

UNIVERSIDADE DO MINHO



## **Visão por Computador**

Sessões práticas com o sistema Khoros/Cantata (v3.2)

João Luís Ferreira Sobral

Departamento de Informática

Braga, Setembro 2003

## Sessão prática 1 - Familiarização com o sistema Khoros

### 1.1 Introdução ao sistema Khoros

O sistema Khoros é uma colecção de funções e ferramentas para processamento e visualização de imagens/dados. O sistema inclui múltiplas interfaces com o utilizador, geradores de código, várias funções de processamento e visualização dos dados. Todas as ferramentas do sistema, bem como o sistema em si, podem ser modificados garantindo a sua extensibilidade e portabilidade.

Da grande variedade de componentes do sistema são de destacar os seguintes:

- a biblioteca de processamento de dados, em especial as funções de processamento de imagem;
- a linguagem de programação visual abstracta, **cantata**, para a utilização do sistema Khoros através da interface gráfica;
- o utilitário **animate** que permite a visualização de sequências de imagens;
- os interpretadores e geradores de código que permitem a extensão do sistema com novas ferramentas ou com novas funções de processamento de dados (**preview**, **composer**, **ghostwriter** e **conductor**).

A biblioteca de processamento de dados contempla principalmente o processamento de imagem, embora também inclua funções para processamento de sinal. A biblioteca de processamento de imagem contém as funções mais comuns no nível mais baixo de visão por computador. De entre a grande variedade de funções destacam-se as funções geométricas (rotação, translação, redução, ...), aritméticas (soma, subtração, divisão, inversão, ...), lógicas (ou, e, ou exclusivo, ...), filtros espaciais 2D (convolução, mediana, ...), filtros de frequências 2D (passa-banda, passa-baixo, passa-alto, ...), histograma (equalização, histograma, ...) e segmentação (binarização dinâmica, determinação de eixos, fecho de contornos, ...). Existem ainda outros grupos de funções, tais como conversões de dados, transformadas e funções baseadas em máscaras. Praticamente todas estas funções processam uma imagem e originam uma imagem como resultado do processamento.

O formato de imagem (dados) usado pelo Khoros designa-se por VIFF (Visualization/Image File Format). O formato contém um cabeçalho seguido dos dados, podendo estes ser de vários tipos; os mais frequentes são: bits, bytes, inteiros e real. Também é possível guardar informação relativa à cor e informação dispersa (pares [x,y]).

## 1.2 Utilização do sistema através da linha de comandos

### 1. Iniciar as variáveis de ambiente

- a. execute o seguinte comando para disponibilizar o sistema Khoros

```
source <directoria do khoros>/khoros
```

### 2. Visualizar e manipular imagens

- a. experimente algumas das ferramentas deste sistema. A sintaxe utilizada é do tipo UNIX: por convenção, todos os argumentos do comando são precedidos por um sinal '-' seguido de uma etiqueta. Por exemplo, **-i** tipicamente indica o ficheiro de entrada e **-o** o ficheiro de saída. A variável `$KHOROS`, refere-se à directoria onde se encontram os ficheiros deste sistema. Visualize uma imagem no ecrã com o seguinte comando

```
putimage -i $KHOROS/sampled/data/images/ball.xv
```

- b. faça desaparecer a imagem com um duplo clique na imagem
- c. reduza o tamanho de uma imagem (um factor de 2) e grave-a com o nome `saida` com o comando

```
kshrink -i $KHOROS/sampled/data/images/ball.xv -o saida  
-wmag 0.5 -hmag 0.5
```

- d. visualize o ficheiro `saida` com o comando anterior
- e. veja os parâmetros possíveis para um comando com a etiqueta **-usage**. Experimente

```
kshrink -usage
```

- f. o `stdin` pode ser utilizado como ficheiro de entrada, especificando '-' como nome do ficheiro. É possível utilizar, de modo semelhante, o `stdout` como ficheiro de saída. Experimente o comando:

```
kshrink -i saida -o - -wmag 0.5 | putimage -i -
```

## 1.3 Utilização do sistema através do ambiente gráfico

Além da interface através da linha de comandos, os principais programas do sistema Khoros disponibilizam uma interface gráfica.

### 1. Visualizar uma sequência de imagens

- a. a visualização interactiva de sequências de imagens é possível através do programa **animate**. As sequências são especificadas através de imagens com várias bandas ou através de ficheiros descrevendo um conjunto de imagens. Esta ferramenta permite a visualização da sequência, imagem a imagem, para a frente ou para trás. Experimente os seguintes comandos (para iniciar a sequência utilize o botão >>):

```
animate -i $KHOROS/sampled/data/sequences/bushes.viff
```

```
animate -i $KHOROS/sampled/data/sequences/hog_heart.viff -cmap  
$KHOROS/sampled/data/sequences/solid.xv
```

## 2. Editar imagens

### a. para editar uma imagem experimente

```
editimage -i $KHOROS/sampled/data/images/feath.eye
```

### b. para desenhar um conjunto pré-definido de pontos experimente

```
xprism -il $KHOROS/sampled/data/signals/strokec.xv
```

```
xprism -il $KHOROS/sampled/data/images/ball.xv
```

```
putplot2 -i $KHOROS/dipcourse/data/2dfft.viff
```

```
putplot3 -i $KHOROS/dipcourse/data/2dfft.viff
```

## 1.4 Utilização do sistema através da interface visual

A invocação dos comandos do sistema Khoros é efectuada através de linhas de comando textuais. O sistema dispõe no entanto de uma linguagem de programação visual, **cantata**, o qual, através de uma janela X.Windows, converte as instruções icónicas do utilizador em linhas de comandos. A programação visual representa com uma caixa cada função seleccionada num menu, como num utilitário de desenho. O encadeamento de várias operações para obter um dado processamento é efectuado no **cantata** ligando as caixas que representam cada operação. Como algumas funções possuem várias entradas ou várias saídas é possível múltiplas ligações entre caixas. Podem ainda existir entradas ou saídas de uso facultativo. Através da linguagem visual é extremamente fácil construir protótipos de processamento.

