

## Conversão de Dados entre os Sistemas SGI e ARC/INFO

DIÓGENES S. ALVES<sup>1</sup>, EUGÊNIO SPER DE ALMEIDA<sup>2</sup>

Divisão de Processamento de Imagens (DPI)  
Instituto Nacional de Pesquisas Espaciais (INPE)  
C.P. 515, CEP 12201-970, São José dos Campos, SP  
Internet: dalves@dpi.inpe.br<sup>1</sup>, eugenio@dpi.inpe.br<sup>2</sup>

**Abstract.** A method for converting SGI *information planes* into ARC/INFO *coverages* is presented, including the conversion of geometric (lines and polygons) and attribute (SGI classes) data. The method is being used for generating ARC//INFO data from deforestation, vegetation and other data of the *Amazonia* Information System, a geo-referenced data base for Amazon ecosystems. A simplified version of C code used is presented.

### Introdução

O uso de sistemas de informações geográficas (SIG) incentiva e, de certa forma, até pressupõe o intercâmbio de dados digitais mantidos por diferentes instituições em diferentes sistemas.

Tal intercâmbio reduz o custo de criação dos acervos de dados digitais e favorece a padronização e a disseminação das bases de dados. Apesar de existirem padrões para intercâmbio de dados, é comum o desenvolvimento de soluções específicas para a conversão de dados entre sistemas diferentes, principalmente quando a adoção de padrões de intercâmbio pode aumentar a complexidade da tarefa e, conseqüentemente, seus custos.

No caso específico do INPE, o desenvolvimento do Sistema de Informações *Amazônia* (Alves et al, 1992), motivou o intercâmbio de dados entre diferentes sistemas, em particular entre o SGI e o ARC/Info.

O *Amazônia* é uma base de dados georeferenciados sobre os ecossistemas da Amazônia, que reúne, em seu estágio atual, 5 conjuntos de dados principais:

- a) TM  
Escala: 1/250.000  
Projeção: UTM  
Conteúdo: áreas desflorestadas entre 1984 e 1991; limites das áreas de florestas, hidrografia, limites estaduais, cobertura de nuvens.  
Fonte: imagens Landsat TM e mapas topográficos

- b) MSS  
Escala: 1/500.000  
Projeção: Lambert  
Conteúdo: áreas desflorestadas em 1975 e 1978  
Fonte: imagens Landsat MSS

- c) VEGE  
Escala: 1/1.000.000  
Projeção: Lambert  
Conteúdo: mapas de vegetação  
Fonte: mapas de vegetação do projeto RADAM

- d) ZOPOT  
Escala: 1/2.500.000  
Projeção: policônica  
Conteúdo: mapa de vegetação  
Fonte: mapa de vegetação do projeto ZOPOT

- e) MUNI  
Escala: 1/2.500.000  
Projeção: policônica  
Conteúdo: mapa de municípios da Amazônia Legal  
Fonte: mapa de municípios, em formato digital, produzido pelo IBGE.

O custo de criação de uma base de dados como essa é elevado e seus dados de grande utilidade para diversas instituições, justificando o desenvolvimento de métodos de intercâmbio de dados. No caso, o ARC/INFO é um dos sistemas de maior popularidade e, estando disponível no próprio INPE, pode também ser utilizado para tratamentos específicos.

O presente trabalho apresenta um método para conversão de dados de um plano de informações do SGI, da categoria mapa poligonal (SGI - manual do usuário) para uma *coverage* do ARC/INFO. O método prevê a conversão de dados geométricos (linhas e polígonos) e da classe de cada polígono, que é inserida no ARC/INFO na forma de atributo de polígono.

### Dados do SGI

O sistema SGI foi desenvolvido no INPE para microcomputadores PC com o sistema operacional DOS. Ele incorpora recursos para o tratamento de dados vetoriais e *raster*, permitindo tratar mapas temáticos, modelos numéricos de terreno (MNT) e também imagens de satélite (de Souza et al, 1990).

Mapas temáticos, também chamados mapas poligonais, são mapas que contêm polígonos e/ou linhas, representando entidades como municípios, classes de vegetação e drenagem.

