENTRÓIDES e POLIGONALIZAR PI.

7.2.2 Entrada de dados para MNT

Na criação de PI's para dados que gerarão MNT, a diferença para o caso anterior consiste na definição da categoria dos dados, sendo obviamente MNT. Com os PI's criados, no módulo ENTRADA DE MNT, o submódulo LEITURA DE ARQUIVOS DO PLANICOMP é acionado. Neste, duas interfaces são apresentadas ao usuário: a que permite entrada de dados de isolinhas e entrada de

pontos isolados. Em ambas as funções são requisitados o nome do PI e o nome do arquivo ASCII gerado no PLANICOMP.

Igualmente ao comentado para dados de mapeamento, da qualidade da restituição depende a quantidade de edição a ser realizada. Se esta for necessária, o SGI possibilita para linhas, suprimí-las ou alterar suas cotas; e para pontos, listá-los, suprimí-los ou também alterar suas cotas.

8 RESULTADOS

As interfaces desenvolvidas neste trabalho foram testadas por dois exemplos de restituição numérica resultantes de digitalização com o PLANICOMP C120. Além destes, um exemplo de perfilamento de modelo fotogramétrico foi utilizado para testar a interface que adquire dados de pontos isolados de um arquivo ASCII para a categoria MNT do SGI. Estes três exemplos foram cedidos pela Empresa Esteio Engenharia e Aerolevantamentos S.A.

O primeiro resulta de um trabalho realizado com fins comerciais, contratado pela CODEPLAN - Companhia de Desenvolvimento do Planato Central. A região geográfica restituída está situada na latitude 15°45'00"Sul e longitude 47°45'00"Oeste. Deste projeto, a região utilizada para verificar as interfaces desenvolvidas corresponde a folha 181 compreendida entre as coordenadas UTM N = 8233040.569m, E = 158658.959m e N = 8237756.947, E = 165426.842m, restituída com a metodologia usualmente adotada pela empresa. Como nestas situações, para a restituição de qualquer tipo de feição, são empregados os modos de digitalização VECTOR, SPLINE e TRACING, somente estes são aqui encontrados.

As feições da superfície terrestre desta região foram digitalizadas em 13 níveis de informação relacionados na folha de controle apresentada na figura 38. Nesta também estão relacionados os números dos arquivos ASCII correspondentes a cada nível, as coordenadas dos cantos da folha, além das informações adicionais exigidas pelo SGI. Com a restituição em mãos foram testadas as interfaces para aquisição de dados poligonais, e isolinhas para dados MNT.

Devido a este primeiro exemplo não contemplar todos os possíveis modos de digitalização previstos na interface para dados poligonais, uma restituição de feições fictícias foi realizada contendo os seguintes modos: VECTOR, TRACING, SPLINE, ARC, CIRC e SQUARED 4 CORNER HOUSE. A escala

de restituição foi 1/8000, com o sistema de lentes que amplia 8 vezes a imagem do modelo, gravada num arquivo ASCII de número 972.

Figura 38: Folha de controle da restituição

Folha/modelo: __181 - CODEPLAN_____ data: __/__/__									
operador: _____									
n° arq	tam. arq	overlay	n° arq	tam. arq	overlay				
618		estr.pavimentas	628		vegetação				
619		estr. sem pavim.	629		cultura				
620		estr. vicinais	630		mestras				
621		edificações	631		secudárias				
622		cercas	632		matas				
623		bueiros	652		secudárias				
624		rios							
625		rios							
626		valas							
627		lago/alagado							
coordenadas dos cantos da folha <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>2</td> <td>3</td> </tr> <tr> <td>1</td> <td>4</td> </tr> </table>			2	3	1	4	sistema de coordenadas: UTM escala do modelo: 1/30000 unidade métrica pts. apoio: m datum horizontal: SAD 69 meridiano central: 45°Oeste ampliação das lentes: 8		
2	3								
1	4								
2.E 158658.959_ 3.E 165357.658_ N_8237655.646 N_8237756.947_									
1.E 158729.533_ 4.E 165426.842_ N_8233040.569 N_8233142.108_									

No terceiro exemplo, o perfilamento de um modelo foi executado com o programa B71 - RECORD TERRAIN PROFILE, para registros incrementais dos pontos pelo modo distância horizontal. Este incremento de distância entre dois pontos de um perfil foi escolhido como 40,0m, sendo o intervalo

entre perfis de 80,0m, com 18 perfis de 2660,0m de comprimento cobrindo o modelo. O azimute adotado foi de 0°, significando que os perfis serão percorridos em direções paralelas ao eixo Y do sistema de coordenadas terrestres. Para possibilitar o armazenamento destes na base de dados do SGI, o formato de saída dos dados foi determinado para RFORM = 2514.

8.1 ENTRADA NO SGI DOS DADOS DA RESTITUIÇÃO DA FOLHA 181 DO PROJETO CODEPLAN

No SGI, a primeira providência a ser tomada após ativado o sistema é definir o projeto ativo. Para tal, após acionada a função DEFINIR PROJETO ATIVO do módulo DEFINIÇÃO, as informações requisitadas ao usuário, e as fornecidas especificamente para este projeto são mostradas nas figuras 39, 40, 41, 42 e 43.

Em seguida, o plano de informação, correspondente ao arquivo ASCII do primeiro nível de informação a ser lido, é criado estando conseqüentemente ativado para a entrada dos dados. As informações a serem fornecidas para esta função são apresentadas nas figuras 44 e 45.

Figura 39: Ativação do Projeto - nome do projeto

30-Dez-92

Definir Projeto Ativo

21:00:31

ESC	FIM
^	VOLTA
CR	
Max. 3 caract.	

Projetos Disponiveis:

Nome do novo projeto: **est**

Figura 40: Ativação do projeto - projeção

30-Dez-92

Definir Projeto Ativo

21:01:10

ESC	FIM
^	VOLTA
0	NENHUMA
1	UTM/SAD
2	UTM/CORREGO
3	UTM/OUTROS
4	MERCATOR
5	GAUSS
6	LAMBERT
	1:1000000
7	LAMBERT
	GERAL
8	POLICONICA
9	CIL. EQUID.
10	POLAR ESTER
11	BIPOLAR OBL
12	ALBERS
13	MILLER

Projetos Disponiveis:

Nome do novo projeto: **est**

Atributos do novo projeto:

Projecao: **1**

Figura 41: Ativação do Projeto - escala

30-Dez-92

Definir Projeto Ativo

21:01:30

ESC	FIM
^	VOLTA
>	1.

Projetos Disponiveis:

Nome do projeto: **est**

Atributos do novo projeto:

Projeção: **utm/sad**Escala: **1/30000**

Figura 42: Ativação do Projeto - datum horizontal e coordenadas envolventes

30-Dez-92 Definir Projeto Ativo 21:01:41

ESC	FIM
^	VOLTA
0 coordenadas geodesicas	
1 coordenadas planas	

Projetos Disponiveis:

Nome do novo projeto: **est**

Atributos do novo projeto:

Projeção: **utm/sad**

Escala: **1/30000**

Unidade: **metros**

Opção para retangulo envolvente: **1**

Figura 43: Ativação do Projeto - retângulo envolvente

30-Dez-92 Definir Projeto Ativo 21:01:41

ESC	FIM
^	VOLTA

Projetos Disponiveis:

Nome do projeto: **est**

Atributos do novo projeto:

Projeção: **utm/sad**

Escala: **1/30000**

Unidade: **metros**

Retangulo envolvente: **coordenadas planas**

X1:158658.959 X2:165426.842

Y1:8233040.569 Y2:8237756.947

Figura 44: Ativação do plano de informação - nome do plano

30-Dez-92

Ativação do plano de informação

21:03:23

ESC	FIM
^	VOLTA
CR	fim
4 caracteres	

Projeto. . : **est**
 Nome do plano . . : **estp**

Parametros do Novo PI
 Categoria:

Escala do mapa:

Criação da tabela
 Numero de classes :

Figura 45: Ativação do plano de informação - parâmetros do PI

30-Dez-92

Ativação do plano de informação

21:07:32

ESC	FIM
^	VOLTA
1	poligonos
2	MNT

Projeto. . : **est**
 Nome do plano . . : **estp**

Parametros do Novo PI
 Categoria: **1**

Escala do mapa: **30000**

Criação da tabela
 Numero de classes : **1**
 Classe 1 Rotulo : **estrada paviment**

Quando a função LER ARQUIVO PLANICOMP é chamada, esta apresenta uma tela ao usuário (fig. 46), na qual constam as informações necessárias à sua execução. A última informação introduzida é justamente o nome do arquivo ASCII onde estão armazenadas as feições do plano de informação em questão.

Figura 46: Leitura do arquivo de restituição

30-Dez-92 Leitura do arquivo de restituição do PLANICOMP em ASCII

ESC	FIM
^	VOLTA
Maximo 8 carac	

Projeto : **est**
 Nome do PI : **estp**
 Ampliação da imagem do modelo: **8**
 Nome do arquivo ASCII-PLANICOMP : **f618.dz7**

Após lido o arquivo ASCII, a imagem das estradas pavimentadas restituídas pode ser visualizada no vídeo de alta resolução, com a cor escolhida pelo usuário. Estes procedimentos, desde criar PI ativo até ler o arquivo ASCII são repetidos para os demais níveis de informação restituídos. Alguns resultados obtidos são apresentados na figura 47, 48, 49 e 50.