Para carregar um exemplo execute:

```
cantata -wksp $KHOROS/sampled/examples/workspaces/vol_elel_time.wk
```

Podem executar este exemplo seleccionando o ícone de execução.

## Sessão prática 2 - Filtros espaciais (filtros no domínio (x,y))

### Nota:

As imagens abingdon.viff e emma.viff e as máscaras para convolução (laplace.viff e realce.viff) podem ser descarregadas da página da disciplina

As restantes imagens encontram-se na directoria \$KHOROS/sampledata/data/images/

Inicie o sistema Khoros (ver sessão 1.2 - 1.a) e o programa Cantata (cantata&)

### 2.1 Operações básicas com o cantata

#### 1. Visualizar uma imagem

- a. crie um Glyph que representa a imagem a visualizar (menu `Input/Output -> Data Files -> User defined`) e posicione-o no ecrã com um clique do rato
- b. crie agora um Glyph para visualizar essa imagem (menu `<Vizualization, Non-Interactive Display, Display Image>`)
- c. ligue a saída do Glyph da imagem à entrada do Glyph de visualização com um clique na seta amarela de cada Glyph
- d. no menu do Glyph da imagem escolha uma imagem e pressione o ícone RUN do espaço de trabalho para executar a sequência criada

#### 2. Gravar e ler o espaço de trabalho

- a. é possível gravar e posteriormente recuperar o espaço de trabalho utilizando os ícones correspondentes ou o menu Files

#### 3. Outras operações com os Glyph

- a. seleccione a bomba do Glyph para o destruir e seleccione o interruptor quando pretender executar só um Glyph
- b. pode copiar, colar e duplicar um ou vários Glyphs. Para seleccionar vários Glyphs marque com o rato uma região no espaço de trabalho contendo os Glyphs desejados ou faça um clique em cada Glyphs, mantendo a tecla de SHIFT premida

### 2.2 Propriedades Estatísticas e Alteração do histograma

#### 1. Visualizar as propriedades estatísticas de uma imagem

- a. utilize o Glyph `<Data Manip, Analysis & Information, Statistics>` para calcular várias propriedades estatísticas de uma imagem. Esta informação é textual e deve ser visualizada com `<Input/Output, Information, File Viewer>`

2. Visualizar o histograma de uma imagem
  - a. seleccione os glyphs <Data Manip, Histogram Operator, Histogram> e <Visualization, Plot Display, 2D Plot> para ver um histograma. Este segundo Glyph desenha o histograma.
  - b. visualize o histograma da imagem fusca.viff
  - c. verifique a resultado da aplicação da equalização do histograma <Data Manip, Histogram Operators, Equalize>
  - d. ligue os dois histogramas ao Glyph <Visualization, Plot Display, Interactive 2D/3D Plot> para comparar os histogramas das duas imagens
  - e. compare o resultado da aplicação da equalização do histograma à imagem (alínea c) com o resultado de uma transformação logarítmica da imagem (<Arithmetic, NonLinear Functions, Logarithm>)

### 2.3 Operador de gradiente e operador laplaciano - Filtros passa-alto

1. Visualizar o resultado do operador de gradiente
  - a. aplique uma convolução <Arithmetic, Linear Transforms, Linear Operator>, com o filtro de gradiente Sobel X <Input/Output, Data Files, 2D Filter Kernels>, à imagem emma.viff (ligue o Glyph da imagem à primeira entrada e o Glyph do filtro à segunda entrada)
  - b. uma vez que as linhas e colunas mais exteriores de uma imagem após a convolução deixam de conter informação relevante, obtém-se melhores resultados se estas forem desprezadas. Seccione (<Data Manip, Size and Region Operators, Extract>, indicando como coordenadas (x,y) (2,2) e como dimensão da imagem extraída (252,252).
  - c. Repita as alíneas a) e b) para o filtro Sobel Y (basta apenas copiar e colar os Glyphs e alterar o filtro)
  - d. calcule o resultado da aplicação simultânea dos dois filtros, calculando a normal das duas imagens (i.é., componentes) com <Arithmetic, Trigonometry, Hypotenuse>
2. Visualizar o resultado do operador laplaciano
  - a. verifique o resultado da convolução da mesma imagem com o operador laplaciano (ligue a imagem laplace.viff à segunda entrada do Glyph)
  - b. repita as operações anteriores com a imagem fusca.viff (para tal basta alterar a imagem no Glyph correspondente)

Nota: no final desta documento encontra as matrizes correspondentes a cada máscara.

## 2.4 Realce de contornos

Existem duas formas de realçar os contornos de uma imagem: através de um filtro especializado ou adicionando a imagem original à obtida pelo operador laplaciano.

