

Interface para processamento de imagem em C++ utilizando Visual Studio .NET 2005



Universidade do Porto

Faculdade de Engenharia

FEUP

Ilda Marisa de Sá Reis
João Manuel R. S. Tavares

OUTUBRO 2007

Agradecimentos

Em primeiro lugar, agradeço ao Prof. João Tavares pela orientação do meu projecto de doutoramento e, em particular, deste trabalho.

À Dra. Isabel Sampaio gostaria de expressar o meu reconhecimento pelo apoio prestado aquando da apresentação dinâmica na barra de estado da interface das coordenadas dos *pixels* da imagem activa.

À Fundação para a Ciência e a Tecnologia agradeço a bolsa de doutoramento, com a referência SFRH/BD/29012/2006, concedida no âmbito do projecto POS_C - Apoiar o Desenvolvimento de Competência.

À Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança agradeço a equiparação a bolseira da Fundação para a Ciência e Tecnologia.

Por último, uma palavra de apreço a todos os que de algum modo contribuíram para a realização deste trabalho.

Resumo

Este relatório tem por objectivo apresentar uma interface, em fase de desenvolvimento, que realiza funções básicas características de uma aplicação do sistema operativo *Microsoft Windows* e implementar um algoritmo simples do domínio do processamento de imagem.

Abstract

The aim of this report is to present an interface which is being in development. It performs basic functions which are typical in software applications of the Microsoft Windows operating system and it implements also a basic algorithm of image processing domain.

Índice

1	Introdução	1
2	Bibliotecas Integradas	2
2.1	Windows Forms	2
2.2	OpenCV	5
3	Interface em Desenvolvimento	9
3.1	Abrir uma imagem	10
3.2	Coordenadas dos <i>pixeis</i> de uma imagem	10
3.3	Criação de controlos próprios	10
3.4	Zoom de uma imagem	11
3.5	<i>Threshold</i> de uma imagem	12
4	Conclusões e Trabalho Futuro	13
4.1	Conclusões	13
4.2	Trabalho Futuro	13
	Bibliografia	14

Capítulo 1

Introdução

A construção desta interface teve como principais objectivos, até este momento, permitir ao primeiro autor a realização de algumas operações básicas do domínio do processamento e análise de imagem e simultaneamente familiarizar-se com uma linguagem de programação orientada aos objectos.

A linguagem de programação empregue no desenvolvimento da aplicação foi a linguagem C++ em ambiente Visual Studio .NET 2005. Os factores que estiveram na génese desta escolha foram a sua robustez, eficiência, portabilidade e a ampla disseminação da mesma. Outro factor merecedor de grande relevo foi a existência de bibliotecas de domínio público, escritas em C/C++, de entre as quais se destaca a biblioteca *OpenCV* [4] particularmente útil para processamento de imagem (otimizada para aplicações em tempo real) e com diversas aplicações em áreas como o reconhecimento e segmentação de objectos.

No futuro pretende-se que esta interface possa integrar outras bibliotecas, como por exemplo, a VTK [8] para visualização e processamento de objectos gráficos a duas e três dimensões; a *Free DICOM* [9] para importação de ficheiros DICOM (formato de dados amplamente difundido na área da medicina para armazenamento e transmissão de imagens médicas).

Este relatório está organizado do seguinte modo: primeiro são descritas as bibliotecas de domínio público que a interface incorpora até ao momento, bem como o procedimento a ter em conta para efectuar a sua instalação; seguidamente são descritas as características e funções da interface e, finalmente, apresentadas algumas conclusões e mencionado o trabalho que se pretende desenvolver.

Capítulo 2

Bibliotecas Integradas

A interface, como já referido anteriormente, foi programada em C++ em ambiente Visual Studio .NET 2005 - o mais recente ambiente de desenvolvimento criado pela *Microsoft* para a sua arquitectura .NET.