Mapas temáticos são usados para representar, num único plano de informações - PI - (Erthal et al, 1988), entidades de um mesmo tema, como por exemplo, municípios (num mapa de municípios) e unidades de vegetação (num mapa de vegetação). Neste caso, o usuário associa uma classe (por exemplo, a classe de vegetação) a cada entidade.

Para associação da classe a um polígono, o SGI exige que um ponto denominado centróide seja colocado no interior do polígono. A classe é então associada ao centróide, o que permite que o polígono seja associado posteriormente à classe do centróide que contém, durante a poligonalização.

### Dados do ARC/Info

O ARC/INFO é um sistema disponível para diversas plataformas, sendo comuns no Brasil implementações em microcomputadores PC (PC ARC/INFO) e estações de trabalho com o sistema operacional UNIX.

Pode-se dizer que um PI da categoria mapa poligonal do SGI corresponde a um *coverage* (ESRI, 1991) no ARC/INFO. Ambos são formados por áreas e linhas, que representam entidades do mapa. O ARC/INFO permite, no entanto, a associação de um número

arbitrário de atributos a cada entidade, enquanto que o SGI, nas versões disponíveis atualmente, limita-se apenas a um atributo - a classe da entidade. Finalmente, vale lembrar que o INFO é o banco de dados do ARC/INFO que gerencia os atributos dos objetos.

Polígonos no ARC/INFO também podem receber um centróide, denominado LABEL, ao qual é associado um atributo especial - o identificador do polígono. Outros atributos podem ser associados a cada polígono tomando por base seu identificador.

### Programa de Conversão

O programa de conversão apresentado no final deste artigo baseia-se em dois princípios básicos:

1. As linhas e os centróides do PI original são convertidos em dois arquivos em formato aceito pelo comando GENERATE do ARC/INFO. Os elementos dos dois tipos são numerados seguindo sua ordem de aparição no PI.
2. A classe de cada polígono dá origem a um terceiro arquivo que associa os números (identificadores) dos polígonos às suas classes.

Tanto linhas como polígonos são introduzidos no *coverage* do ARC/INFO com o comando GENERATE. Já a classe de cada polígono é introduzida por uma sequência de operações que permitem adicionar o atributo no *coverage*, para em seguida introduzir os atributos no INFO.

### Modo de emprego

O programa cria os arquivos seguintes:

Arquivo 1: <projeto>\_<pi>.lin  
Arquivo 2: <projeto>\_<pi>.lab  
Arquivo 3: <projeto>\_<pi>.pol

contendo, respectivamente as LINES do *coverage*, os LABELS do *coverage* (identificadores dos polígonos

associados às coordenadas dos centróides), e as classes de cada polígono.

Para inserir esses dados no ARC/Info, deve-se usar o comando

```
GENERATE <coverage>
```

seguido das opções INPUT LINES e INPUT LABELS, para importar LINES e LABELS.

Em seguida a topologia é gerada (e.g. CLEAN).

Para associar as classes aos polígonos, deve-se criar uma tabela no Info:

```
DEFINE <tabela>, com os itens:
```

```
<cover-id> - 4,5,B
```

e

```
<classe> - 17,17,C
```

e usar JOINITEM <cover.pat> <tabela> <cover.pat> ... para associar as classes aos polígonos.

### Conclusão

O método de conversão apresentado aqui é simples e tem demonstrado sua funcionalidade no âmbito dos trabalhos do sistema de informações *Amazônia* e também de outros usuários.

Ele prevê apenas a conversão de mapas poligonais em *coverages* do ARC/INFO. A conversão de dados da categoria MNT deve ser baseada em princípios diferentes dos aqui utilizados.

Além da conversão de dados do SGI para o ARC/INFO, um programa para a conversão inversa também foi desenvolvido na DPI/INPE. Esse programa não é apresentado aqui, mas vale mencionar que foi usado, por exemplo, para a entrada do mapa de municípios do IBGE, criado em formato ARC/INFO no sistema *Amazônia*.

A conversão de mapas poligonais em *coverages* assim como a operação inversa podem ser facilmente estendidas para tirar proveito do conjunto de atributos armazenados pelo ARC/INFO e pelo SGI em suas versões futuras.