1. Realçar contornos através de um filtro
  - a. efectue a convolução da imagem emma.viff com o kernel realce.viff e visualize o resultado
2. Realçar contornos através da adição da imagem à obtida com o operador laplaciano
  - a. adicione `<Arithmetic, Two Operators Arithmetic, Add>` a imagem resultante do operador laplaciano com a imagem original e visualize o resultado
  - b. compare o resultado obtido com o realce de contornos com a imagem original e com o resultado da aplicação de uma equalização do histograma à imagem original

## 2.5 Filtros de suavização – Filtros passa-baixo

1. Suavizar imagens com filtros lineares
  - a. aplique um filtro de média às imagens emma.viff e abingdon.viff executando a convolução com os kernels avg3x3, avg5x5 e avg7x7 `<Input/Output, Data Files, 2D Filter Kernels>`
  - b. verifique as diferenças resultantes da variação do tamanho dos filtros
2. Suavizar imagens com filtros não-lineares
  - a. experimente utilizar filtros de mediana `<Image Proc, Non-Linear Filters, Median>` com vários tamanhos (3x3, 7x7, 11x11) para suavizar a imagem
  - b. compare com os resultados da alínea 1 anteriores

## 2.6 Combinação de filtros passa-alto com passa-baixo

1. Aplicar um filtro passa-baixo seguido de um passa-alto
  - a. aplique uma detecção de contornos (filtro de gradiente `<Image Proc, Spatial Filters, Gradient>`) à imagem abingdon.viff
  - b. execute a mesma operação depois de aplicar um filtro de mediana, 11x11, à imagem
  - c. explique as diferenças obtidas

## Filtros para convolução

$$\text{Sobel X} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{Sobel Y} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{laplace} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{realce} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{avg3x3} = \begin{bmatrix} ,11 & ,11 & ,11 \\ ,11 & ,11 & ,11 \\ ,11 & ,11 & ,11 \end{bmatrix}$$

$$\text{avg7x7} = \begin{bmatrix} ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \\ ,02 & ,02 & ,02 & ,02 & ,02 & ,02 & ,02 \end{bmatrix}$$

## Sessão prática 3 - Filtros no domínio das frequências

### Nota:

As imagens bola.viff, bola1.viff, bola2.viff, bola3.viff e noisy.viff podem ser descarregadas da página

As restantes imagens encontram-se na directoria \$KHOROS/sampled/data/images/

### 3.1 Introdução à Transformada de Fourier (FFT)

1. Visualizar a imagem a transformar
  - a. Crie um Glyph para seleccionar a imagem bola.viff
  - b. Visualize a imagem
2. Visualizar a magnitude do espectro da imagem
  - a. Aplique a transformação de Fourier <Arithmetic, Linear Transforms, FFT> à imagem anterior
  - b. Calcule a magnitude da transformada <Arithmetic, Complex Operators, Magnitude> e visualize o resultado (espectro da imagem)
  - c. Compare o resultado anterior com o obtido calculando o logaritmo da magnitude+1 e logaritmo da magnitude (altere os parâmetros do glyph). Qual a diferença entre os vários espectros?
  - d. Calcule a fase da transformada com a função <Arithmetic, Complex Operators, Phase> e visualize o resultado

### 3.2 Propriedades da transformada de Fourier

1. Visualizar a imagem a transformar e associar objectos da imagem a partes do espectro
  - a. Visualize a imagem bola.viff
  - b. Visualize o espectro da imagem e tente identifique a parte do espectro correspondente a cada objecto da imagem inicial. Que propriedade utilizou para essa identificação? Se não consegue responder visualize o espectro da imagens bola1.viff, bola2.viff e bola3.viff.
2. Visualizar as alterações do espectro introduzidas por rotações e translações da imagem
  - a. Aplique uma rotação à imagem <Image Proc, Geometric Manip, Rotate> de 45° centrada em 128,128 (para tal seleccione *W center* e *H center*). Verifique a alteração introduzida no espectro.
  - b. Aplique uma translação à imagem de 10 pixels <Data Manip, Reorganize Data, Translate>. Verifique se existem alterações na imagem transformada (veja também as alterações na fase).

### 3.3 Filtros passa-baixo e passa-alto

1. Aplicar um filtro passa-baixo no domínio das frequências
  - a. Visualize a imagem emma.viff
  - b. Aplique a transformada de Fourier à imagem
  - c. Multiplique a imagem `<Arithmetic, Two Operand Arithmetic, Multiply>` por um filtro passa-baixo ideal `<Image Proc, Frequency Filter, Low-Pass>`. Indique um raio de corte de 0.2 nos parâmetros do filtro (este número representa a distância do centro, em percentagem, da frequência de corte).
  - d. Visualize o espectro da imagem resultante
  - e. Visualize o efeito da filtragem na imagem original, aplicando a transformada inversa à imagem filtrada (.../FFT com direcção *Inverse*). Visualize apenas a parte real da transformada inversa. Quais as alterações relativamente à imagem original? Quais os filtros no domínio do espaço (x,y) que produzem um efeitos semelhantes?
  - f. (\*) Compare a imagem obtida efectuando a mesma operação, agora utilizando um raio de corte de 0.3 e utilizando o filtro de Butterworth de ordem 2 e raio de corte de 0.2.
2. Aplicar um filtro passa-alto no domínio das frequências
  - a. Repita a alínea 1, agora aplicando um filtro passa-alto `<Image Proc, Frequency Filter, High-Pass>` à mesma imagem. Quais os filtros espaciais que produzem efeito semelhante?

