Writing Reusable Digital Geometry Algorithms in a Generic Image Processing Framework

Roland Levillain[1,2], Thierry Géraud[1,2], Laurent Najman[2]

[1] EPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicêtre, France

[2] Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge (IGM),
Équipe A3SI, ESIEE Paris, Noisy-le-Grand Cedex, France

Workshop on Applications of Digital Geometry and Mathematical Morphology (WADGMM)

Istanbul, Turkey – August 22, 2010

Intent

Context

- Software tools for Digital Geometry (DG) and Mathematical Morphology (MM).
- Reusability, flexibility (and efficiency).

Observations

- Many software tools for DG and MM.
- But mostly specific (tied to a dimension, a data structure, etc.).
- Little or no reusability, due to a lack of genericity.

Why genericity matters

- A general mathematical algorithm → a single, generic code.
- Quickly experiment methods and data structures at low cost.

Genericity, Image Processing and Digital Geometry

- General idea: design algorithms free of any specific element.
- Use abstractions: concepts [Levillain et al., 2010].
- Application to Image Processing (IP): an Image $I : D \to V$.
- Example: `image2d<bool>`, a model of the Image concept:
  a 2D binary image on a regular grid ($D = \mathbb{Z}^2$, $V = \{\top, \bot\}$).
- Turning a mathematical definition of a morphological dilation:

$$\delta_B(I)(x) = \sup_{h \in B} I(x + h)$$

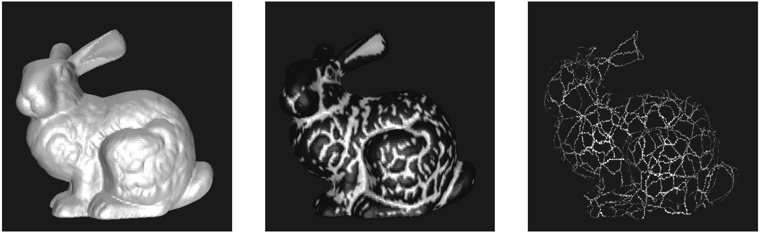
into a generic algorithm [Levillain et al., 2009]:

```
for_all(p) {
  sup = input(p);
  for_all(q)
    sup.take(input(q));
  output(p) = sup;
}
```

Remark: no implementation detail specific to an image type.
⇒ Works on all compatible images!

Applications and Conclusions

- DG and MM algorithms translate easily to generic programs.
- E.g. skeletonization by thinning based on the removal of simple points [Bertrand and Couprie, 2007].



Epilogue: Think Generic!

- Software should be designed with the ability to grow in mind.
- Abstraction may have a cost, but retaining efficiency is possible.
- Our work is available through the Olena project [LRDE, 2009].

Bibliography I

Bertrand, G. and Couprie, M. (2007).
Transformations topologiques discrètes.
In Coeurjolly, D., Montanvert, A., and Chassery, J.-M., editors, *Géométrie discrète et images numériques*, chapter 8, pages 187–209. Hermes Sciences Publications.

Levillain, R., Géraud, Th., and Najman, L. (2009).
Milena: Write generic morphological algorithms once, run on many kinds of images.
In Springer-Verlag, editor, *Proceedings of the Ninth International Symposium on Mathematical Morphology (ISMM)*, Lecture Notes in Computer Science Series, pages 295–306, Groningen, The Netherlands.

Bibliography II

- Levillain, R., Géraud, Th., and Najman, L. (2010).
  Why and how to design a generic and efficient image processing framework: The case of the Milena library.
  In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong.
  Accepted.

- LRDE (2009).
  The Olena image processing library.
  `http://olena.lrde.epita.fr`.