

Relatório Interno:

Introdução à Plataforma de Processamento e Análise de Imagem e Computação Gráfica - CMIS

João Manuel R. S. Tavares

INEB – Instituto de Engenharia Biomédica, Laboratório Sinal e Imagem
INEGI – Instituto de Engenharia e Gestão Industrial, Laboratório de Óptica e Mecânica Experimental
Universidade do Porto, Faculdade de Engenharia, Departamento de Engenharia Mecânica e Gestão Industrial

Rua Dr. Roberto Frias, s/n 4200-465 PORTO – PORTUGAL

Tel. +351225081487

email: tavares@fe.up.pt

url: <http://www.fe.up.pt/~tavares>



**Universidade
do Porto**

**Faculdade de
Engenharia**

FEUP



LOME

Laboratory of Optics and Experimental Mechanics

INEGI / FEUP

Julho 2003

Resumo

Desde 1995 tem vindo a ser desenvolvida na *FEUP* – Faculdade de Engenharia da Universidade do Porto, uma plataforma de análise e processamento de imagem e de computação gráfica designada por *CMIS*.

A opção pelo desenvolvimento de uma nova aplicação, em vez da adaptação de uma aplicação já existente, deveu-se à constatação, por parte do seu autor, 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 base 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.

Procedeu-se então à implementação de uma plataforma de desenvolvimento e ensaio para sistemas operativos *Microsoft Windows*, a linguagem utilizada foi a *C++* e o ambiente de desenvolvimento foi o *Microsoft Visual C++*. Para aumentar as capacidades da plataforma foram incorporadas algumas bibliotecas de domínio público: para operações com imagens *bitmap* as bibliotecas *Cimage* e *Jpeglib*, para cálculo matricial a biblioteca *Newmat*, e para objectos gráficos a biblioteca *VTK*.

A filosofia utilizada no desenho da plataforma permite que a mesma seja facilmente configurável a diferentes géneros de utilizadores, e que os mais diversos tipos de investigadores a utilizem sem grande dificuldade no desenvolvimento e ensaio dos seus próprios algoritmos. Actualmente a plataforma é bastante utilizada por vários alunos da *FEUP*, quer ao nível da Licenciatura quer ao nível de Cursos de Mestrado, para desenvolverem os seus trabalhos, tendo-se verificado as boas condições que a mesma disponibiliza. Deste modo, o número de funções disponíveis na plataforma tem vindo a crescer rapidamente, tendo-se incluindo um grupo relevante de funções adequadas para objectos deformáveis o que também a torna bastante adequada para esse domínio.

Neste relatório é realizada uma descrição da plataforma desenvolvida, assim é descrita a filosofia adoptada, indicadas as entidades actualmente suportadas, descritas de forma resumida algumas das funções já incorporadas de índole mais geral, são indicados os

algoritmos disponíveis mais específicos aos objectos deformáveis e finalmente, em anexo, é descrito o processo de compilação do projecto *CMIS* no ambiente de desenvolvimento *Microsoft Visual C++*.

Palavras-chave – Processamento e análise de imagem, visão por computador, computação gráfica, objectos deformáveis, *Microsoft Visual C++*.

Abstract

Since 1995 it has been developed, in *FEUP* – Faculdade de Engenharia da Universidade do Porto, a platform for analysis and image processing and for computer graphics designed by *CMIS*.

The option for the development of a new application, instead of the adaptation of an already existent, was due to the lack of platforms 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 base 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, they were also important arguments that conditioned the decision for the developing of a new system.

So it was proceeded the implementation of a development and rehearsal platform for *Microsoft Windows* operative systems, the used language is the C^{++} and the development system is the *Microsoft Visual C^{++}*. To increase the capacities of the platform they were incorporate some public domain libraries: for bitmap images operations the *Cimage* and *Jpeglib*, for matrices computation the *Newmat*, and for graphics objects the *VTK*.

The philosophy used in the platform drawing allows that the same is easily configured to different users, and that several types of investigators use it in the development and rehearsal of its own algorithms without great difficulty. Actually the platform is quite used by several students of *FEUP*, for their algorithms developing, having been verified the good conditions that the same offers. This way, the number of available functions in the platform has been coming to grow quickly, and an important group of functions adequate for deformable objects also was being included what turns it quite adapted for this domain.

In this report a description of the developed platform is accomplished, the adopted philosophy is described, indicated the entities actually supported, briefly described some of the generally functions already incorporated, the available algorithms more suitable for deformable objects are presented and finally, in appendix, the *CMIS* project compilation process in the *Microsoft Visual C^{++}* development system is described.

Keywords – Analysis and image processing, computer vision, computer graphics, deformable objects, *Microsoft Visual C^{++}*.

Índice

1 – Introdução	1
2 – Bibliotecas de domínio público integradas	2
3 – Entidades suportadas	4
4 – Interface	5
5 – Integração de novas funções	9
6 – Selecção de funções disponíveis	11
6.1 – Utilitárias	11
6.1.1 – Documentos imagem	11
6.1.2 – Documentos VTK	12
6.1.3 – Configuração	13
6.2 – Processamento de imagem	14
6.3 – Análise de imagem	16
6.3.1 – Genéricas	16
6.3.2 – Para linhas	19
6.4 – Computação e visualização gráfica	22
6.5 – Específicas aos objectos deformáveis	29
6.5.1 – Documentos imagem	30
6.5.1.1 – Para contornos	30
6.5.1.2 – Para superfícies	33
6.5.2 – Documentos VTK	38
7 – Sumário e conclusões	40
Referências	42
Anexo: Compilação do CMIS	44

1 – Introdução

Desde 1995 tem vindo a ser desenvolvido na *FEUP* – Faculdade de Engenharia da Universidade do Porto – uma plataforma de processamento de análise de imagem e computação gráfica designada por *CMIS* [1, 2].

O carácter fechado da maior parte das aplicações existentes para processamento e análise de imagem, bem como o propósito de incorporar novas metodologias por nós desenvolvidas 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 de novos algoritmos, 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++* [3] em ambiente *Microsoft Visual C++* [4]. 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*.

Para a estrutura a utilizar entendeu-se que uma solução adequada seria considerar a plataforma como um conjunto de módulos; assim, existe um módulo principal, ou base, que contém as funções básicas necessárias a um qualquer sistema mínimo de processamento e análise de imagem e outros módulos que disponibilizam funções mais específicas a determinada aplicação. Esta estrutura modular facilita a manutenção e a integração de novas funções.

A acessibilidade por parte do utilizador aos módulos mais específicos, e às funções que os constituem, é possível por configuração da aplicação. Esta configuração, ao nível dos menus e das barras de ferramentas, pode ser realizada durante a execução e permite ajustar o sistema a diferentes tipos de utilizadores e aplicações.

Uma vantagem da plataforma desenvolvida 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.

Na fase actual da plataforma já foram integradas múltiplas contribuições de vários

investigadores, nomeadamente: para a leitura e escrita de imagens com formato diferente do *Bitmap*, [5]; para a escrita em ficheiro dos parâmetros opcionais para as funções disponíveis, para a sua leitura a partir de ficheiros, para o seu armazenamento no fim de cada execução da plataforma e posterior recuperação no início de cada execução [6]; para a segmentação de imagens utilizando contornos activos [6, 7]; para a determinação dos pontos de um contorno sobre uma grelha rectangular [6]; para o mecanismo de configuração da plataforma [6]. Este esforço distribuído tem permitido aumentar rapidamente a funcionalidade do sistema, tendo-se verificado que a integração de novas funções é bastante fácil.

Desde há algum tempo a plataforma tem vindo a ser utilizada como sistema de demonstração em cursos de processamento e análise de imagem, e vários alunos, de licenciatura e de mestrado da *FEUP*, têm-na utilizado como sistema base para o desenvolvimento e ensaio dos seus algoritmos. Este tipo de uso tem vindo a comprovar o valor do sistema em contexto de aprendizagem e a demonstrar a facilidade com que novas funcionalidades podem ser integradas.

Nas secções seguintes serão identificadas as bibliotecas de domínio público incorporadas na plataforma, apresentadas as entidades suportadas, exposta a interface do sistema, explicado o modo como se processa a integração de uma nova função na plataforma, serão discriminadas as funções disponíveis de âmbito geral e também as funções mais específicas ao domínio dos objectos deformáveis¹. Em anexo está descrito o processo de compilação do projecto *CMIS* no ambiente de programação *Microsoft Visual C++*.

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 o sistema base quer para o mais específico ao domínio dos objectos deformáveis.

Para cálculo matricial incorporou-se a biblioteca *Newmat* escrita em C^{++} , na sua versão *11*, [8]; desta forma, as seguintes operações matriciais tornaram-se disponíveis: multiplicação, soma, diferença, concatenação, inversão, transposição, conversão entre tipos diferentes, submatrizes, determinantes, decomposição de *Cholesky*, triangulação *QR*, decomposição *SVD*, valores próprios de uma matriz simétrica, ordenação, transformada rápida de *Fourier*, e interface para os algoritmos do livro *Numerical Recipes in C* [9]. Os tipos de matrizes definidas nesta biblioteca são: rectangulares, triangulares superior e inferior, diagonais, simétricas, de banda, de banda superior e inferior, simétricas de banda, vectores linha e

¹ Muitos dos algoritmos incorporados na referida plataforma permitem que esta possa ser considerada, além de uma aplicação genérica de processamento e análise de imagem e computação gráfica, um banco adequado para o ensaio e estudo de métodos de modelização para objectos deformáveis. Assim as funções mais particulares a este domínio assumirão neste relatório um maior destaque.

coluna. Foram ainda acrescentadas funções para escrita em ficheiro e apresentação dos elementos para cada um destes tipos de matriz.

Actualmente existem vários formatos de ficheiros de imagem utilizados nos mais diversos tipos de aplicações de processamento e análise de imagem. Quando se iniciou o desenvolvimento da plataforma base optou-se pela utilização do formato *BMP* como formato nativo. Rapidamente se constatou que seria desejável incluir suporte para outros formatos; destes destacavam-se [10] 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 [11] e *Jpeglib*, na sua versão 6 [12]. Desta forma a plataforma, desde a sua versão base, suporta este tipo de formatos de ficheiros de imagem.

No decurso do trabalho de desenvolvimento 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;
- segmentação e amostragem de uma entidade poligonal;
- realização de operações lógicas e determinação das fases intermédias (*morphing*) entre duas estruturas;
- identificação dos vértices e dos centros das células que constituem uma dada entidade;
- determinação de algumas características de um objecto poligonal;
- filtragem de uma estrutura;
- representação das normais nos vértices de um objecto; etc.

Dada a sua grande utilização e as suas características de robustez, capacidade, eficiência,

contínua actualização, elevado suporte e manutenção, existência de uma versão em linguagem C^{++} facilmente portada para o ambiente de programação *Microsoft Visual C^{++}* e o facto de ser de domínio público, a escolha recaiu sobre a biblioteca *VTK, The Visualization Toolkit* [13-15]. Esta biblioteca gráfica, na sua versão 4.2.2 em C^{++} , foi incorporada na plataforma, estando assim as suas entidades e algumas das suas funções disponíveis em diferentes módulos do sistema.

Na *Figura 1* estão identificadas as bibliotecas de domínio público integradas na plataforma de desenvolvimento e ensaio.

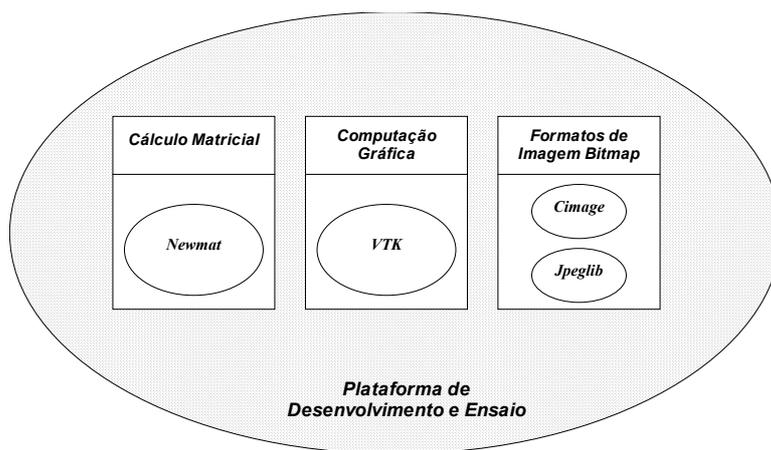


Figura 1 – Identificação das bibliotecas de domínio público integradas na plataforma de desenvolvimento e ensaio - CMIS.

Além das quatro bibliotecas referidas foram também incluídos alguns algoritmos de análise e processamento de imagem que se considerou deverem ser disponibilizados num sistema base; assim foram integrados alguns algoritmos de outros investigadores do grupo: de funções básicas [16, 17] e de detecção de orlas de intensidade [18]; e também foram portados alguns algoritmos incluídos no sistema *XITE – X-based Image Processing Tools and Environment*, [19]. As ferramentas de detecção e seguimento de linhas em imagens de orlas, de simplificação e de seguimento em sequências de imagem desenvolvidas e implementadas durante o trabalho de dissertação de mestrado do autor deste relatório [20-23] foram também portadas para a nova aplicação.

3 – Entidades suportadas

Actualmente várias entidades são suportadas na plataforma 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 da plataforma foram 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*.

Para cada tipo de documento estão suportadas as respectivas funções de escrita em ficheiro e de leitura, de cópia de entidades e de conversão entre algumas das entidades suportadas.

Uma imagem *Bitmap* pode originar um documento *VTK* através de uma sua amostragem e um documento *VTK* pode ser copiado para um documento imagem através da criação de uma imagem *Bitmap* com a cópia do conteúdo de uma das suas janelas de visualização.

Na *Figura 2* estão representados, de forma gráfica, os dois tipos de documentos implementados na plataforma, as entidades que os podem constituir e as operações de cópia, escrita e de leitura, suportadas.

4 – Interface

A interface do sistema segue a filosofia geral das aplicações desenvolvidas para os sistemas operativos *Microsoft Windows*. Dos elementos que a constituem destacam-se os seguintes: a zona dos menus, os menus emergentes de contexto, as barras de ferramentas e a barra de estado (*Figura 3*).

Estes elementos alteram-se dinamicamente conforme o tipo de documento activo e consoante as entidades que os constituem. Desta forma, as funções existentes na plataforma podem estar disponíveis ou não, consoante os seus requisitos estejam satisfeitos ou não.

Para a interacção dos utilizadores com o sistema desenvolvido optou-se por utilizar a seguinte filosofia:

- As operações unárias são aplicadas no documento activo;
- As operações binárias recebem a indicação dos documentos a serem considerados pelo clique com o botão esquerdo do rato sobre uma qualquer janela de visualização dos mesmos;
- Para cada nova operação é criado um novo documento, com o seu título a reflectir a designação do(s) documento(s) original(ais) e a operação efectuada, ficando o(s)

documento(s) original(ais) intacto(s);

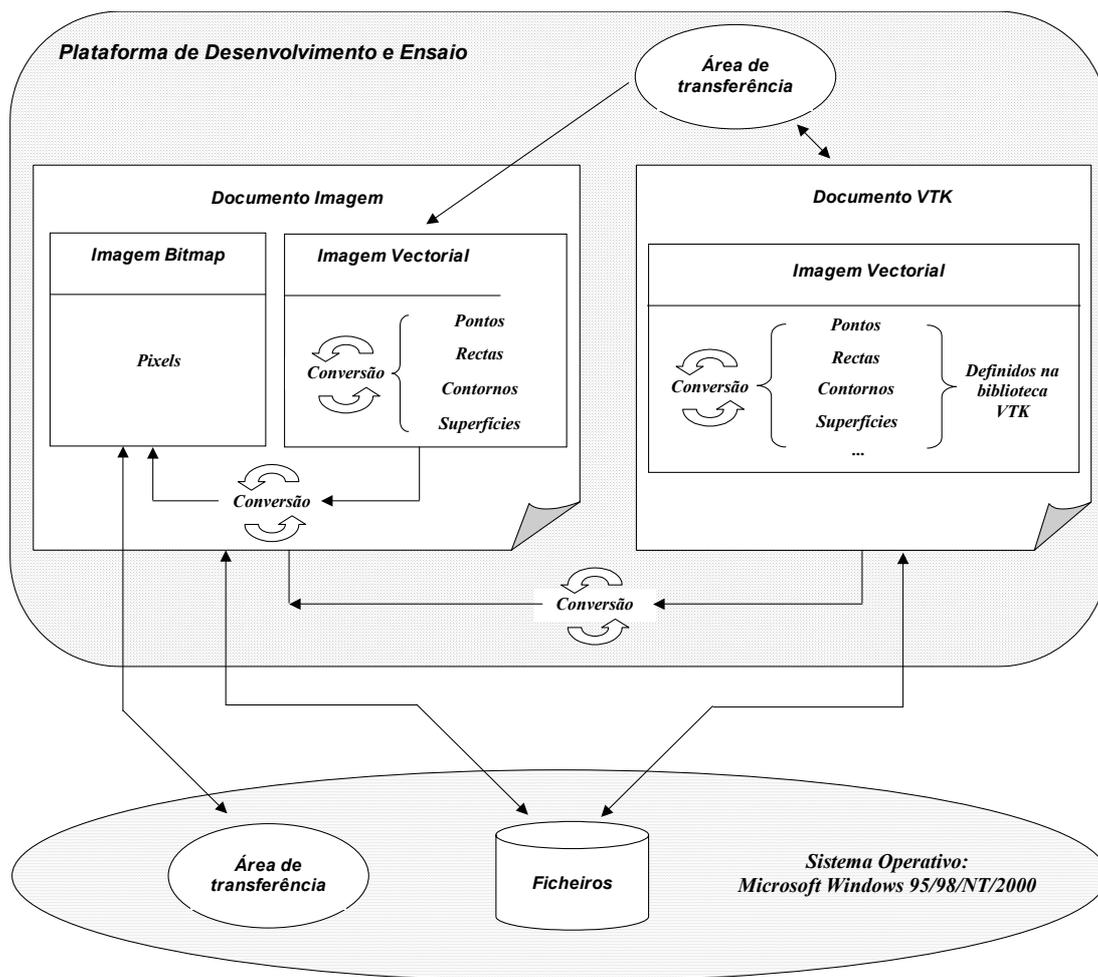


Figura 2 – Entidades suportadas na plataforma de desenvolvimento e ensaio.

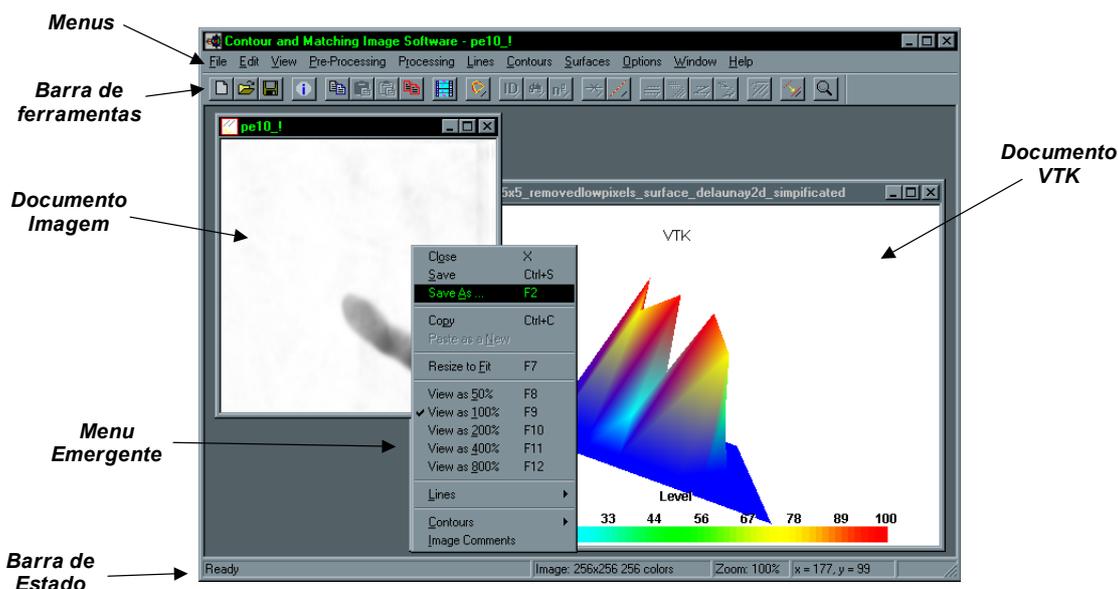


Figura 3 – A interface da plataforma de desenvolvimento e ensaio. (O documento activo é do tipo imagem; assim as funções disponíveis são as existentes para este tipo de documento.)

