

# Apresentação de um Banco de Desenvolvimento e Ensaio para Objectos Deformáveis

João Manuel R. S. Tavares<sup>1</sup>, J. Barbosa<sup>2</sup>, A. Jorge Padilha<sup>3</sup>

<sup>1</sup>Universidade do Porto, Faculdade de Engenharia, Departamento de Engenharia Mecânica e Gestão Industrial, Instituto de Engenharia Biomédica, Laboratório Sinal e Imagem, Rua Dr. Roberto Frias, s/n 4200-465 PORTO – PORTUGAL

[tavares@fe.up.pt](mailto:tavares@fe.up.pt), <http://www.fe.up.pt/~tavares>

<sup>2,3</sup>Universidade do Porto, Faculdade de Engenharia, Departamento de Engenharia Electrotécnica e de Computadores, Instituto de Engenharia Biomédica, Laboratório Sinal e Imagem

[jbarbosa@fe.up.pt](mailto:jbarbosa@fe.up.pt), <http://www.fe.up.pt/~jbarbosa>

[padilha@fe.up.pt](mailto:padilha@fe.up.pt), <http://www.fe.up.pt/~padilha>

**Resumo** - Neste artigo será apresentado um sistema, de análise e processamento de imagem e de computação gráfica, que tem vindo a ser desenvolvido no Laboratório Sinal e Imagem do INEB – Instituto de Engenharia Biomédica desde 1995. A opção pelo desenvolvimento de uma nova aplicação, em vez da adaptação de uma aplicação já existente, deveu-se à constatação, por parte dos seus autores, da falta de plataformas que fossem adequadas não só para serem utilizadas como pacotes fechados, mas também como aplicações abertas que facilmente permitissem a incorporação de novas funções. O desejo do sistema incluir funções comuns na área da análise e processamento de imagem, em conjunto com algumas ferramentas habitualmente existentes no domínio da computação gráfica, e a necessária possibilidade de controlar comodamente as condições de ensaio e de análise dos resultados obtidos por novos algoritmos, foram também argumentos importantes que condicionaram a decisão do desenvolvimento de um novo sistema.

**Palavras-chave:** Computação Gráfica, Objectos Deformáveis, Visão por Computador.

**Abstract** - A system for analysis and image processing and for computer graphics, which has been developed in the Laboratório Sinal e Imagem of INEB – Instituto de Engenharia Biomédica since 1995, will be presented. The option for the development of a new application, instead of the adaptation of an already existent, was due to the lack of systems that were suitable not only for be used as closed packages but also as open applications that easily allowed the incorporation of new users functions. The desire that the system include common functions in the area of analysis and image processing, together with some tools habitually existent in the computer graphics domain, and the necessary possibility of comfortably controlling the rehearsal conditions and analysis of the obtained results by new users algorithms, were also important arguments that conditioned the decision for the developing of a new system.

**Key-words:** Computer graphics, Computer Vision, Deformable Objects.

## 1 – Introdução

Desde 1995, tem vindo a ser desenvolvido no Laboratório Sinal e Imagem do Instituto de Engenharia Biomédica, um banco de desenvolvimento e ensaio com ferramentas variadas que permite o estudo detalhado de diversos métodos de modelização de objectos deformáveis [1].

O carácter fechado da maior parte das aplicações existentes para processamento e análise de imagem, bem como o propósito de incorporar as metodologias desenvolvidas no nosso grupo num sistema de acesso e uso fáceis, conduziu a que se tenha resolvido desenvolver um novo sistema que, além de permitir a implementação e ensaio dos métodos desenvolvidos no âmbito dos objectos

deformáveis, fosse, por si só, uma aplicação genérica de processamento e análise de imagem, e um sistema aberto para outros investigadores desenvolverem, ensaiarem e incorporarem os seus algoritmos.

A linguagem seleccionada para o desenvolvimento da aplicação foi C/C++ [2] em ambiente *Microsoft Visual C++* [3]. A opção por esta linguagem deveu-se essencialmente à sua larga disseminação, robustez, modularidade, eficiência, facilidade de manutenção, à existência de alguns algoritmos já implementados nesta linguagem e que poderiam ser facilmente reutilizados, à existência de algumas bibliotecas de domínio público, implementadas nesta linguagem, que poderiam ser incorporadas, etc. A escolha do ambiente de programação foi essencialmente devida à estreita relação entre

este e os sistemas operativos alvo da aplicação: os sistemas *Microsoft Windows*.

Uma vantagem do sistema desenvolvido está relacionada com a capacidade de controlo das operações incorporadas, por meio da especificação detalhada de parâmetros, e visualização adequada dos resultados obtidos, o que permite o ensaio pormenorizado dos métodos utilizados. Outra vantagem que o sistema apresenta é a integração modular e transparente de funções orientadas para o processamento e análise de imagem, para o tratamento de sequências de imagens (de movimento e/ou de deformação), e de ferramentas de computação (e visualização) gráfica.

Nas secções seguintes serão identificadas as bibliotecas de domínio público incorporadas no banco de desenvolvimento e ensaio, apresentadas as entidades suportadas, exposta a sua interface, explicada a integração de uma nova função, serão discriminadas algumas das funções disponíveis de âmbito geral e também as funções mais específicas aos objectos deformáveis.

## 2 – Bibliotecas de domínio público integradas

No sentido de maximizar a reutilização de código, optou-se por incorporar no sistema algumas bibliotecas de domínio público existentes e que disponibilizam funções de grande utilidade, quer para sistemas genéricos quer mais específicos aos objectos deformáveis.

Para cálculo matricial incorporou-se a biblioteca *Newmat* escrita em C++, actualmente na sua versão 10, [4]; desta forma, as seguintes operações matriciais tornaram-se disponíveis: multiplicação, soma, diferença, concatenação, inversão, transposição, cálculo de determinantes, decomposição de *Cholesky*, triangulação *QR*, decomposição *SVD*, determinação dos valores e vectores próprios de uma matriz simétrica pelo método de *Jacobi* ou de *Householder*, ordenação e transformada rápida de *Fourier*.

Quando se iniciou o desenvolvimento do sistema optou-se pela utilização do formato *BMP* como formato nativo. Rapidamente se constatou que seria desejável incluir suporte para outros formatos [5]; destes destacavam-se o *JPEG*, o *GIF* e o *TIFF*. Dado o elevado número de classes de domínio público disponíveis optou-se por suportar este tipo de formatos através da incorporação de algumas destas bibliotecas. Devido essencialmente à facilidade de integração na aplicação e à sua portabilidade para o ambiente de programação utilizado, optámos pelas bibliotecas em C++ *Cimage*, na sua versão 1.2 [6] e *Jpeglib*, na sua versão 6 [7].

No decurso do trabalho no âmbito dos objectos deformáveis foi reconhecida a necessidade de utilização de algumas estruturas e ferramentas normalmente utilizadas no domínio da computação gráfica; destas destacam-se: classes

definidoras de entidades poligonais 2D e 3D; funções para escrita, leitura e representação dessas entidades; triangulação de um conjunto de pontos não estruturados; simplificação e suavização de malhas poligonais; extracção de contornos de isonível; filtragem, segmentação e amostragem de uma entidade poligonal; realização de operações lógicas e determinação das fases intermédias (*morphing*) entre duas estruturas; determinação de algumas características de um objecto poligonal; etc. Dada a sua grande utilização e robustez, capacidade, eficiência, contínua actualização, elevado suporte e manutenção, facilidade de integração no ambiente de programação *Microsoft Visual C++* e o facto de ser de domínio público, a escolha recaiu sobre a biblioteca *VTK – The Visualization Toolkit* [8-10]. Esta biblioteca gráfica, em C++, foi incorporada no sistema, estando assim as suas entidades e algumas das suas funções disponíveis no banco de desenvolvimento e ensaio.

## 3 – Entidades suportadas

Actualmente várias entidades são suportadas no banco de desenvolvimento e ensaio. Estas entidades podem ser agrupadas em dois grandes grupos: o do tipo *Bitmap* e o do tipo vectorial. No grupo *Bitmap* são consideradas as imagens constituídas por um conjunto de *pixels*; no grupo vectorial estão incluídas as estruturas definidas para pontos, para linhas, para contornos, para superfícies e para as entidades definidas na biblioteca *VTK*.

