Instituto de Matemática Pura e Aplicada

Robust Adaptive Polygonal Approximation of Implicit Curves

Hélio Lopes (PUC-Rio)

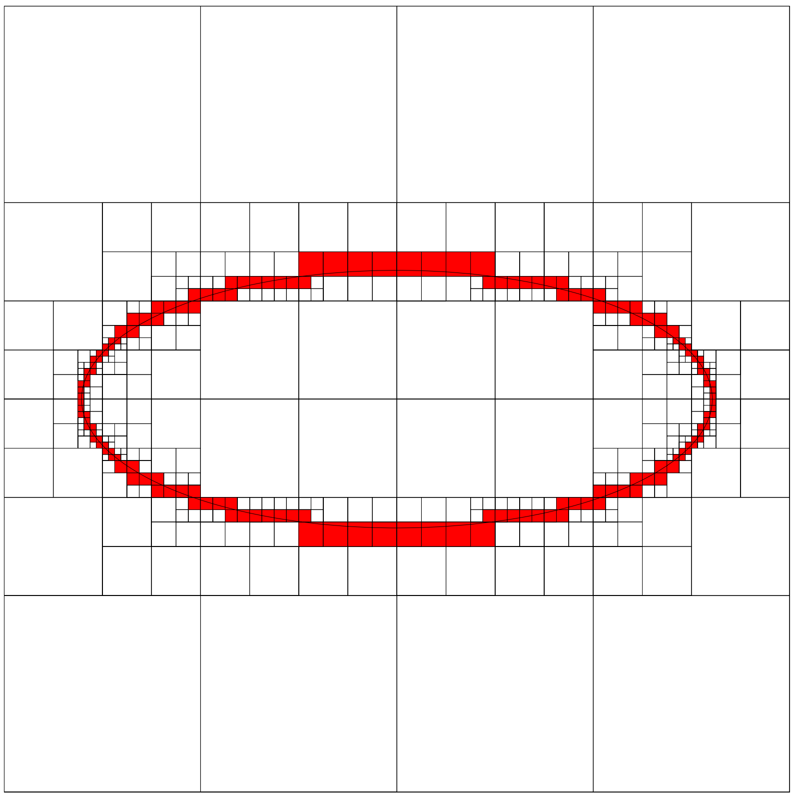
João Batista Oliveira (PUC-RS)

Luiz Henrique de Figueiredo (IMPA)

Seminário de Computação Gráfica

The Problem

Given $f\colon \Omega \subseteq \mathbf{R}^2 \to \mathbf{R}$, compute adaptive polygonal approximation of the curve given implicitly by f: $\mathcal{C} = \{(x, y) \in \mathbf{R}^2 : f(x, y) = 0\}$.

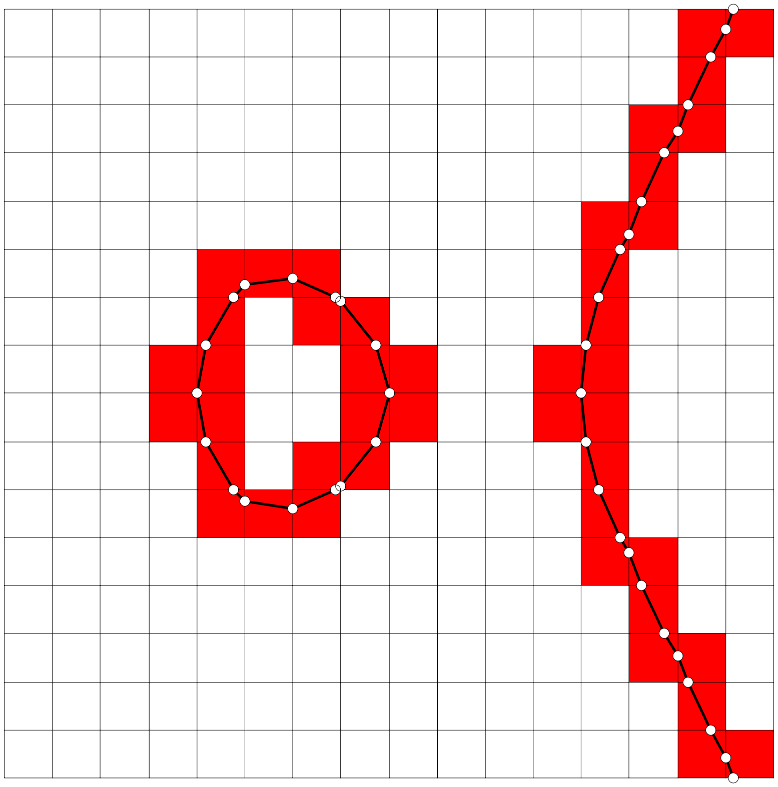


$$f(x, y) = \frac{x^2}{6} + y^2 - 1$$

Goal: Spatial *and* geometric adaption.

A Solution: Enumeration

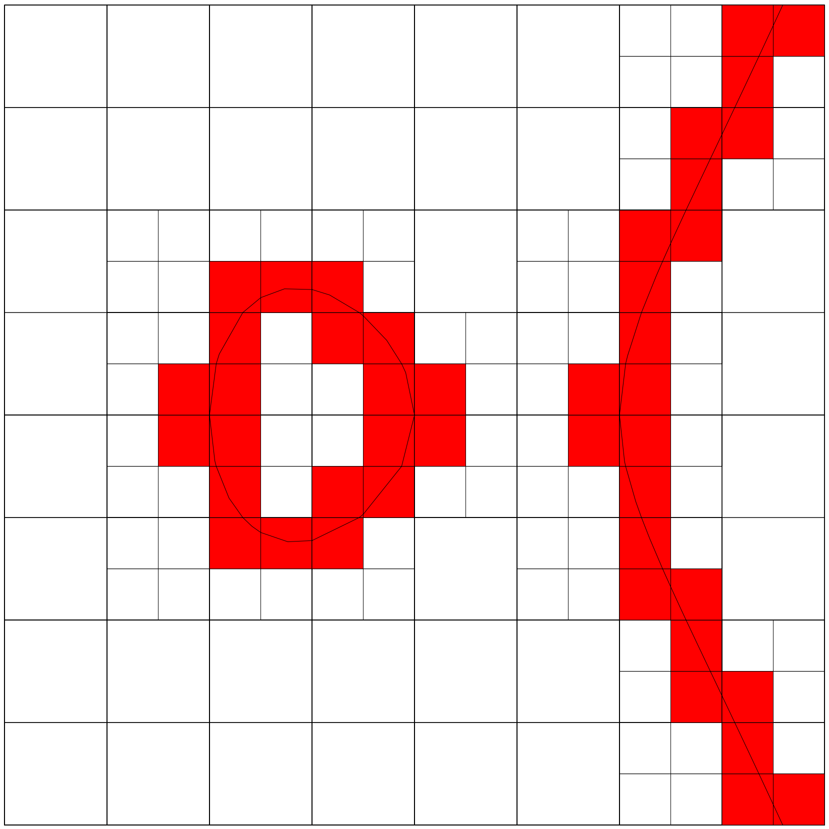
- Decompose Ω into grid of small cells — How small?
- Locate $\mathcal{C}$ by identifying cells that intersect $\mathcal{C}$ — What criteria?



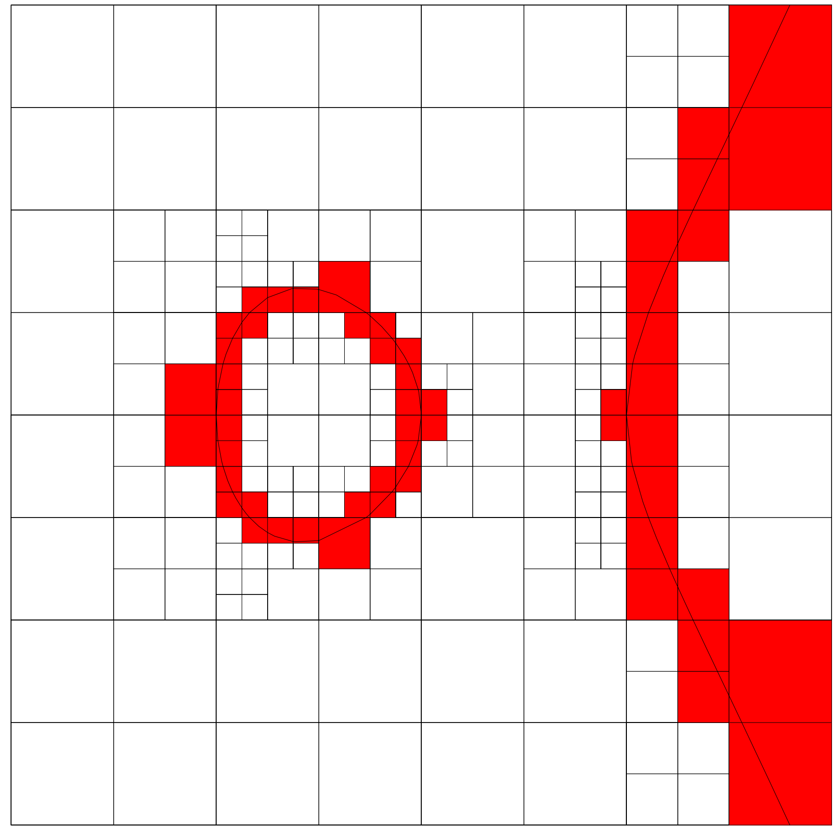
$$f(x,y) = y^2 - x^3 + x$$

Full enumeration is expensive and not robust.

Adaptive Enumeration



Spatial adaption



Geometrical adaption

The Tools

- Oracles
  - ◇ Is this cell away from the curve?
  - ◇ Is the curve approximately flat inside the cell?

- Interval arithmetic
  - ◇ Robust interval estimates for f and ∇f

$$X \subseteq \Omega \Rightarrow F(X) \supseteq f(X) = \{f(x,y) : (x,y) \in X\}$$

  - ◇ $0 \notin F(X) \Rightarrow$ cell X is away from curve

- Automatic differentiation
  - ◇ Efficient gradient computation

Robust Adaptive Enumeration

- Recursive exploration of domain Ω starts with explore(Ω).
- Discard subregions X of Ω when $0 \notin F(X)$. This is a proof that X does not contain any part of the curve $\mathcal{C}$.

```
explore(X):
    if 0 ∉ F(X) then
        discard X
    elseif diam(X) < ε then
        output X
    else
        divide X into smaller pieces X_i
        for each i, explore(X_i)
```

- All output cells have the same size. Only spatial adaption.
- Not new: Suffern–Fackerell (1991), Snyder (1992).

Robust Adaptive Approximation

- Estimate curvature by gradient variation.
- $G =$ inclusion function for the normalized gradient of f.
- $G(X)$ small $\Rightarrow$ curve approximately flat inside X.

explore(X):
  if 0 $\notin F(X)$ then
    discard X
  elseif $\mathsf{diam}(X) < \varepsilon$ or $\mathsf{diam}(G(X)) < \delta$ then
    approx(X)
  else
    divide X into smaller pieces X_i
    for each i, explore(X_i)

- Output cells vary in size. Spatial and geometrical adaption.

