A 1ST PARALLEL ALGORITHM TO COMPUTE THE
MORPHOLOGICAL TREE OF SHAPES OF nD IMAGES
Sébastien Crozet  Thierry Géraud*
EPITA Research and Development Laboratory (LRDE), France
thierry.geraud@lrde.epita.fr
* also with Université Paris-Est, Laboratoire d’Informatique Gaspard-Monge (LIGM), Équipe A3SI, ESIEE Paris, France

The Tree of Shapes [1,4] as a Versatile Tool

Problem statement
• tree of shapes = self-dual morphological tree-based image representation
• a quasi-linear algorithm exists [5], yet it is sequential

Why is it interesting
• tree = easy structure to deal with
• nice properties: invariance to contrast changes and inversion
• numerous and powerful applications (see the banner above)

What our solution achieves
• a 1st parallel version of the quasi-linear algorithm, and ready for nD
• increasing size of data to process  no problème :)

What follows from our solution
• soon, processing 3D images with powerful self-dual morphological tools...

At a Glance

Parallel Sort NEW!

procedure PARALLELSORT(\(Q, Q^\text{ord}, \lambda, \text{ord}\))
\(Q[\lambda] \leftarrow p_{\text{w}}\)
while any queue of \(Q\) is not empty do
while \(Q[\lambda]\) is not empty do
\(p \leftarrow \text{Pop}(Q[\lambda]),\) \(Q^\text{ord}(p) \leftarrow \text{ord}\)
for all \(n \in N_n(p)\) that has not been visited yet do
if \(\lambda \in F(n)\) then \(\text{Push}(Q[\lambda], n)\)
else if \(\lambda < \text{min}(F(n))\) then \(\text{Push}(Q[\lambda \text{max}(F(n))], n)\)
end if
end for
end while
\(\text{ord} \leftarrow \text{ord} + 1\)
\(S^\lambda \leftarrow Q[0..\lambda]\)
\(S^\lambda \leftarrow Q[\lambda..\text{max value}]\)
\(\lambda' \leftarrow \text{highest level having faces on } S^\lambda\)
Run PARALLELSORT(\(F, S^\lambda, Q^\text{ord}(\lambda'), \text{ord}\)) on another thread.
\(\triangleright \) This thread continues with \(S^\lambda\)
end while
Wait for all child processes.
end procedure

An image and its tree of shapes. The nodes \(O\) and \(A\) have already been visited. The hierarchical queue contains the interior contour of \(B\) and \(C\). It is partitioned in two sets \(S^\lambda = \partial B\) and \(S^\lambda = \partial C\).

Reproducible Research
Evangelization from the Church of Mathematical Morphology

our C++ image processing library “Milena” → http://olena.lrde.epita.fr
full source code of our method → http://publis.lrde.epita.fr/crozet.14.icip

Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FLLT</th>
<th>FLST</th>
<th>Géraud et al.</th>
<th>this proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>s</td>
</tr>
</tbody>
</table>

Computation times (in seconds) on a classical image test set of the following algorithms: FLLT [1], FLST [4], Géraud et al. [5], and this paper proposal.

Bibliography