

## Tutorial 4: Fusão de Dados Imagem

### Introdução.

O objectivo principal deste tutorial consiste em estudar algumas das opções de fusão de dados imagem disponíveis na livreria de algoritmos do PCI Geomatica:

1. Transformação IHS
2. PanSharp

### Transformação IHS

Na transformação IHS, a Intensidade (*Intensity*) refere-se ao brilho total da cor, a Tonalidade (*Hue*) refere-se ao comprimento de onda dominante e a Saturação (*Saturation*) especifica a pureza da cor. No método de fusão pela técnica IHS, transformamos três bandas espectrais de baixa resolução espacial do espaço RGB de cores para o espaço IHS. Ao executarmos a transformação estamos a separar a informação acromática (Intensidade) da cromática (Tonalidade e Saturação) de uma imagem padrão RGB (Red = Vermelho, Green = Verde, Blue = Azul). A ideia básica do método consiste então em substituir a componente intensidade pela banda Pan e depois efectuar a operação inversa e regressar ao espaço RGB

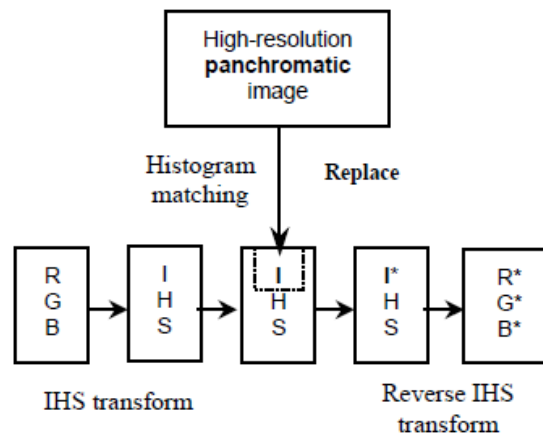


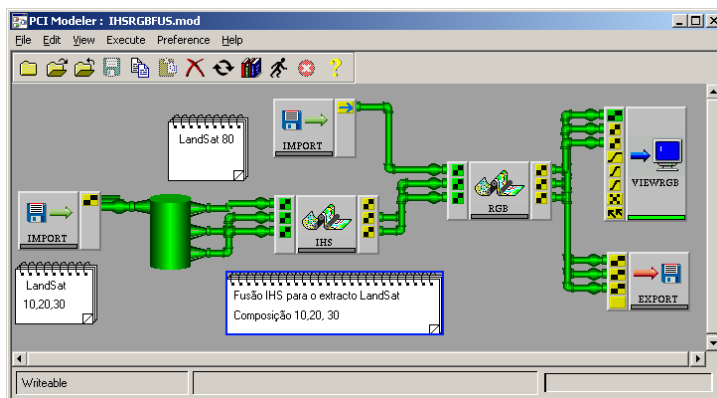
Figura 1: Fluxograma da transformação IHS (Adaptado de Zhang, 2002)

Os passos mais importantes desta transformação são:

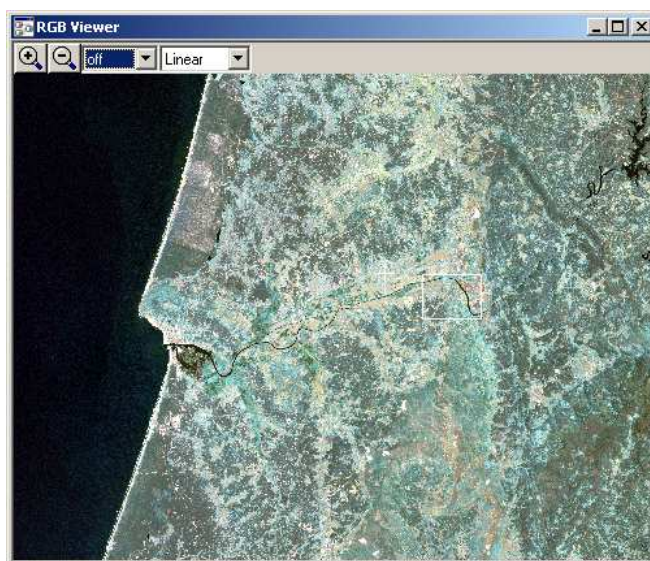
1. Transformar a cor da imagem MS do espaço RGB para o espaço IHS;
2. Substituir a componente da I (Intensidade) pela imagem pancromática (Pan) com uma resolução superior;
3. Efectuar uma transformação inversa com as componentes substituídas do espaço IHS de volta ao espaço original RGB, de forma a obter uma imagem fundida.

Vamos em primeiro lugar utilizar um extracto duma imagem do satélite LandSat. Temos 4 bandas (as bandas 10, 20, 30 e 40 com resolução de 30 m) que pretendemos integrar/fundir com a banda pancromática (banda 80 com resolução de 15m).