A *Framework* .NET é uma plataforma especialmente útil para o desenvolvimento de interfaces gráficas. Um dos seus principais componentes é o *Common Language Runtime* (CLR) que proporciona uma execução segura e um ambiente com alto desempenho para as aplicações. O código que é executado pelo CLR diz-se *managed* enquanto que o código que é executado fora do CLR diz-se *unmanaged*. A única linguagem .NET que permite combinar código *managed* com código não *managed* é a C++, [1, 2]. Este facto contribuiu igualmente para sustentar a opção por esta linguagem pois algumas das bibliotecas que integram a interface estão escritas com código *unmanaged*.

Até ao momento a interface integra duas bibliotecas: *Windows Forms* (código *managed*) e OpenCV (código não *managed*).

2.1 Windows Forms

A biblioteca *Windows Forms* é um importante recurso da *Microsoft Windows .NET Framework* que fornece um conjunto de classes para a construção de interfaces gráficas.

Contrariamente a outras bibliotecas para a criação de interfaces gráficas, como por exemplo a *Microsoft Foundation Classes* (MFC), a biblioteca *Windows Forms* pode ser usada a partir de qualquer linguagem .NET, sendo possível a criação de aplicações gráficas com mistura de linguagens, [1, 2].

Uma aplicação *Windows Forms* consiste numa ou mais janelas chamadas *forms*. Estas janelas podem ser janelas principais, janelas filhas ou caixas de diálogo. A cada *form* pode ser adicionado um conjunto de controlos para construir a interface do programa, como por exemplo, botões, caixas de texto, etc.

O Visual C++ 2005 dispõe de uma área de *design* para criação de aplicações *Windows Forms*, no entanto optou-se por criar a aplicação manualmente porque, tal como referido em [1], consideramos também que o código criado é mais limpo, menos extenso, além de nos permitir entender melhor os princípios gerais por detrás da programação orientada ao objecto (propriedades, eventos, etc.).

Seguidamente é apresentado o procedimento manual para criação de uma aplicação *Windows Forms*, tal como é descrito em [1]:

- Criar um novo projecto com o modelo **CLR Empty Project** do Visual Studio 2005. O nome atribuído ao projecto foi **Exemplo**.
- Após seleccionar a opção propriedades do menu de contexto associado ao nome do projecto na janela **Solution Explorer**, escolher a opção **Reference** em **Common Properties**, como mostra a figura 2.1.

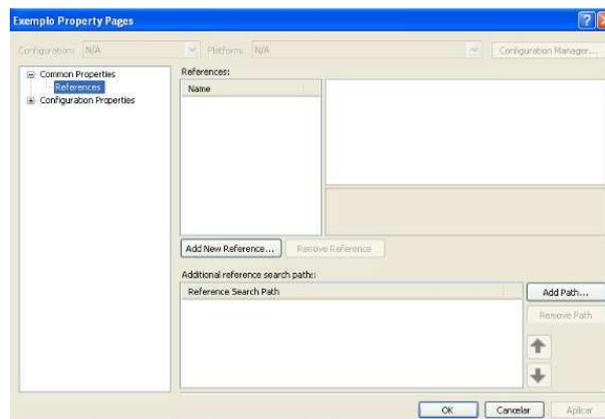


Figura 2.1: Janela *Property Pages* (Referências)

- Na janela anterior dar um clique no botão **Add New Reference** . Na nova janela seleccionar o separador **.NET**, figura 2.2.

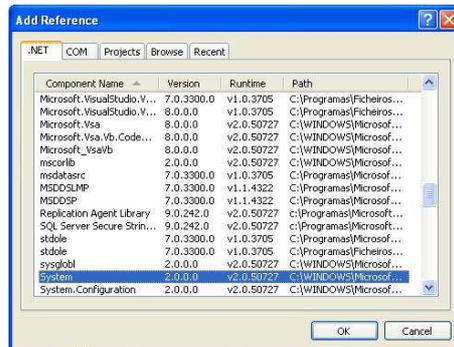


Figura 2.2: Janela que permite escolher referências

Para criar uma aplicação *Windows Forms* é necessário, tal como indicado na figura 2.3, adicionar as referências: *System*, *System.Windows.Forms* e *System.Drawing*. A criação da aplicação estará concluída após premir o botão **OK**.