No caso dos pontos, estes não têm uma estrutura específica, sendo cada elemento definido unicamente pelas suas coordenadas; as linhas são definidas pelas coordenadas dos seus pontos extremos; os contornos podem ser de dois tipos: contornos abertos ou contornos fechados, sendo definidos pelas coordenadas dos pontos que os constituem e pelo seu tipo; as superfícies são definidas por um conjunto de pontos distribuídos ao longo de linhas e colunas; as entidades definidas na biblioteca *VTK* e actualmente utilizadas são: pontos estruturados ou não, entidades poligonais e grelhas de pontos estruturadas ou não.

Na implementação do sistema são considerados dois tipos de documentos: o documento imagem, que pode ser constituído por uma imagem *Bitmap* e/ou por uma imagem vectorial com pontos, rectas, contornos e superfícies; e o documento *VTK*, que pode ser constituído por entidades definidas na biblioteca *VTK*.

Na Fig. 1 estão representados, de forma gráfica, os dois tipos de documentos implementados, as entidades que os podem constituir e as operações de cópia, escrita e de leitura suportadas.

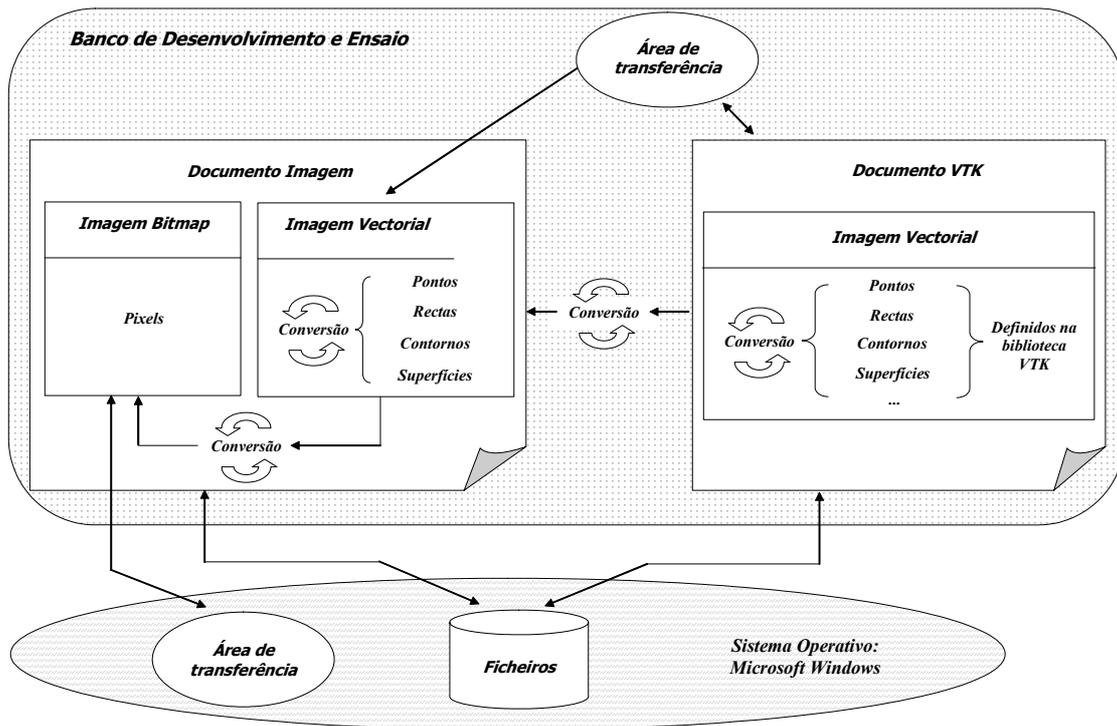


Fig. 1. Entidades suportadas no banco de desenvolvimento e ensaio.

#### 4 – Interface

A interface do banco de desenvolvimento e ensaio segue a filosofia geral das aplicações desenvolvidas para os sistemas operativos *Microsoft Windows*, alterando-se dinamicamente conforme o tipo de documento activo (Fig. 2 e Fig. 3).

Como a classe responsável pela visualização dos documentos VTK [8] deriva, por herança, da correspondente classe do VTK os botões do rato podem ser utilizados para controlar

a visualização dos objectos. Assim, no modo de interacção com a cena do tipo câmara é possível rodar e deslocar a câmara e alterar o nível de *zoom*. Já no modo de interacção com a cena do tipo actor, através dos mesmos botões, é possível aplicar ao objecto uma rotação, um deslocamento e um escalamento. Do mesmo modo algumas teclas assumem determinadas funções para este tipo de documentos, permitindo alterar o tipo de interacção do utilizador com a cena e a representação dos objectos na mesma.

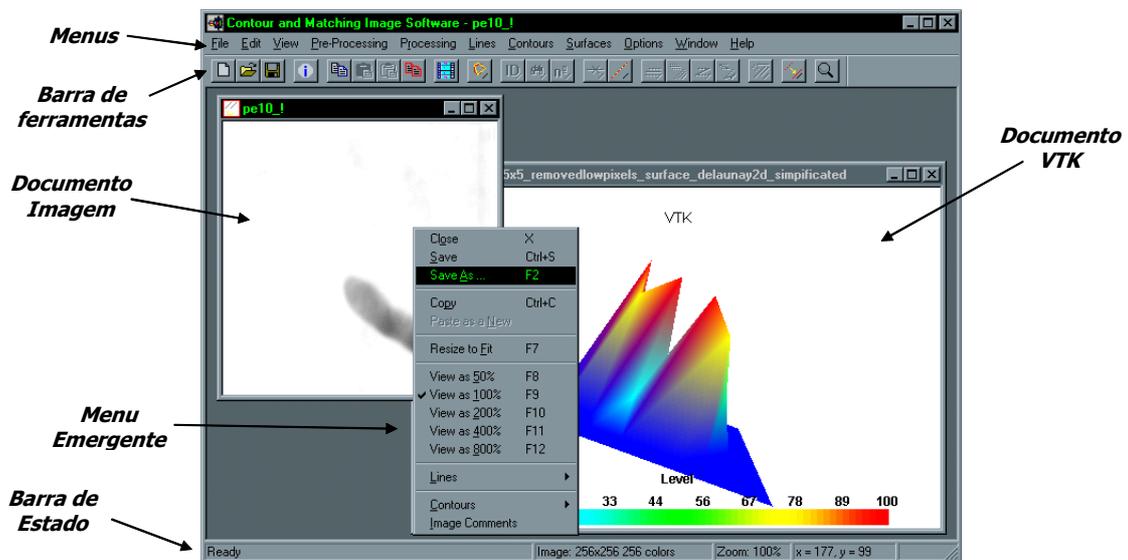


Fig. 2. A interface do banco de desenvolvimento e ensaio para documentos imagem.

