MINISTÉRIO DE MINAS E ENERGIA
Secretaria de Minas e Metalurgia
CPRM - Serviço Geológico do Brasil

RELATÓRIO DE VIAGEM À VENEZUELA

(Programa de Treinamento GIS/SPRING/INPE)

Eduardo Mendes de Oliveira Castro

CPRM
Serviço Geológico do Brasil

Ministério
de Minas
e Energia

Brasil

Março de 1999

Sumário

Anexos:

Anexo 1: Cópia do Diário Oficial da União - Autorização de viagem e afastamento do país conferida ao geólogo Eduardo Mendes de Oliveira Castro pelo Excelentíssimo Senhor Ministro de Minas e Energia Dr. Rodolpho Tourinho Neto.

Anexo 2: Certificado de participação do geólogo Eduardo Mendes de Oliveira Castro no *Workshop* GIS/SPRING/INPE.

Anexo 3: Cartões

Anexo 4: Manual de utilização do *software* SPRING 3.2[1]

---

[1] O manual do *software* SPRING 3.2 está anexado somente à cópia do relatório enviada a Divisão de Documentação Técnica (DIDOTE) na CPRM, localizada no Rio de Janeiro (Avenida Pasteur nº 404 bairro:Urca).

1. Introdução

O presente relatório refere-se à participação do geólogo Eduardo Mendes de Oliveira Castro no programa de treinamento GIS/SPRING/INPE realizado na cidade de Caracas, Venezuela, no período de 22 a 26 de março de 1999. Tal treinamento foi oferecido pela Secretaria "PRO TEMPORE" do Tratado de Cooperação Amazônica e custeado pelo Programa das Nações Unidas para o Desenvolvimento (PNUD). Através de contatos com o Coordenador Geral de Geologia e Recursos Minerais da Secretaria de Minas e Metalurgia (SMM/MME) Sr. Rubens Rulli Costa, a CPRM - Serviço Geológico do Brasil, por intermédio do Chefe de Assessoria de Assuntos Internacionais (ASSUNI) Sr. Samir Nahass, manifestou interesse na participação do referido geólogo, lotado na Divisão de Geoprocessamento (DIGEOP - ERJ). A autorização para a viagem foi concedida pelo Excelentíssimo Senhor Ministro de Minas e Energia Dr. Rodolpho Tourinho Neto e publicada no Diário Oficial da União de 12 de Março de 1999, seção 2 (anexo).

2. Objetivos

A participação do geólogo da CPRM no *Workshop* GIS/SPRING/INPE teve como principais objetivos:

1. Dar continuidade ao treinamento na área de geoprocessamento, possibilitando ao participante aprimorar sua formação técnica.
2. Avaliar o *software* SPRING 3.2 com relação à sua possível utilização em determinados projetos desenvolvidos pela CPRM, nas áreas de Sensoriamento Remoto e SIG.
3. Estabelecer contato com profissionais de outras instituições — nacionais e estrangeiras— que trabalham na área de geoprocessamento.
4. Difundir técnicas aprendidas durante o programa de treinamento para outros profissionais que utilizam as ferramentas de geoprocessamento nas suas atividades dentro da CPRM.

3. Programa de Treinamento

O curso teve a duração de cinco dias, enfocando a utilização do *software* SPRING 3.2 (manual em anexo[2]), desenvolvido pelo Instituto Nacional de Pesquisas Espaciais (INPE), nas áreas de Sensoriamento Remoto e Sistemas de informações Geográficas. Abaixo, encontra-se discriminado o programa de treinamento, cumprido integralmente pelo participante (certificado em anexo)

Local: Oficina Central de Estadística e Informática (**OCEI**)
Av. Boyacá, Edificio Fundación La Salle, Maripérez
Caracas

Dia 22/03/1999 - segunda-feira
Visão geral do *software* SPRING 3.2
Manipulação de imagens com o módulo IMPIMA

Dia 23/03/1999 - terça-feira
Registro de Imagens
Processamento de Imagens

Dia 24/03/1999 - quarta-feira
Classificação de imagens multiespectrais
Manipulação de dados vetoriais

Dia 25/03/1999 - quinta-feira
Modelagem Numérica (Modelo Digital de Terreno - MDT)
Análise e Consulta Espacial de Dados

Dia 26/03/1999 - sexta-feira
Linguagem LEGAL
Geração de Cartas e Impressão (Módulo SCARTA)

---

[2] O manual do *software* SPRING 3.2 está anexado somente à cópia do relatório enviada a Divisão de Documentação Técnica (DIDOTE) na CPRM, localizada no Rio de Janeiro (Avenida Pasteur nº 404 bairro:Urca).

4. Contatos Pessoais

No decorrer do *Workshop*, houve a oportunidade de estabelecer contato com profissionais de diversos países que atuam na área de geoprocessamento. Este contato permitiu o intercâmbio de informações sobre o trabalho desenvolvido por estes especialistas e pelas instituições a que pertencem. Abaixo, encontram-se relacionados os profissionais contactados:

Nassem Nasir - GIS Analyst - Centre for the Study of Biological Diversity
University of Guyana, TurKeyen, Greater Georgetown, Guyana.
e-mail: nnasir@guyana.net.gy

Rajkumar Singh - Enviromental Officer - Enviromental Protection Agency
IAST building, UG Campus, Turkeyen, Greater Georgetown, Guyana
e-mail: raj2singh@usa.net

Susy Lewis - Project Officer (Forestry) - Guyana Natural Resources Agency
41 Brickdam & Boyle Place, Georgetown, Guyana
e-mail: nrmp@solutions2000.net

Doris Marlene Huerta Rodriguez - Coordinadora de GIS - Oficina Central de Estadística e Informática, Dirección de Sistema de Información Geográfica y Estadística (SIGE)
Av. Boyacá, edif. Fundación la Salle, Maripérez, Caracas, Venezuela
tel. (058) 02-7932356
e-mail: dhuerta69@hotmail.com

Enrique Meza - Geógrafo - Oficina Central de Estadística e Informática, Dirección de Sistema de Información Geográfica y Estadística (SIGE)
Av. Boyacá, edif. Fundación la Salle, Maripérez, Caracas, Venezuela
tel. (058) 02-7932356
email: Emeza85@hotmail.com

Jose Gregorio Salas Varela - Geógrafo - Oficina Central de Estadística e Informática, Dirección de Sistema de Información Geográfica y Estadística (SIGE)
Av. Boyacá, edif. Fundación la Salle, Maripérez, Caracas, Venezuela
tel. (058) 02-7932356
e-mail: jsalas2000@hotmail.com

Olimpia Attanasi de Hernández - Directora de Formación y Cooperación Técnica
Oficina Central de Estadística e Informática (OCEI)
Av. Boyacá, edif. Fundación la Salle, Maripérez, Caracas, Venezuela
email: ATANASI@PLATINO.GOV.VE

Taus Mohamed Idries - Deputy Director - Ministry of Natural Resources
Mirandastr 11-15, Paramaribo, Suriname
tel: (0597) 475435/411041
e-mail: nath@sr.net

Roos Clemens Alfons - Head of Departament - Central Bureau of Aerial Mapping
Maystr. 39 p.o. box 971, Paramaribo, Suriname
tel: (0597) 497246
e-mail: nath@sr.net

Sanredjo Edith Ardie - Chief of Subsection: Remote Sensing - Department of Natural Resources.
Planning Division of Forest Service
C.Jongbawwstr. 27, Paramaribo, Suriname.
tel: (0597) 472852/471980

Lubia Vinhas - Analista de Sistemas - Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1758 - Caixa Postal 515- 12227. São José dos Campos - São Paulo
e-mail: lubia@dpi.inpe.br

Thereza C. Carvalho Santos - Centro Integrado de Ordenamento Territorial - CIORD -
Universidade de Brasília
Campus Universitário Darcy Ribeiro Edificio Multiuso II salas AS03/AS07
e-mail: ciord@tba.com.br

5. Considerações Finais

A participação no *Workshop* GIS/SPRING/INPE foi extremamente proveitosa, já que contribuiu sobremaneira para aprimorar a formação do participante. Determinadas técnicas abordadas durante o treinamento poderão ser implementadas imediatamente nos projetos Rio de Janeiro e Tapajós, em fase de execução pela CPRM - Serviço Geológico do Brasil. Dentre estas técnicas, destaca-se a segmentação da imagem para proceder posterior classificação supervisionada por regiões e a utilização de linhas de quebra para geração de MDT (Modelo Digital de Terreno).

Outro ponto positivo desta participação foi o contato com profissionais de áreas afins, que permitiu o intercâmbio técnico entre os participantes.

Com relação ao *software* SPRING 3.2 destaca-se a possibilidade de sua utilização, tendo em vista sua gratuidade, em certas tarefas de geoprocessamento que não necessitem um *software* mais robusto.

Finalmente, vale ressaltar que o envio de técnicos da CPRM para este tipo de evento é altamente valioso, contribuindo sobretudo para a melhoria no quadro funcional da empresa, condição fundamental para fornecer à sociedade produtos de alta qualidade.

6. Agradecimentos

Gostaria de expressar o meu profundo agradecimento ao Chefe do Departamento de Informações Institucionais (DEINF) Sr. Ricardo Moacyr de Vasconcellos, bem como ao Chefe da Divisão de Geoprocessamento (DIGEOP) Sr. Paulo Branco, pela indicação do meu nome e a confiança depositada na minha pessoa.

Agradeço também o auxílio inestimável do chefe da Assessoria de Assuntos Internacionais Sr. Samir Nahass, sem o qual não se concretizaria esta viagem.

Expresso, ainda, minha gratidão, pela aprovação do meu nome, aos diretores da CPRM Dr. Augusto Wagner Padilha Martins (DRI) e Dr. Juarez Milmann Martins (DGM), ao Diretor-Presidente Dr. Carlos Oití Berbert, ao Coordenador Geral de Geologia e Recursos Minerais da Secretaria de Minas e Metalurgia, Sr. Rubens Rulli Costa, e ao Excelentíssimo Senhor Ministro de Minas e Energia, Dr. Rodolpho Tourinho Neto.

Finalmente gostaria de agradecer ao Programa das Nações Unidas para o Desenvolvimento (PNUD), representado pelo Sr. Ricardo Tichauer (Coordenador Residente - Venezuela), à Secretaria "PRO TEMPORE" do Tratado de Cooperação Amazônica, representada pelo Sr. Victor R. Carazo e à Oficina Central de Estadística e Informática (OCEI), representada pelo Sr. Miguel Bolívar Chollett.

Anexo 1

sexta-feira, 12 de março de 1999 — Diário Oficial — Seção 2
1545

omear TANIA MARIA GUIMARÃES E SOUZA MON- exercer o cargo em comissão de Assessor do Secretário Assistência Social, código DAS-102.4.

Nomear HELDO VITOR MULATINHO para exercer o missão de Gerente de Projeto do Departamento de Ca- Secretaria de Estado de Assistência Social, código

lomear TÂNIA HELOISA GUIMARÃES DE FREITAS o cargo em comissão de Coordenador do Departamento Fundo Nacional de Assistência Social, da Secretaria de sistência Social, código DAS 101.3.

Nomear NESTOR ALBINO GREWE para exercer, in o cargo de Superintendente Estadual do Instituto Na- guro Social em Alagoas, código DAS 101.3, sem pro- atuais funções.

WALDECK ORNÉLAS

11/99)

INSTITUTO NACIONAL DO SEGURO SOCIAL

ORTARIAS DE 10 DE MARÇO DE 1999

RESIDENTE DO INSTITUTO NACIONAL DO SE- AL - INSS, no uso da competência que lhe foi conferida S/GM/Nº 5.043, de 11 de fevereiro de 1999, resolve:

nerar o servidor AFONSO PESSÔA PICANÇO, ma- 30.266, do cargo em comissão de Chefe de Divisão de de Pessoal, código DAS-101.2, nº 52-70.079, da Di- ursos Humanos.

ar a servidora RITA GORET DA SILVA, matrícula nº á exercer o cargo em comissão de Chefe de Divisão de de Pessoal, código DAS 101.2, nº 52-70.079, na Di- ursos Humanos.

ar a servidora RITA GORET DA SILVA, matrícula nº ra o encargo de substituta eventual do Coordenador de de Pessoal, na Diretoria de Recursos Humanos, código 53-70.076, nos casos de impedimentos e afastamentos ar.

n consequência, cessar os efeitos da Portaria/INSS/PR 08/96.

CRÉSIO DE MATOS ROLIM

8/99)

Ministério da Saúde

GABINETE DO MINISTRO

ORTARIAS DE 11 DE MARÇO DE 1999

nistro de Estado da Saúde, no uso de suas atribuições,

ar MARIMI TEREZINHA PANTEL MOREIRA, para o de Chefe do Gabinete do Secretário de Gestão de em Saúde, código DAS 101.4, nº 38.0006.

ar VERA ROSANA NUNES VALENTE, para exercer rente de Projeto, código DAS 101.3, nº 38.0021, do de Projetos de Investimento, da Secretaria de Gestão s em Saúde.

JOSÉ SERRA

99)

RIA DE ASSISTÊNCIA À SAÚDE

ARIA Nº 61, DE 10 DE MARÇO DE 1999

etário de Assistência à Saúde, no uso das atribui

rando o contido no Edital de Convocação nº 01, de 1998, publicado no DO, de 07 de outubro de 1998,

rando o término do prazo de apresentação dos pro- ação e funcionamento de Pólos de Capacitação, For- ção Permanente para Pessoal de Saúde da Família

- Designar o Comitê Técnico de Análise e Aprovação Pólos de Capacitação, Formação e Educação Per- ssoal de Saúde da Família, constituído pelos técnicos :

MA DE SOUSA - COSAC/SAS - Presidente
INE JUNIOR - CODRH/SPA - Membro
SSOS NOGUEIRA - IPEA - Membro
ARTE DA FONSECA - SAS/MS - Membro
TSES DA COSTA NETO - COSAC/SAS

Art. 2º - Compete ao Comitê Técnico analisar, emitir parecer técnico, classificar os projetos e encaminhá-los à Unidade Gerenciadora do Projeto REFORSUS (UGP), observado o disposto no Termo de Referência e no Edital de Convocação nº 01, da Secretaria de Assistência à Saúde.

Art. 3º - Esta portaria entra em vigor na data de sua publicação.

RENILSON REHEM DE SOUSA

(Of. El. nº 390/99)

Ministério do Desenvolvimento, Indústria e Comércio

SUPERINTENDÊNCIA DA ZONA FRANCA DE MANAUS

Superintendência de Administração

PORTARIAS DE 10 DE MARÇO DE 1999

O SUPERINTENDENTE ADJUNTO DE ADMINISTRAÇÃO, por delegação de competência atribuída através da Portaria nº 00429/98, de 17 de dezembro de 1998, e tendo em vista o que consta do Processo nº 06100.0686/99, resolve:

Nº 87 - Art. 1º - CONCEDER Pensão Vitalícia a VERÔNICA MARIA DA SILVA CASTRO e Temporária a ISMALIA LISSIANE DA SILVA CASTRO, respectivamente viúva e filha do ex-inativo DEUSDEDITH MONTEIRO DE CASTRO, aposentado no cargo de Motorista Oficial, Classe "A", Padrão I, com fundamento nos artigos 215 e 217, incisos I e II, alínea "a" da Lei nº 8.112/90, com vigência a partir de 10 de fevereiro de 1999, data de falecimento do instituidor.

O SUPERINTENDENTE ADJUNTO DE ADMINISTRAÇÃO, por delegação de competência atribuída através da Portaria nº 00429/98, de 17 de dezembro de 1998, resolve:

Nº 88 - Art. 1º - DECLARAR vago, a partir de 12 de fevereiro de 1999, o cargo de Agente Administrativo, código 26002, Classe "A", Padrão II, em virtude da demissão de seu ocupante ÁTILA PASSOS CRUZ, Matrícula SIAPE nº 677843, do Quadro de Pessoa desta Autarquia, através de Decreto Presidencial, de 11 de fevereiro de 1999, publicado no DOU de 12 seguinte.

Nº 89 - Art. 1º - DECLARAR vago, a partir de 12 de fevereiro de 1999, o cargo de Agente Administrativo, código 26002, Classe "A", Padrão II, em virtude da demissão de seu ocupante JEZIEL VIEIRA DE SOUZA GALVÃO, Matrícula SIAPE nº 677999, do Quadro de Pessoa desta Autarquia, através de Decreto Presidencial, de 11 de fevereiro de 1999, publicado no DOU de 12 seguinte.

Nº 90 - Art. 1º - DECLARAR vago, a partir de 12 de fevereiro de 1999, o cargo de Agente Administrativo, código 26002, Classe "A", Padrão III, em virtude da demissão de seu ocupante RAIMUNDO DE OLIVEIRA GOMES, Matrícula SIAPE nº 678017, do Quadro de Pessoa desta Autarquia, através de Decreto Presidencial, de 11 de fevereiro de 1999, publicado no DOU de 12 seguinte.

Nº 91 - Art. 1º - DECLARAR vago, a partir de 12 de fevereiro de 1999, o cargo de Agente Administrativo, código 26002, Classe "A", Padrão I, em virtude da demissão de seu ocupante RENATO CARVALHO DA SILVA, Matrícula SIAPE nº 678027, do Quadro de Pessoa desta Autarquia, através de Decreto Presidencial, de 11 de fevereiro de 1999, publicado no DOU de 12 seguinte.

JOSÉ OSWALDO DA SILVA

(Of. El. nº 93/99)

Ministério de Minas e Energia

GABINETE DO MINISTRO

DESPACHO DO MINISTRO
Em 11 de março de 1999

Afastamentos do país autorizados na forma do disposto no decreto no 1.387, de 7 de fevereiro de 1995, com a nova redação dada pelos decretos no 1.701, de 14 de novembro de 1995, no 2.349, de 15 de outubro de 1997 e, no 2.809, de 22 de outubro de 1998.

NOME: Eduardo Mendes de Oliveira Castro CARGO/FUNÇÃO: Geólogo/Especialista da Divisão de Geoprocessamento ÓRGÃO: CPRM PAÍS DE DESTINO: Venezuela FINALIDADE DO AFASTAMENTO: Participar do Programa de Treinamento em GIS/SPRING/INPE, oferecido pela Secretaria "Pro Tempore" do Tratado de Cooperação Amazônica e financiado pelo Programa das Nações Unidas para o Desenvolvimento (PNUD/GEF) PERÍODO: 21/03/99 a 27/03/99 TIPO DE AFASTAMENTO: Com ônus limitado ENQUADRAMENTO DA VIAGEM: Artigo 1o, inciso IV

ROBERTO TOURINHO NETO

(Of. El. nº 62/99)

Ministério do Orçamento e Gestão

SECRETARIA EXECUTIVA

PORTARIAS DE 10 DE MARÇO DE 1999

O SECRETÁRIO EXECUTIVO DO MINISTÉRIO DO ÇAMENTO E GESTÃO, no uso da competência que lhe foi delegada pelo art. 2o da Portaria no 29, de 19 de janeiro de publicada no Diário Oficial de 20 de janeiro de 1995 e tendo em o disposto no art. 33, inciso IX, da Lei no 8.112, de 11 de deze de 1990 e no art. 45 da Lei no 9.649, de 27 de maio de 1998, alt pelo art. 1o da Medida Provisória no 1.799-2, de 18 de fevere 1999, resolve:

Nº 54 - Declarar vago o cargo em comissão de Coordenador-Ger Planejamento e Formulação, código DAS 101.4, do Departamen Políticas Regionais, da Secretaria Especial de Políticas Regio deste Ministério, em virtude do falecimento de seu ocupante, LIO DINIZ GUIMARÃES, em 31 de janeiro de 1999.

O SECRETÁRIO EXECUTIVO DO MINISTÉRIO DO ÇAMENTO E GESTÃO, no uso da competência que lhe foi delegada pelo art. 2o da Portaria no 29, de 19 de janeiro de publicada no Diário Oficial de 20 de janeiro do 1995 e tendo em o disposto no art. 45 da Lei no 9.649, de 27 de maio de alterado pelo art. 1o da Medida Provisória no 1.799-2, de fevereiro de 1999, resolve:

Nº 55 - Exonerar, a pedido, PATRICK FARIA COURA do carg Auxiliar, código DAS 102.1, da Coordenação Geral de Planejam e Formulação, do Departamento de Políticas Regionais, da Secre Especial de Políticas Regionais, deste Ministério, a contar de fevereiro de 1999.

MARTUS TAVARES

(Of. El. nº 17/99)

Ministério da Ciência e Tecnologia

GABINETE DO MINISTRO

PORTARIA Nº 79, DE 10 DE MARÇO DE 1999

O MINISTRO DE ESTADO DA CIÊNCIA E TECNO GIA, no uso da competência que lhe foi delegada pelo artigo inciso I, do Decreto nº 2.947, de 26 de janeiro de 1999, resolve

Exonerar, a pedido, a partir do dia 8 de março de 19 IVAN ROCHA NETO, do cargo em comissão de Coordenador-G de Programas, código DAS-101.4, da Secretaria de Desenvolvime Científico deste Ministério.

LUIZ CARLOS BRESSER PEREIRA

(Of. El. nº 96/99)

Ministério Público da União

ATOS DO PROCURADOR-GERAL DA REPÚBLICA

PORTARIAS DE 10 DE MARÇO DE 1999

O PROCURADOR-GERAL DA REPÚBLICA, no uso suas atribuições, tendo em vista o que consta do Ofício OPC 2256/99, de 03-3-99, da Procuradora-Chefe da Procuradoria da pública no Estado de São Paulo, e do Ofício nº 62/99-LA-GA PR/DF, de 9.3.99, do Procurador-Chefe da Procuradoria da Repúbl no Distrito Federal, e por necessidade de serviço, resolve:

Nº 91 - Designar o Procurador Regional da República BRASILIN FERREIRA DOS SANTOS, lotado na Procuradoria da República Distrito Federal, para oficiar no período de 15 de março a 9 de ab de 1999, junto à Procuradoria da República no Município de São Jo do Rio Preto, Estado de São Paulo.

O PROCURADOR-GERAL DA REPÚBLICA, no uso suas atribuições, e tendo em vista o que consta do Ofício nº 047/ de 03-3-99, resolve:

Nº 92 - Designar o Procurador da República SÉRGIO LAURI FERREIRA, lotado na Procuradoria da República no Estado do Am zonas, para participar do VII Congresso Nacional de Estudos T butários, que será realizado na Capital do Estado do Rio de Janei nos dias 25 e 26 de março de 1999.

GERALDO BRINDEIRO

Anexo 2

Secretaría
Pro Tempore
Venezuela

GEF

pnud

FAO
FIAT PANIS

República de Venezuela
Presidencia de la República
Oficina Central de Estadística e Informática

United Nations Development Programme,
Global Environment Facility,
Food and Agriculture Organization,
Amazon Cooperation Treaty
and
La Oficina Central de Estadística e Informática

Certify that:

Eduardo Mendes de Oliveira Castro

Has participated in the GIS/SPRING/INPE Workshop
from March 22 to March 26 1999 for a period of 40 hours

Víctor R. Carazo
Pro Tempore Secretary
Tratado de Cooperación Amazónica

Ricardo Tichauer
Resident Coordinator
United Nations - Venezuela

Miguel Bolivar Chollett
Jefe de la Oficina Central
de Estadística e Informática de
la Presidencia de la República

Caracas, 26 March 1999

Anexo 3

CSBD

Naseem Nasir
GIS Analyst

Centre for the Study of Biological Diversity

49 Chandra Nagar Street
Prashad Nagar
Georgetown, Guyana
Tel:02-60311

CSBD
University of Guyana
Turkeyen Campus
Georgetown, Guyana
Tel/Fax:022-4921

PRESIDENCIA DE LA REPUBLICA

Lic. Olimpia Altanasi de Hernández
Directora de Formación y Cooperación Técnica
Oficina Central de Estadística e Informática

TELFS. DIRECTO: 793.15.03
MASTER: 782.11.33 - EXTS. 114-261
E-MAIL: ATANASI@PLATINO.GOV.VE

Rajkumar Singh
Environmental Officer

http://www.sdnp.org.gy/epa

ENVIRONMENTAL PROTECTION AGENCY
GUYANA

IAST Building
UG Campus, Turkeyen
Greater Georgetown
Guyana
email: epa@sdnp.org.gy

Executive Director	(592) 22-2442
Administrative Director	(592) 22-4224
Operations Director	(592) 22-2277
General	(592) 22-5784
Fax:	(592) 22-2231

Oficina Central de Estadísticas e Informática
Oficina de Sistemas de Información Geográfica y Estadística

Enrique Cimarú Meza González
Geógrafo
Departamento de Digitalización

Tel. Cel: 014-2058486
Telefax Oficina: (02) 793-23-56
E-mail: emeza85@hotmail.com

Oficina Central de Estadísticas e Informática
Oficina de Sistemas de Información Geográfica y Estadística

Doris Marlene Huerta Rodríguez
Geógrafo
Coordinadora de GIS

Tel. Cel: 016-8144368
Telefax Oficina: (02) 793-23-56
E-mail: dhuerta69@hotmail.com

INPE

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Lúbia Vinhas, MsC
Divisão de Processamento de Imagens

Av. dos Astronautas, 1758 - Caixa Postal 515 - 12227-010 - São José dos Campos
SP - Brasil - Tel: (012) 345-6523 - Fax: (012) 345-6468
e-mail: lubia@dpi.inpe.br URL//www.dpi.inpe.br

Centro Integrado de Ordenamento
Territorial - CIORD

Universidade de Brasília - UnB

Prof^{a}. Thereza C. Carvalho Santos - PhD
Diretora

Campus Universitário Darcy Ribeiro
(061) 348-2551

Anexo 4

SPRING-3.2 Training Tour

Outline

CLASS 1 – SPRING OVERVIEW

This class presents a Spring overview. You are going to use a database, named CURSO, which allows you to overview SPRING's conceptual model, the interfaces and control data presentation on drawing areas.

CLASS 2 – IMAGE INPUT - IMPIMA

This class presents how to manipulate images, using a dedicated application named IMPIMA. You are going to read a TM-Landsat image from a CD-ROM. Moreover you are going to learn how to read images using other formats.

CLASS 3 – IMAGE REGISTERING

This class presents how to register a previously read image, using the IMPIMA application.
You are going to learn how to get control points from paper maps, images, digital maps and external landmarks. Besides, you will create a simple database and a project to hold the georeferenced image, corrected geometrically to the desired cartographic projection.

CLASS 4 – IMAGE PROCESSING

Going trough this class you are going to exercise most image processing techniques, using your own database or CURSO database.

CLASS 5 - CLASSIFICATION

This class will show you how to classify a multi-spectral image creating an associated thematic image. Supervised and non-supervised methods will be presented allowing pixel and region based classification algorithms. Besides you are going to learn how to mosaic images.

CLASS 6 – MANAGING VECTOR DATA

This class presents SPRING's vector data structure. You are going to create Thematic Maps, importing data form ASCII files and using interactive edition tools. You will learn how to convert data from raster to vector representations and vice-versa, analyzing its differences.

CLASS 7 - DIGITAL TERRAIN MODELING

This class presents the SPRING's digital (numeric) terrain model and features. A previously loaded map with isovalue lines and quoted landmarks can be found on CURSO database, along with regular grids an triangular nets. You could create your own data, modifying or editing the existing ones, or use the existing data as is.

CLASS 8 – SPATIAL QUERY ANALYSIS

This class presents the SPRING's spatial analysis features, object manipulation interface and LEGAL.

CLASS 9 – MAP DESKTOP EDITOR - SCARTA

This class presents the desktop map module, named SCARTA, which allows you to produce cartographic printed maps from a SPRING's projects.

Class 1 – SPRING Overview

1. Introduction

SPRING , an acronym for **S**istema de **PR**ocessamento de **IN**formações **G**eo-referenciadas (Geo-referenced Information Processing System, is a georeferenced database, for MS-Windows and Unix computational environments with the following characteristics:

- Operates on a geographic database without scale, projection or area limitations, maintaining the identity of objects along the whole database;
- Manages vector and matrix (raster) data and integrates remote sensing and digital terrain modeling data into the GIS environment;
- Allows a friendly and powerful work environment, combining Windows, menus and a spatial language, named LEGAL, an acronym for Liguagem Espacial de Geoprocessamento Algébrico, a Spatial Language for Algebraic Geoprocessing.

SPRING is based on an object oriented model, from which is derived its conceptual model, interfaces and spatial language features as well as a set of powerful algorithms like those for spatial indexing, image segmentation, triangular networks and the interesting functionality provided by Legal, allowing suitable performance for many sound actual applications. The interactive and friendly interface was designed using concepts close to the real world realm and user domain applications. SPRING is designed for MS-Windows and Unix operating environment. All data management in Spring is organized by a common database structure valid for both systems.

Spring have been developed by the Image Processing Division (DPI) at the National Institute for Spatial Research (INPE) in Brazil, with the commitment to provide a powerful tool to improve the use of Remote Sensing technology in Brazil. SPRING is the result of research and technologic development held on INPE since the beginnings of civil use of earth observation satellites.

SPRING is freeware software product that can provide a wide range of GIS, Image Processing and Remote Sensing functionality at low costs. It can be downloaded from DPI web page on http://www.dpi.inpe.br. Other classes of products from spatial technologies available at INPE can be obtained from http://www.inpe.br .

2. SPRING's Database

The SPRING's database is archived in a directory which stores the database definition Design, the data base general data, the spatial data associated with each project. The project's data are stored on sub-directories where the spatial data like points, lines, polygons, images and grids remain.

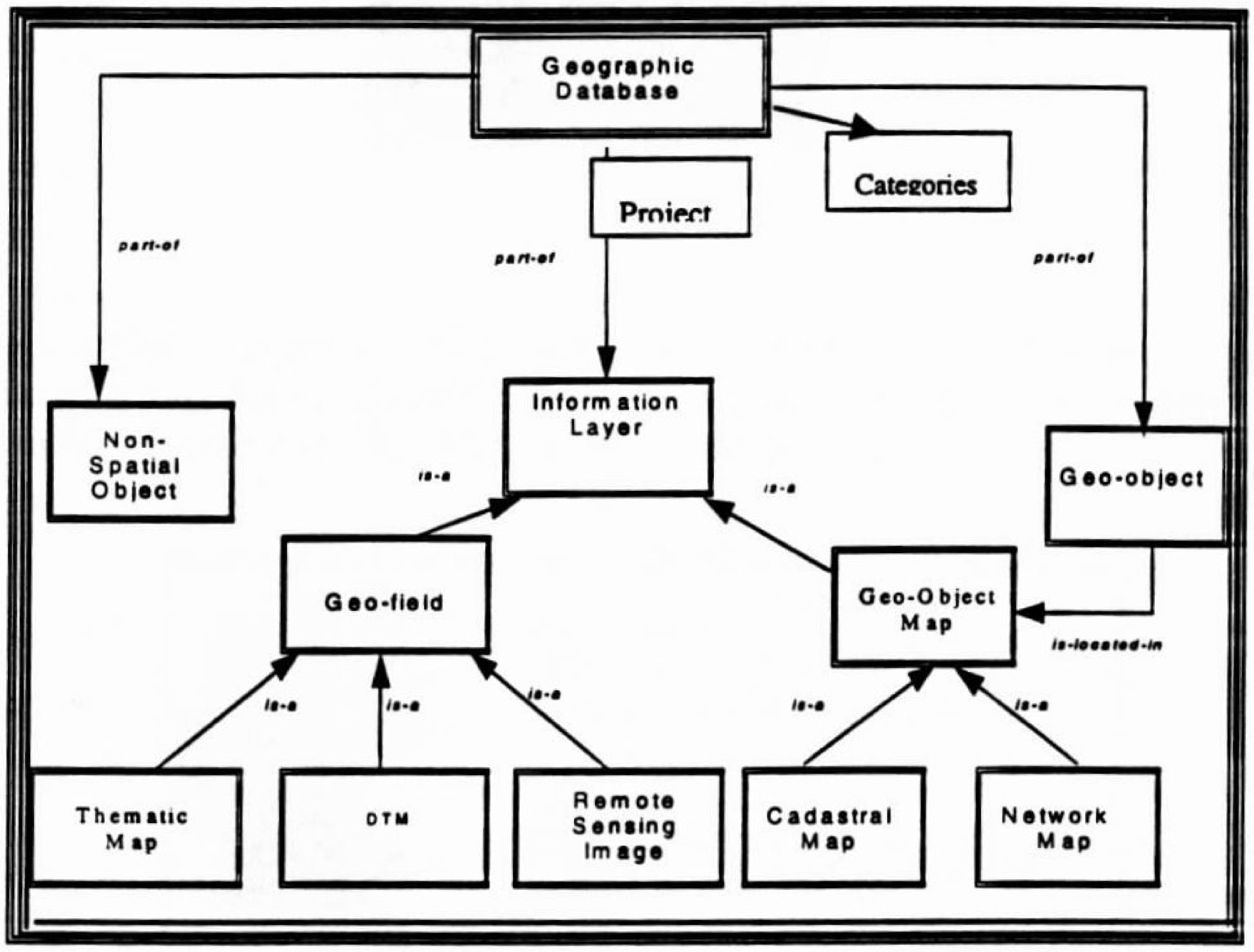
To activate a database the "login" user must have read permission for all files, and only one database can be opened for each user session.

OBSERVATION:

MS-Windows environment: The last database and the last project used in the previous session are automatically activated on starting of a new SPRING section.

UNIX environment: For a database to be automatically activated on starting the environmental variable "SPRINGDB" must refer to its complete folder path.

The following figure presents the SPRING's database structure:



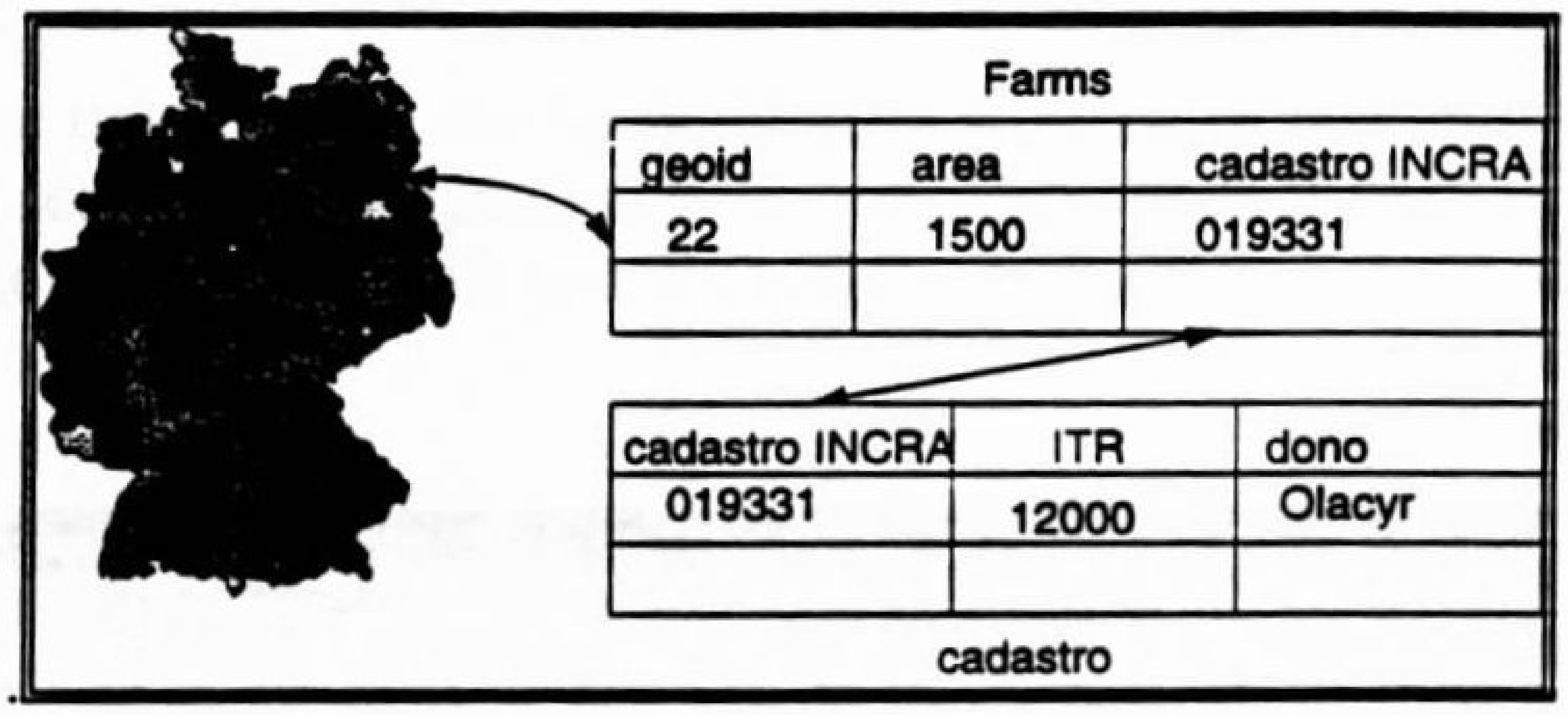
A **geofield** will represent the spatial distribution of some geographic phenomena. Each point in the geographic area covered is associated to a quantitative or qualitative value.

A geofield can be specialized into the following model types:

- **Thematic**: A Geo-field that relates each point of a geographic region to a theme, e.g., a soil map associating a soil type to each point in the covered area. Those types of qualitative values are usually referred to as classes or themes. In the Spring jargon we will use the term **geoclass** when referring to classes or themes.
- **Numeric** or **Digital**: A geo-field that relates each point of a geographic region to a real numerical value, e.g., an altitude grid map giving altitude values at each point of a covered area.

- **Image**: A geo-field that relates each point in the geographic space to quantities representing levels of radiometry into an integer valued scale, normally visualized as grey level picture, ranging 0-255, as shown above.

A **Geo-Object** refers to some data element characterized by a set of attributes along with its geographic location, that can be defined by one or more sub-regions, even disconnected, of the geographic space. It must be unique in the whole database.



Farms

geoid	area	cadastro INCRA
22	1500	019331

cadastro INCRA	ITR	dono
019331	12000	Olacyr

cadastro

An **Infolayer** refers to the information layers containing all data related to geo-field or geo-object representation e.g., the raster representation of an image or a topographic map, the vector representation of a thematic map etc. A **geo-objects map** belong to a kind of infolayers used to represent geo-objects. A **Project** in the Spring database model can be viewed as a meaning collection of infolayers and geo-object maps.

All Spring's interface are mostly driven by button clicks and other sensitive elements. Some few typing is yet required.