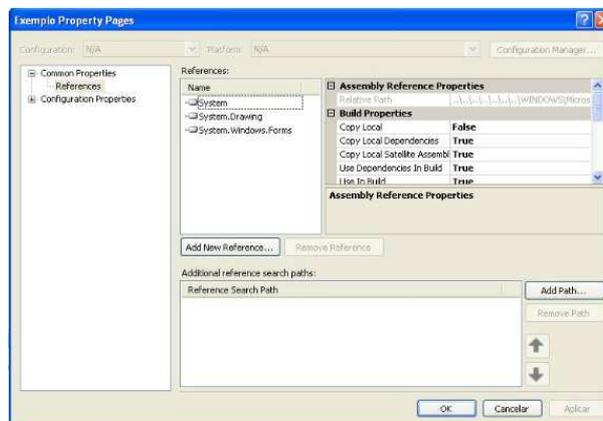


Figura 2.3: Janela *Property pages* com as referências para *Windows Forms*

Neste momento falta apenas adicionar os ficheiros *.h* e *.cpp* para escrever o código da aplicação. Nos ficheiros *header* (*.h*) deve constar as declarações de classes e protótipos dos seus métodos enquanto que nos ficheiros *.cpp* deve constar o código fonte de cada método.

Para adicionar um ficheiro `.cpp` ao projecto proceder-se do seguinte modo:

- Seleccionar a opção **Add -> New File** associada ao menu de contexto da pasta **Source Files**, na janela **Solution Explorer**.
- Seleccionar o modelo **C++ File (.cpp)** da categoria **Code**, como exibido na figura 2.4. Após atribuir um nome ao ficheiro, seleccionar o botão **Add**.

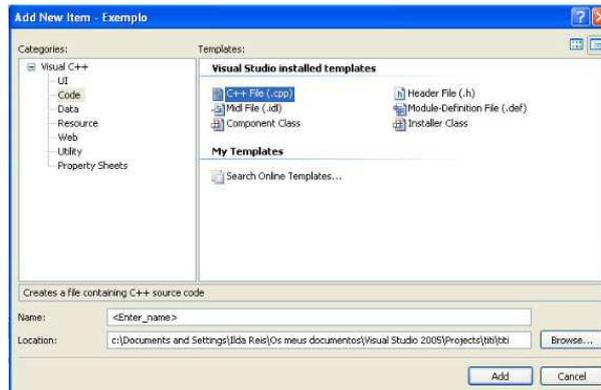


Figura 2.4: Janela *Add New Item*

Procedimento análogo deve ser realizado para inserir um ficheiro *header* no projecto.

2.2 OpenCV

OpenCV - *Open Source Computer Vision* é a segunda biblioteca computacional que a interface integra. Esta biblioteca de domínio público, disponível para *download* em [3], possui um vasto conjunto de funções em C/C++ e implementa alguns dos algoritmos mais usuais de processamento de imagem e visão por computador [4].

A biblioteca concebida especialmente para processamento e análise de imagem em tempo real tem aplicações [6], nas seguintes áreas: interface homem-máquina, identificação de objectos, segmentação e reconhecimento, reconhecimento de faces, reconhecimento de movimento, seguimento de movimento, robótica móvel, entre outras.

Seguidamente é apresentado o procedimento para instalação da biblioteca OpenCV introduzindo as modificações necessárias ao indicado em [5]:

- Abrir o projecto onde se pretende instalar a biblioteca OpenCV. No nosso caso, fá-lo-emos no projecto a que dêmos o nome **Exemplo** (criado na secção 2.1)
- Após seleccionar a opção propriedades do menu de contexto associado ao nome do projecto, na janela **Solution Explorer**, escolher a opção **Configuration Properties -> C/C++ -> General**, tal como indicado na figura 2.5