### 3.4 Aplicações da FFT

1. Remover ruído através de um filtro rejeita-banda
  - a. Visualize a imagem noisy.viff e o respectivo espectro. Este imagem possui ruído numa determinada frequência (pico visível no espectro)
  - b. Aplique um filtro rejeita-banda `<Image Proc, Frequency Filter, Stop-Band>` para remover o ruído da imagem (passos 1c e 1e de 3.3).
  - c. No grupo de filtros de frequências, seleccione um que permita obter uma imagem apenas com o ruído.
2. Remover ruído através de um filtro passa-baixo
  - a. Visualize a imagem abingdon.viff. Retire o ruído desta imagem através da aplicação de um filtro no domínio das frequências.

### 3.5 Teorema da convolução ( $f(x)*g(x) \Leftrightarrow F(u) \times G(u)$ )

1. Visualizar o espectro de filtros para convolução
  - a. Transforme um dos filtros de convolução (p.ex. avg3x3) numa imagem de 256x256 com a função <Data Manip, Size & Region Operators, Pad>
  - b. Aplique a FFT à imagem resultante e visualize o espectro do resultado
  - c. Através do teorema da convolução, interprete o espectro do filtro
  - d. Repita as alíneas anteriores para os filtros avg7x7, Sobel X, Sobel Y, laplace e realce
2. Ilustrar o teorema da convolução
  - a. Calcule a transformada de Fourier da imagem emma.viff e a transformada de Fourier do filtro de convolução avg3x3 (como em 1.a)
  - b. Multiplique as duas transformadas anteriores (i.é., calculando  $F(u) \times G(u)$  )
  - c. Calcule transformada inversa do resultado anterior (i.é.,  $\mathcal{F}^{-1}(F(u) \times G(u))$  ).
  - d. Compare o resultado anterior com a imagem obtida através da aplicação do mesmo filtro no domínio espacial através da convolução (i.é.,  $f(x)*g(x)$ ). Obterá resultados melhores se rejeitar as duas primeiras linhas e as duas primeiras colunas (<Data Manip, Size and Region Operators, Extract>) das imagens obtidas, visualizando assim, apenas a parte das imagens efectivamente processada.

## Sessão prática 4 – Restauração de imagem

### Nota:

As imagens `lenina.viff`, `gauss-kernel-21x1.viff`, `lenina-blur-float.viff`, `map-roi.viff`, `map-roi-4x.viff`, `avg4x4.viff`, `interpolacao.viff` encontram-se na página da disciplina

### 4.1 Restauração de imagem

1. Restaurar uma imagem degradada
  - a. A imagem `lenina.viff` foi degradada, através de uma convolução, com o kernel `gauss-kernel-21x1`. Visualize estas duas imagens (nota: o kernel de convolução é uma imagem 1D). Visualize também a imagem degradada `lenina-blur-float.viff`.
  - b. calcule a transformada (FFT) da imagem degradada e a transformada do filtro (não se esqueça de transformar o filtro numa imagem de 256x256 com a função `<Data Manip, Size & Region Operators, Pad>`. Nota: utilize *Unity Scaling* em FFT).
  - c. divida a transformada da imagem degradada pela transformada do filtro `<Arithmetic, Two Operand Arithmetic, Divide>`, aplique a transformada inversa ao resultado e visualize a imagem restaurada.
2. Restaurar uma imagem degradada com ruído
  - a. o exemplo anterior corresponde à situação ideal. No entanto, na prática a imagem a restaurar irá conter ruído aleatório. Para simular este ruído converta a imagem `lenina-blur-float.viff` para o tipo `unsigned byte` `<Data Manip, Data Conversion, Convert Data Type>` e Subtraia as duas imagens para calcular o ruído introduzido `<Arithmetic, Two Operand Arithmetic, Subtract>` e visualize o resultado.
  - b. aplique o processo da alínea 1 à imagem com ruído. Tente encontrar uma justificação para o resultado obtido, para tal compare os espectros do kernel, da imagem degradada e da imagem restaurada.
3. Restauração de imagem com filtragem inversa e com o filtro de Wiener
  - a. um dos problemas da filtragem inversa surge na presença de valores pequenos na transformada do filtro. Estes valores pequenos, ao serem utilizados como divisores na filtragem inversa, podem amplificar fortemente o ruído existente na imagem. Daí surge a explicação para os resultados anteriores. Uma solução para este problema baseia-se em desprezar estes valores ao efectuar a divisão, tornando o resultado para esses pontos igual a zero. Esta operação é efectuada pelo filtro `<Data Manip, Frequency Filter, InvFilter>`. Utilize este filtro como alternativa à divisão da alínea 2 e visualize o resultado.
  - b. experimente a mesma operação, utilizando o filtro de Wiener `<Data Manip, Frequency Filter, Wiener>`, com o valor de 0.05. Compare o resultado dos dois filtros para o mesmo parâmetro e visualize as alterações introduzidas pela modificação do valor do parâmetro.