- No caso de as operações efectuadas alterarem o conteúdo do documento, é pedido ao utilizador para confirmar a realização da operação seleccionada;
- Na barra de estado surgem mensagens para o utilizador indicando a utilidade da operação seleccionada, a operação a efectuar, etc., e para documentos do tipo imagem é também indicado o nível de *zoom* actualmente utilizado, a dimensão e o número de cores da imagem activa;
- Quando se pressiona o botão direito do rato sobre um documento do tipo imagem surge o menu de contexto;
- Efectuando-se um duplo clique no botão esquerdo do rato sobre uma imagem *Bitmap* surge na secção direita da barra de estado o valor do respectivo *pixel*;
- Clicando-se no botão esquerdo do rato sobre uma imagem *Bitmap* e mantendo-o pressionado enquanto se arrasta o rato, define-se uma zona rectangular de selecção na imagem, com a possibilidade de visualizar-se as suas dimensões (*Figura 4*), a qual poderá ser posteriormente copiada e duplicada.

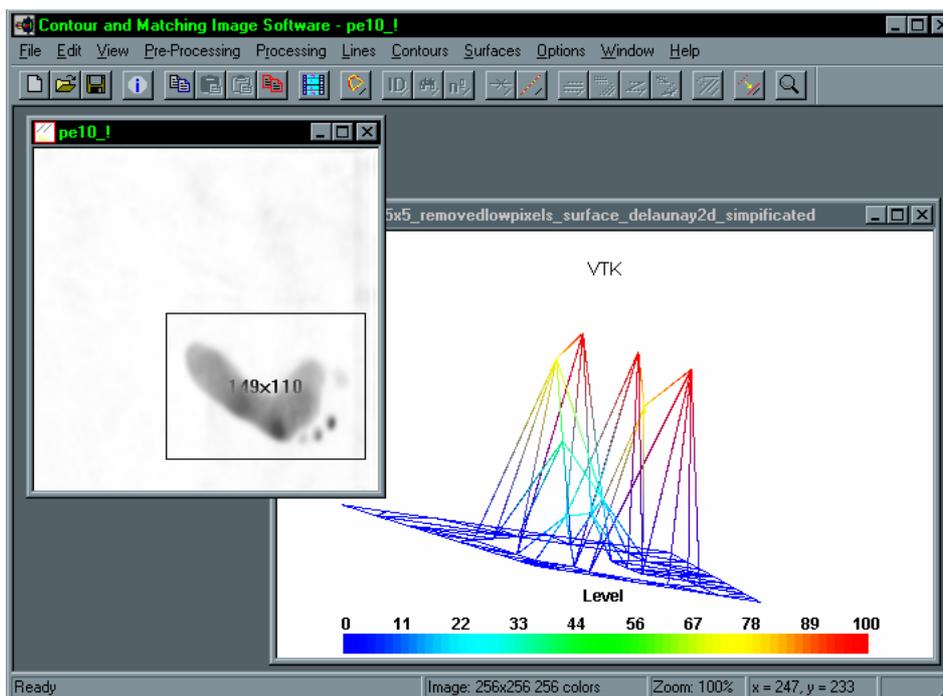


Figura 4 – Selecção de uma zona rectangular numa imagem Bitmap.

A interface da plataforma de desenvolvimento e ensaio muda dinamicamente quando passa a estar seleccionado um documento do tipo *VTK* (*Figura 5*). Como a classe responsável pela visualização destes tipos de documentos 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 em torno do ponto focal da câmara, deslocar a câmara ao longo do plano da vista actual, alterar o nível de *zoom* e

executar uma rotação plana da câmara em torno do seu eixo óptico. Já no modo de interacção com a cena do tipo actor², através dos mesmos botões, é possível aplicar ao objecto: uma rotação em torno da origem, uma deslocação plana, um escalamento uniforme, uma rotação plana e uma translação.

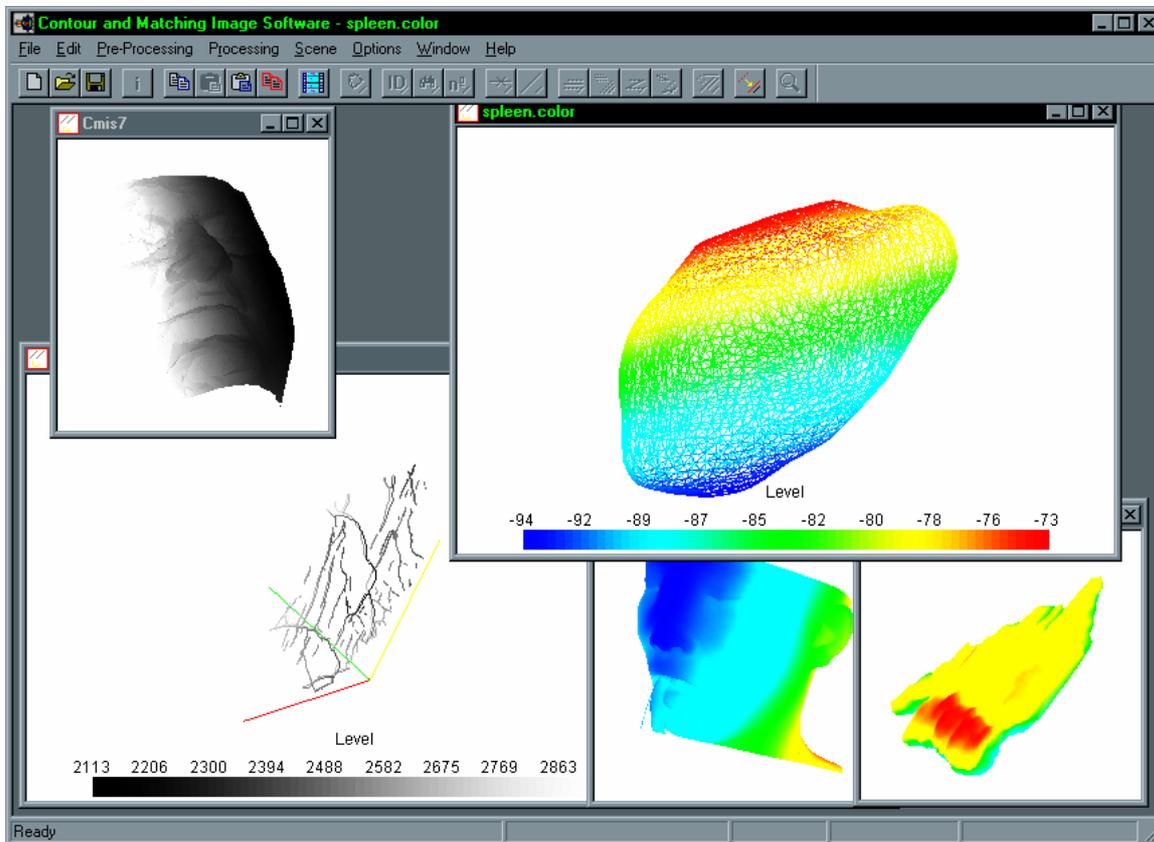


Figura 5 – Interface da plataforma de desenvolvimento e ensaio quando está activo um documento VTK.

Também devido à herança das classes *VTK* algumas teclas assumem determinadas funções para este tipo de documentos. Essencialmente estas teclas permitem alterar o tipo de interacção do utilizador com a cena e a representação dos objectos na mesma: comutar entre o modo do tipo actor (os eventos do rato alteram o actor actualmente seleccionado) ou do tipo câmara (os eventos do rato alteram a posição da câmara e o ponto focal), reinicializar a vista da câmara, representar todos os actores em modo de estrutura em arame ou em modo de superfície e representar o paralelepípedo que circunscreve o actor seleccionado.

Sempre que uma operação necessita de um tempo de execução significativo, são apresentados ao utilizador diálogos com a indicação do progresso do processamento (*Figura 6*).

Como já foi referido, as funções disponíveis na plataforma dependem dos módulos que estão configurados como disponíveis. Esta configuração, realizada em tempo de execução,

² Cada objecto *VTK* presente na cena de visualização é, por vezes, designado no âmbito do *VTK* por actor.

permite definir os menus disponíveis, as funções que constituem cada um destes menus, as barras de ferramentas disponíveis, os botões que as constituem e os itens que formam os menus emergentes, tornando assim o sistema adaptável a diferentes níveis de utilizadores e a diferentes tipos de aplicação.

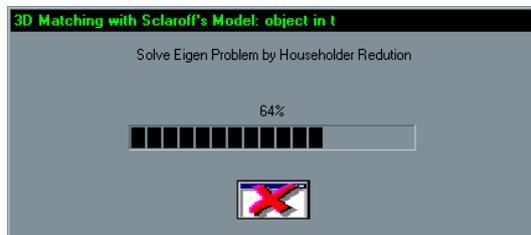


Figura 6 – Exemplo de um diálogo de progresso utilizado na plataforma.

5 – Integração de novas funções

Na *Figura 7* estão representadas as classes principais da plataforma *CMIS*; estas classes são responsáveis pelo controlo da aplicação, pela gestão da sua janela principal, pelos documentos imagem e *VTK* e pelas janelas de visualização de cada um destes documentos.

Com a filosofia utilizada na plataforma, 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, os passos a realizar para a integração de uma nova função são:

1. Acrescentar o(s) ficheiro(s) de implementação da nova função ao projecto do sistema no *Microsoft Visual C++*;
2. No *Microsoft Visual C++*, acrescentar um item para a função no menu apropriado;
3. Utilizando o *MFC ClassWizard* do *Microsoft Visual C++*, associar o item criado com uma das classes principais da plataforma³ e definir a função dessa classe a ser chamada quando o item for seleccionado;
4. Identificar quais as condições a serem verificadas para a função ser considerada como disponível;
5. Identificar qual o módulo da plataforma que conterá a função;

³ Caso a função esteja relacionada com os parâmetros globais da plataforma esta deve ser associada à *CMainFrame*; caso esteja associada com os documentos do tipo imagem e independente dos eventos associados ao rato deve ser associada à classe *CCmisDoc*; caso esteja associada a este tipo de documentos e dependente dos eventos referidos deve ser associada à classe *CCmisView*, sendo a chamada da função efectuada no interior da função associada ao evento pretendido; caso esteja associada com os documentos do tipo *VTK* e independente dos eventos associados ao rato deve ser associada à classe *CVtkDoc*; caso esteja associada com este tipo de documentos e dependente dos referidos eventos deve ser associada à classe *CVtkView*, sendo a chamada da função efectuada no interior da função associada ao evento pretendido.

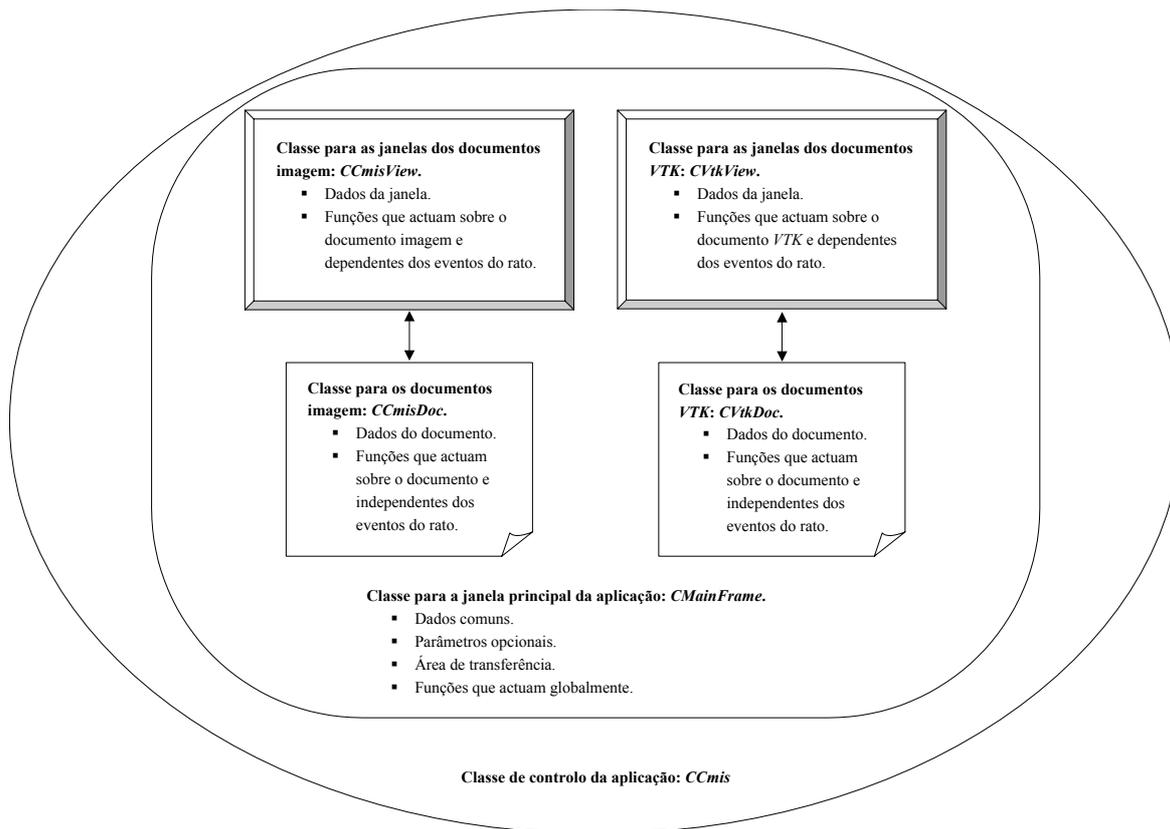


Figura 7 – Classes principais da plataforma de desenvolvimento e ensaio - CMIS.

6. Caso seja necessário, desenhar no *Microsoft Visual C++* uma caixa de diálogo para entrada dos parâmetros da função; utilizando o *MFC ClassWizard* do *Microsoft Visual C++*, criar a classe responsável pela interface desse diálogo com o utilizador; podem-se definir os parâmetros por omissão na classe *CMainFrame* e efectuar a sua inicialização na função *defaultValues()* da mesma; caso se pretenda que a plataforma guarde o valor dos parâmetros utilizados quando a execução terminar e os retome no início de nova execução, deve acrescentar-se o procedimento de escrita na função *InitInstance()* da classe *CCmisApp* e o de leitura na função *DestroyWindow()* da classe *CMainFrame*;
7. Caso seja necessário, utilizar um diálogo de progresso com a indicação da fase de processamento actual;
8. Caso seja útil, para facilitar a interface com o utilizador, enviar mensagens para a primeira zona (secção esquerda) da barra de estado, informando-o com os passos a seguir;
9. Recompilar a plataforma de desenvolvimento e ensaio no *Microsoft Visual C++*.

6 – Selecção de funções disponíveis

Nesta secção serão brevemente descritas algumas das funções actualmente disponíveis na plataforma de desenvolvimento e ensaio. Em primeiro lugar serão descritas as funções de índole mais geral e seguidamente, as funções mais específicas ao domínio dos 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 novos documentos, abrir documentos já existentes a partir dos seus ficheiros, fechar o documento activo e todos os documentos abertos, gravar o documento activo e todos os documentos abertos, alterar o título de um documento, escolher os parâmetros de escrita dos documentos em ficheiros (por exemplo, o factor de qualidade para ficheiros do tipo *JPEG*), adquirir uma imagem utilizando um periférico de digitalização (por exemplo, uma *webcam* ou um *scanner*), escolher e configurar a impressora, realizar a previsão da impressão do documento, imprimir o documento, abrir um documento presente na lista dos últimos documentos abertos e terminar a execução.

Para a edição de documentos imagem estão essencialmente disponibilizadas as funções associadas à cópia das entidades que um documento deste tipo pode conter. Assim, é possível: copiar uma imagem *Bitmap* ou apenas uma zona desta; repor a imagem *Bitmap* copiada no documento actual ou criar um novo documento com a mesma; duplicar o documento activo; copiar, e posteriormente inserir num outro documento, as entidades vectoriais consideradas neste tipo de documentos; realizar um novo documento com uma cópia da imagem *Bitmap* activa utilizando um diferente número de *bits* para a definição do valor dos *pixels* que a constituem, isto é, com diferente resolução de intensidade; realizar um novo documento com uma imagem de cinzentos de 8 *bits* resultante da conversão de uma dada imagem de cor.

As funções relacionadas com a visualização dos documentos imagem disponíveis na plataforma são as seguintes: activar ou desactivar a visualização das barras de ferramentas e de estado; alterar o factor de *zoom*; redesenhar as entidades presentes na janela activa; activar ou desactivar o desenho de duas rectas ortogonais que se cruzam no cursor do rato de forma a facilitar a identificação dos elementos presentes (*Figura 8*); visualizar um conjunto de documentos, seleccionados pelo utilizador, numa mesma janela com um intervalo de tempo de visualização entre cada um especificado e com total controlo sobre a exibição da sequência (*Figura 9*), sendo possível visualizar-se ao mesmo tempo mais do que uma série de imagens devido a utilizar-se na implementação *multithreads* com sincronização e envio de mensagens entre si [24] (esta função é útil para se visualizar uma série de imagens independentes como se tratasse de um filme); visualizar uma descrição da imagem activa (*Figura 10*), podendo o

utilizador editar texto que pode ficar guardado com a imagem no mesmo ficheiro⁴ e ser posteriormente recuperado quando a mesma for aberta na plataforma.

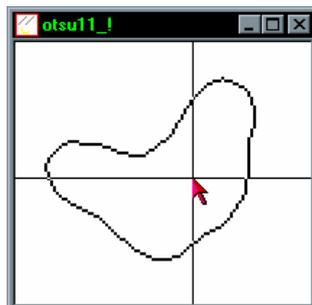


Figura 8 – Desenho de duas rectas ortogonais centradas no cursor do rato para auxílio de localização das entidades presentes.

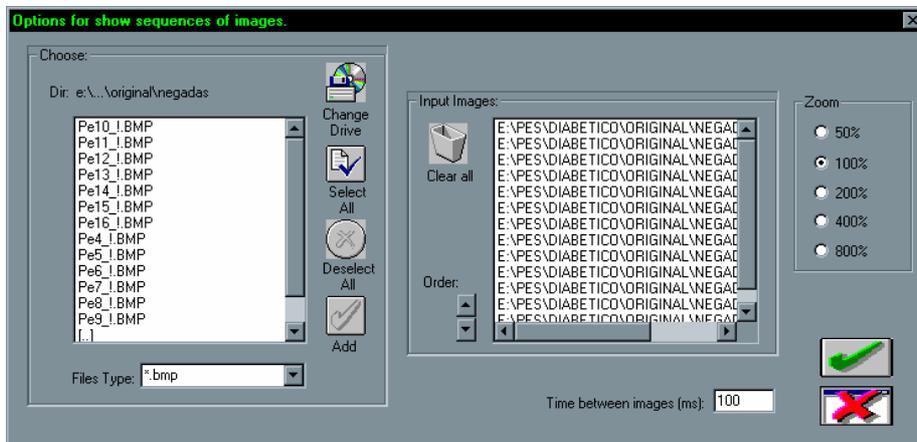
6.1.2 – Documentos VTK

As funções básicas associadas aos documentos *VTK* são também relacionadas com a entrada e saída de dados: criar um novo documento, abrir um documento gravado previamente em ficheiro, acrescentar um novo objecto *VTK* definido num determinado ficheiro no documento activo, criar um novo documento com dois objectos seleccionados pelo utilizador, importar objectos 3D definidos em ficheiros *ascii* (com o formato X, Y, Z; - coordenadas dos vértices do objecto por linha), fechar o documento activo e todos os documentos abertos, gravar o documento activo e todos os documentos abertos, definir o título do documento activo, seleccionar e configurar a impressora, realizar a previsão da impressão do documento actual, imprimir o documento, abrir um documento presente na lista de últimos documentos abertos e terminar a aplicação.