This Spring tour makes use of the following notation to indicate the sequences of operations the user must perform:

#Spring-3.2 or icon Spring( ) : spring version to execute
[Menu] – Menu option to select in the main window
{Field Name: String} – Field to fill in with string
(Button Name) – Button to click on
Field Name: (Button Name) – Button to select among others in a field
(<u>List Name</u> : Item) – Specific Item to be selected in a named scrolled list
(<u>List Name</u> : Item1, Item2, Item3, ...) – Named items that must be selected in a non-exclusive scrolled list
Window Name – Activated window in **bold** letters

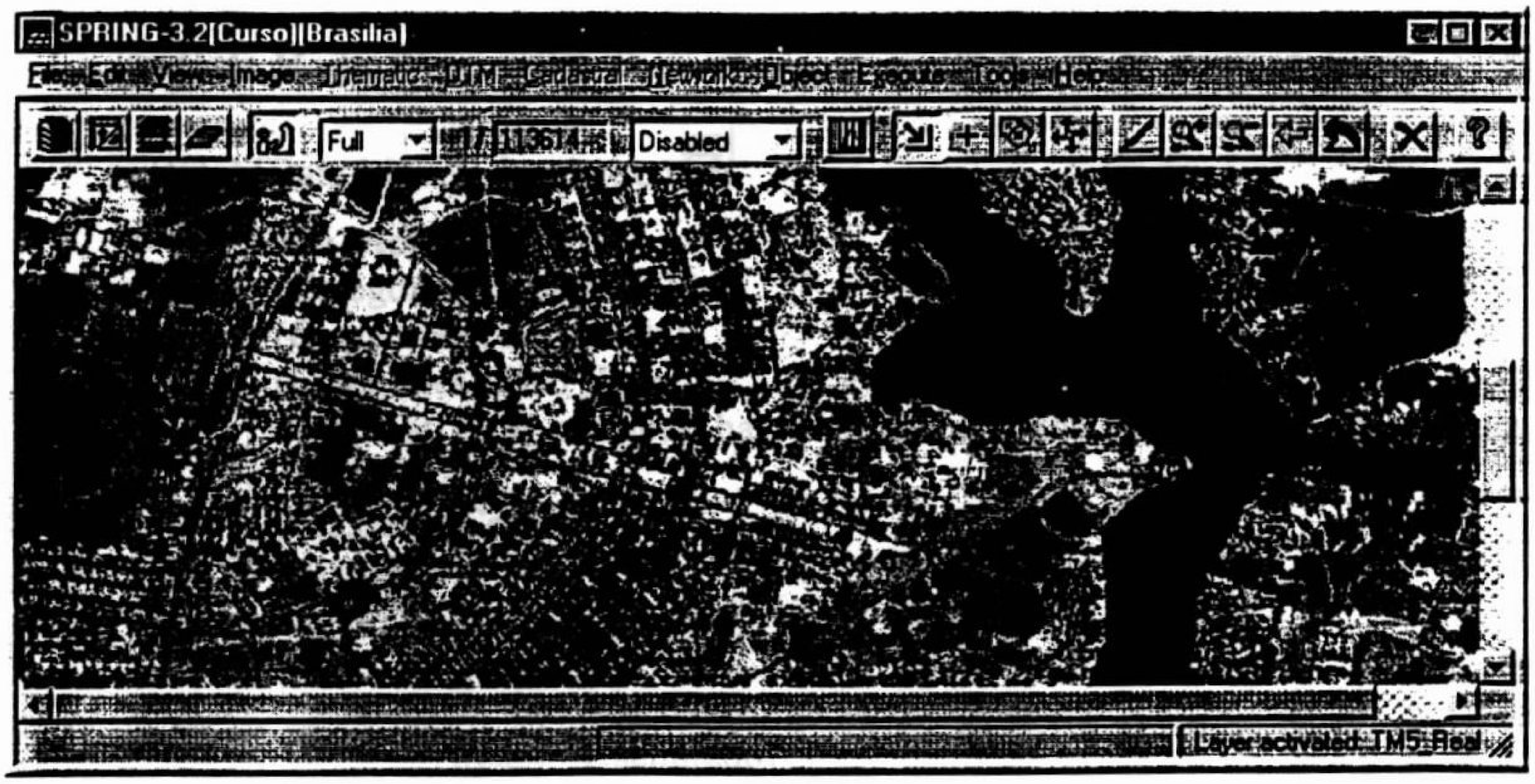
⇒ ***Activating a Database***

#Spring-3.2 or icon ***Spring( )***

- [File] [Database] or button

Database
- (Databases: CURSO)
- (Active)

The following figure shows the Spring's main window, which allows the access to all functionality by selecting the options of the menu bar or tool bar. The menu bar is sensitive and only the allowed operations are shown.



3. SPRING's Data Model

Before loading any data into Spring you need to create a Database and to define its Data Model, in the same way that you must do in a conventional database. The Data Model defines classes of homogeneous data that will be used in your application, named **Categories**. Each Category must hold elements that are modeled by only one of the concepts: Thematic geo-field, Numeric geo-field, Image geo-field, Geo-object, Cadastral geo-objects map, Network geo-object map, and Non spatial objects, which are a specialization of the main concepts: geo-fields an geo-objects. Therefore, each Category must be associated with only one of these concepts, named ***Models***.

Actually all Spring infolayers, geo-object maps and the geo-objects will be associated to some category.

Categories may be of any available model type: Thematic, Digital and Image – or of a geo-object map type also referred to as a **cadastral map** or even as a pure geo-object type.

The visualization characteristics to be assigned to each entity (Area, line, point and text) in Spring are specified from the **Data Model** interface through the **Visual** button.

All categories must not be created at this first moment, they could be created during the operation. However once created and populated with data it can not be deleted. The deletion of a category and all data hold by it, is a time consuming operation better accomplished in batch (not available on 3.2 version).

For a thematic category a set of geo-classes must be defined along with their *Visual* characteristics. For geo-objects categories the attribute names and types should be defined. A table will be created in the database to hold actual attributes of each object to be introduced later.

⇒ Data Model

- [File] [Data Model...] or button ▤

Data Model
- (Categories : Imagem_TM)
- (Categories : Altimetria)
- (Categories : Uso_Terra)
*Note that each category has a model and that a thematic one defines classes.
- (Classes : Cerrado)
- (Visual...)

Graphical Presentation Properties

- Area:(*:SOLIDO)
- Area: (*:Hachurado)
- Area :(Color...)

Color

- Select a color

*Try to change visuals for line, points and texts

- (Close)

Data Model

- (Close)

4. Projects

A project defines a work area. To create a project you must inform the project's name, the cartographic projection that you intend to use, and the extent of the work area given as a bounding box. Actually the system creates a sub-directory, under the database directory previously created with the same name of your database, to hold the graphical information related with this project. To create a project you need only to have an active database. No restrictions are imposed on the number of databases that you create, but only one can be activated each turn.

A project holds an arbitrary number of Infolayers, which stores geo-fields and geo-object maps on the geographic area covered by project. All data hold are conform the cartographic projection specified in the project. Therefore any data imported to the project will be remapped to the proper projection automatically.

⇒Activating a Project

- [File] [Project...] or button

Projects

- (Project : Brasilia)
- (Projection...)

Projections

- (Systems : UTM)

Note that you can choose 13 different projection systems and different earth ellipsoids. Missing ellipsoids can be included inserting the parameters into the ASCII file named datum in the Spring home sub-directory **/etc.** (). Depending upon the selected cartographic system you must fill the parameters that will become visible in the interface: **Hemisphere**, **Origin Latitude**, **Origin Longitude , or Standard Parallels.**

- (Close)

Projects

- (Coordinates: Geographic or Planimetric)

* Note that a bounding box can be defined in Planimetric coordinates (meters) or geodetic coordinates (degrees, minutes and seconds).

- (Active)

NOTE : Once a project is activated a window named **"Control Panel"** is presented together with Spring's main window. Using this window you control the visualization of data and enable the menu functions that could be applied to the selected INFOLAYER active.

5. Infolayer Visualization

Using the **"Control Panel"** window you can select which INFOLAYER and associated representation you want to visualize or select to use in operation.

To select an Infolayer just click on the scrolled list Category the data desired class of data and after select an Infolayer. It then become active and can be used in a geographical operation. Only allowed operations will become available in the interface.

Categories – Only the categories which hold some Infolayer will be presented. Selected a category all Infolayers are presented. A (V) appearing before the category name indicates that some of its Infolayers are selected to be visualized. Double clicking on visualized category will suppress the visualization of all Infolayers of that Category.

Infolayers - The representations available for the selected infolayers are shown immediately below the correspondent scrolled list..

Once selected an Infolayer the allowed operations become accessible in the menu bar. You do not need to visualize a data to operate on it, just active it.

⇒ ***Visualizing a monochromatic image in the main window***

Control Panel or button

- (Categories: Imagem_TM)
- (Layers: TM5)
- (M) , to visualize in grey levels.

SPRING-3.2

- (Draw) or button

Control Panel or button

- (Show: Window 2)
- (Active: Window 2)
- (Categories: Imagem_TM)
- (Layers: TM5_Realce)
- (M)

SPRING-3.2

- (Draw) or button

Compare both images.

⇒ ***Visualizing and airborne photo in main***

Control Panel

- (Active: Window 1)
- (Categories: Imagem_foto)
- (Information Layer: Foto1)
- (M)

SPRING-3.2

- (Draw) or button

⇒ ***Visualizing a synthetic (8 bit with luts) image on main window.***

Control Panel or button

- (Categories: Imagem_TM)
- (Layer: Comp_3B_4R_5G)
- (Synthetic Image)

SPRING-3.2

- (Draw) or button

On this image each pixel is associated with a lookup color table.

⇒ ***Creating a RGB color composition in the main window***

Control Panel or button

- (Categories: Imagem_TM)
- (Layer: TM3_Realce)
- (R) to visualize using a red lookup color table (LUT).
- (Layer: TM4_Realce)
- (G) to visualize using a green LUT.
- (Layer: TM5_Realce)
- (B) to visualize using a blue LUT.

SPRING-3.2

- (Draw) or button

NOTE: Besides the main window the user could use more 4 windows, labeled for 2 to 5. The fifth Windows is reserved to present the image being registered (see registering Images). Any window can be shown or hide using the window control of **"Control Panel"**.

Zooming a region of the work area

Once we have drawn some Infolayer we can zoom a region of the work area, using an eye glass. Select a *zoom* factor (2,4,8) on bottom of the Control Panel . Moving the cursor over the area the corresponding zoom image will appear on a small window named ZOOM. This facility only replicates the information shown on the window.

Visualizing a work area with better resolution

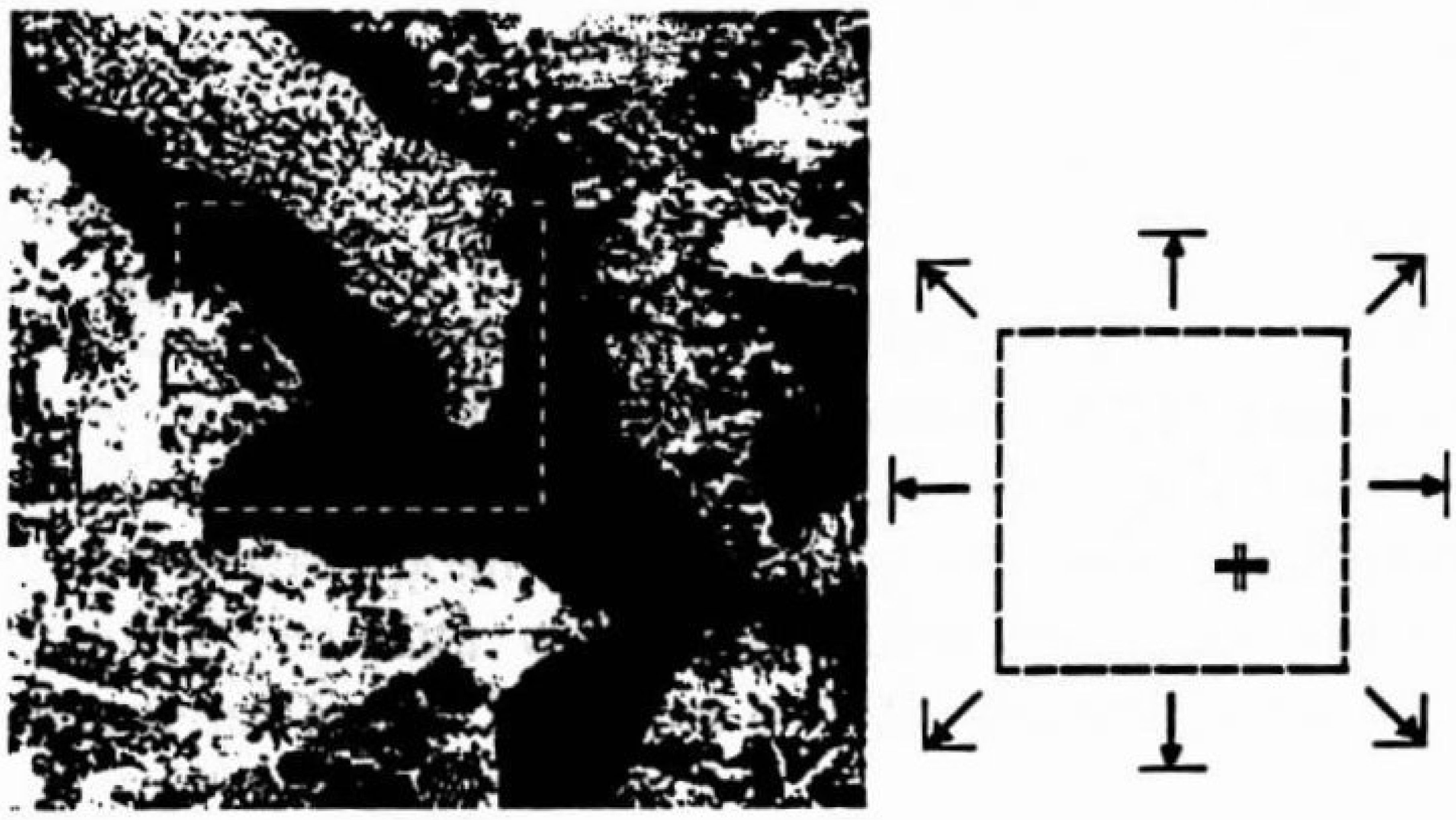
The **Area Cursor** button or **[View][Area Cursor]** is used to define a specific area and show it with a better resolution.

It is active when the cursor format were (↘ p/)

In order to define the specific area, click once over the upper left corner of the area, then move to the lower right corner and click again. A rectangle will be draw over the image. This area will be shown, fitting the window, after clicking on **Draw** button [Execute][Draw].

NOTES: To **disable** the **Area Cursor**, just click over the button or **[View][Area Cursor]**, or the **mouse's right button.**

To move the rectangle over the image, click inside it and move it. To change the size of the region (rectangle) click over one of the side of the rectangle and move it to the new position and click again (the rectangle will increase or decrease). See the figure below.



Visualization Windows

The visualization Windows or canvases and the visualized data can be shown using the **Control Panel** selection lists, each canvas has its own visualization controls. The window footer, on the right side shows the active Infolayer name. On the left side the coordinates pointed by the cursor can appear, depending on the scale mode selection. A total of five canvas can be used to display data and whenever one is selected, the **Control Panel** will indicate by a ticking in the infolayers list which of them participates in the canvas content.

Auto/Full/Scale Menu Choice

The presentation scale of an Infolayer can be controlled by the choice menu located on the footer of drawing window:

- **Auto:** the whole information selected will be fitted in the canvas.
- **Full:** each pixel of an visualized image will show using a dot in the screen. All image pixels will be shown without resampling or replication.
- **Scale:** the information will be shown in a particular graphic scale supplied by the user in the appropriated text field. If the drawing area becomes greater than the Windows, scrolled bars are automatically supplied to pan over the whole picture. A message ask you to reduce the scale if the picture generated uses to much system memory.

Navigation

As a project is cartographicaly defined any pixel on the canvas can have its coordinates shown. The choice menu **Disable/Planimetric/Geodetic** enables or disables the navigation feature.

Draw , Reset and Previous

The **Draw** button or **[Execute][Draw]** shows on the canvas all data selected for visualization in the **Control Panel** interface, for the selected **InfoLayer**.

The **Reset** button or **[Execute][Reset]** fits the project area into the whole canvas, resetting all previously done zoom operation and turning the scale mode to Automatic (**Auto**).

The **Previous** button or **[Execute][Previous]** undo the last action over the drawing area (zoom, roaming and so on).

Roaming over a drawing area

The **Roaming** button or **[View] [Roaming Cursor]** allows panning of the visualized area over the whole project area, maintaining the selected scale. In panning mode, the cursor appearance changes to a Cross (+). Click on a point over the canvas and drag it to a desired position. The point will move to the new position while the rest of the visualized data will move accordingly. To disable the pan feature, just click on the mouse's right button or over the **Roaming** button again.

6. **Exit** from SPRING

To exit SPRING just click on **[File] [Exit]** menu or use the usual way to close MS-Windows applications and then confirm you really want to close it. All changed data will be already saved that time.

Class 2 – Reading Images

1. Digital Image

Radiance is the radiant flux coming from a source target in a certain direction per unit area.

A digital image is defined by a two-dimensional function of the reflected radiance I(x,y) of a scene. Image intensity at each location (x,y) is described by I, usually called **gray level,** which is represented by a **finite non-negative integer**. Each point imaged by the sensor corresponds to a small area named 'pixel' (picture cell), which should have a geographic location assigned to it. Each pixel is digitally coded according to the reflected energy on the target for different bands of the electromagnetic spectrum.

2. Image Characteristics

An image can be represented by a 2D matrix in which rows and columns define pixel locations. A finite number of bits is then used to represent scene radiance for each pixel.

The continuous radiance of a scene is represented by discrete gray levels which are given by specific number of bits per pixel. Modern sensors usually provide images coded using 8 or 10 bits (256 or 1024 digital levels).

3. Bands and Resolution

SPRING directly reads Landsat, SPOT, NOAA, and ERS-1 images. Each type of image has its own **resolution**. Analog images like paper photographs can also be handled by importing their corresponding digital versions (TIFF or RAW).

Radiometric resolution defines the sensor's ability to distinguish spectrally similar or spatially near responses. It applies to the spatial, spectral and radiometric character of imagery.

Spatial Resolution: it is the smaller angular or linear separation between two objects. A resolution of 20 meters requires the objects to be separated 20 meters at least in order to be discriminated by the system.

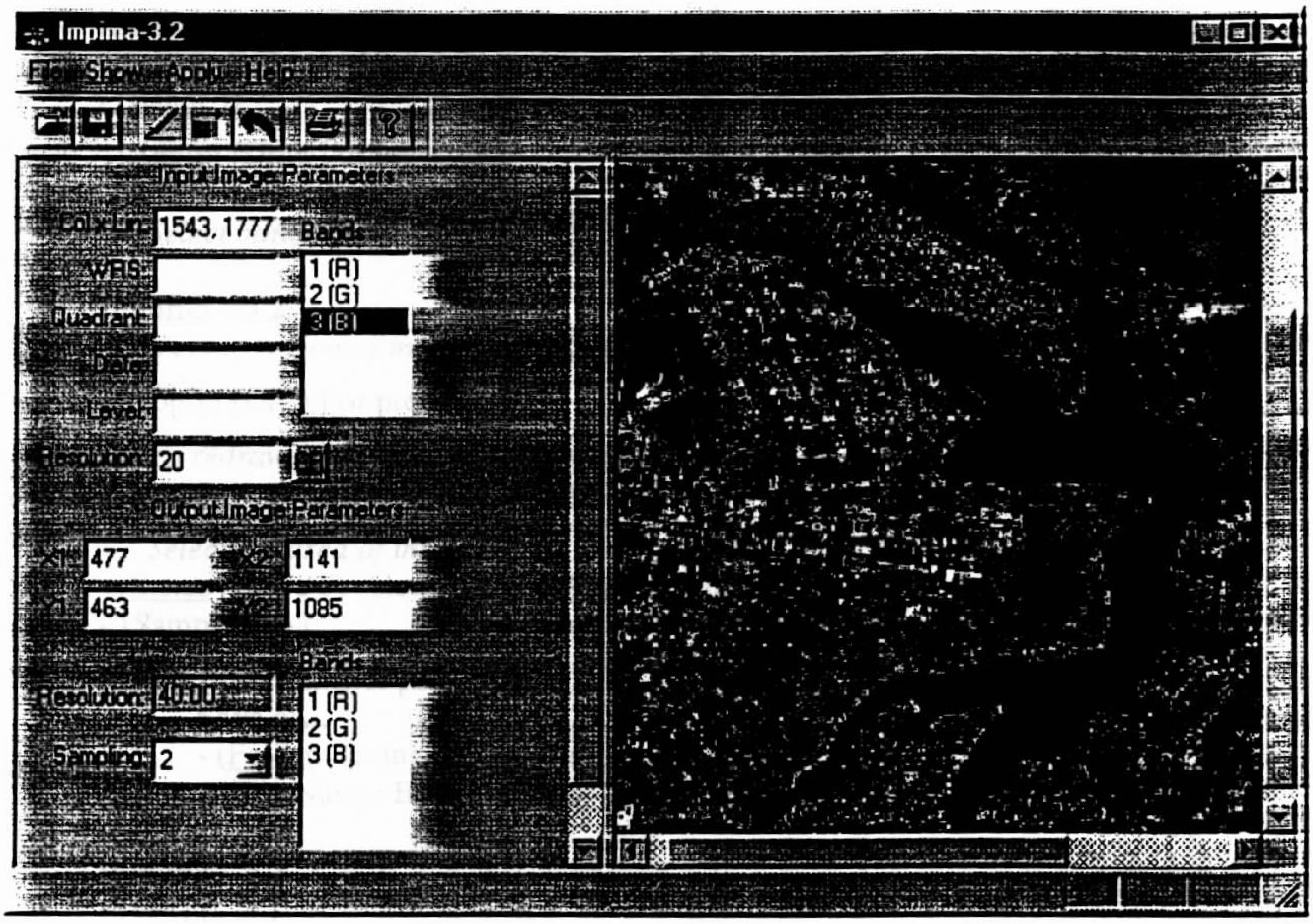
Spectral Resolution: it represents a measure of the width of the spectral bands of a sensor. A sensor that operates from 0.4 to 0.45 *u*m has a smaller spectral resolution than another one operating from 0.4 to 0.5 *u*m

Radiometric Resolution: it is associated to the ability of the sensor to distinguish between two intensity levels of the returning signal. A resolution of 10 bits is better than one of 8 bits.

The **Impima** Spring module reads digital images and convert them to the Spring native GRIB format. The output from Impima are GRIB files consisting of files that describe image data characteristics.

4. Image Reading and Conversion (Windows)

⇒ *Initializing IMPIMA*
- # Spring 3.2- *Impima*
* The figure shows the IMPIMA interface.



Col x Lin: 1543, 1777
WRS:
Quadrant:
Date:
Level:
Resolution: 20
Bands: 1 (R) 2 (G) 3 (B)

X1: 477 X2: 1141
Y1: 463 Y2: 1085
Resolution: 40.00
Sampling: 2
Bands: 1 (R) 2 (G) 3 (B)

Next section will describe the steps to read an image from CD-ROM and the item 4.2 describe the steps to convert a TIFF image to GRIB (*.grb files)

4.1. Reading Image from CD-ROM

⇒ Selecting and visualizing a image from CDROM:
Impima – 3.2

- [File] [Open...] or press
  Open
  ** The CDRom drive must be available*
  - (Type of File : | Directory of volume (vold*.dat))
  - (Folder | 221/071X)
  - (Folder | 940607)
  - (File | Voldir.dat)
  - (Open)

Impima – 3.2
*Observe the **Input Image Parameters***
- (Band| 5)
- [Apply] [Draw] or press

⇒ ***Select bands and region of interest***

Impima – 3.2
- *Select the region of interest with area cursor over the image*
- [Apply] [Draw] or press
- *To redraw image press*
- *To enhance image contrast press*
- *Select the area of interest.*
- (Bands | 3, 4 e 5) select 3,4,5 in ***Output Image Parameters***
- {Sampling: 1}
- [File] [Save as...] or press
  Save as...
  - (Folder | springdb\Imagens)
  - {File Name: Brasilia.grb}
  - (Save)

Impima – 3.2
- [File] [Exit]

4.2. Converting (TIFF) Image to GRIB Image

⇒ ***Selecting and Showing TIFF image***

Impima – 3.2
- [File] [Open...] or press
  Open
  - (File of Type: | file TIFF (*.tif))
  - (Folder | springdb\Imagens)
  - (File | brasi.tif)
  - (Open)

Impima – 3.2
* *Observe the **Input Image parameters**.*
- (Bands | 3)
- {Resolution: 30}
- [Apply] [Draw] or press

⇒ ***Select bands and region of interest to be saved***

Impima – 3.2
- *Select region over the image using the area cursor*
- [Apply] [Draw] or press
- *To redraw image press*
- *To enhance the image contrast press*

- (Bands| 1, 2 e 3) in ***Output Image parameters***
- {Sampling: 1}
- [File] [Save as...] or press 💾

    Save as

    - (Folder | springdb\Imagens)
    - {File Name: Brasilia.grb}
    - (Save)

Impima – 3.2

- [File] [Exit]

NOTE: After reading or converting the image format to the grib format (*.grb) you can import this image to a Spring project. If you want to georeference it you must to take some control points and after that to import it as we will see in the next class.

Class 3 –Image Registration/Georeferencing

1. Image Registration/Georeferencing

Registration is a mathematical transformation that relates the image coordinates to a map or another registered image. If the projection system is specified in the interface the process is called *georeferencing*, i.e., a special kind of registration.

Georeferencing is a mathematics transformation that relates the image coordinates to a map coordinate system (lat/long).

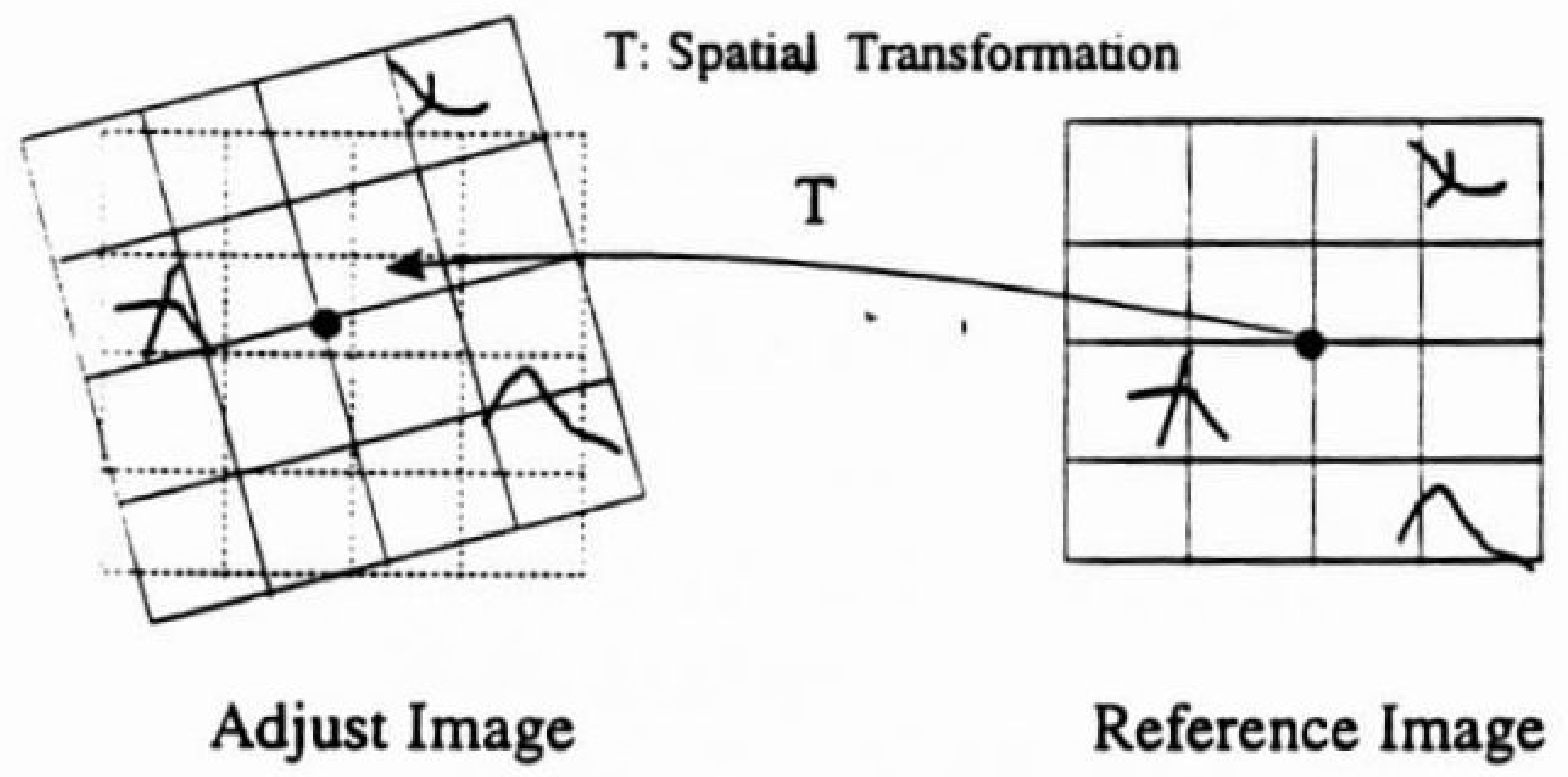
This transformation eliminates image distortions generated mainly by sensor system and platform movements. Typical distortions are caused by: perspective, earth rotation, earth curvature, platform motion (changes in roll, pitch, yaw, altitude and velocity), scale changes due the topography, and so forth.

Why do you need the registering?

- To integrate images of different sensor system.
- To integrate images of different times. Temporal analysis.
- To integrate images of different position. (To get tri-dimensional information.)
- To do a "mosaic" of images

What do you need for registering?

1. **Control Points** – spatially small features on the map that can also be identified on the image. These features could be road intersections, airport runway intersections, bends in river etc. The control points (as pairs – on the map and image) can be used to estimate the polynomial coefficients (6 for first order) by substitution into the mapping polynomials to yield sets of equations in those unknowns.
2. **Mapping function** – resample function, typically a first or second polynomial function to define a new pixel from some neighbors pixels
3. **Interpolation process** – Nearest-neighbor, Bilinear or Cubic interpolation.



First Order

$xr = a_1x + b_1y + d_1$

$yr = a_2x + b_2y + d_2$

Second order

$xr = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$

$yr = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2$

$Ncp = (N+1)(N+2)/2$

$N = 1$ => 3 pts

$N = 2$ => 6 pts

Using SPRING , you can get the **control points** in three ways: using a map placed on a digitizer table (TABLE); using another georeferenced infolayer (CANVAS) or giving the coordinates by keyboard (KEYBOARD).

KEYBOARD : The Control Points are informed in plan or geographic coordinates directly from keyboard. You can take them from a topographic map or GPS. It is not necessary to activate a *project*. In this case the system will ask you for the projection system to be used in this georeferencing.

DIGITIZER TABLE : It is necessary to have a map (cartographic map) that cover the image area. This map must be put on a digitizer table. It is not necessary to activate a *project*. In this case the system will ask you for the projection system to be used in this georeferencing.

CANVAS : In this case you have to activate one *project* and use one infolayer as a reference. This InfoLayer must be already georeferenced and must cover the image area, and can be a *Image* or a *thematic* map (for example, road map or river map), where you can see features of the image (control points).

2. Image Georeferencing

This section describe the steps in georeferencing the Image read in the last class (Brasilia.grb), using a topographic map placed in a digitizing table (item2.1). If the table is not available, control points can be taken by using the keyboard, or another existing and georeferenced image or a thematic map..

Initially you have to create a new *database* and a *category* of image model in this database.

⇒ ***Creating a Data Base:***

- *# Spring(*)
- [File] [DataBase...] or button

Data Base

- (Directory) {C:\springdb} – usually all databases are put under *springdb* directory. If this directory do no exist select another one or create it.
- {Name: TEST} – or other name (maximum of 32 characters).

- (Create)
- (Activate)
  - * *if there is other active database and/or project active, say Yes to close them.*

⇒ ***Creating Categories:***

- *# Spring( )*
- [File] [Data Model...] or button

Data Model

- {Category - Name: Image_TM}
- (Model ⇔ Image)
- (Category - Create)
- (Apply) (Close)

2.1. Acquiring control points using the keyboard

- [File][Image Registration...]

** If you don't have an active project, the projection system will be required, otherwise the project's projection will be considered.*

Projections

- (Systems | UTM)
- (Earth Model | SAD69)
- {Origin- Long: w 45 00 00}
- (Apply)

- [File][Image Registration...]

**Two interfaces will be showed: Image Registration and Image Selection.*

Image Selection

- (Directory: (C:\springdb\Imagens) (CR)
- (File | Brasilia.grb)
- (Select)

*You must to select the CANVAS 5 in a *Control Panel* interface.

Control Panel

CANVAS CONTROL
(Show) (5)

**The CANVAS 5 is reserved to make a registration.*
All bands of this image will be showed below. Select one band.

- (Bands | 5) (select band 5 or other one)
- (M) (monochromatic)

SPRING - CANVAS 5 (visualize this band on CANVAS 5)

[Execute] [Draw] or button - the band 5 will be visualized on CANVAS 5.

** You can apply a contrast in this image.*

Select Image

- (Contrast...)

Contrast (CANVAS 5)

- Click and drag the mouse on (M) channel.
- (Apply)
- (Close)

*Apply the contrast again if it is necessary.

Select Image

- (Close)

⇒ ***Getting the control points by Keyboard:***

Image Registration

- (Acquisition ⇔ Keyboard)
- (Operation ⇔ Create)
- {Name: pc1} – control point 1 (or use the names in the table below)
- (CR)
- (OK)- *in a message "Digit reference coordinates!"*

- (Plane)
- {X(m): 189127.2}, {Y(m): 8260695.1}
- (CR)

* *A green point will show up in the image.*

- Click and Drag the green point to the right position in the image according to the figures below.
- (Save) – PS: Save it although there is not enough control point.
- Repeat this process for at least 6 control points.
- *It will be better if you take these points distributed in the image.*

CPs	Name	X (meters)	Y (meters)
1	trevo_torto	189127.2	8260695.1
2	ponte_asanorte	189773.2	8258734.3
3	trevo_cpdex	186106.4	8253422.9
4	trv_zoologico	185944.6	8246038.4
5	pte_gsalomao	189544.2	8246578.1
6	lago_jaburu	196520.2	8251208.2
7	barragem	201670.7	8251500.6
8	ilha	197639.3	8255736.0
9	trv_nordeste	201244.2	8260185.0
10	trv_unb	192014.9	8252928.3

CPs	Name	X (meters)	Y (meters)
11	trv_cemiterio	185134.9	8249891.9
12	trv_esfazenda	197860.9	8244118.9
13	trv_esfazenda2	198175.0	8245925.3

The above figures present the control points position in the image and a zoom of each position.

2.2. Acquiring the points using a Digitizing table:

In order to get the control points through the cartographic map placed in a digitizing table, you have to configure and calibrate the table .

=> *Preparing Digitizer Table*

- *# Spring(⊞)*

[Tools][Environment Configuration...] on the major menu of "**Spring**" to configure the model and kind of digitizer table. (Ex: Digigraf, Vangogh)

PS: See the details of configuration digitizer on configuration manual or in help on-line.

=> *Calibrating the Digitizer table:*

- *# Spring(⊞)*
- [Tools][Table Calibrating]
    Calibration
    - (Projection...)
        Projection
        - (Systems I UTM)
        - (Models I SAD69)
        - {Origin - Long: o 45 0 0}
        - (Apply)
    Calibration
    - (Coordinates ⇔ Plane)
    - (Point 1)
    - *Click a known point 1 on the map over the table*
    {X(m): XXXXX}, {Y(m): YYYY} – *PS: XXXX,YYYY is the point coordinates. Now, enter the real plane coordinates of this point.*
    - *Repeat for points 2, 3 e 4 (known points)*
    - (Apply)
    - (Test)
    - Click on the map a known point and compare the real coordinates with the coordinates presented by the system. *Redo the calibration if the coordinates of the point of test presented by the system is not right.*
    - (Close)

⇒ Select the image to be georeferenced and apply a contrast:

- [File][Image Registration...]

** Two interfaces will be showed Image Registration and Image Selection.*

Image Selection

- (Directory: (C:\springdb\Imagens) CR
- (File | Brasilia.grb)
- (Select)

*You must to select the CANVAS 5 in a *Control Panel* interface.

Control Panel

CANVAS CONTROL (box)
(Show) (5)

**The CANVAS 5 is reserved to make a registration.*
All bands of this image will be showed below . Select one band.

- (Bands | 3) (select band 3 or other one)
- (M) (monochromatic)

SPRING - CANVAS 5 (visualize this band on CANVAS 5)

[Execute] [Draw] or button ⊿ - the band 3 will be visualized on CANVAS 5.

** You can apply a contrast in this image.*

Select Image

- (Contrast...)

Contrast (CANVAS 5)

- Click and drag the mouse on (M) channel.
- (Apply)
- (Close)

*Apply the contrast again if it is necessary.

Select Image

- (Close)

⇒ Acquiring Control Points using Digitizer table:

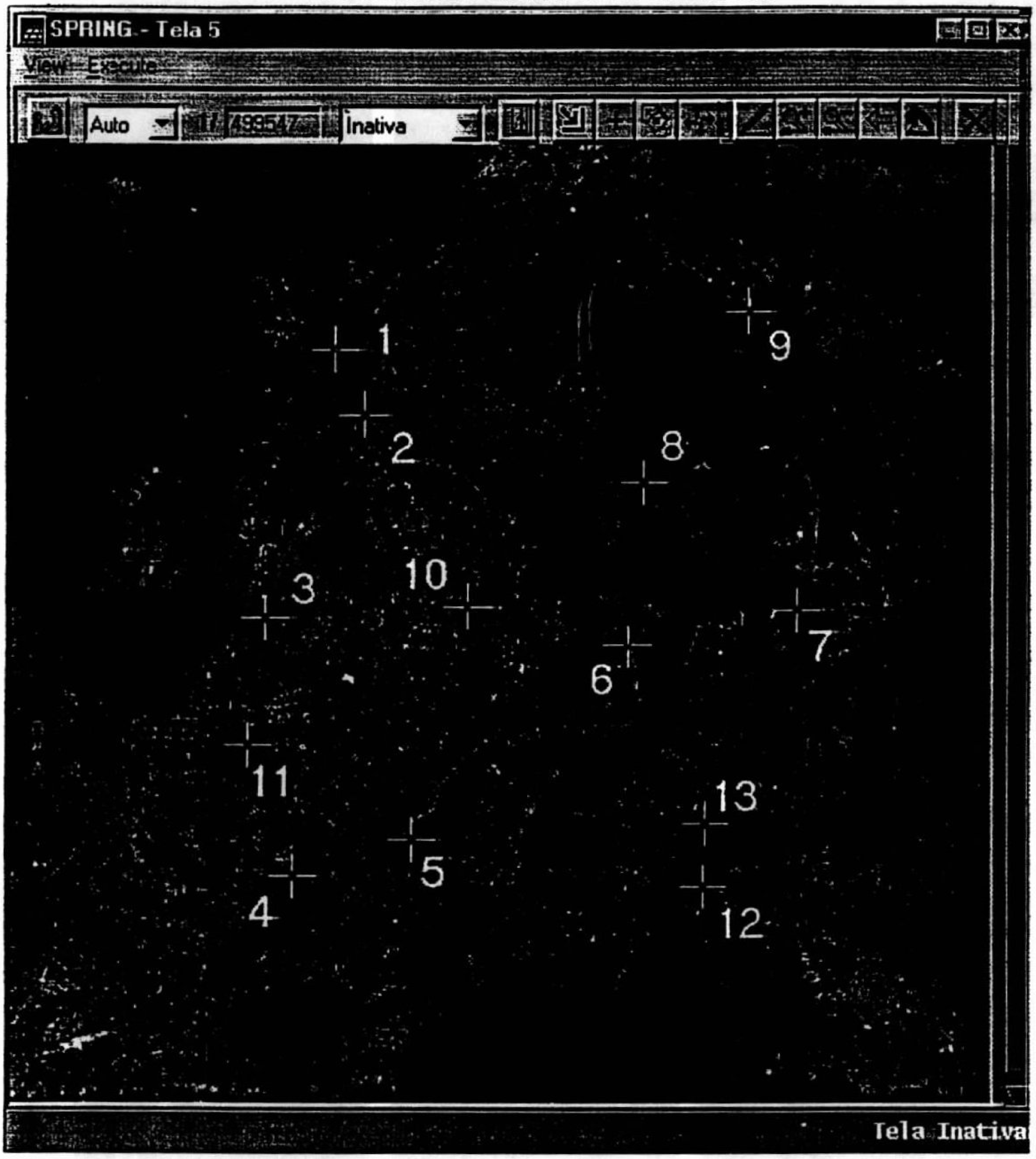
Image Registration

- (Acquisition ⇔ Digitizer)
- (Operation ⇔ Create)
- {Name: pc1}
- (CR)
- (Ok) – click OK on message asking you to enter the coordinates of the pc1.
- Select the control point on the map placed in a digitizer table. Click the button 1 in a digitizer mouse over this point.