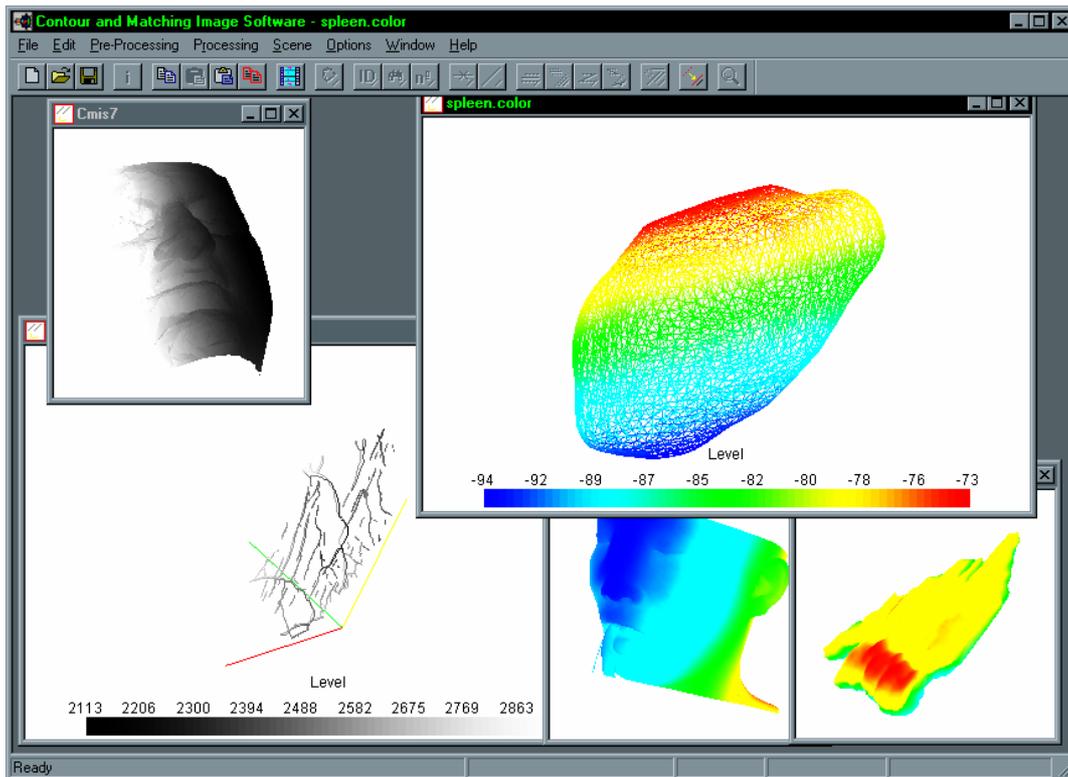


Fig. 3. Interface do sistema quando está activo um documento VTK.

## 5 – Integração de novas funções

Na Fig. 4 estão representadas as classes principais do banco de desenvolvimento e ensaio; estas classes são responsáveis pelo controlo da aplicação, pela gestão da sua janela principal, pelos documentos e pelas janelas de visualização.

Com a filosofia utilizada sempre que um utilizador pretenda acrescentar-lhe uma nova

função, já não necessita de se preocupar com a interface com os dados de entrada e de saída a considerar, nem com o modo como é realizada a gestão e manipulação dos mesmos. Assim, basicamente, o programador apenas necessita de acrescentar os ficheiros com o código fonte ao projecto e proceder à ligação da nova função no sistema.

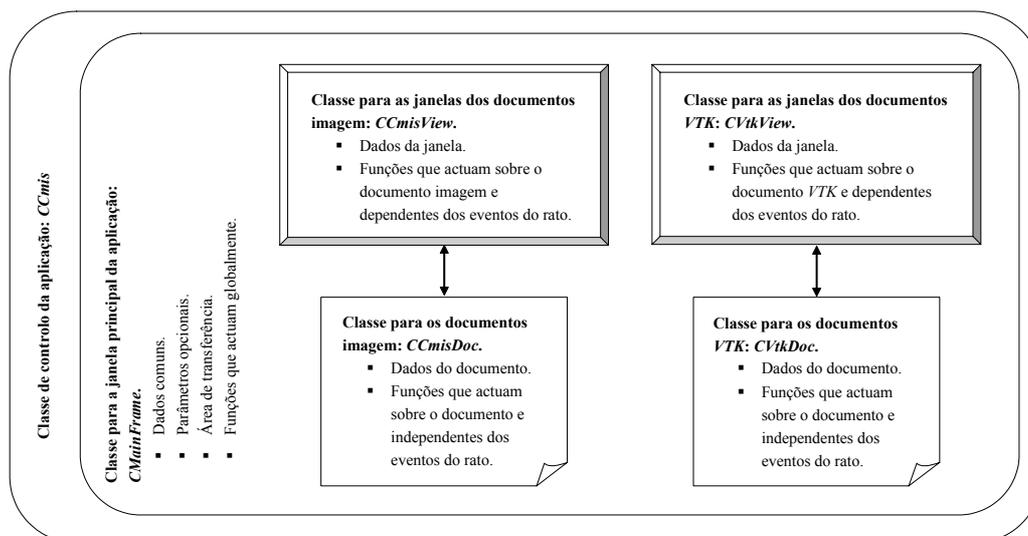


Fig. 4. Classes principais do banco de desenvolvimento e ensaio.

## 6 – Selecção de funções disponíveis

Nesta secção serão brevemente descritas algumas das funções actualmente disponíveis no banco de desenvolvimento e ensaio. Em primeiro lugar serão descritas as funções de índole mais geral e seguidamente, as funções mais específicas aos objectos deformáveis.

### 6.1 – Utilitárias

#### 6.1.1 – Documentos imagem

As funções básicas disponíveis para este tipo de documento estão relacionadas com entrada e saída de dados: criar, abrir, gravar e fechar documentos; realizar a aquisição de imagens; alterar o título; realizar a previsão e a impressão.

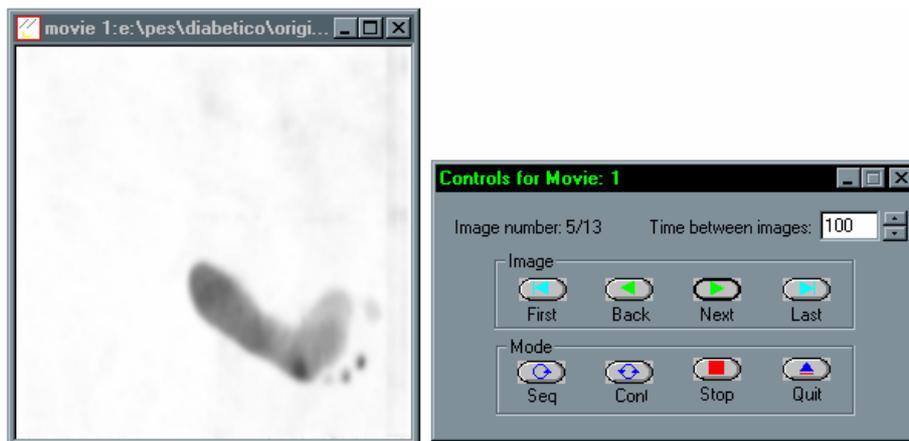


Fig. 5. Visualização de uma série de documentos imagem previamente seleccionados.

#### 6.1.2 – Documentos VTK

As funções básicas associadas aos documentos VTK são também relacionadas com a entrada e saída de dados: criar, abrir, fechar, gravar, um documento; importar um objecto; acrescentar um objecto VTK definido num determinado ficheiro no documento activo; criar um novo documento com dois objectos seleccionados; exportar um actor num formato diferente do VTK; definir o título; realizar a previsão e a impressão.

Para a edição deste tipo de documentos estão disponíveis as seguintes funções: copiar o conteúdo de uma janela de visualização; copiar e inserir um objecto copiado; criar um novo documento com um objecto copiado; duplicar um documento; alterar algumas propriedades de um objecto como a origem, a posição, a orientação, a escala e a opacidade; retirar um objecto da cena; determinar o número de objectos presentes; visualizar algumas características e a constituição de um objecto (Fig. 6).

Para edição estão essencialmente disponibilizadas as funções associadas à cópia das entidades suportadas. Assim, é possível: copiar uma imagem ou apenas uma zona desta; repor a imagem copiada no documento actual ou criar um novo documento com a mesma; duplicar um documento; copiar, e posteriormente inserir num outro documento, entidades vectoriais.

As funções relacionadas com a visualização dos documentos imagem disponíveis no sistema são basicamente as seguintes: alterar o factor de zoom; visualizar um conjunto de documentos, seleccionados pelo utilizador, numa mesma janela (Fig. 5); visualizar uma descrição da imagem activa, podendo o utilizador editar texto que pode ficar guardado juntamente com a imagem e ser posteriormente recuperado quando a mesma for aberta no sistema.

## 6.2 – Processamento de imagem

Como se pretende que o banco de desenvolvimento e ensaio seja por si só um sistema base para processamento de imagem, foram integradas funções consideradas como úteis para um tal sistema. Seguidamente serão destacadas algumas dessas funções.