Interval arithmetic

- Quantities represented by intervals:

$$x = [a, b] \Rightarrow x \in [a, b]$$

- Primitive operations:

$$\begin{aligned}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] \times [c, d] &= [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}] \\
[a, b] \;/\; [c, d] &= [a, b] \times [1/d, 1/c] \\
[a, b]^2 &= [0, \max(a^2, b^2)], \qquad a \le 0 \le b \\
&= [\min(a^2, b^2), \max(a^2, b^2)], \qquad \text{otherwise} \\
\exp\, [a, b] &= [\exp(a), \exp(b)].
\end{aligned}$$

- Automatic extensions:

$$x_i \in X_i \Rightarrow f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

- Several good implementations available in the Web

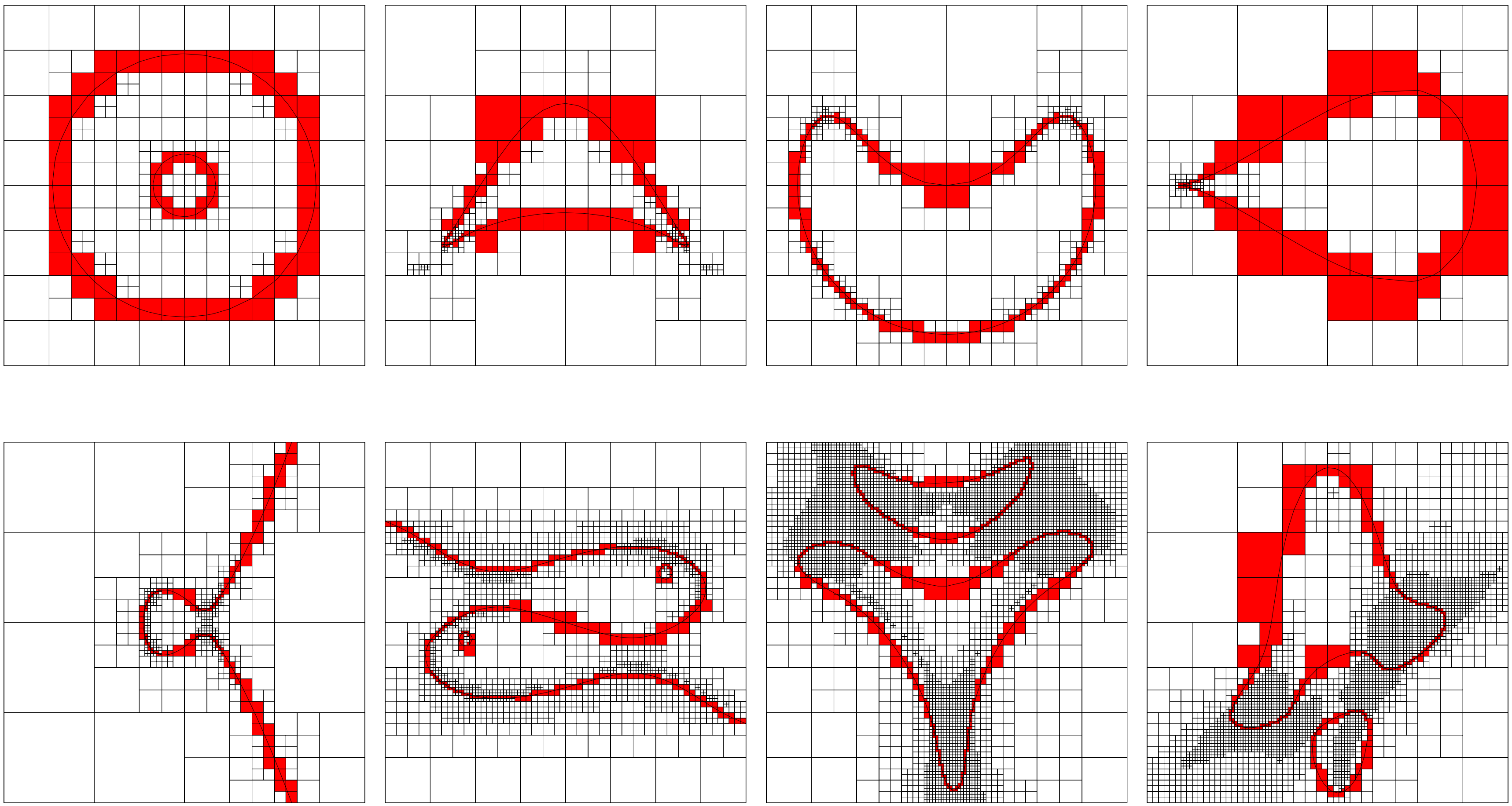
Automatic differentiation

- Symbolic differentiation: exact, complex, slow
- Numerical differentiation: approximate, ill-conditioned
- Automatic differentiation: exact, well-conditioned, fast
- Operates on tuples $(u_0, u_1, \ldots, u_n)$, $\quad u_i = \left.\frac{\partial u}{\partial x_i}\right|_{x_i = a_i}$

$$
\begin{aligned}
(u_0, u_1, u_2) + (v_0, v_1, v_2) &= (u_0 + v_0, u_1 + v_1, u_2 + v_2) \\
(u_0, u_1, u_2) \cdot (v_0, v_1, v_2) &= (u_0 v_0, u_0 v_1 + v_0 u_1, u_0 v_2 + v_0 u_2) \\
\sin(u_0, u_1, u_2) &= (\sin u_0, u_1 \cos u_0, u_2 \cos u_0) \\
\exp(u_0, u_1, u_2) &= (\exp u_0, u_1 \exp u_0, u_2 \exp u_0)
\end{aligned}
$$

- Automatic extensions
- Several good implementations available in the Web
- Operate with intervals and get estimates for gradient!