Figura 47: Armazenamento da restituição das estradas pavimentadas no SGI

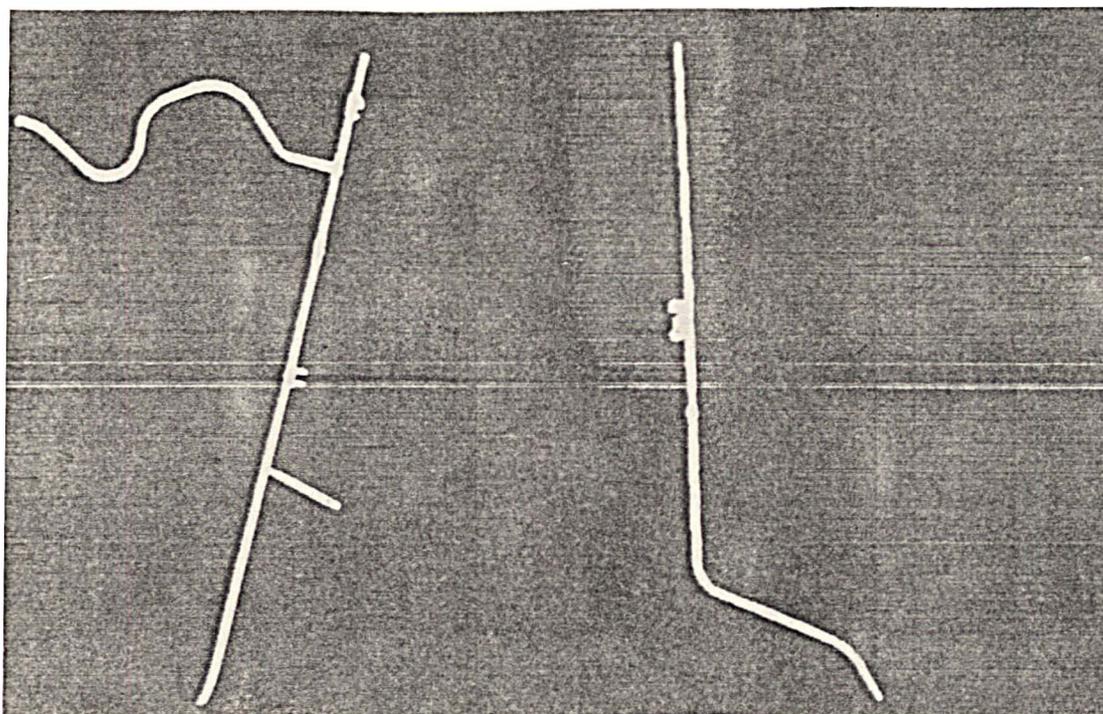


Figura 48: Armazenamento da restituição das estradas pavimentadas, vicinais e construções no SGI

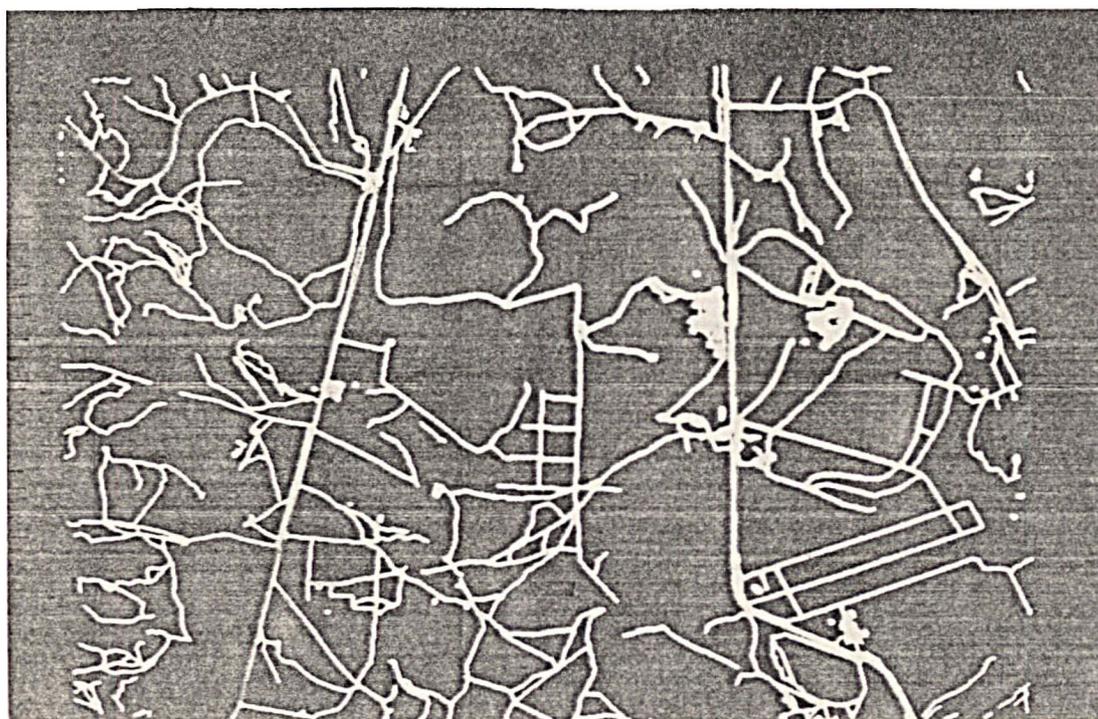


Figura 49: Armazenamento da restituição da planimetria no SGI

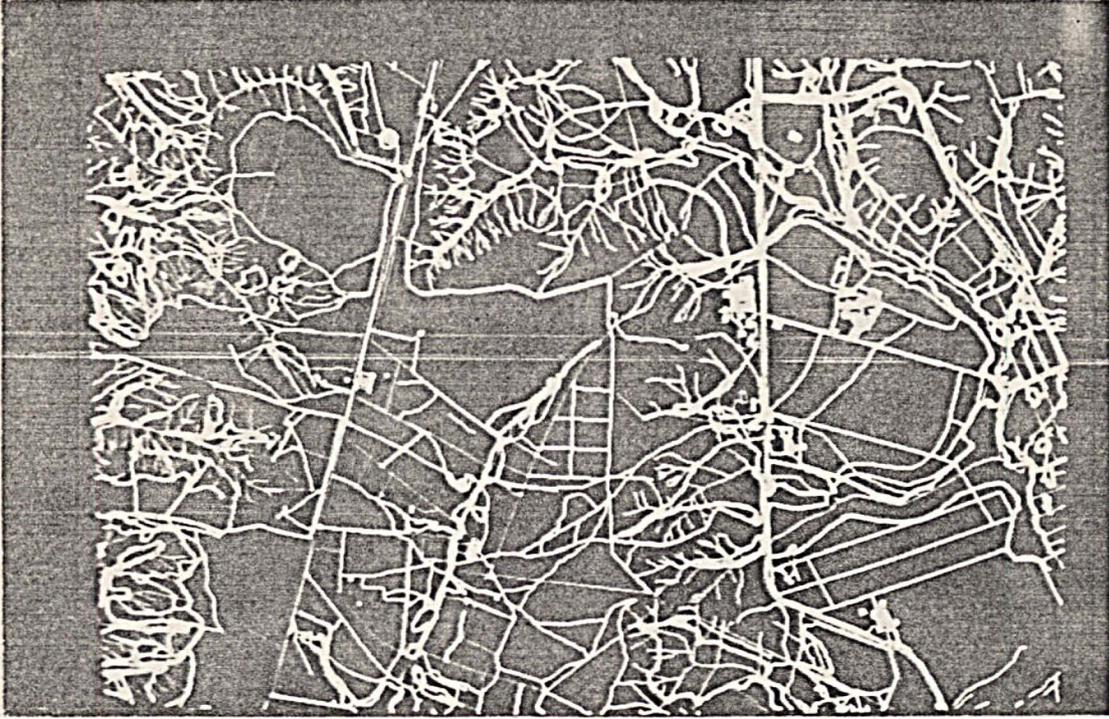
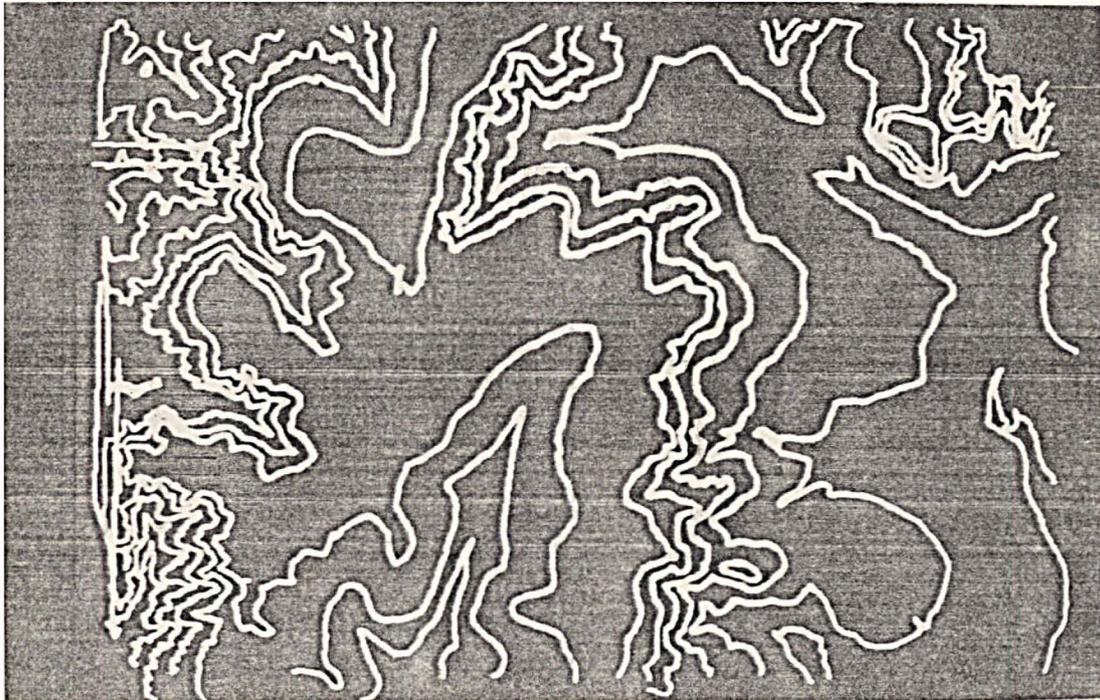


Figura 50: Armazenamento da restituição das curvas de nível mestras no SGI



Pela observação destas restituições, a existência de falhas na digitalização das feições é percebida. Isto ocorre por serem estes os primeiros resultados obtidos com o PLANICOMP C120, sobre os quais não havia sido realizado qualquer trabalho de edição. Devido a isto, foram também armazenados no SGI estes mesmos níveis de informações, nos quais edições foram executadas. Estes podem ser observados nas figuras 51, 52 e 53. Obviamente, quanto melhor a qualidade da restituição menor o trabalho de edição a ser realizado no SGI. Por outro lado, por melhor que seja a restituição, algum tempo à edição no SGI deve ser dispensado, pois por este trabalho de edição os PI's são preparados para a construção da topologia, capacitando-os aos aplicativos dos submódulos CONVERSÃO, MANIPULAÇÃO e SAÍDA.

