Universidade do Minho
Escola de Engenharia/Departamento de Informática

Mestrado Integrado em Engenharia Biomédica

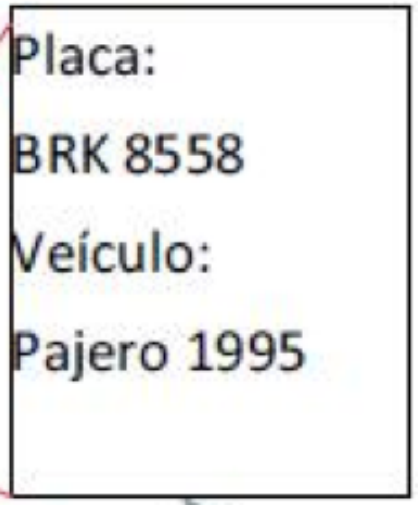
Algoritmos de Processamento de Imagens

IMAGIOLOGIA

INTRODUÇÃO

- Processamento de imagens é um processo onde a entrada do sistema é uma imagem e a saída é um conjunto de valores numéricos, que podem ou não compor uma outra imagem.

- Visão computacional procura emular a visão humana, portanto também possui como entrada uma imagem, porém, a saída é uma interpretação da imagem como um todo, ou parcialmente.



Veiculo em Imagem Escura

Após uma equalização e histogramas, em nível de cinza, onde a placa do veículo pode ser lida

Informação da placa e do veículo no retângulo

INTRODUÇÃO

- Processos de visão computacional geralmente iniciam com o processamento de imagens.
- Processamento ocorre em três níveis:
  - baixo-nível: operações primitivas (redução de ruído ou melhoria no contraste de uma imagem);
  - nível-médio: operações do tipo segmentação ou Classificação;
  - alto-nível: tarefas de cognição normalmente associadas com a visão humana.

Universidade do Minho
Escola de Engenharia/Departamento de Informática

Mestrado Integrado em Engenharia Biomédica

Tutorial - OpenCV

IMAGIOLOGIA

OpenCV

- **OpenCV** (***Open*** *Source* ***C****omputer* ***V****ision*) é uma biblioteca, de código aberto, desenvolvida inicialmente pela Intel em 2000.
- Biblioteca desenvolvida nas linguagens de **programação C/C++**.
- Dá suporte a programadores que utilizem Java, Python e Visual Basic.
- Possui mais de 500 funções.

OpenCV

- Foi idealizada com o objetivo de tornar a visão computacional acessível a utilizadores e programadores em áreas tais como a interação humano-computador em tempo real e a robótica.
- Desenvolvimento de aplicações na área de Visão por Computador.
- Implementa ferramentas de interpretação de imagens, indo desde operações simples como um filtro de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em 3D.

INSTALAÇÃO

- Linux
  http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation
- Windows
  http://docs.opencv.org/doc/tutorials/introduction/windows_install/windows_install.html#windows-installation
- Mac OS

  mac ports - http://www.macports.org/

  sudo port install opencv

ECLIPSE + CDT

- Instalar Eclipse
  http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/junosr2

- Criar novo projecto
  - Criar pasta src
  - Criar novo file .cpp
  - Configurar os *includes* para o projecto

ABRIR/ALTERAR IMAGEM

- Abrir imagem
  - *Mat imagem = imread(String filename);*
  - *Imagem.empty();* – Verificar se a imagem abriu;

- Alterar imagem
  - *Mat cinzento;*
  - *cvtColor(imagem, cinzento, CV_BRG2GRAY);*

MOSTRAR/GUARDAR IMAGEM

- Mostrar imagem
  - namedWindow("Nova imagem", CV_WINDOW_AUTOSIZE);
  - imshow("Nova Imagem", imagem);

- Guardar imagem
  - imwrite ("../../imagens/Cinzento.jpg", cinzento);

VIDEO

- Captura
  - *VideoCapture cap;*
  - *cap.open(0);* // Abrir webcam
  - *cap.open(String Filename);* // Abrir de um ficheiro
  - *cap.isOpened();* // verificação
  - *cap >> frame;* // captura um frame para um nova Mat
- Alteração
  - Feita em cada frame…

VIDEO

- Gravar
  - *VideoWriter video; //* Declarar
  - *video.open(const string& filename, int fourcc, double fps, Size frameSize, bool isColor=true);*
  - *video.isOpened(); //* Verificação
  - *video << frame; //* Escreve o frame no ficheiro

cap.get(<flag>); // para aceder a caracteristicas do video como tamanho, fps, etc

char c = waitKey(33); // espera pelo utilizador durante 33 milisegundos

EXEMPLOS

- resize(frame, out, Size(0,0), 1/size, 1/size);
- Histogram Equalization // Link
- Filtering // Link

Sites de interesse:

http://opencv-srf.blogspot.pt/p/opencv-c-tutorials.html

http://docs.opencv.org/index.html

http://dasl.mem.drexel.edu/~noahKuntz/openCVTut1.html

http://www.laganiere.name/opencv1Tut/