GRUPO DE INTELIGENCIA COMPUTACIONAL (GIC), PERTENECIENTE A LA UNIVERSIDAD
DEL PAÍS VASCO (UPV-EHU)

Euclidean skeletons of digital image and volume data in linear time by integer medial axis transform. A C++ implementation

Andoni Beristain Iraola

beristainandoni@yahoo.es

April 2009

Contents

Universidad del País Vasco   Euskal Herriko Unibertsitatea

1. Introduction

This document describes a C++ implementation of the algorithm for Euclidean Skeleton computation based on the Integer Medial Axis transform (IMA), proposed by Hesselink and Roerdink from the University of Groningen in the paper cited in the References section of this document.

This document as well as the source code are provided without any kind of warranty.

The piece of code included is composed of these files:

Skeleton.h: Header file containing the listing of functions for skeleton computation.

Skeleton.h: Source file containing the implementation of the skeletonization algorithm.

SkeletonExample.cpp: Source file showing an example about how to use the skeletonization function in Skeleton.cpp. Makes use of the OpenCV library.

Stdafx.h: Include file for standard system include files.

Stdafx.cpp: Source file that includes just the standard includes.

Agarrar_A.bmp: Example binary image. 320*240 grayscale image.

SkeletonExample.sln and SkeletonExample.vcproj: Solution and project file for Visual Studio 2005.

2. Requirements

The code in Skeleton.h and Skeleton.cpp does not have any special library dependency.

The example project requires Visual Studio 2005 or later and the Open Computer Vision (OpenCV) library, which is used for image loading and presentation.

3. Usage

The skeletonization procedure implemented is restricted to 320*240 sized images, but can be modified to fit other sizes easily. Just modify the `I_WIDTH` and `I_HEIGHT` defines in `Skeleton.h.`

To use the skeletonization function in Skeleton.h and Skeleton.cpp call:

```
EuclideanSkeleton(uchar inI[I_SIZE],uchar* outI);
```

Where `inI` is the input image array and `outI` is the output image array. Both arrays are the row ordered transform of the image matrix into an array. For the input image, 0 valued positions correspond to background and the rest correspond to foreground. For the output image (skeleton), 0 valued positions correspond to background and `128 (define SKELETON)` valued positions correspond to skeleton.

The `SkeletonExample` project produces a command line executable which takes the path to a binary image as input and shows both the input image and the skeleton in different windows. The calling convention is:

Skeletonexample.exe IMAGE_PATH.

4. References

- *W. H. Hesselink, J. B. T. M. Roerdink: Euclidean skeletons of digital image and volume data in linear time by integer medial axis transform. IEEE Trans. Pattern Anal. Machine Intell. 30 (2008) 2204--2217.* –This is the base article of the implementation described here.

- http://opencv.willowgarage.com/wiki/ --Open Computer Vision library (OpenCV).