** A green point will show up in the image.*

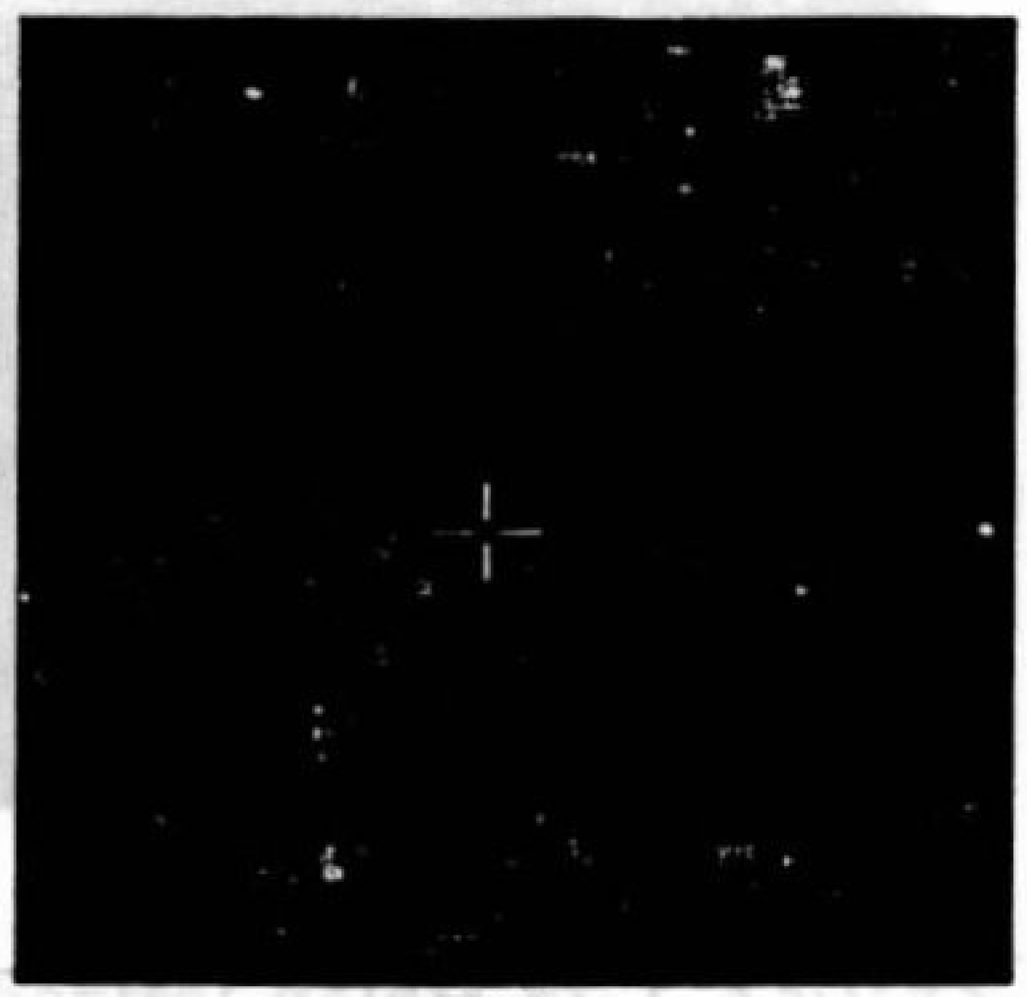
- Click and Drag the green point to the right position in the image.
- Repeat this process for at least 6 control points.

CONTROL POINTS IN A IMAGEM



CP1

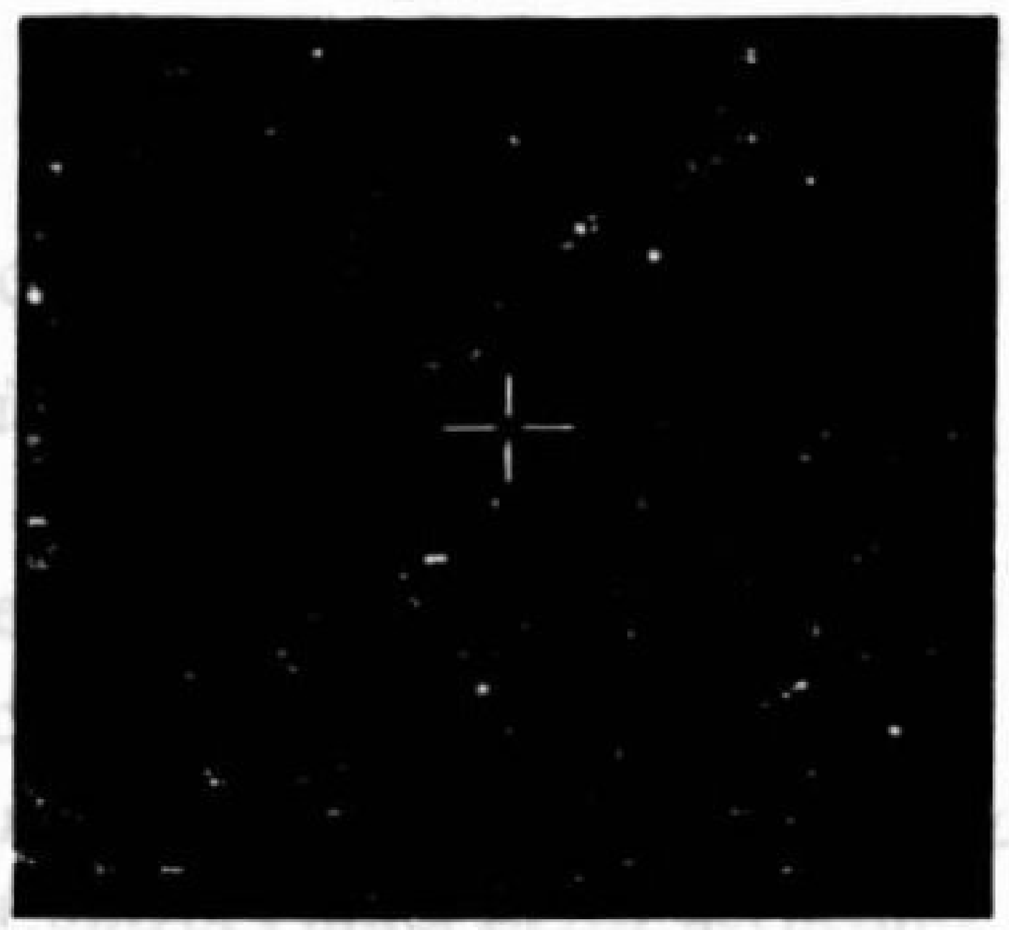
CP2



CP3

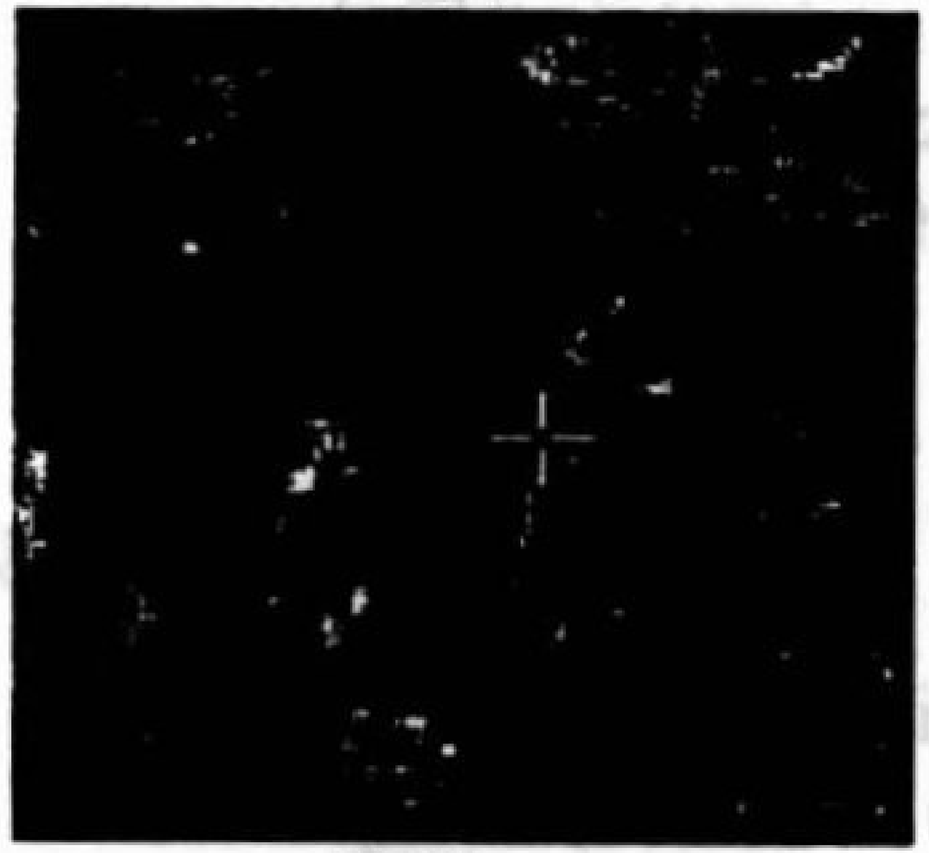
CP4

CP5

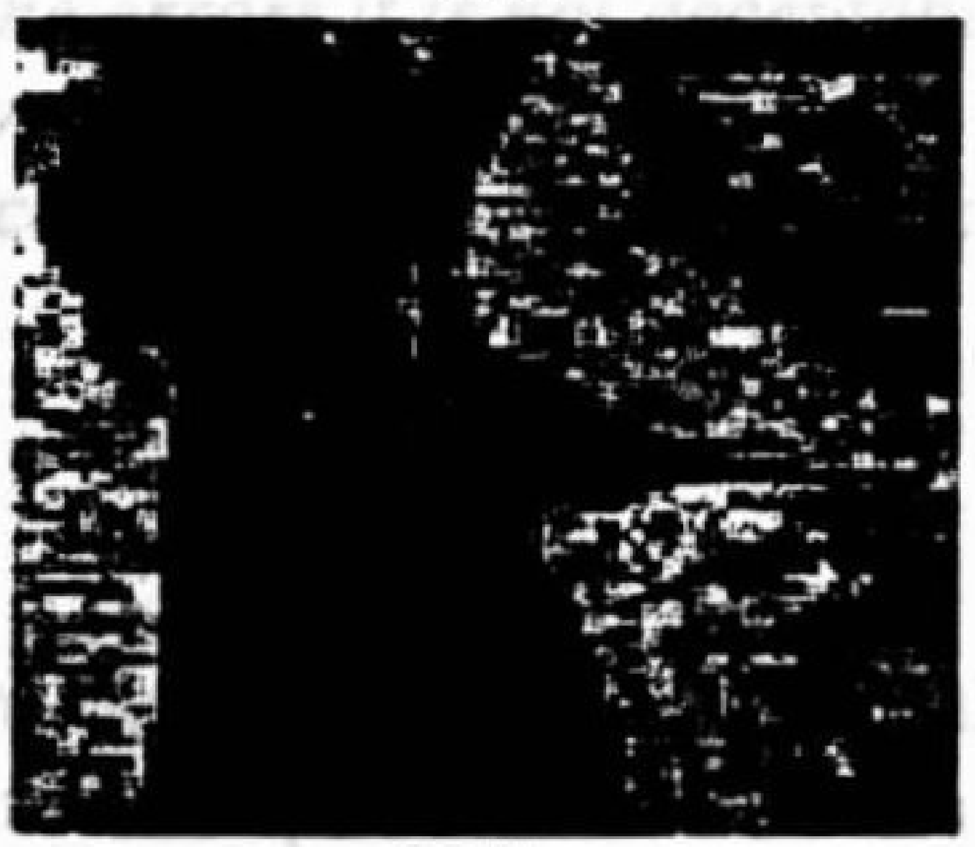


CP6

CP7



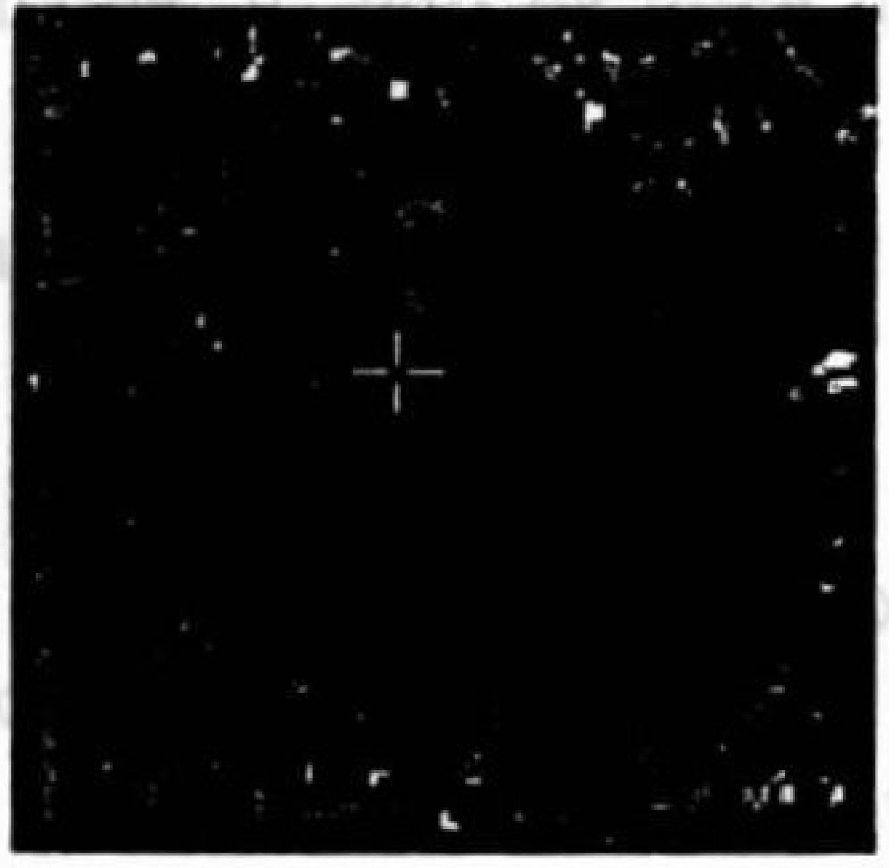
CP8



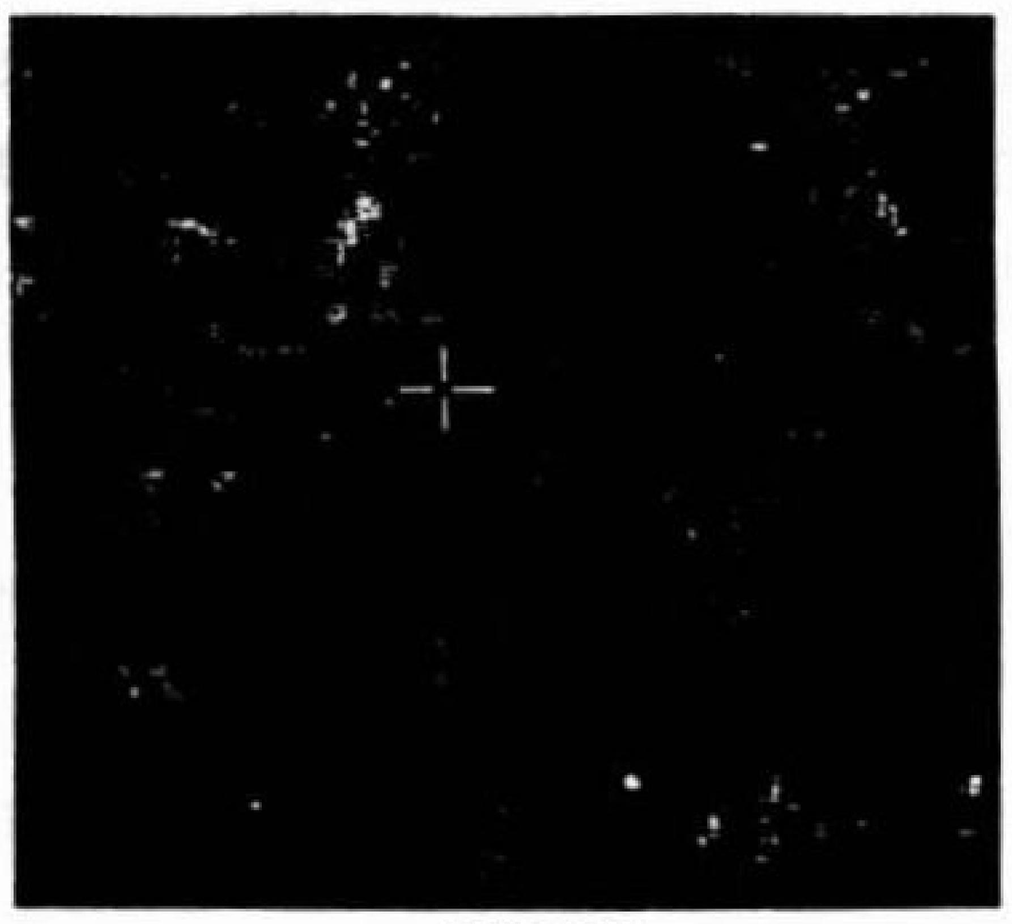
CP9

CP10

CP11



CP12



CP13

PS : In order to acquire control points by CANVAS, it is necessary to activate a project and a infolayer which will be used as a reference to extract the control points.

2.3. Selecting some control points to register the image

⇒ ***Defining mapping function***

Image Registration

- (Polynomial Degree ⇔ 1)

- (Operation ⇔ Select)

- *Select at least 6 control points and see the errors in:*

Test Points Error and Control Points Error according to the NOTES below.

- *Edit points when the errors it is not acceptable (use all the Operations: Create, Delete, Move, Rename, Select)*

- (Save) - save *the combination of points which the minimum error.*

- (Close)

NOTES:

- After selection of the third point, the system start calculating the errors associated to each point (the error units is "pixel" resolution in x and in y direction) and will put them in front of its name. It will be done for points not selected too (**Test Points**).
- It is advised to use the maximum points acquired if they are well distributed over the image.
- The user must check the value Control Point Errors to control precision. Example: in a urban area it is acceptable a error around 0.5 pixel if the resolution is 30 meters. In forest areas, it is acceptable a error around 3 pixels for the same resolution, because it is sometimes more difficult to get control points and the variation in a forest area is not like in urban area.
- Other parameter used to check the georeference error is the SCALE. The acceptable error is defined by 0,5mm of the scale. As a example, if the scale used in the mapping is 1:50.000, the acceptable error is 25 meters (0,5mm x 50.000m) so that for 10 meters image resolution, 2 pixels of error (20 m) would be acceptable (it is less than 25m).

- You can repeat the selection of control points until you get the best combination, with acceptable control points error value. You can test other points, other polynomial degree and move, delete or create points in order to get the acceptable error value.

2.4. Importing the georeferenced IMAGE as an Information Layer to SPRING

It is necessary to define a project in SPRING that cover the image area.

⇒ ***Defining a Project:***
SPRING-3.2

- [File] [Project...] or button
    Projects
    - {Name: DFederal} (maximum of 32 characters)
    - (Projection...)
        Projections
        - (<u>System</u> I UTM)
        - (<u>Earth Model</u> I SAD 69)
        - {Origin - Long: o 45 0 0}
        - (Apply) (this secondary window will be closed)
        Ps: *The most of secondary interface-window is closed after press the APPLY button, but there is some of them which you have to press the button CLOSE.*

    Projects
    - (Coordinates ⇔ Geographical)
    - {Long1: o 47 57 30}, {Long2: o 47 47 00}
    - {Lat1: s 15 52 30}, {Lat2: s 15 41 55}
    *Long1/Lat1 is the lower left corner of the bounding box.
    *Long2 and Lat2 is the up right of the bounding box.
    - (Create)
    - (Activate)

⇒ ***Importing the Image (georeferenced) to the Project:***
SPRING-3.2

- [File][Import GRIB files...]
    Import Images
    - (Directory...: (C:\springdb\Imagens)
    - (Files I Brasilia.grb)
    - (Images I <select the band 3>)
    - (Category...)
    *Select the category of this data (in this case the category must be defined in a Data Model as a Image Model)
        Category List
        - (<u>Categories</u> I Image_TM)
        - (Apply)

Import Images

- {Info Layer: **TM3**} (it can be the same of the image band)

- (Apply) **Observe that this interface is not closed after to press Apply.* **Repeat for bands 4 and 5, giving the infolayer name* **TM4** *and* **TM5** respectively.

NOTE:

- After registration (geo-reference in this case because you defined the projection system) and importing the image (bands 3,4,5), observe that the infolayer TM3, TM4 and TM5 are showed in the list of infolayers at the interface **"Control Panel"**.
- If one image is not enough to cover the project area of interest, it is available the Mosaic option. You can get the adjacent images and do the same process, i.e., you can read, georeference (using the same projection system) and import the adjacent images, but now click in the Mosaic option.

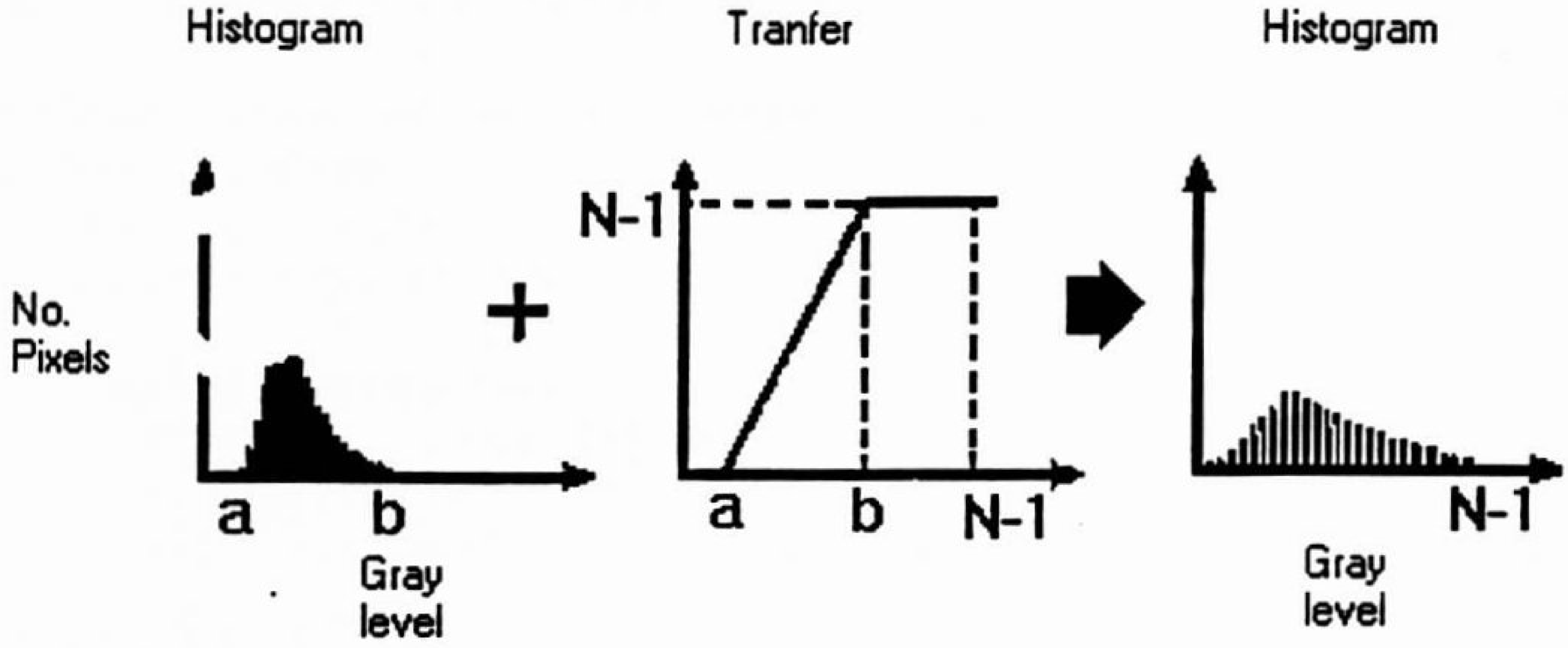
CLASS 4 – Image Processing

1. Image Enhancement

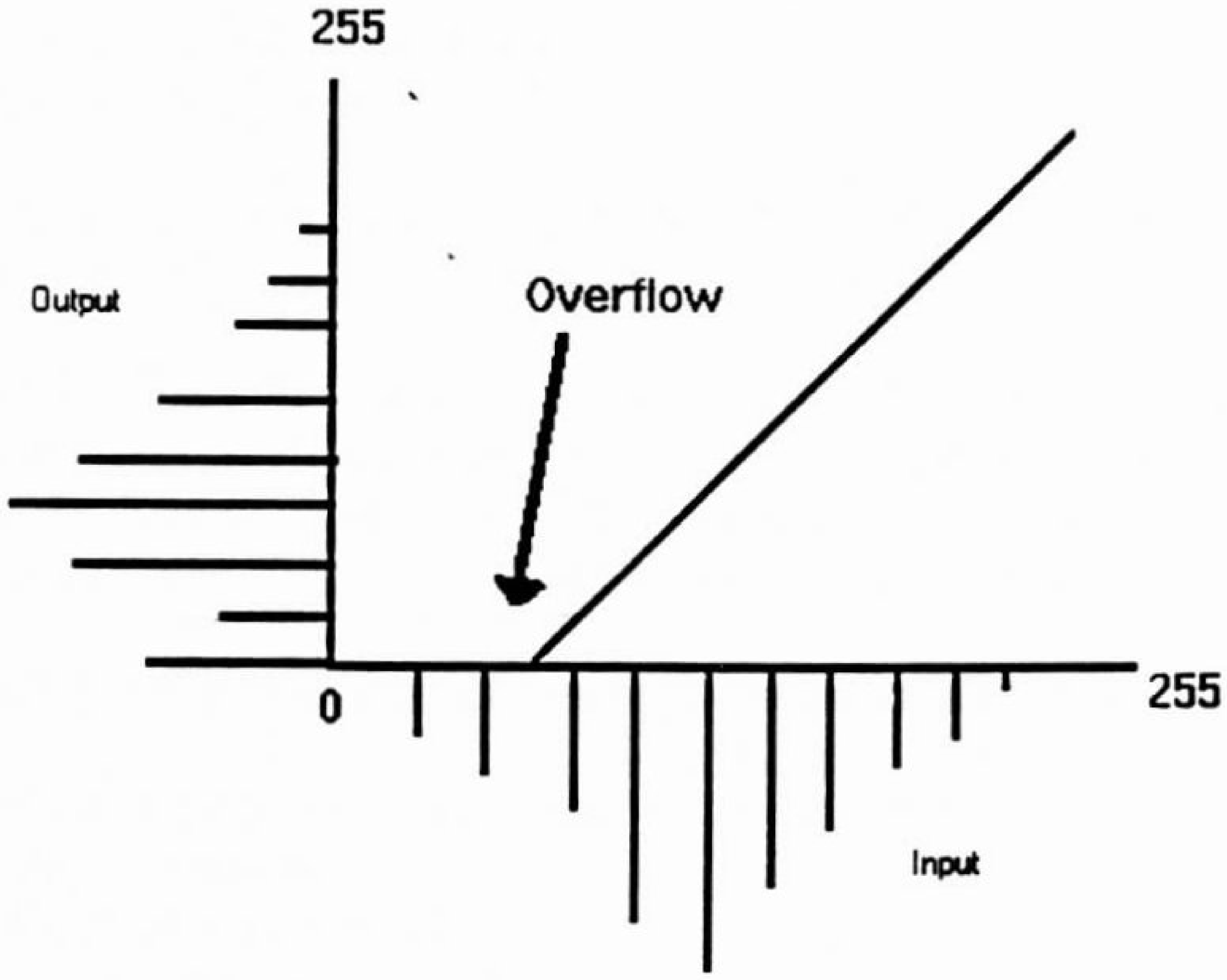
Image enhancement process consists in improving the visual appearance of an image, or its conversion to a better suited form to analysis by the human eye or a machine. It is usually used as a pre-processing stage in pattern recognition systems.

The **contrast manipulation** technique is a punctual operation that applies a radiometric transfer function to each pixel of an image in order to increase the discrimination of the objects in the image.

The **histogram** of an image describes its statistic distribution of pixels per gray scale levels. One means of enhancing images is the technique of histogram modification in which the original image is rescaled by some transfer function so that the histogram of the enhanced image follows some desired form.



Usually the enhance in the original image is related to the inclination of the curve being applied to it. A 45° line slope function will result in no enhancement or compression of the gray levels in the image.
An overflow occurs when an amount of pixels with different gray levels are mapped to the same level. A linear transfer function with a very low slope degree may cause an overflow in the enhanced image. One should be careful about overflow of pixels in a level, whenever it is not a desired result, because it means lost of information. See figure bellow.



1.1. One band image enhancing

⇒ ***Defining a linear contrast enhancement:***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

Control Panel window

- (<u>Categories</u> I Image_TM)
- (<u>Layers</u> I TM4)
- (M) – stands for monochromatic image

or [Execute][Draw]

[Image][Contrast...]

Contrast window

- (Operation ⇔ Linear)
- *Select, by left button mouse clicking, the minimum value for an interval of gray levels;*
- *Select, by right button mouse clicking, the maximum gray level value for that interval.*

(Apply)

**The mapping of the above defined interval into [0-255] define a linear transfer function to be composed with the original histogram to generates a new one: that of a new enhanced image*

**The enhancing effect is visualized in the active canvas*

⇒ ***Saving the enhanced image:***

- {Name: TM4_realce_linear}

- (Save Image ⇔ Band)

(Execute) (Save)

**You may save only part of the area by defining a rectangular region over the image with the area cursor, before saving.*

NOTE:: Once an enhancing option is applied to a canvas it keeps in effect even after the contrast interface window is closed, so that for a new visualized image in the same enhancing option will be applied. The (Execute)(Reset) option should be explicitly set before the visualization of a new image.

1.2. Multi Spectral Enhancement (RGB color components)

⇒ ***Visualizing a three color components composition:***

Spring3.2 window

Control Panel window

- (Categories | Image_TM)
- (Layers | TM4), (R)
- (Layers | TM5), (G)
- (Layers | TM3), (B)

◿ or [Execute][Draw]

⇒ ***Defining enhancement options to each band:***

[Image][Contrast...]

Contrast window

- (Operation ⇔ Linear)
- (Band ⇔ **Red**)
- *Left button mouse clicking select the minimum value for a of gray level interval*
- *Right button mouse clicking select the maximum gray level value for that interval.*

(Apply)

Repeat for **green and **blue** channel images*

⇒ ***Saving a synthetic color image from the three components:***

- (Save Image ⇔ Synthetic)
- (Name: comp_453)

(Execute)(Save)

**The synthetic image based on the selected bands and contrast options is generated as a new infolayer entry available in the Control Panel*

Analyze also* **MinMax, Square Root, Square, Logarithmic, Negative, Histogram Equalization *enhancing options. The* **Editing** *operations:* **Remove, Add** *and* **Move** *applies to points in a piecewise linear function graph in order to adjust it to desired enhancement conditions.*

2. Reading Pixel Values

This option allows the reading of gray level values of pixels in the neighborhoods of a certain pixel. It is useful in studies of the spectral reflectance characteristic behavior of known objects in a scene relative to bands in multispectral images.

⇒ ***Reading mouse selected pixel values:***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

  Control Panel window

  - *Select category* **Image_TM**
  - *Select image infolayer* **TM5**

    Canvas (1 to 5)

    - *Visualize selected image* **TM5**

[Image][Pixel Value]

- *Select a pixel of the image with a click*

  Pixel Values window

  - *Analyze showed pixel values for different targets*

⇒ ***Reading pixel values given by coordinates:***

Pixel Values window

(Set Pixel Position...)

Set Cursor Position window

- (Coordinates ⇔ Plane)
- {X:196219.3}, {Y:8254121.6}

(Apply)

*Note a green "cross" showing the pixel position according to the coordinate.

⇒ ***Saving Pixel Values***

Pixel Values window

(Save...)

Save As window

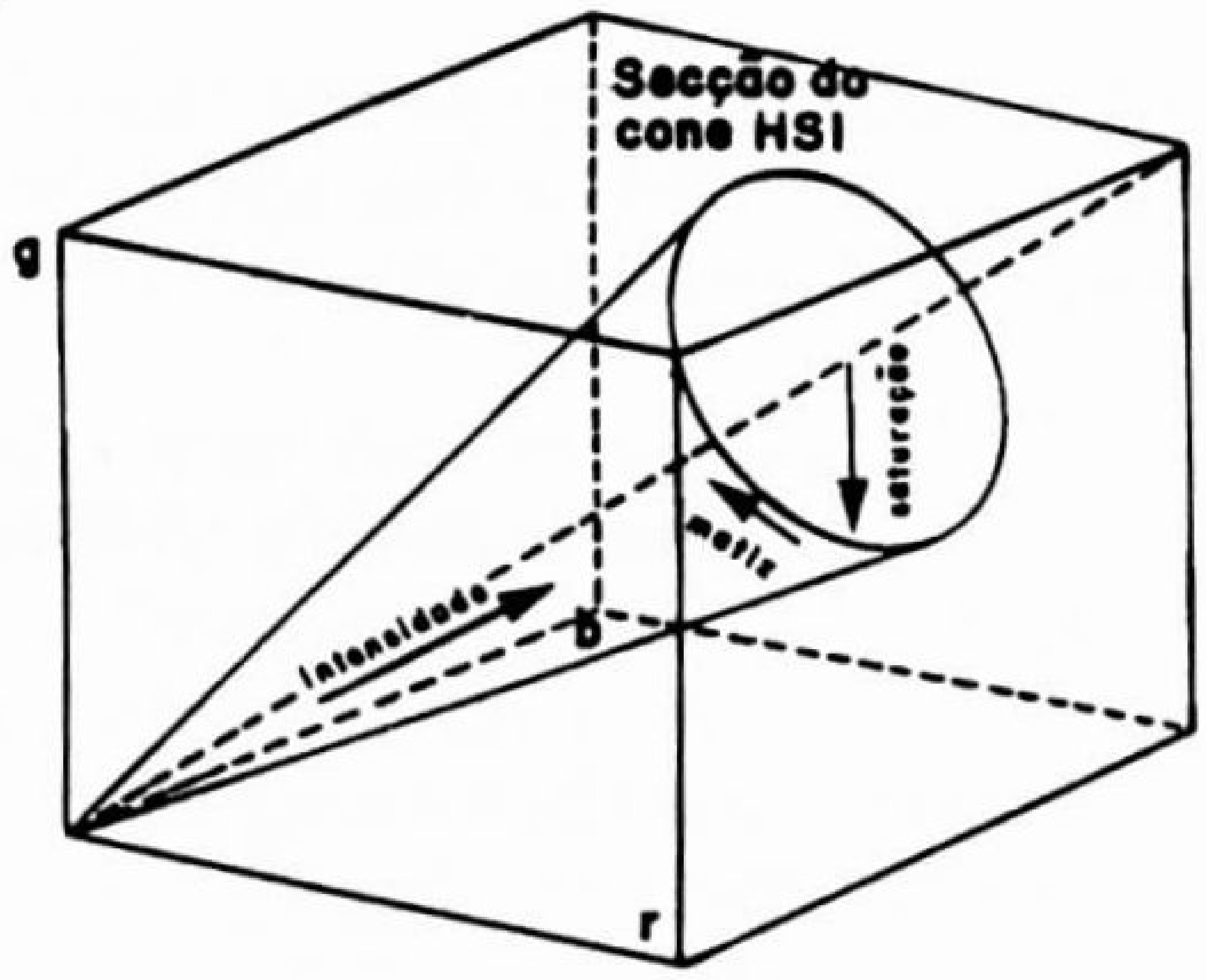
- Select a file name to store the values

NOTE: The same procedure for pixel value reading can be done over a color image resulting from 3 bands, but not over a synthetic image.

3. IHS Color System

The IHS system has been used within the image processing community as a quantitative means of specifying the **intensity, hue** and **saturation** of a color. These parameters are more significant to the human eye than the proportion of red, green and blue in the image.

The **intensity** (I) measures the total energy in all the wavelengths of the light. It defines the brightness perception when seen by the human eye.
The **hue** (H) measures the average wavelength of the light being reflected or emitted. It defines the color of the object.
The **saturation** (S) measures the range of wavelength around the average in which the light is being reflected or emitted. It defines the pureness of the color.
The IHS coordinate system can be graphically represented by a cone. It is spatially related to the RGB system according to the figure bellow:



The distance of a point to the cone origin represents the intensity. The radial distance of the point to the cone central axis represents the saturation. The hue is represented by a radial circumference with radius equal the saturation around the intensity axis. These three parameters are independent and can be analyzed and changed separately in order to achieve a better adjust of the colors to the human visual system. In many cases it is used to produce integrated images using different sensors or geophysical data.

⇒ ***RGB-IHS Transformation:***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

  Control Panel window

  - *Select category* **Image_TM**
  - *Select image infolayer* **TM5_Amostra**

    Canvas (1 to 5)

    - *Visualize selected image* **TM5_Amostra**

[Image][IHS<->RGB Transform...]

IHS Transform window

- (Transformation ⇔ RGB->IHS)
- (Input Images ⇔ R) //replace by "channels"

  Categories and Layers window

    - (Categories | Image_TM)
    - (Infolayers | TM5_Amostra)

  (Apply)

  **Repeat for channel G with infolayer* TM4_Amostra *and channel B with* TM3_Amostra ·

- (Output Layer: Tihs)
- (Output Resolution ⇔ R | G | B) = {X: 30}, {Y:30}, the same resolution as that of input infolayers.

(Apply)

To gain resolution on the output components of the IHS->RGB transformation use the **Panchromatic** SPOT band (Category Image_Spot) as the I-component (intensity) Layer. Then the output resolution can be set to X=10 and Y=10.