No grupo das operações geométricas, foram integradas funções para: a rotação (sendo possível optar por um dos seguintes métodos de interpolação: "vizinho mais próximo", bi-linear ou cúbico), o espelho e o verso, e a alteração das dimensões.

Para alterar o brilho dos pixels que constituem uma imagem foram incorporadas as seguintes funções: aumentar e diminuir o brilho e o contraste; atribuir um determinado valor aos pixels que apresentem valor inferior a um dado limiar, ficando os restantes com o valor original; conversão de imagens a cores em imagens de cinzentos equivalentes.

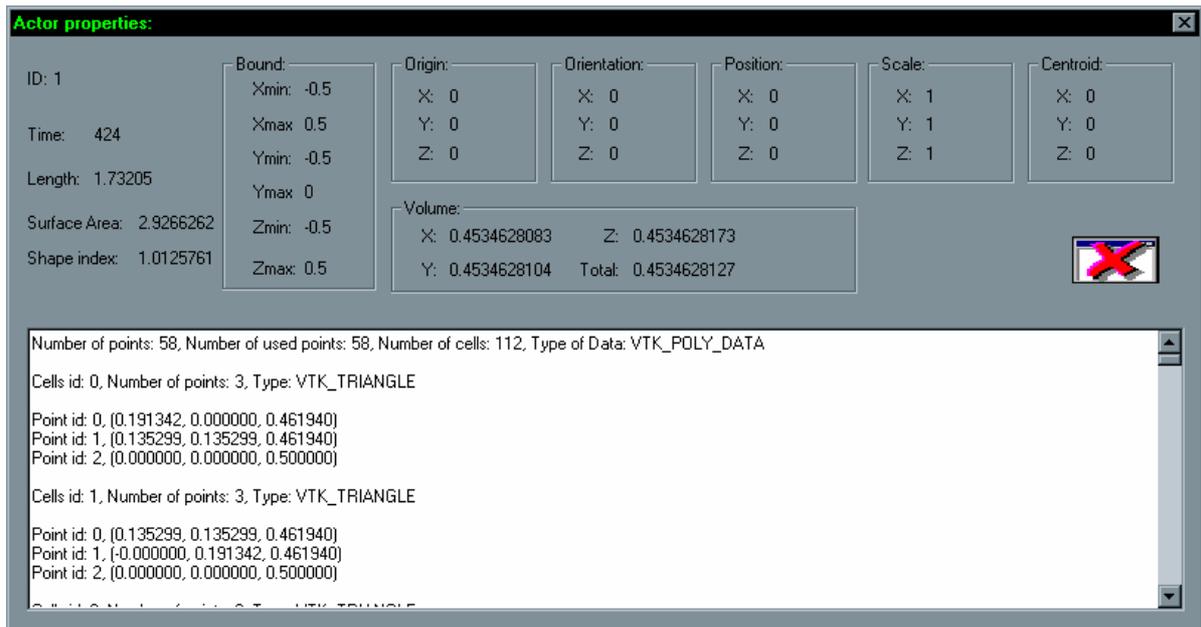


Fig. 6. Visualização das propriedades de um objecto VTK.

Por vezes pode ser útil adicionar ou multiplicar uma imagem com ruído do tipo Gaussiano ou do tipo aleatório, ou criar uma nova imagem com um desses tipos de ruído. Funções para essas operações foram disponibilizadas no sistema, sendo facilmente configuradas.

Vários tipos de filtros [11] foram implementados no sistema; destes podem ser destacados os seguintes: de média, de mediana, de mínimo, de máximo, do mais próximo do máximo e do mínimo; de gradiente [12] de *Roberts*, de *Prewitt*, de *Grad* (diferenças de 1ª ordem) e de *Sobel*; de Laplaciano, e de determinação das passagens por zero do Laplaciano; de convolução; de Gaussiano e de LoG com ou sem normalização.

Para imagens de níveis de cinzento foram também integradas no sistema as operações morfológicas [11] de erosão, de dilatação, de fecho e de abertura.

A detecção de orlas de intensidade utilizando os operadores [12-19] de *Deriche*, de *Shen & Castan*, de *Canny*, e de *Spacek*, é bastante útil em várias aplicações de processamento de imagem. Estes operadores foram também implementados sendo a sua parametrização facilmente conseguida.

Como as operações lógicas e aritméticas são bastante úteis, nomeadamente para a comparação e verificação de resultados, foram também integradas no sistema.

### 6.3 – Análise de imagem

Nesta subsecção são descritas algumas das funções relacionadas com a análise de imagem integradas no banco de desenvolvimento e ensaio. Em primeiro lugar, são descritas as funções de

índole mais geral, seguidamente são apresentadas as funções directamente relacionadas com linhas.

#### 6.3.1 – Genéricas

As funções básicas para a análise de imagem integradas são as seguintes: calcular o histograma de uma imagem, determinar a distância entre dois *pixels*, determinar o máximo e o mínimo valor dos *pixels* presentes, determinar o número de *pixels* que apresentam um determinado valor.

Uma operação que tem muitas vezes utilidade num sistema de análise de imagem é a binarização. Assim foram integradas no sistema as seguintes técnicas: determinação dos *pixels* com valor de brilho não inferior a um limiar especificado ou dentro de um intervalo definido; determinação dos *pixels* com valor de brilho não inferior ao limiar determinado pelo método de *Otsu* [20] ou pelo método de *Kittler* [21]; binarização utilizando-se histerese de amplitude; binarização de objectos pelos métodos de *White & Rohrer's* [22].

Também foram integradas no sistema funções para processar os objectos constituídos por *pixels* de valor uniforme presentes numa imagem, nomeadamente para: esqueletizar [23], determinar as linhas centrais [24], determinar o rectângulo que circunscreve um objecto, realizar o preenchimento, determinar os *pixels* fronteira, anular os *pixels* interiores.

Em muitas aplicações da visão por computador assume uma importância relevante a análise das regiões presentes numa imagem. Assim resolveu-se incluir no sistema uma operação [25] que permite: determinar as regiões presentes, verificar algumas características de

cada região determinada e criar uma nova imagem

a partir de uma região seleccionada (Fig. 7).

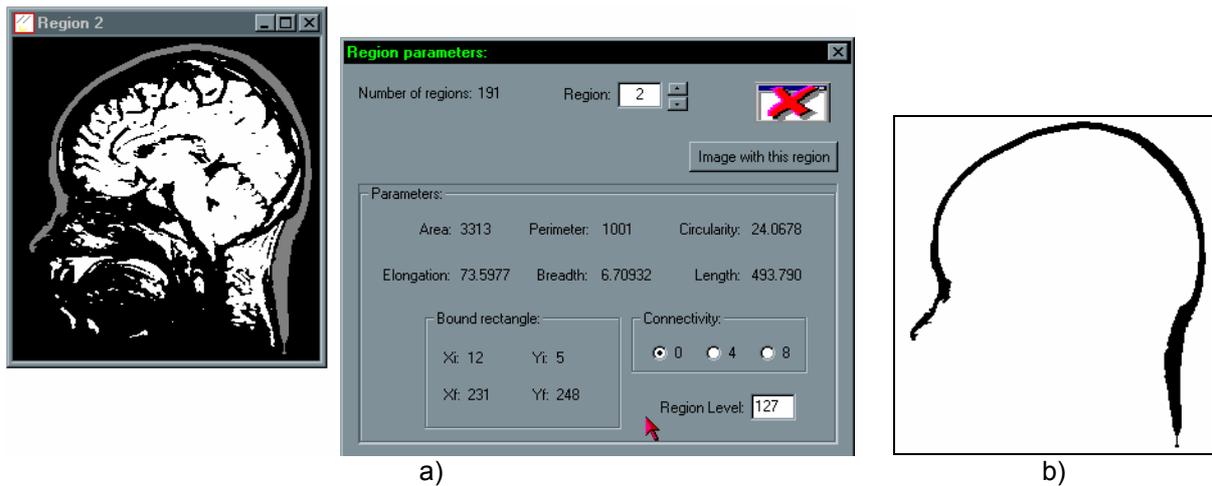


Fig. 7. Análise das regiões de uma imagem: a) Imagem original e alguns parâmetros da região seleccionada; b) Imagem (negada) criada a partir da região seleccionada.