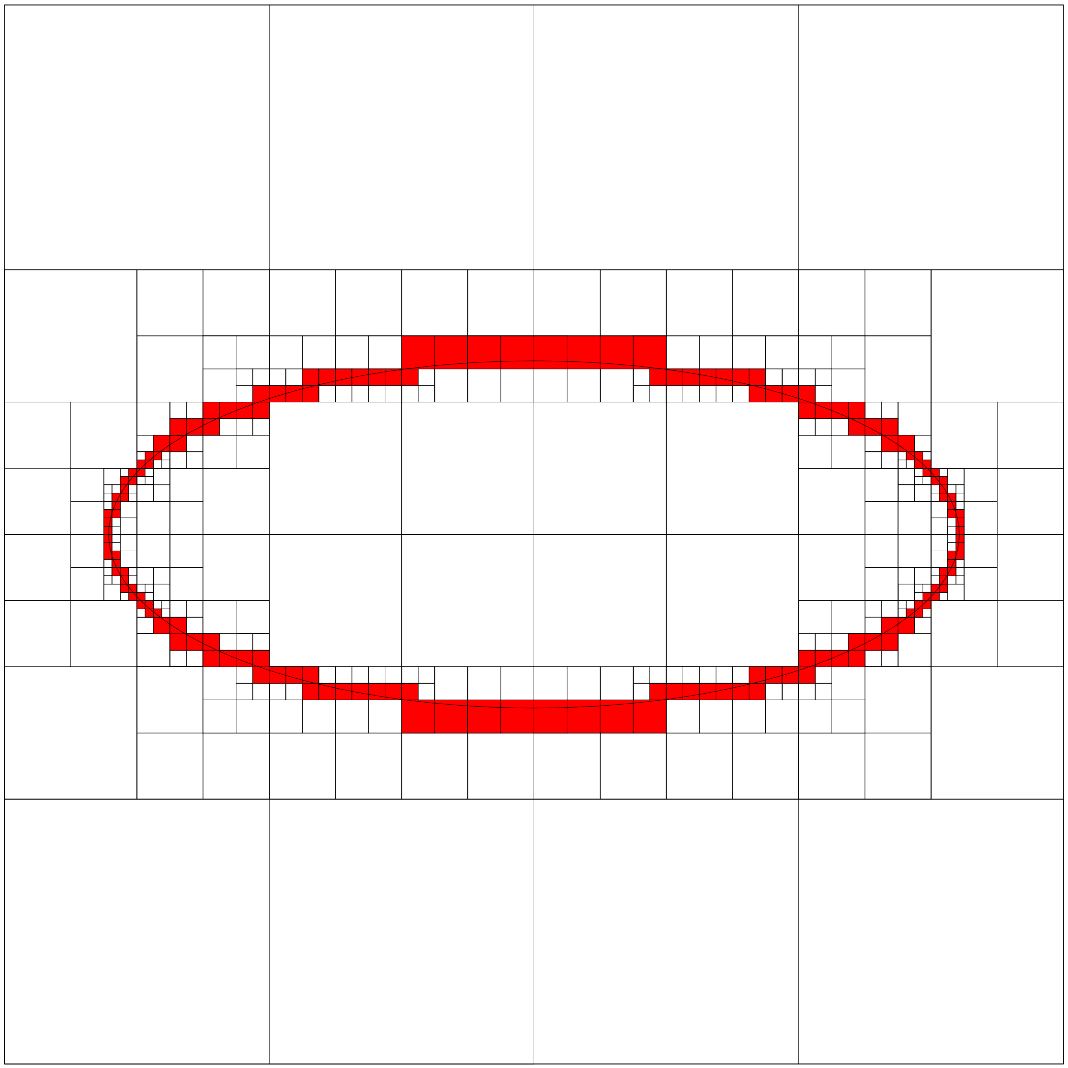
Results

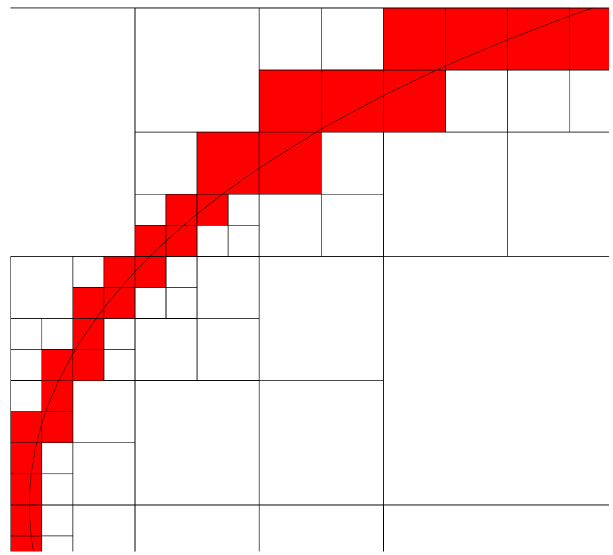


Large white cells = spatial adaption

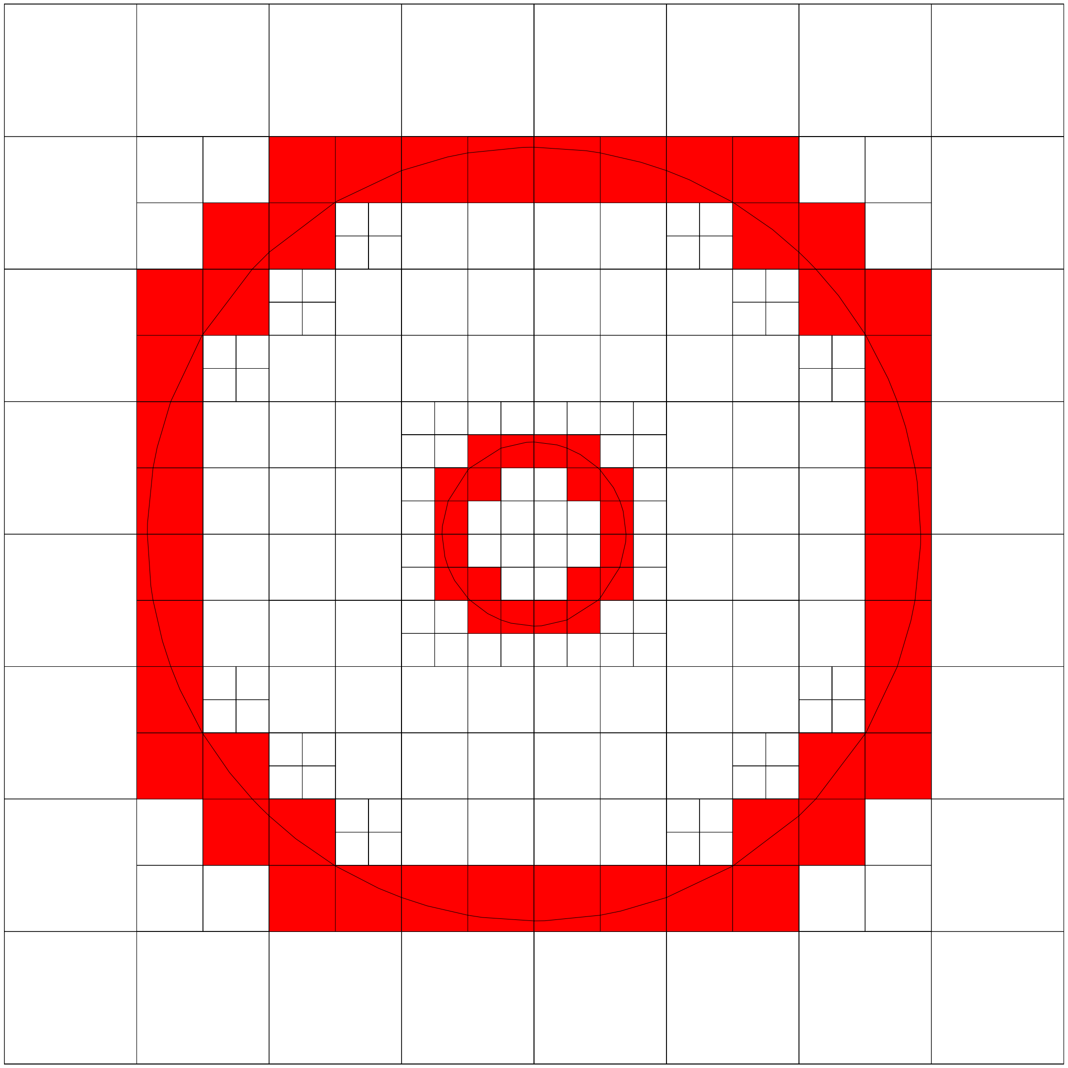
Large red cells = geometrical adaption

Results: Ellipse

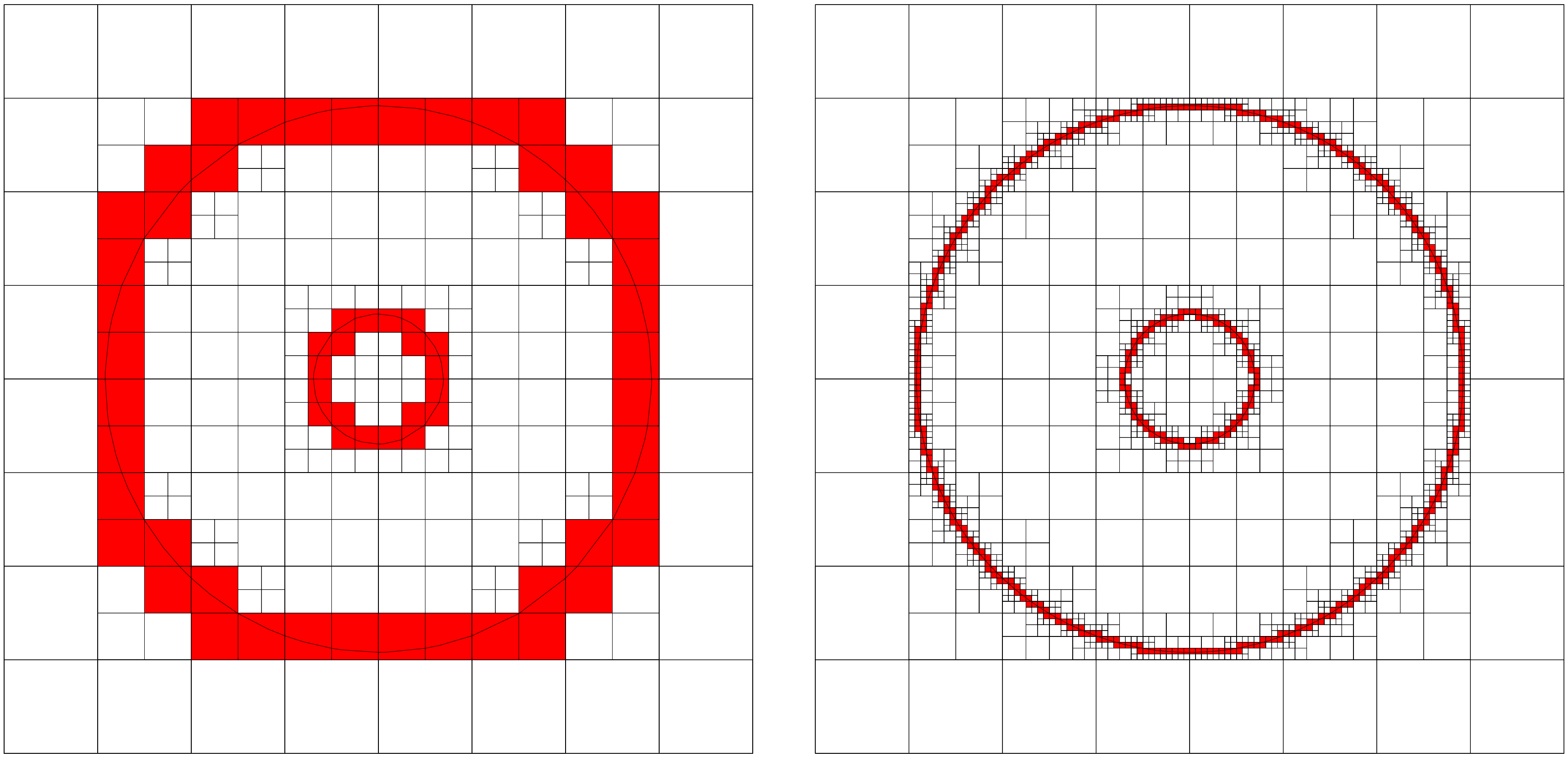




Results: Two circles



Results: Two circles

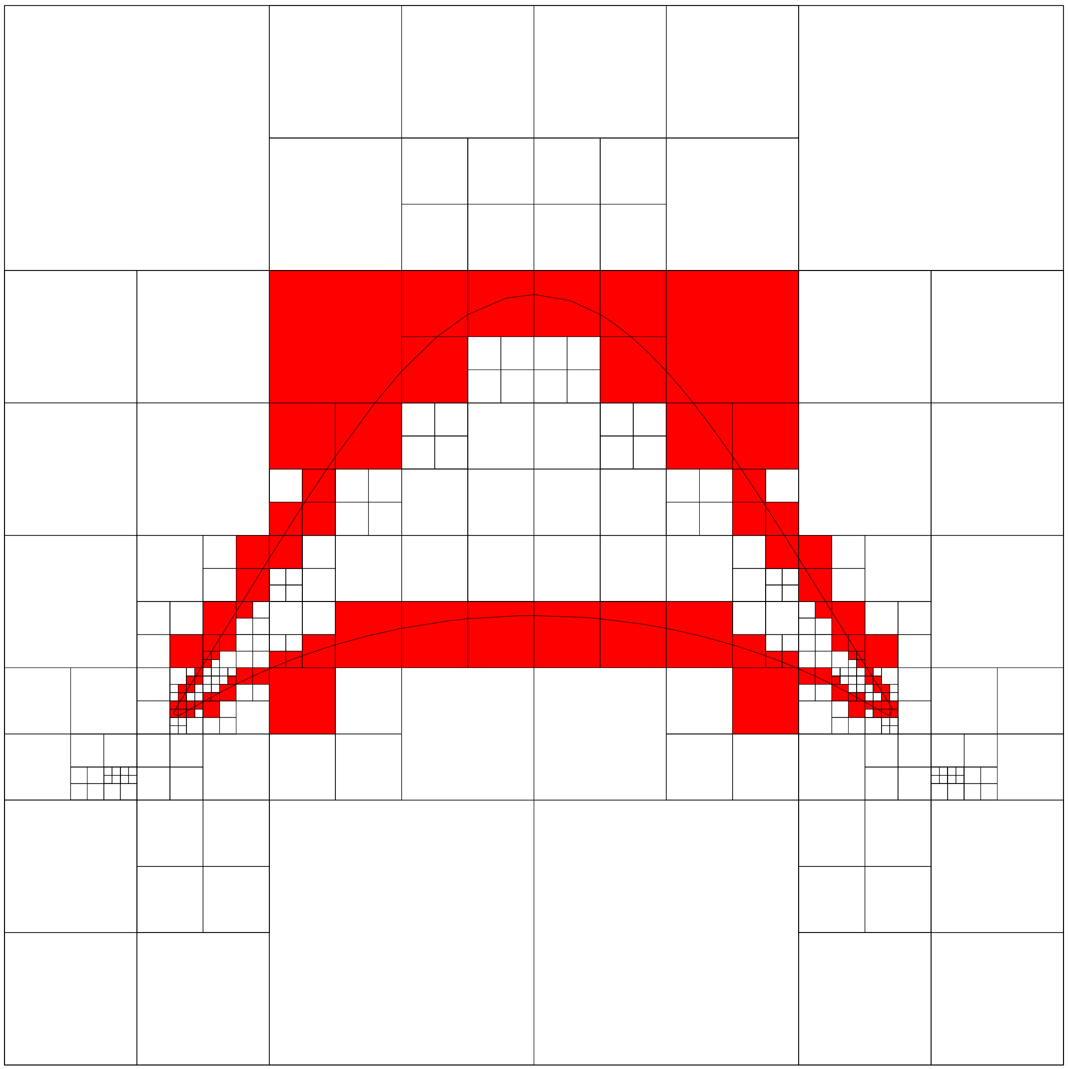


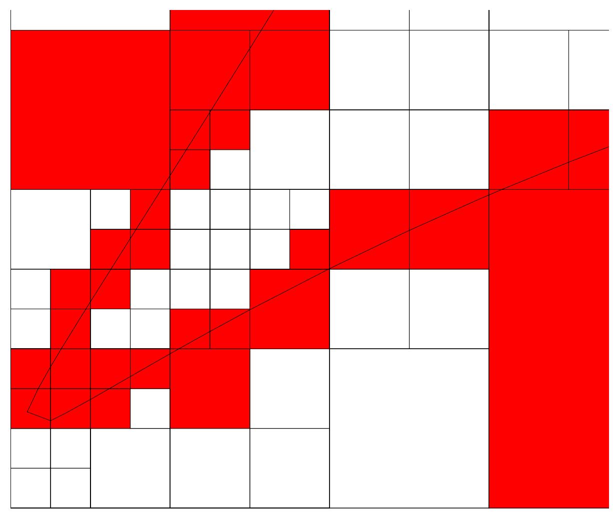