Figura 51: Armazenamento da restituição editada das estradas pavimentadas, vicinais e construções

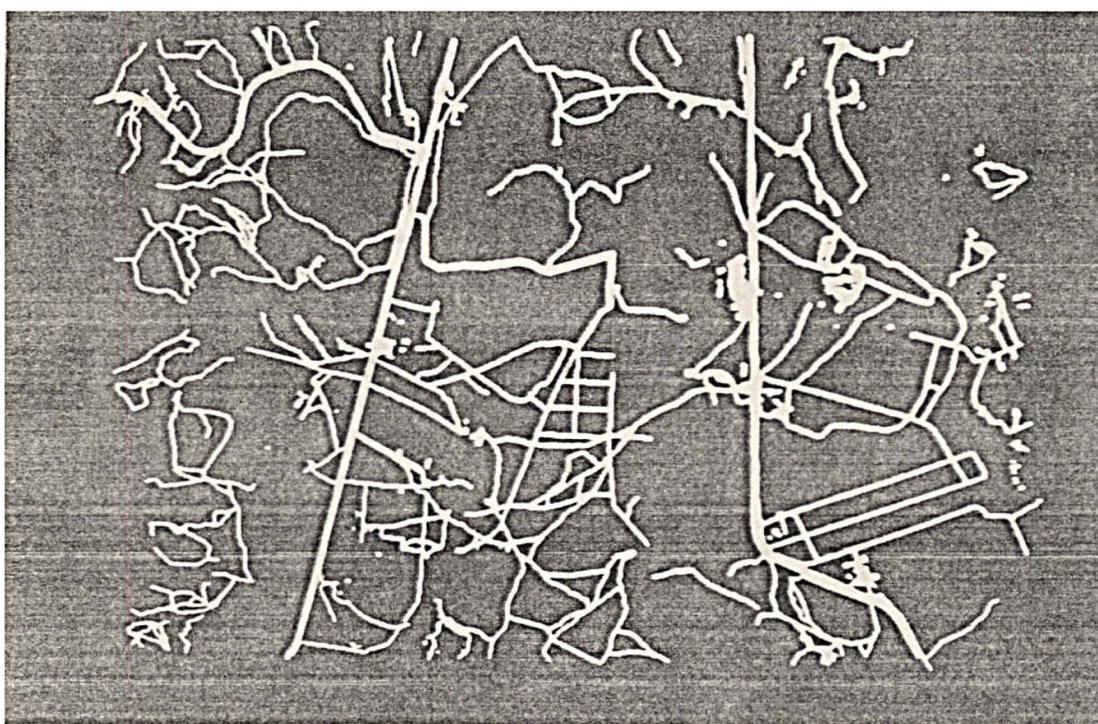


Figura 52: Ampliação de um detalhe das estradas pavimentadas, vicinais e construções

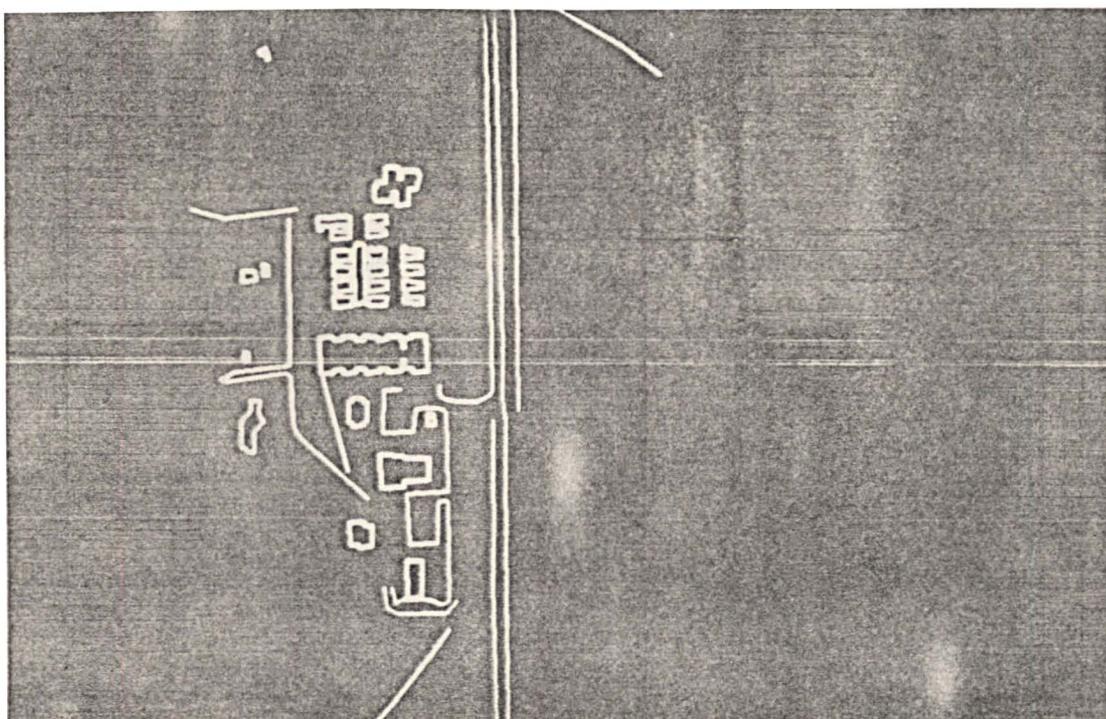
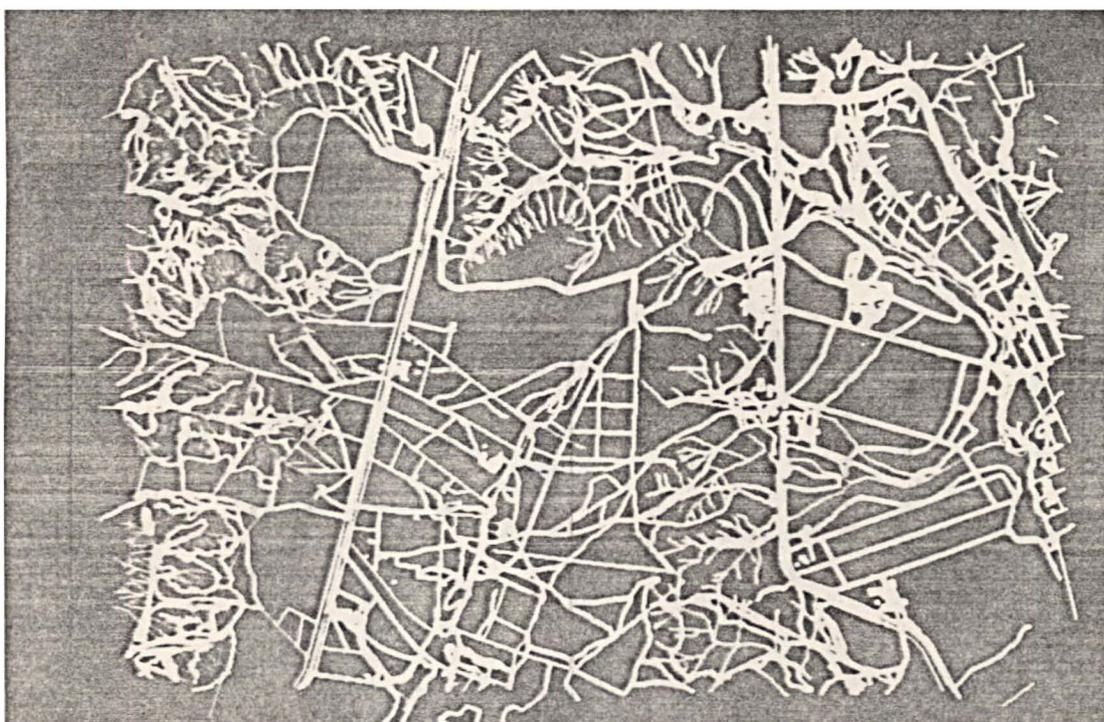


Figura 53: Armazenamento da restituição editada da planimetria no SGI



8.2 ENTRADA NO SGI DA RESTITUIÇÃO DO EXEMPLO COM FEIÇÕES FICTÍCIAS

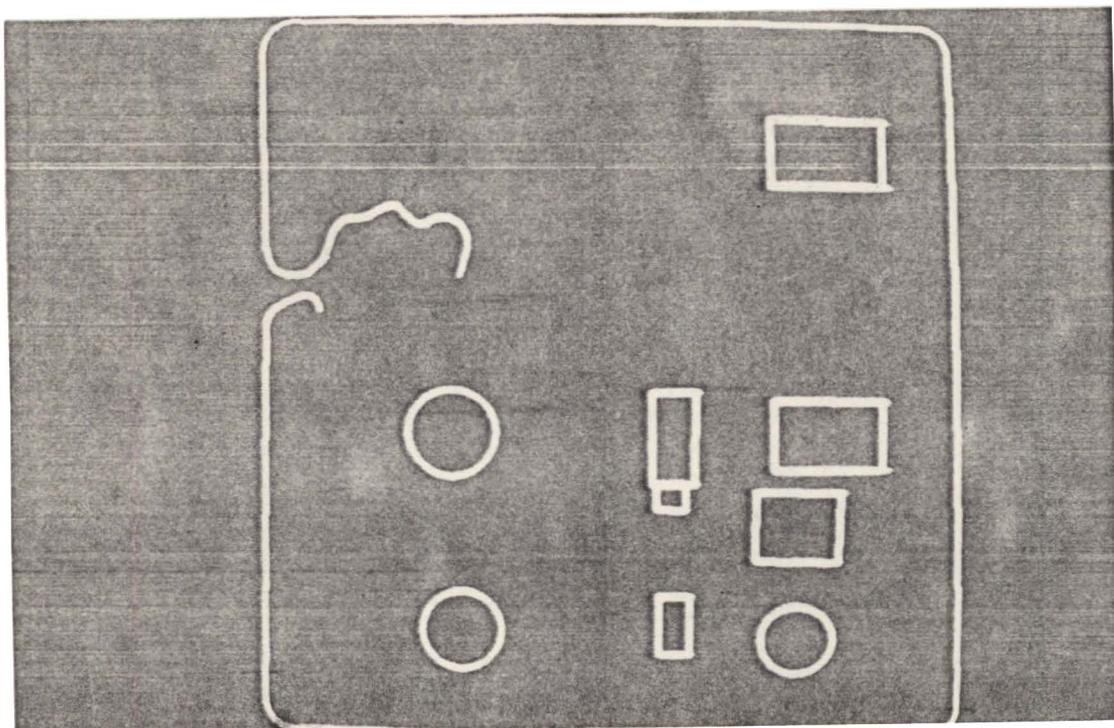
Como já mencionado, um exemplo contendo feições fictícias digitalizadas com os modos: VECTOR, TRACING, SPLINE, ARC, CIRC e SQUARED 4 CORNER HOUSE serviu como meio para verificar totalmente a interface para entrada de dados de mapeamento no SGI. Isto é verdadeiro, uma vez que o modo de digitalização CLOSED POLYGON têm, assim como o SQUARED 4 CORNER HOUSE, no arquivo ASCII os mesmos códigos do modo VECTOR, ou seja, 212 e 213.