⇒ ***IHS-RGB Transformation***

IHS Transform window

- (Transformation ⇔ IHS-> RGB)
- (Input Images ⇔ I) //replace by "channels" in the interface

  Categories and Layers window
  - (Categories | Image_SPOT)
  - (Infolayers | PAN)

  (Apply)

- (Input Images ⇔ H) //replace by "channels" in the interface

  Categories and Layers window
  - (Categories | Image_TM)
  - (Infolayers | Tihs-H)

  (Apply)

- (Input Images ⇔ S) //replace by "channels" in the interface

  Categories and Layers window
  - (Categories | Image_TM)
  - (Infolayers | Tihs-S)

  (Apply)

- {Output Layer T: _inversa}
- (Output Resol ⇔ I) = {X: 10}, {Y:10}

(Apply)

**Compare the original color images and the transformed one.*

4. Filtering

Filtering techniques are local transformations that are characterized by operations over a neighborhood. The new value for a pixel is derived from the values of the pixel itself and of a set of surrounding pixels.

These filters are implemented as masks, defined and then moved over the image row by row and column by column. The products of the pixel value covered by the mask at a particular position and the mask entries are taken and summed to give the filter response.

The filters available in SPRING and its respective mask are shown in the table bellow. Additionally the user can also define new masks.

Filter	Option	Mask
Linear	Low-pass	3x3, 5x5 or 7x7
	Edge directional	NW, W, SW, N, S, NE, E or SE
	Edge non-directional	High, Middle or Low
	TM enhancement	-
	Mask	Mask editor.
Non-Linear	Edge Detection	Sobel or Roberts
	Morphological Dilatation	Mtot, Mx, M+, M-, Ml, Md or Me
	Morphological Median	
	Morphological - Erosion	
	Masks	Element Editor.
Radar		Define Type, Window (3x3, 5x5, 7x7, 9x9), Image and Number of Looks.
		Define Type, Window (3x3, 5x5, 7x7, 9x9), Image and Number of Looks.
		Definer o Type, Window (3x3, 5x5, 7x7, 9x9), Image, Number of Looks e Correlation Coefficient.

⇒ ***Using pre-defined filters:***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

  Control Panel window

  - *Select category* **Image_TM**
  - *Select image infolayer* **TM4**

    Canvas (1)

- *Visualize selected image* **TM4**

[Image][Filtering...]

Filtering window
- (Type ⇔ Linear)
- (Linear Filters ⇔ Low-Mid Pass)
- (3x3)
- {Image Output}
- {Name: tm4_average}
- (Bounding Box...)
- (Number of Iterations ⇔ 1)

(Apply)

*Visualize image **tm4_average** filtered to compare. Try using other predefined filter types

⇒ ***Editing new user defined filters***

Control Panel window
- *Select category* **Image_SPOT**
- *Select image infolayer* **PAN**

Canvas (1)
- *Visualize selected image* ~~**TM4**~~ PAN

Filtering window
- (Type ⇔ Linear)
- (Linear Filters ⇔ Mask)

(Select...)

Filters window
- (Directory: C:\springdb\Dados)
- (X ⇔ 7), (Y ⇔ 7)

(Create...)

Filter Edition window
- Name: urb}

**Provide the filter values*

- Provide the filter values

0	0	0	0	0	0	0
0	0	0	-2	0	0	0
0	0	1	-2	1	0	0
-2	-2	-2	25	-2	-2	-2
0	0	1	-2	1	0	0
0	0	0	-2	0	0	0
0	0	0	0	0	0	0

(Save)

(Apply)

- {Output Layer: **SPOT_m** }
- (Bounding Box...)
- *Select the filtering area over the image, if it is not informed the area the whole image will be filtered.*
- (Number of Iterations ⇔ 1)

(Apply)

*Visualize the image **SPOT_m** filtered to compare. Edit more masks and test

5. Noise Elimination

The filtering process is useful to remove non systematic noise, as can be seen in the following example. Note that some completely dark noise (value=0) and white (value=255) were created.

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Rondonia.**

  Control Panel window
  - *Select category* **Image_TM**
  - *Select image***TM5_Ruido~~A~~** B

    Canvas (1 to 5)
    - *Visualize selected image***TM5_Ruido~~A~~** B

    * Compare with the images **TM5_RuidoB** and **TM5_RuidoC.**

⇒ ***Filtering to remove noise***

[Image][Filtering...]

Filtering window

- (Type ⇔ Linear)
- (Linear Filters ⇔ Low-Mid Pass)
- (3x3)
- {Image Output}
- {Name TM5_NoNoiseA }
- (Bounding Box...)
- (Number of Iterations ⇔ 1)

(Apply)

*Test the effect of bigger mask and also of non-linear filters.

Noise (completely dark or clear points) can appear randomly distributed or in a systematic way (vertical or horizontal stripes). They can be caused by failures of the detector, limitations of the sensor electronic system etc.
It's elimination can be done by defining **Upper** and **Lower Thresholds** used to cut undesired values based in the neighboring pixel values.

Lower Threshold (LT) choice
A pixel is considered noise if its value is, at least, LT smaller than the values of its upper and lower neighbor pixels. In this case it will be replaced by the average between its neighbors.

Upper Threshold (UT) choice
A pixel is considered noise if its value is, at least, LT greater than the values of its upper and lower neighbor pixels. In this case it will be replaced by the average between its neighbors.

⇒ ***Noise removal***

Spring3.2 window
- *Activate database* **Curso**
- *Activate project* **Rondonia.**

  Control Panel window
  - *Select category* **Image_TM**
  - *Select image* **tm5_ruido**

    Canvas (1 to 5)
    - *Visualize selected image* **tm5_ruido**

[Image][Noise Elimination...]

Noise Elimination window
- (Lower ⇔ 65), (Upper ⇔ 40)
- (Image Output}
- (Name TM5_NoNoiseB)

**Select the noisy area over the image*
(Apply)

Visualize the image* **TM5_NoNoiseB *in another* **Canvas** *to check If noise has been removed or not. Ttry different upper and lower thresholds.*

6. Image Statistics

The statistical analysis function is applied on samples over the image, its goal is to calculate and show some statistical parameters of selected images. The parameters are: Moments, Median, Moda, Standard Deviation, Covariance and Correlation Matrix, Auto-Correlation Matrix and Cross-Correlation Matrix.

⇒ ***Executing Statistical Analysis:***

Spring3.2 window
- *Activate database* **Curso**
- *Activate project* **Brasilia.**

  Control Panel window
  - *Select category* **Image_TM**
  - *Select image* **TM5**

    Canvas (1 to 5)
    - *Visualize selected image* **TM5**

[Image][Statistics...]

Sample Statistical Analysis window

- {Name: estat1}
- (Layers | TM3, TM4, TM5)

[Create File...]

Sample Acquisition

- {Sample Name: sample1}
- (Acquisition ⇔ Cursor)
- *Acquire a sample over the image defining a rectangle over the image, same as for zoom operation.*

*Note that a "yellow rectangle" defines the sample area.
(Acquire)

⇒ ***Analyzing the sample moments***

- (Calculations ⇔ Moments)

(Calculate and Present...)

Sample Statistics window

**Analyze the histogram and the statistical values (mean, variance, standard deviation and moments)*

**Select different layers and analyze the statistical values*

⇒ ***Analyzing sample moments***

- (Calculations ⇔ Covariance and Correlation Matrices)

(Calculate and Present...)

Sample Statistics window

*Analyze the matrices. Analyze different samples and parameters

7. Image Restoration

The restoration is a technique to perform a radiometric correction in order to correct distortions inserted by the optical sensor during the process of acquisition of the digital image. This distortion causes a blurring effect that should be removed.
The restoration is done using a linear filter, where the values of the mask are estimated based on the characteristics of the sensor and not empirically as in the traditional filters. A specific filter is designed to each sensor and spectral band. This option is available just for projects without projection.

⇒ ***Restoring image***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

Control Panel window

- *Select category* **Image_SPOT**
- *Select image infolayer* **PAN**

  Canvas (1)
  - *Visualize selected image* **PAN**

[Image][Restoration...]

Retoration window

- (Image ⇔ SPOT-P)
- {Name: PAN_R_5}
- (Pixel ⇔ 5)

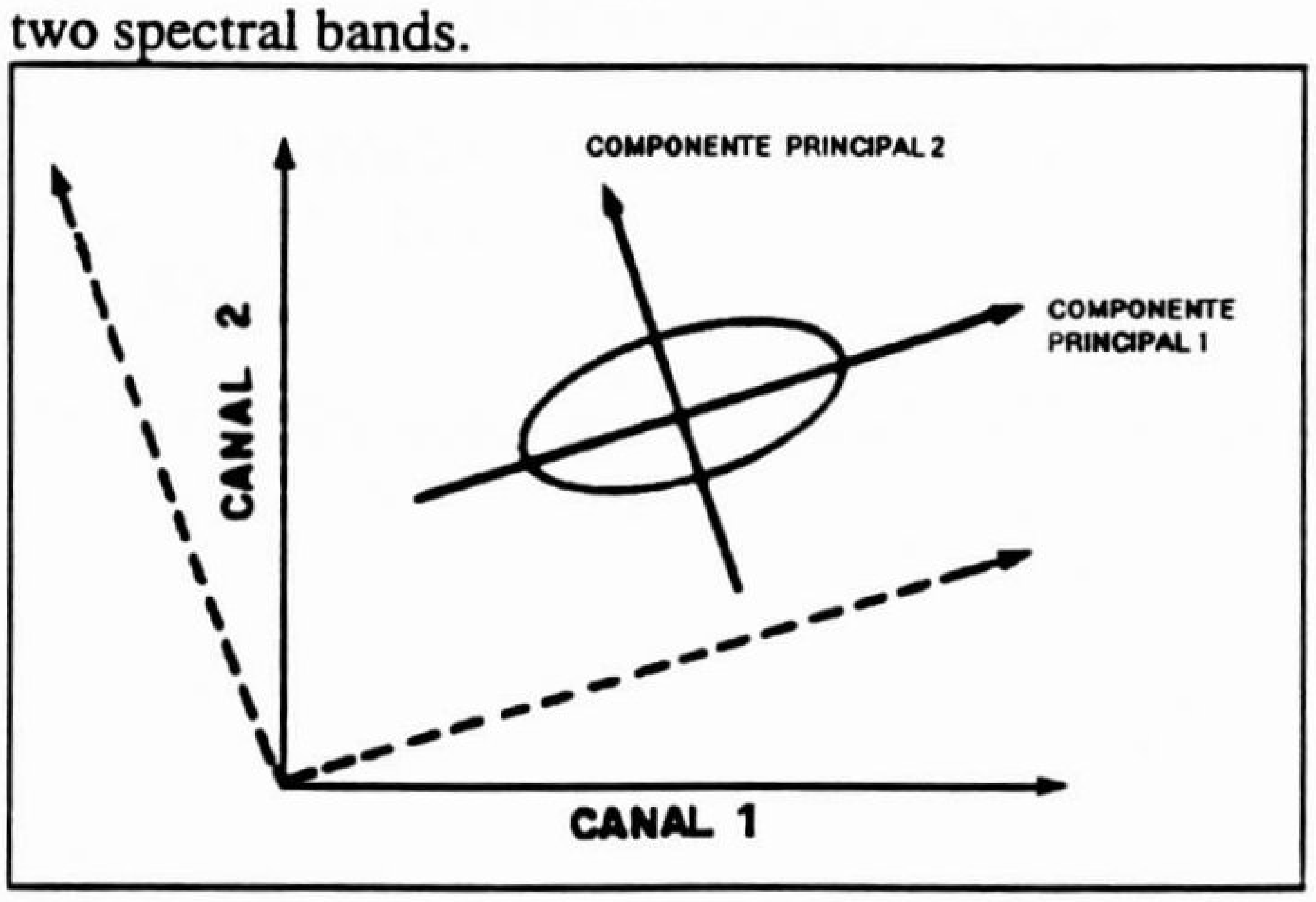
(Apply)

*Visualize the image **PAN_R_5** in **Canvas 2** to compare

8. Principal Components

This is a technique to remove or reduce spectral redundancy in a multispectral image. A new collection of bands is generated where each band has only the information not present in the other bands. The spectral redundancy is due mainly to the shadow effects resulting from the topography of the area, the overlapping between two adjacent bands and the spectral characteristic of the surface.

The position of pixels points in a multispectral space can be described by vectors, whose components are the individual spectral responses in each band. The Principal Components Transformation rotates the original co-ordinate system in order to find a new co-ordinate system in the multispetral vector space in which the data can be represented without correlation. This can be seen in the figure bellow for the case of two spectral bands.



The first component contains the brightness information associated to the topography shadows and the big variations in the general spectral reflectance of the bands. It contains most of the variance of the data that was spread in the various bands.

The second and subsequent components contain less contrast of the elements of the image and are without topography information. The third and fourth components contain, typically, less of the structure of the image and more noise. That indicates that the information is compressed in the first component.
The last component contains basically the noise existent in the original data.

⇒ ***Principal Components Transformation:***

Spring3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia.**

  Control Panel window

  - *Select category* **Image_TM**
  - *Select image infolayer* **TM5**

    Canvas (1)

    - *Visualize selected image* **TM5**

[Image][Principal Components...]

Principal Components window

- (Layers | **TM3**, **TM4** e **TM5**) – *select layers to analyze*
- (Samples ⇔ Cursor)

**Select an area over the image*

(Acquire)

*Acquire more samples

(Parameters...)

Principal Components Parameters window

- *Save parameters in the file* **princo1**

- (Output Image: tmpc)
- (Enhance ⇔ Yes)

(Apply)

*Visualize the resulting image **tmpc**. Test with no enhance

Class 5 - Classification

1. Pixel based Classification

Classification is the process of extracting information from images by recognizing patterns and homogeneous objects. The pixel based classifiers use just the spectral information of each pixel to recognize patterns and homogeneous regions.

The final result of a classification process is a digital image that can be seeing as a map of classified pixels, represented by regions (polygons) in graphic symbols or different colors.

The most common technique for pixels based multispectral classification are:

Maximum Likelihood, Minimum Distance and Parallelepiped.

Maximum Likelihood classification uses the spectral mean and covariance matrix of the classes of interest. These parameters are estimated from the training samples selected by the user for each class. It is assumed that the classes have normal distribution.

The general steps to do classification using SPRING are described below,:

1. **Creating a Context file** to store which bands of an image and which method (pixel or region) will be used to in the classification process as well as the training samples in the case of pixel classification.
2. **Acquire Training Samples** for each desired theme identified in the image.
3. **Analyzing Samples** by checking the confusion matrix of each spectral sample acquired.
4. **Executing the classification.** The image is classified based on context file and spectral samples acquired for each theme.
5. **Executing Post-Classification** to extract isolated pixels based in a **weight** and in a **threshold** defined by the user.
6. **Mapping to Thematic Classes.** The classified image resulting of the above process is converted into a thematic map. Its a mapping from Image to Thematic model type

The following steps illustrate some classification in the Spring environment.

Pixel based Classification

 # Spring 3.2()

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Display a RGB color composition of three image bands* : **TM4_Realce, TM5_Realce** and **TM3_Realce** in **RGB** channels.

[Image][Classification...]

Classification window

(Directory...: C:\springdb\dados) or other one

(Create...)

Context Creation window

{Name: contx1} (or any other name)

(Analysis Type ⇔ Pixel)

(Bands | TM3, TM4 e TM5) – select bands 3,4,5

(Apply)

(Context | contx1) – select file contx1

(Training...)

⇒ ***Creating themes or classes***

Training window

{Name: vegetation} (class/theme name)

(Color..)

⇒ ***Color Selection***

- *Select a color for the* ***vegetation*** *theme*

(OK)

(Type ⇔ Acquisition)

(Create)

⇒ ***Getting rectangular samples***

(Contour ⇔ Rectangle)

- *Select the vegetation samples over the image using the Area cursor (Set the area cursor and draw a rectangular sample over the image).*

(Get)

⇒ ***Getting polygonal samples***

(Contour ⇔ Polygonal)

- *Digitize the polygonal contour sample over image. Use the right mouse button to close the polygon.*

(Get)

- *Acquire other samples for* **vegetation** *theme, some types of urban areas, say* **urban1, urban2,** and **water.**

(Save)

⇒ ***Visualization of theme samples***

(Themes | vegetation)

(Type ⇔ Test)

- Select some of the listed samples

(View)

⇒ ***Changing of theme samples***

(Change)

- Select one of the listed samples to edit a change

(Save)

...

(Close)

(Classification...)

⇒ ***Using the acquired samples to classify image***

Image Classification window

{Name: tmc-thematic}

(Create)

(Classifier ⇔ Maxver) (Maximum likelihood)

(Acceptance-Threshold: 99%)

(Sample Analysis ...)

Sample Analysis window

- *Check the Acquisition and Test samples of each theme.*
- *Check the 'Theme Confusion Matrix' and 'Sample Confusion Matrix' for each theme.*
- *For those samples with high confusion, return to training process, reject the sample and acquire or another one for this theme.*

(Classify)

- *Display **tmc-thematic** image after classification.*

(Close)

⇒ ***Eliminating classification noise:***

(Post-Classification...)

Post-Classification window

(Classified Image | tmc-thematic)

(Weight ⇔ 2)

(Threshold ⇔ 5)

(Apply)

- *Display **tmc-thematic** image after post-classification.*
- *Test other weights and thresholds.*

⇒ ***Mapping themes in classes (Classified Image in Thematic image)***

(Mapping...)

Class Mapping window

(Classified Images | tmc-thematic)

(Category | Uso_Terra)

(Themes | vegetation)

(Classes | Mata)

- *Repeat for other themes.*

(Apply)

- *Display **tmc-thematic-T** infolayer of Uso_Terra category (thematic model)*

(Close)

2. Segmentation

In this process, the image is partitioned into regions. Regions are a connected set of pixels that are spread in two directions and present some uniformity or similarity.

The segmentation can be performed by using Region Growing, Edge Detection or Watershed algorithms.

Region Growing

Region growing is a technique of grouping data (pixels), where just spatially adjacent data are coalesced to form a region. Initially, the segmentation process considers each pixel as a distinct region. The algorithm grows the regions by aggregating two regions each time. This aggregation takes place only if the Euclidian distance between the spectral means of both regions is below a threshold specified by the user.

Edge Detection

The segmentation by edge detection uses a resultant image of edge extract process. The edge extraction algorithm could be a Sobel filter. This algorithm considers the gradient of gray level of the original image in order to generate the gradient image or border intensity image.

The algorithm also considers a threshold defined by user to following the edges. When it finds a pixel which value is greater than this threshold, the edge detection process starts. It looks around to determine which is the greatest pixel value and follows this direction until it finds other edge.

The result of this process is a binary image: value 1 is assigned to edges and value 0 is assigned to regions (or interior of edges). The binary image will be labeled such as parts of image which value is 0 will be regions limited by values 1 of the image – generating the *labeled image.*

Watershed

The segmentation by basin detection uses a resultant image of a filter process where the edges is enhanced. It could be used a Sobel filter where the gradient of the original image is considered to generate the gradient image or image of enhanced borders.

The watershed process take the enhanced image represented by a topographic surface where the gray level of pixels standing for the elevation. Now, start flooding progressively the basin of the minimum at the lowest altitude. When the water level reaches the altitude of the second minimum, we flood simultaneously the two basins until the elevation of the third minimum is reached, and so forth. In addition, dams are erected at the places where the waters coming from two different basins would merge. At the end of this flooding procedure, each minimum is completely surrounded by dams, which delimit its associated catchment basin. The set of dams thus obtained corresponds to the watersheds.

Each basin delimited by dams is labeled as one region and must be classified using *region* classifiers.

The major steps to classify an image using segmentation process is described below:

1. **Creating Segmented Image** – generate a *labeled image* in regions analyzing the gray level of each pixel and grouping them in regions based in some parameters of similarity and in statistical analysis.
2. **Creating Context file** – in this file is specified which bands and which labeled image will be used in a classification process by region and also it storage the training samples if it is a supervised classification.
3. **Training Classification** – acquire regions (as polygonal samples) of themes over image.
4. **Analysis Sample** – check the confusion matrix of each spectral sample acquired.
5. **Extracting Regions**: this algorithm extract the statistical average and variance of each region in a labeled image, considering the bands defined in a context file.
6. **Select Region Classifier** – select one classifier by region.
7. **Execute the Classification** - take the samples and its statistical parameters, the context file and regions extracted and make the classification.
8. **Mapping to Classes** – convert the classified image (Image model category) in a classified thematic map (thematic model category).

⇒ ***Image Segmentation***

Spring-3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia**

[Image][Segmentation...]

Segmentation window

- (Bands | **TM3_Amostra, TM4_Amostra, TM5_Amostra**
- (Method ⇔ Crescimento de regiões)
- (Similarity ⇔ 8)
- (Area (pixels) ⇔ 10)
- {Segmented Image: seg-8-10}
- (Arc Smoothing ⇔ Yes)

(Apply)

- *Display Labeled Image* ***seg-8-10*** *over a color composed image used in this segmentation.*

- *Test other values of similarity and area.*

3. Region Classification

The region classifiers considers, beside of spectral information of each pixel, the spatial information that involve the relation among pixels and its neighborhood.

It is a kind of simulation of the visual interpreter, and it tries to recognize homogenous areas in the image based on spectral and spatial information. The borders are used initially in order to separate the regions and the spectral and spatial properties are used to aggregate areas with the same texture.

The ISOSEG classifier is the algorithm which is available in SPRING to classify regions of a segmented image. It is a algorithm of aggregation of data without supervision and it is applied in a set of regions that are characterized by their area and statistical attributes – mean, covariance matrix.

⇒ Region Classification
Spring-3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Display image* **TM3_Amostra**

[Image][Classification...]
Classification window
⇒ Creating Context file in some directory
(Directory...: C:\springdb\dados) (CR) * Don't forget this directory
(Create...)
Context Creation window
{Name: clasreg}
(Classification: ⇔ Regions)
(Bands I **TM3_Amostra, TM4_Amostra, TM5_Amostra**) – *select band*
(Segmented Image I **seg-8-10**)
(Apply)
⇒ Extracting regions
(Region Extraction)

⇒ Unsupervised Classification
(Classification...)
Image Classification window
{Name: tm345sub-isoseg}
(Create)
(Classifier ⇔ Isoseg)
(Acceptance-Threshold ⇔ 99%)
(Classify)

- *Display classified image* **tm345sub-isoseg.**

⇒ Supervised Classification
Spring-3.2 window

- *Activate database* **Curso**

- *Activate project* **Brasilia**
- *Display color composition of TM3_Amostra(B), TM4_Amostra(R) and TM5_Amostra(G), and labeled image seg-8-10*

[Image][Classification...]

Classification window

(Directory...: C:\springdb\dados) (CR)

*Directory where you saved the context file (clasreg.ctx)

(**Contexts**: clasreg)

(Training...)

Training window

{Name: vegetation}

(Color...) – Select appropriate color .

(OK)

(Create)

(Type ⇔ Acquisition)

- *Click inside the region over the image with vegetation*

(Get)

- *Get enough samples for vegetation theme*
- ***Get samples for all themes: Urban1, Urban2 and Water.***

=> *Classifying*

(Classification...)

***Image* Classification window**(by regions)

{Name: tmsub-bata}

(Create)

(Classifier ⇔ Battacharya)

(Acceptance-Threshold ⇔ 99%)

(Classify)

- *Display* ***tmsub-bata*** *classified image on Canvas 1 and a color composition on Canvas 2.*

⇒ Mapping themes of classified image to classes of Uso_Terra category defined in Data Model)

(Mapping...)

Class Mapping window

(Categories | Uso_Terra)

(Classified Images | tmsub-bata)

(Themes | Vegetation)

(Classes | Mata)

- *Repeat for each theme.*

(Apply)

- *Display* **tmsub-bata-T** *thematic Layer* (category: Uso_Terra)

4. Mosaic of Images

Spring-3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia**

⇒ Creating one empty infolayer to use in mosaic tool

[Edit][InfoLayer...]

Layers widow
(Category | Image_TM)
{Name: tm5-mosaic}
The *Bounding box will be the project's bounding box*
{X(m): 30}, {Y(m): 30}
(Create)
(Close)

⇒ Mosaic of images

[Image][Mosaic...]

Mosaic windows
(Projects | Brasilia)
(Categories | Image_TM)
(Origin InfoLayer | TM5_Amostra)
(Apply)
* *Doing image mosaic with the second part*
(Project | Brasilia)
(Categories | Image_TM)
(Origin Infolayer | TM5_Parte2)
(Mosaic)
(Apply)
* *Doing image mosaic with the third part*
(Projects | Brasilia)
(Categories | Image_TM)
(Origin Infolayer | TM5_Parte3)
(Mosaic) *DON'T FORGET TO CLICK IN MOSAIC
(Apply)

- *Display* **tm5-mosaic** image
- *Display all image's parts in CANVAS 2,3,4.*

Class 6 – Vector Data Manipulation

1. Editing Vector Data

Editing vector data in SPRING applies to **thematic, cadastral, networks,** and **digital** model types. These vector representations are used to accurately describe geographic objects by means of **points, lines, and areas** (or polygons), which are the basic entities used to define thematic classes, geographic objects, and digital terrain model samples (contour lines and spot heights)..

Vector data can be also imported, as shown in the following examples for ASCII or DXF format files.

1.1 Importing ASCII files

This example explains the major steps towards importing a drainage map and a road map. Note that a category and its thematic classes must be defined for the road map, as it has not been created before.

⇒ ***Creating Category of Thematic model***
Spring-3.2 window
- *Activate database* **Curso**
- *Activate project* **Brasilia**

[File] [Data Model...] or buttom

Data Model window
{Categories - Name: Vias_acesso}
(Model ⇔ Thematic)
(Categories ⇔ Create)
{Thematic Classes - Name: Principais}
(Thematic Classes ⇔ Create)
{Thematic Classes - Name: Secundarias}
(Thematic Classes ⇔ Create)
{Thematic Classes - Name: Urbanas}
(Thematic Classes ⇔ Create)
(Apply) - (to create category and its thematic classes)

⇒ ***Defining visual characteristics for each line of a thematic class***
{Categories: Vias_acesso}
{Thematic Classes: Principais}
[Visual...]

Graphical Presentation Properties window
(LINES | CONTINUOUS)
(WIDTH | 1)
(Color...)

Color Selection window
- Select a color
(OK)

(Apply)
- *Repeat for each Thematic Class in the Data Model window*

(Close)

(Apply)
(Close)

⇒ Importing LINES from ASCII file Mavias.L2D to a Spring Thematic Layer
Spring-3.2 window
- *Activate project* **Brasilia**
[File][Import...]
Import window
External Data
(Directory...: C:/springdb/Dados) -
(Model ⇔ Thematic)
* Importing lines from Mavias.L2D file
(Format ⇔ ASCII : Mavias.L2D)
(Entity ⇔ Lines with topology.), (Units ⇔ m -meters), {Scale: 25000}
- *Projection and Bounding box will be the same of the active project*

SPRING
- *Project: it will be the active project*
(Category...)
Category List
(Categories | Vias_acesso)
(Apply)
Import window
{Layer: Map_vias}
(Apply)

⇒ Importing IDENTIFYERS/LABels for each line from a file to the active thematic layer
Spring-3.2 window
[File][Import...]
Import window
**The same settings used for lines importation, except:*
(Format ⇔ ASCII: Mavias.LAB)
(Entity⇔ Labels)
(Apply)
(Close)

⇒ Show thematic data imported to Mapa_vias layer
Control Panel window
(Categories | Vias_acesso)
(Layers | Map_vias)
(Lines), (Classes)
(Select)
Class Selection
- *Select some/all/None class*
(Enable Canvas 1)
(Show Canvas 1)
[Execute][Draw] or button ◺

* Repeat the same process to import lines and labels from the drainage map. Thematic category for drainage and all classes are already defined in the database named **CURSO**.

⇒ Importing LINES from ASCII file to a Spring Thematic Layer
Spring-3.2 window
- Activate database **Curso**
- Activate project **Brasilia**
[File][Import...]
    Import window
    (Directory...: C:/springdb/Dados)
    (Model ⇔ Thematic)
    (Format ⇔ ASCII: Drenagem.L2D)
    (Entity⇔ Lines with topology.), (Units ⇔ m -meters), {Scale: 25000}
    - Projection and Bounding box will be the same of the activated project
    SPRING
    - Project: Brasilia (it will be the activated project)
    (Category...)
        Category List
        (Categories | Drenagem)
        (Apply)
    {Layer: ***Map_drenagem***}
    (Apply)

⇒ Importing IDENTIFYERS/LABels for lines from a file to the active thematic layer
Spring-3.2 window
[File][Import...]
    Import window
    - Idem to lines importation, except:
    (Format ⇔ ASCII: DRENAGEM.LAB)
    (Entity⇔ Labels
    (Apply)
    (Close)

⇒ Show thematic data imported to Map_drenagem layer
Control Panel window
(Categories | Drenagem)
(Layers | Mapa_rios)
(Lines), (Classes)
(Select)
    Class Selection
    - Select some/all/None
Control Panel window
(Enable - 1)
(Show - 1)
Spring-3.2 window
[Execute][Draw] or button ⊿

2. Graphic Edition

Some parameters must be set up properly before editing arcs:

A) **Graphical Editor** or **Verification**

B) **Edit** : Lines, Points, or Breaklines;

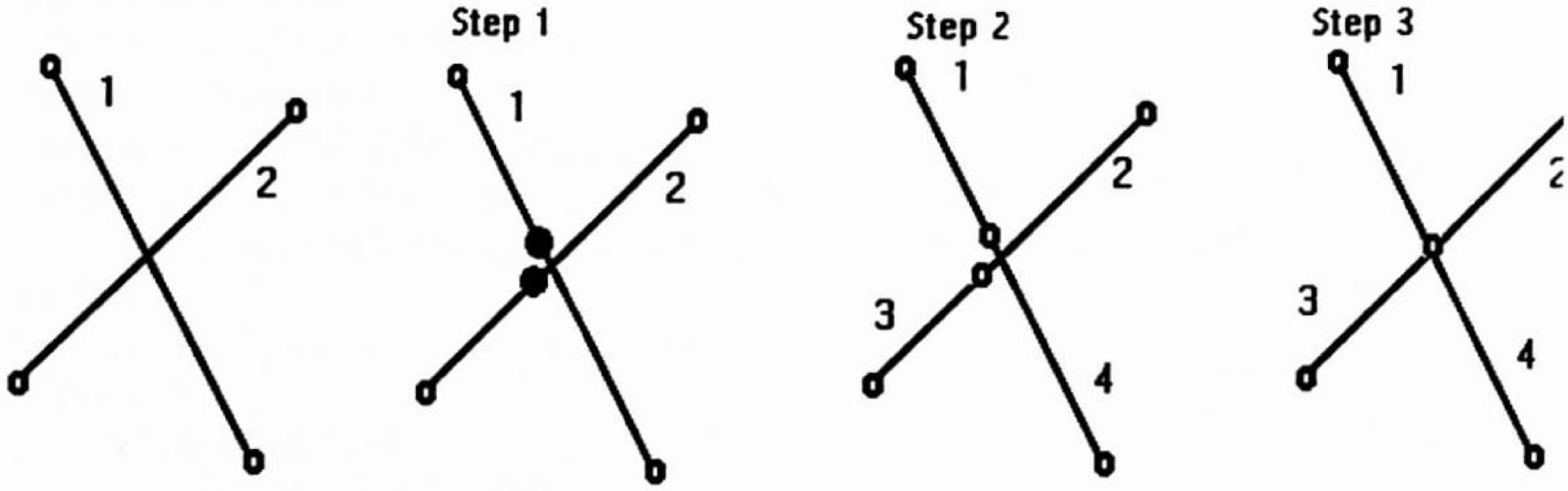
C) **Mode** : Continuous or Step

D) **Topologia** : Automatic or Manual;

E) **Digitizing Step (mm)** : 0,00 a 4,00;

F) Editing : **Create Line, Create Closed Line, Add Point, Move Point, Break Line, Join Lines, Delete Line, Delete Point, Clean Area, Concatenate Lines.**

The following figure shows the necessary steps to edit two intersecting arcs in order to define a single node. In **step 1** one point is added to each arc. The two arcs are then broken into four arcs in **step 2.** In **step 3** the four endnodes are finally joined to form a single node. The number of arcs on each step are also indicated in the figure.



Original

Add one point in each arc

Breaking the arcs generating 4 arcs and 2 nodes

Joint 2 nodes in one

The next practice class involves line adjustment and polygon generation for a landuse map. Topology is defined according to the map in **annex 1.** Some files need to be imported to provide the information required to form a landuse map. First you must edit and import an ASCII file that contains coordinates for the landuse infolayer boundaries. Then you must import both the drainage (file **Agua.L2D**) and urban area (file **Urbano.L2D**) boundaries.

The landuse map must contain only the classes **Cerrado, Agua, and Urbano**, as depicted in annex 1.

- **OBS** : For the sake of practicing, one of the arcs of the landuse map must be adjusted manually, according to the procedure already described.

The boundary file for the landuse map (**LIM_USO.L2D**, for instance) is presented below. Use a text editor (**vi, asedit, textedit,** notepad, etc.), add those boundary coordinates, and save the file in the directory "**/springdb/Dados**".

```
LINES
INFO
// ASCII file generated by SPRING-3.2
// project Brasilia  Infolayer Molde
INFO_END
 s 15  52  30   w 47  57  30
 s 15  41  55   w 47  57  30
 s 15  41  55   w 47  47  00
 s 15  52  30   w 47  47  00
 s 15  52  30   w 47  57  30
END
END
```

⇒ Importing thematic data to map_uso layer
Spring-3.2 window
- *Activate database* **Curso**
- *Activate project* **Brasilia**

[File][Import...]

Importing window
(Directory...: C:\springdb\Dados
(Model ⇔ Thematic)
(Format ⇔ ASCII : LIM_USO.L2D)
(Entity⇔ Lines without adjust.), (Units ⇔ m -meters), {Scale: 25000}

- *Projection and Bounding box will be the same of the active project*

SPRING

- *Project: Brasilia (the active project)*

(Category...)
Category List
(Categories | Uso_Terra)
(Apply)
{Layer: Map_uso}
(Apply)

*Repeat this process to import the drainage and urban area boundaries, but do not forget to click on the **Mosaic** option this time in order to transfer those lines to the same infolayer.

=> Importing limits of water to Map_uso layer
(Format ⇔ ASCII : LIM_Water.L2D) (agua.L2D)
(Entity⇔ Lines without adjust.), (Units ⇔ m - meters), {Scale: 25000}
(Category...)
Category List
(Categories | Uso_Terra)
(Apply)
Importing window
{PI: Map_uso)
(Mosaic)
(Apply)
=> Importing limits of urban area to Map_uso layer
(Format ⇔ ASCII : Lim_Urban.L2D)
(Entity⇔ Lines without adjust.), (Units ⇔ m - meters), {Scale: 25000}

(Category...)
Category List
(Categories | Uso_Terra)
(Apply)
{Layer: Map_uso}
(Mosaic)
(Apply)

⇒ Show the layer Map_uso
Control Panel
(Categories | Uso_Terra)
(*<u>Layer</u>* | Map_uso)
(Lines)

Spring-3.2 window
[Execute][Draw] or button

⇒ Adjusting lines and generating polygons in the Map_uso layer
- *Activate thematic layer* **Map_uso** *created before*

[Edit][Vector Editing...]
Topological Editor
(Graphical Editor)
(Snap(mm) ⇔ 0.50)
(Show Nodes)
(Adjust)
- *Check the message in the bottom of this interface*

(Verification)
(Verify ⇔ Nodes)
** you will see blue and green nodes*
Spring-3.2 window
[Execute] [Draw] or button
(Graphical Editor)
LINES EDITOR
- Use the Operations: Delete Line, Break Line, Join Line, Add Point, Delete Point and Move Point to correct the nodes/lines position.
- *Activate* **Area cursor** *e* **SPRING-3.2** [Execute] [Draw] or button *to zooming the edition area - Ps: After zooming, deactivate the area cursor in order to transfer mouse control to the edition interface.*

(Adjust)
- *Check the message on the bottom of this interface – if there is some open node, edit again and correct it.*

(Generate Polygons)

OBS: While editing lines or adjusting arcs you can select the option **Show Nodes** in the dialog box **Topological Editor** to show the nodes in the active window. After clicking on **<u>Draw</u>** all adjusted nodes are shown as green squares and all disconnected nodes appear as blue crosses. Changing the digitizing factor or threshold will increase the sizes of squares and crosses accordingly.

⇒ Associating Thematic Classes to Polygons
Topological Editor window
(Classes...)
    Thematic Classes Editor window
    (Classes | Agua)
    (Operation ⇔ Associate)
        (Entity | Polygon)
    - *Click over the polygon that correspond to water (agua) in edition area*
    - *Use* (Operation ⇔ Dissociate) *if you make a mistake and click over the polygon that you want to dissociate.*
    - *Repeat this association process for other Classes (Urbano and Cerrado)*
    (Close)

*Close the interface **Topological Editor** if your map is ready
* Show the thematic map (now it has all polygons associated to the thematic classes defined in the Data Model)

3. Editing the Soils Map

Soils map creation must conform to annex 2 contents. You are supposed to import the boundaries for the soils infolayer. If you have access to a digitizing table you should use it to digitize the soils map. Otherwise, use the mouse to digitize your 'freestyle' soils map directly on screen. Remember to define the appropriate thematic category and classes.

⇒ Defining Thematic Model to soil Map
Spring-3.2 window
- *Activate database* **Curso**
- *Activate project* **Brasilia**

[File] [Data Model...]
    Data Model
    {Categories - Name: Soil}
    (Model ⇔ Thematic)
    (Categories ⇔ Create)
    {Thematic Classes - Name: LEd1}
    (Thematic Classes ⇔ Create)
    {Thematic Classes - Name: LVd1}
    (Thematic Classes ⇔ Create)
    {Thematic Classes - Name: Cd12}
    (Thematic Classes ⇔ Create)
    {Thematic Classes - Name: Cd1}
    (Thematic Classes ⇔ Create)
    {Thematic Classes - Name: Cd17}
    (Thematic Classes ⇔ Create)
    {Thematic Classes - Name: Cd14}
    (Thematic Classes ⇔ Create)
    (Apply) - to create in a database the thematic category **Soil** and its classes.

    **** Defining one visual for each thematic class***
    Data Model window
    {Categories: Soil}

{Thematic Classes: LEd1}

(Visual...)

Graphic Presentation Properties

- Set visual characteristics for Areas

(AREAS | SOLID) – or other filling pattern of polygon

(AREAS - Color...)

Color Selection

- Select one color

(OK)

(Apply)

- *Define one visual to each thematic classes defined above*

(Close)

Data Model

(Close)

⇒ Importing the limits of map of the soil

[File][Import...]

Importing window

(Directory...: /springdb/Dados)

(Model ⇔ Thematic)

(Format ⇔ ASCII : Molde.L2D)

(Entity⇔ Lines without adjust.), (Units ⇔ m -meters), {Scale: 25000}

- Projection and Bounding box will be the same of the activated project

SPRING

- *Project: Brasilia (it will be the active project)*

(Category...)

Category List

(Categories | Soil)

(Apply)

Importing window

{Layer: Map_soil)

(Apply)

Editing Vectors on the Soils Map

⇒ Digitizing table calibration

Spring-3.2 window

[Tool][Table Calibration...]