### Agradecimentos

Os autores agradecem todos aqueles que contribuíram para o uso dos sistemas de conversão, em particular Marcos e Ana Lúcia do IBGE, cujos dados estão sendo de grande importância para o *Amazônia*, e Bob da EletroNorte, pelas sugestões. Igualmente, gostariam de registrar seu reconhecimento ao EGO da Helen pela impressão sem falhas, apesar do grande volume de trabalho.

### Bibliografia

Alves, D.S. et al, 1992 *The Amazonia Information System*, XVII Congress of the ISPRS, Archives, Washington D.C., August 1992

de Souza, R.C.; Câmara Neto, G.; Alves, D.S., 1990 *O Desenvolvimento de Sistemas de Informação Geográfica e de Processamento Digital de Imagens no INPE*, Simpósio Brasileiro de Geoprocessamento, Anais, São Paulo, Maio de 1990, pp 168-173

Erthal, G.J.; Câmara, G.; Alves, D., 1988 *Modelo de Dados Geo-Relacional: Uma Visão Conceitual de um Sistema Geográfico de Informações*, I Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI), Anais, Petrópolis, Brasil, Abril de 1988

Manual de programação SGI, sem data, Engespaço Indústria e Comércio, aprox. 350 pag.

ESRI, 1991 *ARC/INFO Data Model, Concepts, & Key Terms*, ARC/INFO (R) Users Guide, ESRI, Inc, 1991

Anexo - Listagem simplificada do programa

## ANEXO - Listagem do Programa

/\*\*\*\*\*\*

Resumo : Converte um PI do projeto ativo para o formato ARC

Entradas : Nome do plano de informacao

Saidas : Arquivo de linhas (arcos e ilhas)  
arquivo de centroides, contendo X, Y, classe  
arquivo de poligonos, contendo polig-id, classe

Notas: 1. variaveis em maiusculas sao constantes  
2. diversas variaveis sao struct definidas pelo SGI  
3. as definicoes de variaveis, struct e constantes  
que nao sao primordiais para a compreensao do codigo  
sao omitidas

\*\*\*\*\*/

main () {

dia\_titulo("Conversao SGI -> ARC");

/\*-----  
Inicializacao do projeto ativo  
-----\*/

if(sgi\_le\_contexto(nome\_projeto) != ERRO\_NORMAL)  
{  
dia\_erro ("Selecione projeto");  
exit (ZAP);  
}

if(dir\_carga (nome\_projeto,0) != ERRO\_NORMAL)  
{  
dia\_erro ("Diretorio nao existe");  
exit (ZAP);  
}

dia\_poe\_string (LIN,COL,"Nome do projeto : ",nome\_projeto);

/\*-----  
Definicao do plano de informacoes a ser convertido  
-----\*/

pergl: switch (dia\_peg\_string(LIN+2,COL,help,"Digite nome do plano : ",  
nome\_pi,COMPR\_PI))  
{  
case VOLTA: goto pergl;  
case FIM: exit (ZAP);  
...  
default: break;  
}

```

/*      Verifica se o plano existe
if(! pi_existencia(0,nome_pi)
{
    dia_erro("Plano nao existe");
    goto perg1;
}

/*-----
Criacao dos arquivos de saida:
a) arquivo de LINES (*) com extensao .lin
b) arquivo de LABELS (*) com extensao .lab
c) arquivo das classes dos poligonos, com
    extensao .pol
(*) - para uso com o comando GENERATE do ARC/INFO
obs: nome_arq e' formado por <nome_projeto>_<nome_pi>
-----*/

monta_nome_arquivo (nome_projeto,nome_pi,nome_arq);
for(;;) {
    if ((saida_lin=cria_arquivo(nome_arq,".lin")) == NULL) {
        dia_erro("Erro de criacao do arquivo de LINES");
        exit (ZAP);
    }
    if ((saida_lab=cria_arquivo(nome_arq,".lab")) == NULL) {
        dia_erro("Erro de criacao do arquivo de LABELS");
        exit (ZAP);
    }
    if ((saida_pol=cria_arquivo(nome_arq,".pol")) == NULL) {
        dia_erro("Erro de criacao do arquivo de POLIGONOS");
        exit (ZAP);
    }
}

/* geracao do arquivo de linhas */
if (GENERATE_LIN(nome_projeto,nome_pi,nome_arq,saidalin,per)
    != ERRO) {
    /* geracao dos arquivos de centroides e de classes */
    GENERATE_CEN(nome_projeto,nome_pi,nome_arq,saidacen,saidapol,per);
}
exit (ZAP);
}

```

```

/*****
      GENERATE_LIN - gera arquivo de linhas
*****/

```

```

GENERATE_LIN(nome_regiao,nome_pi,nome_arq,saida,per)

```

```

...