Após criados o projeto e o plano de informação para receber estes dados, a função LER ARQUIVO PLANICOMP foi chamada, e o resultado de sua execução é mostrado na figura 54. Pela análise deste resultado e do arquivo ASCII da restituição apresentado no anexo 4, é possível afirmar que para qualquer modo de digitalização previsto na interface, esta responde de acordo com o esperado. Isto conduz a algumas conclusões, relacionadas às situações previstas no desenvolvimento da interface:

- a) No início do arquivo há um ponto registrado com o código 213, porém não pertencente a feição alguma. Consequentemente foi eliminado pelo programa;
- b) Para os quatro arcos de círculo existentes nos cantos da área, a digitalização dos três pontos pertencentes a estes foi realizada, em três casos no sentido horário e um no anti-horário. Em ambas as situações a função calculou devidamente os demais pontos da feição;
- c) Para os círculos, apesar de nos três casos, os três pontos pertencentes a cada um terem sido digitalizados no sentido anti-horário, há pontos localizados em qualquer dos quadrantes do círculo;
- d) As splines, armazenadas pelo modo SPLINE ou TRACING, foram desenvolvidas como o esperado, tanto nas interpolações entre os pontos intermediários como nas extrapolações nos pontos iniciais e finais.
- e) A digitalização das retas entre os arcos de círculos com o modo VECTOR, pelos pontos inicial e final destas (2 pontos em cada uma), após os arcos de círculo terem sido armazenados, prova o armazenamento correto dos pontos destas feições na

base de dados do SGI, uma vez que a conexão destes com os arcos de círculo está de acordo com o esperado.

Figura 54: Resultado do teste usando os modos de digitalização aceitos pela interface



8.3 ENTRADA NO SGI DE ISOLINHAS PARA MNT

Para verificar esta interface, a restituição das curvas mestras do projeto CODEPLAN foi utilizada. Com isso, no mesmo projeto criado no SGI para testar a interface para entrada de dados de mapeamento, ou seja projeto est, foi criado um PI para dados MNT. Esta etapa é mostrada na figura 55.

Após as edições necessárias, executou-se os seguintes passos para gerar o modelo numérico do terreno:

- a) ORGANIZAR PONTOS;
- b) GERAR GRADE REGULAR;

- c) REFINAR GRADE REGULAR;
- d) VISUALIZAR 3D.

Como resultado foi gerada imagem tridimensional do terreno mostrada na figura 56.

Figura 55: Ativação do plano de informação - parâmetros do PI

30-Dez-92

Ativação do plano de informação

21:07:32

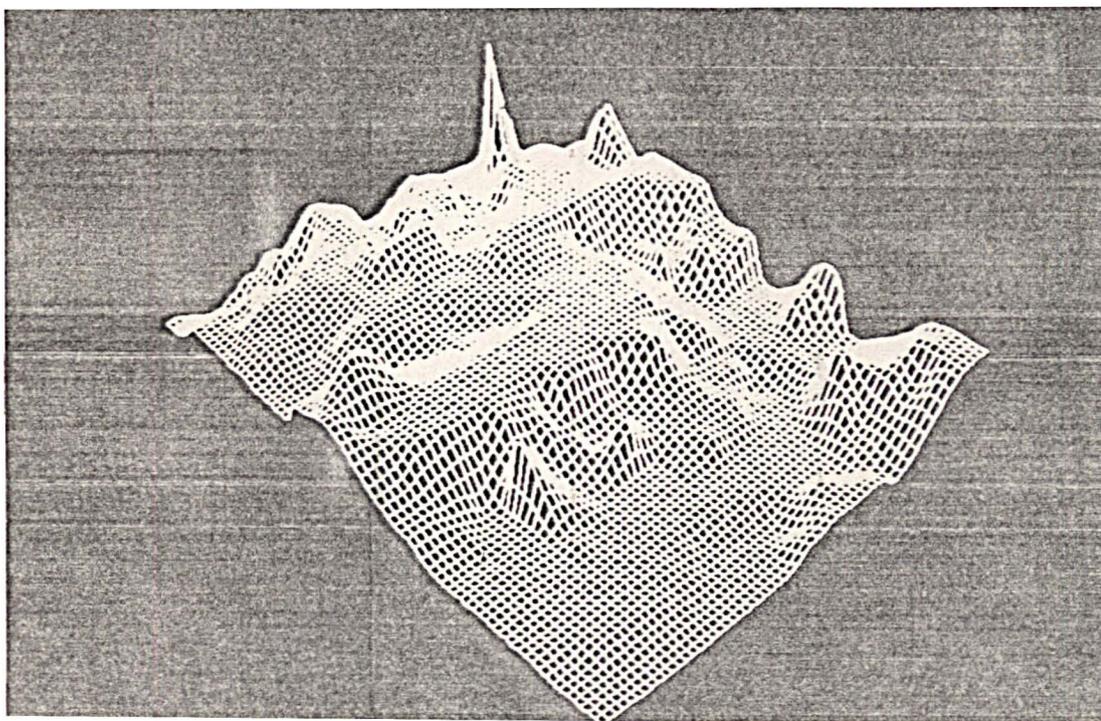
ESC	FIM
^	VOLTA
1	poligonos
2	MNT

Projeto. . : est
Nome do plano . . : mnt1

Parametros do Novo PI
Categoria: 2

Escala do mapa: 30000

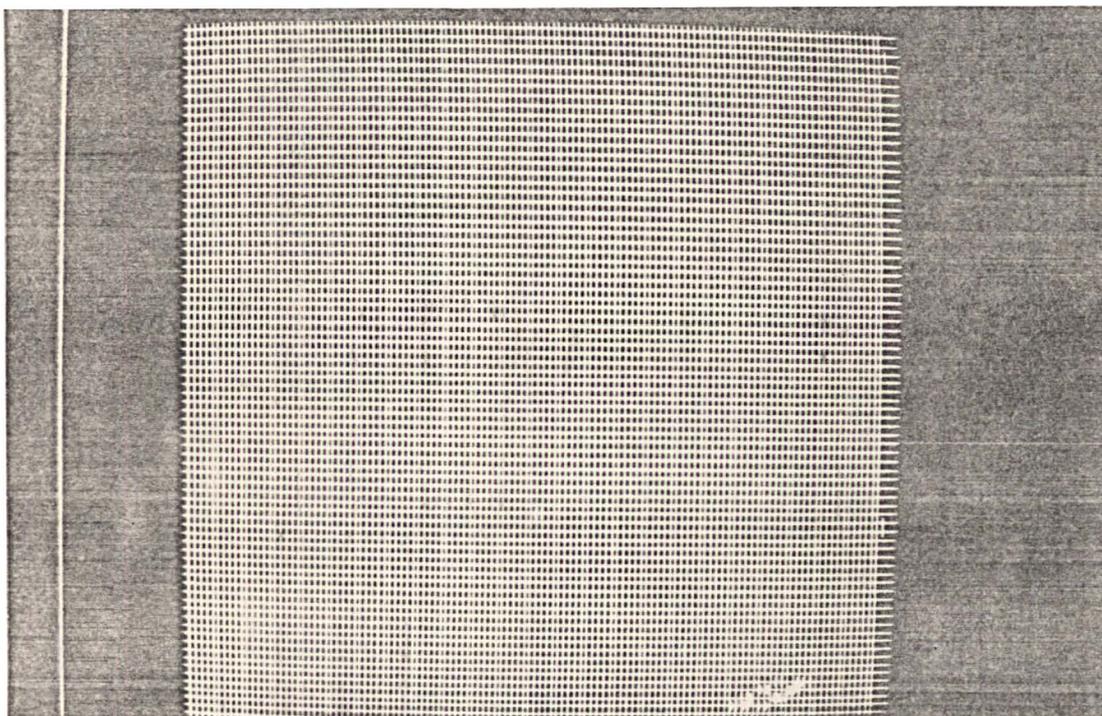
Figura 56: Imagem tridimensional gerada a partir das curvas de nível mestras



8.4 ENTRADA NO SGI DE PONTOS ISOLADOS PARA MNT

Como os dados, neste caso, correspondem a uma outra região geográfica, um novo projeto e plano de informação foi criado. A denominação do projeto foi pts e do PI mnt, este último, naturalmente da categoria MNT. Após estes criados, e as informações necessárias à execução da função armazenadas, o arquivo de dados foi definido, e conseqüentemente lido pela interface. O primeiro resultado desta etapa, ou seja, a imagem dos pontos armazenados na base de dados, é apresentada na figura 57.

Figura 57: Armazenamento de pontos obtidos por perfilamento de um modelo fotogramétrico no PLANICOMP



As etapas seguintes, incluindo as edições necessárias, foram semelhantes às descritas no item 8.3, com inclusão da geração e visualização das isolinhas a partir dos pontos armazenados. Estes resultados são mostrados nas figuras abaixo.

Figura 58: Isolinhas e imagem sintética geradas a partir da leitura de pontos de perfil