Calibration window

*Follow the same steps defined in Registration/Georeference (Class 3)

- Place map 2 on the table and digitize lines/Adjust/Generate Polygons

⇒ Preparing the edition using CANVAS (if table is not available)

- Show image **Comp_3B_4R_5G**
- Activate layer **Map_soil**
- Use Map 2 as a reference to digitizing lines over image.

⇒ Digitizing lines

Spring-3.2 window

- *Activate project* **Brasilia** *and layer* **Map_soil** *of the category* **Soil**

[Edit][Vector Editing...] or [Thematic][Vector Editing]
    Topological Editor window
    (Graphic Editor)
    (Edit ⇔ Lines)
        LINE EDITOR
    ⇒ ***Editing in continuous mode***
    (Mode ⇔ Continuous)
    (Topology ⇔ Manual)
    (Digit. Step(mm) ⇔ 0.50)
    (Operation | (Create Line or Create Closed Line)
(Set Table cursor) *-only if you are using digitize table*

- *Digitize lines*

    ⇒ ***Adjusting lines and generating polygons***
    (Snap(mm) ⇔ 0.50)
    (Show nodes)
    (Adjust])   * *you will see blue and green nodes*

    - *Check the message at the bottom of this interface – if there is some open polygon, edit again and correct it.*

    (Verification)
    (Verify – Nodes)
[Execute][Draw] or button Z

    - *Use all edition tools available in Operation to correct the map edition* (Delete Line, Break-line, Join Line, Add Point, Delete Point, Move Point)

    (Adjust)
    (Generate Polygons)

⇒ ***Associating Thematic Classes to polygons***
    Topological Editor
    (Classes...)
        Thematic Classes Editor
        (Classes | LEd1)
        (Operation ⇔ Associate)
        (Entity | Polygon)
        - *Click over the polygon that correspond to Led1 class in edition area*
        - *Use* Operation ⇔ Dissociate *if you make a mistake and click over the polygon that you want to dissociate.*

    - *Repeat this association process for all Classes according to the map 2.*
    - Show the resulting **Map_soil**

4. Format Conversion

Although both raster and vector data formats are available to all data models, Spring currently supports conversion between raster and vector for the thematic data model only.

Vector Format

The vector representation attempts to accurately describe geographic entities in terms of boundaries, locations, and dimensions.

SPRING-3.2 makes use of different data models that may have some kind of vector representation:

Category/Model	Vector Format Representations
Thematic:	Points, Lines, and Classes
DTM:	Samples and TIN
Cadastral:	Points and Lines
Network:	Points and Lines

Raster Format

The raster format comprises a set of cells arranged in a 2D matrix that occupy contiguous locations. Each cell – here defined as image element, matrix element, or 'pixel' – has its location defined by a column and a row and has an attribute assigned to it.

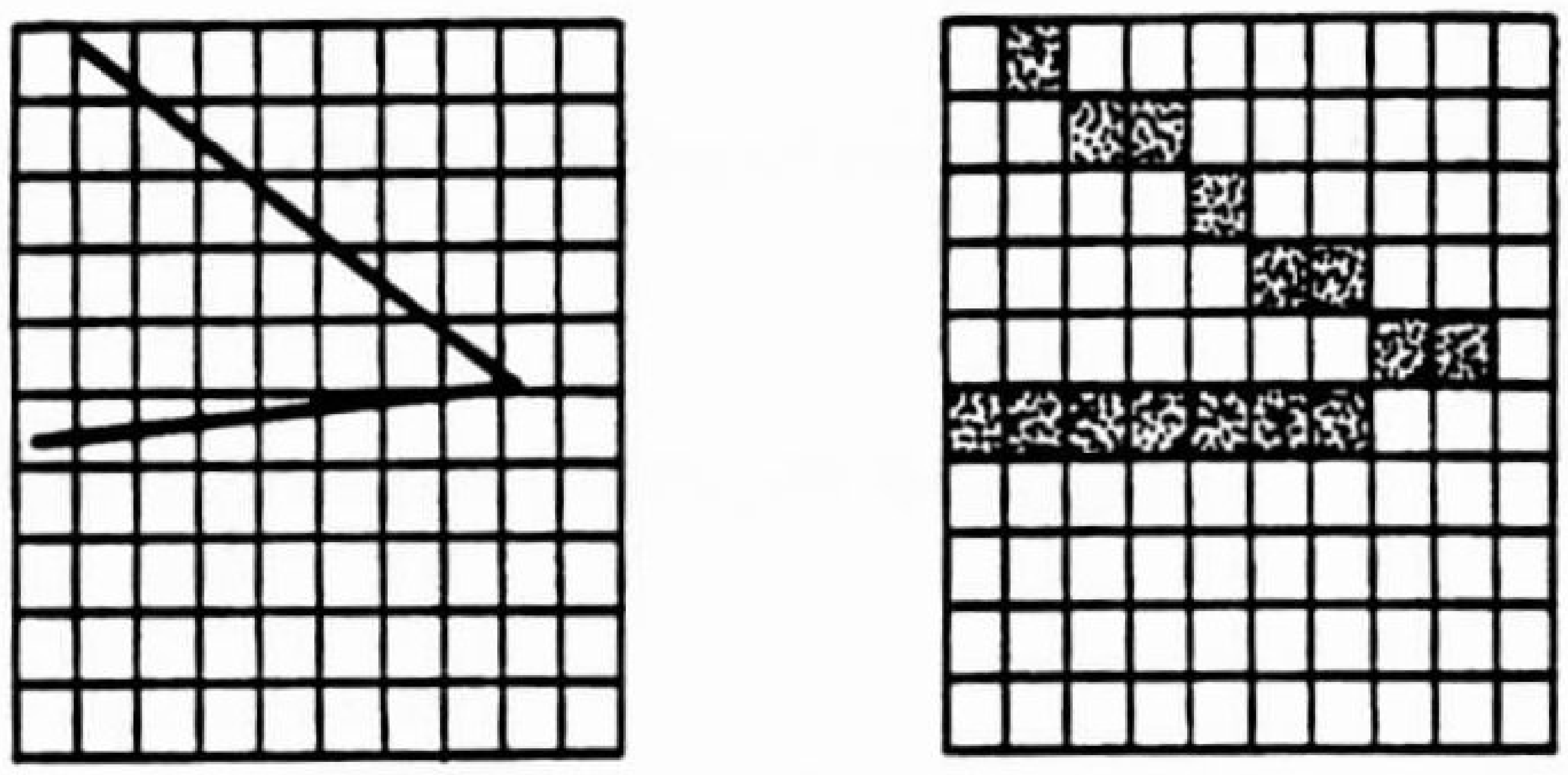
Raster and vector representations are not exactly equivalent for a certain data set. Converting from vector to raster usually implies in worsening accuracy because continuous boundaries become a sequence of discrete elements according to the spatial resolution of the resulting image. This is somehow compensated by the efficiency of geographic analysis operations in the raster domain.

Vector-Raster Conversion

The raster-vector conversion creates a thematic image representation from the infolayer classes. Each infolayer contains only one thematic image representation. A new conversion is required if vector are edited or classes are redefined.

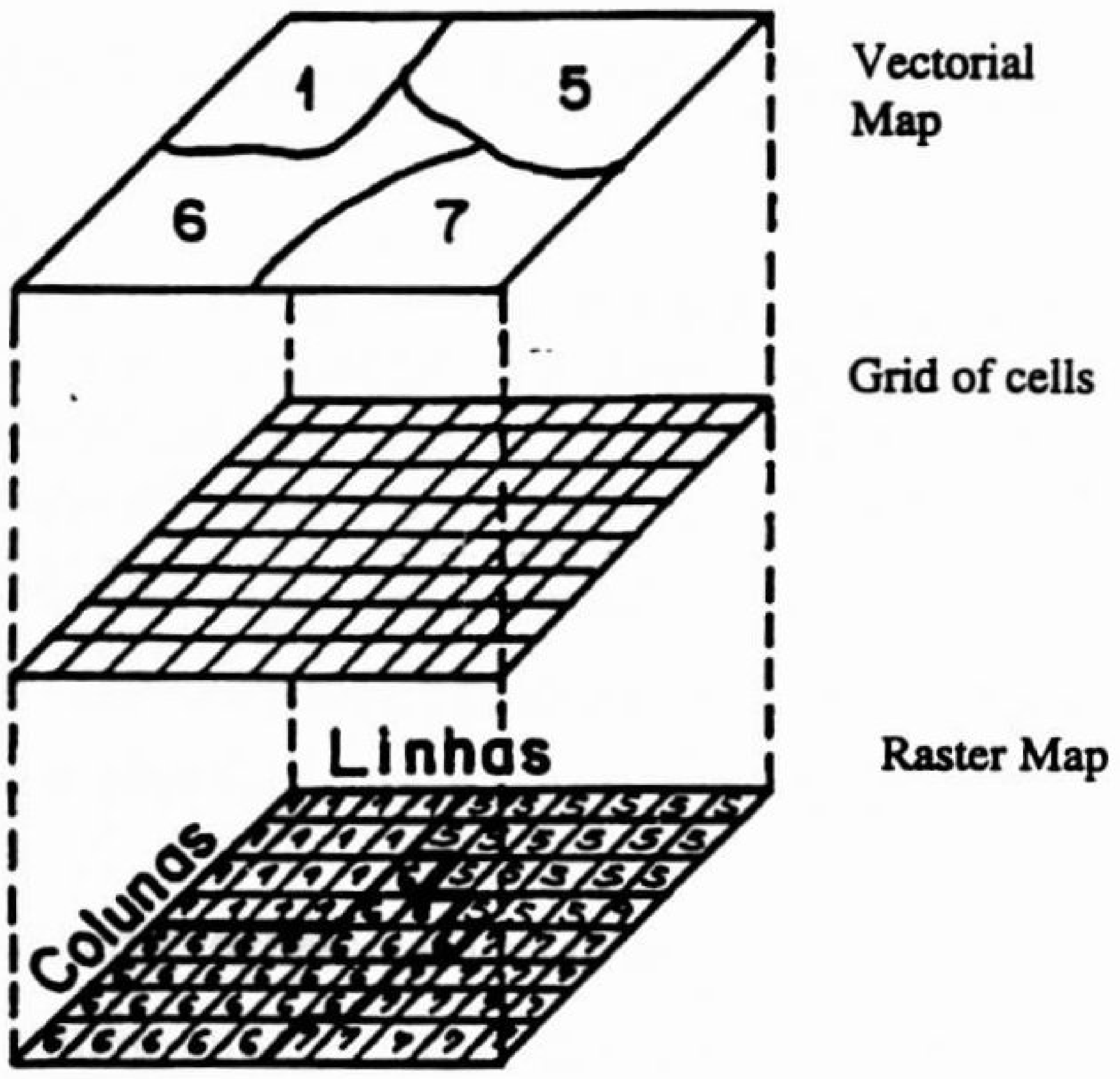
The conversion may be outlined by superimposing a vector or linear element to a matrix structure. Pixels crossed by the linear elements are identified and coded according to line classes.

Example:



Vector Format Raster Format

Converting polygons requires a previous knowledge of each polygon bounding box, as well as pixel size. This information is then used to define the grid or matrix to be superimposed to the polygons. Each pixels has a class (some attribute) assigned to it.



⇒ *Converting Thematic map from* **Vector** *to* **Raster** *representation*
Spring-3.2 window

- *Activate database* **Curso**
- *Activate project* **Brasilia**

    Control Panel window

    - *Activate o Thematic* layer **Map_Soil**

[Thematic][Vector->Raster...]

Vetor->Raster
{Horizontal: 30}, {Vertical: 30}
(Apply)
⇒ ***Show resultant*** **Image** ***representation of*** **Map_soil** ***in Canvas 2***
Control Panel window
CANVAS CONTROL
(Show | 2)

- Select **Map_Soil** layer and its **Image** representation

[Execute][Draw] or button

* Repeat this process to Map_uso layer

⇒ ***Converting thematic map from Raster to Vector representation***
Control Panel window

- *Activate the* **Image** *representation of some thematic layer for instance:* **tm345sub-isoseg-temática**

Spring-3.2 window
[Thematic][Raster ->Vector...]
RASTER ->VECTOR
(Arc Smoothing ⇔ Yes)
(Apply)
- *Show the vector representation of this layer*

Class 7 – Digital Terrain Modeling

1. Digital Terrain Modeling

A Digital Terrain Model – **DTM** – is a mathematical representation of the spatial distribution of a phenomenon occurring in a determined area of the earth surface. The phenomenon is considered an attribute of the spatial data. In general, the attribute represented varies continuously in the surface. Follow examples of DTM use:

a) Altimetry data storing in order to create topographic maps;

b) Cut and fill analysis for civil engineering projects like roads and dams;

c) Derivation of slope and aspect maps, from topographic data, to be used in geomorphologic and suitability analyses;

d) Analysis of geophysical and geochemical variables;

e) Planar projection of the three-dimensional representation combined with other spatial data.

In the digital terrain modeling process one can distinguish three steps: data acquisition; model generation; and applications, also referred as analysis, performed over the model.

Digital Terrain Model Edition

In this part the user will be in contact with all the DTM edition tools to create and edit isolines and 3-d sample points. The user can use his imagination and some other data presented in the database in order to create an experimental sample set. The *Brasilia* project contains a real topographic sample set that will be used later.

⇒ Preparing for canvas edition (in the case that there is no digitize table)

- # Start – Programs – Spring 3.2 – Spring(⊞) *#spring (* **UNIX** *)*

SPRING-3.2

- Activate the database **Curso**

- Activate the project **Brasilia**

- Create Infolayer **DTM_teste** *of the* **Altimetria** *category*

- Activate Infolayer **DTM_teste** *just created*

⇒ Editing vectors

SPRING-3.2

- [Edit][Vectors...]

Topological Editor

- (Operation ⇔ Graphic Edition)

⇒ Editing isolines

Topological Editor

- (Edit ⇔ Lines)
- (Mode ⇔ Continuous)
- (Digit. Factor (mm) ⇔ 0.50)
- {Value Z: 100}
- (Create Line) *or* (Create Closed Line)

IMPORTANT: If you will edit lines using the digitizer table, set the Digitizer cursor

- Digitize (table or canvas) isolines with z-values equal **100**
- Repeat for all isolines with other Z-values

⇒ ***Verifying isolines***

Topological Editor
- (Verification)
- (Verify ⇔ Lines)

  Lines
  - *Click over the line, in the canvas, to be checked*
  - *Checking the Z value*
  - {Value Z: 100}, (Draw) - *if the correct value is* **100**

⇒ ***Editing 3-D points***

Topological Editor
- (Operation ⇔ Graphic Edition)
- (Edit ⇔ Points)
- {Points Edition ⇔ Value Z: 50}
- (Points Edition ⇔ Create)
- **Control Panel** (Digitizer cursor) - *PS: Only in the case of edition using digitize table*
- *Digitize (table or canvas) points with z value* **50**
- *Repeat for all other points with different z values*

⇒ ***Checking 3-D points***

Topological Editor
- (Checking)
- (Check ⇔ Points)

  Points
  - *Select, in the canvas, the point to be checked*
  - *Check the z value*
  - {Value Z: 50}, (Draw) - *if the correct value is 50*

⇒ ***Editing breaklines***

Topological Editor
- (Operation ⇔ Graphic Edition)
- (Edit ⇔ breaklines)
- (Mode ⇔ Continuos)
- (Digit. Factor (mm) ⇔ 0.50)
- (Create L)
- **Control Panel** (Digitizer Cursor) – *P.S.: Only for edition using the digitize table*
- *Digitize (table or canvas) breaklines*

* The procedures that follow (grid generation) must be executed on data created and edited by the user.

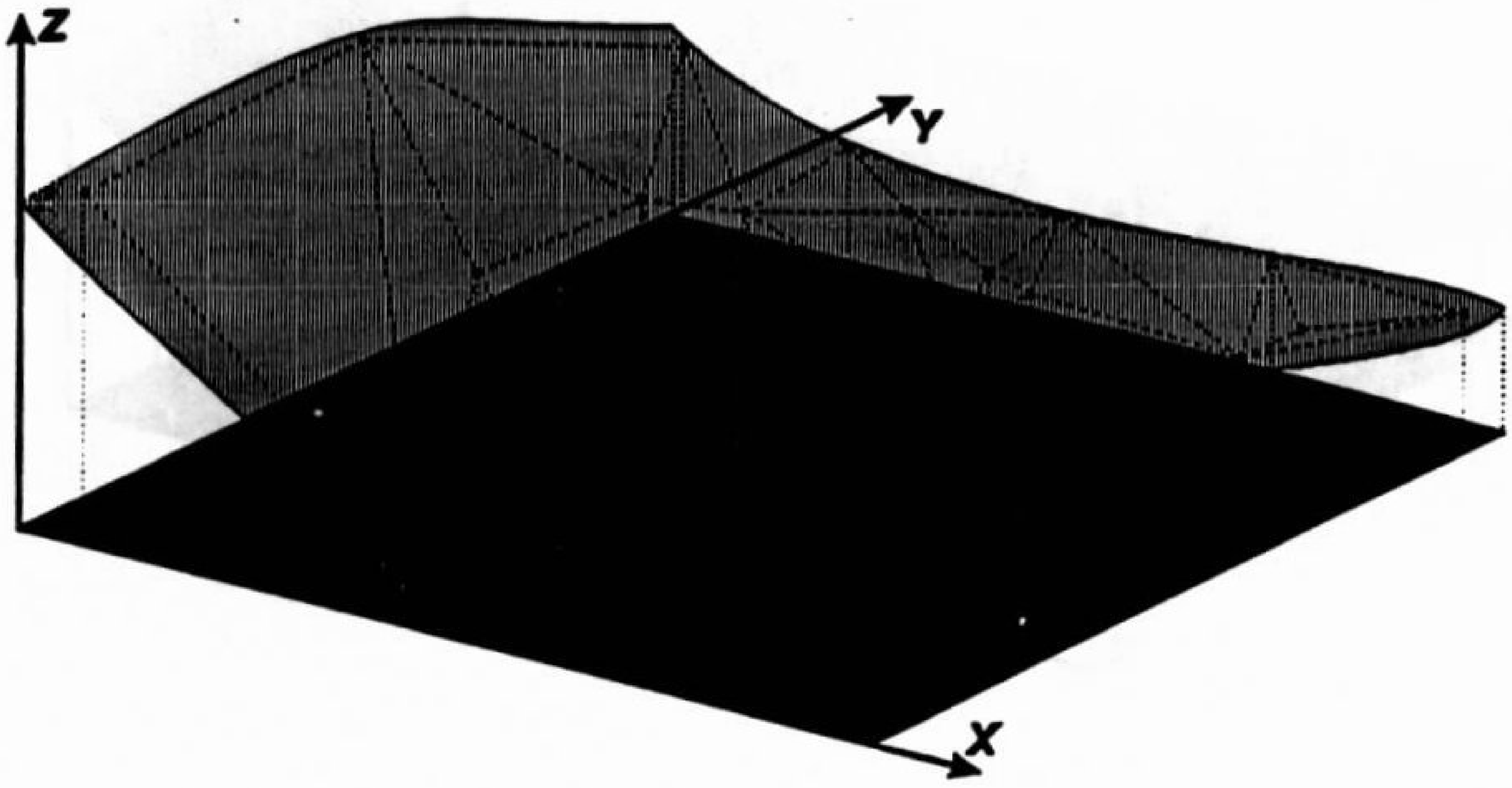
2. Grids and interpolators

The rectangular grids are, in general, used for qualitative applications, as for example, planar geometric visualization of the 3-D model. The TIN (Triangular Irregular Network) must be used when the accuracy of the model is important or in qualitative applications..

The rectangular and triangular interpolators, used in the SPRING for digital terrain model generation, were specified according the different DTM representations: 3-D sample (points-3D or/and isolines), rectangular grid or triangular network.

Triangular Grid

In the TIN model, the triangle vertices are, in general, the samples of the surface attribute. Besides, this model allows the incorporation of complementary information about the morphological structures of the attribute, such as: ridges lines, drainage lines and, discontinuities in the surface. The morphological structures are considered, together with the 3-D samples, during the process of TIN generation in order to create a better model representation of the surface.



⇒ ***TIN creation***

- *# Start – Programs – Spring 3.2 – Spring(▦)* *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate Infolayer* **DTM_teste** *of the* **Altimetria** *category*
- [DTM][Triangular...]

⇒ ***Generating TIN without breaklines***

TIN Generation

- (Input ⇔ Sample)
- {Output Layer: dtm-tin}
- (Bounding Box...)
- *Select an area on the layer*
- (Method ⇔ No Constraints)
- (Apply)

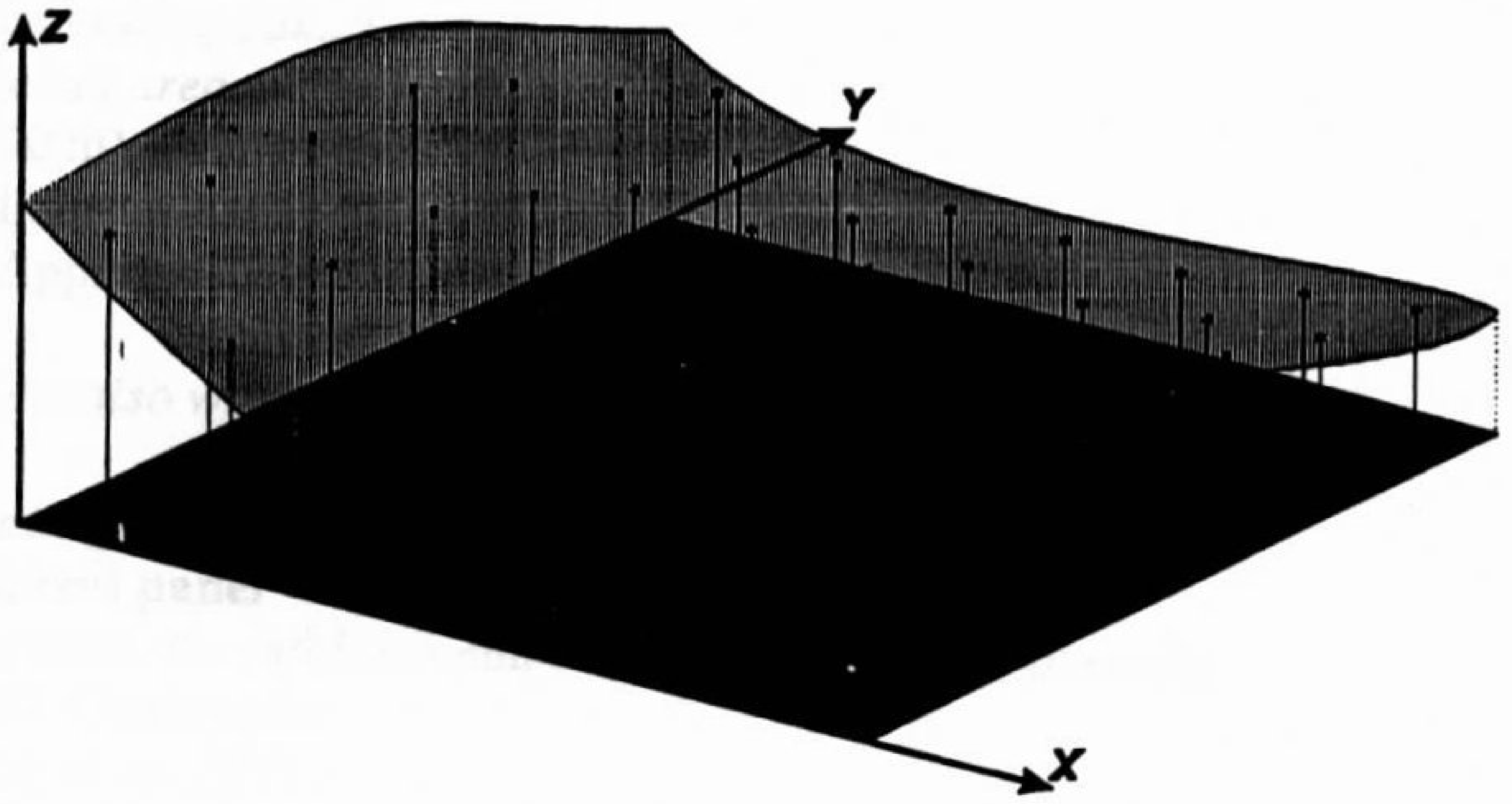
⇒ ***Generating TIN with break lines***

TIN generation
- (Input ⇔ Sample)
- {Output Layer: dtm-brk}
- (Bounding Box...)
- (Method ⇔ With Constraints)
- (Apply)

- *Visualize the grids just created.*

Rectangular Grid

A rectangular, or regular, grid is a model that represents the surface using a rectangular faced polyhedron structure. The polyhedron vertices will be the sample points only if the sampling was made in the same spatial positions of the grid. In general these vertices are points interpolated from the sample points.



⇒ ***Rectangular Grid Creation***

- *# Start – Programs – Spring 3.2 – Spring(*⊞) *#spring (* **UNIX** *)*

SPRING-3.2
- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **dtm** *of the* **Altimetria** *category*
- [DTM][Rectangular Grid...]

⇒ ***Generating the rectangular grid from samples***

Grid Generation

- (Input ⇔ Sample)
- {Output Layer: dtm-grd}
- (Bounding box...)
- *Select are on the layer*
- {X(m): 100}, {Y(m): 100}
- (Interpolator ⇔ Nearest Neighborhood)
- (Apply)

- *Test with other interpolators*

⇒ ***Generating rectangular grid from another rectangular grid***

Control panel

- *Activate the infolayer* **dtm-grd** *of the* **Altimetria** *category*

Grid Generation

- (Input ⇔ Grid)
- {Output Layer: dtm-grd-50}
- (Bounding box...)
- *Select area on the layer*
- {X(m): 50}, {Y(m): 50}
- (Interpolator ⇔ Bilinear)
- (Apply)

- *Test also with bicubic interpolator*

⇒ ***Generating rectangular grid from triangular grid***

Control panel

- *Activate the infolayer* **dtm-tin** *from the category* **Altimetria**

Grid Generation

- (Input ⇔ TIN)
- {Output Layer: dtm-grd-tin}
- (Bounding box...)
- *Select an area on the layer*
- {X(m): 100}, {Y(m): 100}
- (Interpolator ⇔ Linear)
- (Apply)

- *Test also the quintic interpolator without breaklines and with breaklines (in case the TIN was created with constraints).*

3. DTM applications

The DTM applications are obtained directly from the rectangular or triangular grid models. They are organized by functions that can be activated from the SPRING main menu. Before running any of these application the user must have the rectangular or the triangular model representation created.

NOTE: The user can use the infolayer **Mapa_altimetrico** that was already created in the Brasilia project.

3.1 Image Generation for the Digital Terrain Model

The user can create *gray level* or *shaded* images from the grid models. A category from the IMAGE model must exist in order to create these images. Because the IMAGE categories in the database are already used by other images, we recommend the user **to create** a new one (example: **Image_DTM)** in a **Data Model interface**

⇒ Generating Grey Level DTM Images

- *# Start – Programs – Spring 3.2 – Spring()* *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **dtm-grd** *from the category* **Altimetria**
- [DTM][Image Generation...]

⇒ Generating a DTM Grey Level image

DTM -> Image

- (Image ⇔ Gray Level)
- (Output category...)

  Categories list

  - (Categories | Imagens_DTM) - Category user defined
  - (Apply)
- {Output layer: ima-dtm}
- (Apply)

⇒ Generating Shaded DTM Images

DTM -> Image

- (Image ⇔ Shaded)
- (Output category...)

  Categories list

  - (Categories | Imagens_DTM) - Category user defined
  - (Apply)
- {Output layer: som-dtm}
- {Azimuth (degrees): 45}
- {Elevation (degrees): 45}
- {Height Exaggeration: 10}
- (Apply)

- *Visualize the images just created*

3.2 Slope and Aspect Maps Creation

For any spatial position **Slope** is the maximum slant, related to the horizontal plane, calculated for the position. This means the maximum variation rate of the elevation in each position. The slope can be presented in degrees (0 to 90°) or in percents (%). The **Aspect** is the slope direction and is measured in degrees (0 a 360°).

The Slope and the Aspect are the two components of the gradient vector, respectively, the absolute value and the direction. These components are calculated from the first and second derivatives obtained directly from the grid (rectangular and triangular) models.

IMPORTANT: It is recommended that the user create two new NUMERICAL model categories (example: **Grd_decli** and **Grd_exp**), in order to accommodate the new infolayers created in the previous exercise.

⇒ ***Slope Map Generation***
- *# Start – Programs – Spring 3.2 – Spring(*⊞) *#spring (* **UNIX** *)*

SPRING-3.2
- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **dtm-grd** *of the* **Altimetria** *category*
- [DTM][Slope...]

⇒ ***Generating slope em degrees or in percents from a rectangular grid***

Slope
- (Input ⇔ Grid)
- (Output ⇔ Slope)
- (Unit ⇔ Percentage) or (Unit ⇔ Degrees)
- (Output category...)

    Categories list
    - (Categories | Grd_decli) - Category user defined.
    - (Apply)

Slope
- {Output layer: dec-grd}
- (Apply)

⇒ ***Generating aspect from a rectangular grid***

Control panel
- *Activate the infolayer* **dtm-grd** *of the* **Altimetria** *category*

Slope
- (Input ⇔ Grid)
- (Output ⇔ Aspect)
- (Output category...)

    Categories list
    - (Categories | Grd_exp) - Category user defined.
    - (Apply)

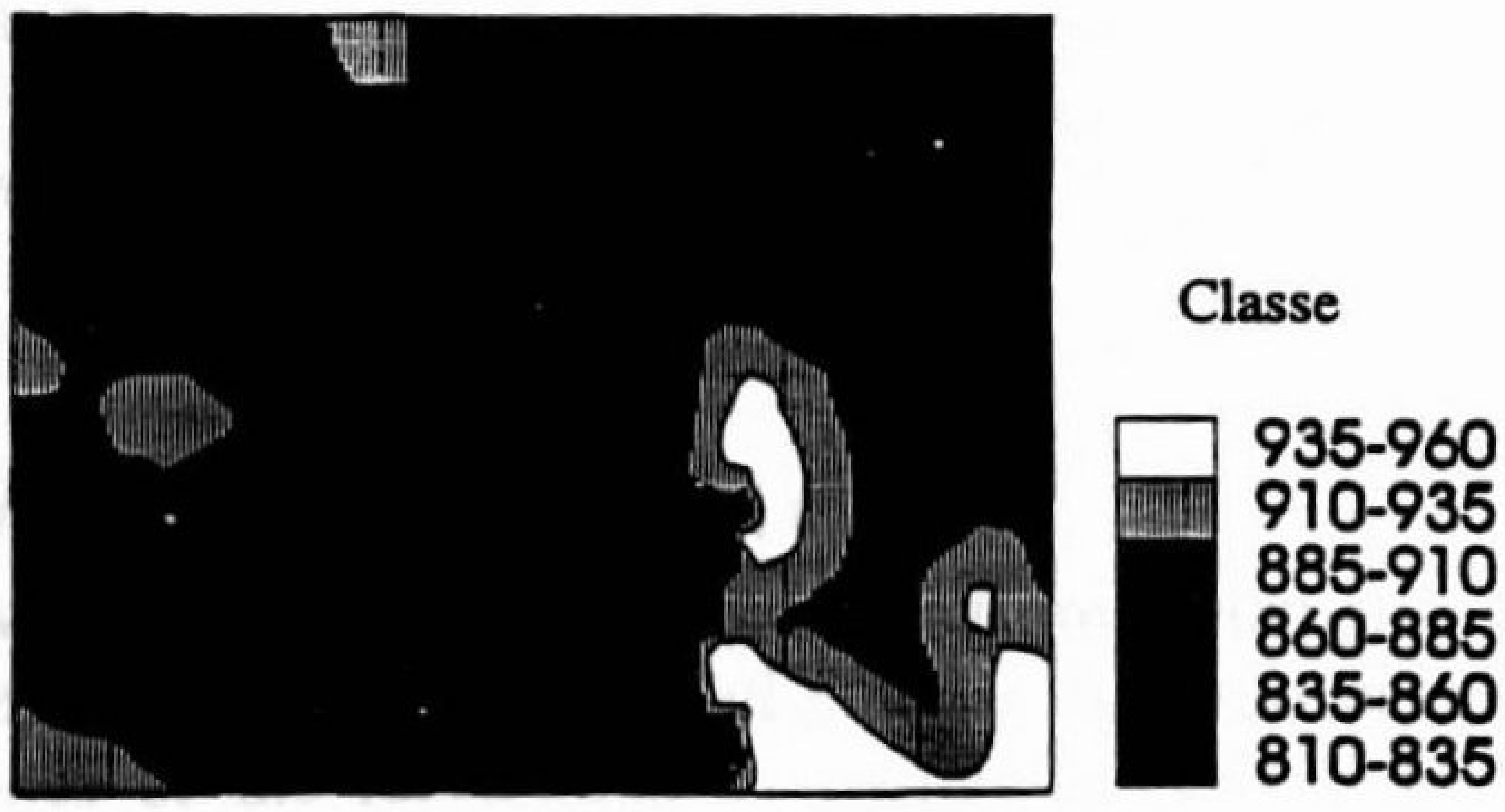
Slope
- {Output layer: exp-grd}
- (Bounding box...)
- *Select an area on the layer*
- {X(m): 100}, {Y(m): 100}

- (Apply)
- *Test the slope generation using a triangular model*
- *Test the aspect generation using a triangular model*

IMPORTANT: In order to obtain thematic maps with slope or aspect classes the user must run the slicing process that will be explained below.

3.3 Digital Terrain Model Slicing

The purpose of *DTM slicing* is to create thematic images from rectangular or triangular models. The thematic image classes correspond to intervals, named *slices* in the SPRING environment, of the total z variation for the surface. The classes, that will be associated to the slices, must be created along with a category belonging to a THEMATIC model. The figure below illustrates this concept.



⇒ ***Rectangular Grid Slicing***

- *# Start – Programs – Spring 3.2 – Spring()* *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **dtm-grd** *of the* **Altimetria** *category*
- [DTM][Slicing...]

  DTM Slicing

  - (Output category...)

    Categories list

    - (Categories | Hipsometria) *Go to the **Data Model** and create the Hipsometria category .
    - (Apply)

  DTM Slicing

  - {Output layer: fat-dtm-grd}
  - (Slice Definition...)

    Slice Definition

    - (Step ⇔ Fixed)
    - {Start: <min Z value>}
    - * see the *Zmin and Zmax value* in dtm-grd infolayer [Edit] [Infolayer] or buttom (Representation)

- {End: <max Zvalue>}
- {Step: 50} or (max Z – min Z)/<number of slices desired>
- (Insert)
- (Apply)

⇒ ***Associating slices to thematic classes***

DTM Slicing

- (Associating Slices with Classes...)

Slices-Classes Association

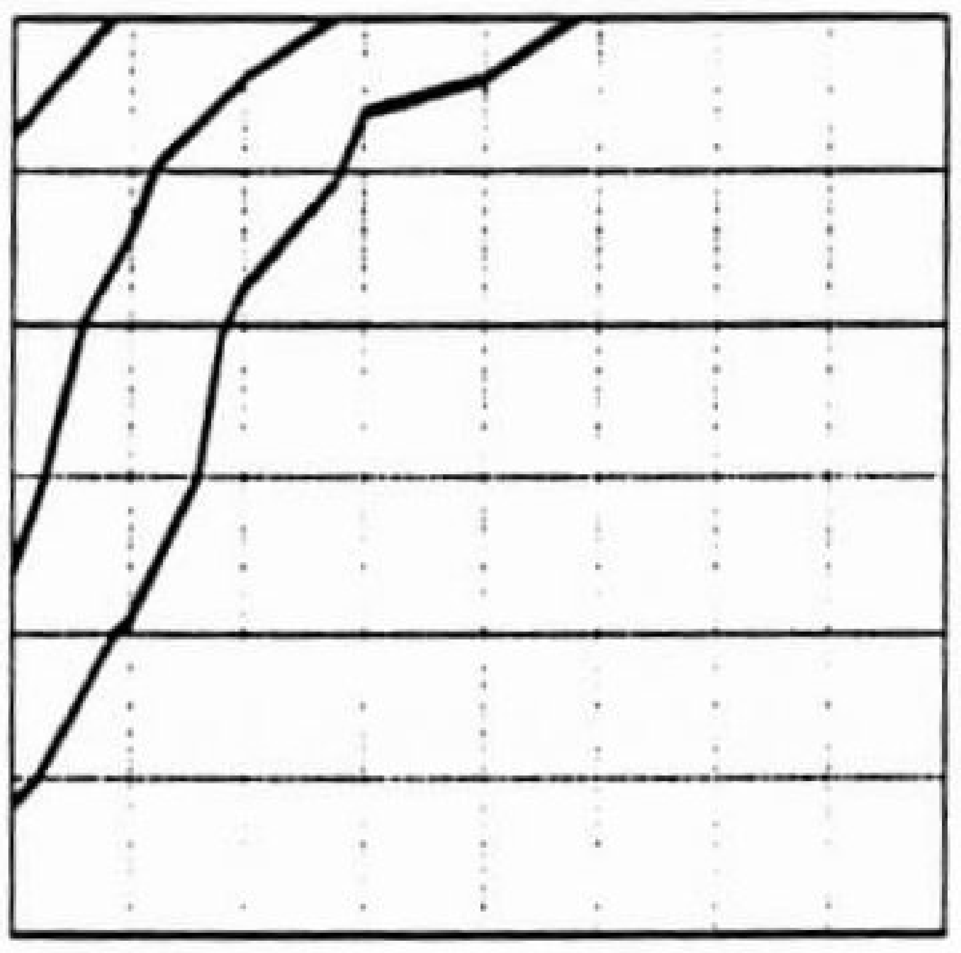
- (Classes | 1) **if there is **none** classes defined, go to the* **Data Model** ▦ *and define thematic classes to Hipsometria category.*
- (Association: Slices -Classes | classe1 -> 1)
- *Repeat the association process for other slices*
- (Apply)

DTM Slicing

- (Apply)
- *Visualize the layer* **fat-dtm-grd**

3.4 Automatic Isoline Maps creation from DTM models

The SPRING allows the creation of *isolines maps* from rectangular and triangular digital terrain models. The method used, called *cell method,* is based in the intersection of each isoline with the sides of the rectangular or triangular cells from a DTM model. This process creates segment lines that are connected to generate *close isolines* or *open isolines*, when the extremes of the line reach the limits of the infolayer area.



⇒ ***Generating Isolines Maps***

- *# Start – Programs – Spring 3.2 – Spring(*▦) *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **dtm-grd** *of the* **Altimetria** *category*
- [DTM][Contour Generation...]