Para a edição deste tipo de documentos estão disponíveis as seguintes funções: copiar o conteúdo da janela de visualização activa para a área de transferência do sistema operativo; criar um documento do tipo imagem através de uma nova imagem *Bitmap* com o conteúdo existente na área de transferência; copiar um objecto *VTK* existente no documento activo; inserir no documento actual um objecto *VTK* copiado; criar um novo documento com um objecto copiado; duplicar o documento activo; alterar algumas propriedades de um objecto como a sua origem, a posição, a orientação, a escala e a opacidade; repor os valores por omissão das propriedades origem, posição, orientação e escala de um objecto; repor os valores por omissão destas propriedades em todos os objectos presentes num documento; apagar um dado actor; apagar todos os actores de um documento; determinar o número de objectos que constituem um documento; visualizar algumas características e a constituição de um objecto: coordenadas máximas e mínimas, comprimento da diagonal do paralelepípedo que o delimita, origem, orientação, posição, escala, coordenadas do centróide, número de pontos e de células, tipo de objecto, identificação de cada ponto e de cada célula, área da superfície, índice de

⁴ Esta informação *ASCII* é guardada no fim do ficheiro da imagem e é desprezada por outras aplicações durante a leitura da mesma.

forma do objecto relativamente a uma esfera e volume total e projectado segundo planos alinhados com cada um dos três eixos [14, 15] (Figura 11); visualizar as características de um objecto: número de pontos, coordenadas máximas e mínimas e do centróide, área da superfície, índice de forma e volume total e projectado (Figura 11).



a)



b)

Figura 9 – Visualização de uma série de documentos imagem: a) Selecção das imagens a serem visualizadas, especificação do factor de zoom e do valor do intervalo entre a visualização de cada uma; b) Visualização (Do lado direito, a caixa de diálogo de controlo da visualização e na qual é possível alterar-se o intervalo de tempo.).

6.1.3 – Configuração

De forma a adequar-se o funcionamento da plataforma desenvolvida a vários tipos de aplicação e a diferentes níveis de utilizador é possível configurar vários parâmetros da mesma. Destes podem-se destacar parâmetros para: os detectores de orlas de intensidade de *Deriche*, de *Chen & Castan*, de *Canny* e de *Spacek*; o seguimento de segmentos de recta ao longo de sequências de imagens; escolha das imagens resultantes dos operadores de gradiente de *Robert*, de *Prewitt*, de *Sobel* e de *Grad*; a configuração do funcionamento global da plataforma.

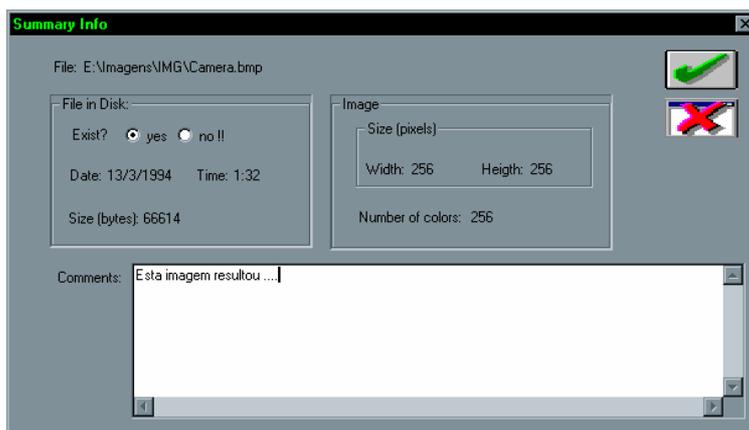


Figura 10 – Visualização de algumas características de uma determinada imagem.
(É possível adicionar e editar comentários associados à mesma.)

6.2 – Processamento de imagem

Como se pretende que a plataforma *CMIS* 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 em torno de um ponto especificado utilizando vários tipos de interpolação, o espelho e o verso, e a alteração das dimensões.

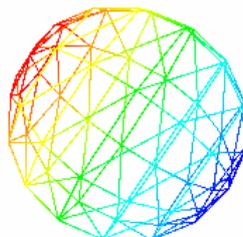
Para alterar o brilho dos *pixels* que constituem uma imagem foram incorporadas as seguintes funções: aumentar e diminuir o brilho de um determinado valor especificado; aumentar e diminuir o contraste; atribuir um determinado valor aos *pixels* que apresentem valor inferior a um dado limiar ficando os restantes com o valor original (útil para a remoção de *pixels* que possam ser considerados como ruído).

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.

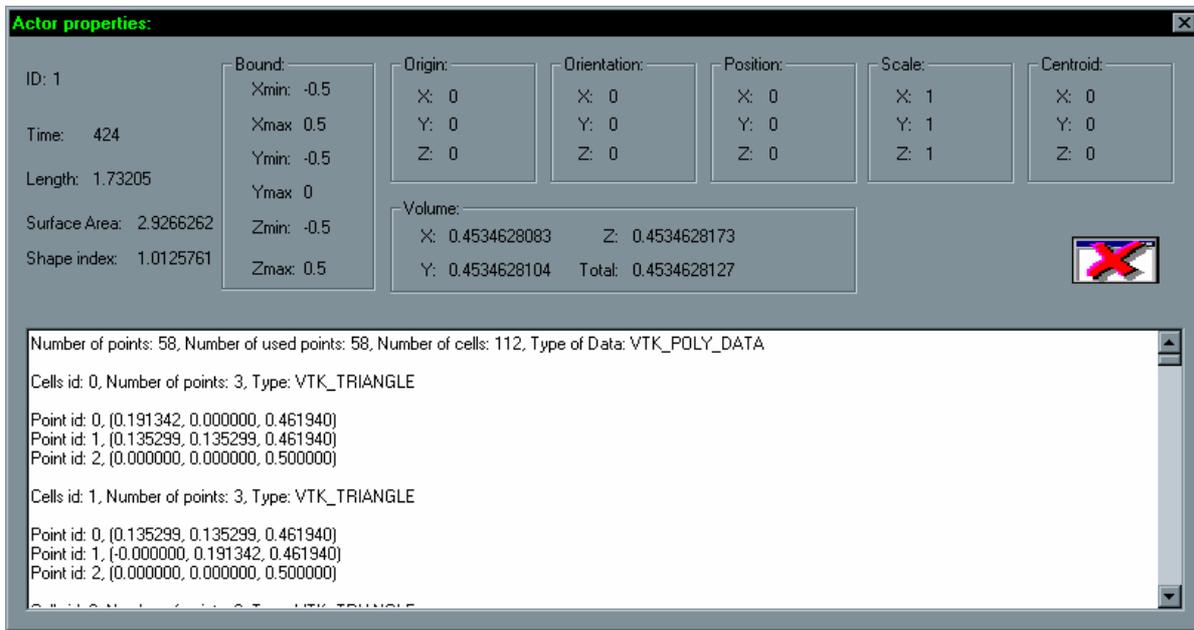
Vários tipos de filtros [25, 26] foram implementados na plataforma de desenvolvimento e ensaio; destes podem ser destacados os seguintes:

- de média, de mediana, de mínimo, de máximo, do mais próximo do máximo e do mínimo, sendo o tamanho do núcleo definido pelo utilizador;
- de gradiente⁵ [21] de *Roberts*, de *Prewitt*, de *Grad* (diferenças de 1ª ordem) e de *Sobel*;

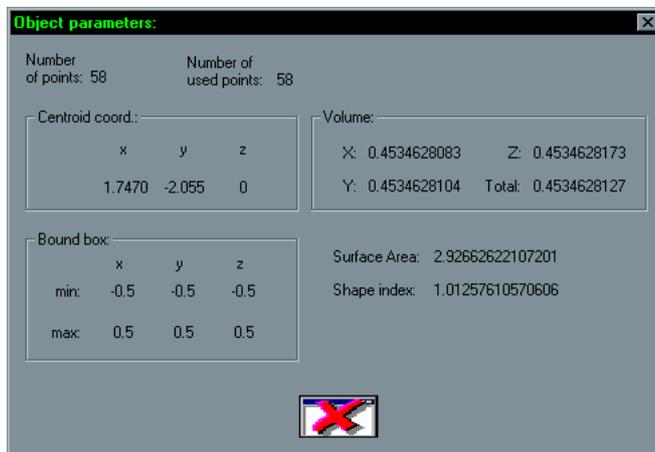
⁵ Por omissão são criadas cinco novas imagens: uma para o gradiente segundo a direcção *x*, outra segundo a direcção *y*, uma para a magnitude, uma quarta para a direcção, e uma última para a combinação da magnitude e da direcção; as imagens que resultam destes filtros podem ser seleccionadas nas opções de configuração da plataforma.



a)



b)



c)

Figura 11 – Visualização das propriedades de um objecto VTK: a) Objecto original; b) Alguns dos seus parâmetros e descrição da sua constituição; c) Alguns dos seus parâmetros.

- de Laplaciano, com núcleo definido pelo utilizador ou seleccionado a partir de núcleos *standard*, e de determinação das passagens por zero do Laplaciano;
- de convolução, com núcleo definido pelo utilizador (*Figura 12*);

- de Gaussiano e de *LoG*, seleccionados pelo utilizador, com ou sem normalização.

Também foram integradas no sistema as operações morfológicas de erosão, de dilatação, de fecho e de abertura, com núcleo definido pelo utilizador (*Figura 12*).

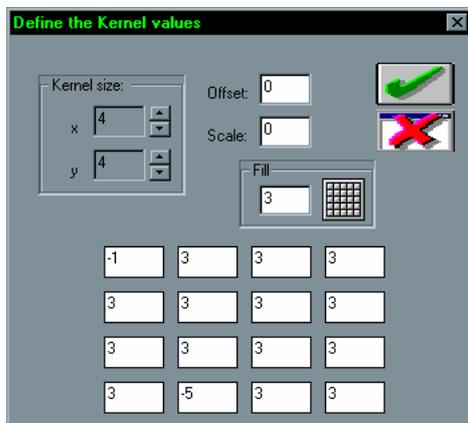


Figura 12 – Definição do núcleo de um filtro de convolução.

A detecção de orlas de intensidade utilizando os operadores [18, 21] 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 na plataforma desenvolvida, sendo a sua parametrização facilmente conseguida.

Como as operações lógicas e aritméticas são bastante úteis, nomeadamente para a comparação e verificação de resultados, foram integradas na plataforma as operações lógicas: negação, ou, e, e ou exclusivo, e as operações aritméticas entre duas imagens de adição, subtracção, multiplicação, divisão e máximo, e, entre uma imagem e um valor numérico especificado de adição, subtracção, multiplicação e divisão.

6.3 – Análise de imagem

Nesta subsecção são descritas as funções relacionadas com a análise de imagem integradas na plataforma de desenvolvimento e ensaio. Em primeiro lugar, são descritas as funções incorporadas 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 na plataforma são as seguintes: calcular o histograma de uma imagem (*Figura 13*), determinar a distância entre dois *pixels* definidos pelo utilizador através do rato (*Figura 14*), determinar o máximo e o mínimo valor dos *pixels* que constituem uma imagem, determinar o número de *pixels* que apresentam um determinado valor numa imagem.

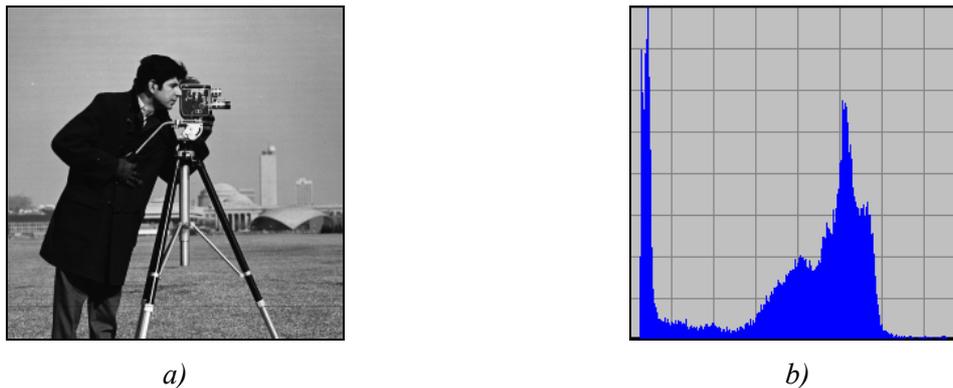


Figura 13 – Histograma de uma imagem: a) Imagem original; b) Histograma determinado.

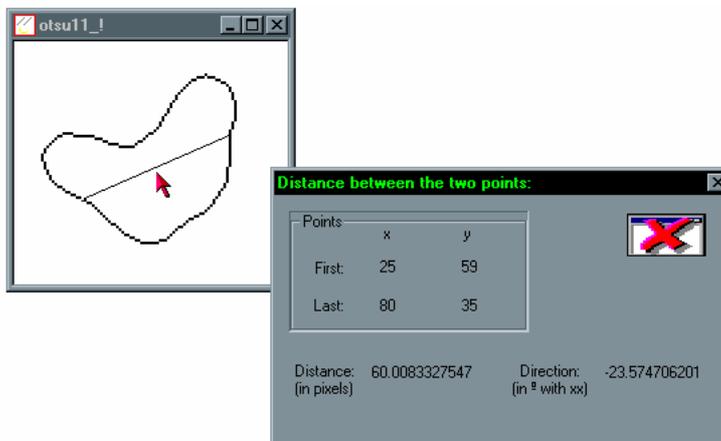


Figura 14 – Apresentação da distância, e da direcção, entre dois pixels especificados pelo utilizador.

Uma operação que tem muitas vezes utilidade num sistema de análise de imagem é a binarização. Assim foram integradas no sistema desenvolvido várias técnicas para se realizar a referida operação: determinação dos *pixels* com valor de brilho não inferior a um limiar; determinação dos *pixels* com valor de brilho dentro de um intervalo definido; determinação dos *pixels* com valor de brilho não inferior ao limiar determinado pelo método de *Otsu* [27] ou pelo método de *Kittler* [28]; determinação dos *pixels* com valor de brilho não inferior a um limiar definido interactivamente pelo utilizador (ao mesmo tempo é possível binarizar-se interactivamente mais do que uma imagem devido a utilizar-se na implementação *multithreads* com sincronização e envio de mensagens entre si) (Figura 15); binarização utilizando-se histerese de amplitude; binarização de objectos pelos métodos de *White & Rohrer's* [29].

Também foram integradas no sistema desenvolvido funções para processar os objectos constituídos por *pixels* de valor uniforme presentes numa imagem, nomeadamente para: esqueletizar, determinar as linhas centrais, determinar o rectângulo que circunscreve um objecto, realizar o preenchimento, determinar os *pixels* fronteira, anular os *pixels* interiores, amostrar utilizando uma grelha rectangular, determinar os pontos exteriores de um objecto previamente amostrado por uma grelha quadrada.

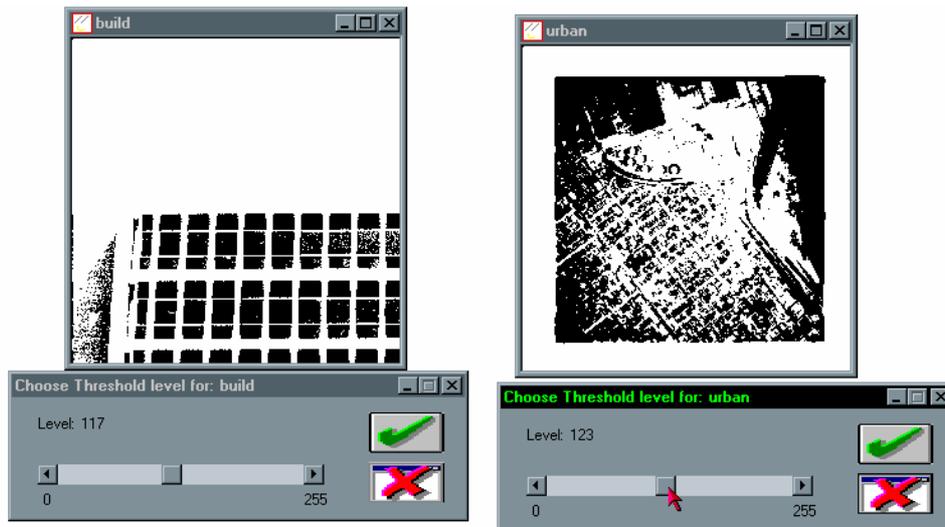


Figura 15 – Binarização interactiva de imagens Bitmap.

Em muitas aplicações da visão por computador assume uma importância relevante a análise das regiões presentes numa imagem. Assim resolveu-se incluir na plataforma desenvolvida uma operação [19], implementada utilizando-se *multithreads* com sincronização e envio de mensagens entre si, que permite: determinar as regiões presentes numa imagem, verificar algumas características de cada região determinada e criar uma nova imagem a partir de uma região seleccionada (Figura 16).

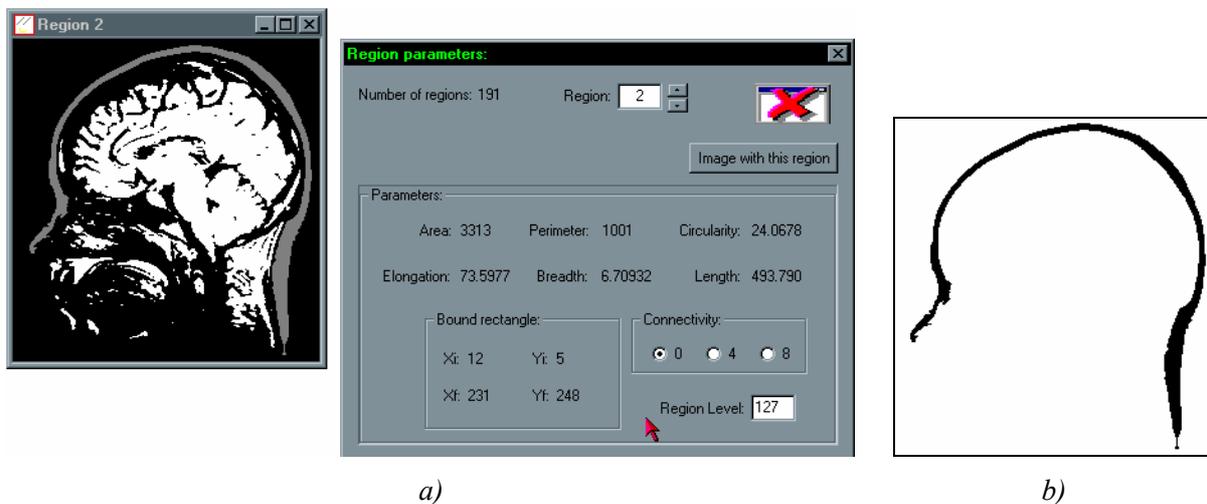


Figura 16 – Análise das regiões de uma imagem: a) Imagem original e alguns parâmetros da região seleccionada; b) Imagem (negada) criada a partir da região seleccionada.

Em visão por computador é muitas vezes necessário a detecção de pontos de calibração em imagens 2D que, após o emparelhamento com os correspondentes pontos 3D que lhes deram origem, são utilizados na calibração de câmaras ou projectores. Assim foi também portada para a plataforma uma função que determina, com resolução ao nível do *subpixel*, os vértices dos quadrados regulares que constituem uma imagem [20, 21], Figura 17. As orlas da imagem de entrada podem ser realçadas antes da fase de detecção dos vértices, e as

coordenadas dos pontos 2D determinados podem ser escritas num ficheiro de texto.