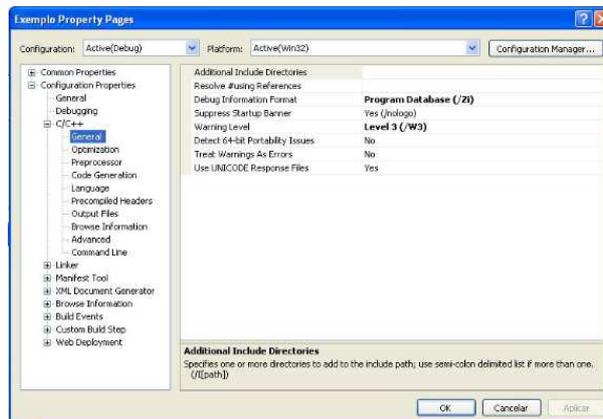


Figura 2.5: Janela *Property Pages* (Geral)

- Na janela anterior, quando seleccionar a opção **Additional Include Directories** aparece do lado direito uma caixa com reticências que ao ser premeida faz aparecer uma janela como a exibida na figura 2.6.



Figura 2.6: Janela *Additional Include Directories*

Nesta nova janela é necessário inserir os seguintes directórios:

C:\...\OpenCV\cv\include

C:\...\OpenCV\cvaux\include

C:\...\OpenCV\cxcore\include

C:\...\OpenCV\otherlibs\highgui

- A seguir, ao escolher a opção **Configuration Properties -> Linker -> General -> Additional Library Directories** aparece a janela indicada na figura 2.7

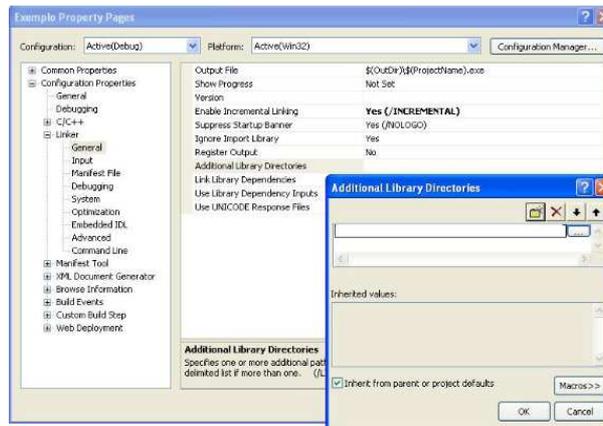


Figura 2.7: Janelas *Property Pages* e *Additional Library Directories*

onde devem ser incluídos os directórios:

C:\...\OpenCV\lib

C:\...\OpenCV\bin

- Finalmente, ao escolher a opção **Configuration Properties -> Linker -> Input -> Additional Dependencies** surge a janela apresentada na figura 2.8

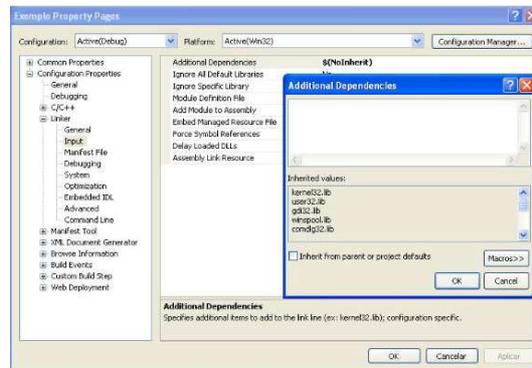


Figura 2.8: Janelas *Property Pages* e *Additional Dependencies*

onde devem ser adicionados as seguintes dependências: *cv.lib*, *highgui.lib*, *cxcore.lib* e *cvaux.lib*

Neste momento a biblioteca OpenCV está integrada, mas para que as suas funções possam ser usadas é necessário no início do ficheiro *.cpp* fazer o *#include* dos ficheiros onde constam essas funções.

Capítulo 3

Interface em Desenvolvimento

A interface segue a filosofia geral das aplicações desenvolvidas para o sistema operativo *Microsoft Windows*. É *Multiple Document Interface* (MDI), isto é, permite abrir vários *forms*, numa instância da aplicação, como se pode ver na figura 3.1 que se segue:



Figura 3.1: Interface de Processamento de Imagem

Os elementos que constituem a janela principal, *main form*, são: menu, menu de contexto e barra de estado. De acordo com a janela activa, *child form*, os menus adaptam-se dinamicamente podendo assim determinadas opções estar ou não activas.

