A First 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


All those results are from Yongchao Xu: http://www.lrde.epita.fr/wiki/User:Xu

At a Glance

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

Why is 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 problemo :)

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

Algorithmic Scheme of the Sequential Version [5]

function \( \text{ComputeTree}(f, p) \)
\( F \leftarrow \text{IMMERSE}(f) \)
\( (R, F') \leftarrow \text{SORT}(F, p) \)
\( \text{par} \leftarrow \text{UNIONFIND}(\text{reverse}(R)) \)
return \( \text{CANONICALIZE}(\text{par}, R, F') \)
end function

Algorithmic Scheme of the Parallel Version NEW!

function \( \text{ComputeTree}(f, p) \)
\( F \leftarrow \text{PARALLELIMMERSE}(f) \)  \( \triangleright \text{trivial} \)
\( \lambda \leftarrow \text{mean}(F(p)) \)
\( Q[\lambda] \leftarrow p \)
\( F^{\text{ord}} \leftarrow \text{PARALLELSORT}(F, Q, \lambda, 0) \)
\( \text{par} \leftarrow \text{PARALLELMAXTREE}(F^{\text{ord}}) \)  \( \triangleright \text{see [2] and [3]} \)
return \( \text{CANONICALIZE}(\text{par}, F^{\text{ord}}) \)
end function

Parallel Sort NEW!

procedure \( \text{PARALLELSORT}(F, Q, F^{\text{ord}}, \lambda, \text{ord}) \)
\( Q[\lambda] \leftarrow p \)
while any queue of \( Q \) is not empty do
\( p \leftarrow \text{Pop}(Q[\lambda]) \)
\( F^{\text{ord}}(p) \leftarrow \text{ord} \)
for all \( n \in N(p) \) 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 \( \text{Push}(Q[\lambda], n) \)
else \( \text{Push}(Q[\text{max}(F(n))], n) \)
end if
end for
end while
\( \text{ord} \leftarrow \text{ord} + 1 \)
\( S^\lambda_{\text{ord}} \leftarrow Q[\lambda..\lambda] \)
\( \lambda' \leftarrow \text{highest level having faces on } S^\lambda_{\text{ord}} \)
\( \lambda' \leftarrow \text{smallest level having faces on } S^\lambda_{\text{ord}} \)
end if
\( Q \leftarrow S^\lambda_{\text{ord}} \)
end procedure

Example

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 \( S^A_{\text{ord}} = \partial B \) and \( S^C_{\text{ord}} = \partial C \).

Reproducible Research

Evangelization from the Church of Mathematical Morphology

our C++ image processing library “Milena” → http://olenalrde.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>Run time</td>
<td>0.12 s</td>
<td>0.03 s</td>
<td>0.01 s</td>
<td>0.005 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