Meaningful Disjoint Level Lines Selection
Yongchao Xu\textsuperscript{1,2}, Edwin Carlinet\textsuperscript{1,2}, Thierry Géraud\textsuperscript{3}, Laurent Najman\textsuperscript{2}
\textsuperscript{1}EPITA Research and Development Laboratory (LRDE), France
\textsuperscript{2}Université Paris-Est, Laboratoire d’Informatique Gaspard-Monge (LIGM), ESIEE Paris, France
\{firstname.lastname\}@lrde.epita.fr, l.najman@esiee.fr

Tree of shapes $T$ \cite{1,2}: a versatile tool for many applications

Shape filtering (ICPR 2012).
Energy-driven simplification (ICIP 2013).
Object segmentation (ICIP 2012).
Hierarchical image segmentation (ISMM 2013).

These results are from the PhD work \cite{3} supervised by T. Géraud & L. Najman available in http://www.lrde.epita.fr/wiki/User:Xu

At a glance

Motivation
- Significant contours of objects $\Leftrightarrow$ segments of level lines \cite{1}
- Inclusion relationship $\Rightarrow$ tree of shapes $T$ \cite{2}: a versatile representation
- The knowledge of tree structure is fundamental for a deep tree analysis

Problem
- The number of shapes is about as large as the number of pixels

Objective
- Select a subset of level lines representing the main tree structure

Contribution
- An efficient algorithm for extracting meaningful and disjoint level lines
- A simplified image providing an intuitive idea about main tree structure

Basic idea

Select a subset of meaningful and disjoint level lines from the tree of shapes $T$ to represent the main tree structure; Two main ideas:
1. $\forall N \in T$, find its lowest ancestor shape $N'$: Smallest Enclosing Shape $\text{ses}(N)$, such that $N' \subseteq N$, $\partial N' \cap \partial N = \emptyset$.
2. $\forall N \in T$ in some Order, select $N$ if it is not deactivated by any descendant, and none of $[N \sim \text{ses}(N)]$ is selected, then deactivate $[N \sim \text{ses}(N)]$.

Algorithm overview: three main steps
1. Tree of shapes construction: use the union-find-based algorithm in \cite{4} to compute the set of all level lines.
2. SES computation: bottom-up traversal updating based on the nodes’ depth
3. Level lines selection: sequential test based on the status of $[N \sim \text{ses}(N)]$

Smallest Enclosing Shape (SES) computation

The algorithm in \cite{4} works on Khalimsky grid $K_d$. A shape is represented by a 2-face; $\text{parent}$: inclusion relationship; $\text{getCanonical}$: canonical element.

\begin{verbatim}
COMPUTE_SSES(parent, S, depth)
foreach x in K_d do SES(x) ← getCanonical(x)
foreach 2-face x in reverse order of S do
  if depth(x) < depth(SES(x)) then
    SES(x) ← getCanonical(x)
  q ← parent(x);
  if depth(SES(x)) < depth(SES(q)) then
    SES(q) ← SES(x)
return SES
\end{verbatim}

Basic idea

Final disjoint level lines $S'$ selection

\begin{verbatim}
SELECT_LEVEL_LINES(parent, SES, Order)
foreach x in K_d do status(x) ← Null;
S' ← \emptyset;
foreach canonical element x in Order do
  if status(x) ≠ Unactive then
    y ← parent(x);
    while y ≠ SES(x) and status(y) ≠ Active do y ← parent(y);
    if y = SES(x) then
      status(x) ← Active;
      S' ← S' ∪ \{x\};
    y ← parent(x);
    while y ≠ SES(x) do status(y) ← Unactive; y ← parent(y);
    return S'
\end{verbatim}

Comparison with different selection orders

References