```

```

{
struct          desc_linhas      pdes;
struct          arq_linhas       plin;
double         offx, offy;
float          vx[NUM_PT], vy[NUM_PT];

```

```

/*-----
      Abertura do arquivo de linhas
      e inicializacoes
-----*/

```

```

if(linha_abre_arq(&pdes,&plin,nome_arq) != ERRO) {
    dia_erro("Erro na abertura do arquivo de linhas");
    return(ERRO);
}
if(pdes.des_nlinhas==0) {
    dia_erro("Arquivo de linhas nao existe ou vazio");
    return(ERRO);
}
dir_rec_offset  (0,&offx, &offy); /* offset do projeto*/
npt = pdes.des_nlinhas ;          /* numero de linhas do PI */
linha_rewind(&pdes);

```

```

/*-----
      Escreve linhas no arquivo .lin
      i,j - numeros da linha e ponto
-----*/

```

```

for(j=0,i = 1; i <= npt; i++) {
    if(linha leseq_head(&pdes,&plin,&ndalin) == ERRO) {
        dia_erro("ERRO - leitura das linhas do PI");
        return (ERRO);
    }
    if(linha_le(&pdes,&plin,ndalin,vx,vy,&npontos) == ERRO) {
        dia_erro("ERRO - leitura das linhas do PI");
        return (ERRO);
    }
    fprintf(saida,"%d\n", i);
    for(j=0;j<npontos;j++)
        fprintf(saida,"%15.6lf %15.6lf\n",
            (double)vx[j]+offx,(double)vy[j]+offy);
    fprintf(saida,"END\n");
}
fprintf(saida,"END\n");
linha_fecha_arq(&pdes);
return(OK);
} /* GENERATE_LIN */

```

```

/*****
      GENERATE_CEN - gera arquivos de centroides e classes
*****/

GENERATE_CEN(nome_regiao,nome_pi,nome_arq,saida,saida2,per)
...
{
struct          arq_tabela  ptab;
struct          desc_tabela pdtab;
char            rotulo [MAX_CLS][COMPR_CLS];
float           x,y;
double          offx,offy;

    dir_rec_offset  (0,&offx, &offy);
    /* Leitura da tabela de classes */
    if(tab_abre_arq(&pdtab,&ptab,nome_arq) == ERRO) {
        dia_erro("Arquivo tabela nao existe");
        return(ERRO);
    }
    if(pdtab.des_nclasses==0) {
        dia_erro("Arquivo tabela nao existe ou vazio");
        return(ERRO);
    }
        /* pdtab.des_nclasses - numero de classes */
    for (i=0;i<=pdtab.des_nclasses ;i++) {
        if(tab_le_atributos(&pdtab,&ptab,i) == ERRO) {
            dia_erro("Erro na leitura do arquivo tabela");
            return (ERRO);
        }
        strcpy (rotulo[i], ptab.tab_rotulo);
    }
    /* Leitura do arquivo de centroides */
    if(cent_le_arq(&nccents,pcen,nome_arq) == ERRO) {
        dia_erro("Arquivo de centroides nao existe");
        return(ERRO);
    }

    /*-----
      Escreve LABELS e CLASSES, nccents - numero de centroides
    -----*/
    for(i=ini;i<=nccents-1;i++) {
        if(cent_le_dados(nccents,pcen,&indice,&x,&y,i) == ERRO) {
            dia_erro("Numero de centroide invalido");
            return (ERRO);
        }
        fprintf(saida,"%4d %15.6lf %15.6lf\n",
            i+1, (double)x+offx, (double)y+offy) ;
        fprintf(saida2,"%4d,%s\n", i+1, rotulo[indice]);
    }
    fprintf(saida,"END\n");
    return(OK);
} /* GENERATE_CEN */

```