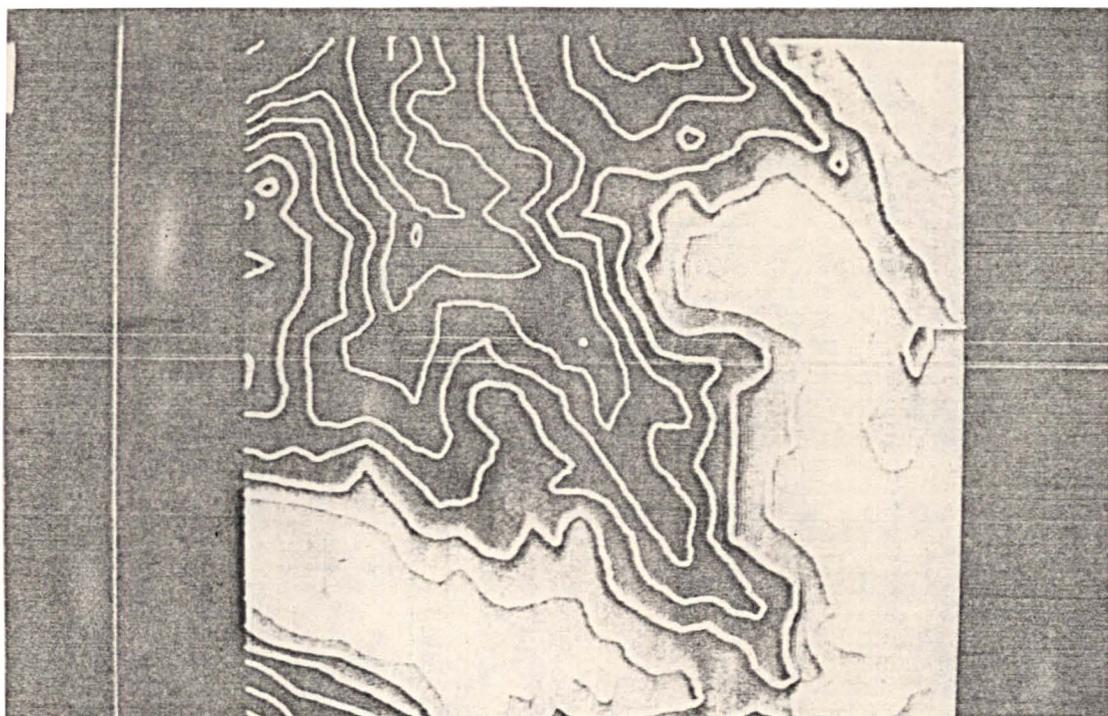
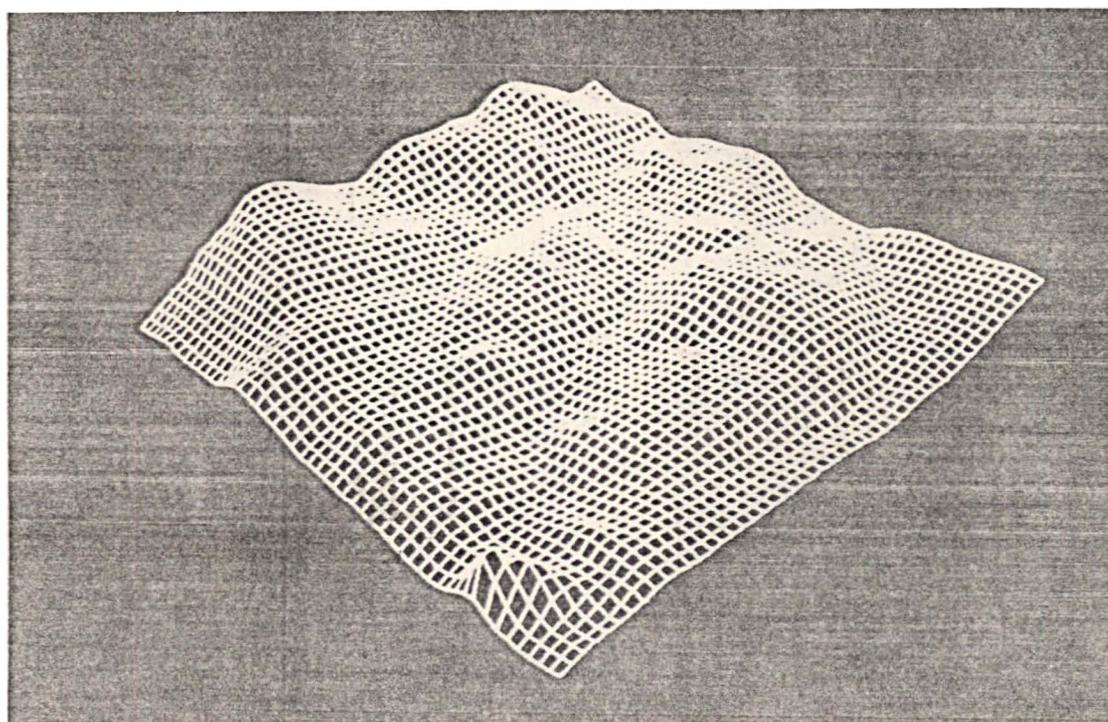


Figura 59: Imagem tridimensional do terreno obtida a partir das curvas de nível da figura 58.



9 CONCLUSÕES E SUGESTÕES PARA ESTUDOS FUTUROS

Das considerações sobre as soluções possíveis ao problema proposto para este trabalho, concluímos que a solução em batch inserida no SGI apresentou as melhores justificativas, pois:

- a) quando a maioria das instituições ou empresas usuárias de SIG's necessitam de produtos fotogramétricos em seus projetos, a coleta de dados fotogramétricos e a introdução destes nos sistemas são procedimentos realizados em locais e épocas distintas;
- b) a metodologia de trabalho do operador fotogrametrista no PLANICOMP não é alterada, ou seja, não exige qualquer adaptação e;
- c) para usuários conhecedores do SGI, o aprendizado sobre as poucas informações adicionais necessárias à entrada de dados fotogramétricos está baseado em conceitos cartográficos exigidos nos demais aplicativos do sistema.

Pela análise das potencialidades dos dois sistemas, percebemos que sendo o PLANICOMP um sistema fotogramétrico apoiado computacionalmente, seus recursos extrapolam a restituição de produtos fotogramétricos para mapeamento, os quais permitem também o registro de pontos do terreno para a geração de modelos numéricos do terreno. De acordo com as opções de tipos e categorias de dados possíveis de serem armazenados na base de dados do SGI, os dados obtidos com o PLANICOMP compatibilizam-se tanto na categoria de dados poligonais, como para MNT. Conseqüentemente, duas interfaces foram desenvolvidas, uma para cada categoria de dados, ou seja, entrada de dados de mapeamento e entrada de isolinhas para MNT; e uma terceira já existente no sistema, utilizada para entrada de registro de pontos isolados para MNT.

Dos testes realizados para verificar o comportamento da interface desenvolvida para entrada de dados de mapeamento, os quais englobaram todas as situações, que julgamos possíveis, de ocorrer nos arquivos ASCII de restituições, concluímos que:

- a) as possíveis inconsistências nos registros de pontos de feições digitalizadas pelo modo VECTOR, quando testadas nos exemplos de restituição;
- b) as cinco situações previstas para a determinação das tangentes nos pontos extremos das linhas digitalizadas nos modos TRACING e SPLINE;
- c) a determinação dos pontos interpolados e extrapolados nos modos SPLINE e TRACING;
- d) a determinação dos pontos pertencentes aos arcos de círculo ou círculo, a partir de três pontos digitalizados tanto no sentido horário como anti-horário;

comportaram-se como esperado, ou seja, as feições restituídas foram devidamente armazenadas na base de dados do SGI, sem inclusão ou omissão de pontos, ou ainda deformação das feições originais.

Quanto a interface de entrada de isolinhas para MNT, por armazenar pontos digitalizados com os modos TRACING e SPLINE sem determinar pontos interpolados entre estes, e por ter sido desenvolvida posteriormente à interface para dados poligonais, o tempo dispendido em seu desenvolvimento foi menor, pois alcançamos os resultados esperados mais facilmente.

Pelos resultados apresentados podemos afirmar que os objetivos almejados com o trabalho foram alcançados, observando ainda que por ser o SGI um software para sistemas de informações geográficas totalmente nacional, apoiado em tecnologia de baixo custo, tanto projetos realizados com suas atuais potencialidades, como os intencionados a ampliá-las, devem ser incentivados. Diante disso, propomos como estudos futuros:

- a) capacitar a base de dados do SGI para o armazenamento de feições pontuais;
- b) desenvolver interface para ler e armazenar os arquivos ASCII de textos das restituições;

- c) ampliar a capacidade de edição do SGI, tanto para o armazenamento de dados poligonais, como para isolinhas para MNT;
- d) implementar, através da mesma metodologia desenvolvida neste trabalho, interfaces para restituições realizadas com o estereorestituidor analítico da Wild;
- e) estudar a possibilidade de entrada de dados, no SGI, adquiridos com a câmara CCD;
- f) desenvolver e implementar metodologias para atualização cartográfica através da Fotogrametria analítica, utilizando-as para atualizar a base de dados do SGI.

ANEXOS

ANEXO 1 - Programação em C do algoritmo para leitura dos dados de restituição e armazenamento destes na base de dados do SGI	107
ANEXO 2 - Funções em C: pega_registro, ARC_CIRC, CIRCULO, ARCO, BETA_ARC, SPLI_DZ7	113
ANEXO 3 - Programação em C do algoritmo para leitura e armazenamento de isolinhas para MNT	121
ANEXO 4 - Arquivo ASCII do exemplo de restituição contendo todos os modos de digitalização	125

ANEXO 1

PROGRAMAÇÃO EM C DO ALGORITMO PARA LEITURA DOS DADOS DE RESTITUIÇÃO E
ARMAZENAMENTO DESTES NA BASE DE DADOS DO SGI

```

/* Programa para transformar formato de dados PLANICOMP -> SGI */
/* PLANISPL.C */

#include <stdio.h>
#include <math.h>
#include <float.h>
#include <sgi21.h>
#include <dia.h>

#define TRUE 1
#define FALSE 0
#define TAM_LIN_MAX 1000
#define FIM_ARQ -2
#define INVALIDO -3
#define PENDURAR xsave=x; ysave=y; ea=cod
#define s0(v1,v2,v3) 3.0*v1 - 3.0*v2 + v3

/*-----*/
/* NOME: planisgi */
/* AUTOR: Claudia Robbi */
/* RESUMO: Leitura e reformatacao de arquivo ASCII, gerado por res */
/* tituicao analitica no PLANICOMP serie C120/130, para */
/* armazenar na base de dados do SGI */
/* ENTRADA: arquivo ASCII de linhas gerado no PLANICOMP */
/* SAIDA: armazenamento na base de dados do SGI */
/* OBS: 1)o arquivo de entrada deve estar no formato RFORM=163511*/
/* do PLANICOMP */
/* 2)este arquivo deve ser finalizado com EOF */
/*-----*/
struct CIRC {
    double x0,
           y0,
           r2,
           r;
};

typedef struct CIRC CIRCO;
CIRCO PC;

struct VEC {
    double x[TAM_LIN_MAX],
           y[TAM_LIN_MAX];
    int npt;
};

typedef struct VEC VETOR;
VETOR V;

void ARCO(double[], double[]);
void ARC_CIRC(double[], double[], float, int);
void CIRCULO(double[], double[], float, int);
double pot(double, int);
double sqrt(double);
double atan(double);
double sin(double);
double cos(double);
double BETA_ARC(double, double, double, double);
void pega_registro( FILE *, double *, double *, int * );
SPLI_DZ7(VETOR *, int );