341 boxes, 64 leaves

2245 boxes, 464 leaves

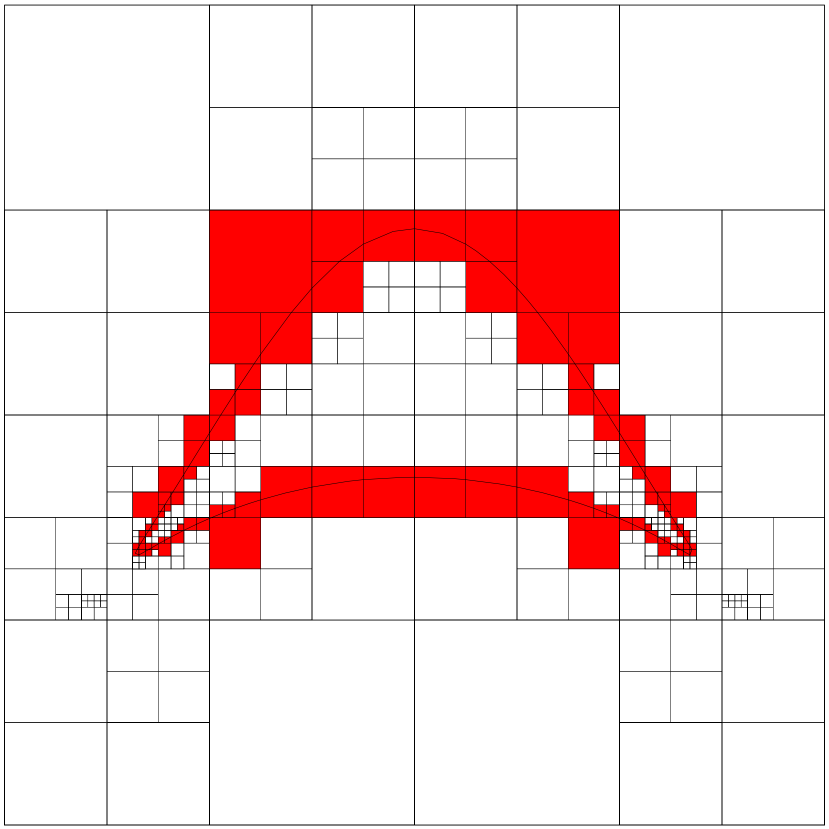
efficiency: 6.6 for boxes, 7.2 for leaves

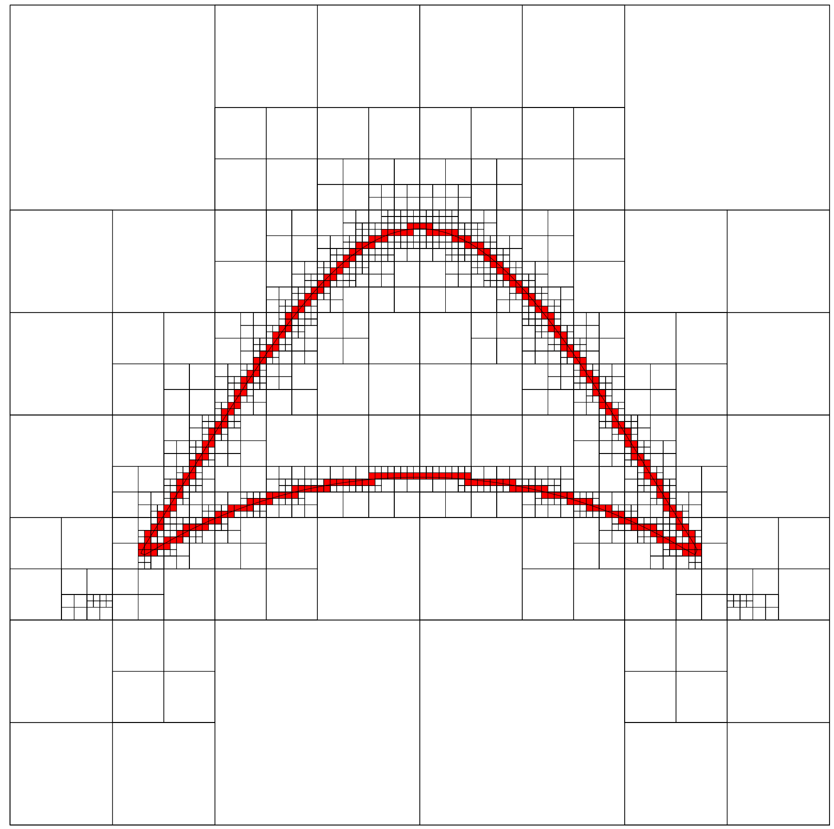
Results: Bicorn





Results: Bicorn



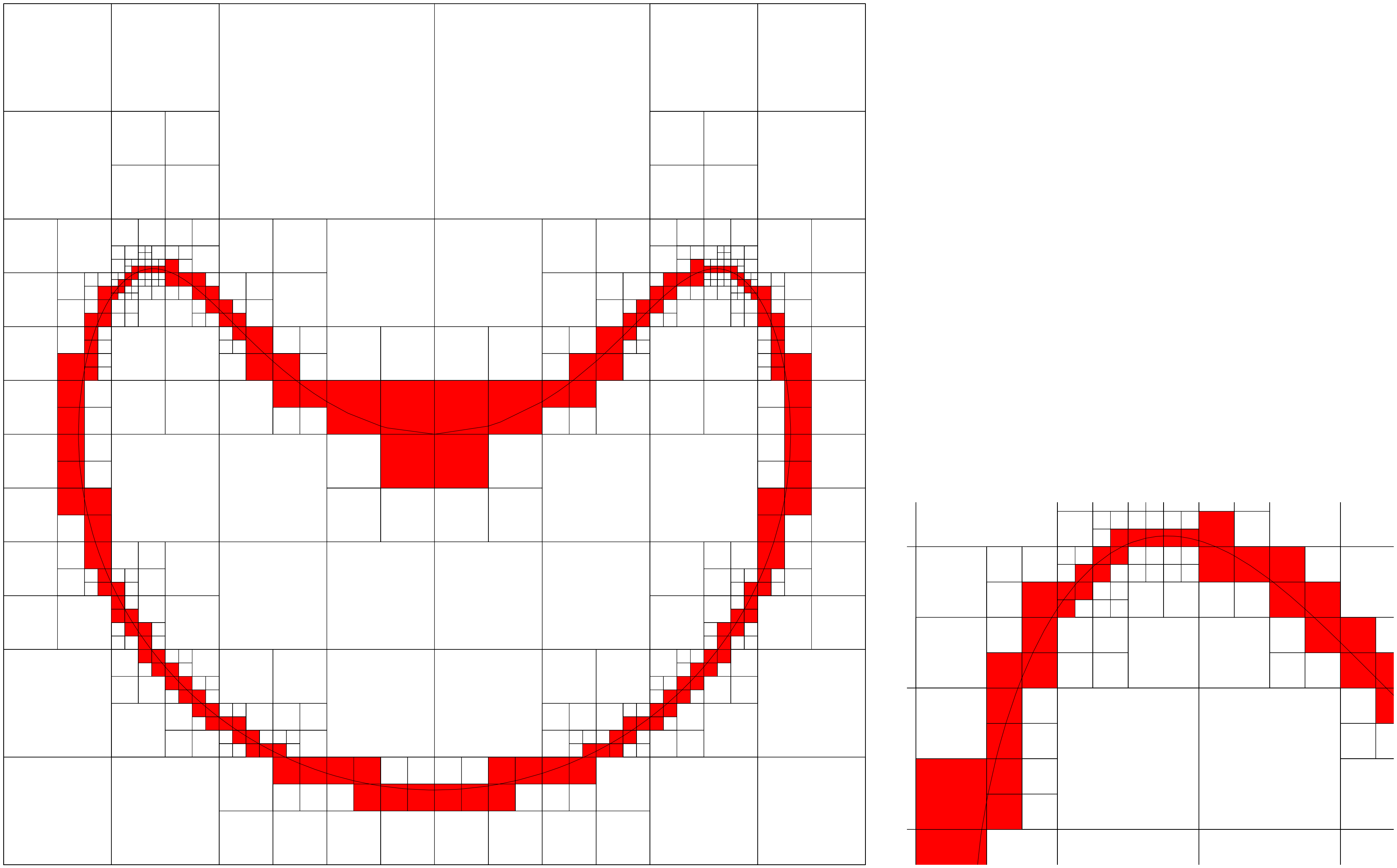


453 boxes, 94 leaves

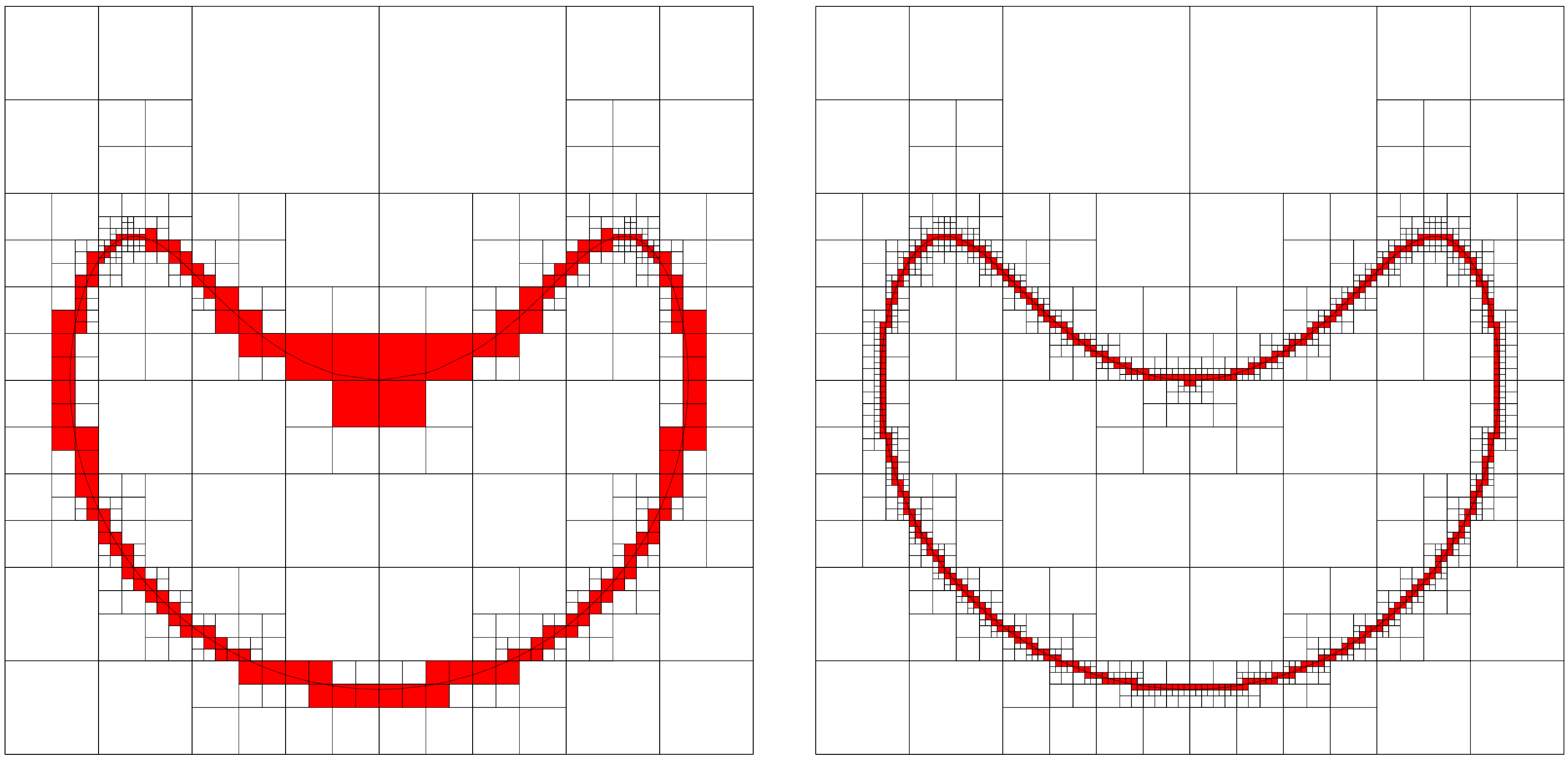
1717 boxes, 300 leaves

efficiency: 3.8 for boxes, 3.2 for leaves

Results: “Clown smile”