⇒ ***Generating isolines in the canvas with fixed step***

Contour Lines Generation

- (Generate ⇔ Screen)
- (Input ⇔ Grid)

- (Step ⇔ Fixed)
- {Vmin: Zmin} * see the Zmin and Zmax value in dtm-grd
  [Edit] [Infolayer] or button
  (Representation)
- {Vmax: Zmax}
- {Step: 500}
- (Insert)
- (Apply)

⇒ ***Generating de isolines in the canvas with variable step***

Contour Lines Generation

- (Generate ⇔ Screen)
- (Input ⇔ Grid)
- (Step ⇔ Variable)
- {Value: 100}
- (Insert)
- *Insert the values* **250, 300, 500, 750, 800 or other values according to your data**
- (Apply)

- *Test also with input TIN*
- *Test also the Contour generation on file*

3.5 Profiles analysis

The profile analysis is performed over predefined trajectories. The trajectories can be edited or the user can choose them from line data elements stored in the database. The lines from stored data must belong to infolayers defined for categories of the THEMATIC, CADASTRAL or NETWORK model.

⇒ ***Profile***

- *# Start – Programs – Spring 3.2 – Spring()* *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **Mapa_altimetrico** *of the* **Altimetria** *category*
- [DTM][Profile...]

⇒ ***Generating profile from trajectory edited on the canvas***

Profile

- (Input ⇔ Grid)
- (Trajectory ⇔ Editor)
- (Lines ⇔ Create)
- *Digitize trajectory on the canvas - P.S. Up to 5 trajectories.* ***Press the left bottom of the mouse to edit the trajectory points and the right bottom to end the edition.***
- {Graphic Title - Profile}
- {Axis Y: Z Value}
- {Unit: m}
- (Apply)

⇒ ***Generating profile from trajectory defined by infolayer lines***

- *Visualize the infolayer* **Mapa_vias**, *of the* **Vias_acesso** *category, that contains the trajectory lines.*
- *Activate the infolayer* **Mapa_altimetrico** *of the* **Altimetria** *category*

Profile
- (Input ⇔ Grid)
- (Trajectory ⇔ Layer)
- (Model ⇔ Thematic)
- (Layer...)

  Categories and Layers
  - (Categories I Vias_acesso)
  - (Information Layers I Mapa_vias)
  - (Apply)

Profile
- (Select Trajectory)
- *Select a trajectory line on the canvas*
- {Graphic Title - Profile2}
- { Y axis: Z Value}
- {Unit: m}
- (Apply)

3.6 Volume Calculation

The volume calculation is performed over closed polygons that belong to infolayers from categories of the THEMATIC AND CADASTRAL models. Besides the polygons, the volume calculation requires a triangular or rectangular terrain model. The volume calculation is based on a numerical integration of the DTM values inside the polygon areas. The *cut and fill volumes* are evaluated in relation to a user defined Z value, named here as the Z base. The cut volume is the volume below the Z base while the fill volume is the volume above the Z base. The *ideal Z value* is the Z value that equals the fill and cut volumes.

⇒ ***Volume***

- *# Start – Programs – Spring 3.2 – Spring()* *#spring (* **UNIX** *)*

SPRING-3.2
- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Visualize the layer* **Mapa_rios** *that contains some polygons*
- *Activate the infolayer* **relevo** *of the* **Altimetria** *category*
- [DTM][Volume...]

  Volume Calculation
  - (DTM Data - Grid)
  - (Layer...)

    Categories and Layers
    - (Categories I Drenagem)
    - (Information Layers I Mapa_rios)
    - (Apply)

  Volume Calculation

- {Base: 600} *Z value that you intend to cut
- (Volume ⇔ Corte/Aterro)

⇒ ***Calculating volume for a specific polygon***

Volume Calculation

- (Calculation Option ⇔ Partial)
- *Select a polygon on the canvas using "double-click"*
- (Apply)
- (Save...)
- *Select or define the file name to save the volume information*
- (Copy...)
- *Select the file name that contains volume information*

Important: Apply again with the **Base** value suggested by the system

⇒ ***Calculating volume for all the polygons***

Note: don't do this exercise, because it takes a long time

Volume Calculation

- (Calculation Option ⇔ Total)
- (Apply)

3.7 Planar projection of the DTM with texture

The **planar projection of the DTM with texture,** named **3D viewing** in the SPRING menu, requires the use of two maps: the *relief model* and the *texture image*. The relief model must be a rectangular grid model while the texture image must belong to infolayer from an IMAGE model category. The data from the relief model is used to calculate the transformations that create the planar projection. The texture map will cover the surface given to it a realistic appearance.

⇒ ***Visualization 3D***

- *# Start – Programs – Spring 3.2 – Spring(*⊞) *#spring (* **UNIX** *)*

SPRING-3.2

- *Activate the database* **Curso**
- *Activate the project* **Brasilia**
- *Activate the infolayer* **Mapa_altimetrico** *of the* **Altimetria** *category*
- [DTM][3D Viewing...]

3D Viewing

- (Texture Layer...)

Categories and Layers

- (Categories | Image_TM)
- (Image Layers | TM5_Realce) – *as the texture image*
- (Apply)

⇒ ***Parallel projection viewing***

3D Viewing

- (Projection ⇔ Parallel)
- {Azimuth: 225}
- {Elevation: 45}
- {Vertical Exaggeration: 0.4}
- (Apply)
- *Test with other azimuth, elevation and vertical exaggeration values*

⇒ ***Perspective projection viewing***

3D Viewing

- (Projection ⇔ Perspective)
- {X: 400000}, {Y: 7442000},{Z: 250}
- {Azimuth: 130}
- {Aperture: 60}
- {Vertical Exaggeration: 0.5}
- (Apply)
- *Test with other values*

⇒ ***Visualizing in parallel stereo projection***

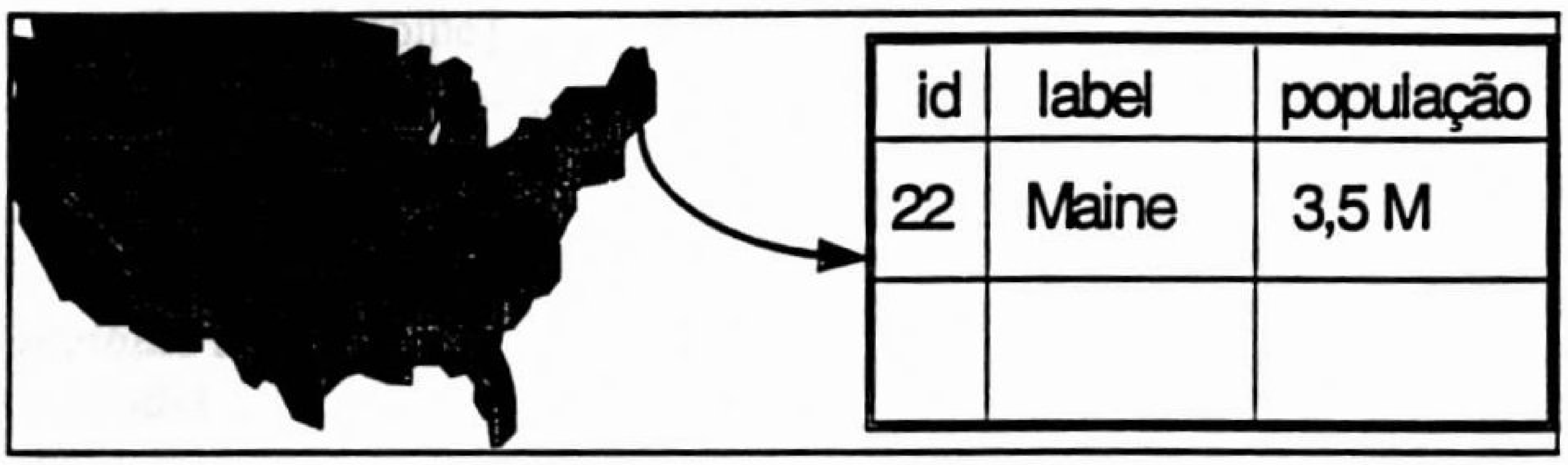
3D Viewing

- (Texture Layer...)
- *Select infolayer* **relevo-som** *as the image of texture*
- (Projection ⇔ Stereo Pair)
- {Azimuth: 225}
- {Elevation: 30}
- {Distance among Projections.: 5}
- {Vertical Exaggeration: 1}
- (Apply)
- *Test with other values*

Class 8 - Spatial Analysis and Data Base Query

1. Querying a Geographical Data Base

SPRING stores the conventional attributes of geographical objects as tables in a relational data base management system (RDBMS). In order to allow the connection between the location and the descriptive attributes, each spatial object has a unique identifier **(id)**. When a geographical database is queried, both spatial and relational subsystems are searched and the result is a composition of results. The link between a relational database and the spatial component is shown in Figure 1.



id	label	população
22	Maine	3,5 M

In this part of the class, the user will learn how to query a geographical database. Initially, he will define two data types, one of the "Object" category and one of the "Cadastral" category". The spatial file containing the lines, which define the cadastral map, will be imported, but the user will do the labeling of the polygons.

1.1 Data Base Definition (Cadastral e Object models)

⇒ ***Creating data of types Cadastral e Object***
SPRING-3.2

- [File] [Data Model...] or icon
  - Data Model
  - - {Categories - Name: Blocks}
  - - (Model ⇔ Object)
  - - (Categories ⇔ Create)

Creating Categories of Cadastral type

- **Data Model**
  - - {Categories - Name: Urban_Cadastre}
  - - (Model ⇔ Cadastral)
  - - (Categories ⇔ Create)
  - - (Execute)

⇒ ***Defining object attributes***
SPRING-3.2

- [File] [Data Model...] or icon
  - **Data Model**
  - - (Categories ⇔ Blocks)
  - - (Attributes...)
  - ***Defining attributes of type*** **integer**

Category Attributes

- {Name: Population}
- (Type ⇔ Integer)
- (Insert)
- {Name: Schools}
- (Type ⇔ Integer)
- (Insert)

Defining text attributes

- {Name: Type}
- (Type ⇔ Text)
- {Size : 20}

Defining real attributes

- {Name: Income}
- (Type ⇔ Real)
- (Insert)
- (Apply), (Close)

Finishing attribute definition

Data Model

- (Close)

1.2 Generation of a cadastral map and object labeling

⇒ Importing cadastral data (lines in ascii format)

SPRING-3.2

- [File][Import...]

Import

- (Directory...: /springdb/Data)
- (Model ⇔ Thematic)
- (Format | ASCII : Quadras.L2D) ***Quadras=Blocks**
- (Entity | Lines with topol.), (Unit | Meters), {Scale : 25000}
- *Projection and Bounding Box - Not needed, takes from the active project*
- *Project - Nor needed, active project*
- (Category...)

Categories List

- (Categories: Urban_Cadastre or Cadastro_Urbano)
- (Execute)

Import

- {Layer: Blocks_Map)
- (Execute)
- (Close)

⇒ Non-Spatial Attributes Edition

Showing the information layer

Control Panel

- (Categories | Urban_Cadastre)
- (Information Layers| Blocks_Map)
- (Lines)
- [Execute][Draw] *or* ✎

Disabling the cross/area cursor

Control Panel

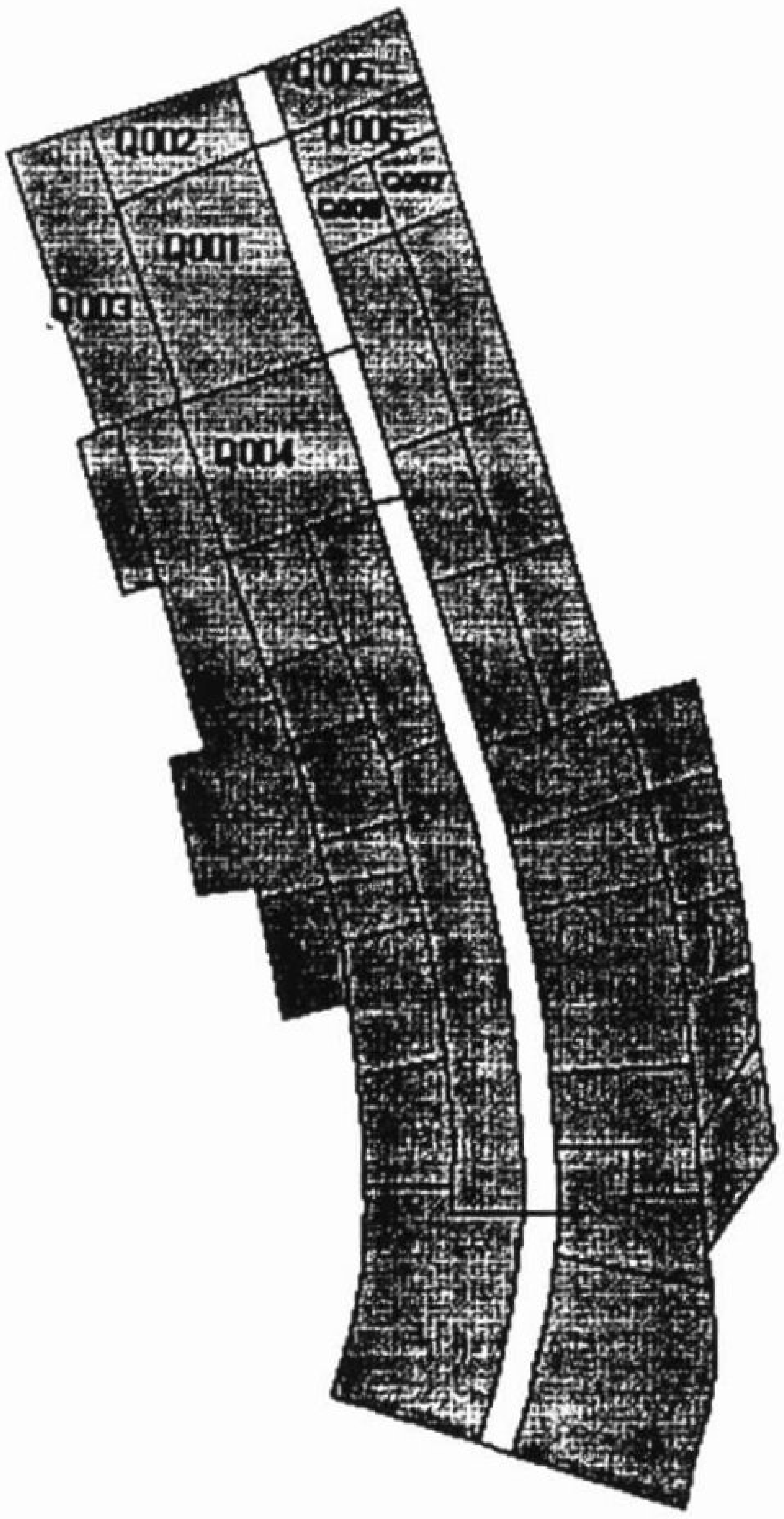
- (Cross/Area Cursor)

Creating and associating objects to their spatial representation

SPRING-3.2

- [Edit][Object...]
  - **Edit Objects**
  - *Creating objects*
    - (<u>Object</u> <u>Classes</u> | Block)
    - {Label: Q001} *Keep Name: in blank
    - (Selection Mode ⇔ Label)
    - (Create)
    - **<u>Associating objetcs with graphical representation</u>**
    - (Operation ⇔ Associate)
    - (Entity ⇔ Polygon)
    - *Click on the object* **Q001** *on screen - Observe the message on the bottom. If it is "Polygon is not found", re-create the topology (Generate Polygons) in*
      - *-[Edit][Vector Editing]*
      - ***Topological Editor***
        - *(Adjust)(Generate Polygons)*
    - (Update)

- Repeat for the other objects using the following map:


Q005
Q002
Q006
Q007
Q008
Q001
Q003
Q004

Editing object attributes

SPRING-3.2

- [Edit][Object...]
  - **Edit Objects**
  - (Object Classes | Block)
  - (Selection Mode ⇔ Screen)
  - *Click on an object on screen (canvas window)*
  - *Observe the parameters Label and Name in the Edit Objects Interface.

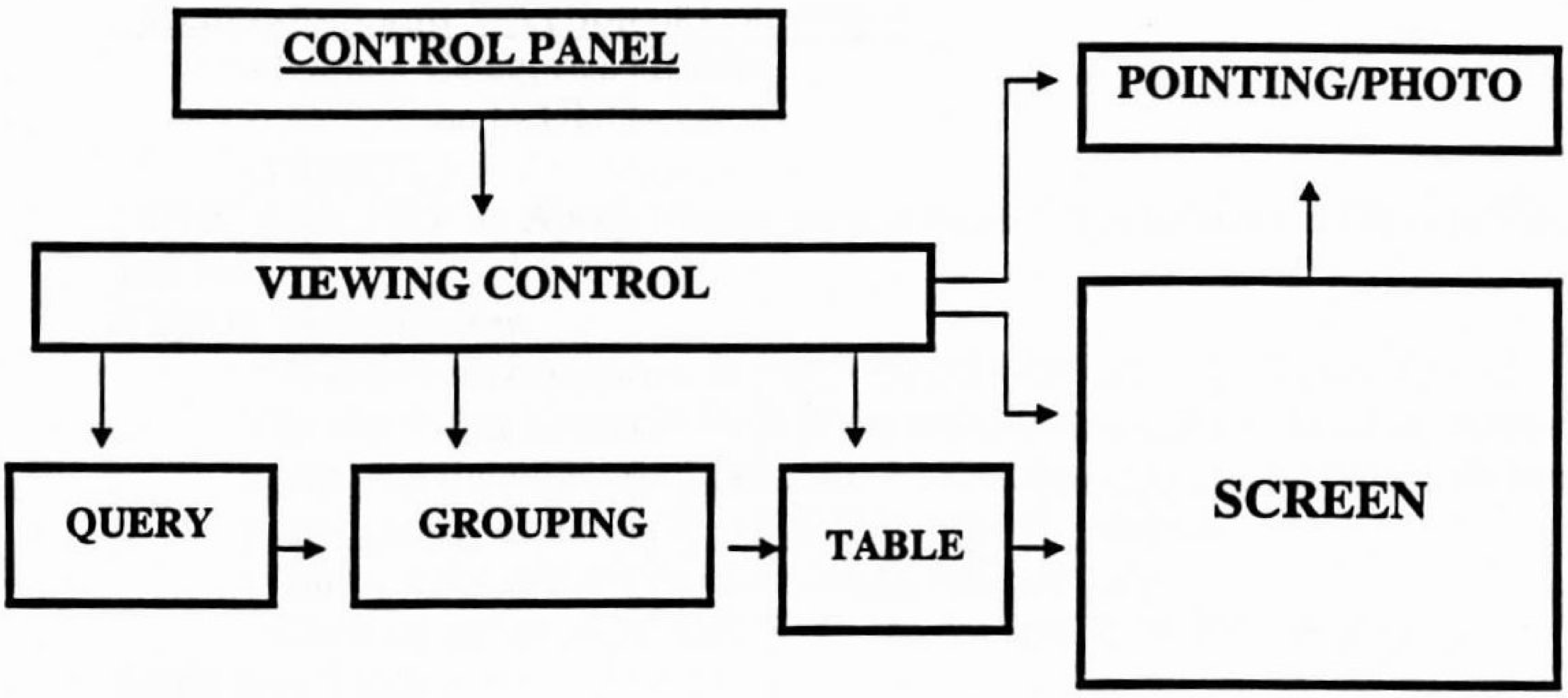
⇒ ***Editing conventional object attributes***

- **Edit Objects**
- (Attributes...)
  - **Attributes**
  - (Attributes | Population)
  - {Value: 500}
  - (Attributes | Schools)
  - {Value: 2} *number of schools by block
  - (Attributes | Type)
  - {Value: Residence} *type of block: Industrial, or Hotels, or Leisure
  - (Attributes | Income)
  - {Value: 2476.51}
  - (Execute)

- Repeat these two items for all objects (Q001- Q007) giving different values for Schools, Type, Income attributes.

1.3 Querying the DataBase

The query function is controlled in the **"Control Panel"**, and include five main modules: **Viewing Control, Query, Grouping, Table and Pointing/Photo**, as illustrated in the figure below.



The functions of each module and its interaction with the user as follows.

The **Viewing Control** module control the visualization of objects, indicating which objects is shown and how they are shown. This module is also responsible for controlling legend exhibition and by controlling the graphical presentation sequence, as well as determining the attribute table, which is shown.

The **Query, Grouping** and **Table** modules modify the graphical presentation of objects, as follows:

- **Query** module: filters the objects, which do not satisfy user-defined criteria.
- **Grouping** module: group objects based on one of its attributes, and each resulting group receives a distinctive color-coding.
- **Table** module: shows the attributes in a tabular format e enables the user to select and alter the color coding of any object.

Data Base Query

SPRING-3.2
- *Activate database:* ***Curso***
- *Activate project:* **Brasilia**

⇒ ***Showing objects for the query operation***
Control Panel
- (Categories I Urban_cadastre or Cadastro Urbano)
- (Information LayersI Blocks_Map)
- (Lines)
- (Execute)(Draw) or ✎
- (Objects)

⇒ ***Selecting objects by pointing***
Control Panel
- (Query...)

Generation and Selection of Collections
  - (Object Categories I Blocks)
  - (Collection I TUDO -ALL)
  - (Apply...)

NOTE: After click on Apply bottom, two interfaces will be shown: Objects Visualization and Table.

Objects Visualization

* It shows all categories of object model selected in the Control Panel.
The *check-box* button in front of the categories's name is used to activate and deactivate the objects visualization of each category, and the *triangle* button is used to show or hiding the LEGEND of each category.
- Click on the *check-box* of the BLOCKS category.
- Click on the *triangle* button and see the legend of this category.

Attributes Table

* It shows all attributes of the category selected in the interface: Objects Visualization.

⇒ Selecting and deselecting an object (pointing on SCREEN)

- Don't forget to *disable point/area cursor (if active).*

- Point and click inside a polygon (block, Q001-Q007) on screen and observe that the object in the TABLE interface will be highlighted with the green color (the green color can be changed if you click on the upper left corner of the TABLE interface with the right bottom mouse).

- Click again on the same polygon on the screen and observe that its color will be changed to BLUE- default color and in the TABLE interface this object will not be highlighted anymore. So, it is possible to select or deselect an object on the screen and on the table pointing and clicking on it.

⇒ Selecting and deselecting an object (pointing on TABLE)

- Click on the first column of the TABLE interface and observe that the same object in the screen will be highlighted with the green color.

- Click again on the first column of the TABLE interface and observe that this object will not be highlighted anymore and the color of the respective object on the screen will be changed to BLUE- default color (or the color defined in DATA MODEL interface (visual bottom)).

⇒ Attributes table

Table

⇒ Saving attributes Table in a ascii file

- (File | Save Table...)

⇒ Showing all or just the selected objects

- (Show | All)
* *Show all objects of this category.*
- (Show | Selected)
* *Show just the objects selected by the Query module*

⇒ Selecting the color

- Click on the upper left corner with the right mouse bottom
- Select the color.

⇒ Selecting many lines on the Table

- Click on the first column of the table with the left mouse bottom and drag vertically until the last desired line, or
- Select the first line,
- Scroll until the page that contain the last line deisred,
- Press the *SHIFT* key and click on the last line.

NOTE: These operations highlight the objects on the TABLE interface and its representation in the SCREEN in a current color. If the objects was selected before, these operations deselect them in the TABLE interface and on the screen they are showed with the category color . In this example, the category color is blue (it can be changed through the Data Model interface – visual bottom).

⇒ Showing Statistics (just for numerical attributes)

- Press the right bottom mouse on the numerical attribute name (first line of the TABLE interface)
- (Statistics...)

⇒ Ordering the TABLE by attribute

- Press the right bottom mouse on the attribute name (first line of the TABLE interface)
- (Ordering)
- (Increasing) or (Decreasing)

⇒ Showing or Hiding Columns

- Press the right bottom mouse on the attribute name (first line of the TABLE interface)
- (Show Columns...)
- Click on the columns to select/deselect

* The highlighted columns will be shown

- (Apply)

⇒ Showing Histogram (just for numerical attributes)

- Firstly deselect all lines selected before.
- Press the left bottom mouse on the attribute name (first line of the TABLE interface) *it will select the column.
- Press the right bottom mouse on the attribute name (first line of the TABLE interface) and select the operation
- (Graphics...) *the histogram will be shown

⇒ Showing Scatter Plot

* *Firstly deselect all lines selected before and select two numerical columns.*

- Press the left bottom mouse on the **first** attribute name (first line of the TABLE interface) *it will select the column.
- Press the left bottom mouse on the **second** attribute name (first line of the TABLE interface) *it will select the column.
- Press the right bottom mouse on the one of the attributes selected (first line of the TABLE interface) and select the operation
- (Graphics...) *the Scatter Plot will be shown

⇒ Showing Pie Chart

- Select at least one line.
- Press the right bottom mouse on the attributes name (first line of the TABLE interface) and select the operation
- (Graphics...) *the Pie Chart will be shown

⇒ Selecting objects by the Aggregation module

Object Visualization

- [Edit][Aggregation...]

Aggregation Objects: Blocks

- (Attributes | Income)
- (Equal Steps)
- (5 steps)

- (Apply)
- *Analyze result*

* *Test other grouping options*

* ***Reset** aggregation before closing the window*

- (Reset)
- (Apply)
- (Close)

⇒ Attribute Selection

Object Visualization

- [Query...]

Object Selection (by query sentence)

- (<u>Attributes</u> | Income)
- (<u>Operation</u> | >)
- (<u>Values</u> | 2000) - *or other value, according to the data given by the user.*

* *Observe that the logical expression is shown during your selection.*

- (Apply)
- *the selected objects are shown on the active screen according to the defined expression.*

* ***Cancel** the query before closing the interface:*

- (Cancel) – *click many times to remove all components of the query sentence.*
- (Close)

⇒ Inserting a JPEG, GIF, HTML files or URL's as a new attribute of the Object.

SPRING 3.2

- Select a graphical representation of one object on the screen by double-clicking on the respective polygon with the left mouse bottom. A new interface will be shown with all attributes name.

Attributes: BLOCKS

- Press the right bottom mouse anywhere on this interface and select the operation.
- (Insert ⇔ JPEG/GIF/HTML...)

Open File

- Select a file of one of these formats.
- (Open)

* Observe on the interface the new attribute (PHOTO/HTML)

- Press the right bottom mouse anywhere on this interface and select the operation.
- (Insert ⇔ URL...)

Insert URL address

- { URL : http://www.inpe.br} or www.inpe.br
- (Apply)

* Observe on the interface the new attribute (PHOTO/HTML)

⇒ ***Showing JPEG, GIF, HTML files or URL's.***

Attributes: BLOCKS

- Press the right bottom mouse on the line PHOTO/HTML and select the operation,
- (View)

⇒ ***Deleting JPEG, GIF, HTML files or URL's associated with the object.***

Attributes: BLOCKS

- Press the right bottom mouse on the line PHOTO/HTML and select the operation,
- (Delete)

* *The table below is a resume of the principals query options, as follows:*

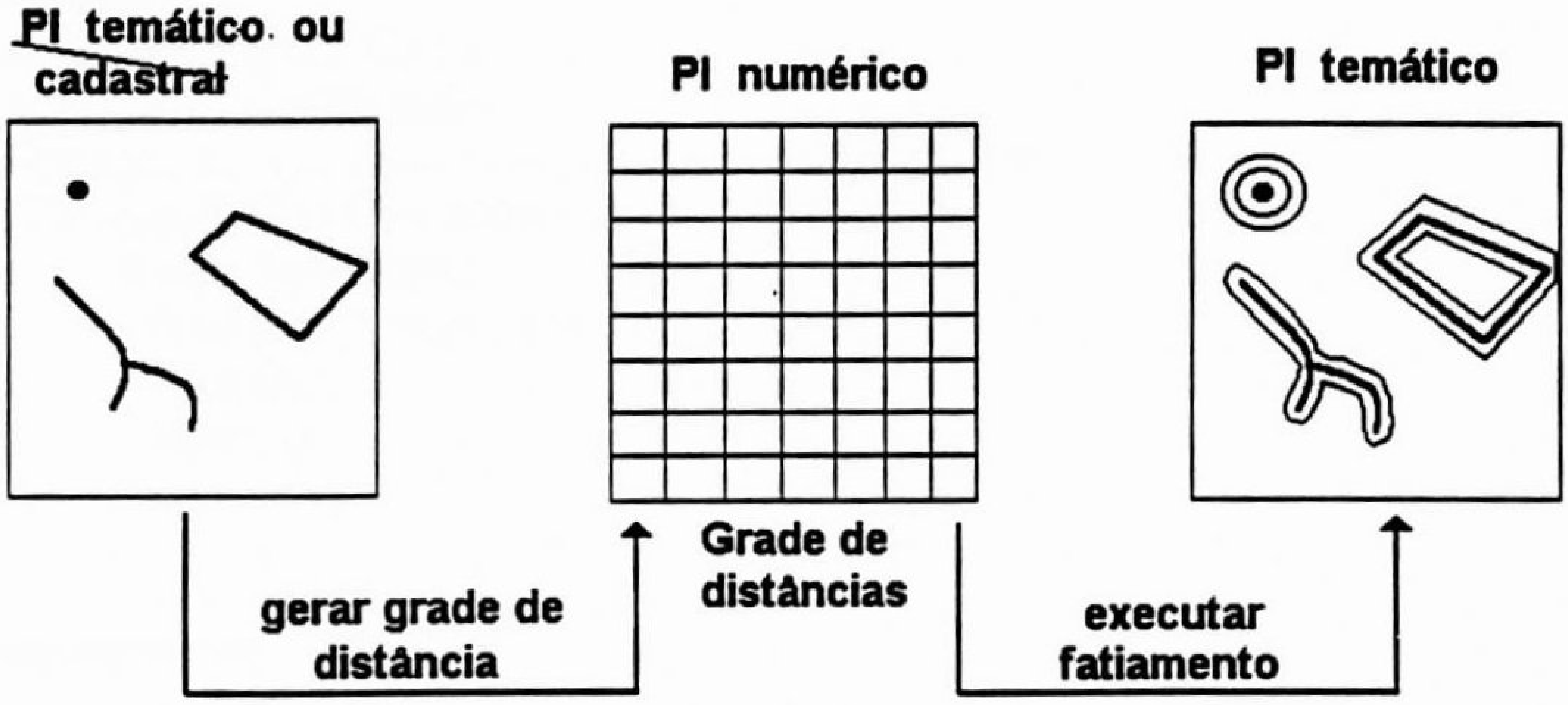
Mouse Action	**Pointer Location**	**Action/Result**
Click (LB)	First column	Selects an object on the table with the color defined in the color rectangle, or deselect the object and select the graphical representation of this object on the screen.
Click and Drag vertically (LB)	First column of Many lines on the table	Selects many objects on the table (or deselect a previous choice) and on the screen.
Click (RB)	Upper left corner	Change the color box by clicking on it
Click and Press with (RB)	Attribute name	Shows Operations: Statistics, ordering, graphic and Select /Deselect visible columns.
Click (LB)	Attribute name	Select the whole column
Click (RB)	Attribute name /Graphics...	Graphical interface with a histcgram of attribute values
Select one column and select some lines on table clicking and dragging the left button	Attribute name /Graphics...	Graphical interface with a pie chart
Select two columns and click on the second column with (RB), choose the graphics option.	Attribute name (second column) /Graphics...	Graphical interface with a Scatter Plot

LB: Left Bottom RB: Right bottom

2. Spatial Analysis

2.1 Buffer Map

A buffer map shows a set of geographical regions, which are contained, within a specified distance of selected geographical objects, as shown in the figure below.



In order to obtain a buffer map, please follow these steps:

1. Identify the graphical element (point, line or polygon) which will be used as a reference for computing the distances.
2. Create a numerical grid of values which represent the Euclidean distance from the selected object
3. Slice the grid in relation to the desired buffer zones.

⇒ *Generating a Buffer Map*

SPRING-3.2
- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Visualize the layer* **RiversMap** *of category* **Drainage**
- [Thematic][Buffering..]
  Distance Maps
  - *Select reference objects*
  - (Category...)
  - *Select numerical output category*
  - {PI: distancemap}
  - {X(m): 200}, {Y(m):200}
  - (Execute)

- *Visualize the output grid*
- *Slice the grid to obtain a map of buffer zones*

2.2 Area Calculation

⇒ *Executing an Area Calculation*

SPRING-3.2
- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Visualize thematic layer* **Soil_Map** *from category* **Soils**
- [Thematic][Area Calculation...]
  Area Calculation
  - (Thematic Image), (Vectors)
  - (Execute)
  - (Save...)

2.3 Measurements

Distance between two points

Indicate two points with the mouse, and SPRING will calculate the Euclidean distance between them.

⇒ ***Executing measurements***

SPRING-3.2

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Visualize any information layer*
- [Tools][Measurements...]

  Measurements

  - (Option ⇔ Distance
  - *Click on any two points in the active screen. For each pair of points the distance values are calculated*
  - (Close)

Polygon Area or Perimeter

The "Measurements" tool enables the user to calculate the area and the perimeter of any polygon shown on thematic or cadastral maps.

Executing area and perimeter measurements

SPRING-3.2

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- *Visualize any information layer*
- [Tools][Measurements...]

  Measurements

  - (Option ⇔ Polygon)
  - *Click on any polygon in the active screen, and its area/perimeter values are displayed.*
  - (Close)

2.4 Cross Tabulation

The cross tabulation operation enables the user to calculate the intersection area of two thematic information layers (in raster format), with the same horizontal and vertical resolution, and the same number of lines and columns. It compares the classes of these two layers, determining the distribution of their intersection.

⇒ ***Executing a cross tabulation operation***

SPRING-3.2

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- ***It is not required to have a layer on the active screen, but the desired layer must be the active one. Activate the Thematic layer Soils_Map of category Soils.***
- [Thematic][Cross Tabulation...]

Cross Tabulation

- (Intersection Layer...)

Categories and Information Layers

- (Categories | Land_Use)
- (Information Layers| Land_Use_Map)
- (Execute)

Cross Tabulation

- (Execute)

** The result is shown on a separate window.*

3. LEGAL

LEGAL stands for "Linguagem Espacial de Geoprocessamento Algebrico" (Spatial language for Algebraic Geoprocessing). It provides an infrastructure to describe spatial analysis models as sequences of declarations, instantiations and operations represented as "algebraic" expressions similar to the well known arithmetic, relational and boolean expressions involving Thematic, Digital, Image and Object data model types. A program in LEGAL is a sequence of sentences separated by ";" edited manually to a text file provided in the Spring main window as follows

Spring-3.2 window

[Edit][LEGAL]

Algebra window

The sentences in a program can be grouped in three classes:

Declarations:

in this part, the working variables of the program are defined. Each variable must be explicitly declared; the user must give a name and associate it to an existing category in the Data Model.

Instantiations:

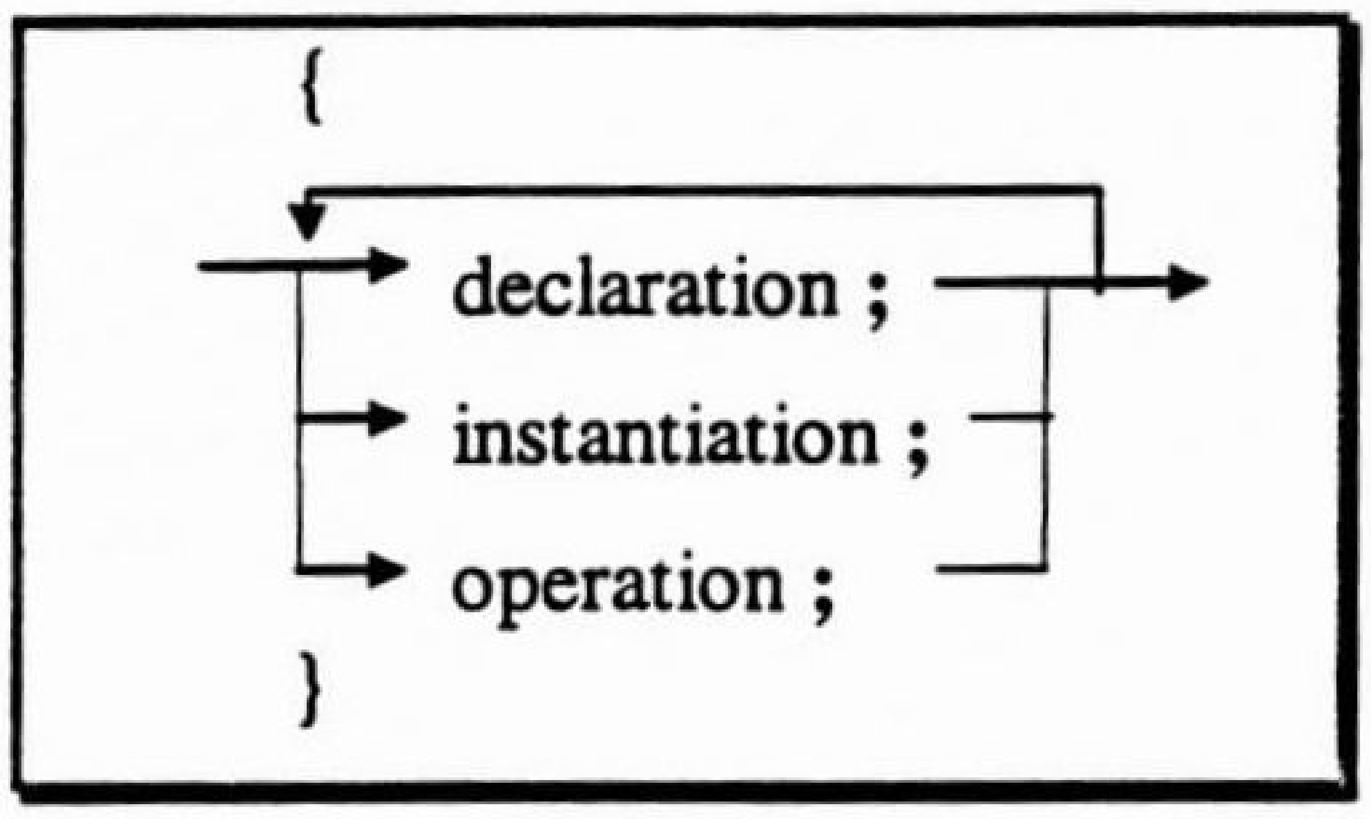
in this part, the current data is retrieved or new layers are created, which can be associated to the results of operations in LEGAL.

Operations:

In this part, the Map Algebra operations are actually performed

Sentences in LEGAL may involve **symbols** (for example, '{', '(', ';', ','), **operators** (for example, '+', '*', '&&', '||' , '<', '<=', '!='), **reserved words** (for example, **New, Thematic, Name, ResX**), **Variable names** and **Data names** (layers). All layer names, categories and thematic classes must be written within quotes (""). All reserved words must start with capitals (for example, **Thematic**).