## 4.2 Interpolação de imagem

1. Interpolação por replicação de pixels
  - a. visualize a imagem map-roi.viff e o respectivo espectro. A imagem map-roi-4x.viff resulta da expansão desta imagem 4 vezes, sem interpolação. Visualize também esta imagem e o respectivo espectro.
  - b. efectue uma convolução da imagem expandida com o kernel avg4x4; visualize a imagem resultante e o respectivo espectro.
2. Interpolação por replicação pesada de pixels
  - a. repita a alínea 1, agora utilizando o kernel interpolacao.viff
3. Interpolação no domínio das frequências
  - a. aplique a transformada de Fourier à imagem map-roi-4x.viff
  - b. aplique um filtro passa baixo (Butterworth de ordem 6 e raio 0.10) à transformada (Multiplique a transformada <Arithmetic, Two Operand Arithmetic, Multiply> por um filtro passa-baixo <Image Proc, Frequency Filter, Low-Pass>)
  - c. calcule a transformada inversa e visualize a imagem resultante. Qual das três alternativas lhe parece produzir melhores resultados?

## Sessão prática 5 – Processamento de imagens a cores

### 5.1 Pseudo-cor

#### Atribuir cores a uma imagem em tons de cinzento

1. Visualizar a imagem a colorir
  - a. Crie um Glyph para seleccionar a imagem retina.viff
  - b. Visualize a imagem
2. Criar uma imagem para referência com todos os tons de cinzento
  - a. Crie uma imagem de 128x20 pixels em tons de cinzento utilizando o Glyph *Khoros 1, Generate Data, Piecewise Linear*  
Altere os seguintes parâmetros: Number Rows=128, Number of Cols=10, Sampling Freq=1, Maximum Value=255, Y Period=128 e Y Rise Time=128
  - b. Converta a imagem para unsigned byte com o Glyph *Data Manip, Data Conversion, Convert Type*
  - c. Use o operador *Data Manip, Size & Region Operators, Pad* para acrescentar uma borda branca com um pixel  
Altere os seguintes parâmetros: Padded Width=10+2, Padded Height=128+2, Width Offset=1, Height Offset=1 e Real Pad Value=255
3. Juntar as duas imagens
  - a. Utilize o operador *Data Manip, Size & Region Operator, Inset* para juntar as duas imagens
  - b. Visualize a imagem resultante
4. Colorir a imagem a visualizar
  - a. Adicione um mapa de cores à imagem anterior com o Glyph *Visualization, Map Display & Manip, Autocolor*
  - b. Visualize a imagem
  - c. Experimente utilizar outros mapas de cores (designadamente o mapa SA-Pseudo)

## 5.2 Cor completa

### 5.2.1. Compor uma imagem RGB partindo dos três componentes

1. Criar 3 imagens sintéticas de um círculo, cada uma com o centro numa posição diferente.
  - a. Crie um Glyph para gerar um círculo *Khoros 1, Generate Data, Circle Image*  
Altere os parâmetros: Rows=128, Columns=128, Diameter=80, X of center=64 e Y of center=70
  - b. Efectue uma translação do círculo: *Data Manip, Reorganize Data, Translate*. Altere os parâmetros: Width Offset=0, Height Offset=-30
  - c. Duplique duas vezes os Glyphs anteriores e altere os parâmetros de translação de cada um deles para (-25,15) e (25,15)
  - d. Visualize cada um dos círculos
2. Combinar as três imagens numa imagem RGB.
  - a. Junte as três imagens com o Glyph *Data Manip, Size & Region Operators, Append*. Ligue cada uma das saídas de Translate a uma entrada de Append.  
Altere o parâmetro Dimension de Append para “Elements”
  - b. Transforme a imagem resultante numa imagem RGB com o Glyph *Data Manip, Object Attributes, Set Attribute*. Altere o parâmetro Colorspace para RGB
  - c. Visualiza a imagem resultante

### 5.2.2. Decompor uma imagem RGB nos três componentes

1. Visualizar a imagem Mandrill.viff
  - a. Selecciona a imagem a cores com o Glyph *Input/Output, Data Files, RGB Images*  
Altere o parâmetro do Glyph para seleccionar a imagem correspondente.
  - b. Visualize a imagem
2. Ler o cabeçalho da imagem para verificar que contém três elementos numa imagem RGB
  - a. Utilize o Glyph *Input/Output, Information, Data Object Info* para gerar um texto que descreve as características da imagem
  - b. Utilize o Glyph *Input/Output, Information, File Viewer* para visualizar a informação gerada pelo Glyph anterior
3. Extrair o plano R (Red)
  - a. Utilize o Glyph *DataManip, Size & Region Operators, Extract*  
Altere os parâmetros: Em Specifying Region Size desactivar Width e Height e seleccionar Elements, deixando o valor em 1; em Specifying Region Origin seleccionar Elements deixando o valor em 0 (i.é., Element Origin=0)
  - b. Visualize a imagem resultante
4. Repetir 3 para os planos G e B
  - a. Extraia e visualize o plano G (Elemento 1 – Element Origin=1) e o plano B (Elemento 2 – Element Origin=2).