Results: “Clown smile”

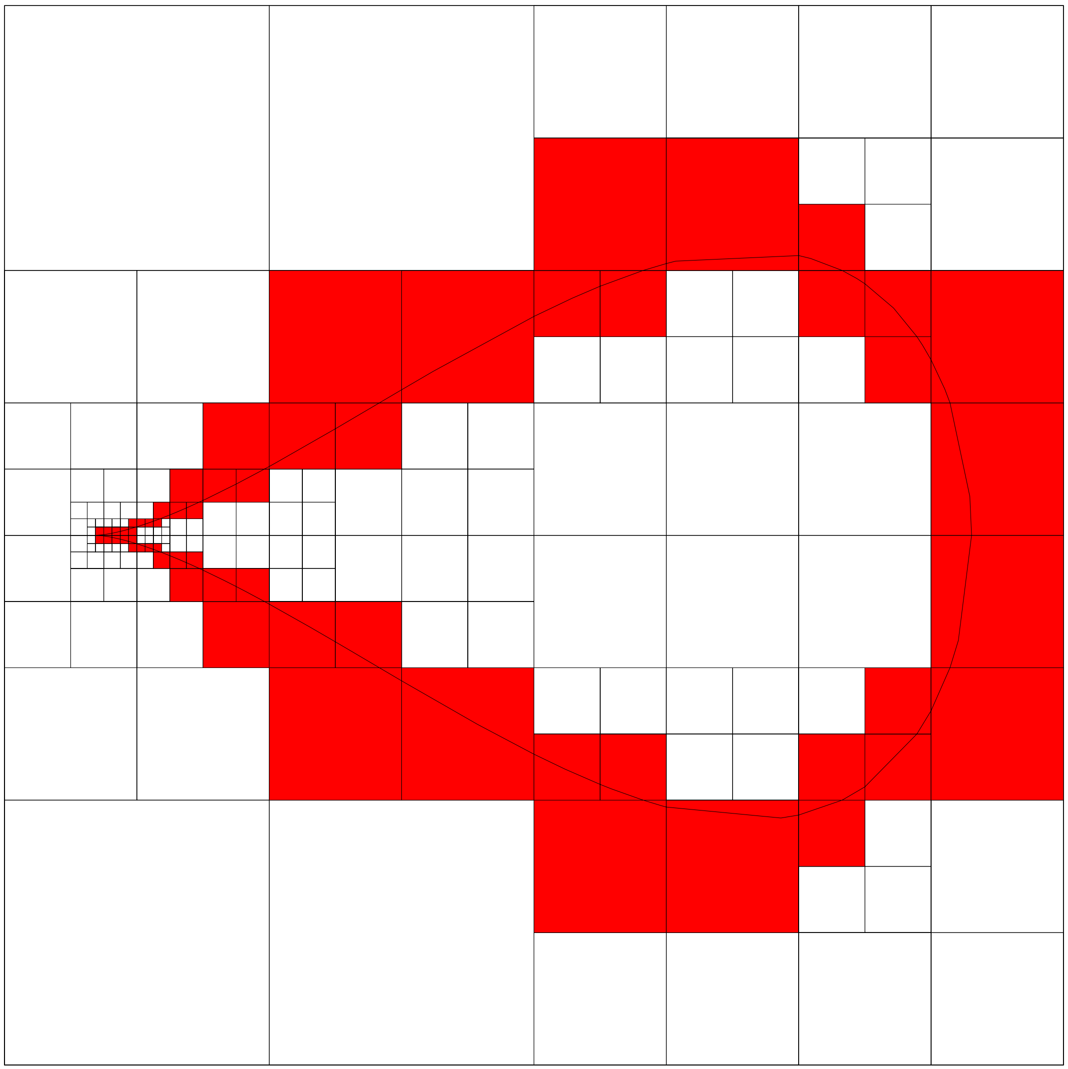


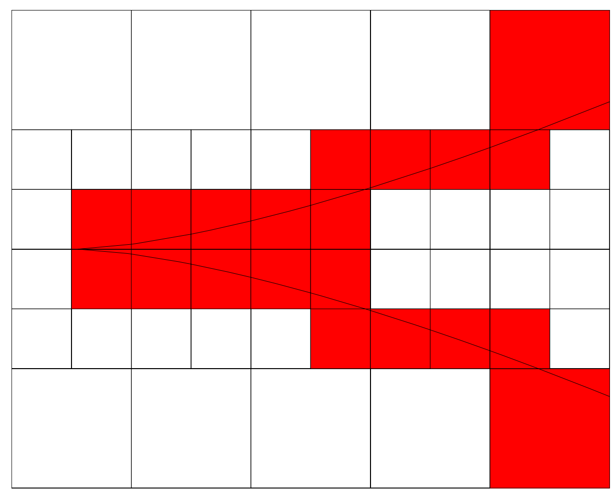
709 boxes, 164 leaves

1781 boxes, 414 leaves

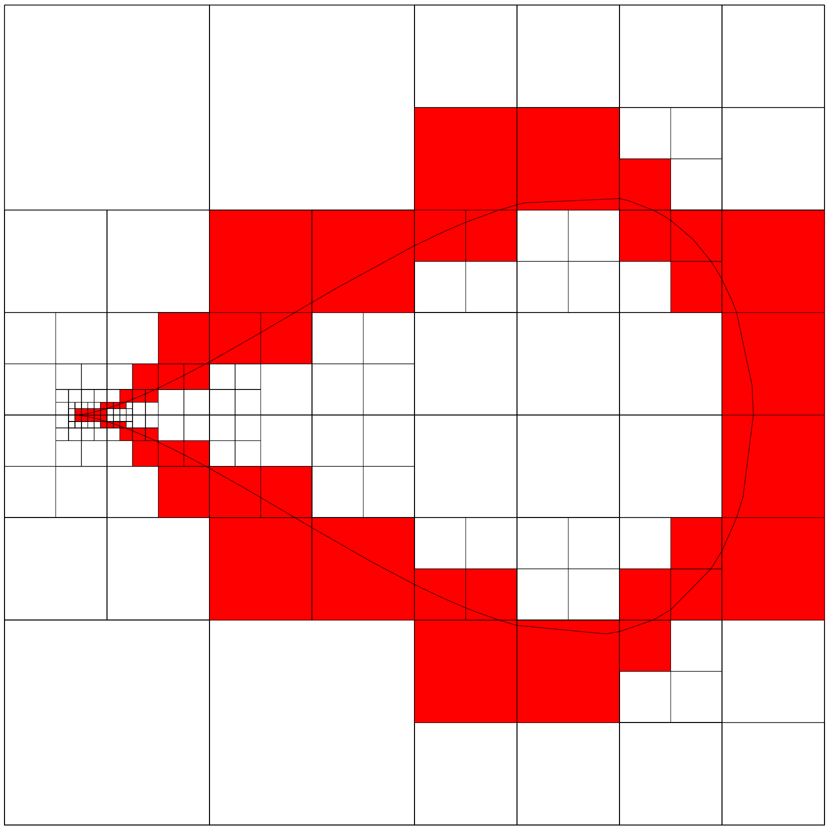
efficiency: 2.5 for boxes, 2.5 for leaves