Program structure:



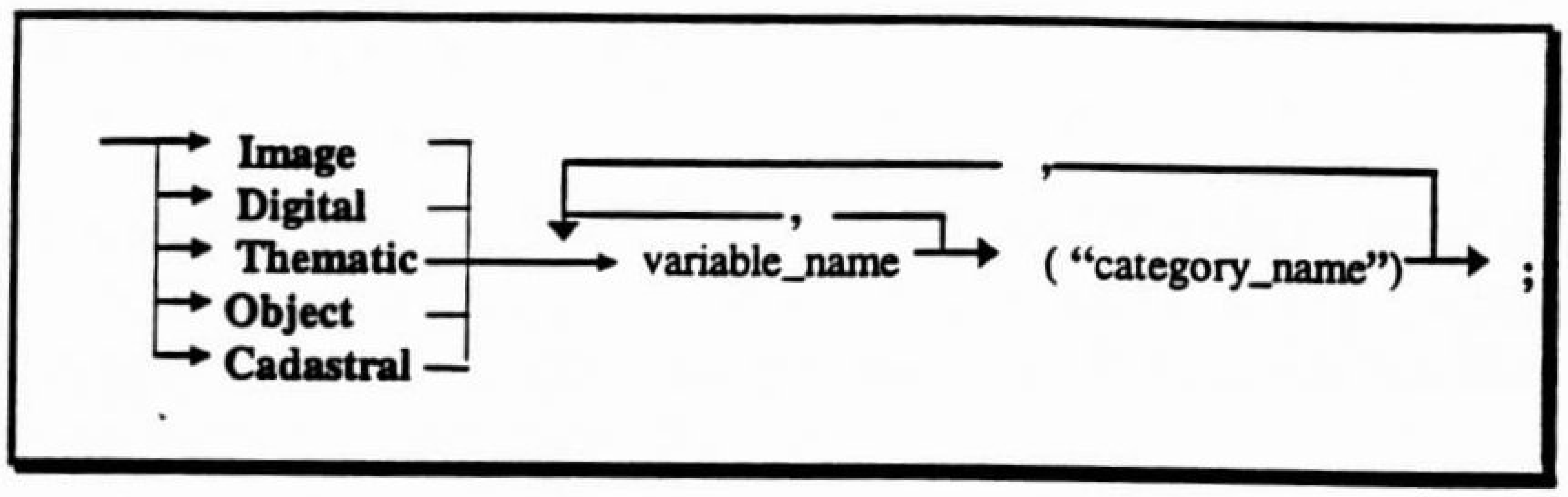
Valid sentences in the language counts a very big number so that its functionality must be described by some grammar description schema, In the syntactic diagram adopted here a sentence is constructed by simply writing any sequence of items that occurs while following the arrows from left to right, sometimes returning to repeat part of the diagram until reaching the right side of the diagram.

Declaration

All data referred to in a program must be previously defined in a **declaration sentence**. Data will always associated to a **variable** that must be associated to a valid Spring data **model type** and an existing **category**, to indicate the model type and specific category of the infolayer to be associated to it by an **instantiation sentence** later.

The diagram bellow indicates that a declaration must contain one of the reserved words: Image, Thematic, Digital, Object or Cadastral to indicate the **model type**, then a list of one or more **variable names** and the **category** to which it reffers. It also indicates that more than one such arrangements of variable names and categories can be declared to the same data model type in the same declaration sentence. Any kind of such sentences will be terminated by a semicolon (";").

Syntactic diagram for field variable declaration:



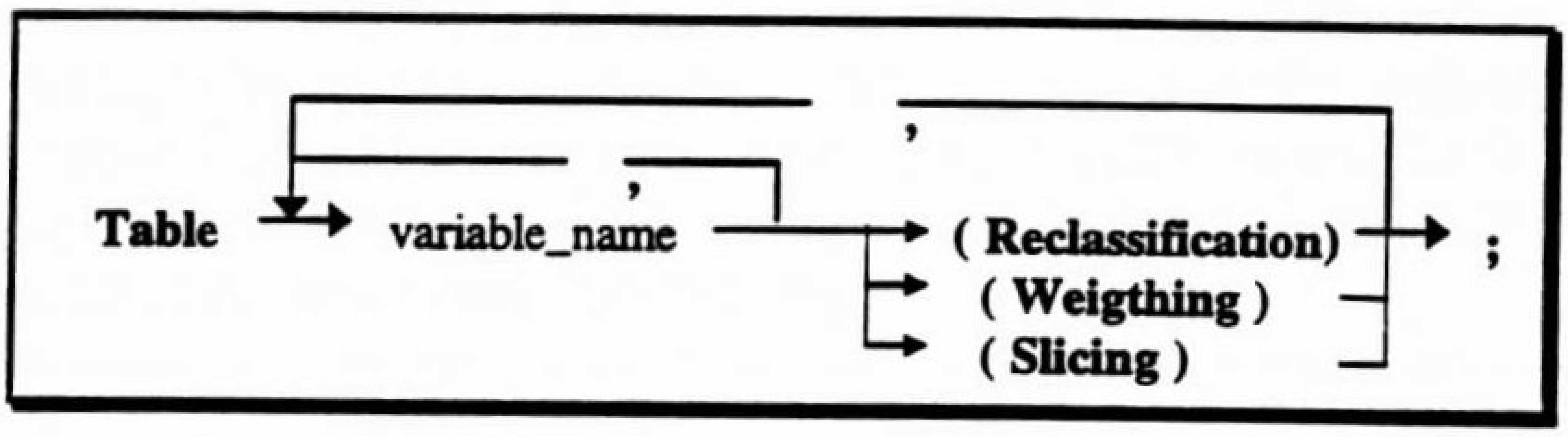
Examples:

Image banda3, banda4, ivdn ("LANDSAT"), pan ("SPOT");

Thematic soilMap("Soil"), geo("geology");

Digital alti1 ("Altimetry");

There is another kind of variable in LEGAL to be considered, the **Table variables**, used along with table operations as we will see later. Those variables acts as mapping rules that associates values of some model type to another model type. The operations based on tables, Reclassification, Weighting and Slicing, are typically transformations between different model types.



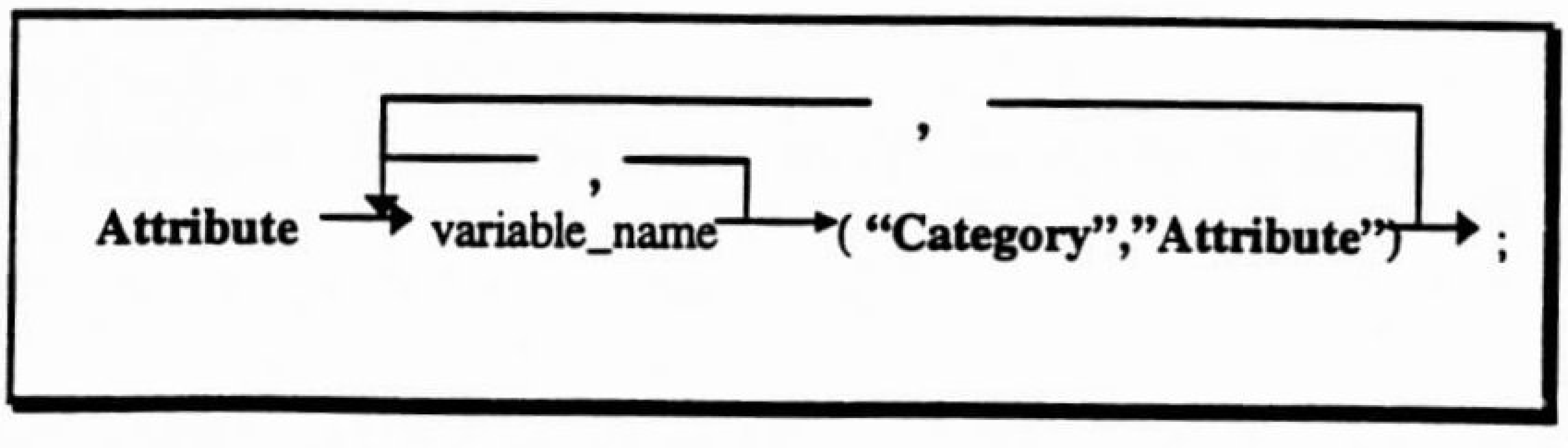
Examples:

Table reclass1, reclass2 (Reclassification);

Table slices (Slicing);

It can be observed here that the reserved words Reclassification, Weighting and Slicing are used in place of the category names for field declarations as showed earlier. The variable concept here is used as an auxiliary tool to implement table operations.

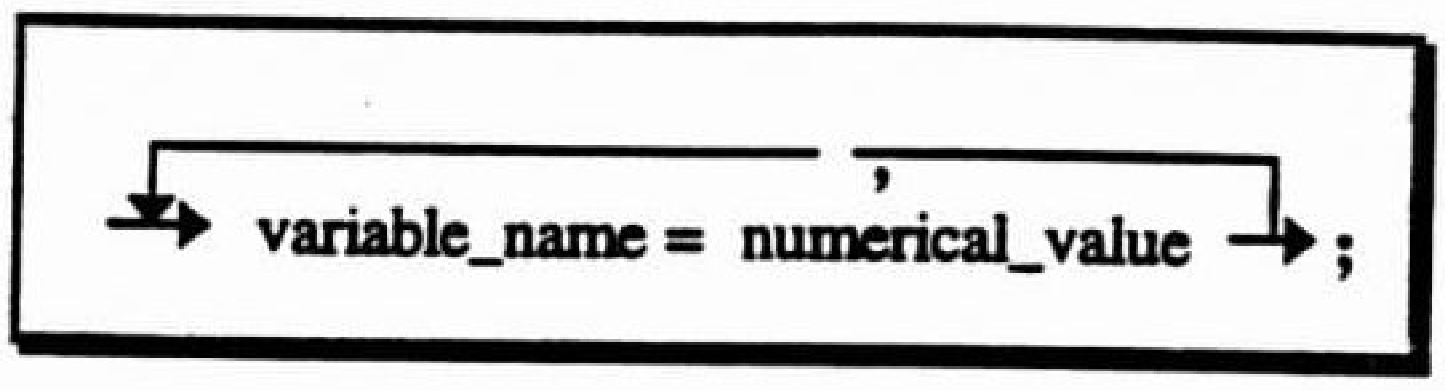
There is another kind of transformation that makes use of the Table variable concept as well as the concept of an **Attribute variable**, the **Reclassification by Attributes,** where a field of type Thematic or Digital can be obtained from the numerical values of some attribute of an Spring geoobject, through direct assignment to a Digital grid or through a mapping, based on a predefined **Slicing** table, into a Thematic map.



Example:

Attribute Values (“Parcels”,”CouncilTax”);

Yet another kind of variable can be defined in LEGAL, the **Number** variables. Those kind of variables refers to numerical values to be used as constants along the program or as variables to be assigned freely through valid number expressions. In practice we can say that its declarations coincide with its instantiation.

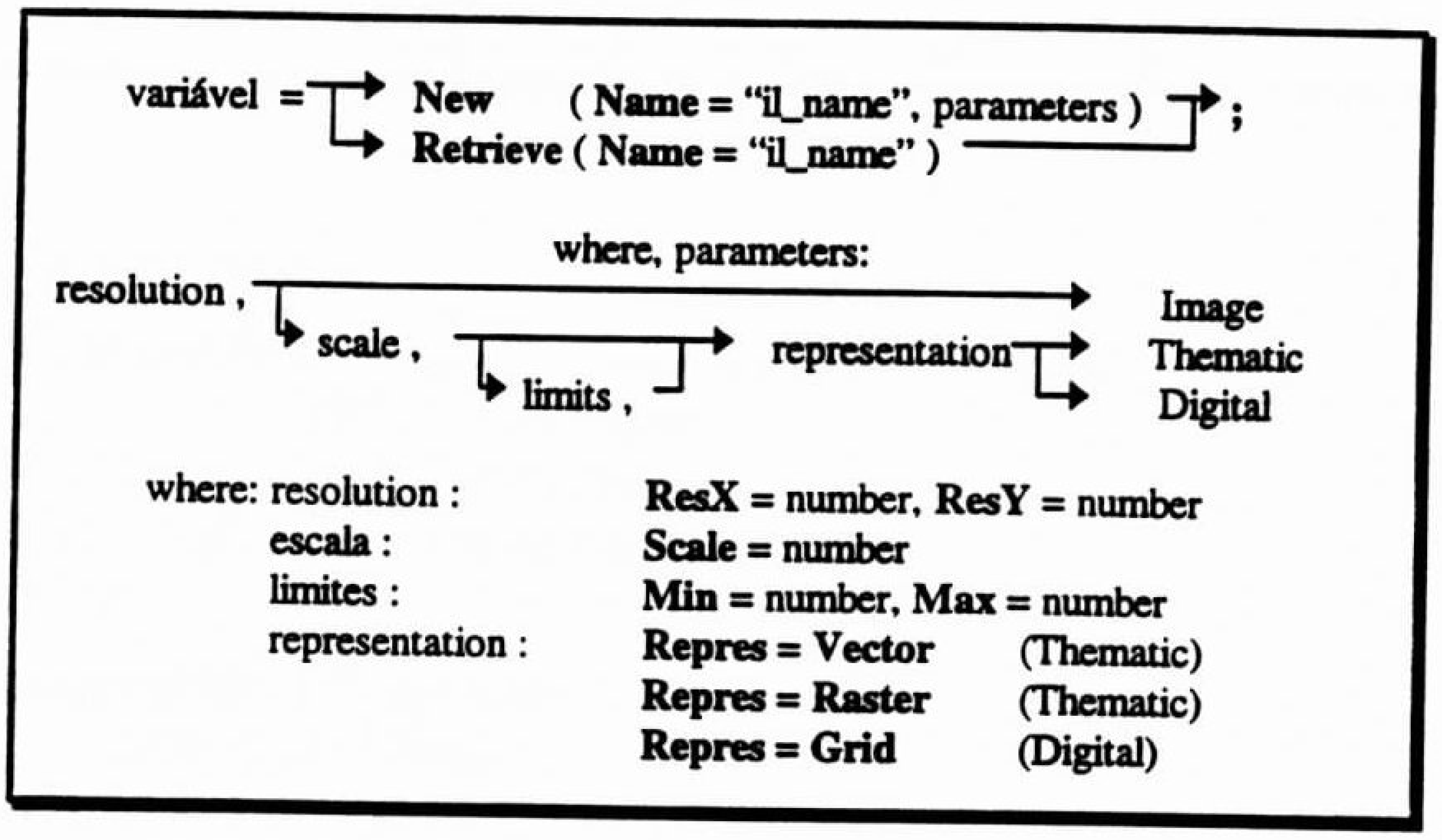


Example:

pi=3.14, d=1.234454;

Instantiation

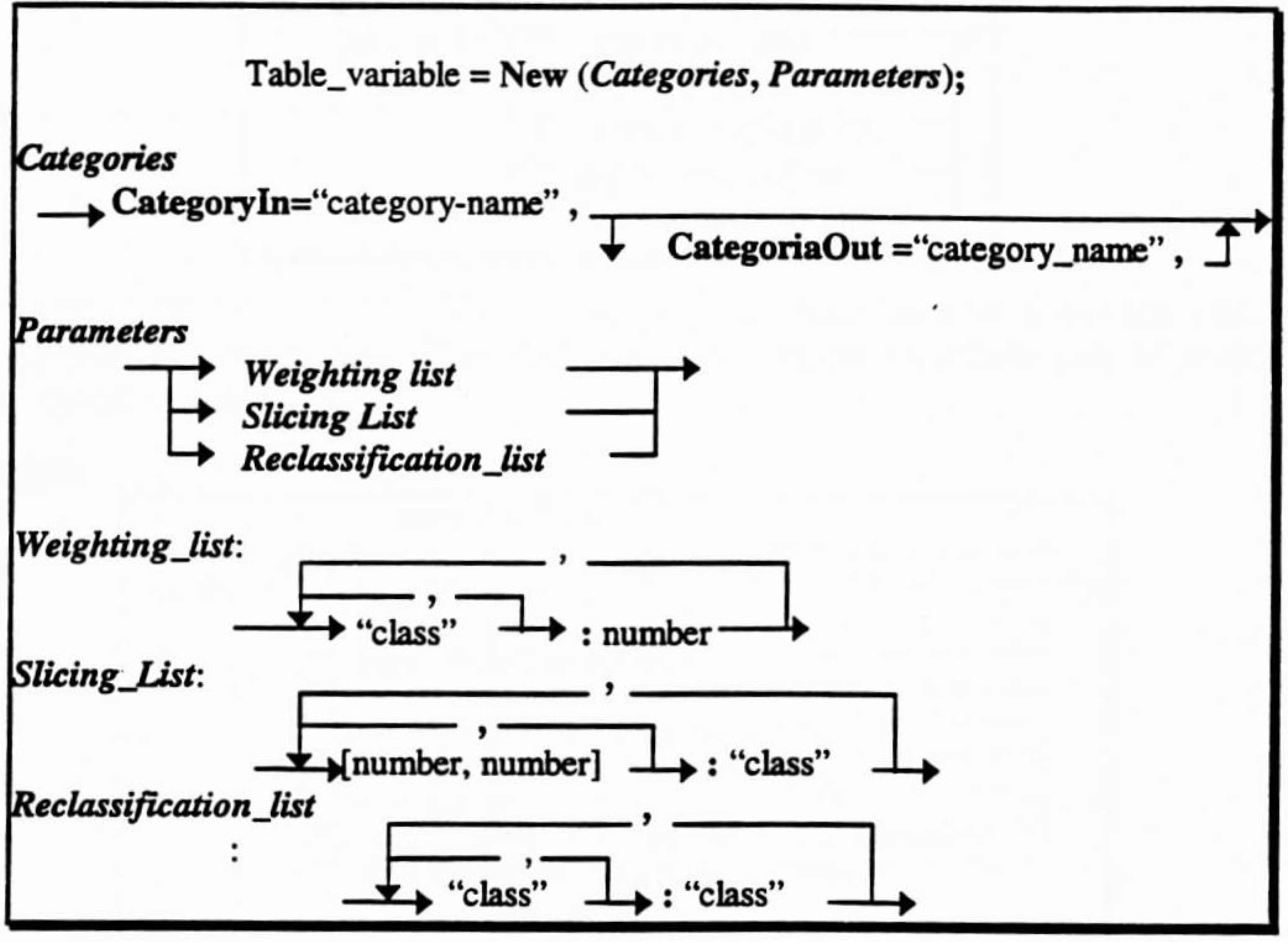
Instantiation consists of the association of a previously declared variable to a real object in the spring environment. It may be a field - an **infolayer** of a valid model type -, a **table** consisting of combined pairs of numerical and/or thematic classes values. Field variables are instantiated through the commands **Retrieve** and **New** whose action consists of the association of an existing infolayer to a variable or the creation of a new infolayer before associating to the variable. Those new infolayers will typically store the results of algebraic expressions involving other variables. The following syntactic diagram describe the rules for constructing sentences for field variable instantiation.



Examples:

tem = **Retrieve** (**Name** = "rivers");
top = **Retrieve** (**Name** = "topography");
imga = **Retrieve** (**Name** = "TM4");
imgb = **New** (**Name** = "TM4a", **ResX**=30, **ResY**=30);
soils = **New** (**Name** = "Soils", **ResX**=50, **ResY**=50, **Scale**=100000);
alt = **New** (**Name**="Altimetry", **ResX**=50, **ResY**=50, **Scale**=1000, **Min**=0, **Max**=100);

The instantiation of Tables consists of the actual creation of the mapping between two model types. The **Reclassification Table** will define the mapping between two sets of geoclasses of maybe two different Thematic Categories. The **Weighting Table** will define the mapping between geoclasses of a Thematic Category and a set of numerical values to be assigned to an infolayer of a Digital Category. The **Slicing Table** defines the mapping of ranges of numerical values found in an infolayer of Digital model type into a set of geoclasses of a Thematic Category. The following syntactic diagram shows the valid paths to be followed while instantiating tables.



Example of reclassification table:

Reclass1 = **New(CategoryIn** = "Vegeta", **CategoryOut** = "Vegeta",
"Of" : "DenseForest",
"Db", "Ds1", "Ds2", "Ds4", "Dm" : "WetForest",
"sd", "sp", "sA" : "Savanna");

Example of slicing list:

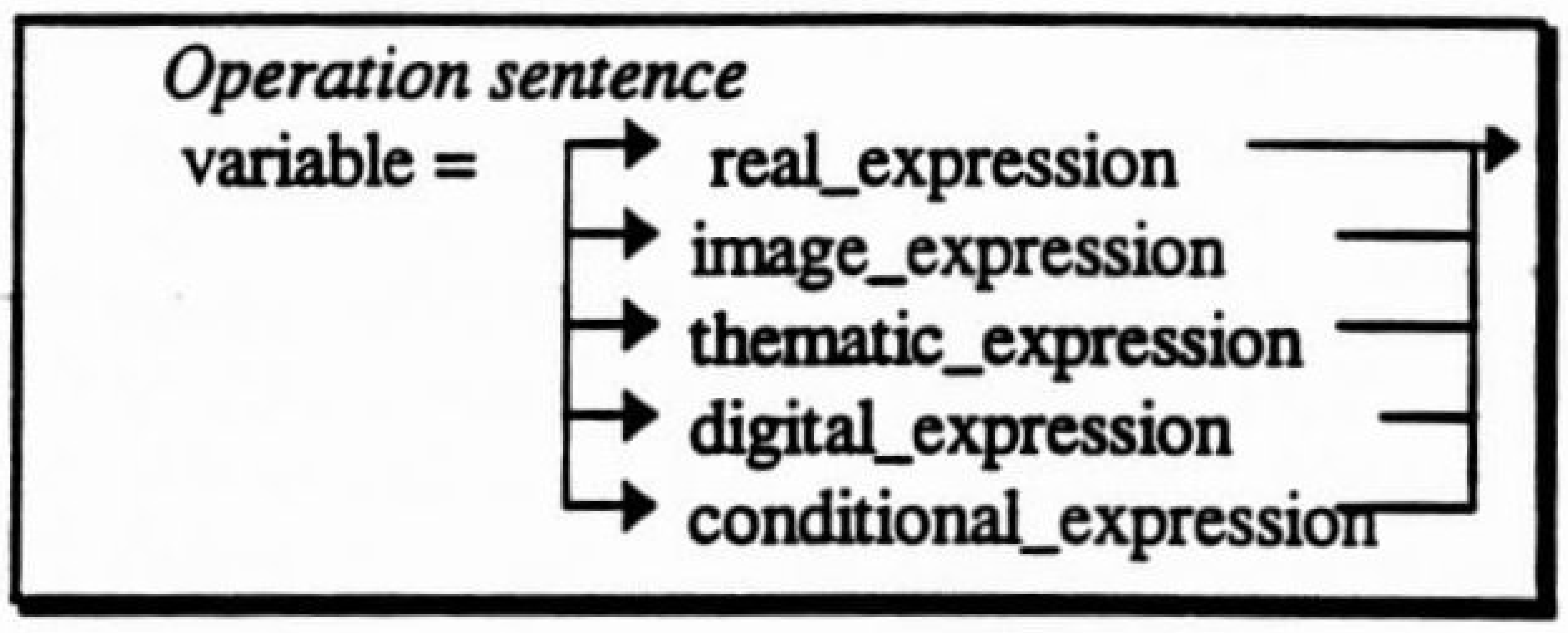
Slices = **New** (**CategoryOut** = "Vegetation",
[0.0, 0.2] : "Forest",
[0.2, 0.45], [0.8, 1.0] : "OpenForest",
[0.45, 0.8] : "Savanna");

Example of weighting table:

weights = **New**(**CategoryIn** = "Vegetation",
"Forest" : 0.2,
"OpenForest" : 0.43
"Savanna") : 0.456);

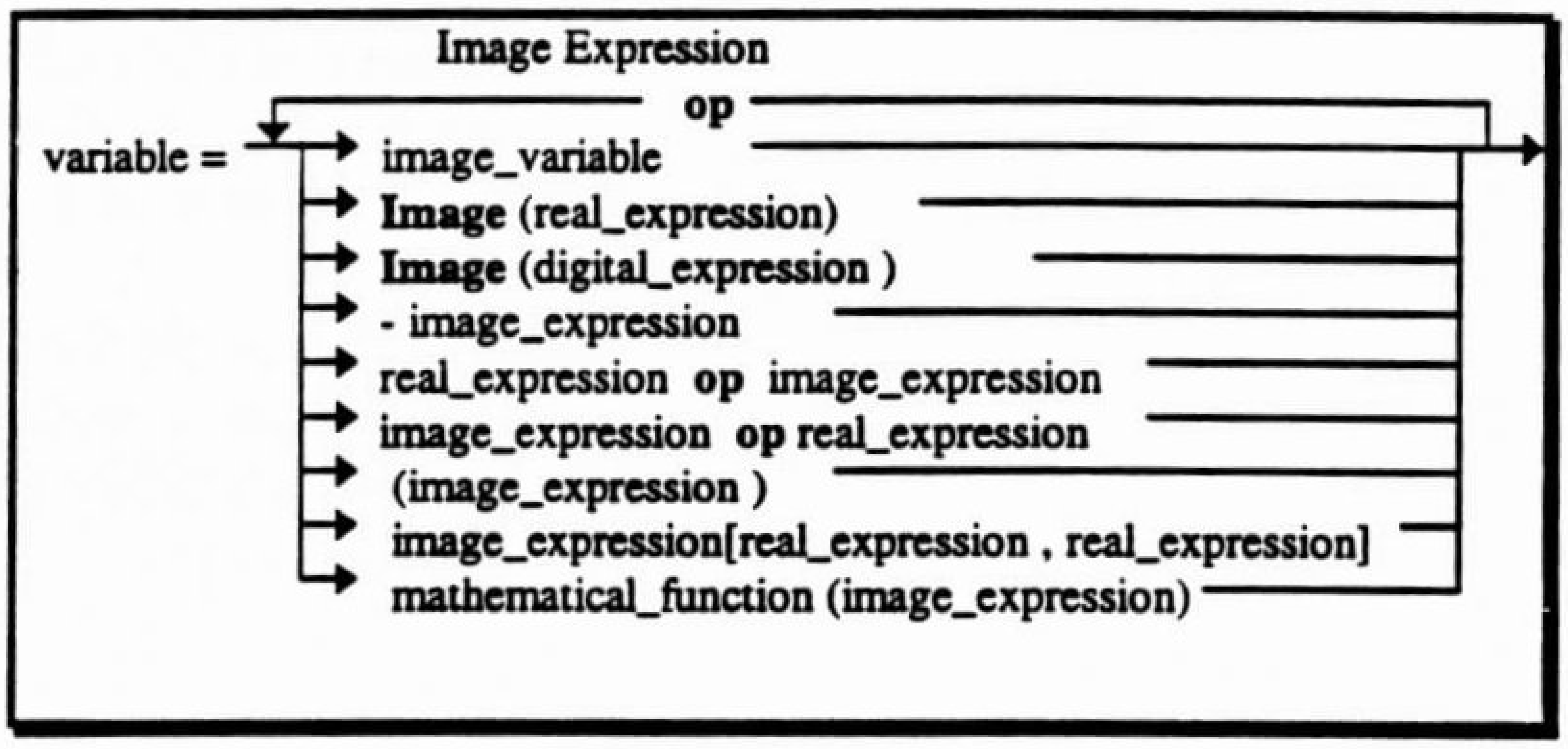
Operation

After the variables have been declared and instantiated, the user will define which operations should be performed over the declared data. In an operation, a variable receives the result of an expression evaluation, which may include previously instantiated variables of many different model types, as seen in the diagram. Result variables as well, can participate of any other expression in an **operation sentence.**



The arithmetic operators '+', '-', '*', '/'e '^', as well as mathematical functions (sin, tangent, etc.), operate over each individual element of the raster representation of some image or digital model type variables.

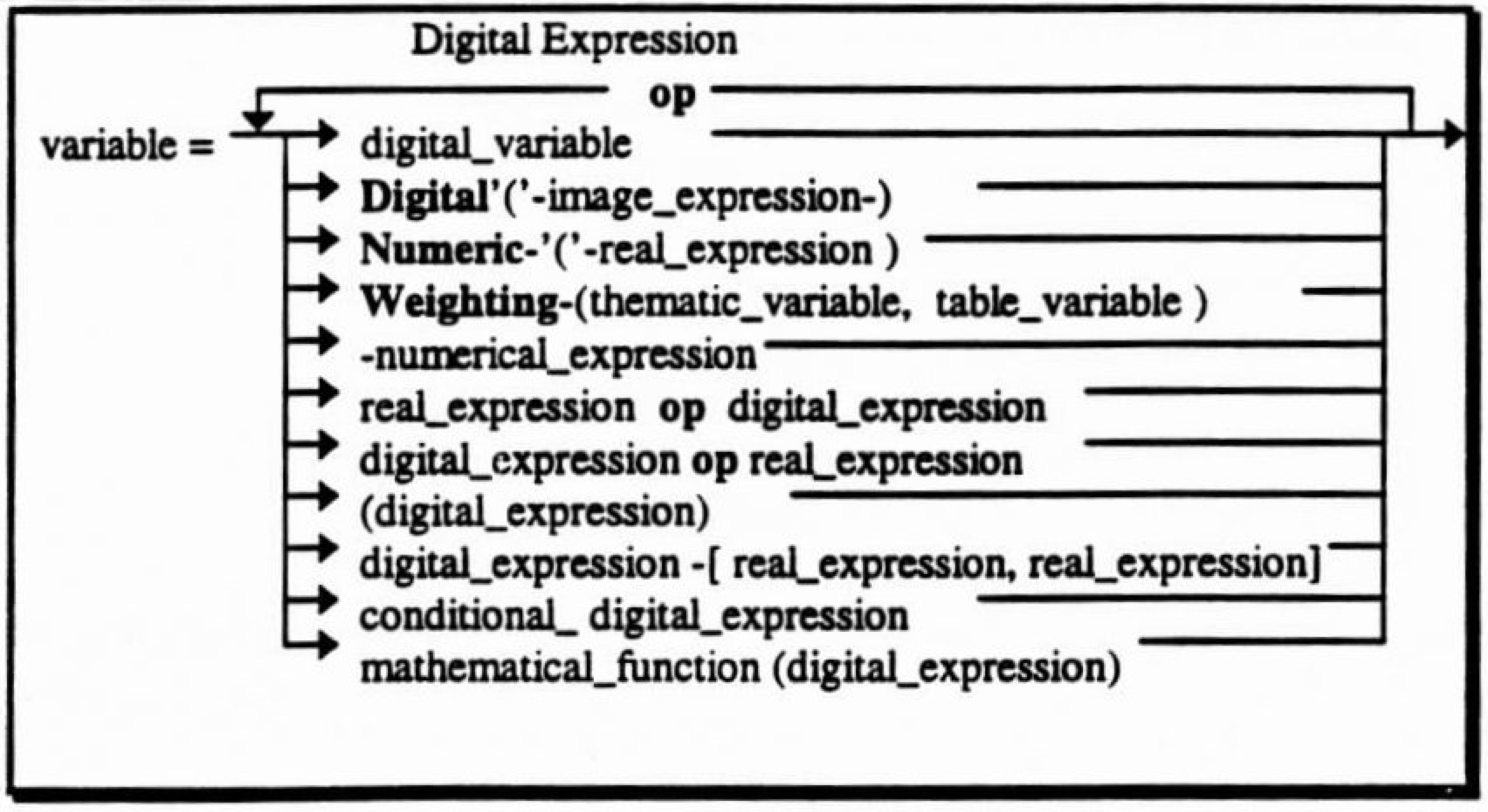
Image Expression



Example of Image expression:

ima1 = **Image**(grade1); //convert grid to image
ima3 = ima2 + 20;
indx = (tm4–tm3) / (tm4+tm3)
reslt = **abs(sin** (ima1) ∗ 255);

Digital Expression

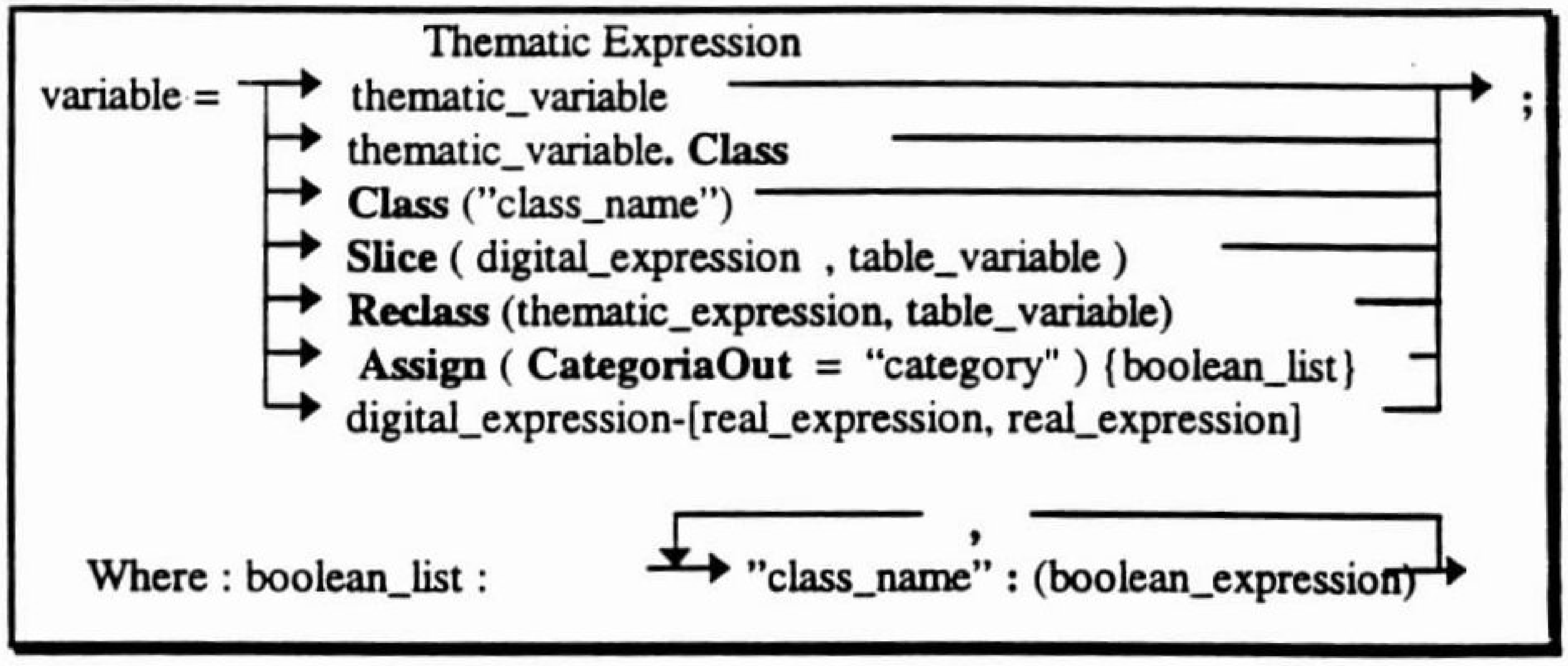


Example of digital expressions:

ph_fe1 = **Digital**(banda_spot2);

grid_sum = (soil_grid + slope_grid)/2;
sin_grid = **sin**(grid1);

Thematic Expression



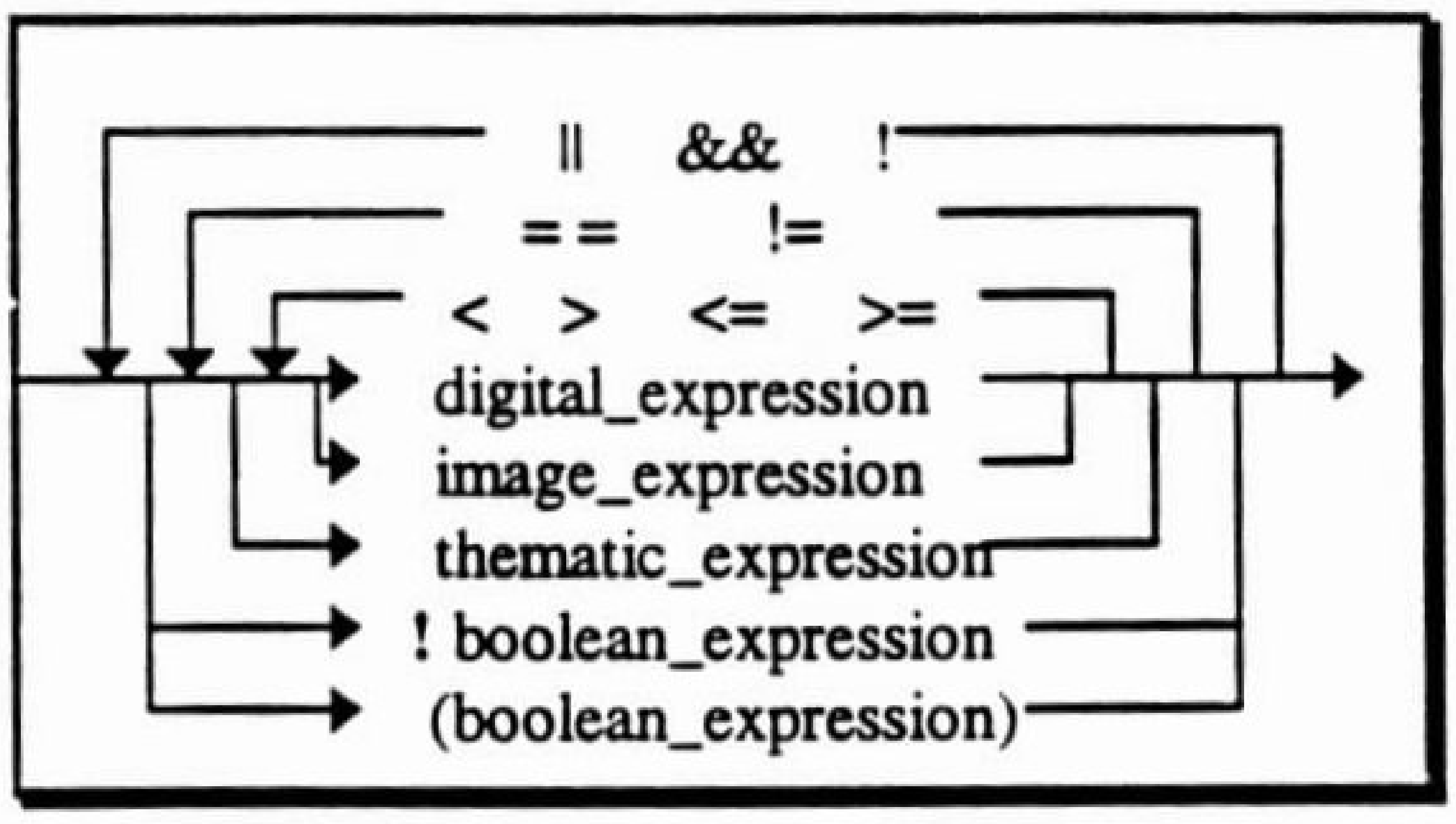
Examples of thematic expressions:

Slope_cl = **Slice**(decliv,tab_decliv);
cut = **Reclass** (cover, tab_recl);
suit = **Assign** (**CategoryOut** = "Suitability")
{
"Good" : (soil.Class == "Lr" &&
slope.Class == "O-3"),
"NotFit" : (soil.Class == "Aq" &&
slope.Class == ">8")};

Boolean Expression

The values of a boolean expression must be either TRUE or FALSE, resulting from a comparison between image or grid values by means of the logical operators '<', '>', '<=', '>=', '==' and '!='; or from the comparison between classes of thematic maps, by means of the operators '==' and '!='.

Boolean expressions can be combined by means of the operators '&&' (AND), '||' (OR) and '!' or '~' (NOT).



The boolean expressions constitutes part of some operation sentences, like those involving conditional expressions.