## Sessão prática 6 – Compressão de imagem

### 6.1 Compressão de imagem (com perda) através da Transformada de Fourier

#### 6.1.1. Comprimir uma imagem através da transformada de Fourier

1. Transformar a imagem para o domínio de Fourier e remover coeficientes.
  - a. Converta a imagem em `emma.viff` para o domínio de Fourier *Arithmetic, Linear Transforms, FFT*
  - b. Extraia uma zona central de 128x128 pixels com a função *Data Manip, Size & Region Operators, Extract*  
Altere os seguintes parâmetros: dimensão da imagem a extrair = 128x128, região de origem = (64,64)
  - c. Qual o grau de compressão utilizado?
2. Visualizar a imagem após a remoção dos coeficientes.
  - a. Converta a imagem extraída para 256x256 com a função *Data Manip, Size & Region Operators, Pad*  
Altere os seguintes parâmetros: *offset*=(64,64)
  - b. Aplique a transformada inversa
  - c. Visualize a imagem resultante

#### 6.1.2. Comparar imagens com vários graus de compressão

1. Repetir o processo anterior para blocos de 96x96 e 64x64.
  - a. Duplique os glyphs anteriores e altere os parâmetros de *Extract* para 96x96 e origem em (80,80) e o *offset* de *Pad* para (80,80)
  - b. Duplique novamente os glyphs anteriores e altere os parâmetros de *Extract* para 96x96 e origem em (80,80) e o *offset* de *Pad* para (96,96)
  - c. Visualize cada uma das imagens e compare a sua qualidade
  - d. Qual o grau de compressão utilizado em cada imagem?

#### 6.1.3. Comparar metodologias de escolha de coeficientes a eliminar

1. Elaborar um processo que permita remover os coeficientes baseado no seu valor e comparar o resultado com os resultados anteriores.
  - a. Utilize uma binarização do espectro para obter um filtro passa-baixo que permita remover as componentes da transformada cuja magnitude é inferior a 0.02
  - b. A quantidade de coeficientes removida pode ser obtida contando (i.é., histograma) o número de elementos a zero no filtro obtido anteriormente

## Sessão prática 7 - Operações básicas de segmentação

### 7.1 Segmentação baseada nos contornos

A divisão de uma imagem em regiões pode ser efectuada com base nos contornos existentes na imagem. Para que essa operação seja efectuada com sucesso é necessário que os contornos formem regiões fechadas. Em geral, quando se determinam os contornos de uma imagem surgem vários segmentos (correspondentes a contornos) sem ligação entre si, sendo necessário aumentar os vários segmentos para que estes se liguem uns aos outros, formando regiões fechadas. Uma técnica para efectuar esta operação baseia-se no gradiente da imagem.

1. Determinar os contornos de uma imagem através do gradiente
  - a. Visualize a imagem emma.viff
  - b. Aplique o operador de gradiente (*Image Proc, Spatial Filters, Gradient Operator*) à imagem. Altere o parâmetro que indica o tipo de gradiente para Prewitt
  - c. Aplique uma binarização (*Data Manip, Threshold and Clip Operators, Thres Above*) ao resultado anterior. Altere o nível de binarização (Cutoff Value) para 60
  - d. Visualize a imagem resultante
2. Determinar os contornos através de um filtro especializado
  - a. Aplique o filtro GEF (*Khoros 1, Spatial Filters*) novamente à imagem emma
  - b. Visualize a imagem resultante e compare com a imagem obtida na alínea anterior
3. Fechar os contornos
  - a. Utilize o operador *Khoros 1, Segmentation, Edge Close* para fechar os contornos resultantes do filtro GEF. À segunda entrada deste filtro deve ligar o resultado do operador de gradiente (sem binarização), uma vez que esta informação é a base para fechar os contornos
  - b. Compare a imagem obtida com a da alínea 2
4. Detectar os contornos depois de remover ruído da imagem
  - a. Repita a alínea 3, mas agora depois de aplicar um filtro de mediana *Image Proc, Nonlinear Filters (7x7)* à imagem emma
  - b. Compare o resultado com o obtido em 3

## 7.2 Segmentação baseada no histograma

1. Segmentar uma imagem com base no seu histograma
  - a. Aplique um filtro de mediana (7x7) à imagem *abingdon.viff* e visualize o seu histograma (*Data Manip, Histogram Operators e Visualization, Plot Display*)
  - b. Com base no histograma da imagem seleccione um valor para efectuar a binarização da imagem por forma a separar o corpo da cruz do fundo da imagem.. Verifique que existem dois picos no histograma, correspondendo um deles ao corpo da cruz e o outro ao fundo da imagem.
  - c. Visualize o resultado da aplicação da binarização da imagem
  - d. Pode escolher o ponto óptimo de binarização se visualizar o histograma em modo texto com os operadores *Print Data e File Viewer (Input/Output, Information)*

## 7.3 Operações morfológicas

1. Erosão e dilatação de imagens
  - a. Visualize o resultado da aplicação de operações de dilatação e de erosão (*MMACH, Basic*) à imagem binarizada na alínea anterior.  
As operações morfológicas geralmente necessitam de um kernel morfológico. Utilize o Glyph *MMACH, Tools, Disk Str.El.* para gerar o kernel necessário para cada operação e ligue-o à 2ª entrada dos glyphs
  - b. Compare o resultado das duas operações anteriores com a imagem original
2. *Opening e Closing*
  - a. Repita o exercício anterior, agora com as operações de *Opening e closing* (ambas estão em *MMACH, First*)
  - b. Experimente agora aplicar uma operação de *closing* seguida de *opening*
3. Determinação de contornos
  - a. Os contornos de uma imagem podem ser determinados com base no operador morfológico de erosão. Esta técnica consiste em subtrair a imagem erudida à imagem original. Aplique esta técnica a uma das imagens produzidas da alínea anterior. Compare esta técnica com a técnica tradicional, baseada em filtros passa-alto.