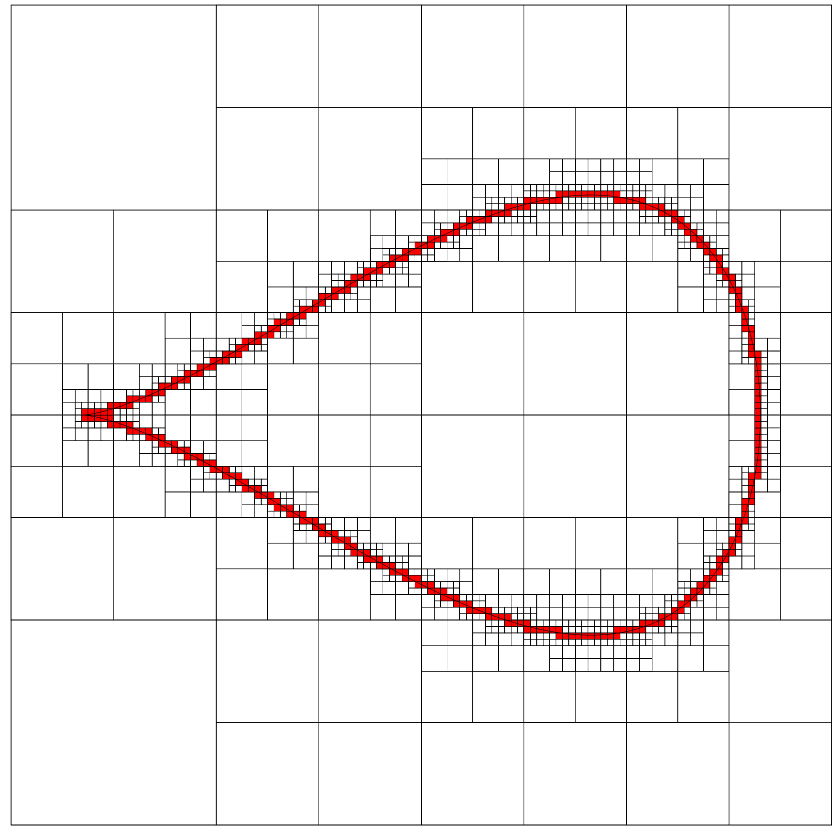
Results: Pear





Results: Pear



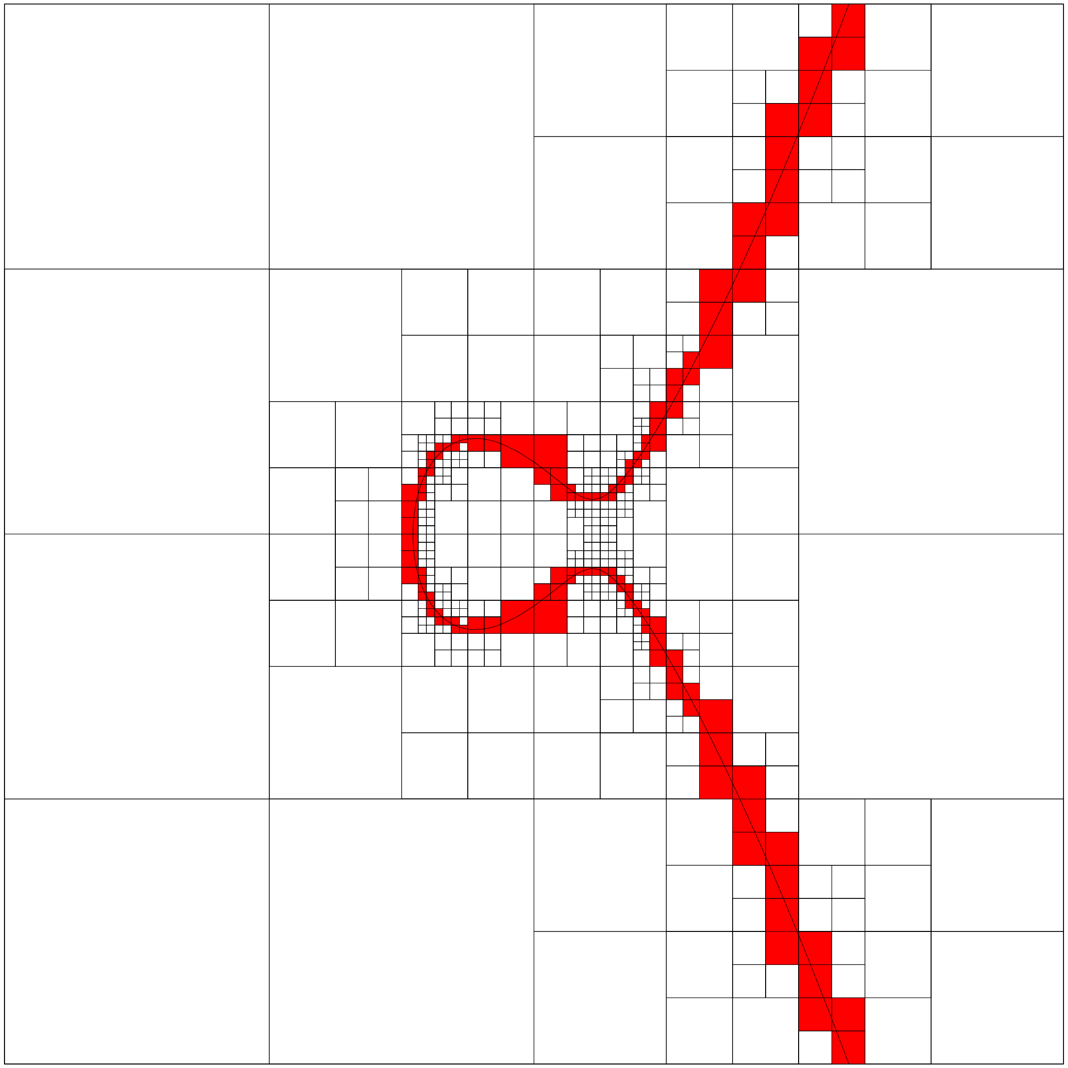


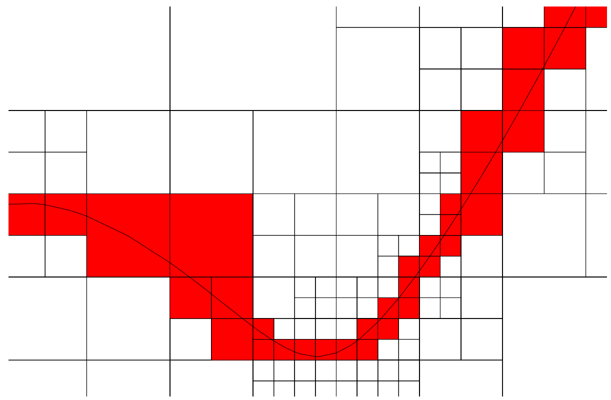
237 boxes, 60 leaves

1773 boxes, 348 leaves

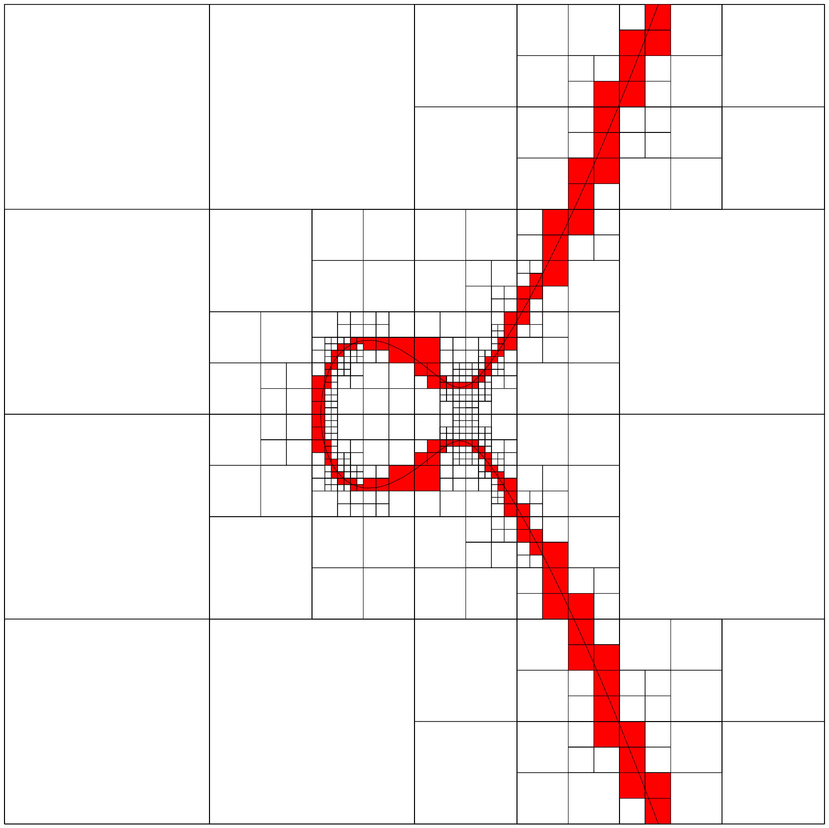
efficiency: 7.5 for boxes, 5.8 for leaves