Conditional Expression

Conditional expression :

variable = → image_conditional_expression →
→ digital_ conditional_expression
→ thematic_conditional_expression

where:

thematic_conditional_expression:
(boolean_expression) ? thematic_expression: thematic_expression
image_conditional_expression
(boolean_expression) ? image_expression: image_expression
digital_conditional_expression:
(boolean_expression) ? digital_expression: digital_expression

Examples of conditional expressions:

Imag_out =(ta.**Class** == "forest") ? **Image** (TM5): 0;

3.1 Editing and Executing a Program in LEGAL

Editing a program in LEGAL is made by using a simple text editor (or by means of an external text editor, such as "**notepad**".

SPRING-3.2

- *Activate database* **Curso**
- *Activate project* **Brasilia**
- [Edit][LEGAL...]
  Algebra
  - *(Directory... :* **/springdb/Data)**
  - {Name: image_cut}
  - (Create...)
    Model Editor
    - {PROGRAM: - *Edit a program showed below*}
    - (Save)

  Algebra
  - (Execute)

  ** in case of a syntax error in the program, the edit window will be displayed, indicating the line where the error occurred.*

  **Observe the category/classes name in the Data Model Interface, correct the errors, and save and execute again (Lake =Lago in Portuguese)*

  *Draw the resultant Layer showed in a Control Panel - lackcut

```
//Use the image TM5 and cut the Lake area creating a new image.
{
Image originalimage,  lakeimage (“ Imagem_TM” );
Thematic Limites (“ Drenagem” );

originalimage = Retrieve (Name = “ TM5” );
lakeimage = New (Name= “ lakecut” , ResX = 30, ResY = 30);
Limites = Retrieve( Name = “ drenagem” );

~~lakecut~~ = Limites .Classe == “ Lake” ? originalimage : 255 ;

}
```

lakeimage

SPRING-3.2

- *Activate database* **Acre**
- *Activate project* **Assis**
- [Edit][LEGAL...]
    - **Algebra**
    - *(Directory... :* **/springdb/acre/Legal)**
    - (Programl Ponderacao) *there are more examples
    - (Edit...)
        - **Model Editor**
        - {PROGRAM: - *Look at the programs sentence*}
        - * change the name of the NEW layers.
        - (Save)
    - **Algebra**
    - (Execute)

This Program transforms Thematic Maps in Digital Maps using Weighting Tables:

```
// Declaration of the variables that will hold the thematic
// input layers of Geology, Geomorphology, Soil, Vegetation(uso-terra)

Thematic  InputGeologia           ("Geologia");
Thematic  InputGeomorfologia      ("Geomorfologia");
Thematic  InputPedologia          ("Solos");
Thematic  InputUso                ("Uso_Terra");

// Declaration of the variable that will hold the
// Weighting Layers of  Geology,  Pedology,
// Use/Vegetation

Digital  GeologyWeighting         ("MNT");
Digital  PedologyWeighting        ("MNT");
Digital  UseWeighting             ("MNT");
Digital  GeomorphWeighting        ("MNT");

// Declaration of the variables that will hold the Weighting tables
// with theme name and numeric values of stability / vulnerability
// associated to each theme [1,3]

Table  TabGeology         (Weighting);
Table  TabPedology        (Weighting);
Table  TabUse              (Weighting);
Table  TabGeomorfologia  (Weighting);

// Instantiation
// Retrieve of Layers:
//     Geology, Geomorphology, Pedology e
//     Use/ Vegetation

InputGeologia   =       Retrieve  (Name= "Geolog2");
InputGeomorfologia  =  Retrieve  (Name= "Geomorfo2");
InputPedologia  =       Retrieve  (Name= "Solos2");
InputUso   =            Retrieve  (Name= "Uso1");


// Creating new empty layers, which will hold the results
// of Weight operation

GeologyWeighting   =      New (Name= "geol_pond", ResX =120, ResY =120,
                                    Scale= 250000, Min = 0, Max = 3);
GeomorphWeighting =      New (Name= "geom_pond", ResX =120, ResY =120,
                                    Scale= 250000, Min = 0, Max = 3);
PedologyWeighting  =     New (Name= "solo_pond", ResX = 120, ResY =120,
                                    Scale= 250000, Min = 0, Max = 3);
UseWeighting =           New (Name= "vege_pond", ResX =120, ResY =120,
                              Scale= 250000, Min = 0, Max = 3);
```

```
//Creating Weighting Tables for each category of data

TabGeology   =  New (CategoriaIni = "Geologia",
                      "Q-Aluviao-Indif" :  3.00,
                      "T-Form-Solimoes" :  2.60);

TabGeomorfologia  =  New (CategoriaIni = "Geomorfologia",
                      "A2-Terraco_fluvial" :              1.00,
                      "B1-Colinosa_Crista_Tabular" :      1.60,
                      "B2-Colinosa_Crista"         :      1.80,
                      "C3-Planicie_Terraco-fluvial":      2.00,
                      "C2-Planicie-fluvial"        :      3.00);

TabPedology  =  New (CategoriaIni = "Solos",
                      "Pvae-Podz_ver_amar_eu"    : 2.00,
                      "Pvaa-Podz_ver_amar_al"    : 2.00,
                      "Hge-Hidro_gley_eu"        : 3.00);

TabUse  =  New(CategoriaIni = "Uso_Terra",
                      "Q-Queimada"                      : 3.00,
                      "AA-Ativid_Agropec"               : 2.80,
                      "C-CapoeiraN"                     : 2.10,
                      "C-CapoeiraV"                     : 1.40,
                      "Fapb-Aber_Palm_Bambu"            : 1.20,
                      "Fapc-Aber_Palm_Cipo_Bambu"       : 1.20,
                      "Faapb-Aluv_Aber_Palm_Bambu"      : 1.20,
                      "Fapd-Aber_Palm-Densa"            : 1.10,
                      "Faapd-Aluv_Aber_Palm-Densa"      : 1.10);

// Operation

GeologyWeighting        = Weight  (InputGeologia, TabGeology);
GeomorphWeighting       = Weight  (InputGeomorfologia, TabGeomorfologia);
PedologiaPonderado      = Weight  (InputPedologia, TabPedology);
UseWeighting            = Weight  (InputUso, TabUse);
}
```

This program uses a Digital map and a cadastral map as a spatial restriction to update the attribute table:

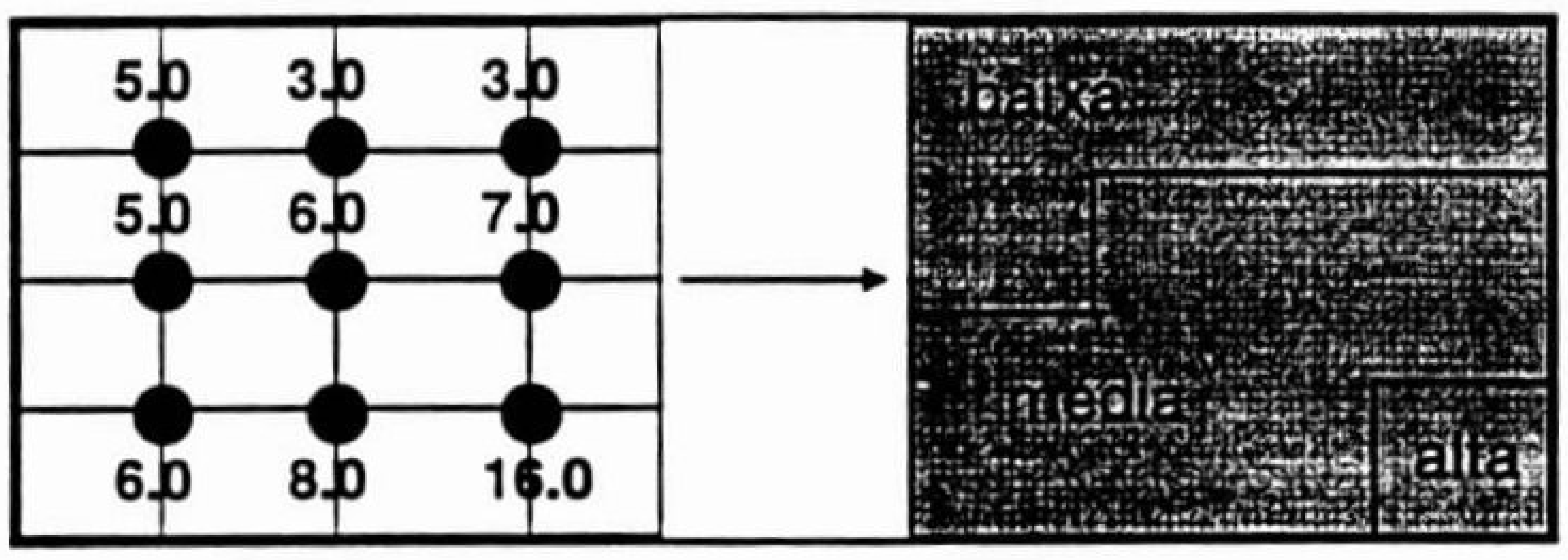
```
{
//This program update the attribute GEOL_VUL using the
//Operator ATUALIZE(Update) and MAIORIA ZONAL (MaiZ)

//Declaration of Variables

Object        zonas ("utb_Obj");
Cadastral     mapacadastral ("utb_Cad");
Digital       geol ("MNT");

//Instantiation

mapacadastral = Retrieve (Name = "utb");
geol = Retrieve (Name = "geol_pond");

//Updating the attribute "GEOL_VUL" of all objects

zonas. "GEOL_VUL" = Atualize (geol, zonas OnMap mapacadastral,
MaiZ);

}
```

3.2 Generic examples of LEGAL Programming

```
//Boolean Operation between two thematic layers
{
//Defining the variables
Thematic soilMap(“Soils”), slopeMap(“Slope”),
         suitMap(“Suitability”);

//Retrieving data
slopeMap = Retrieve (Name = “soil”);
soilMap  = Retrieve (Name = “soil”);

//Creating a New Layer
suitMap = New(Name=“suitb”, ResX=200, ResY=200, Scale=100000);

//Defining the relation between the classes
suitMap = Boolean (CategoryOut = “Suitability”)
      {
      “Good”: ( soilMap.Class      == “Cd1”
                && slopeMap.Class == “0-3” ),
      “Average”: ( soilMap.Class  == “Cd1”
                && slopeMap.Class == “3-8”),
      “Low”: (soilMap.Class       == “Cd1”
                && slopeMap.Class == “8-20”)
      };
}
```

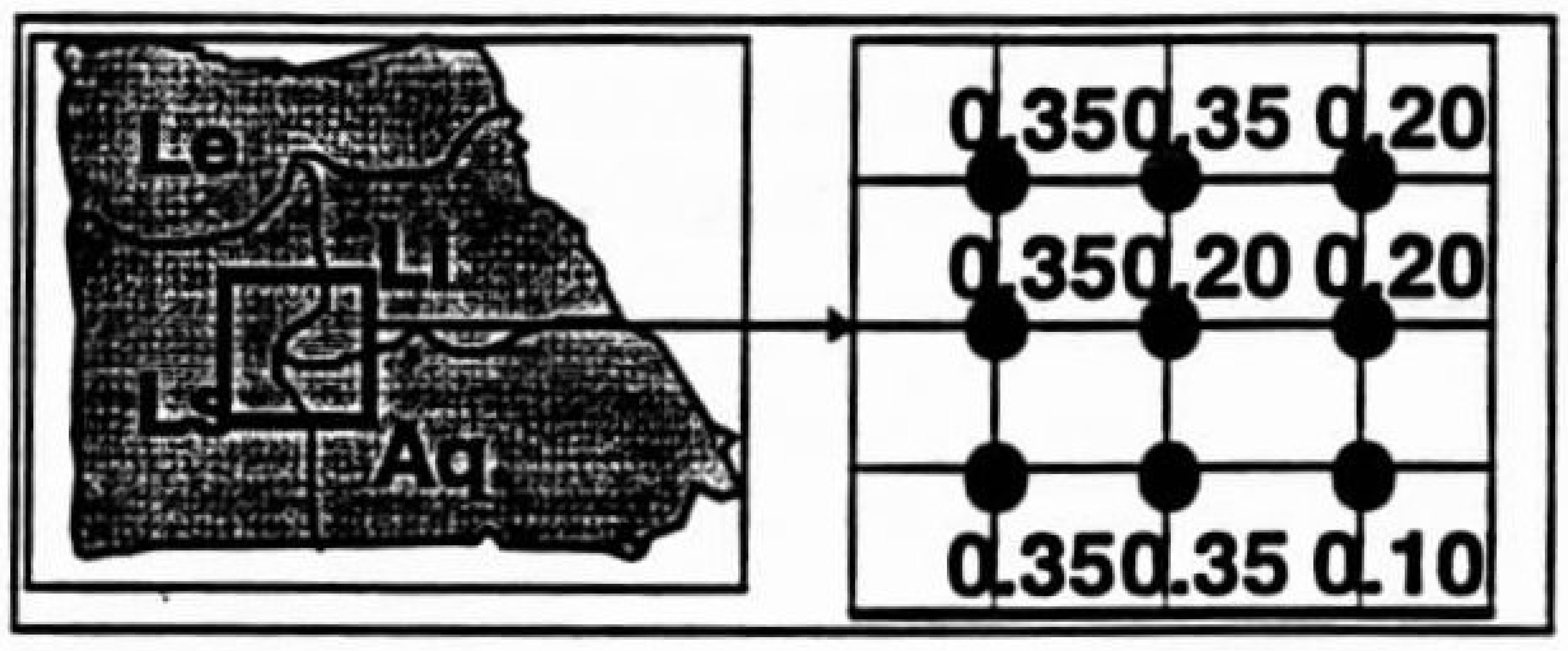
Slicing example (Slope Map)

```
{
Digital  slopeGrid ("Slope");
Thematic slopeMap ("ThematicSlope");
Table    gridToThemes (Slicing);

grd = Retrieve(Name="slope-30x30");
slc = New(Name= "classes_30x30", ResX=30, ResY=30, Scale=100000);

gridToThemes = New(CategoryIn="Slope",
            CategoryOut = "ThematicSlope",
            [0.0,3.0]     : "A-0a3",
            [3.0, 8.0]    : "B-3a8",
            [8.0, 12.0]   : "C-8a12",
            [12.0, 20.0]  : "D-12a20",
            [20.0, 45.0]  : "E-20a45",
            [45.0 ,90.0]  : "F>45",
            [90 , 900]    : "F>45" );

slc = Slice(grd,gridToThemes);
}
```



Weighting example

```
{
Thematic soilMap("Soils");
Digital erosionMap ("Erodibility");

Table soilErosionFactor (Weighting) ;

solo = Retrieve (Name = "soilMap1989");
erosionMap =New( Name="erosion", ResX=30, ResY=30, Scale=100000, Min=0,
                 Max=1);

soilErosionFactor = New (CategoryIn = "Soils",
      CategoryOut= "Erodibility",
      "LV1" : 0,
      "AR"  : 0.2,
      "LV2" : 0.6,
      "PV1" : 0.95);

erosionMap = Weight (solo, soilErosionFactor);

}
```



Land Use Reclassification Example

```
{
Thematic  use, recl ("LandUse");
Table rectab (Reclassification);

use = Retrieve (Name=LandUse1997");
recl = New (Name= "recla_alg", ResX=30, ResY=30, Scale = 100000);

rectab = New (CategoryIn = "LandUse", CategoryOut = "LandUse",
       "Decidous forest": "Forest",
       "Wet Forest": "Forest",
       "Rice": "Agriculture",
       "Wheat" : "Agriculture" );

recl = Reclassify (use, rectab);
}
```

Conversion of images expressed in digital counts to an image expressed in reflectance values, using the equation proposed by Markham & Baker (1987):

```
{
Image IV255, ima3, ima4, tm3, tm4,IV2("Image_TM");
Digital re3("Digital"), re4("Digital"), IVNAl ("Digital") ;

tm3 = Retrieve (Name="tm3_030895");
tm4 = Retrieve (Name="tm4_030895");

re3=New(Name="tm3_refl",ResX=30,ResY=30,Scale=100000,Min=0,Max=50);
re4=New(Name="tm4_refl",ResX=30,ResY=30,Scale=100000,Min=0,Max=50);

IVNAl=New(Name="IV_Refalg",ResX=30,ResY=30,Scale=100000,Min=0,Max=50);
IV2 = New (Name = "IVDN_RefAlg", ResX = 30, ResY = 30);
IV255 = New (Name = "IV255", ResX = 30, ResY = 30);

ima3 = New (Name = "ima3reflect", ResX = 30, ResY = 30);
ima4 = New (Name = "ima4reflect", ResX = 30, ResY = 30);

c1 = 1.0119;
c2 = 0.607735;

re3=Digital((PI*(C1^2)/155.7*C2)*((tm3/255)*(20.43-(-0.12))+(-0.12)));
re4=Digital((PI*(C1^2)/104.7*C2)*((tm4/255)*(20.62-(-0.15))+(-0.15)));

ima3= Image (re3 * 255);
ima4= Image (re4 * 255);

IVNAl = (re4-re3)/(re4+re3);
IV2= Image ((re4-re3)/(re4+re3));
IV255 = Image (255*((re4-re3)/(re4+re3)));
}
```

Calculation of vegetation index from a TM image

```
{
Image tm3, tm4, viimg("TM_Image");
Digital ndvi("Numerical");
Thematic veget("Vegetation");

tm3 = Retrieve (Name = "tm3_86");
tm4 = Retrieve (Name = "tm4_86");

viimg = New (Name = "Vegetation", ResX = 120, ResY = 120);
viimg = 40*((tm4-tm3)/(tm4+tm3))+64;

Table slice(Slicing);
slice = New(CategoryOut = "Vegetation",
        [0.0,0.2] : "Non_forest",
        [0.2,0.5] : "Transition",
        [0.5,1.0] : "Forest");
veget = New(Name="SoilCoverage",ResX=120,ResY=120,Scale=250000);
veget = Slice(Digital((tm4-tm3)/(tm4+tm3)), slice);
}
```

Class 9 – Chart Generation and Plotting

1. Chart Elements

Title

The Title describes the chart purpose and it should be in a prominent position.

Size

The chart Size depends on the goals of the chart and on the print device available.

There are standard paper size and format to represent drawings. A0 is the basic format from which all other formats are derived. The following table presents the more frequent used formats. It considers the external margin that is the paper cut indication.

Standard drawing format

Format	Height	Length
A0	841 mm	1189 mm
A1	594 mm	841 mm
A2	420 mm	594 mm
A3	297 mm	420 mm
A4	210 mm	297 mm

Scale

The chart scale should de defined considering which information will be contended in the chart. The right scale depends on the original data resolution, as well as the detail level desired.

The Scale should be placed in a prominent position in the chart.

The scale can be represented as a fractional number (Scale 1:300.000) or a graphic symbol.

The graphic scale is a line segment divided in a way that one can measure distances in the chart. Dimensions of objects presented in the chart can be easily visualized by this type of scale indication. The following example indicates which distance is related to 3 km in the chart.

The advantage in using the graphic scale is that it can be reduced or amplified with the chart drawing, conserving the scale relation.

Normally, scales are classified based on the represented subject. The following table shows a general scale classification based on the size and on the representation.

Based on Size	Base on Representation	Scale	Applications
Big Scale	Detail Scale	Up to 1:25.000	Cadastral Maps, Detailed surveys or topographical layer.
Medium Scale	Semi-detail Scale	From 1:25:0000 up to 1:250.000	Topographical Maps
Small Scale	Synthesis or Recognition Scale	From 1:250.000 up to smaller.	General Charts and Colorgraphics Charts.

Legend

Legend is a class relating non-spatial attributes to spatial attributes. Non-spatial attributes can be visually indicated by colors, symbols or shadows, in the way that it is defined on the legend.

Localization

As better as the object represent in the chart is adjusted to its space, the better is the chart precision. For this reason, the chart should have a coordinate system. A grid of geographical or latitude and longitude coordinates is normally used.

Balance e Lay-out

The chart elements should be positioned in a logical way to call the attention to the chart focus. This is the chart balance: nothing is so dark or light, nothing is so small or big, nothing is so short or long.

Lay-out is the process to achieve a suitable balance. It should be done how many Layouts the user thinks it is convenient.

Pattern contrast

Different pattern is used to represent different chart regions. Lines, points or line and points combination can compose patterns. It is not recommended to represent irregular areas with line patterns that are not very different in space between lines and direction one from another. A chart represented by point pattern is more stable and its contours are clearly distinct.

Color

Color is the strongest variable in a chart; it is easily perceived and intensively selective. It is the most delicate to work with and the most difficult to use. According to how much emphasis is desired to one data, the color is chosen. Some colors are more perceptive than others colors. The human eye is more sensitive to the red, followed by green, yellow, blue and purple.

The user should consult the more appropriate color to represent that type of the data in the chart. Examples: roads are represented in red, rivers and seas in blue, forest types in green, climatic charts represent tropical regions in red and arid regions in yellow.

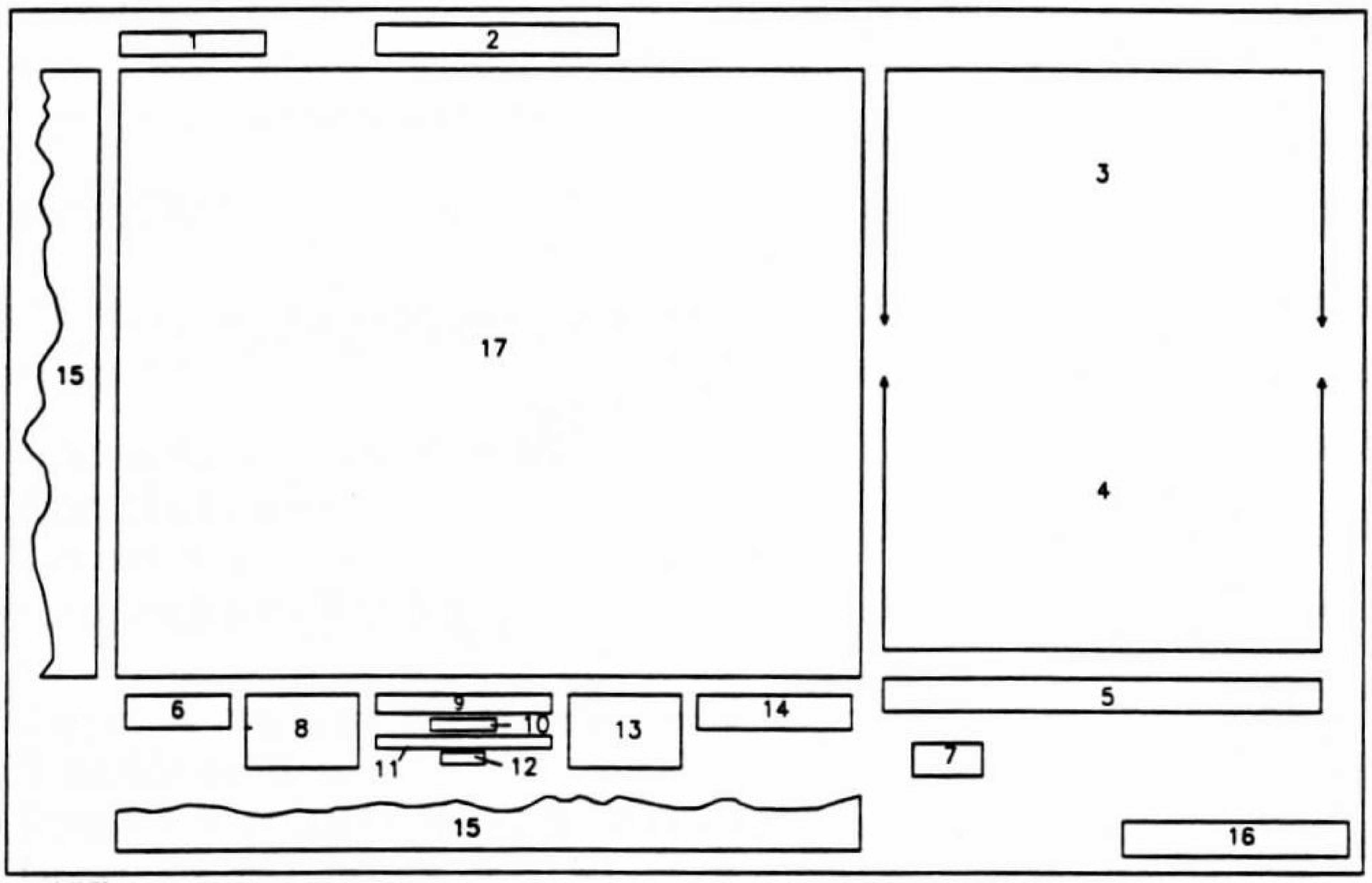
Clearness and Legibility

Clearness and legibility are chart qualities when an information can be easily found, differentiated from the others and memorized without any effort. Legibility can be obtained through the appropriate line, forms and color choice, and through their precise delineation. Lines should be clear, thin and uniform. Color, pattern and shadow should be easily distinct and correctly registered. The symbol shapes should not be confused.

One should differentiate mainly information from the basic map (background information). High graphic density should be avoided because it makes the chart complicate and difficult to read.

Chart presentation

Chart presentation is very important.



Where:

1 – Name and sheet code
Ex: São Paulo - SF-23-Y-C-VI-2
2 – Institution and executor
3 – Lithographic Units Description
4 – Geological Conventions
5 – Planialtimetric Conventions
6 – Base-map information
7 – Magnetic Declination
8 – Sheet localization
9 – Chart name
10 – Numeric scale
11 - Graphic Scale
12 – Execution Year
13 – Sheet articulation
14 – Project and executor team
15 – Areas to geological section localization
16 – Chart title and bibliographic reference (optional)

2. Chart Generation (SCARTA)

⇒ ***Starting Chart generation***

- *# Spring 3.2– Scarta( )* *#scarta (* **UNIX** *)*

SCARTA-3.2

- [File][Database] or press
  - **Data Base**
  - (Data Base | Curso)
  - (Active)

SCARTA-3.2

- [File] [Load Project...] or press
  - **Projects**
  - (Project | Brasilia)
  - (Active)

SCARTA-3.2

- [File] [Create Chart...] or press
  - **Create Chart**
  - {Name: Carta_uso}
  - (Create)

  Note: A new interface window will be shown where you can define the Paper and Data characteristics.

2.1 Chart Characteristics

⇒ ***Defining paper and data Characteristics***

SCARTA-3.2

- [Edit], [Characteristics...] or press
  - **Chart Generation**
  - (Size⇔ A4)
  - (Orientation ⇔ Vertical)

⇒ ***Defining chart useful area with 1 cm border***

Chart Generation

{Drawing Area (cm) – X1= 1.0 - Y1= 1.0}

{Drawing Area (cm) – X2= 29.40 – Y2= 20.79}

- (Restart)

⇒ ***Defining data area***

Chart Generation

- {Scale: 1/ 500000}
- (Position ⇔ Coordinates)
- {X1: 2.35}, {Y1: 2.35} - *PS: This is the initial point (lower left) of data on paper*
- (Coordinates⇔ Plane)
- *Click and move the lower left corner of data area*
- *Click and move the up right corner of data area*

⇒ ***Displaying Chart and data area***
SCARTA-3.2

- [Apply] [Draw] or press
- *Change chart Characteristics if it is nessesary*

2.2 Data presentation at Chart

Choose which data will be showed in the chart: select which Layers on the Control Panel you want to show. In this example the Layers will be: **Mapa_uso, Mapa_vias** and **Mapa_rios**, defined during the training.

SCARTA-3.2

- [Show] [Control Panel...] or press
  Control Panel
  - (Categories | Uso_Terra)
  - (Layer| Mapa_uso)
  - {Priority: 100} - (CR)
  - (Lines) - (Classes)
  - (Categories | Drenagem)
  - (Layer| Mapa_rios)
  - {Priority: 150} - (CR)
  - (Lines) - (Classes)
  * Observe the Priority of presentation: it can be changed.

- ***Displaying Chart***
SCARTA-3.2

- [Execute] [Draw] or press

2.3 Chart Elements

⇒ ***Editing text, symbols and legends using paper XYcoordinates***
SCARTA-3.2

- [Edit] [Elements...] or press
  Chart Elements Editor
  - (Paper (cm))
  - *"Double-click" on screen to recover X and Y of position to insert element and symbols. Ps. Elements inserted using Geographical Coordinates will not be recovered.*

⇒ ***Inserting Texts***
Chart Elements Editor
- [Insert] [Texts...]
  Inserting Text
  -{Text: Brasilia Soil Map }
  - (Insert)

⇒ ***Changing text Characteristics***
Chart Elements Editor
- *Select the TEXT you want to change on canvas using mouse*

- [Define][Characteristics...]
- *Select the TEXT you want to change on canvas using mouse*
    Chart Elements Characteristics
    - (TEXT ⇔ Unclick it) It must be disable if you want to change the color/thichness etc.
    - (TEXT ⇔ Color...)
    - *Select any color*
    - {Text - Height(mm): 6}
    - {Text - Thickness(mm): 0.20}
    - (Update)

⇒ ***Moving text to the position defined by coordinate***

Chart Elements Editor

Position Control

- X: 7.5 - Y: 23.5}
- (Set Position)

⇒ ***Inserting Symbols***

Chart Elements Editor

- [Insert] [Symbols...]
    Inserting Symbols
    - {Symbols: Aeroporto-276a} or other one
    - (Insert)
    - *Select symbol and move it to desired position*

⇒ ***Changing Symbols Characteristics***

Chart Elements Editor

- [Define][Characteristics...]
    Chart Elements Characteristics
    - *Select the symbol you want to modify on canvas using mouse*
    - {SYMBOL- Height(mm): 10}
    - {SYMBOL - Rotation(graus): 45}
    - (Update)
    - *Move the symbol to desired position*

⇒ ***Inserting legends for all thematic classes***

Chart Elements Editor

- [Insert] [Legends...]
    Insert Legend
    - {Initial Position(cm)- X: 21}, {Initial Position(cm)- Y: 17}
    - {Spacing : 3.5} *PS: Vertical Space among Legends*
    - (Categories | Uso_Terra)
    - (Classes | Agua)
    - [Insert] [Selected] **PS: on the top of this interface*
    - (Classes | Urbano1)
    - [Insert] [Selected]
    - (Classes | Cerrado)

⇒ ***Changing Characteristics of Legends***

Chart Elements Editor

- *Select the set of legends you want to modify on canvas, using mouse.*
- [Define][Characteristics...]

  Chart Elements Characteristics

  - {LEGEND- Height(mm): 5 - Width(mm): 10}
  - {LEGEND- Distance(mm): 15}
  - (LEGEND- Box Position: - Leftward)
  - {TEXT - Height(mm): 5}
  - (TEXT - Color...)
  - *Select color* **Preta**
  - (Update)
  - (Close)

⇒ ***Changing Text of Legends***

Insert Legend

- {Categories: Uso_terra}
- {Classes: Agua}
- {Legend Items: Agua }
- {Text: Water }, (CR)

Change* **Urbano *to* **Urban area**

⇒ ***Align legends text (left/right/central...)***

[Edit][Elements...]

Chart Elements Editor

- Select all legends on the canvas –draw a rectangle bounding legends.
- (Align)(Align Left)
- (Close)

2.4 Showing Graticule

⇒ ***Editing Graticule Coordinates***

SCARTA-3.2

- [Edit] [Graticule...]

  Graticule Definition

  - (Plane)
  - (Define Coordinates⇔ Plane)
  - {X: 181972}, {Y: 8241711} - *Ps: Initial graticule coordinates*
  - {|dx|: 5000}, {|dy|:5000}
  - (Define ...)

    Plane Graticule

    => ***Defining Divisions characteristicas***

    GRATICULE CHARACTERISTICS

    - {Subdivisions per Division: 1}
    - {Defining : Division}
    - (Color...) (OK)
    - *Select* **Black** *color to Division*
    - (Continue) (*type of principal lines)

    COORDINATES CHARACTERISTICS

    - (Coordinates Show)
    - {Height(mm): 4}

- {Distance(mm): 2}
- (Presentation ⇔ Reduced Coordinates}
- (Apply)

Graticule Definition

- (Apply)

=> ***Defining Divisions characteristics***

- (Define: ⇔ Subdivisions)
- (Color...)
- *Select* **sky** *color to subdivisions*
- (Continue)
- (Coordinates Show)
- {Height(mm): 3}
- {Distance(mm): 2}
- (Presentation ⇔ Reduced Coordinates)
- (Apply)

Graticule Definition

- (Apply)

⇒ ***Displaying Planes coordinates graticule***

Graticule Definition

- (Apply)

⇒ ***Editing geographical coordinates graticule:***

Graticule Definition

- (Geographic)
- (Define Coordinates⇔ Geographic)
- {Long: o 47 58 0}, {Lat: s 15 53 0} - *PS: Initial graticule coordinates*

-{|dx|: o 0 1 00}, {|dy|:s 0 1 00}

- (Define ...)

Geographic Grid

- {Subdivisions per Division: 4

⇒ ***Defining divisions:***

Geographic Grid

- (Define ⇔ Division)
- (Color...)
- *Select black color to Division*
- (Crossings)
- (Coordinates Show)
- {Height(mm): 4}
- {Distance(mm): 5}
- (Apply)

⇒ ***Defining Subdivisions:***

Geographic Grid

- (Define ⇔ Subdivision)
- (Color...)

- *Select* **sky** *color to subdivisions*
- (Crossigns)
- (Coordinates Show)
- {Height(mm): 3}
- {Distance(mm): 5}
- (Apply)

- (Apply)

⇒ ***Displaying geographical coordinates graticule***

Graticule Definition

- (Apply)

2.5 Editing Drawing Area

⇒ ***Editing drawing area***

SCARTA-3.2

- [Edit], [Drawing Area...]

Define Drawing Area

- *Define width and color of drawing area contour line.*
- *Define width and color of data area contour line.*
- (Apply)

2.6 Editing Lines

⇒ ***Editing Lines***

SCARTA-3.2

- [Edit] [Lines...]

=>*Defining line attributes and creating lines*

Lines Editor

- *Define line thickness and color.*
- (**Step (mm)** - 2)
- (**Create** - Yes), (**Orthogonal** - Yes) , (**Assistant** - Extremes)
- *Draw a rectangle bounding legends.*

- *Test creating non-orthogonal lines.*
- *Test creating lines using middle, near and normal assistance.*

⇒ ***Eliminating Lines***

Lines Editor

- (**Create** - No)
- *Select lines to be deleted*
- (**Delete**)

⇒ ***Moving Lines***

Lines Editor

- (**Create** - No)
- *Select the lines you want to move and drag them to desired position*

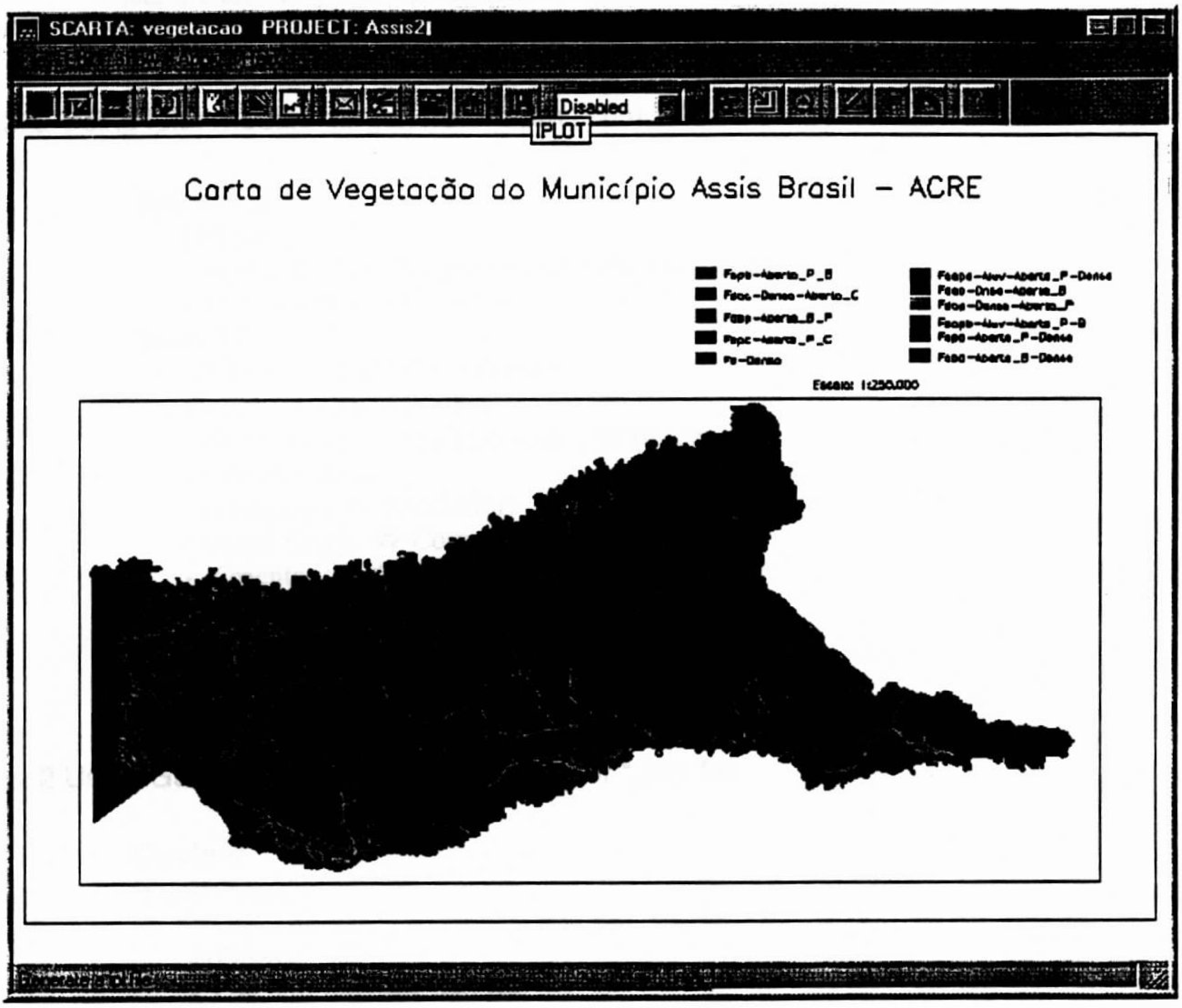
=> *Defining Divisions*

2.7 Saving Created Chart

⇒ ***Saving created Chart***
SCARTA-3.2
- [File] [Save Chart] or press

⇒ ***Saving created Template***
SCARTA-3.2
- [File] [Save Template]

Example:



3. Printing Chart

3. 1 Generating *.ipl file to be used in IPLOT module

SCARTA-3.2
- [File] [Generating IPLOT...]
SPRING: Save as
Select the directory and the file name (*.ipl)
(Save)

3. 2 Using Iplot module to generate *.prn (*.ps) file

Iplot-3.2
[File][Open...]
- Select the *.ipl file generated by **Scarta** module
Ps: You will not see the file
Iplot-3.2
[File][Generate File to Plotter]
Generate File to Plotter
- Select all options related with your printer
Defaults values:
- (Language ⇔ PostScript)
- (Axes Origin ⇔ Corner)
- (Orientation ⇔ Landscape)
- (Generate File)
SPRING : Save as
- Select the directory and give the file name (*.prn)
- (Save)

3. 2 Using GSView to see and print *.prn (or *.ps) file

Gsview
[File][Open]
- Select directory and where you generated the file (*.prn)
- (Open)
- You will see the Chart file
[File] [Print]