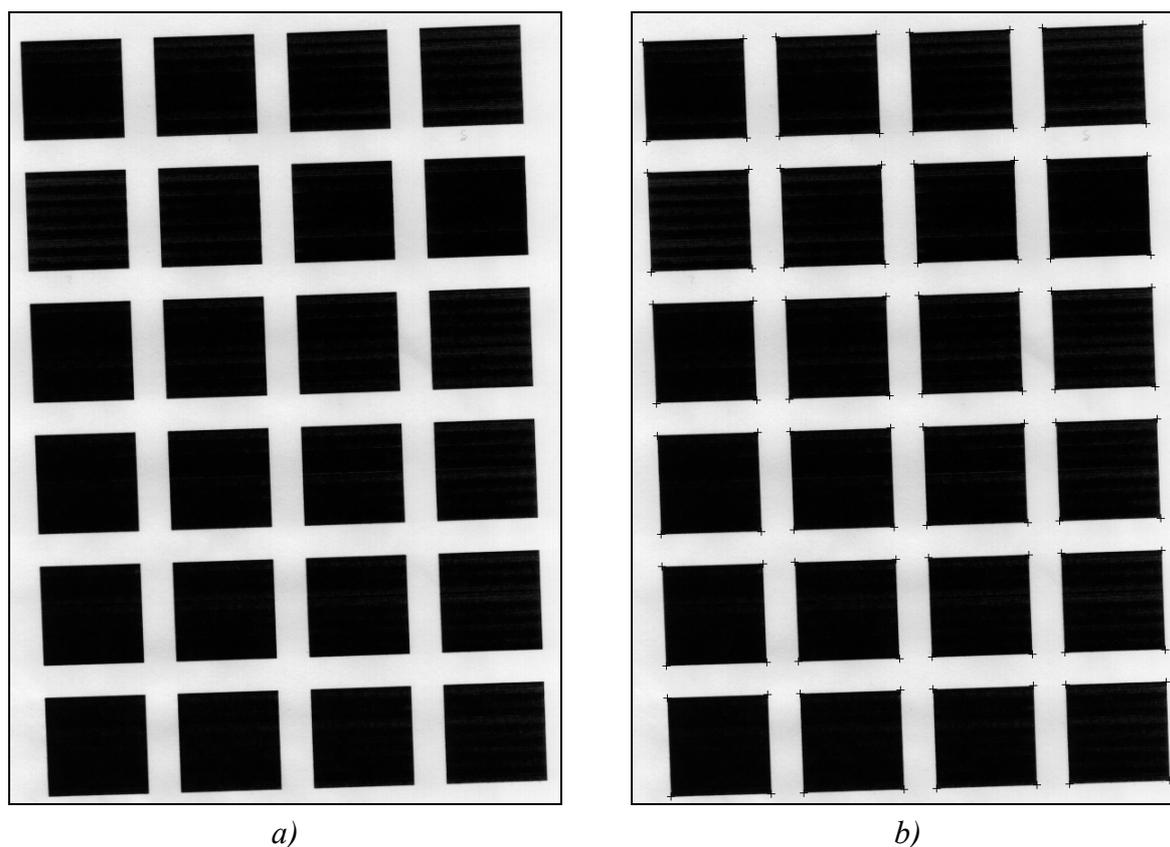


Figura 17 – Determinação dos vértices dos quadrados regulares presentes numa imagem: a) Imagem original; b) Imagem original com a indicação dos vértices determinados.

Na área da segmentação de objectos os contornos activos, vulgarmente designados por *snakes*, são bastante utilizados [2]. Com uma importância tão relevante, e com resultados satisfatórios em vários domínios de aplicação, incluíram-se na plataforma algoritmos de contornos activos baseados em regiões e baseados em orlas de intensidade (*Figura 18*).

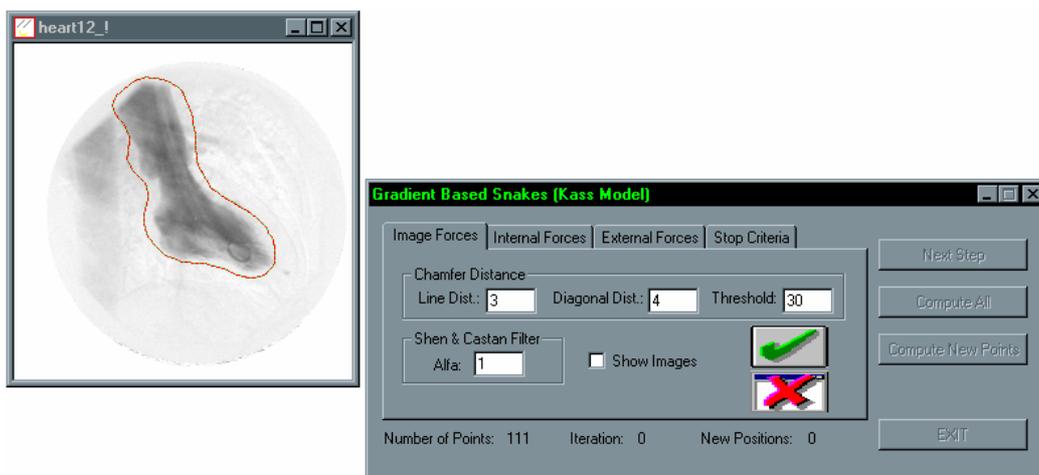
Os *pixels* resultantes da segmentação podem posteriormente ser convertidos em contornos ou em linhas tornando-se assim possível utilizar as ferramentas desenvolvidas e existentes no sistema para estas entidades (*Figura 19*).

Ao analisar a *Figura 18* é possível verificar a facilidade existente para visualizar e alterar os parâmetros do método envolvido, sendo assim facilitada a interacção do utilizador e permitindo um controlo correcto sobre todo o processo.

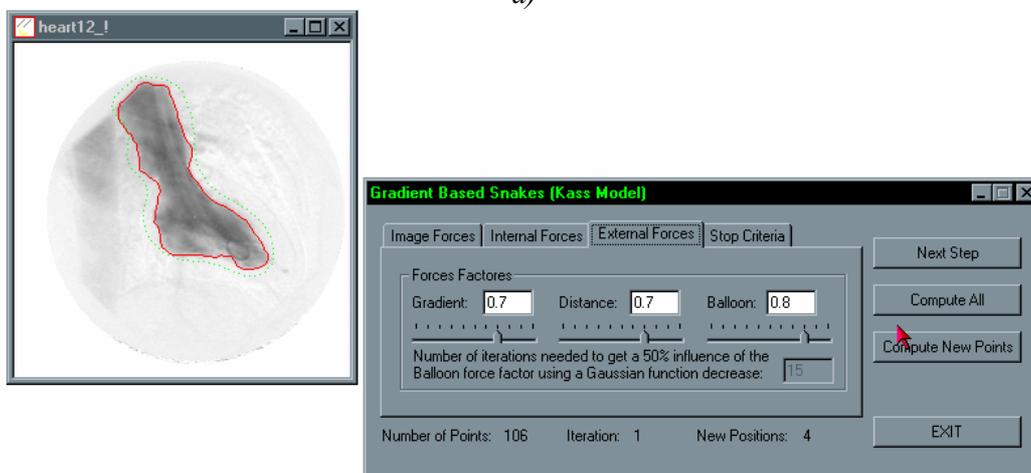
6.3.2 – Para linhas

Para entidades do tipo linha estão essencialmente disponíveis as ferramentas implementadas durante a dissertação de mestrado do autor deste relatório [20, 21]; nomeadamente para a detecção e o seguimento de linhas em imagens de orlas, a aproximação poligonal de linhas utilizando faixas dinâmicas, a simplificação de segmentos de recta [23], e o seguimento de segmentos de recta ao longo de sequências de imagens [22]. Seguidamente descreve-se, de

forma sucinta, as funções disponíveis para este tipo de entidade.



a)



b)

Figura 18 – 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.

Coordinates:		Number of pixels: 381
x	y	
First Pixel:	132 96	Area: 6416
Medium Pixel:	129.51 108.97	Perimeter: 444.374878

Figura 19 – Alguns parâmetros do contorno resultante da conversão da snake final da Figura 18.

Em qualquer instante o utilizador pode verificar os parâmetros de um determinado segmento de recta, determinar o número de segmentos de recta que constitui um documento, determinar um segmento de recta presente num documento a partir das coordenadas dos seus pontos extremos ou a partir da sua identificação, apagar ou inserir um segmento de recta, determinar os segmentos de recta que constituem um documento e têm comprimento superior

a um determinado número de *pixels*, converter os segmentos de recta especificados em contornos.

Para detectar, seguir e aproximar as linhas presentes em imagens de orlas, foi implementado um algoritmo que utiliza no seguimento histerese de amplitude e critérios de direcção e, na aproximação poligonal, um algoritmo baseado em faixas dinâmicas. As imagens de entrada a considerar deverão ser resultantes de um qualquer detector de orlas de intensidade aplicado na imagem original; este detector deverá criar uma imagem para as amplitudes e uma outra para as direcções.

Na *Figura 20* é possível verificar-se os segmentos de recta resultantes da detecção, seguimento e aproximação poligonal das linhas existentes numa imagem original.

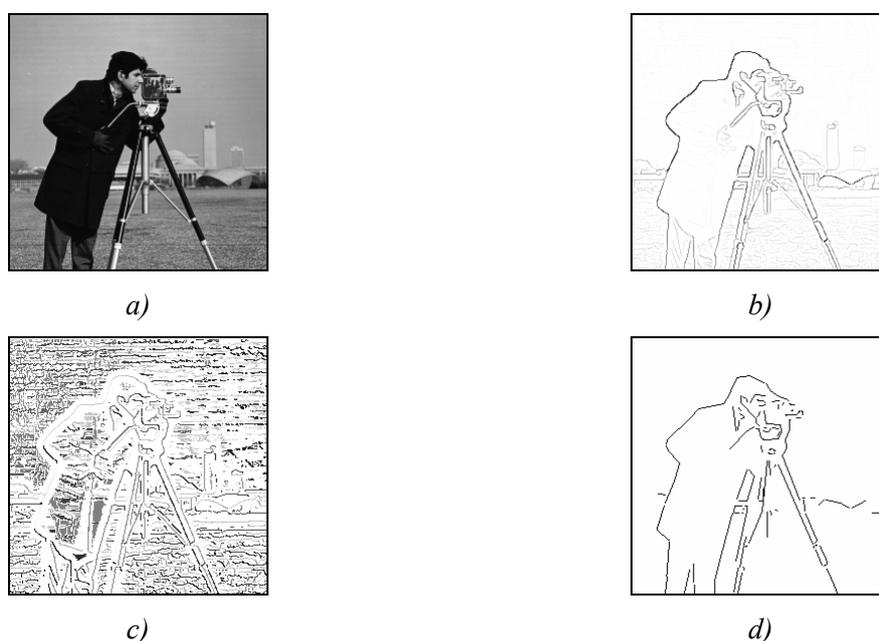


Figura 20 – Detecção, seguimento e aproximação poligonal de linhas: a) Imagem original; b) Imagem (negada) de amplitudes resultante do detector de Deriche; c) Imagem (negada) de direcções resultante do detector de Deriche; d) Aproximação resultante, utilizando-se os parâmetros definidos por omissão.

No sistema desenvolvido dois segmentos de recta podem ser simplificados por dois processos: um, utilizando o algoritmo que considera, na determinação do segmento de recta resultante, o comprimento e a posição de cada um dos segmentos a fundir; e um outro que determina o segmento de recta resultante através da ligação dos pontos mais extremos dos segmentos de recta a simplificar. Cada um destes dois tipos de simplificação pode também ser aplicado de uma só vez a todos os segmentos de recta presentes num documento.

Na *Figura 21* está representado um exemplo de fusão de dois segmentos de recta e na *Figura 22* um exemplo idêntico para a simplificação.

O seguimento de segmentos de recta ao longo de uma sequência é possível utilizando uma implementação da metodologia baseada em filtragem de *Kalman*, distância de *Mahalanobis* e

em restrições geométricas. Com a implementação utilizada, é possível especificar todos os parâmetros para o controlo da metodologia (*Figura 23*) e verificar, durante a execução, todo o processo de seguimento (*Figura 24*), o que com a implementação anteriormente realizada não era conseguido de forma tão fácil e cómoda [20, 21].

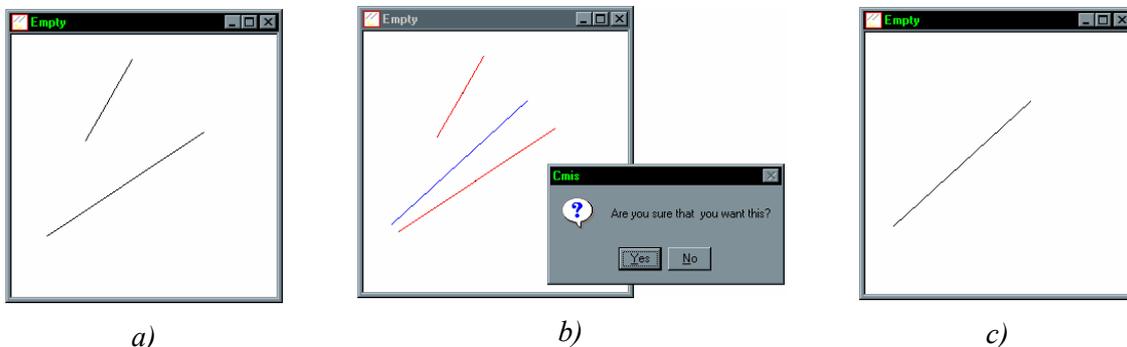


Figura 21 – Fusão de dois segmentos: a) Segmentos originais; b) Confirmação da fusão; c) Segmento resultante.

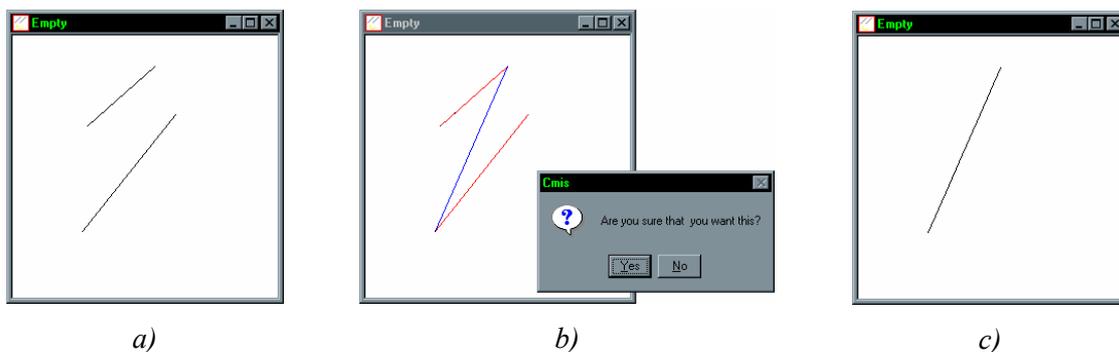


Figura 22 – Simplificação de dois segmentos: a) Segmentos originais; b) Confirmação da simplificação; c) Segmento resultante.

6.4 – Computação e visualização gráfica

Uma das características interessantes da plataforma 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.

De seguida, são referidas algumas das funções disponíveis para objectos gráficos:

- eliminação de pontos duplicados e primitivas degeneradas;
- simplificação de objectos, por meio de uma técnica de decimação [14, 15, 30] (*Figura 25*) ou por amostragem; também, em sentido oposto, foi incluída uma função para a

subdivisão das células de um objecto;

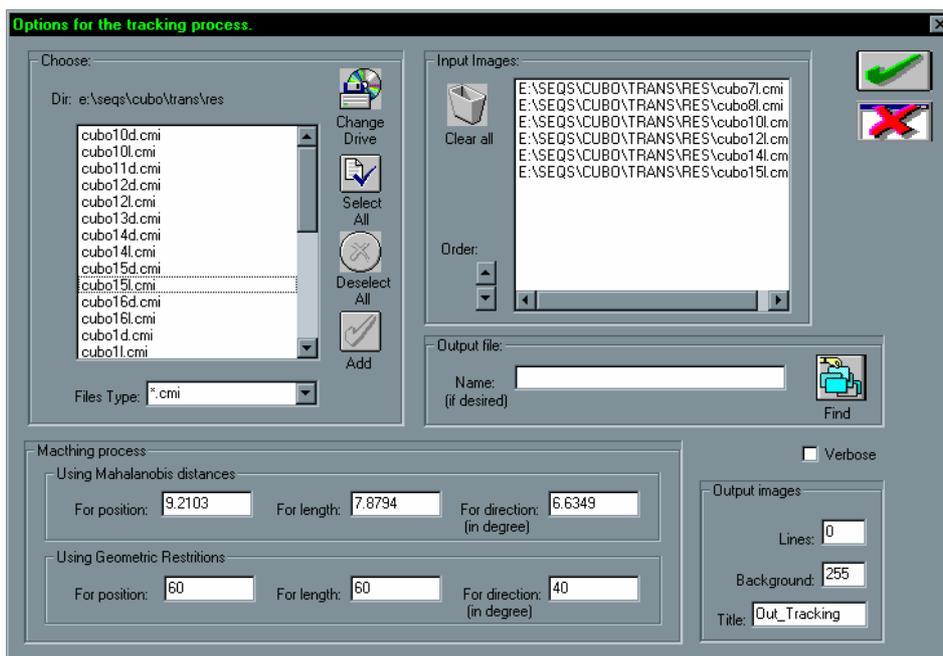


Figura 23 – Especificação dos ficheiros, e dos parâmetros a considerar, no processo de seguimento dos segmentos de recta que os constituem.

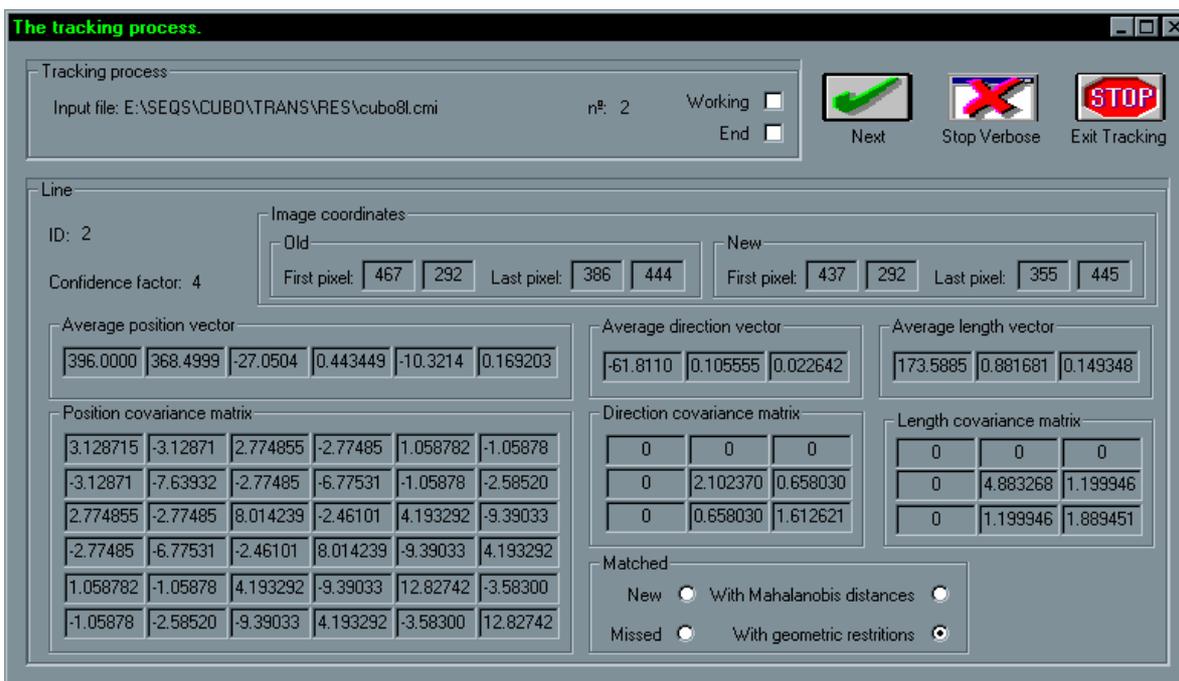
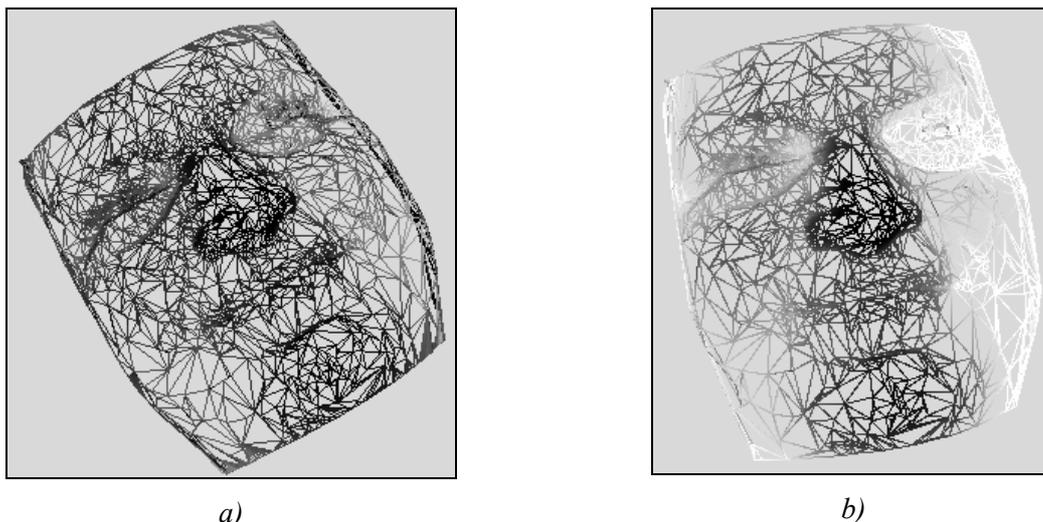


Figura 24 – Descrição do processo de seguimento de segmentos de recta ao longo de uma sequência.