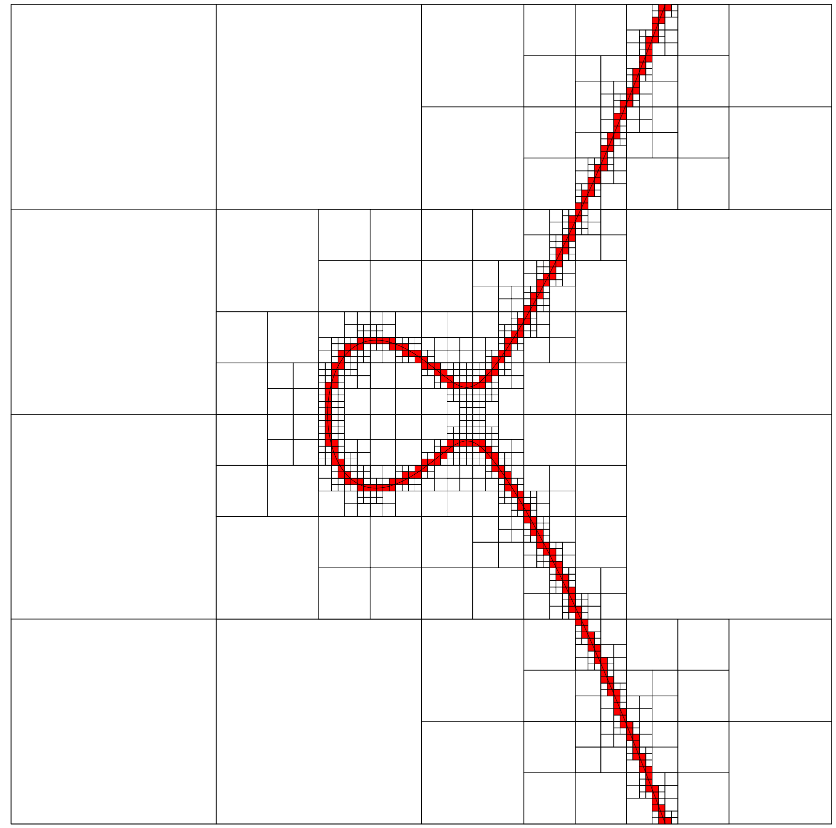
Results: Cubic





Results: Cubic



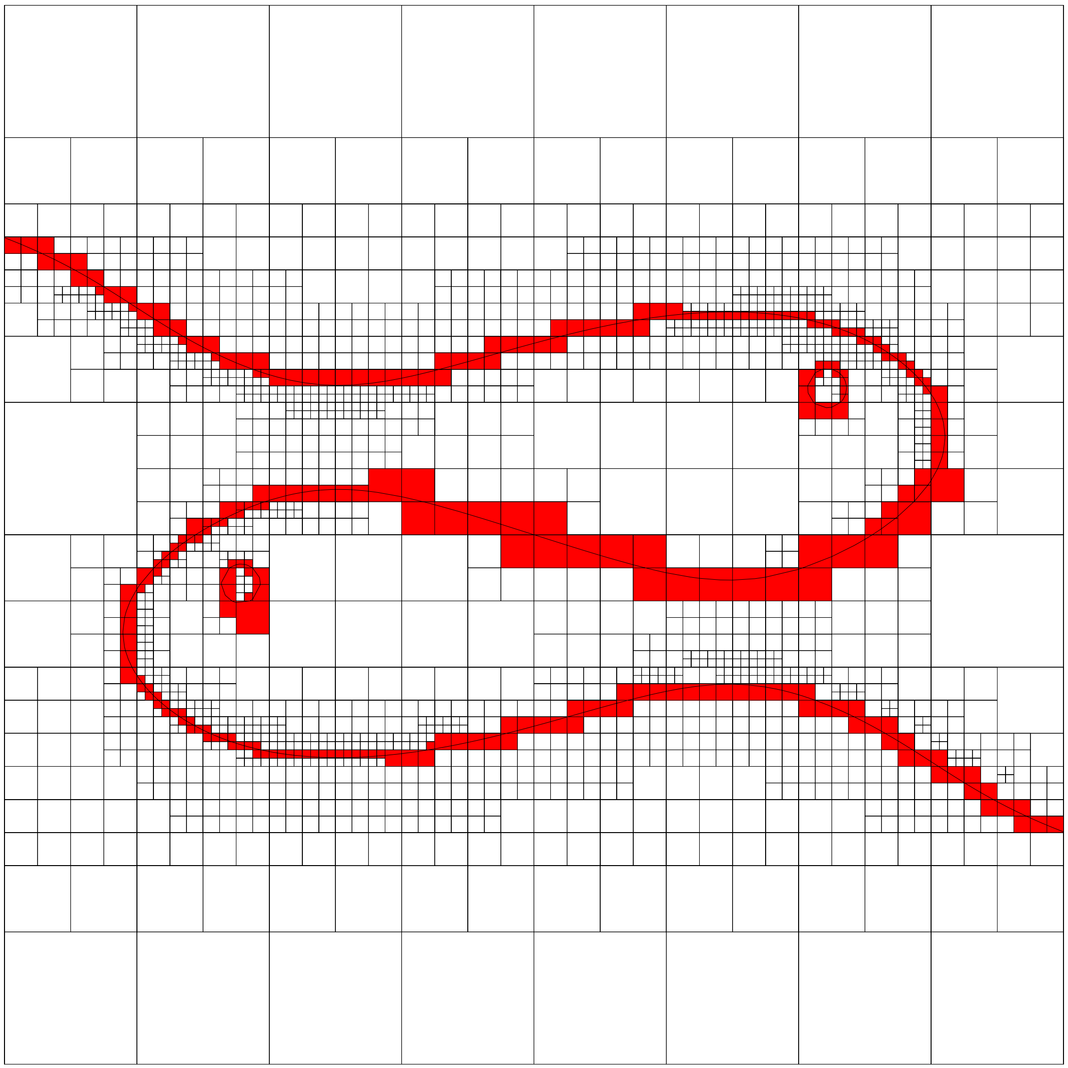


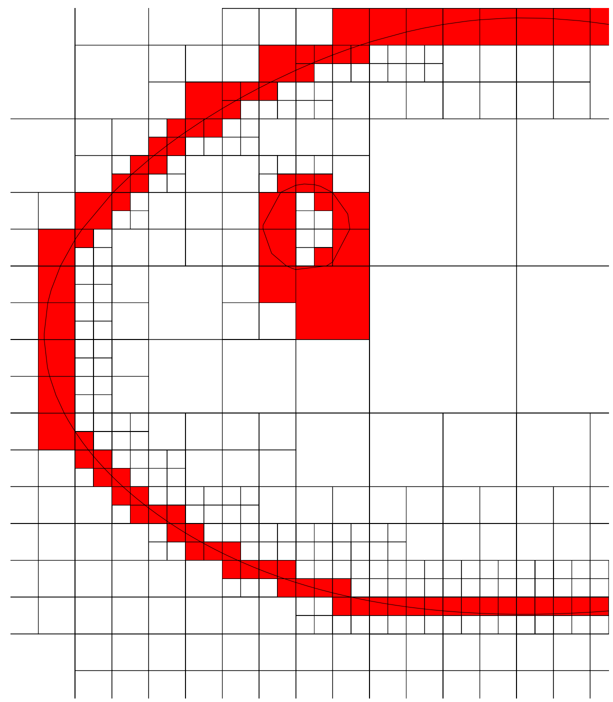
709 boxes, 128 leaves

1341 boxes, 262 leaves

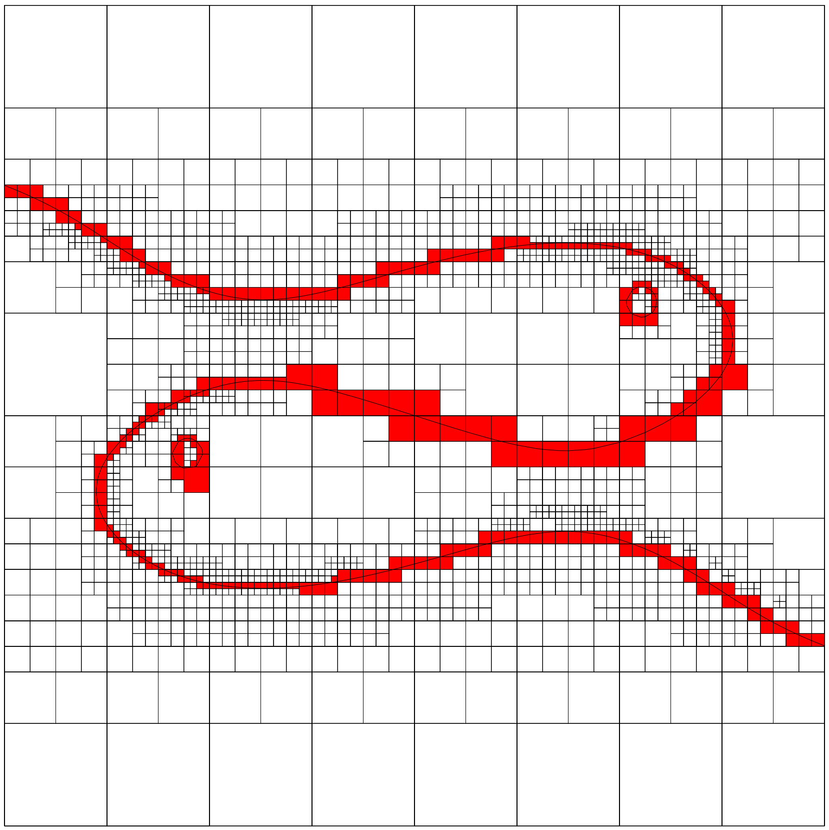
efficiency: 1.8 for boxes, 2.0 for leaves