No domínio da segmentação de objectos, incluindo objectos deformáveis, os contornos activos (vulgarmente designados por *snakes*, [1, 24, 26-33]) são bastante utilizados. Com uma importância tão relevante, e com resultados

satisfatórios em vários tipos de aplicações, incluíram-se no sistema algoritmos de contornos activos baseados em regiões e baseados em orlas de intensidade para imagens em níveis de cinzento (Fig. 8).

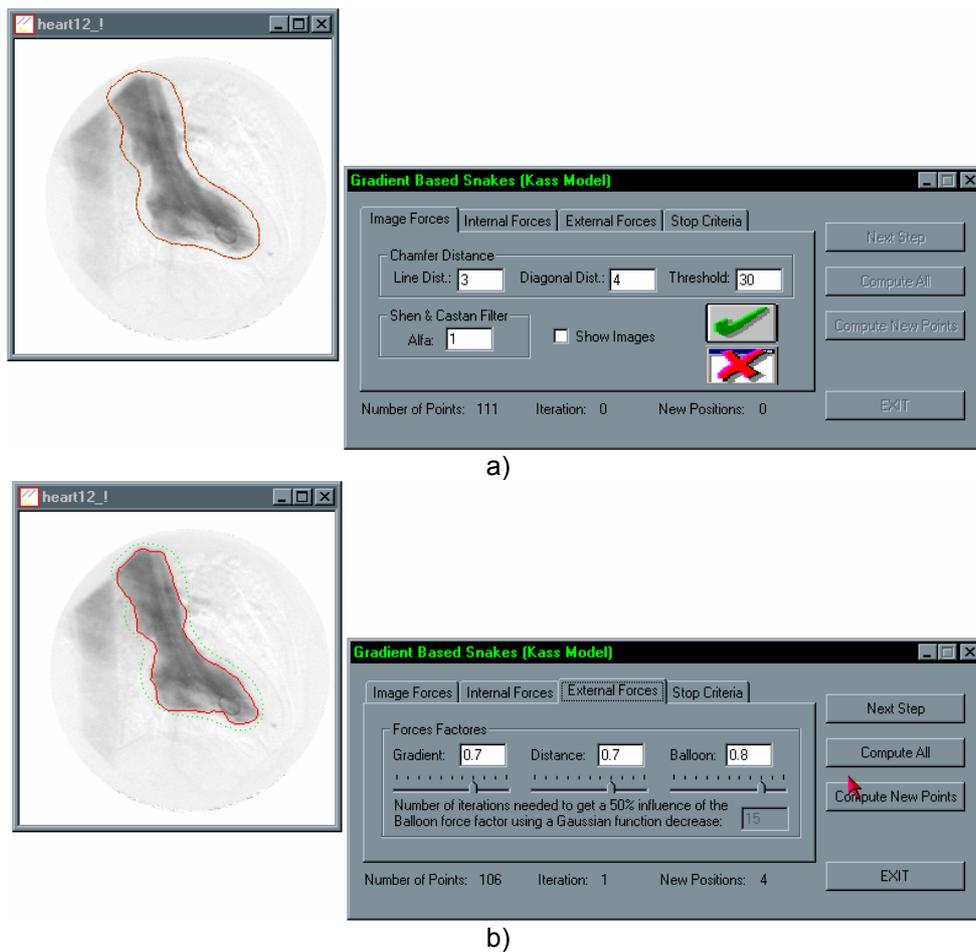


Fig. 8. Segmentação de um objecto utilizando o modelo de *snake* de Kass: a) Definição da *snake* inicial, utilizando o rato, e aplicação do método; b) *Snake* final obtida.

### 6.3.2 – Para linhas

Neste secção descreve-se, de forma sucinta, as funções disponíveis para linhas; detalhes em [12, 34-36].

Em qualquer instante o utilizador pode verificar os parâmetros de um determinado segmento de recta, determinar o número de segmentos de recta presentes, determinar um segmento de recta presente, apagar ou inserir um segmento de recta, determinar os segmentos de recta que têm comprimento superior a um determinado número de *pixels*.

Para detectar, seguir e aproximar as linhas presentes em imagens de orlas, foi implementado um algoritmo [35] que utiliza no seguimento histerese de amplitude e critérios de direcção e, na aproximação poligonal, um algoritmo baseado em faixas dinâmicas.

No sistema desenvolvido dois segmentos de recta podem ser simplificados por dois processos: um que considera, na determinação do segmento de recta resultante, o comprimento e a posição de cada um dos segmentos a fundir [35, 36]; e um outro que determina o segmento de recta resultante através da ligação dos pontos mais extremos dos segmentos de recta a simplificar.

O seguimento de segmentos de recta ao longo de uma sequência é possível utilizando uma implementação de uma metodologia [12, 34, 35] baseada em filtragem de *Kalman*<sup>1</sup>, distância de *Mahalanobis* e em restrições geométricas. Com a implementação utilizada, é possível especificar todos os parâmetros de controlo e verificar, durante a execução, todo o processo de seguimento.

### 6.4 – Computação e visualização gráfica

Uma das características interessantes da aplicação desenvolvida é a integração, num só sistema, de ferramentas de processamento e análise de imagem com ferramentas e entidades habitualmente existentes em computação gráfica. Com esta integração, totalmente transparente, torna-se possível utilizar numa só aplicação imagens *Bitmap* e entidades usuais em computação gráfica e, com a utilização de adequadas conversões, utilizar as ferramentas que estão normalmente disponíveis em cada domínio aumentando-se assim a funcionalidade do sistema global.

Algumas das funções disponíveis para objectos gráficos que podem ser destacadas são: simplificação de objectos, subdivisão das células de um objecto, diminuição do ruído presente por suavização Laplaciana [10], visualização das normais, binarização de um objecto gráfico,

---

<sup>1</sup> São utilizados três filtros de *Kalman* independentes: um para a posição do ponto médio, outro para o comprimento e um terceiro para a direcção.

realização de operações lógicas, visualização da transformação de um objecto gráfico num outro, extracção do contorno exterior e de contornos de isonível, triangulação 2D e 3D de *Delaunay* [10].

No sistema desenvolvido é possível ao utilizador controlar uma série de parâmetros de visualização e de representação dos objectos *VTK* [8, 10] como por exemplo: definir a legenda e a respectiva formatação; activar ou desactivar a identificação dos pontos e das células dos objectos; activar ou desactivar a representação dos eixos coordenados; escolher do tipo de vista pretendida para a cena; seleccionar o tipo de representação desejado (*Points*, *Wireframe*, *Surface*); optar pela interpolação desejada (*Flat*, *Gouraud*, *Phong*); formatar e activar, ou desactivar, a barra de cores para os valores escalares associados aos pontos que constituem os objectos representados; adicionar/remover fontes de luz; definir a luz ambiente da cena. As cenas *VTK* podem também ser exportadas em vários formatos, aumentando assim a possibilidade de ligação do sistema a outras aplicações existentes na área da computação gráfica.

### 6.5 – Especificas aos objectos deformáveis

Nesta subsecção serão descritas sucintamente as funções disponíveis no banco de desenvolvimento e ensaio consideradas como mais específicas aos objectos deformáveis. Em primeiro lugar, serão descritas as funções relacionadas com os documentos do tipo imagem e seguidamente as relacionadas com os documentos *VTK*.

#### 6.5.1 – Documentos imagem

##### i) Para contornos

A detecção de contornos dos objectos presentes numa imagem foi implementada utilizando-se histerese da amplitude, com a possibilidade de efectuar a aproximação poligonal dos contornos detectados através do algoritmo de faixas dinâmicas, e da determinação da curvatura e do *chain-code* [11].

A qualquer momento o utilizador pode inserir ou apagar um contorno, determinar o número de contornos presentes, verificar algumas características de um contorno, aplicar a um contorno uma dada transformação rígida 2D.

A determinação da cobertura convexa (*Convex Hull*) [10] de um conjunto de pontos foi também implementada no sistema, pois é uma operação bastante utilizada no domínio dos objectos deformáveis.