Utilizando o Modeler e os algoritmos IHS e RGB podemos reproduzir o fluxograma indicado na figura 1



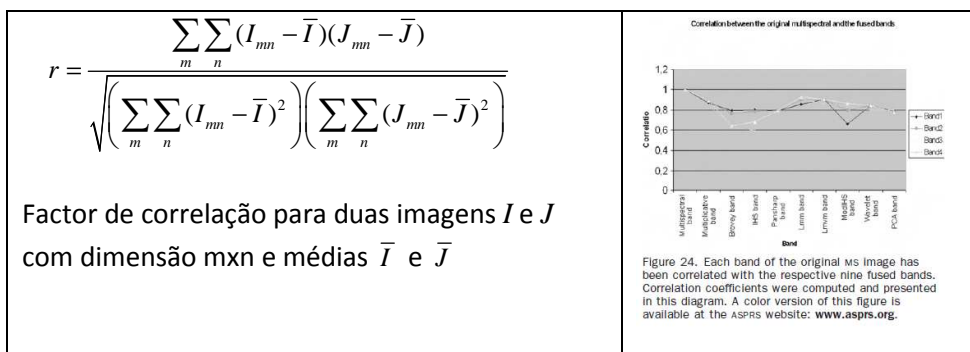
O resultado da execução do modelo para a composição 123 (ou 10,20,30) é dado na figura seguinte:



Note-se que poderíamos realizar esta fusão de um modo ainda mais simples utilizando apenas o algoritmo fuse.

### Avaliação da transformação

O objectivo é utilizar o coeficiente de correlação entre duas imagens para avaliarmos quantitativamente os resultados da fusão. Calculando o coeficiente de correlação para cada par de bandas antes e depois da fusão podemos construir um gráfico como o indicado na figura.



Para calcularmos o coeficiente de correlação entre duas bandas teremos de utilizar o seguinte script easi:

```
!===== D E T E C A O R E M O T A =====
!
```

Gil Goncalves

```

! Este script calcula a correlação entre duas bandas de uma geodatabase pix
!  $r = \frac{\sum_m \sum_n (I_{\{mn\}} - \bar{I})^2}{\sum_m \sum_n (I_{\{mn\}} - \bar{I})^2}$ 
!
! Exemplo: Suponhamos que dois ficheiros
! fich1="antes.pix" com as 4 bandas que pretendemos fundir: R,G,B,I
! fich2="depois.pix" com as 4 bandas fundidas : R,G,B,I
!
! este script escreve num ficheiro texto correlacao.txt a correlacao entre
! cada uma das bandas.
! Nota1: que é suposto que para a banda pancromatica o factor de correlação
! é 1 visto que esta banda não se altera no procedimento
! Nota2: é importante que as bandas estejam pela mesma ordem nos dois
! ficheiros
!=====

! Variáveis locais
local int fid, fid1, fid2
local int i, j
local string fich1, fich2, tmp1, tmp2
local double meanI, meanJ
local double r

!=====
!
!===== Variaveis a personalizar =====
! ficheiros pix que contem as 4 bandas antes e depois da fusão
fich1 = "LandSatExtrB10B40.pix"
fich2 = "FusaoIHSAula.pix"
!=====

!=====
! M E A N
!=====
! Função para o calculo da média dos valores dos pixeis numa imagem
define function MeanIm(fid,banda)
local integer mlines, npixels
local integer i, j
local double meanI, soma
npixels=DBPixels(fid)
mlines=DBLines(fid)
local float databuffer[npixels]
soma=0
for i=0 to mlines-1
    call DBReadLine(fid,banda,i,0,npixels,databuffer)
    for j=1 to npixels
        soma = soma + databuffer[j]
        !print j,"-", databuffer[j]
    endfor
endfor
meanI=soma/(mlines*npixels)
return(meanI)
enddefine
!=====
! C O R R E L A T I O N
!=====

```

```

! Função para o calculo da correlação entre duas bandas
define function Correlation(fidI, bandaI, fidJ, bandaJ)
local double meanI, meanJ
local integer mlines, npixels
local integer i, j
local double num,den1,den2,r
! Numero de pixeis da linha
npixels=DBPixels(fidI)
! Numero de linhas da imagem
mlines=DBLines(fidI)
local float databufferI[npixels]
local float databufferJ[npixels]
! Media da imagem I
meanI=MeanIm(fidI,bandaI)
! Media da imagem J
meanJ=MeanIm(fidJ,bandaJ)
num=0
den1=0
den2=0
for i=0 to mlines-1
  call DBReadLine(fidI,bandaI,i,0,npixels,databufferI)
  call DBReadLine(fidJ,bandaJ,i,0,npixels,databufferJ)
  for j=1 to npixels
    num = num + (databufferI[j]-meanI)*(databufferJ[j]-meanJ)
    den1= den1 + (databufferI[j]-meanI)*(databufferI[j]-meanI)
    den2= den2 + (databufferJ[j]-meanJ)*(databufferJ[j]-meanJ)
    !print j,"-", databuffer[j]
  endfor
endfor
r=num/sqrt(den1*den2)
return(r)
enddefine
!=====

!=====
! Gestão do erro
try
! 1. Abrir o ficheiro antes da fusão
tmp1 = NormalizePath(GetCWD() + "/" + fich1)
tmp2 = NormalizePath(GetCWD() + "/" + fich2)
fid1 = DBOpen(tmp1,"r+")
fid2 = DBOpen(tmp2,"r+")
fid = TEXTOpen( "correlacao.txt", "w" )

for i=1 to 2
  r=Correlation(fid1,i,fid2,i)
  ! Escrever o factor num ficheiro
  call TextWrite( fid, F$STRING(r) )
  ! Escrever no ecran
  print "r= ", r
endfor
! 2. Get the
! Close file
Call DBClose(fid1)
Call DBclose(fid2)

```

```
Call TEXTClose(fid)
```

```
OnError
```

```
    print GetLastErrorMessage()
```

```
EndOnError
```