Seguidamente será apresentado de forma detalhada as principais funcionalidades actuais da interface.

3.1 Abrir uma imagem

Ao seleccionar-se a opção *Open File ...* do menu *File*, surge uma caixa de diálogo modal, de acordo com a ilustrada na figura 3.2, que permite ao utilizador escolher uma imagem.

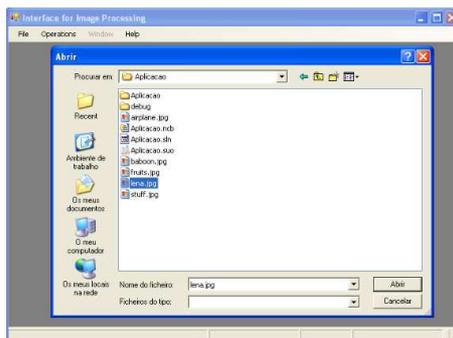


Figura 3.2: Caixa de diálogo para escolha da imagem a abrir

Essa imagem é apresentada num novo *child form* que contém uma instância da classe *PictureBox* e cuja propriedade *Image* permite a visualização da mesma. Porém, se a imagem escolhida já se encontrar aberta é simplesmente activado o *form* que a contém.

3.2 Coordenadas dos *pixels* de uma imagem

Sempre que um *form* que contenha uma imagem for activado, o utilizador tem a possibilidade de visualizar dinamicamente, na barra de estado, as coordenadas do *pixel* indicado pelo apontador do rato. Note-se que as coordenadas só serão visíveis para o *form* activo, isto é, se o utilizador deslizar o rato sobre uma imagem inactiva não é disparado o evento que permite visualizar as suas coordenadas.

3.3 Criação de controlos próprios

As aplicações Windows Forms têm classes para criar variadíssimos tipos de controlos, tais como, *ComboBox*, *Form*, *GroupBox*, *Label*, *ListBox*, *MessageBox*, *PictureBox*, *ToolBar*, entre outros.

À medida que a interface ia sendo construída constatamos que seria desejável

guardar informação adicional para além da que os controlos existentes permitem. Por exemplo, notamos que a maior parte dos filtros são aplicáveis a imagens em tons de cinzento, pelo que seria boa ideia guardar desde logo a imagem em tons de cinzento em vez de fazer a conversão para essa escala de tons sempre que quiséssemos aplicar novo filtro.

A solução para o problema acima descrito passou pela criação de um controlo próprio. Este controlo é uma instância de uma nova classe que deriva da classe *UserControl* do espaço de nome *System.Windows.Forms*.

3.4 Zoom de uma imagem

Quando um *form* contendo uma imagem está activo, o utilizador pode visualizar na barra de estado se a imagem está no seu tamanho real ou, pelo contrário, se encontra ampliada ou reduzida. A mesma informação pode ser consultada no menu de contexto associado à imagem cujo acesso é permitido quando o utilizador posiciona o rato sobre a imagem e pressiona o botão direito do rato. O menu de contexto dá ainda a possibilidade de alterar a escala de visualização da imagem, tal como é ilustrado na figura 3.3.



Figura 3.3: Menu de contexto da imagem que permite alterar a escala de visualização

3.5 *Threshold* de uma imagem

Threshold [6] é uma função que é usada essencialmente com dois objectivos: converter uma imagem em tons de cinzentos, *grayscale*, numa imagem binária (em geral, preto e branco) ou eliminar ruído de uma imagem, por exemplo, ‘corrigir’ o valor dos *píxeis* que estão fora de uma gama de valores.

Esta função fica disponível na interface sempre que o utilizador selecciona uma imagem à qual ainda não tenha sido aplicado um *threshold*. No nosso caso, o objectivo da função *threshold* é binarizar a correspondente imagem em tons de cinzentos. O resultado desta operação é colocado num novo *form* cujo título reflecte a designação do nome original da imagem e da operação realizada. Por defeito, o valor do *threshold* é zero, mas o utilizador tem à sua disposição uma *trackbar* que lhe permite modificar o valor e visualizar em tempo-real as modificações na imagem. A figura 3.4 mostra dois *forms*, num deles é apresentada a imagem a cores e no outro o *threshold* de nível 120 da correspondente imagem em tons de cinzentos.