No banco de desenvolvimento e ensaio foram integradas funções que permitem a análise da vibração de um contorno em modo livre utilizando os modos próprios da forma do objecto

[1, 37-40], ou a modelização física do contorno por intermédio de um elemento finito isoparamétrico 2D de *Sclaroff* [1, 41-44] ou por intermédio de um conjunto de elementos finitos axiais lineares agrupados [1, 45]. Após o cálculo dos modos próprios de vibração, o utilizador pode visualizá-los, determinar os deslocamentos obtidos considerando um determinado conjunto de modos e criar uma imagem considerando esses deslocamentos.

A determinação das correspondências entre dois contornos utilizando o método de mapeamento segundo a distância mínima [1, 37] foi implementada no sistema desenvolvido. Além das correspondências obtidas é possível verificar a estimacão para a transformação rígida existente entre os dois contornos, obtida utilizando uma implementação do método de *quaternions* unitários [1, 46, 47].

Também foi implementada uma função para determinar as correspondências entre dois contornos utilizando o método baseado na análise modal da forma [1, 37-40], ou utilizando modelização física dos objectos, por intermédio de dois elementos finitos isoparamétricos 2D de *Sclaroff* [1, 41-44] ou por intermédio de dois

conjuntos de elementos finitos axiais lineares agrupados [1, 45] com posterior emparelhamento modal. Com esta função é possível determinar-se as correspondências entre os dois contornos, a transformação geométrica rígida existente entre os dois, estimar os deslocamentos nodais por minimização da energia de deformação do modelo físico considerado, determinar o valor da energia de deformação necessária para os alinhar, criar uma imagem com os resultados obtidos e alterar as condições utilizadas para o emparelhamento modal.

O diálogo disponível para o utilizador verificar os resultados de emparelhamento, alterar as condições do emparelhamento modal, criar as imagens de resultados pretendidas e para executar a estimacão dos deslocamentos nodais e cálculo da energia de deformação, está representado a título de exemplo na Fig. 9. Analisando a referida figura, é possível comprovar as boas condições de ensaio disponíveis no sistema desenvolvido.

Também a título de exemplo, na Fig. 10 estão representados dois contornos originais e uma imagem com as correspondências obtidas entre estes.

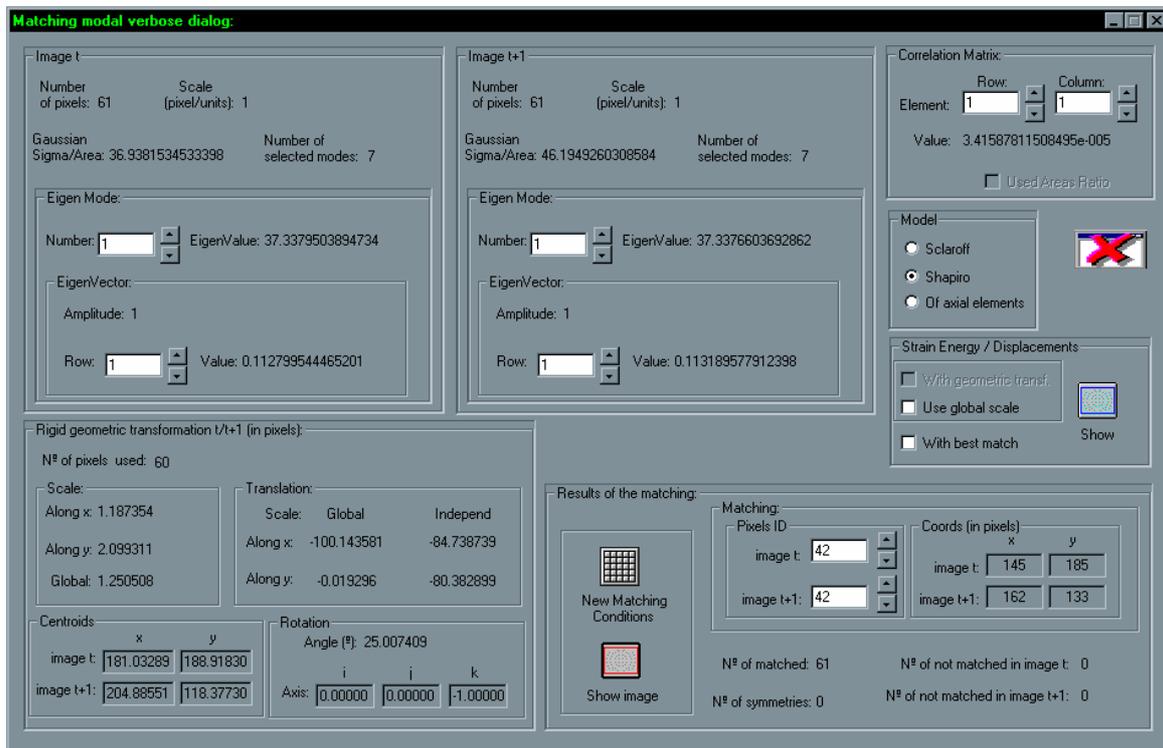


Fig. 9. Verificação dos resultados obtidos na determinação das correspondências entre dois contornos.

## ii) Para superfícies

### a) Criação de superfícies

Para inserir uma superfície, o utilizador especifica as localizações dos seus nodos na

imagem original, sendo o nível de brilho de cada *pixel* seleccionado tomado como a sua terceira coordenada.

Em vez de utilizar a inserção manual, é também possível criar automaticamente uma superfície por amostragem rectangular ou

amostragem adaptativa [1] de uma imagem *bitmap*. Na amostragem adaptativa também é considerado o nível de intensidade de cada *pixel* como a sua terceira coordenada, sendo a

amostragem realizada por análise dos perfis de intensidade radiais centrados nos pontos de máximos locais de brilho.

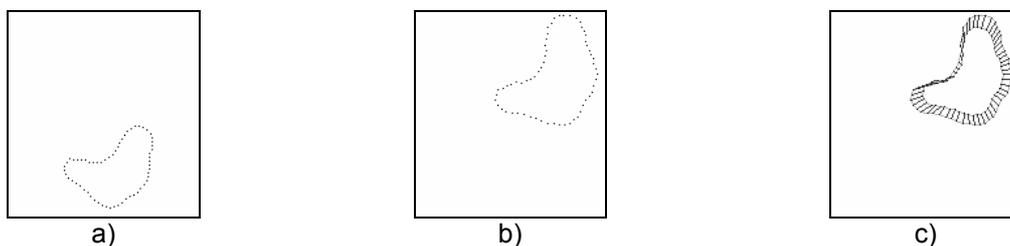


Fig. 10. Determinação das correspondências entre dois contornos: a) Contorno 1; b) Contorno 2; c) Correspondências determinadas após aplicação aos pixels do contorno 1 da rotação e da translação determinadas.

### b) Utilitárias

Em qualquer instante o utilizador pode apagar uma superfície, realizar a normalização das coordenadas dos *pixels* de uma superfície ao longo do eixo z, atribuir uma cota uniforme aos *pixels* que constituem uma superfície, determinar o número de *pixels* que a constituem, identificar um determinado *pixel* de uma superfície e medir os seguintes parâmetros: número de *pixels*, coordenadas do primeiro *pixel* e do centróide, dimensões máximas e mínimas.

No âmbito dos objectos deformáveis é muitas vezes necessário aplicar a uma superfície uma transformação geométrica rígida 3D, esta transformação foi também implementada no sistema.

### c) Reconstrução/determinação da estrutura

Após a determinação dos pontos que constituem uma dada superfície é por vezes necessário determinar a estrutura de malha triangular que melhor se adapta a tal conjunto de pontos. Assim foram implementadas funções para realizar a triangulação 2D de *Delaunay* [10], considerando ou não a cota dos pontos originais como terceira coordenada dos pontos do objecto resultante, e a triangulação 3D de *Delaunay*.

A ferramenta para reconstrução de superfícies existente no *VTK* foi também disponibilizada no sistema. Assim, sempre que um dado conjunto de pontos não estruturados pertença a uma superfície de um sólido 3D, pode-se reconstruir a superfície utilizando-se esta função.