```

```
FILE *p1;
```

```
main()
```

```
{
```

```
    struct desc_linhas    deslin;  
    struct arq_linhas     stlin;
```

```
    char    nome_pi[COMPR_NOME_ARQ],    /*nome do plano de informacao*/  
           nome_regiao[COMPR_NOME_ARQ], /*nome da regioa*/  
           nome_arq[COMPR_NOME_ARQ],    /*nome do arquivo formado pelo*/  
                                           /*nome do plano e nome da regioa*/
```

```
           nome_asc[20],  
           c[3],  
           help[200],  
           ch;
```

```
    int    ea=0,          /* estado anterior do elemento grafico lido*/  
          cod,          /* codigo do elemento grafico */  
          fator_ampli = 8,  
          unidade,  
          i, np, cat, ndalin,  
          arco_sgi = FALSE,  
          spline;
```

```
    float  vx[TAM_LIN_MAX],  
          vy[TAM_LIN_MAX],  
          escala,  
          xmin,xmax,ymin,ymax;
```

```
    double xa[3],ya[3],    /*vetor para gerar o arco de circulo*/  
           xc[5],yc[5],    /*vetor para gerar o circulo*/  
           x,y,            /*coordenadas lidas no PLANICOMP*/  
           xsave, ysave,  
           xd,yd,offx,offy;
```

```
    memset(&deslin,0,sizeof(deslin));  
    memset(&stlin,0,sizeof(stlin));
```

```
    dia_titulo("Leitura do arquivo de restituicao do PLANICOMP em ASCII");
```

```
/* obtem nome da regioa no arquivo contexto */
```

```
    if(sgi_le_contexto(nome_regiao) != ERRO_NORMAL)
```

```
    {
```

```
        dia_erro("Defina projeto ativo");  
        goto fim_prog;
```

```
    }
```

```
        .  
        .  
        .
```

```
    }
```

```
spline = 0;
```

```
V.npt = 0;
```

```
while( 1 ) {
```

```
    pega_registro( p1, &x, &y, &cod );
```

```

if( cod == INVALIDO )
    continue;

switch( ea ) {
    case 0: /* inicio */
        break;

    case 213: /* inicio do vetor */
        switch( cod ) {
            case 212: V.x[V.npt] = xsave;
                    V.y[V.npt] = ysave;
                    V.npt += 1;
                    if(V.npt > 1000)
                        arco_sgi = TRUE;
                    break;
            case FIM_ARQ: goto fim_prog;
        }
        break;

    case 212: /* sequencia de pontos do vetor */
        V.x[V.npt] = xsave; V.y[V.npt] = ysave; V.npt += 1;
        switch( cod ){
            case 213:
            case 230:
            case 241:
            case 341:
            case FIM_ARQ: arco_sgi = TRUE;
                        break;
        }
        break;

    case 230: /* sequencia de pontos da spline */
        switch( cod ){
            case 230:
            case 231: if(spline == 2 || spline == 3)
                    spline = 0;
                    V.x[V.npt] = xsave;
                    V.y[V.npt] = ysave;
                    if(V.npt != 0){
                        if(V.x[V.npt]==V.x[V.npt-1]&&V.y[V.npt]==V.y[V.npt-1]){
                            V.npt = V.npt - 1;
                        }
                    }
                    V.npt += 1;
                    if(V.npt > 180){
                        SPLI_DZ7(&V, spline);
                        if(spline == 0) spline = 1;
                        arco_sgi = TRUE;
                    }
                    break;
        }
        break;

    case 231: /* ultimo ponto da spline */
        V.x[V.npt] = xsave; V.y[V.npt] = ysave;
        if(V.npt != 0){
            if(V.x[V.npt]==V.x[V.npt-1]&&V.y[V.npt]==V.y[V.npt-1]){
                V.npt = V.npt - 1;
            }
        }
}

```

```

    }
    V.npt += 1;
    if(V.npt == 1 && spline == 0){
        V.npt = 0;
        break;
    }
    if(V.npt == 2 && spline == 0){
        arco_sgi = TRUE;
        break;
    }
    if(spline == 0) spline = 3;
    else if(spline == 1) spline = 2;
    SPLI_DZ7(&V, spline);
    arco_sgi = TRUE;
    break;
}
case 241: /* primeiro ponto da arco de circulo */
    switch( cod ){
        case 242: xa[0] = xsave; ya[0] = ysave;
                break;
    }
    break;

case 242: /* segundo ponto do arco de circulo */
    switch( cod ){
        case 243: xa[1] = xsave;
                ya[1] = ysave;
                break;
    }
    break;

case 243: /* terceiro ponto do arco de circulo */
    xa[2] = xsave;
    ya[2] = ysave;
    ARC_CIRC(xa,ya,escala,fator_ampli);
    arco_sgi = TRUE;
    break;

case 341: /* primeiro ponto do circulo */
    switch( cod ){
        case 342: xc[0] = xsave; yc[0] = ysave;
                break;
    }
    break;

case 342: /* segundo ponto do circulo */
    switch( cod ){
        case 343: xc[1] = xsave;
                yc[1] = ysave;
                break;
    }
    break;

case 343: /* terceiro ponto do circulo */
    xc[2] = xsave;
    yc[2] = ysave;
    CIRCULO(xc, yc, escala, fator_ampli);
    arco_sgi = TRUE;
    break;

```

```
}
```

```
PENDURAR;
```

```
if(arco_sgi == TRUE)
```

```
{  
  .  
  .  
  . }  
}
```

```
PENDURAR;
```

```
if(arco_sgi == TRUE)
```

```
{  
  .  
  .  
  . }  
}
```

```
PENDURAR;
```

```
if(arco_sgi == TRUE)
```

```
{  
  .  
  .  
  . }  
}
```

ANEXO 2

FUNÇÕES EM C: PEGA_REGISTRO, ARC_CIRC, CIRCULO, ARCO, BETA_ARC, SPLI_DZ7

```

        void pega_registro( FILE *p1, double *x, double *y, int *cod )
    {
        float  flixo;
        int    lixo;
        static int nreg = 0;

        if( fscanf( p1, "%d %lf %lf %f %d", &lixo, x, y, &flixo, cod ) == EOF )
            *cod = FIM_ARQ;

        switch( *cod ) {
            case 213: case 212: case 230: case 231: case 341:
            case 342: case 343: case 241: case 242: case 243:
                break;
            case FIM_ARQ:
                break;
            default:
                *cod = INVALIDO;
                break;
        }

        return;
    }
}

```

```

/* Funcao que gera pontos de um arco de circulo a partir de 3 pontos deste*/
void ARC_CIRC(double xa[], double ya[], float escala, int fator_ampli)

```

```

{
    int horario=1,
        i=0;

    double alfa, delta, dist,
           pi,beta;

    pi = 4. * atan(1.);

    xa[1] = xa[1] - xa[0];
    ya[1] = ya[1] - ya[0];
    xa[2] = xa[2] - xa[0];
    ya[2] = ya[2] - ya[0];
    ARCO(xa,ya);

    xa[1] = xa[1] + xa[0];
    ya[1] = ya[1] + ya[0];
    xa[2] = xa[2] + xa[0];
    ya[2] = ya[2] + ya[0];

    beta = BETA_ARC(PC.x0,xa[0],PC.y0,ya[0]);

    if(xa[0] >= PC.x0)
    {
        if(((ya[0]>=PC.y0)&&(xa[0]>xa[1])) || ((ya[0]<=PC.y0)&&(ya[0]<ya[1])))
            horario = 0;
    }
    else if(xa[0] <= PC.x0)
    {
        if(((ya[0]>=PC.y0)&&(ya[0]>ya[1])) || ((ya[0]<=PC.y0)&&(xa[0]<xa[1])))
            horario = 0;
    }

    alfa = (0.0002 * escala)/(PC.r * fator_ampli);
}

```

```

delta = 2. * PC.r * sin(alfa/2.);
V.x[0] = xa[0];
V.y[0] = ya[0];
V.npt = 0;
do
{
    dist = sqrt((pot(xa[i]-xa[i+1],2))+(pot(ya[i]-ya[i+1],2)));
    if(dist > delta)
    {
        do
        {
            V.npt += 1;
            if(horario == 1)
            {
                beta = beta - alfa;
                if(beta < 0.)
                    beta = 2. * pi + beta;
            }
            else if(horario == 0)
                beta = beta + alfa;
            V.x[V.npt] = PC.x0 + (PC.r * cos(beta));
            V.y[V.npt] = PC.y0 + (PC.r * sin(beta));
            dist = sqrt((pot(V.x[V.npt]-xa[i+1],2))+(pot(V.y[V.npt]-ya[i+1],2)));
        }
        while (dist > delta);
    }
    i += 1;
    V.npt += 1;
    V.x[V.npt] = xa[i];
    V.y[V.npt] = ya[i];
}
while(i < 2);
V.npt += 1;

return;

```

}

/* Funcao que gera pontos de um circulo a partir de 3 pontos deste*/
void CIRCULO(double xc[], double yc[], float escala, int fator_ampli)

{

```

int horario=1,
    i=0;

```

```

double alfa, delta, dist,
        pi, beta;

```

```

pi = 4. * atan(1.);

```

```

xc[1] = xc[1] - xc[0];
yc[1] = yc[1] - yc[0];
xc[2] = xc[2] - xc[0];
yc[2] = yc[2] - yc[0];
ARCO(xc,yc);

```

```

xc[0] = PC.x0 - PC.r;
xc[2] = PC.x0 + PC.r;
yc[0] = PC.y0;
yc[2] = PC.y0;

```

```

xc[1] = PC.x0;
xc[3] = PC.x0;
yc[1] = PC.y0 + PC.r;
yc[3] = PC.y0 - PC.r;
xc[4] = xc[0];
yc[4] = yc[0];

beta = pi;
alfa = (0.0002 * escala)/(PC.r * fator_ampli);
delta = 2. * PC.r * sin(alfa/2.);
V.x[0] = xc[0];
V.y[0] = yc[0];
V.npt = 0;
do
{
    do
    {
        V.npt += 1;
        beta = beta - alfa;
        if(beta < 0.)
            beta = 2. * pi + beta;
        V.x[V.npt] = PC.x0 + (PC.r * cos(beta));
        V.y[V.npt] = PC.y0 + (PC.r * sin(beta));
        dist = sqrt((pot(V.x[V.npt]-xc[i+1],2)+(pot(V.y[V.npt]-yc[i+1],2)));
    }
    while (dist > delta);
    i += 1;
    V.npt += 1;
    V.x[V.npt] = xc[i];
    V.y[V.npt] = yc[i];
}
while(i < 4);
V.npt += 1;

return;
}

```

/*Funcao para calcular o raio e o centro de um arco dados 3 pts.*/

void ARCO(double x[],double y[])

```

{
    double c1, c2;

    c1 = (-pot(x[1],2)-pot(y[1],2))/2.;
    c2 = (-pot(x[2],2)-pot(y[2],2))/2.;

    PC.x0 = (y[1]*c2-y[2]*c1)/(y[2]*x[1]-y[1]*x[2]);
    PC.y0 = (c2*x[1]-c1*x[2])/(y[1]*x[2]-y[2]*x[1]);
    PC.r2 = pot(x[1]-PC.x0,2)+pot(y[1]-PC.y0,2);
    PC.r2 = PC.r2+pot(x[2]-PC.x0,2)+pot(y[2]-PC.y0,2);
    PC.r2 = PC.r2/2.;
    PC.r = sqrt(PC.r2);

    PC.x0 = PC.x0 + x[0];
    PC.y0 = PC.y0 + y[0];

    return;
}