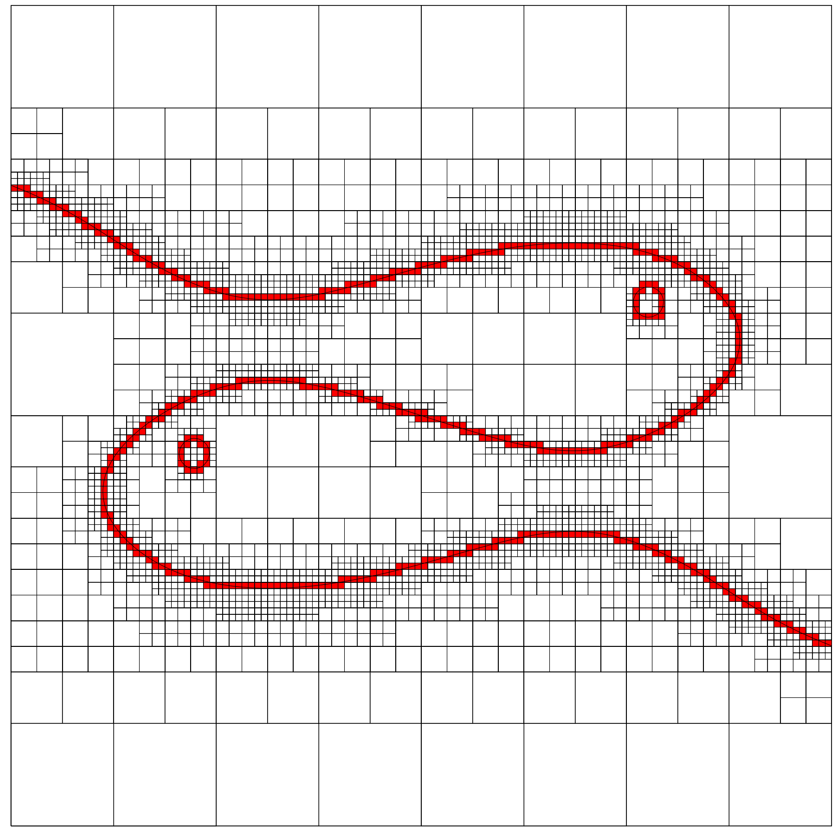
Results: Pisces logo





Results: Pisces logo



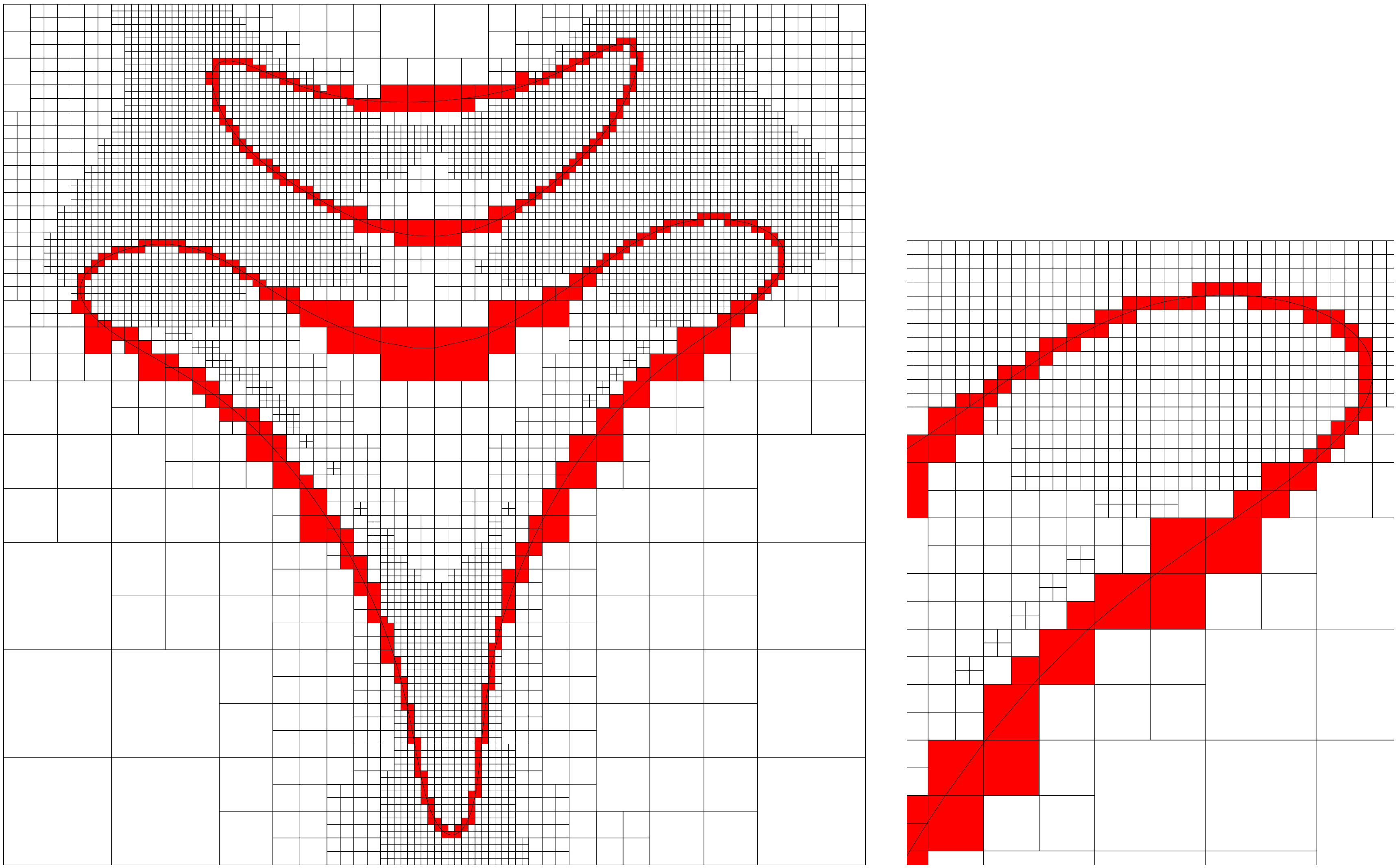


2621 boxes, 280 leaves

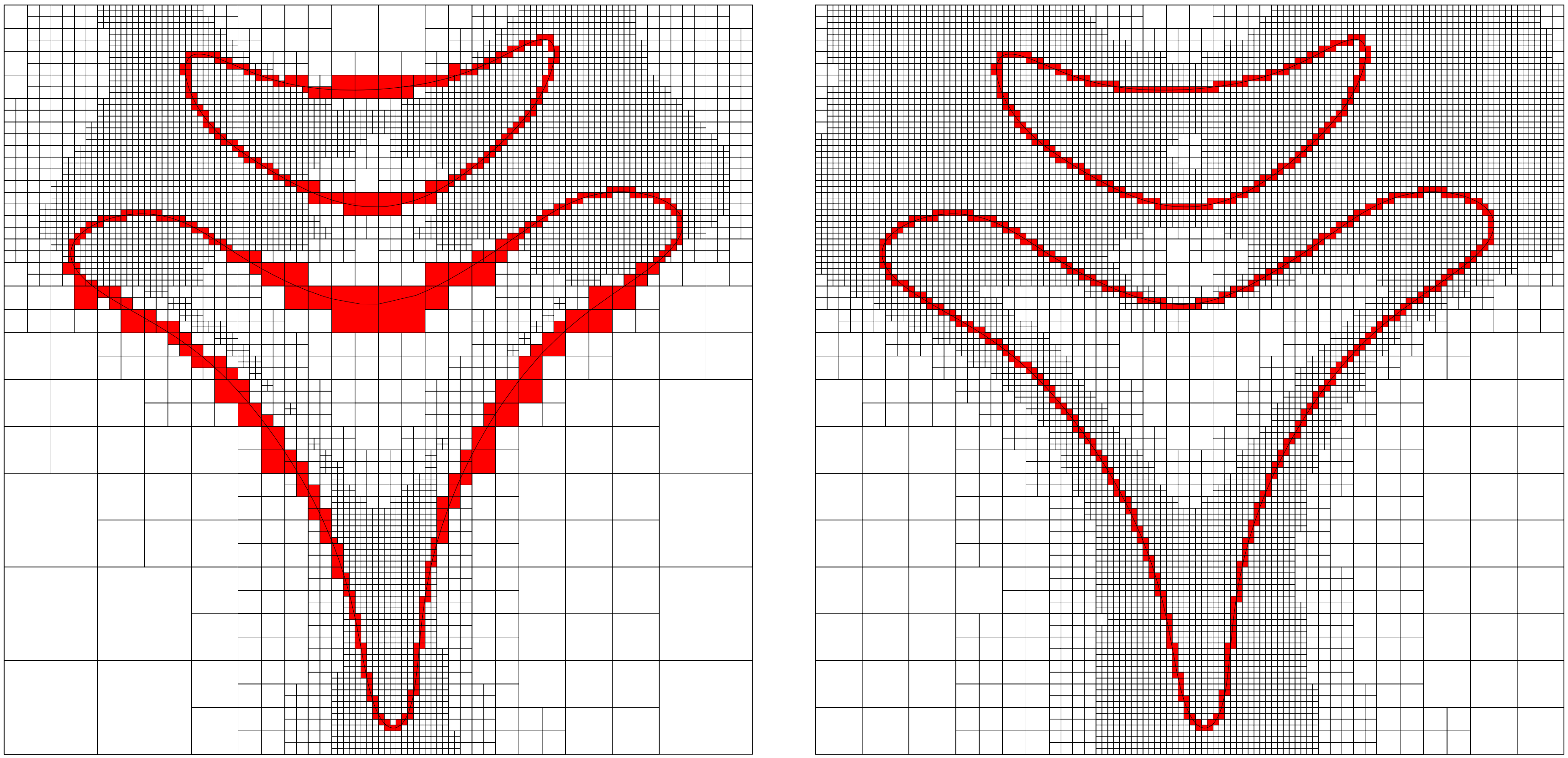
4477 boxes, 488 leaves

efficiency: 1.7 for boxes, 1.7 for leaves

Results: Mig outline



Results: Mig outline

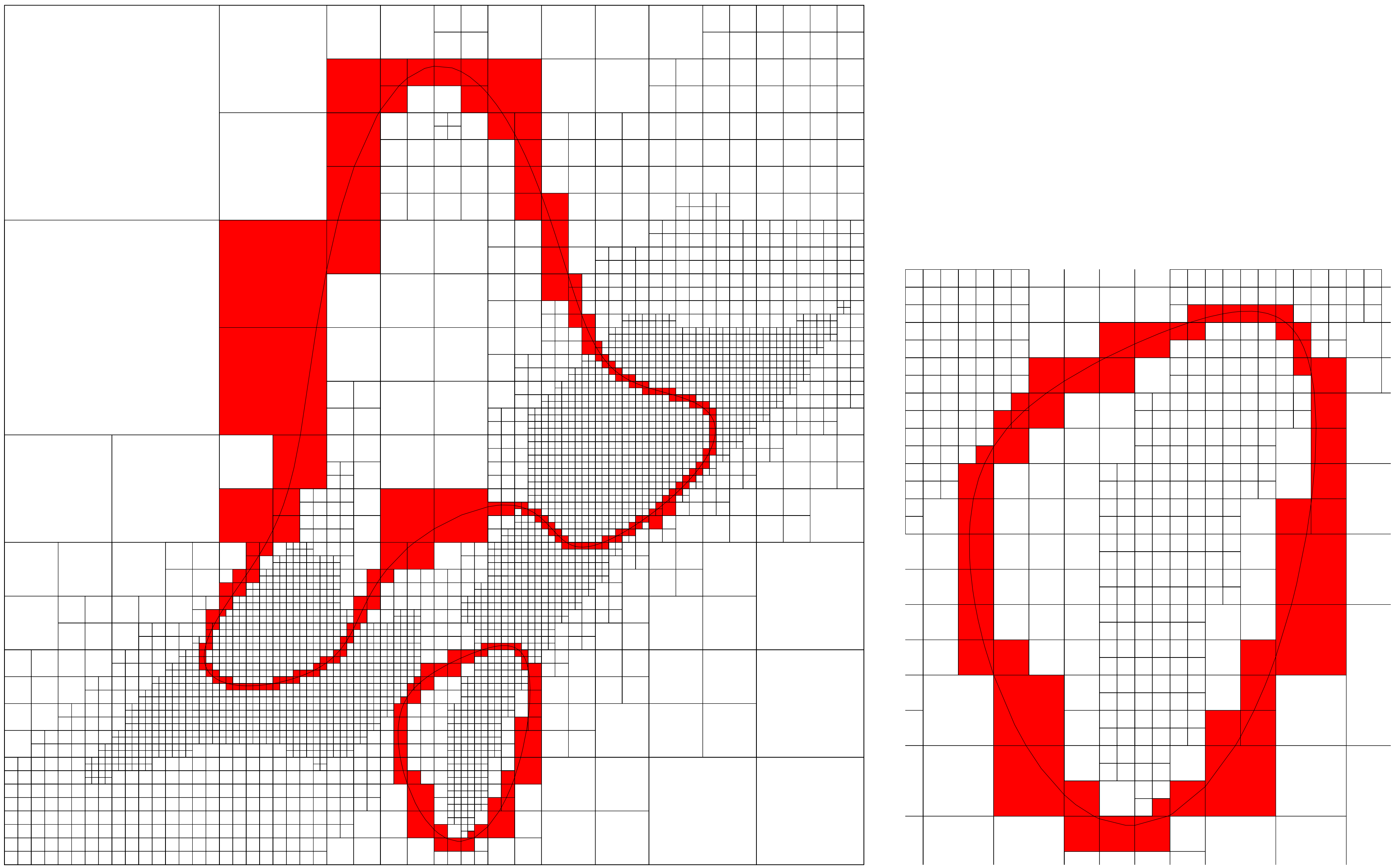


7457 boxes, 425 leaves

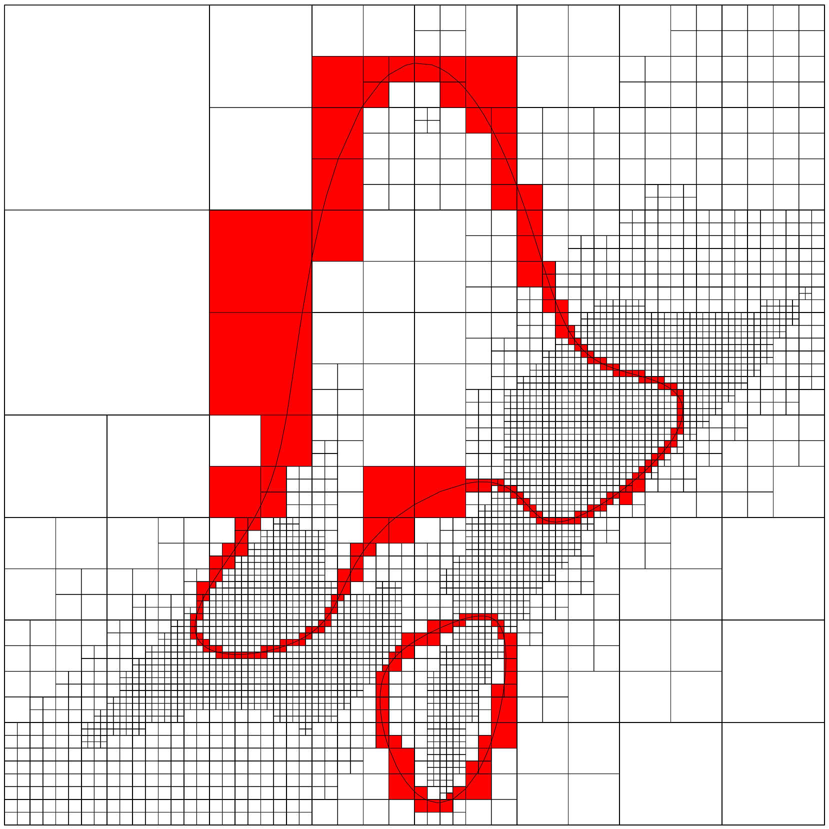
12121 boxes, 622 leaves

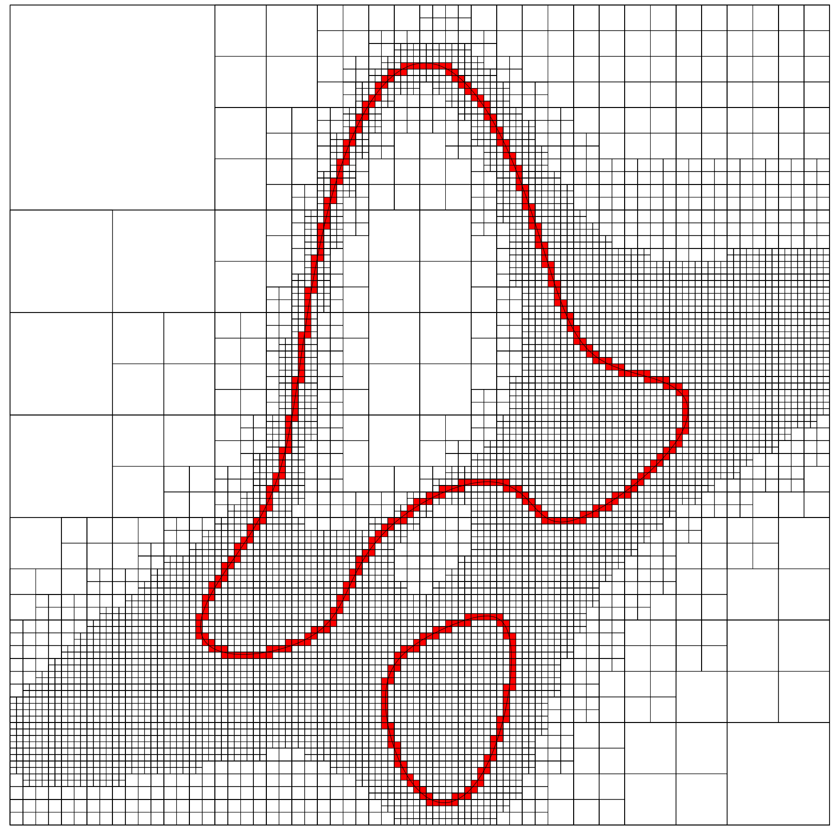
efficiency: 1.6 for boxes, 1.5 for leaves

Results: Curve from Taubin’s paper



Results: Curve from Taubin's paper





4505 boxes, 233 leaves

9161 boxes, 446 leaves

efficiency: 2.0 for boxes, 1.9 for leaves

Conclusion

- Robust adaptive approximation of implicit curves
- First algorithm to combine spatial and geometrical adaption
- Can use cache trees for speed when generating multiple level curves

Future work

- Surfaces?
  - ◇ difficult topological problems
- Use affine arithmetic to reduce overestimation
- Piecewise cubic approximation
  - ◇ Hermite formulation
  - ◇ no need to test gradient variation inside whole cell
  - ◇ need to test values over whole cubic segment