## Sessão prática 8 – Descrição de regiões

### Nota:

As imagens `textura1.viff`, `textura2.viff`, `textura3.viff`, `tx1.viff`, `tx1.viff`, `tx2.viff`, `tx3.viff`, `tx4.viff`, `drawing.viff` e `blobs.viff` podem ser descarregadas da página da disciplina

## 8.1 Texturas

### 8.1.1 Descrição de texturas através do modelo estatístico

1. Segmentar uma imagem com texturas através das suas propriedades estatísticas
  - a. Visualize a imagem `textura1.viff`. Consegue segmentar as regiões desta imagem através da determinação de contornos ou da binarização com base no histograma?
  - b. Aplique o operador que calcula a média em janelas de 11x11 (*Khoros 1, Feature Extraction, Spatial Analysis*) à imagem. Altere os parâmetros correspondentes neste filtro (i.é., dimensão da janela). Verifique que já pode aplicar uma binarização à imagens para separar as texturas existentes
  - c. Obtenha uma imagem apenas com a superfície de uma das texturas, aplicando uma binarização à imagem obtida. Indique o valor mínimo e máximo da média da textura escolhida como ponto inferior e superior da binarização.
  - d. Obtenha uma imagem apenas com uma das texturas, aplicando a operação de AND (*Arithmetic, Logical Operations, And*) entre a imagem original e a imagem com a superfície da textura
  - e. Repita todo o processo anterior, agora para a imagem `textura2.viff`. Acha que pode utilizar a mesma propriedade estatística para esta imagem? Experimente utilizar outras propriedades estatísticas
  - f. Qual o modelo que acha mais adequado para descrever as texturas da imagem `textura3.viff`

### 8.1.2 Descrição de texturas através do modelo espectral

1. Isolar uma textura através do processamento do seu espectro
  - a. Visualize a imagem `tx1.viff` e o respectivo espectro. Verifique que existem dois picos de frequências altas
  - b. Aplique um filtro rejeita banda (*Image Proc*) para remover os picos do espectro e visualize o resultado
  - c. Desenvolva um método para obter apenas a descrição da textura e verifique a semelhança entre a textura gerada a partir da descrição e a textura original
  - d. Repita a sequência anterior para a imagem `tx2.viff`

2. Comparar espectros de texturas
  - a. Visualize a imagem tx2.viff e o respectivo espectro. Considerando a posição dos picos no espectro, qual o tipo de partição (angular ou radial) do espaço de frequências mais adequada para descrever a textura
  - b. Compare os espectros das texturas tx1.viff e tx3.viff. O que pode concluir quanto à relação entre o tamanho do grão da textura e a posição dos picos no espectro?
  - c. Acha viável a descrição da textura existente na imagem tx4.viff através do espaço de frequências? Porquê?

## 8.2 Etiquetagem de regiões

O processo de etiquetagem de regiões consiste em atribuir um tom único a cada região, por forma a permitir posteriormente efectuar um processamento isolado de cada região.

1. Atribuir um tom a cada região de uma imagem
  - a. Visualize a imagem drawing.viff. Utilize o operador *Data Manip, Size & Region operators, Expand* para aumentar a dimensão da imagem, facilitando a sua visualização
  - b. Aplique a operação de etiquetagem à imagem (*MMACH, Basic, Labeling (mmach)*) e visualize o resultado (utilize também a expansão da imagem para facilitar a sua visualização)
  - c. Visualize novamente a imagem atribuindo cores a cada região através do operador *Visualization, Map Display & Manip, Autocolor*. Altere o tipo de mapa para *Random*
  - d. Duplique os glyphs de etiquetagem e de visualização anteriores e altere o parâmetro da fase de etiquetagem para *4-neighborhood*. Quais as diferenças entre esta imagem e a imagem obtida na alínea c)?
2. Determinar o número de regiões presentes na imagem
  - a. Utilize um operador que lhe permita obter informação estatística sobre a imagem para identificar o número de regiões presentes nas duas imagens resultantes do processo de etiquetagem
3. Classificar as regiões de uma imagem em função da sua área
  - a. Visualize a imagem blobs.viff e efectue a etiquetagem das regiões presentes na imagem
  - b. Calcule e visualize o histograma da imagem etiquetada. Altere o parâmetro *Number of bins* do histograma para o número de regiões da imagem (ignorando o fundo da imagem, i.é., *Minimum=1*)
  - c. Calcule o histograma do histograma para agrupar as regiões pela sua área (i.é., número de pixels). Altere os parâmetros do histograma: *bin width=400, Number of Bins=4*
  - d. Utilize os operadores *Print Data* e *File Viewer* (em *Input/Output, Information*) para ver a informação produzida. Interprete essa informação
  - e. Experimente parâmetros diferentes para a alínea c)

## Sessão prática 9 - Detecção de formas

### Nota:

As imagens scan1.viff, scan1.tpl, marilyns2.viff, marilyn2.tpl, marilyns.viff, marilyn.tpl, pieces.viff e proto.viff podem ser descarregadas da página da disciplina