```

```
/*Funcao para calcular potenciacao*/
```

```
double pot(double x, int n)
```

```
{  
    int i;  
  
    double t;  
  
    i = 1;  
    t = x;  
    do  
    {  
        t = t * x;  
        i += 1;  
    }  
    while(i < n);  
  
    return(t);  
}
```

```
/*Funcao que calcula beta para arco e circulo*/
```

```
double BETA_ARC(double x1, double x2, double y1, double y2)
```

```
{  
  
    double pi,  
           beta;  
  
    double num, den;  
  
    pi = 4. * atan(1.);  
    num = y2 - y1;  
    den = x2 - x1;  
  
    if(num > 0)  
    {  
        if(den > 0)  
            beta = atan(num/den);  
        else if(den < 0)  
            beta = pi + atan(num/den);  
        else if(den == 0)  
            beta = pi/2.;  
    }  
    else if(num < 0)  
    {  
        if(den < 0)  
            beta = pi + atan(num/den);  
        else if(den > 0)  
            beta = 2.*pi + atan(num/den);  
        else if(den == 0)  
            beta = 3.*pi/2.;  
    }  
    else if(num == 0)  
    {  
        if(den < 0)  
            beta = pi;  
        if(den > 0)  
            beta = 0.;  
    }  
  
    return(beta);  
}
```

}

SPLI_DZ7(VETOR *V, int spline)

{

int n_seg, npt,
i, j, k,
l;

double a[5], b[5],
w[3],
sen_teta[2], cos_teta[2],
A, B, r, aux,
p0, p1, p2, p3,
q0, q1, q2, q3,
sx[TAM_LIN_MAX], sy[TAM_LIN_MAX],
u, du,
ult_pt_x, ult_pt_y;

float intervalo;

if (spline == 2 || spline == 3)

{

ult_pt_x = V->x[V->npt - 1];
ult_pt_y = V->y[V->npt - 1];

}

else if (spline == 0 || spline == 1)

{

ult_pt_x = V->x[V->npt - 3];
ult_pt_y = V->y[V->npt - 3];

}

if (spline == 0)

{

npt = V->npt - 1;

}

else if (spline == 1)

{

npt = V->npt + 2;

}

else if (spline == 2)

{

npt = V->npt + 4;

}

else if (spline == 3)

{

npt = V->npt + 1;

}

if (spline == 0 || spline == 3)

{

for (i=0; i<V->npt; i++)

{

sx[i+2] = V->x[i];
sy[i+2] = V->y[i];

}

}

else if (spline == 1 || spline == 2)

```

(
  for(i=0; i<V->npt; i++)
  {
    sx[i+5] = V->x[i];
    sy[i+5] = V->y[i];
  }
)

if(V->x[0] == V->x[V->npt-1] && V->y[0] == V->y[V->npt-1]) /*linha fechada*/
(
  for(i=1,j=0; i>=0; i--,j++)
  {
    sx[i] = sx[V->npt-j];
    sy[i] = sy[V->npt-j];
  }
  for (i=2; i<=3; i++)
  {
    sx[V->npt+i] = sx[i+1];
    sy[V->npt+i] = sy[i+1];
  }
)

if(V->x[0] != V->x[V->npt-1] && V->y[0] != V->y[V->npt-1]) /*linha aberta*/
(
  if(spline == 0 || spline == 3)
  {
    for(i=1; i>=0; i--)
    {
      sx[i] = s0(sx[i+1], sx[i+2], sx[i+3]);
      sy[i] = s0(sy[i+1], sy[i+2], sy[i+3]);
    }
  }
  if(spline == 2)
  {
    for(i=5; i<=6; i++)
    {
      sx[V->npt+i] = s0(sx[V->npt+i-1], sx[V->npt+i-2], sx[V->npt+i-3]);
      sy[V->npt+i] = s0(sy[V->npt+i-1], sy[V->npt+i-2], sy[V->npt+i-3]);
    }
  }
  if(spline == 3)
  {
    for(i=2; i<=3; i++)
    {
      sx[V->npt+i] = s0(sx[V->npt+i-1], sx[V->npt+i-2], sx[V->npt+i-3]);
      sy[V->npt+i] = s0(sy[V->npt+i-1], sy[V->npt+i-2], sy[V->npt+i-3]);
    }
  }
)

l = 0;
for(i=2; i<npt; i++)
(
  k = i - 2;
  for(j=0; j<=4; j++,k++)
  {
    a[j] = sx[k+1] - sx[k];
    b[j] = sy[k+1] - sy[k];
  }
)

```

```

k = 0;
for(j=2; j<=3; j++,k++)
{
    w[k] = fabs(a[j]*b[j+1] - a[j+1]*b[j]);
    w[k+1] = fabs(a[j-2]*b[j-1] - a[j-1]*b[j-2]);

    A = w[k]*a[j-1] + w[k+1]*a[j];
    B = w[k]*b[j-1] + w[k+1]*b[j];

    aux = sqrt((A*A) + (B*B));
    if(aux == 0.) aux = 1.;
    sen_teta[k] = B / aux;
    cos_teta[k] = A / aux;
}

r = sqrt((a[2]*a[2]) + (b[2]*b[2]));

p0 = sx[i];
p1 = r * cos_teta[0];
p2 = (3.0*(a[2]))-(r*(cos_teta[1]+2.0*cos_teta[0]));
p3 = (-2.0*(a[2]))+(r*(cos_teta[1]+cos_teta[0]));

q0 = sy[i];
q1 = r * sen_teta[0];
q2 = (3.0*(b[2]))-(r*(sen_teta[1]+2.0*sen_teta[0]));
q3 = (-2.0*(b[2]))+(r*(sen_teta[1]+sen_teta[0]));

n_seg = 5;

du = 1.0/(float)n_seg;

for(u=0.0; u<1.0; u+=du,l++)
{
    V->x[l] = (((((double)p3 * u + (double)p2) * u + (double)p1) * u + (double)p0);
    V->y[l] = (((((double)q3 * u + (double)q2) * u + (double)q1) * u + (double)q0);
}
}

V->npt = l;

if(spline == 0 || spline == 1)
{
    for(i=0; i<=4; i++)
    {
        sx[i] = sx[npt - 2 + i];
        sy[i] = sy[npt - 2 + i];
    }
}

V->x[V->npt] = ult_pt_x;
V->y[V->npt] = ult_pt_y;
V->npt = V->npt + 1;

return;

```

ANEXO 3

PROGRAMAÇÃO EM C DO ALGORITMO PARA LEITURA
E ARMAZENAMENTO DE ISOLINHAS PARA MNT

```

/* Programa para transformar formato de dados PLANICOMP -> SGI */
/*      para isolinhas      */

#include <stdio.h>
#include <dia.h>
#include <math.h>
#include <sgi21.h>
#include <float.h>

#define TRUE 1
#define FALSE 0
#define TAM_LIN_MAX 1000
#define FIM_ARQ -2
#define INVALIDO -3
#define PENDURAR xsave=x; ysave=y; zsave=z; ea=cod
/*-----*/
/*      NOME:  planisgi      */
/*      AUTOE: Claudia Robbi */
/*      RESUMO: Leitura e reformatacao de arquivo ASCII, gerado por*/
/*              restituicao analitica no PLANICOMP serie C120/130, */
/*              para arquivo de isolinhas no formato MNT do SGI  */
/*      ENTRADA: arquivo ASCII de linhas gerado no PLANICOMP      */
/*      SAIDA:  arquivo de isolinhas para MNT do SGI              */
/*      FUNCOES :sgiger4.o      */
/*      OBS:   o arquivo de entrada deve estar no formato        */
/*              RFORM=163511 do PLANICOMP                        */
/*              este arquivo deve ser finalizado com EOF         */
/*              Projeto e plano de informacao ja devem existir   */
/*-----*/

struct VEC {
    double x[TAM_LIN_MAX],
           y[TAM_LIN_MAX],
           z[TAM_LIN_MAX];
    int npt;
};

typedef struct VEC VETOR;
VETOR V;

void pega_registro( FILE *, double *, double *, double *, int * );

FILE *p1;

main()
{
    struct desc_linhas  deslin;
    struct arq_linhas   stlin;
    struct desc_poligonos despol;
    struct arq_poligonos stpol;

    char nome_pi[COMPR_NOME_ARQ], /* nome do plano de informacao */
          nome_regiao[COMPR_NOME_ARQ], /* nome da regioao */
          nome_arq[COMPR_NOME_ARQ], /* nome do arquivo formado pelo*/
                                           /* nome do plano e nome da regioao*/

          nome_asc[20],
          c[3],
          help[200];

```

```

int    ea=0,          /* estado anterior do elemento grafico lido*/
      cod,           /* codigo do elemento grafico */
      i, cat, ndalin, ik, vaux[3],
      arco_sgi = FALSE;

float  vx[TAM_LIN_MAX],
      vy[TAM_LIN_MAX],
      xmin,xmax,ymin,ymax,
      cotaz;

double x,y,z,        /*coordenadas lidas no PLANICOMP*/
      xsave, ysave,zsave,
      xd,yd,offx,offy;

memset(&deslin,0,sizeof(deslin));
memset(&stlin,0,sizeof(stlin));

dia_titulo("Leitura do arquivo de isolinhas do PLANICOMP em ASCII");

/* obtem nome da regio no arquivo contexto */
if(sgi_le_contexto(nome_regiao) != ERRO_NORMAL)
{
    dia_erro("Defina projeto ativo");
    goto fim_prog;
}
.
.
.

while( 1 ) {

    pega_registro( p1, &x, &y, &z, &cod );

    if( cod == INVALIDO )
        continue;

    switch( ea ) {
        case 0: /* inicio */
            break;

        case 230: /* sequencia de pontos da spline */
            switch( cod ){
                case 230:
                case 231: V.x[V.npt] = xsave;
                        V.y[V.npt] = ysave;
                        V.z[V.npt] = zsave;
                        if(V.npt != 0)
                        {
                            if((V.x[V.npt]==V.x[V.npt-1])&&(V.y[V.npt]==V.y[V.npt-1]))
                            {
                                V.npt = V.npt -1;
                            }
                        }
                        V.npt += 1;
                        if(V.npt > 1000)
                        {
                            for(i=0; i<V.npt; i++)
                                cotaz = cotaz + V.z[i];
                            cotaz = cotaz/((float)(V.npt - 1));
                        }
            }
    }
}

```

```

        arco_sgi = TRUE;
    }
    break;
}
break;

case 231: /* ultimo ponto da spline */
    V.x[V.npt] = xsave;
    V.y[V.npt] = ysave;
    V.z[V.npt] = zsave;
    if(V.npt != 0)
    {
        if((V.x[V.npt]==V.x[V.npt-1])&&(V.y[V.npt]==V.y[V.npt-1]))
        {
            V.npt = V.npt - 1;
        }
    }
    V.npt += 1;
    if(V.npt == 1){
        V.npt = 0;
        break;
    }
    for(i=0; i<V.npt; i++){
        cotaz = cotaz + (float)V.z[i];
    }
    cotaz = cotaz/(float)(V.npt - 1);
    arco_sgi = TRUE;
    break;
}

PENDURAR;

if(arco_sgi == TRUE)
{
    for(i=0; i<V.npt; i++)
    {
        vx[i] = (float)(V.x[i]-offx);
        vy[i] = (float)(V.y[i]-offy);
    }
    .
    .
    .
}