- corte de um objecto gráfico por um plano que passa por uma dado ponto e cuja normal tem direcção especificada;
- diminuição do ruído presente na estrutura de um objecto gráfico por suavização Laplaciana;



*Figura 25 – Simplificação de um objecto: a) Objecto original (2943 pontos);
b) Objecto simplificado (2075) pontos.*

- conversão entre diferentes tipos de estrutura de objectos gráficos;
- visualização das normais nos vértices de um objecto gráfico;
- deformação geométrica de um objecto gráfico utilizando o escalar associado a cada um dos seus pontos: cada ponto é deslocado ao longo da normal que por ele passa, sendo este deslocamento definido por um valor, arbitrado pelo utilizador, multiplicado pelo escalar associado ao ponto (*Figura 26*);

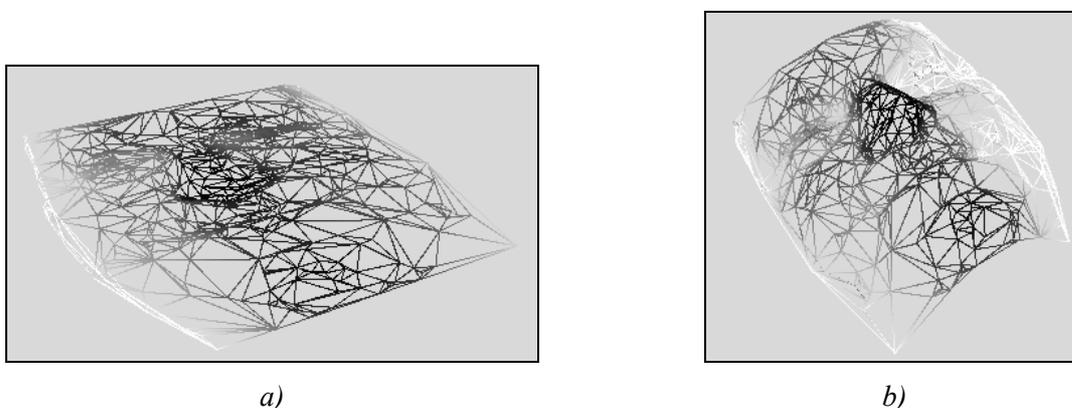


Figura 26 – Deformação da geometria de um objecto utilizando os escalares: a) Objecto original com valores escalares associados aos seus pontos; b) Objecto resultante.

- projecção de um objecto gráfico 3D num dos três planos coordenados;
- alteração das dimensões de um objecto gráfico por normalização das coordenadas dos seus pontos;
- encolhimento, num determinado valor, das células que constituem um objecto gráfico (*Figura 27*);



Figura 27 – Encolhimento das células de um objecto: a) Objecto original; b) Objecto resultante.

- determinação do centro das células de um objecto gráfico;
- binarização de um objecto gráfico, quer ao nível dos seus pontos quer ao nível das seus células, por valor mínimo, por valor máximo ou por uma faixa de valores (Figura 28);

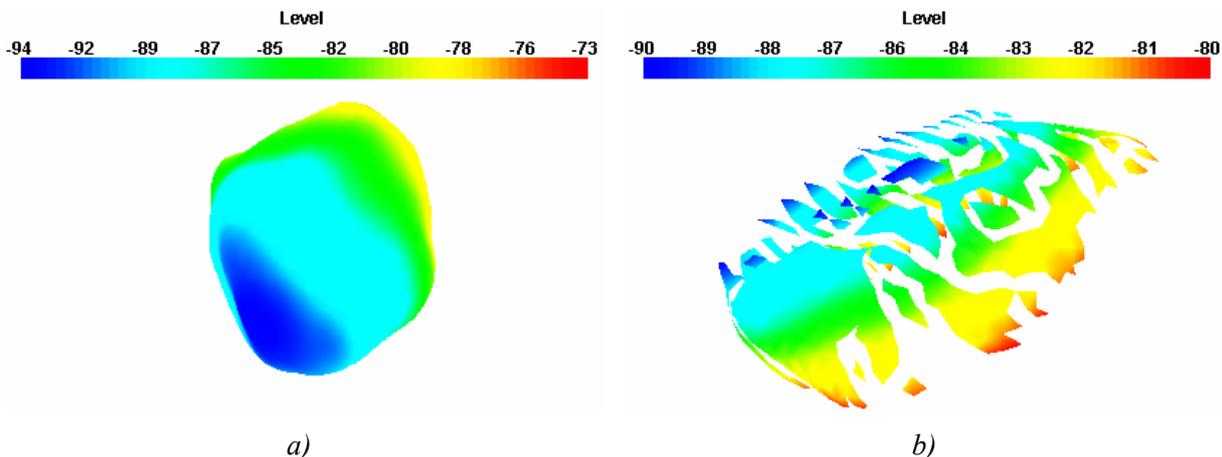


Figura 28 – Binarização das células de um objecto: a) Objecto original; b) Objecto resultante da binarização das células com valores entre -90.0 e -80.0 .

- realização das operações lógicas de união, diferença, intersecção e união de magnitudes (Figura 29) entre dois objectos gráficos;
- visualização da transformação, de forma faseada, de um objecto gráfico num outro (esta transformação, geralmente designada por *morphing*, foi implementada permitindo que os objectos intermédios sejam visualizados numa mesma janela e desfasados entre si de um intervalo de tempo especificado pelo utilizador, podendo o número de etapas utilizadas na transformação ser também definido);
- amostragem de um objecto gráfico nos pontos que constituem um outro objecto;
- aplicação a um objecto gráfico de uma transformação 3D rígida constituída por uma rotação em torno de um determinado ponto, um escalamento em relação a um dado

ponto e uma translação (Figura 30);

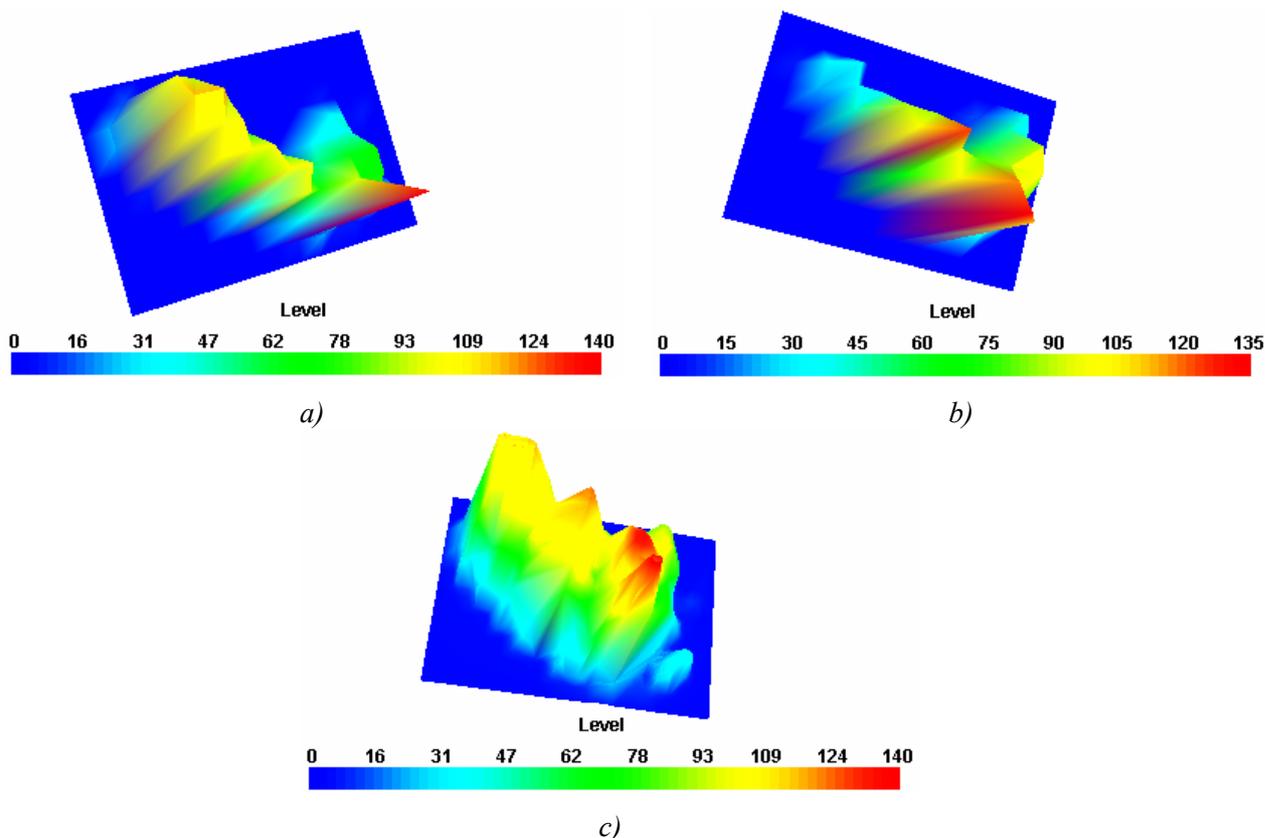


Figura 29 – União de dois objectos: a) Objecto original 1; b) Objecto original 2; c) Objecto resultante.

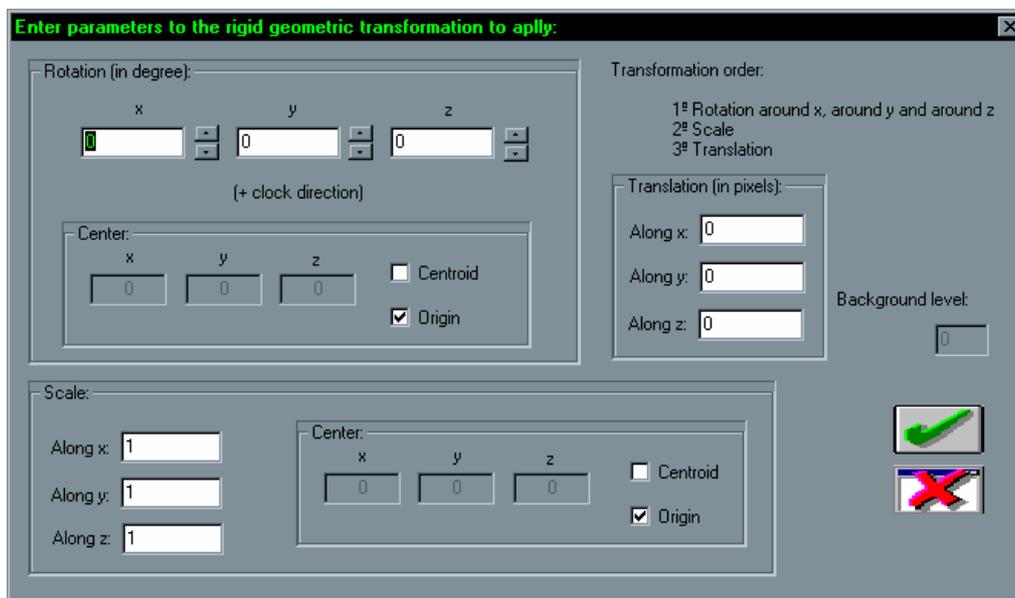


Figura 30 – Especificação dos parâmetros da transformação rígida a aplicar.

- extracção do contorno exterior, de contornos de isonível e de arestas específicas, de um objecto gráfico (Figura 31 e Figura 32);

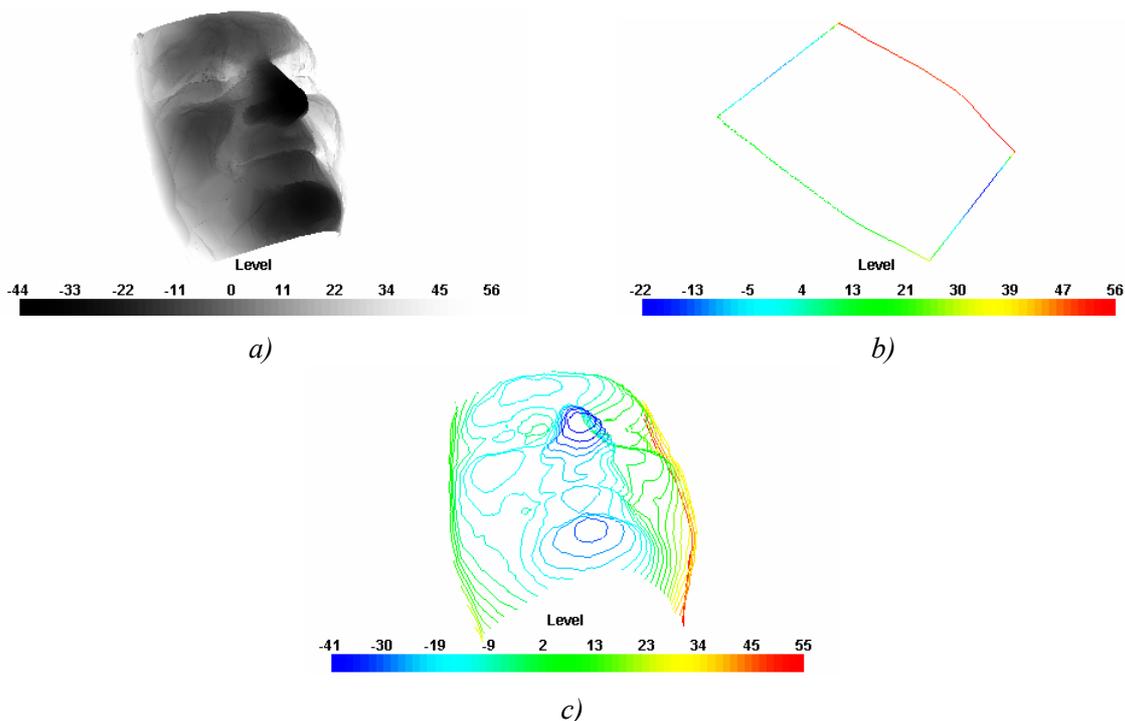


Figura 31 – Determinação dos contornos de um objecto: a) Objecto original; b) Contorno exterior; c) Alguns contornos de isonível.



Figura 32 – Determinação das orlas de um objecto: a) Objecto original; b) Arestas determinadas.

- triangulação 2D e 3D de Delaunay e reconstrução de superfícies de sólidos tridimensionais [Schroeder, 1998, 1999];
- extrusão rotacional (Figura 33);

Na plataforma desenvolvida é possível ao utilizador controlar uma série de parâmetros de visualização e de representação dos objectos *VTK*, nas respectivas janelas de visualização. As funções disponíveis para tal controlo são as seguintes:

- Definir o texto, e a respectiva formatação, a inserir na janela de visualização do documento (Figura 34).
- Activar ou desactivar a identificação dos pontos visíveis dos objectos (Figura 35).
- Activar ou desactivar a identificação das células visíveis dos objectos.

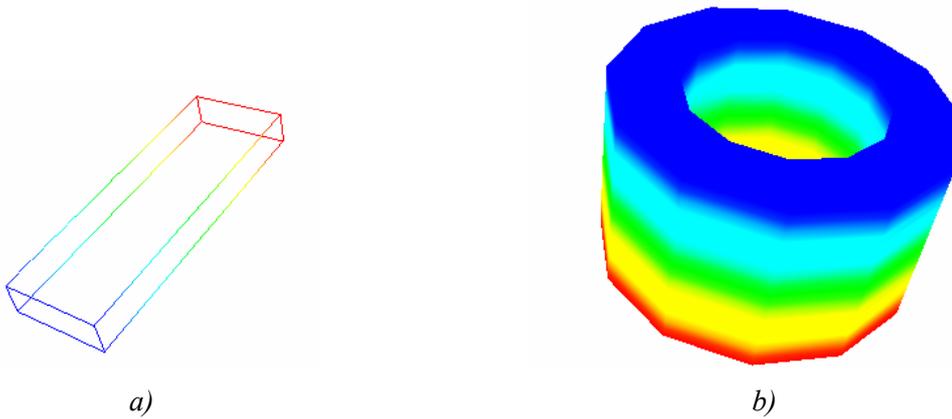


Figura 33 – Extrusão rotacional de um objecto: a) Objecto a extrudir; b) Objecto resultante da extrusão de 360° com translação nula e raio nulo.

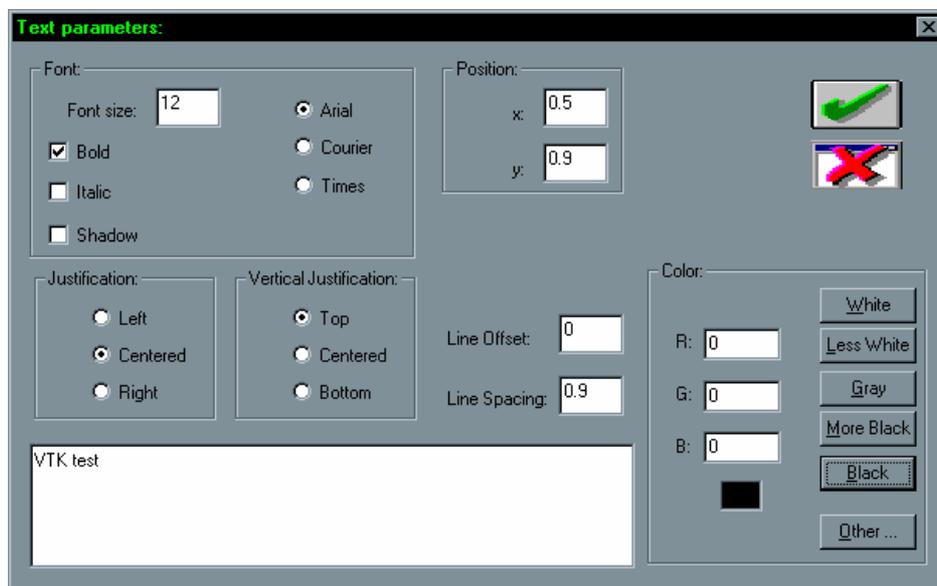


Figura 34 – Especificação e formatação do texto a inserir numa cena.

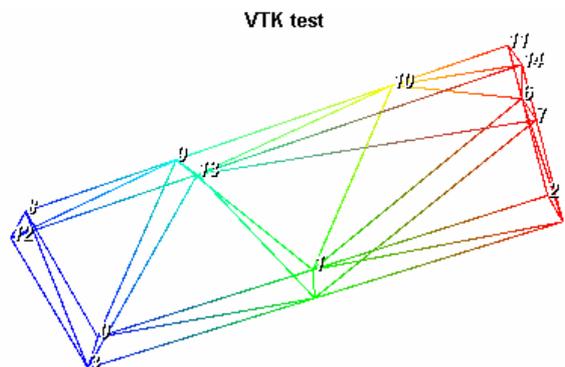


Figura 35 – Identificação dos pontos de um objecto.

- Activar ou desactivar a representação dos eixos coordenados; é possível ao utilizador especificar a origem e a escala.
- Escolher do tipo de vista pretendida para a cena: de cima, de baixo, de frente, de trás,

da esquerda, da direita ou isométrica.

- Activar ou desactivar a barra de cores para os valores escalares associados aos pontos que constituem os objectos representados; é possível especificar a sua posição na janela, o título e sua formatação, o número de etiquetas e a formatação numérica das mesmas (Figura 36).