### 9.1 *Template matching* (através da correlação)

1. Detectar um padrão pré-definido numa imagem
  - a. Visualize a imagem scan1.viff. Pretende-se detectar os zeros presentes na imagem.
  - b. Aplique uma correlação (*Arithmetic, Linear Transforms, Linear Operator*) entre a imagem scan1.viff e o *template* scan1.tpl. Os máximos da imagem resultante correspondem aos pontos onde a semelhança com o *template* é maior
  - c. Aplique uma binarização à imagem resultante da correlação para obter uma imagem só com os pontos máximos da imagem (i.é., regiões)
2. Detectar e contar o número ocorrências de um padrão pré-definido numa imagem
  - a. Visualize a imagem marilyns2.viff. Pretende-se desenvolver um processo que conte o número de marilyns presentes nesta imagem
  - b. Aplique uma correlação entre a imagem anterior e a imagem marilyn2.tpl. Normalize os valores resultantes com o operador *Data Manip, Data Conversion*
  - c. Aplique uma binarização à imagem resultante de forma a identificar as regiões onde existem máximos. Quantas figuras consegue reconhecer por este processo?
  - d. É possível efectuar automaticamente a contagem das regiões existentes na imagem resultante da correlação. Para tal existe o operador *Khoros1, Classification, Labeling* que atribui uma etiqueta a cada região identificada. Neste operador uma região é considerada um conjunto de pixels contíguo, podendo recorrer a operadores morfológicos para que seja identificado o número correcto de regiões. Deve também indicar ao operador para não desprezar as regiões com um reduzido número de pixels (alterando o parâmetro *Mimimun Region Size* para 0). Aplique este operador à imagem obtida na alínea c) e veja a imagem resultante. Quantas regiões foram identificadas?
3. Detectar o número ocorrências de um padrão através da correlação normalizada
  - a. O resultado anterior pode ser melhorado se inicialmente for utilizada a correlação normalizada (ver fórmula), onde  $F(x,y)$  representa a imagem inicial,  $t$  o *template*, o a operação de correlação e  $i$  representa uma máscara com a dimensão do *template*, em que todos os elementos são 1. Quantas regiões são identificadas com esta alteração?
  - b. Acha viável utilizar a correlação para as imagens marilyns.viff e marilyn.tpl. Como poderiam ser utilizados os resultados obtidos anteriormente para diminuir o tempo perdido a determinar a posição exacta das marilyns na imagem
4. Experimente uma técnica, no domínio das frequências, para reduzir o tempo de processamento da correlação.

## 9.2 Classificação de regiões

1. Etiquetar e atribuir cores às regiões da imagem
  - a. Visualize a imagem *pieces.viff*, efectue a etiquetagem das regiões da imagem com o operador *MMACH, Basic, Labeling (mmach)* e visualize a imagem resultante.
  - b. Visualize novamente a imagem atribuindo cores a cada região através do operador *Visualization, Map Display & Manip, Autocolor*.
2. Remover ruído da imagem
  - a. Insira a operação de *Closing (MMACH, First)* antes de efectuar a etiquetagem para remover ruído da imagem. Utilize *Disk Str.El. (MMACH, Tools)* para gerar o *kernel* necessário para a operação morfológica e ligue-o à 2ª entrada do glyph de *closing*
  - b. Visualize a imagem resultante e compare-a com a obtida em 1.a)
3. Calcular a propriedade estatística “Primeiro Momento Invariante” de cada região
  - a. Calcule o “Primeiro Momento Invariante” de cada região, aplicando o operador *Khoros1, Feature Extraction, Shape Analysis* à imagem etiquetada. Nos parâmetros do glyph seleccione o tipo de análise desejada (*Invariant Moments*).
  - b. Visualize o resultado da análise em ASCII, ligando a quarta saída do glyph ao operador *Input/Output, Information, File Viewer*
4. Extrair uma característica
  - a. A saída *VIFF Statistics File* (3ª saída do glyph) produz um ficheiro binário, contendo todas as estatísticas da imagem. Este ficheiro contém 61 elementos para cada objecto, um para cada característica. O elemento 42 (M1) corresponde ao primeiro momento
  - b. Para extrair os dados estatísticos relativos ao primeiro momento utilize o operador *Data Manip, Size & Region Operators, Extract*. Altere os parâmetros do glyph para: em *Specify Region Size* desligar *Width* e *Height*, ligar *Elements*; em *Specify Region Size* desligar *Width* e *Height*, ligar *Elements* e colocar o valor 42
  - c. Visualize o resultado com o operador *Visualization, Plot Display, Display 2D Plot*. Após a visualização, altere o tipo de gráfico para *Discrete* no botão *Options*. Pode ver os mesmos resultados em modo texto com os operadores *Input/Output, Information, Print Data* e *Input/Output, Information, File Viewer*
5. Classificar cada região de acordo com uma característica
  - a. Visualize o ficheiro *proto.viff* que contém as descrições utilizadas como base de comparação. Utilize os operadores *Print Data* e *File Viewer*
  - b. Classifique os objectos da imagem com o operador *Khoros1, Classification, Minimum Distance*. Ligue a primeira entrada ao resultado obtido em 4.b e a segunda entrada ao ficheiro protótipo
  - c. Visualize as regiões classificadas com os operadores *Input/Output, Information, Print Data* e *Input/Output, Information, File Viewer*