```

ANEXO 4

**ARQUIVO ASCII DO EXEMPLO DE RESTITUIÇÃO CONTENDO
TODOS OS MODOS DE DIGITALIZAÇÃO**

-1	669951.492	7179581.158	919.760	212
-1	669951.494	7179573.607	919.760	212
-1	669947.953	7179573.607	919.448	212
-1	669904.611	7179652.209	914.094	213
-1	669904.611	7179652.209	914.094	241
-1	669900.691	7179651.361	914.118	242
-1	669899.344	7179648.033	914.142	243
-1	669898.666	7179566.967	917.380	241
-1	669900.021	7179564.010	917.380	242
-1	669902.984	7179563.133	917.427	243
-1	669980.168	7179561.787	915.822	241
-1	669983.238	7179562.816	915.762	242
-1	669984.494	7179566.422	915.679	243
-1	669983.246	7179648.205	914.085	241
-1	669982.221	7179650.328	914.025	242
-1	669979.889	7179651.391	914.013	243
-1	669979.889	7179651.391	914.025	213
-1	669904.611	7179652.209	914.087	212
-1	669899.344	7179648.033	914.135	213
-1	669899.463	7179624.910	915.044	212
-1	669899.455	7179615.891	915.427	213
-1	669898.666	7179566.967	917.376	212
-1	669902.984	7179563.133	917.424	213
-1	669980.168	7179561.787	915.819	212
-1	669984.494	7179566.422	915.676	213
-1	669983.246	7179648.205	914.079	212
-1	669899.463	7179624.910	915.039	230
-1	669899.463	7179624.910	915.039	230
-1	669899.514	7179624.799	915.051	230
-1	669899.719	7179624.207	915.111	230
-1	669899.820	7179623.621	915.195	230
-1	669900.023	7179623.191	915.219	230
-1	669900.584	7179622.281	915.338	230
-1	669901.080	7179622.008	915.374	230
-1	669901.691	7179621.742	915.410	230
-1	669903.145	7179621.506	915.458	230
-1	669904.029	7179621.602	915.553	230
-1	669904.656	7179621.779	915.565	230
-1	669905.342	7179622.215	915.565	230
-1	669906.270	7179622.967	915.625	230
-1	669906.693	7179623.428	915.613	230
-1	669907.100	7179623.834	915.577	231
-1	669899.455	7179615.891	915.422	230
-1	669899.455	7179615.891	915.422	230
-1	669899.451	7179615.916	915.422	230
-1	669899.529	7179616.221	915.422	230
-1	669899.732	7179616.670	915.422	230
-1	669900.061	7179617.135	915.422	230
-1	669900.455	7179617.512	915.422	230
-1	669900.998	7179617.838	915.470	230
-1	669902.262	7179618.506	915.590	230
-1	669902.965	7179618.770	915.602	230
-1	669903.848	7179619.043	915.602	230
-1	669904.361	7179619.088	915.602	230
-1	669905.080	7179619.037	915.602	230
-1	669905.553	7179618.822	915.602	230
-1	669905.867	7179618.312	915.566	230
-1	669906.129	7179617.723	915.554	230
-1	669906.248	7179617.457	915.554	231
-1	669906.270	7179622.967	915.608	230
-1	669906.668	7179623.488	915.608	230
-1	669907.014	7179624.105	915.608	230
-1	669907.338	7179624.934	915.608	230

-1	669907.701	7179625.928	915.608	230
-1	669908.018	7179626.693	915.608	230
-1	669908.354	7179627.238	915.608	230
-1	669908.613	7179627.553	915.608	230
-1	669908.973	7179627.844	915.608	230
-1	669909.490	7179628.164	915.608	230
-1	669910.457	7179628.281	915.608	230
-1	669911.615	7179628.381	915.608	230
-1	669912.609	7179628.641	915.608	230
-1	669913.637	7179629.225	915.608	230
-1	669915.145	7179629.717	915.608	230
-1	669916.326	7179629.562	915.608	230
-1	669917.260	7179628.621	915.608	230
-1	669918.129	7179627.855	915.608	230
-1	669919.426	7179627.582	915.608	230
-1	669920.707	7179628.207	915.608	230
-1	669922.393	7179628.328	915.608	230
-1	669923.686	7179627.781	915.608	230
-1	669924.533	7179626.664	915.608	230
-1	669924.850	7179625.115	915.608	230
-1	669924.451	7179623.215	915.608	230
-1	669923.982	7179621.766	915.608	230
-1	669923.811	7179621.441	915.608	231
-1	669925.334	7179607.070	919.502	341
-1	669917.230	7179602.414	920.137	342
-1	669925.480	7179597.402	919.443	343
-1	669925.582	7179581.523	919.658	341
-1	669918.748	7179576.832	921.085	342
-1	669925.391	7179572.117	919.768	343
-1	669968.773	7179575.178	920.833	341
-1	669963.521	7179579.607	919.887	342
-1	669963.508	7179571.018	919.864	343
-1	669969.490	7179585.289	918.668	213
-1	669969.490	7179585.289	918.668	213
-1	669959.346	7179585.502	918.728	212
-1	669959.535	7179594.498	918.752	212
-1	669969.682	7179594.287	918.752	212
-1	669969.490	7179585.289	918.668	212
-1	669951.084	7179592.232	918.046	213
-1	669947.625	7179592.236	918.142	212
-1	669947.629	7179595.646	918.453	212
-1	669951.090	7179595.643	918.453	212
-1	669951.084	7179592.232	918.046	212
-1	669947.143	7179595.502	918.979	213
-1	669947.170	7179607.258	919.051	212
-1	669952.187	7179607.246	919.027	212
-1	669952.162	7179595.492	919.027	212
-1	669947.143	7179595.502	918.979	212
-1	669974.645	7179597.143	919.111	213
-1	669961.555	7179597.229	919.075	212
-1	669961.613	7179606.025	919.075	212
-1	669974.703	7179605.939	919.075	212
-1	669974.645	7179597.143	919.111	212
-1	669974.779	7179631.965	917.035	213
-1	669961.402	7179632.070	917.035	212
-1	669961.465	7179639.887	917.035	212
-1	669974.844	7179639.781	917.035	212
-1	669974.779	7179631.965	917.035	212
-1	669947.953	7179573.607	919.448	213
-1	669947.951	7179581.158	919.353	212

-1	669951.492	7179581.158	919.760	212
-1	669951.494	7179573.607	919.760	212
-1	669947.953	7179573.607	919.448	212

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 AKIMA, H. A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. Journal of the Association for Computing Machinery, 17(4):589-602. out.1970.
- 2 ALLAM, M. M. A Review of Data Acquisition Systems Present and Future and their Effect on Cartographic Information Systems. 14th Congress of the International Society for Photogrammetry Commission IV, Working group 1. Hamburgo, 1980. 21-35.
- 3 BONJOUR, J. D. & NEWBY, P. R. T. Architecture and Components of Photogrammetric Workstations. Photogrammetric Record. 13(75):389-405. abr.1990.
- 4 BURROUGH, P. A. Principles of Geographical Information Systems for Land Resources Assessment. 5. reimp. Oxford, Clarence Press, 1990. 194p. (Cap.1,2 e 4)
- 5 CHURCHMAN, C. W. Introdução à Teoria dos Sistemas. 2.ed. Petrópolis, Editora Vozes Ltda, 1972. 309p. (Cap.1)
- 6 DALE, P. F. Geographical Information Systems and Photogrammetry: Organisational Issues. Photogrammetric Record, 13(77):679-683. abr.1991.
- 7 DATE, C. J. Introdução a Sistemas de Banco de Dados. 4. ed. Rio de Janeiro, Editora Campus Ltda, 1986. 513p. (Cap. 1, 2 e 3)
- 8 EREZ, M. T. & DORRER, E. Photogrammetric Data Acquisition Using an Interactive Computer Graphics System. Photogrammetric Engineering and Remote Sensing. 50(2):183-188. feb.1984.
- 9 FURTADO, A. L. & SANTOS, C. S. Organização de Banco de Dados. 4 ed. Rio de Janeiro, Editora Campus Ltda, 1983. 281p. (Cap. 1, 2, 3, 4 e 5)
- 10 HOROWITZ, E. & SAHNI, S. Fundamentos de Estruturas de Dados. 4. reimp. Rio de Janeiro, Editora Campus Ltda, 1984. 494p. (Cap. 1, 2, 3, 4, 5 e 6)

- 11 INSTITUTO DE PESQUISAS ESPACIAIS. SIG - Manual do usuário. São José dos Campos, ano 1990.
- 12 INSTITUTO DE PESQUISAS ESPACIAIS. SIG - Manual de programação. São José dos Campos, ano 1990.
- 13 LEITE, L. L. P. Introdução aos Sistemas de Gerência de Banco de Dados. 1. reimp. São Paulo, Editora Edgard Blücher Ltda, 1980. 138p. (Cap. 1, 2 e 3)
- 14 MOLENAAR, M. Status and problems of geographical information systems. The necessity of a geoinformation theory. ISPRS Journal of Photogrammetry and Remote Sensing, 46(2):85-103, abr.1991.
- 15 YONG, C. S. Banco de Dados: organização, sistemas, administração. São Paulo, Editora Atlas S.A., 1983. 398p. (Cap. 1 e 3)
- 16 WALKER, A. S. Input of Photogrammetric Data to Geographical Information Systems. Photogrammetric Record, 12(70):459-471. oct.1987.
- 17 WARNER, W. S. et alii. GIS Data Capture from Aerial Photographs: a case study. Photogrammetric Record, 13(77):691-701. abr.1991.
- 18 WIRTH, N. Algoritmos e Estruturas de Dados. Rio de Janeiro, Editora Prentice-Hall do Brasil Ltda, 1989. 255p. (Cap. 1, 2, 3 e 4)
- 19 ZEISS WEST GERMANY. PLANICOMP C100/120/130 - Manual de manejo 51-1088s, Oberkochen
- 20 ZEISS WEST GERMANY. DZ7-AS - Reference Manual 51-1129, Oberkochen
- 21 ZEISS WEST GERMANY. GRAPH F1 subroutine library for programming graphical tasks 51-1116, Oberkochen
- 22 ZEISS WEST GERMANY. GEFIO Operating Instructions 51-1136e, Oberkochen