Milena
Generic image processing library

LRDE

1 Introduction

Milena is a programming framework for discrete mathematical morphology written in C++. It is part of the Olena project which aims at building a scientific computation platform oriented towards image processing, image recognition and artificial vision.

Milena is designed with two major goals in mind:

- Be as simple as calling C routines for end users.

- Be modular enough to be extended with respect to algorithms and data structures.

2 Targeted audience

This library targets several audiences:

- *End users* of morphological tools who want to apply and assemble algorithms to solve image processing, pattern recognition or computer vision problems.

- *Designers* of morphological operators who build new algorithms by using constructs from their software framework (language, livraries, toolboxes, programs, etc.).

- *Providers* of data structures who are interested in extending their framework with new data types (images, values, structuring elements, etc.).

3 Key features

Olena is:

- **Generic**. If a morphological operator admis a general definition whatever the context (topology of the image, structuring element, etc.), then this algorithm should have a corresponding single implementation.

- **Close to theory**. Reading (and writing) algorithms should eventually become natural to scientists used to mathematical morphology notations.

- **Efficient** (with respect to run time speed and memory usage), when it is possible. Dedicated and efficient implementations of morphological algorithms for certain cases are known and should be selected whenever possible.

- **User-friendly**. Users should not have to address memory-related issues or deal with a program silently failing because of an arithmetic overflow. The tool should handle these situations, and help the user diagnose any problem.

- **Reliable**. Programming by contract helps debbuging user's programs. By default, a debug mode is enabled and check the data and access validity at runtime. Since Olena tends to be as static as possible, many static checks are also done at compile time.

- **Free**. Milena is *free* and *open-source*. It is released under GNU GPL V2.

4 Library content

4.1 Generic basic image types

Common basic image types are provided: 1-D, 2-D, 3-D images. A N-D image class is also available. These class are provided with a border in order to make them fast in algorithms using structural elements.

4.2 Morphers

Morphers are generic, composable and lightweight objects built on one or several images, that can be used as

- **mixins**: a morpher can add extra data (e.g. a neighborhood) or operations (e.g., an ordering on the values) to an image;

- **adapters**: e.g., a slice morpher can be used to view a slice of a 3-D image (spacemap) as a 2-D image (bitmap);

- **modifiers**: a morpher can add a mask to an image, to restrict its (iterable) domain;

- **lazy function applications**: a morpher can present an image seen through a function, either bijective or not;

- etc.

4.3 Generic image processing algorithms

- Morphological algorithms: dilation, erosion, watershed, leveling, etc;

- Influence zone;

- Labeling;

- etc.

4.4 Auxiliary tools

Since Olena is intended to *designer* of algorithms and *provider* of new data structures, various generic auxiliary tools are available in the library.

- Topologies (grid, graph, etc.);

- Points and delta-points;

- neighborhoods and windows;

- accumulators;

- etc.

5 Learn more

Olena's **official website**: `http://olena.lrde.epita.fr`
Olena's **Trac**: `http://trac.lrde.org/olena`
Milena's **documentation**: `http://www.lrde.epita.fr/dload/doc/milena-1.0`

Mailing lists:

- olena@lrde.epita.fr - Question and comments;

- olena-bug@lrde.epita.fr - Bug reports;

- olena-patches@lrde.epita.fr - Patches.

Contacts:

- Thierry Geraud - thierry.geraud@lrde.epita.fr

- Roland Levillain - roland.levillain@lrde.epita.fr

Copyright