### d) Emparelhamento de superfícies

A análise da vibração de uma superfície em modo livre foi também implementada no sistema. Esta análise pode ser efectuada utilizando-se os modos próprios da forma, ou modelização física, por intermédio de um único elemento finito

isoparamétrico 3D de *Scleroff* ou por intermédio de um conjunto de elementos finitos axiais lineares agrupados. Após o cálculo dos modos próprios de vibração o utilizador pode: visualizá-los; determinar os deslocamentos obtidos considerando um determinado conjunto de modos; criar uma imagem *Bitmap*, considerando os deslocamentos associados a um único ou a um conjunto de modos.

A determinação das correspondências entre duas superfícies utilizando o método de mapeamento segundo a distância mínima foi também implementada no banco de desenvolvimento e ensaio.

Também foi implementada a determinação das correspondências entre duas superfícies utilizando o método baseado na análise modal da forma, ou utilizando o método baseado em modelização física, por intermédio de dois elementos finitos isoparamétricos 3D de *Scleroff* ou por intermédio de dois conjuntos de elementos finitos axiais lineares agrupados, e posterior emparelhamento modal. Com esta implementação é possível: determinar-se as correspondências; estimar a transformação geométrica rígida 3D existente; estimar os deslocamentos nodais, por minimização da energia de deformação do modelo físico considerado; determinar o valor da energia de deformação necessária para as alinhar; criar uma imagem *Bitmap* com os resultados obtidos; alterar as condições do emparelhamento modal e visualizar os novos resultados obtidos.

### 6.5.2 – Documentos *VTK*

Para documentos do tipo *VTK* foram integradas mais algumas funções no sistema desenvolvido relacionadas com a análise de vibrações de um objecto em modo livre, a determinação das correspondências entre dois objectos, a estimação da transformação rígida existente, a determinação dos deslocamentos nodais e o valor da energia de deformação necessária para os alinhar. Seguidamente são

descritas de forma sumária as funções implementadas.

Foi implementada a análise (2D ou 3D) da vibração de um objecto em modo livre utilizando os modos próprios da forma, ou utilizando a modelização física do objecto. Após o cálculo dos modos próprios de vibração, o utilizador pode visualizá-los, determinar os deslocamentos obtidos considerando um determinado conjunto de modos e criar um novo documento considerando os deslocamentos associados a este conjunto.

A determinação das correspondências entre dois objectos utilizando o método de mapeamento segundo a distância mínima foi também implementada. Nesta determinação pode-se considerar apenas as coordenadas do objecto no plano  $xy$  (utilizando-se a metodologia 2D), ou considerar-se as três coordenadas de cada nodo dos objectos (utilizando-se a metodologia 3D). Além das correspondências obtidas, o utilizador pode visualizar a transformação rígida estimada,

alterar as condições de emparelhamento e criar novos documentos com os resultados obtidos.

Também foi implementada a determinação (2D ou 3D) das correspondências entre dois objectos *VTK* utilizando o método baseado na análise modal da forma, ou o método baseado em modelização física, e posterior emparelhamento modal. Com a respectiva função é possível verificar-se as correspondências obtidas, analisar e estimar a transformação geométrica rígida existente, estimar os deslocamentos nodais, por minimização da energia de deformação do modelo físico considerado, determinar o valor da energia de deformação necessária para os alinhar, criar um novo documento com os resultados obtidos, e alterar as condições de emparelhamento modal, visualizando os novos resultados obtidos.

Na Fig. 11 é possível verificar-se um exemplo da representação das correspondências obtidas entre dois objectos *VTK*.

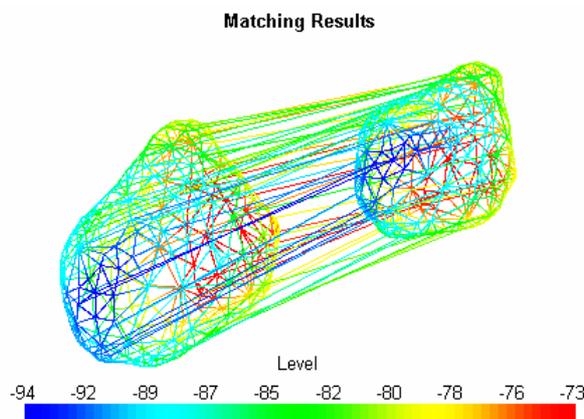


Fig. 11. Exemplo da representação das correspondências determinadas entre dois objectos *VTK*.

## 7 – Sumário e conclusões

O sistema descrito neste artigo apresenta duas características importantes. A primeira é a integração, num mesmo sistema, de entidades e de ferramentas normalmente consideradas em dois domínios distintos: o da visão por computador e o da computação gráfica; esta integração torna possível aos utilizadores disporem e operarem num só sistema com um vasto conjunto de entidades e de ferramentas. A segunda característica que o mesmo apresenta é a facilidade com que as funções incorporadas podem ser parametrizadas e controladas; os resultados obtidos podem ser verificados e visualizados, conseguindo-se assim realizar ensaios exaustivos dos métodos envolvidos.

Actualmente a aplicação já incorporou contribuições de outros investigadores do grupo e serviu de sistema de desenvolvimento e ensaio a vários alunos da Faculdade de Engenharia da Universidade do Porto, nos seus trabalhos de

licenciatura e de mestrado, na área da análise e processamento de imagem. Confirmou-se que a integração de novas funções, descrita de forma breve neste artigo, é fácil e transparente o que torna mais rápida a adaptação de novos investigadores.

O sistema desenvolvido também tem sido utilizado como aplicação de demonstração em cursos de processamento e análise de imagem, revelando-se uma aplicação com valor na aprendizagem.

Com o sistema apresentado neste artigo atingiu-se um dos objectivos principais que orientou o seu desenvolvimento: a criação de um sistema para a análise e processamento de imagem, que constituísse, por si só, uma aplicação independente e utilizável em diferentes contextos e, ao mesmo tempo, permitisse que outros investigadores facilmente a utilizassem para o desenvolvimento dos seus próprios algoritmos, ensaio e posterior incorporação.

Ao longo do desenvolvimento do sistema

alguns dos seus algoritmos, mais pesados em termos computacionais, têm sido paralelizados<sup>2</sup> [24, 26, 48-52], utilizando-se o ambiente *WPVM – Windows Parallel Virtual Machine* [53], sendo os resultados já obtidos bastante encorajadores.

Os investigadores que desejem utilizar o sistema, assim como aceder ao seu código fonte, podem solicita-lo aos autores.