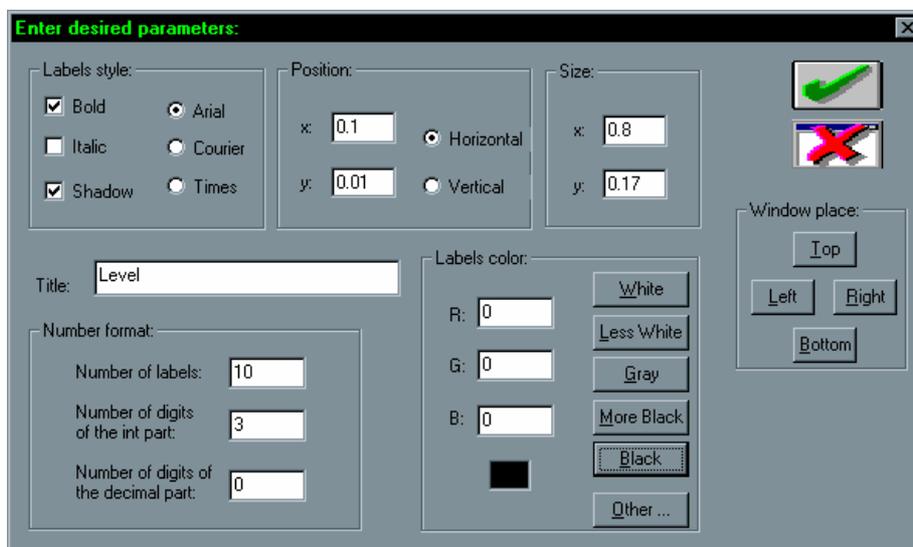


Figura 36 – Formatação da barra de cores associada aos escalares dos objectos.

- Seleccionar do tipo de tabela de cores utilizada na representação dos objectos: a cores, a preto e branco, de saturação ou de tonalidade; é possível o utilizador definir o número de níveis a utilizar.
- Seleccionar a cor do fundo da janela activa.
- Seleccionar a representação a utilizar para os objectos: por pontos, por estruturas em arame, por superfícies.
- Definir o tipo de interpolação a utilizar para os objectos [30]: *Flat*, *Gouraud* ou *Phong*.
- Definir o tamanho da janela de visualização activa; as janelas de visualização dos documentos podem também ser redimensionadas com a utilização do rato.

6.5 – Específicas aos objectos deformáveis

Nesta subsecção serão descritas sucintamente as funções disponíveis na plataforma consideradas como mais específicas ao domínio dos 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

6.5.1.1 – 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 [20, 21], e da determinação da curvatura e do *chain-code* [26]. A respectiva função deve ser utilizada sobre uma imagem de amplitudes resultante de um qualquer detector de orlas de intensidade aplicado na imagem original.

Na *Figura 37* está representado um exemplo de detecção dos contornos presentes numa imagem original.

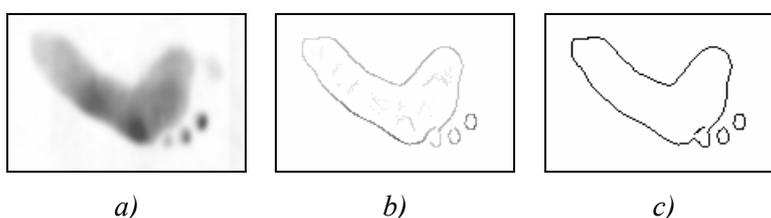


Figura 37 – Exemplo de detecção de contornos: a) Imagem original; b) Imagem de amplitudes resultante de um detector de orlas de intensidade aplicado na imagem original; c) Contornos determinados.

A qualquer momento o utilizador pode inserir um novo contorno, aberto ou fechado, utilizando o rato para identificar os *pixels* que o constituem. No caso de contorno fechado os *pixels* são automaticamente ligados entre si.

Também é possível ao utilizador apagar um contorno especificado, definir os contornos existentes como pertencentes a um único contorno fechado ou aberto, determinar o número de contornos presentes, verificar algumas características de um contorno especificado (*Figura 38*) e identificar um *pixel* pertencente a um contorno; foi igualmente implementada a transformação rígida 2D de contornos, com especificação de rotação, escala e translação.

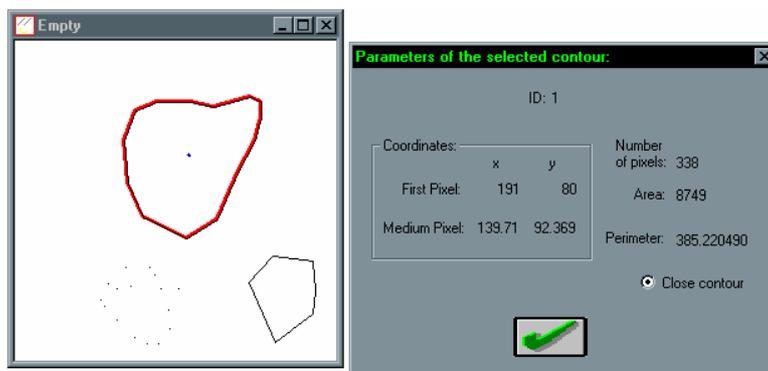


Figura 38 – Verificação de algumas características de um contorno.

Foram implementadas funções para a determinação do *chain-code* e do valor da curvatura dos *pixels* que constituem um dado contorno, operações bastante utilizadas no domínio dos

objectos deformáveis.

A determinação da cobertura convexa (*Convex Hull*) [14, 30] de um conjunto de pontos é também uma operação bastante utilizada em visão por computador, nomeadamente no domínio dos objectos deformáveis; assim esta operação foi implementada e pode ser utilizada, por exemplo, para determinar o contorno exterior de um conjunto de pontos não estruturados.

Na *Figura 39* está representado um exemplo da determinação do *Convex Hull* de um conjunto original de pontos.

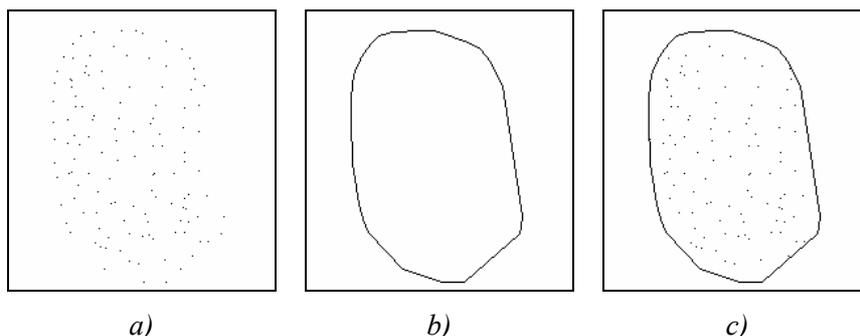


Figura 39 – Determinação do Convex Hull de um conjunto de pontos: a) Conjunto original; b) Convex Hull determinado; c) Representação sobreposta do Convex Hull determinado e do conjunto original de pontos.

Na plataforma 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 [2, 31-34], ou a modelização física do contorno por intermédio de um elemento finito isoparamétrico 2D de *Sclaroff* [2, 35-38] ou por intermédio de um conjunto de elementos finitos axiais lineares agrupados [2, 39]. 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 considerando os deslocamentos associados a este conjunto e criar uma imagem para cada modo de vibração considerando os deslocamentos associados.

Na *Figura 40* está visível o diálogo disponível ao utilizador para este verificar os resultados obtidos e construir, segundo os seus próprios critérios, as imagens de resultados.

A determinação das correspondências entre dois contornos utilizando o método de mapeamento segundo a distância mínima, [2], foi implementada na plataforma desenvolvida. Além das correspondências obtidas é possível verificar a estimação para a transformação rígida existente entre os dois contornos, obtida utilizando uma implementação do método de *quaternions* unitários [2, 40, 41].

Na *Figura 41* pode-se analisar o diálogo apresentado ao utilizador para este verificar os resultados obtidos, criar as imagens de resultados pretendidas e, caso deseje, alterar as condições de emparelhamento utilizadas.

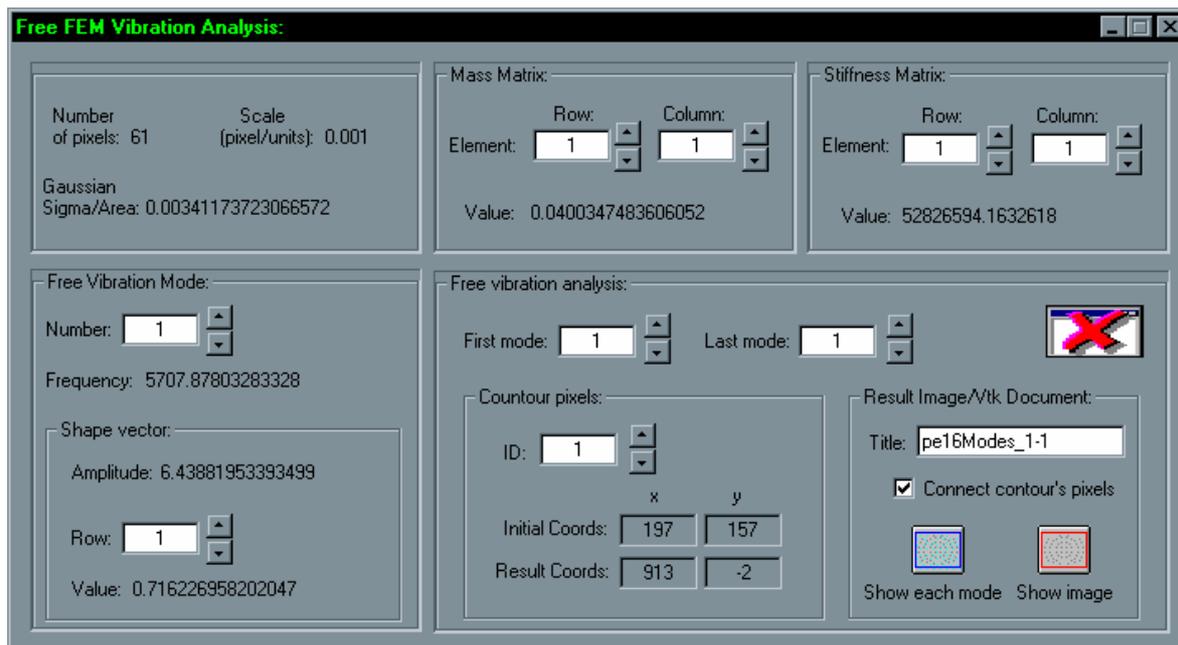


Figura 40 – Diálogo disponível ao utilizador para este verificar a análise da vibração de um contorno em modo livre.

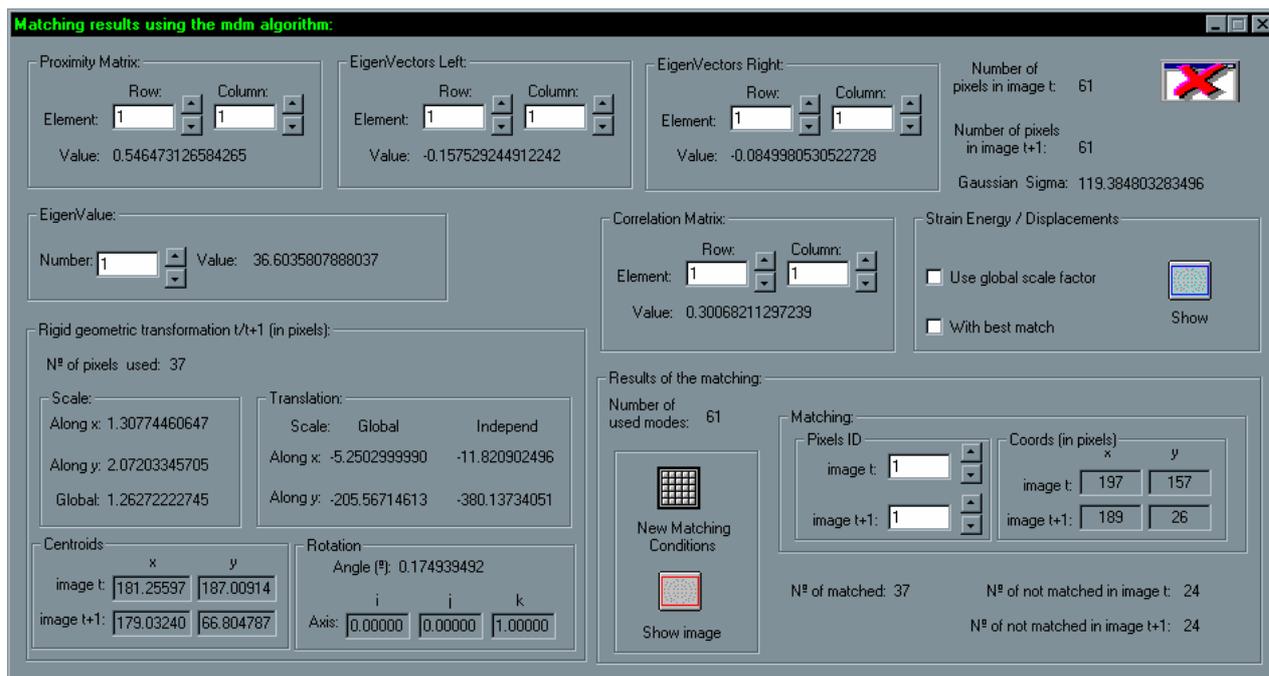


Figura 41 – Resultados da determinação das correspondências entre dois contornos utilizando o método do mapeamento segundo a distância mínima.

Também foi implementada uma função para determinar as correspondências entre dois contornos utilizando o método baseado na análise modal da forma [2, 31-34], ou utilizando o método baseado em modelização física, por intermédio de dois elementos finitos isoparamétricos 2D de *Sclaroff* [2, 35-38] ou por intermédio de dois conjuntos de elementos finitos axiais lineares agrupados, com posterior emparelhamento modal [2, 39]. 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 (caso seja utilizado o método da análise modal da forma, é criado um modelo físico para cada contorno sendo os parâmetros e o tipo de modelização pretendidos especificados pelo utilizador), 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.

Na *Figura 42* é possível visualizar-se o diálogo para o utilizador especificar a metodologia, e as respectivas condições pretendidas para determinar o emparelhamento entre dois contornos. Analisando-se a referida figura, é possível verificar os vários parâmetros a utilizar e a facilidade com que os mesmos podem ser especificados pelo utilizador.

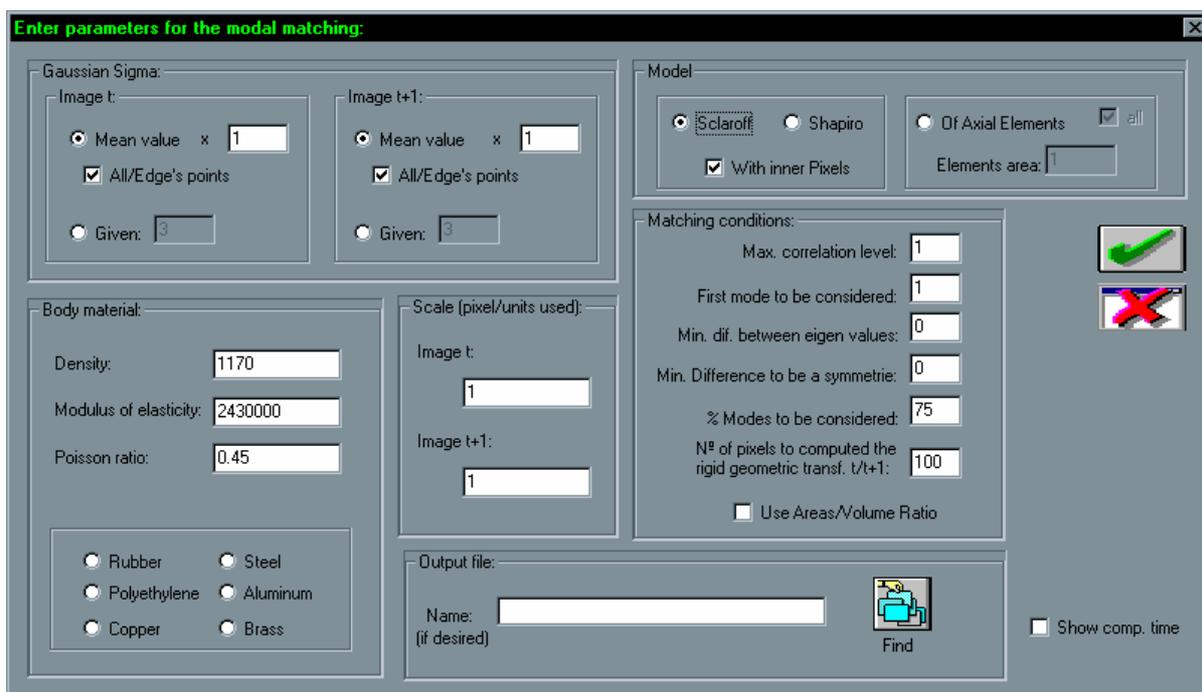


Figura 42 – Escolha do tipo de modelização física e dos seus parâmetros pretendidos e das condições desejadas para o emparelhamento modal.

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 estimação dos deslocamentos nodais e cálculo da energia de deformação, está representado na *Figura 43*. Analisando a referida figura, é possível, mais uma vez, comprovar as boas condições de ensaio disponíveis na plataforma desenvolvida.

Na *Figura 44* estão representados dois contornos originais e uma imagem com as correspondências obtidas entre estes.

6.5.1.2 – Para superfícies

Criação de superfícies

Para inserir uma superfície, constituída por n linhas e m colunas, o utilizador pode especificar,

através de cliques no botão esquerdo do rato, as localizações pretendidas para os seus nodos na imagem *bitmap* original, sendo o nível de brilho de cada *pixel* seleccionado tomado como a sua terceira coordenada.

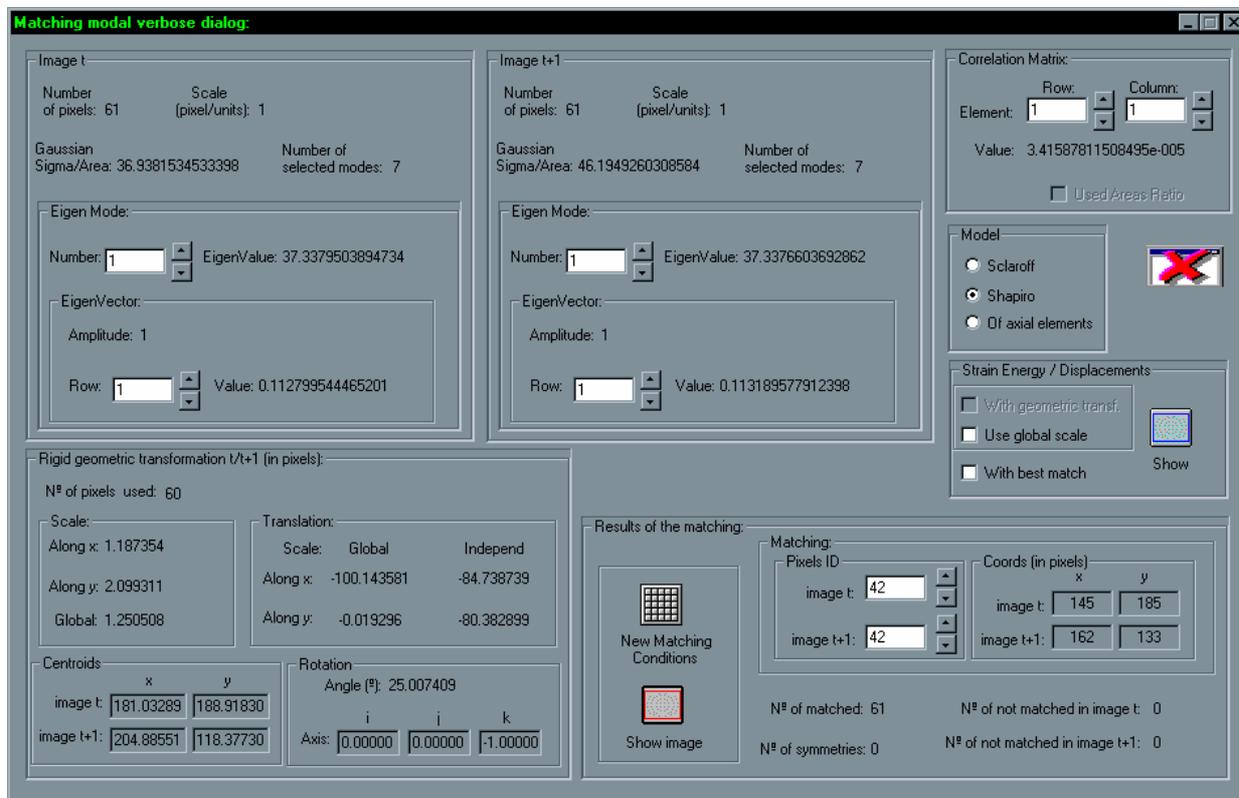


Figura 43 – Verificação dos resultados obtidos na determinação das correspondências entre dois contornos.