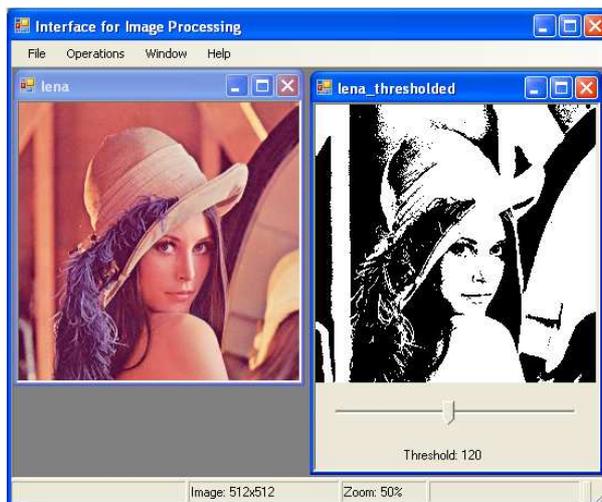


Figura 3.4: *Threshold* dinâmico da imagem correspondente em tons de cinzentos (nesta imagem o valor do *threshold* é 120)

Como esta operação cria uma nova imagem, a opção *Save Image* do menu *File* é activada e no caso do utilizador optar por gravá-la o nome sugerido é o que consta no título do *form* seguido do valor do *threshold* (por exemplo, na figura acima o nome sugerido seria *lena_thresholded120.jpg*).

Capítulo 4

Conclusões e Trabalho Futuro

Neste capítulo final são apresentadas as principais conclusões acerca do desenvolvimento da interface e indicadas as futuras implementações que nos propomos concretizar.

4.1 Conclusões

A construção desta interface, como referido no capítulo 1, permitiu ao autor deste trabalho familiarizar-se com o Visual C++. A incorporação da biblioteca computacional OpenCV revelou-se uma mais valia pela panóplia de funções disponibilizadas. Por outro lado, o facto de usar código não *managed* acarretou algumas dificuldades adicionais, como por exemplo, conversão entre diferentes tipos de dados (nomeadamente *String* para *char*** e *Image* para *IplImage**). Para a resolução destes problemas contribuiu a partilha de ideias no grupo de discussão do OpenCV, [7].

4.2 Trabalho Futuro

Pretende-se no futuro melhorar significativamente a interface no que diz respeito à implementação e integração de novos algoritmos de processamento e análise de imagem, especialmente no que concerne a algoritmos baseados em *level sets*, [10].

Bibliografia

- [1] ISABEL SAMPAIO, ALBERTO SAMPAIO, *Visual C++/CLI - Curso Completo*, FCA - Editora de Informática, Lda, 2007
- [2] JULIAN TEMPLEMAN, ANDY OLSEN, *Microsoft Visual C++ .NET Step by Step*, Microsoft Press, 2003
- [3] Open Computer Vision Library, versão 1.0,
<http://sourceforge.net/projects/opencvlibrary>, 2006
(último acesso em Outubro/2007)
- [4] Manual de OpenCV Online,
<http://isa.umh.es/pfc/rmvision/opencvdocs>
(último acesso em Outubro/2007)
- [5] OpenCV with Visual C++ .NET 2003
<http://opencvlibrary.sourceforge.net/VisualC%2B%2B>
(último acesso em Outubro/2007)
- [6] INTEL, *Open Source Computer Vision Library - Reference Manual*, Intel Corporation, 2001
- [7] Grupo de discussão do OpenCV
<http://tech.groups.yahoo.com/group/OpenCV/>
(último acesso em Outubro/2007)
- [8] The Visualization Toolkit, <http://public.kitware.com/VTK/>
(último acesso em Outubro/2007)
- [9] Idoimaging.com Free Medical Imaging Software,
<http://www.idoimaging.com/index.shtml>
(último acesso em Outubro/2007)
- [10] J.A. SETHIAN, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*, Cambridge University Press, 1999.