## Referências

1. Tavares, J.M.R.S., *Análise de Movimento de Corpos Deformáveis usando Visão Computacional*, in *Faculdade de Engenharia*. 2000, Universidade do Porto. <http://www.fe.up.pt/~tavares/downloads/publications/teses/tese%20phd.zip>
2. Stroustrup, B., *The C++ Programming Language*. 1995: Addison-Wesley Publishing Company.
3. Young, M.J., *Mastering Microsoft Visual C++ 6*. 1998: Sybex.
4. Davies, R., *Newmat, A matrix library in C++*. 1999. <http://webnz.co.nz/robert/>
5. Rimmer, S., *Supercharged Bitmapped Graphics*. 1991: Windcrest/McGraw-Hill.
6. others, J.S.a., *CImage Class Version 1.2*. 1997. <http://web.ukonline.co.uk/Members/julian.smart>
7. Lane, T.G., *Jpeg Library*. 1996.
8. Will Schroeder, Ken Martin, and B. Lorensen, *The VTK User's Guide*. 1999: Kitware Inc. <http://www.kitware.com>
9. Will Schroeder, Ken Martin, and B. Lorensen. *The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization*. in *Visualization '96*. 1996.
10. Will Schroeder, Ken Martin, and B. Lorensen, *The Visualization Toolkit*. 2nd ed. 1998: Prentice Hall.
11. Schalkoff, R.J., *Digital Image Processing and Computer Vision*. 1989: John Willey & Sons, Inc.
12. Tavares, J.M.R.S., *Obtenção de Estrutura Tridimensional a Partir de Movimento de Câmara*, in *Faculdade de Engenharia*. 1995, Universidade do Porto. <http://www.fe.up.pt/~tavares/downloads/publications/teses/tese%20msc.zip>
13. Correia, M.F.P.V., *Análise de Movimento em Sequências de Imagens*, in *Departamento de Engenharia Electrotécnica e de Computadores*. 1995, Faculdade de Engenharia da Universidade do Porto.
14. Deriche, R., *Using Canny's Criteria to derive a Recursively Implemented Optimal Edge Detector*. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1987. **1**(2): p. 167/187.
15. Haralick, R.M., *Digital Step Edges from Zero-crossings of Second Directional Derivatives*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1984. p. 58/68.
16. J. S. Chen and G. Medioni, *Detection, Localization and Estimation of Edges*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1989. p. 191/198.
17. V. S. Nalwa and T.O. Binford, *On detecting edges*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986. p. 699/714.
18. J. Shen and S. Castan, *An Optimal Linear Operator for Step Edge Detection*, in *CVGIP: Graphical Models and Image Processing*. 1992. p. 112/133.
19. Canny, J.F., *A Computational Approach for Edge Detection*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986. p. 679/698.
20. Otsu, N., *A Threshold Selection Method from Gray-Level Histograms*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1978. p. 62/66.
21. Josef Kittler and J. Illingworth, *Minimum error thresholding*. *Pattern Recognition Letters*, 1986. **19**: p. 41/47.
22. J. M. White and G.D. Rohrer, *Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction*. *IBM Journal of Research and Development*, 1983. **27**(4): p. 400/411.
23. O'Gorman, L., *KxK Thinning*. *CVGIP*, 1990. **51**: p. 195/215.
24. Barbosa, J.M.G., *Paralelismo em Processamento e Análise de Imagens Médicas*, in *Faculdade de Engenharia*. 2000, Universidade do Porto.
25. Tor Lonnestad and O. Milvang, *XITE: X-based Image Processing Tools and Environment User's Manual*. 1992.
26. G. Barbosa, João Manuel Tavares, and A.J. Padilha. *Paralellisation of active contours*. in *RecPad'96 - 8th Portuguese Conference on Pattern Recognition*. 1996. Guimarães, Portugal.
27. Michael Kass, Andrew Witkin, and D. Terzopoulos, *Snakes: Active contour models*. *International Journal of Computer Vision*, 1988: p. 321/331.
28. Olof Henricsson and W. Neuenschwander, *Controlling growing snakes by using key-points*, in *IEEE 1051-4651/94*. 1994, IEEE.
29. W. Neuenschwander, G.S. P. Fua, and O. Kübler, *Making Snakes Converge from Minimal Initialization*, in *IEEE 1051-4651/94*. 1994, IEEE.

---

<sup>2</sup> Actualmente, as paralelizações implementadas são de dados existindo as devidas funções para os distribuir. A máquina virtual é definida de forma automática e inclui os processadores disponíveis na rede. O número de processadores escolhidos para cada tarefa é determinado dinamicamente (em *runtime*); se esse valor for um, então é executada a versão sequencial do algoritmo. Esta tarefa é realizada por um escalonador que trata de toda a parte de execução.

30. Song Chun Zhu and A. Yuille, *Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1996.
31. B. Basclé and R. Deriche. *Features Extraction using Parametric Snakes*. in *IEEE 11th International Conference on Pattern Recognition*. 1992. Netherlands.
32. Andrew Blake and M. Isard, *Active Contours*. 1998: Springer-Verlag.
33. Cohen, L.D., *NOTE On active contour models and balloons*, in *CVGIP: Image Understanding*. 1991. p. 211/218.
34. João Manuel R. S. Tavares and A.J. Padilha. *Matching lines in image sequences with geometric constraints*. in *RecPad'95 - 7th Portuguese Conference on Pattern Recognition*. 1995. Aveiro, Portugal.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/recpad%2095%201.pdf>
35. Tavares, J.M.R.S., *Algumas Ferramentas para Visão Tridimensional por Computador*. 1995, Faculdade de Engenharia da Universidade do Porto.  
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/ferramentas%203d.zip>
36. João Manuel R. S. Tavares and A.J. Padilha. *A new approach for merging edge line segments*. in *RecPad'95 - 7th Portuguese Conference on Pattern Recognition*. 1995. Aveiro, Portugal.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/recpad%2095%202.pdf>
37. Larry Shapiro and J.M. Brady, *Feature-based correspondence: an eigenvector approach*, in *Butterworth-Heinemann Ltd*. 1992.
38. Shapiro, L., *Report: Towards A Vision-Based Motion Framework*. 1991, Robotics Research Group, Department of Engineering Science, Oxford University.
39. Larry Shapiro and J.M. Brady, *A Modal Approach to Feature-based Correspondence*. 1992, Robotics Research Group, Department of Engineering Science, Oxford University.
40. João Manuel R. S. Tavares and A.J. Padilha, *Relatório Interno: Apresentação de uma Abordagem Modal para a Determinação da Correspondência entre Pixels de dois Contornos Não Rígidos*. 1997, Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto.  
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/shapiro.zip>
41. João Manuel R. S. Tavares and A.J. Padilha, *Relatório Interno: Abordagem Modal para a Determinação da Correspondência entre Objectos Não Rígidos Utilizando o Modelo FEM de Sclaroff*. 1998, Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto.  
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/sclaroff.zip>
42. Stan Sclaroff and A. Pentland, *Modal Matching for Correspondence and Recognition*, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1995.
43. Sclaroff, S.E., *Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals*, in *Massachusetts Institute of Technology*. 1995.
44. Stan Sclaroff and A. Pentland, *Object Recognition and Categorization Using Modal Matching*. 1994, Massachusetts Institute of Technology, Media Laboratory.
45. João Manuel R. S. Tavares and A.J. Padilha, *Relatório Interno: Apresentação de uma Modelização por Elementos Finitos Lineares do Tipo Axial*. 1998, Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto.  
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/axial.zip>
46. Horn, B.K.P., *Closed-form Solution of Absolute Orientation using Unit Quaternions*. *Journal of the Optical Society of America A*, 1987. 4: p. 629/642.
47. João Manuel R. S. Tavares and A.J. Padilha, *Relatório Interno: Apresentação de um Método para a Determinação da Transformação Rígida entre dois Contornos Constituídos por Pixels*. 1997, Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto.  
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/horner.zip>
48. J. Barbosa, J. Tavares, and J. Padilha. *Self-Tuned Parallel Processing System for Heterogeneous Clusters*. in *International Conference on Parallel and Distributed Processing Techniques and Applications*. 2001. USA: CSREA Press.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/pdpta01.pdf>
49. J. Barbosa, J. Tavares, and J. Padilha. *A Group Block Distribution Strategy for a Heterogeneous Machine*. in *International Symposium on Parallel and Distributed Computing and Networks (AI2002)*. 2002. Áustria.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/iasted.zip>
50. Jorge G. Barbosa, João Manuel Tavares, and A.J. Padilha. *Parallel Image Processing System on a Cluster of Personal Computers*. in *VECPAR 2000 - 4th International Meeting on Vector and Parallel Processing*. 2000. Faculdade de Engenharia da Universidade do Porto, Porto, Portugal.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/vecpar00.zip>
51. Jorge G. Barbosa, João Manuel Tavares, and A.J. Padilha. *Linear Algebra Algorithms in a Heterogeneous Cluster of Personal Computers*. in *HCW 2000 - 9th Heterogeneous*

*Computing Workshop*. 2000. Cancun, Mexico.  
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/hcw00.pdf>

52. Jorge Barbosa and A. Padilha. *Algorithm-Dependant Method to Determine the Optimal Number of Computers in Parallel Virtual Machines*. in *Vector and Parallel Processing -*

*VECPAR'98, Third International Conference*. 1998. Porto, Portugal.

53. Alves, A., et al. *WPVM: parallel computing for the people*. in *HPCN'95 High Performance Computing and Network Conference*. 1995: Springer Verlag LNCS.  
<http://dsg.dei.uc.pt/wpvm>