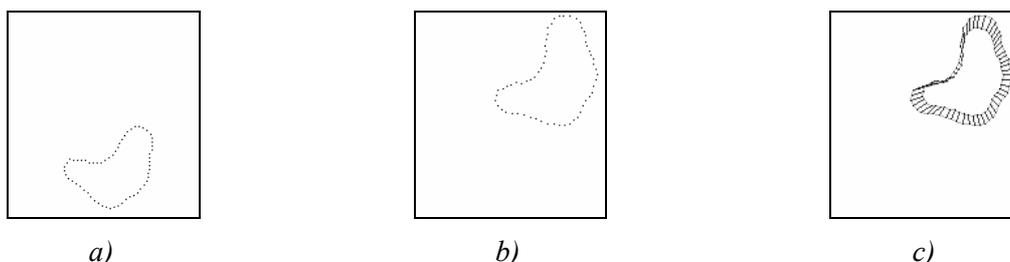


Figura 44 – Determinação das correspondências entre dois contornos: a) Contorno 1; b) Contorno 2; c) Correspondências determinadas após aplicação aos pixels do contorno 1 da rotação e da translação determinadas.

Em vez de utilizar esta inserção manual, é também possível criar automaticamente uma superfície por amostragem rectangular de uma imagem *bitmap*, utilizando dois níveis de detalhe: um para a zona interior do rectângulo que circunscreve o objecto e um outro para a zona exterior, ou considerando o mesmo nível de detalhe para toda imagem [2].

Na *Figura 45* é possível verificar-se a visualização de uma superfície utilizando-se a aplicação de domínio público *Win4Dv* [42], útil para a visualização de malhas poligonais, resultante de uma amostragem rectangular uniforme de uma imagem original.

Também foi implementada uma função para a criação automática de superfícies utilizando-se amostragem adaptativa [2]. Nesta amostragem 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.

Na *Figura 46* está representada a superfície resultante da amostragem adaptativa da imagem original da *Figura 45*.

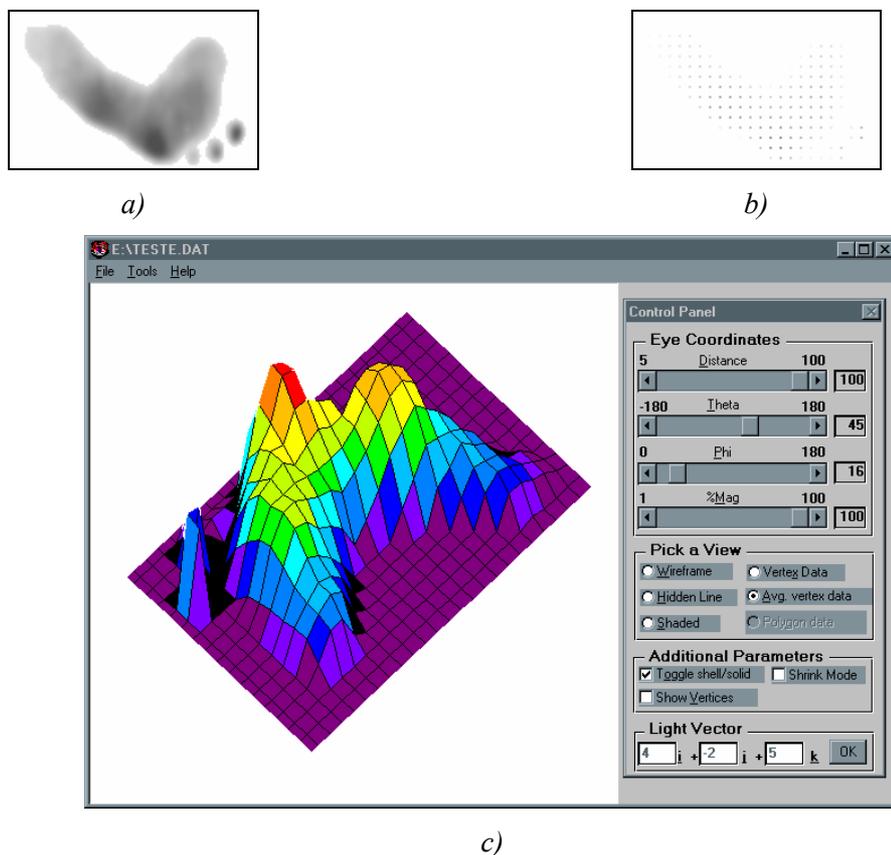


Figura 45 – Visualização 3D de uma superfície: a) Imagem original (negada); b) Amostragem resultante; c) Superfície visualizada na aplicação Win4Dv.

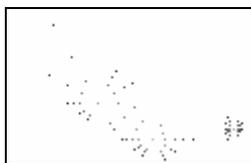


Figura 46 – Criação de uma superfície para a imagem original da Figura 45 utilizando-se amostragem adaptativa.

Utilitários

Em qualquer instante o utilizador pode apagar uma superfície, converter uma superfície num contorno aberto, 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 verificar se é ou não uma superfície com multiresolução e medir os seguintes parâmetros: número de *pixels*, coordenadas do primeiro *pixel* e do centróide, dimensões máximas e mínimas.

Muitas vezes é necessário aplicar a uma superfície uma transformação geométrica rígida 3D, constituída por uma rotação em torno de um determinado ponto, um escalamento em relação a um dado ponto e uma translação. Esta transformação foi implementada na plataforma desenvolvida sendo os parâmetros da transformação pretendida facilmente especificados pelo utilizador.

Como já foi referido, a aplicação de domínio público *Win4Dv* é útil para se visualizar objectos de malha poligonal; assim foram integradas funções para exportar uma ou duas superfícies especificadas sendo, neste caso, possível definir qual a translação a utilizar entre elas e qual a escala a considerar para cada uma, para visualização 3D na referida aplicação.

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* [14, 15, 30], considerando ou não a cota dos pontos originais como terceira coordenada dos pontos do objecto resultante, e a triangulação 3D de *Delaunay*.

Na *Figura 47* estão representados estes tipos de triangulação considerando os pontos originais da *Figura 46*.

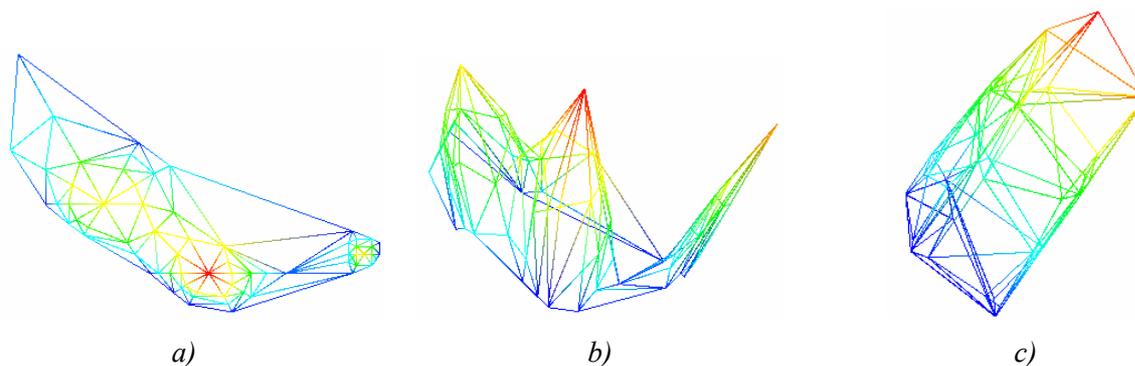


Figura 47 – Triangulações de Delaunay dos pontos da Figura 46: a) Triangulação 2D; b) Triangulação 2D utilizando a cota como a terceira coordenada; c) Triangulação 3D.

A ferramenta para reconstrução de superfícies [14, 15] existente no *VTK* foi também disponibilizada na plataforma. 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.

Emparelhamento de superfícies

A análise da vibração de uma superfície em modo livre foi também implementada na plataforma desenvolvida. 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 *Scalaroff* 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*, ou um ficheiro para posterior visualização 3D na aplicação *Win4Dv*, considerando os deslocamentos associados a um único ou a um conjunto de modos.

Na *Figura 48* está representado o diálogo para o utilizador verificar os resultados da análise e criar as imagens de resultados pretendidas.

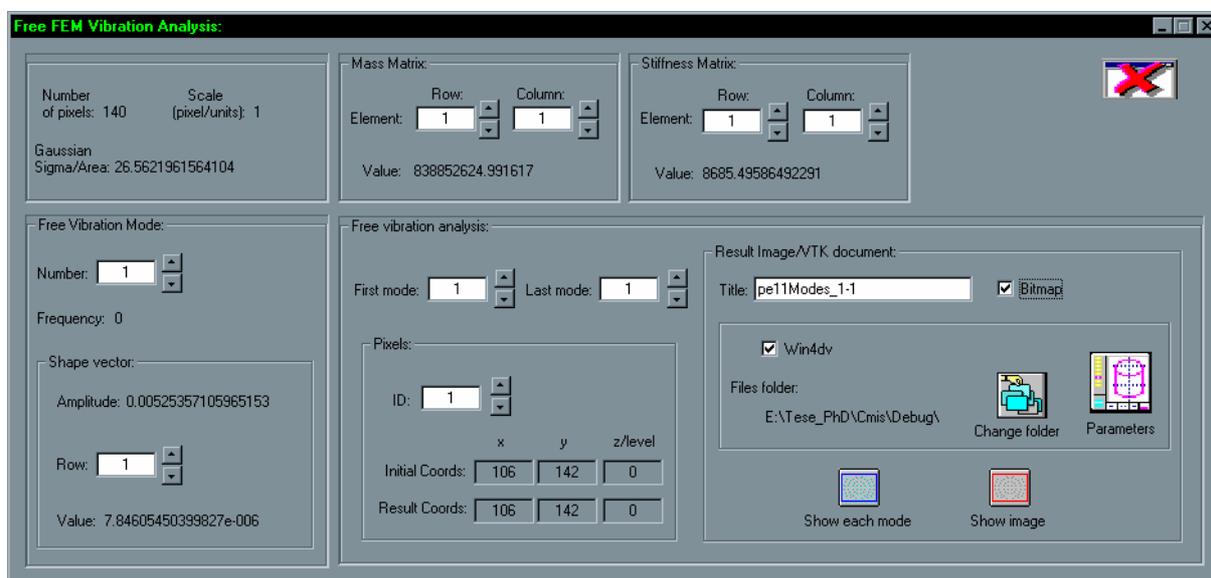


Figura 48 – Análise da vibração de uma superfície em modo livre.

A determinação das correspondências entre duas superfícies utilizando o método de mapeamento segundo a distância mínima foi também implementada na plataforma de desenvolvimento.

Na *Figura 49* está representado o diálogo que permite ao utilizador verificar os resultados obtidos para o emparelhamento e a transformação rígida estimada, alterar as condições de emparelhamento e construir as imagens de resultados que pretenda.

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 *Scalaroff* 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 entre as duas

superfícies; 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*, ou um ficheiro para a aplicação *Win4Dv*, com os resultados obtidos; e alterar as condições do emparelhamento modal e visualizar os novos resultados obtidos.

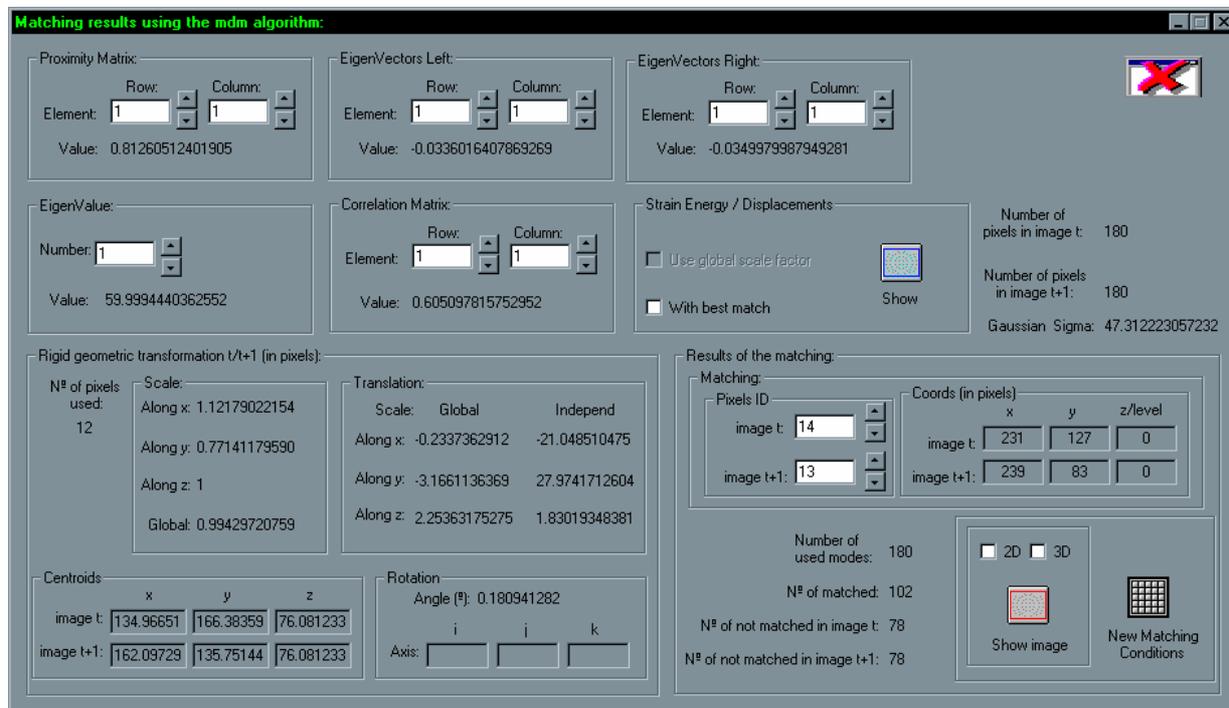


Figura 49 – Determinação das correspondências entre duas superfícies utilizando-se o método de mapeamento segundo a distância mínima.

Na *Figura 50* está visível o diálogo apresentado ao utilizador para controlar os resultados obtidos no emparelhamento de duas superfícies, e na *Figura 51* é possível verificar-se um exemplo de visualização das correspondências obtidas utilizando-se a aplicação *Win4Dv*.

6.5.2 – Documentos VTK

Para documentos do tipo *VTK* foram integradas mais algumas funções na plataforma desenvolvida 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 entre os dois, a determinação dos deslocamentos nodais e o valor da energia de deformação necessária para os alinhar. Seguidamente são descritas as funções implementadas.

Foi implementada a análise (2D ou 3D) da vibração de um objecto⁶ 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, criar um novo

⁶ Na análise 2D as coordenadas ao longo do eixo z dos objectos são desprezadas.

documento considerando os deslocamentos associados a este conjunto, e criar um novo documento para cada modo de vibração considerando os deslocamentos associados.

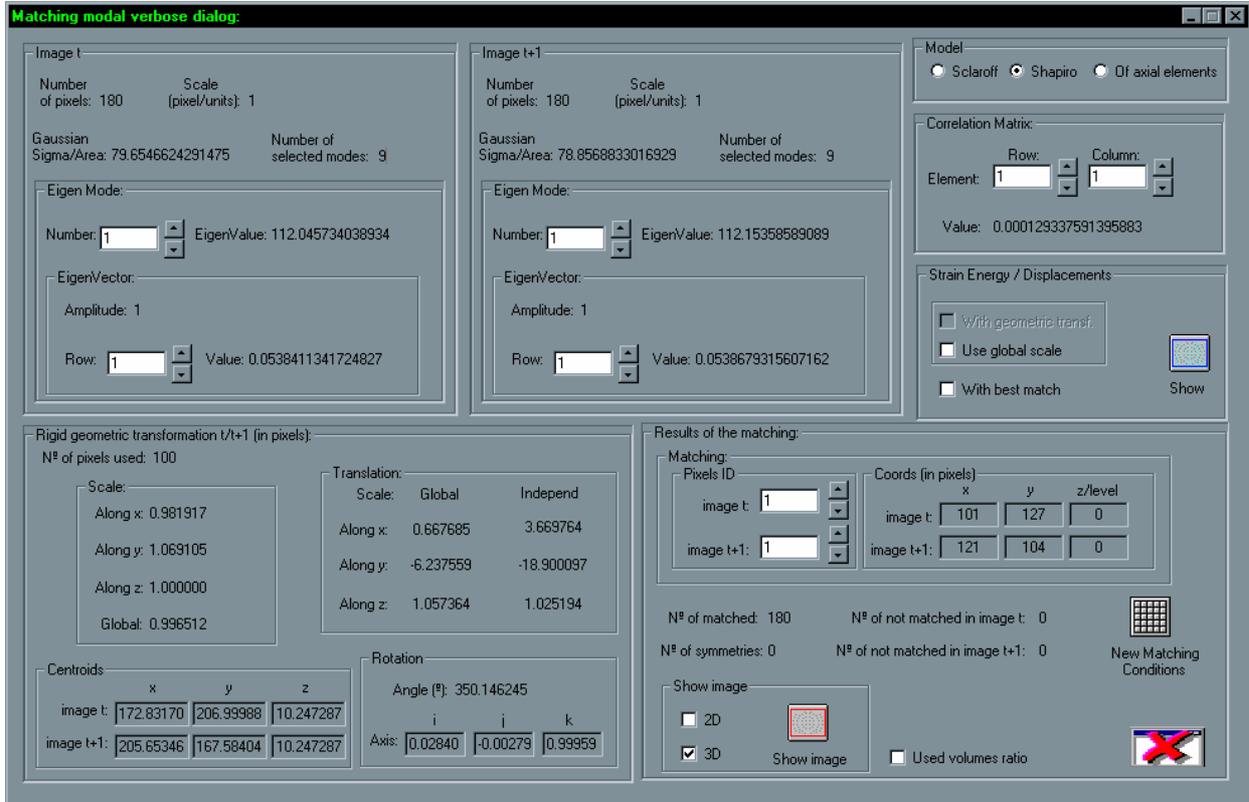


Figura 50 – Determinação das correspondências entre duas superfícies.

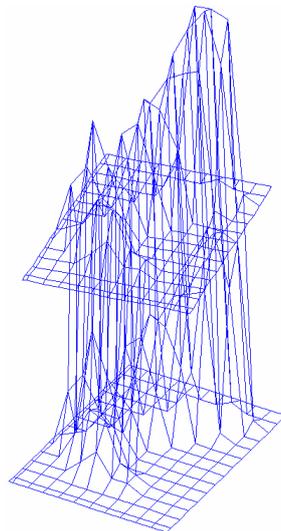


Figura 51 – Visualização 3D, através da aplicação Win4Dv, das correspondências obtidas entre duas superfícies.

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* utilizando⁷ 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 entre os dois objectos, 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 *Figura 52* é possível verificar-se um exemplo da representação das correspondências obtidas entre dois objectos 3D.

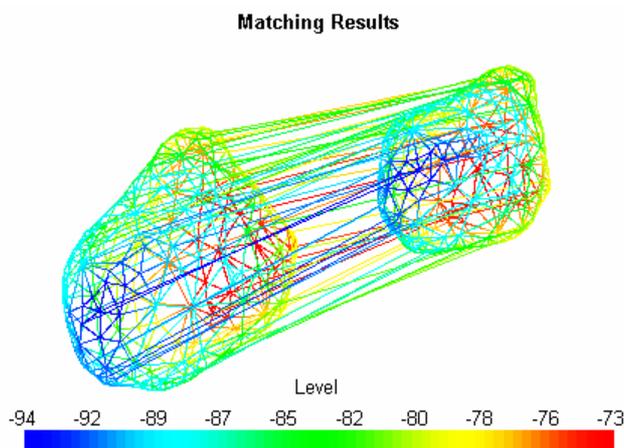


Figura 52 – Exemplo da representação das correspondências determinadas entre dois objectos.

7 – Sumário e conclusões

Neste relatório foi apresentada a plataforma *CMIS*, implementada em linguagem *C++* no ambiente integrado de desenvolvimento *Microsoft Visual C++*, para sistemas operativos *Microsoft Windows* que tem vindo a ser desenvolvida na Faculdade de Engenharia da Universidade do Porto desde 1995.

A plataforma integra algumas bibliotecas de domínio público que foram apresentadas e referenciadas. Com estas bibliotecas ficam disponíveis estruturas e operações para cálculo matricial, operações para leitura e escrita de imagens em vários formatos, e estruturas e operações normalmente comuns em aplicações de processamento e análise de imagem e em

⁷ No caso 2D, são apenas consideradas as suas coordenadas no plano *xy*.

aplicações de computação gráfica.

A filosofia utilizada na estrutura e implementação da plataforma permite que esta seja facilmente configurada e adaptada a níveis diferentes de utilizadores e de aplicação, e a sua manutenção não seja demasiado complexa.

A plataforma desenvolvida 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 a plataforma 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.

As funções integradas no sistema foram descritas de forma sucinta neste relatório. Assim, em primeiro lugar foram apresentadas as funções de índole mais geral e comum a um sistema de processamento e análise de imagem, a seguir foram descritas as funções geralmente associadas a sistemas de computação gráfica e, por fim, foram relatadas as funções mais directamente relacionadas com o domínio dos objectos deformáveis. Este maior destaque às funções relacionadas com os objectos deformáveis, deveu-se ao facto que o estado actual da plataforma permite que esta seja considerada como um banco adequado para estudo e ensaio de metodologias para esse tipo de objectos.

Actualmente a aplicação *CMIS* já incorporou contribuições de outros investigadores e serviu de plataforma 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 relatório, 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 estado actual da plataforma apresentada atingiu-se um dos objectivos principais para 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. Por outro lado, o facto de investigadores com objectivos diversos partilharem o mesmo sistema base incentiva a sua interacção e facilita as operações de actualização e de manutenção do sistema.

Em resumo, em vez de um sistema fechado, desenvolveu-se um sistema aberto que é interessante para várias aplicações e utilizadores, nomeadamente como banco de ensaio com ferramentas variadas, de processamento e análise de imagem e de computação gráfica, permitindo o estudo detalhado de novos algoritmos. Com as ferramentas desenvolvidas e implementadas no âmbito dos objectos deformáveis conseguiu-se estender as potencialidades da plataforma a esse domínio.

Referências

- [1] J. M. R. S. Tavares, J. Barbosa, and A. J. Padilha, "Apresentação de um Banco de Desenvolvimento e Ensaio para Objectos Deformáveis," in *RESI – Revista Electrónica de Sistemas de Informação*, vol. 1, 2003.
http://www.fe.up.pt/~tavares/downloads/publications/artigos/banco_resi.pdf
- [2] J. M. R. S. Tavares, "Análise de Movimento de Corpos Deformáveis usando Visão Computacional," in *Faculdade de Engenharia: Universidade do Porto*, 2000.
<http://www.fe.up.pt/~tavares/downloads/publications/teses/tese%20phd.zip>
- [3] B. Stroustrup, *The C++ Programming Language*: Addison-Wesley Publishing Company, 1995
- [4] M. J. Young, *Mastering Microsoft Visual C++ 6*: Sybex, 1998
- [5] T. Pinto, "Relatório da Bolsa de Investigação 1998/1999," Instituto de Engenharia Biomédica 1999
- [6] P. S. Vieira, "Relatório da Bolsa de Investigação 1998/1999," Instituto de Engenharia Biomédica 1999
- [7] G. Barbosa, João Manuel Tavares, and A. J. Padilha, "Paralellisation of active contours," presented at RecPad'96 - 8th Portuguese Conference on Pattern Recognition, Guimarães, Portugal, 1996
- [8] R. Davies, "Newmat, A matrix library in C++," 1999. <http://webnz.co.nz/robert/>
- [9] William H. Press, Saul A. Teukolsky, William T. Vetterling, and B. P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing*: Cambridge University Press, 1992
- [10] S. Rimmer, *Supercharged Bitmapped Graphics*: Windcrest/McGraw-Hill, 1991
- [11] J. S. a. others, "CImage Class Version 1.2," 1997.
<http://web.ukonline.co.uk/Members/julian.smart>
- [12] T. G. Lane, "Jpeg Library," 1996
- [13] Will Schroeder, Ken Martin, and B. Lorensen, "The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization," presented at Visualization '96, 1996
- [14] Will Schroeder, Ken Martin, and B. Lorensen, *The Visualization Toolkit*, 2nd ed: Prentice Hall, 1998
- [15] Will Schroeder, Ken Martin, and B. Lorensen, *The VTK User's Guide*: Kitware Inc., 1999. <http://www.kitware.com>
- [16] R. Lima, "Processamento de Imagem na Análise de Documentos de Engenharia," in *Faculdade de Engenharia: Universidade do Porto*, 1995
- [17] L. P. G. Reis, "Sistema Integrado de Detecção e Seguimento de Contornos," in *Faculdade de Engenharia: Universidade do Porto*, 1995
- [18] M. F. P. V. Correia, "Análise de Movimento em Sequências de Imagens," in *Departamento de Engenharia Electrotécnica e de Computadores: Faculdade de Engenharia da Universidade do Porto*, 1995

- [19] Tor Lonnestad and O. Milvang, "XITE: X-based Image Processing Tools and Environment User's Manual," October 21 1992
- [20] J. M. R. S. Tavares, "Algumas Ferramentas para Visão Tridimensional por Computador," Faculdade de Engenharia da Universidade do Porto 1995.
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/ferramentas%203d.zip>
- [21] J. M. R. S. Tavares, "Obtenção de Estrutura Tridimensional a Partir de Movimento de Câmara," in *Faculdade de Engenharia: Universidade do Porto*, 1995.
<http://www.fe.up.pt/~tavares/downloads/publications/teses/tese%20msc.zip>
- [22] João Manuel R. S. Tavares and A. J. Padilha, "Matching lines in image sequences with geometric constraints," presented at RecPad'95 - 7th Portuguese Conference on Pattern Recognition, Aveiro, Portugal, 1995.
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/recpad%2095%201.pdf>
- [23] João Manuel R. S. Tavares and A. J. Padilha, "A new approach for merging edge line segments," presented at RecPad'95 - 7th Portuguese Conference on Pattern Recognition, Aveiro, Portugal, 1995.
<http://www.fe.up.pt/~tavares/downloads/publications/artigos/recpad%2095%202.pdf>
- [24] J. Richter, *Advanced Windows*: Microsoft Press, 1995
- [25] J. S. Lim, *Two-Dimensional Signal and Image Processing*: PTR Prentice Hall, 1990
- [26] R. J. Schalkoff, *Digital Image Processing and Computer Vision*: John Willey & Sons, Inc., 1989
- [27] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. SMC, 1978, pp. 62/66
- [28] Josef Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition Letters*, vol. 19, pp. 41/47, 1986
- [29] 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*, vol. 27, pp. 400/411, 1983
- [30] v. Foley and H. Feiner, *Computer Graphics*: Addison-Wesley, 1991
- [31] Larry Shapiro and J. M. Brady, "A Modal Approach to Feature-based Correspondence," Robotics Research Group, Department of Engineering Science, Oxford University 1992
- [32] Larry Shapiro and J. M. Brady, "Feature-based correspondence: an eigenvector approach," in *Butterworth-Heinemann Ltd*, vol. 10, 1992
- [33] L. Shapiro, "Report: Towards A Vision-Based Motion Framework," Robotics Research Group, Department of Engineering Science, Oxford University 1991
- [34] 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," Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto 1997.
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/shapiro.zip>
- [35] 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," Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto 1998.
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/sclaroff.zip>
- [36] Stan Sclaroff and A. Pentland, "Object Recognition and Categorization Using Modal Matching," Massachusetts Institute of Technology, Media Laboratory 267, 1994
- [37] Stan Sclaroff and A. Pentland, "Modal Matching for Correspondence and Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

- vol. 17, 1995
- [38] S. E. Sclaroff, "Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals," in *Massachusetts Institute of Technology*, 1995
- [39] João Manuel R. S. Tavares and A. J. Padilha, "Relatório Interno: Apresentação de uma Modelização por Elementos Finitos Lineares do Tipo Axial," Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto 1998.
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/axial.zip>
- [40] 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," Instituto de Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto Junho 1997.
<http://www.fe.up.pt/~tavares/downloads/publications/relatorios/horner.zip>
- [41] B. K. P. Horn, "Closed-form Solution of Absolute Orientation using Unit Quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629/642, 1987
- [42] S. Shenoy, "Win4Dv v1.0: Viewer for 3D polygonal meshes and data," Department of AEEM, Iowa State University, IA 50010, USA, 1995

Anexo: Compilação do CMIS

Neste anexo é descrito o processo de compilação da plataforma *CMIS* no ambiente *Microsoft Visual C++*.

Em primeiro lugar é preciso descompactar o ficheiro contendo programa fonte⁸ *CMIS* mantendo, se possível, os mesmos directórios dos vários ficheiros que integram a aplicação. Neste conjunto de ficheiros estão já os ficheiros das bibliotecas *Cimage*, *Jpeglib* e *Newmat*.

A seguir à descompactação do programa fonte, é preciso instalar a biblioteca⁹ *VTK*. Para realizar esta operação, existem duas soluções possíveis: uma que consiste em instalar a biblioteca previamente compilada; a segunda, preferível, é compilar localmente a biblioteca. Esta segunda solução é composta por duas etapas: descompactação dos ficheiros fonte e compilação no *Microsoft Visual C++* da biblioteca segundo as instruções contidas do ficheiro *README.html* (incluído no ficheiro comprimido que contém os ficheiros fonte do *VTK*).

Após a instalação, ou da compilação, da biblioteca *VTK* deve-se abrir o projecto *CMIS* no *Microsoft Visual C++* e acrescentar os directórios dos ficheiros fonte da biblioteca *VTK* nos directórios a incluir no projecto (item opções do menu ferramentas), *Figura 53*.

A fase seguinte consiste em acrescentar as bibliotecas do *VTK* para a sua ligação ao projecto *CMIS*. Assim para a versão de *debug* do *CMIS* deverão ser incluídas, *Figura 54*, as seguintes bibliotecas (os directórios poderão ter outras designações):

⁸ O programa fonte deverá ser solicitado ao autor deste relatório. O programa previamente compilado, com o respectivo processo de instalação, está disponível em: <http://www.fe.up.pt/~tavares>.

⁹ Disponível em <http://www.kitware.com>.

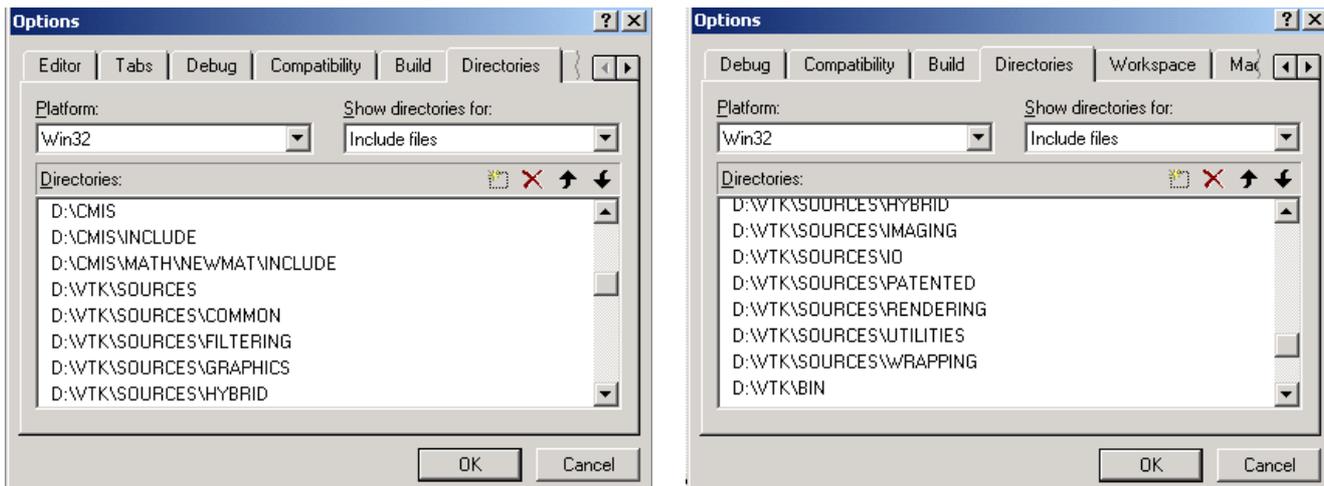


Figura 53 – Inclusão dos directórios dos ficheiros fonte do VTK no Microsoft Visual C++.

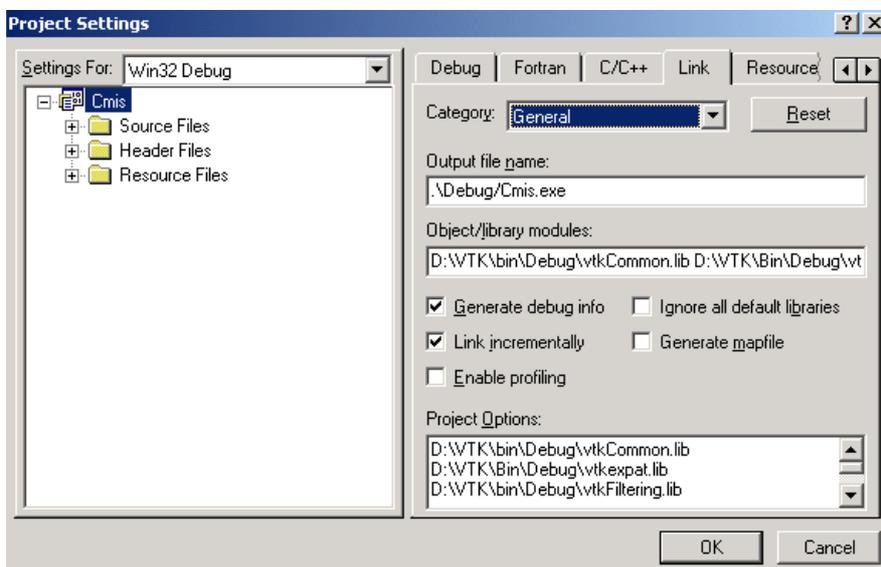


Figura 54 – Inclusão dos bibliotecas VTK no Microsoft Visual C++ para a versão debug.

- D:\VTK\bin\Debug\vtkCommon.lib
- D:\VTK\Bin\Debug\vtkexpat.lib
- D:\VTK\bin\Debug\vtkFiltering.lib
- D:\VTK\Bin\Debug\vtkfreetype.lib
- D:\VTK\bin\Debug\vtkGraphics.lib
- D:\VTK\Bin\Debug\vtkftgl.lib
- D:\VTK\bin\Debug\vtkHybrid.lib
- D:\VTK\bin\Debug\vtkImaging.lib
- D:\VTK\bin\Debug\vtkIO.lib
- D:\VTK\bin\Debug\vtkjjpeg.lib
- D:\VTK\bin\Debug\vtkPatented.lib

- *D:\VTK\bin\Debug\vtkpng.lib*
- *D:\VTK\bin\Debug\vtkRendering.lib*
- *D:\VTK\bin\Debug\vtktiff.lib*
- *D:\VTK\bin\Debug\vtkzlib.lib*

Acrescentar as bibliotecas para a versão *release* do *CMIS* tem um procedimento semelhante, *Figura 55*, sendo as bibliotecas a considerar:

- *D:\VTK\bin\Release\vtkCommon.lib*
- *D:\VTK\Bin\Release\vtkexpat.lib*
- *D:\VTK\bin\Release\vtkFiltering.lib*
- *D:\VTK\Bin\Release\vtkfreetype.lib*
- *D:\VTK\bin\Release\vtkGraphics.lib*
- *D:\VTK\Bin\Release\vtkftgl.lib*
- *D:\VTK\bin\Release\vtkHybrid.lib*
- *D:\VTK\bin\Release\vtkImaging.lib*
- *D:\VTK\bin\Release\vtkIO.lib*
- *D:\VTK\bin\Release\vtkjpeg.lib*
- *D:\VTK\bin\Release\vtkPatented.lib*
- *D:\VTK\bin\Release\vtkpng.lib*
- *D:\VTK\bin\Release\vtkRendering.lib*
- *D:\VTK\bin\Release\vtktiff.lib*
- *D:\VTK\bin\Release\vtkzlib.lib*

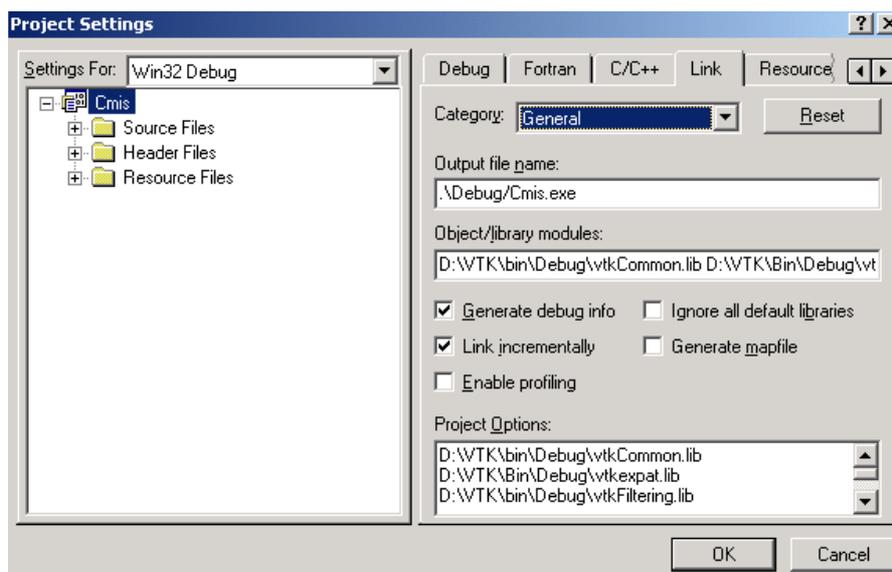


Figura 55 – Inclusão dos bibliotecas VTK no Microsoft Visual C++ para a versão release.

Finalmente deverá ser incluído o directório que contém as *dll*'s da biblioteca *VTK*, preferencialmente na versão *debug*, no *path* do sistema operativo, *Figura 56*.

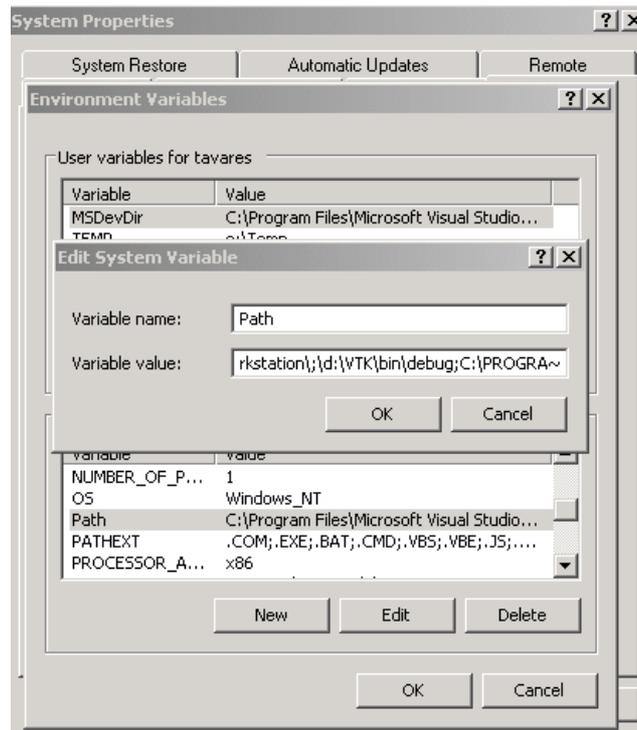


Figura 56 – Inclusão do directório das dll's da biblioteca VTK no path do sistema operativo.