Robust Approximation of Offsets and Bisectors of Plane Curves

João Batista S. de Oliveira (PUC-RS)
Luiz Henrique de Figueiredo (IMPA)

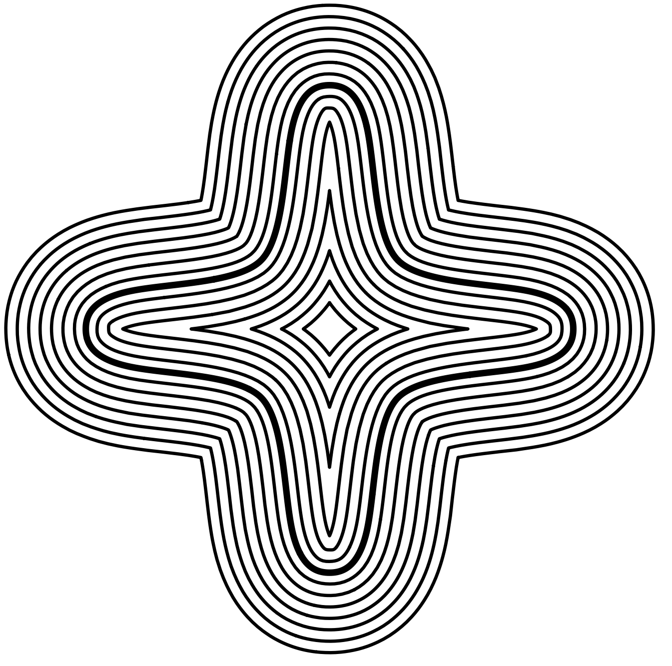
Offsets

The r-*offset* of a plane curve Γ:

$$\mathcal{O} = \{p \in \mathbf{R}^2 : d(p, \Gamma) = r\}$$

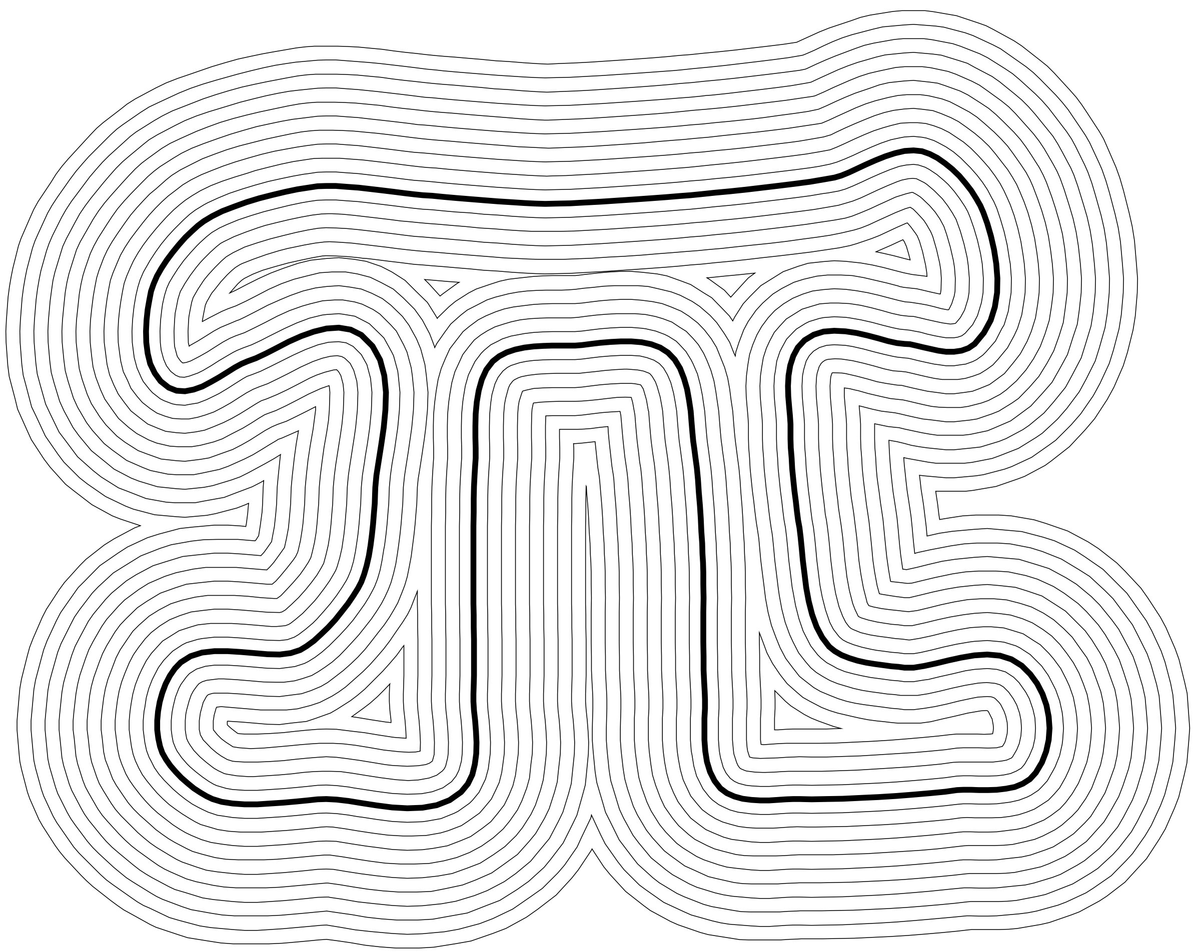
Distance of point to curve:

$$d(p, \Gamma) = \min\{d(p, q) : q \in \Gamma\}$$



Finding offset curves is a *global* problem.

Offsets are complicated



J.-H. Lee, S. J. Hong, M.-S. Kim, *The Visual Computer* (2000) 16:208–240.

Local formulation

Curve Γ given as the *trace* $\gamma(I)$ of a parametric curve $\gamma\colon I \to \mathbf{R}^2$.

Distance of point to curve:

$$d(p, \Gamma) = \min\{d(p, \gamma(t)) : t \in I\}.$$

Global minimization problem on interval I. Too hard.

Local formulation: measure distance to γ along its normal.

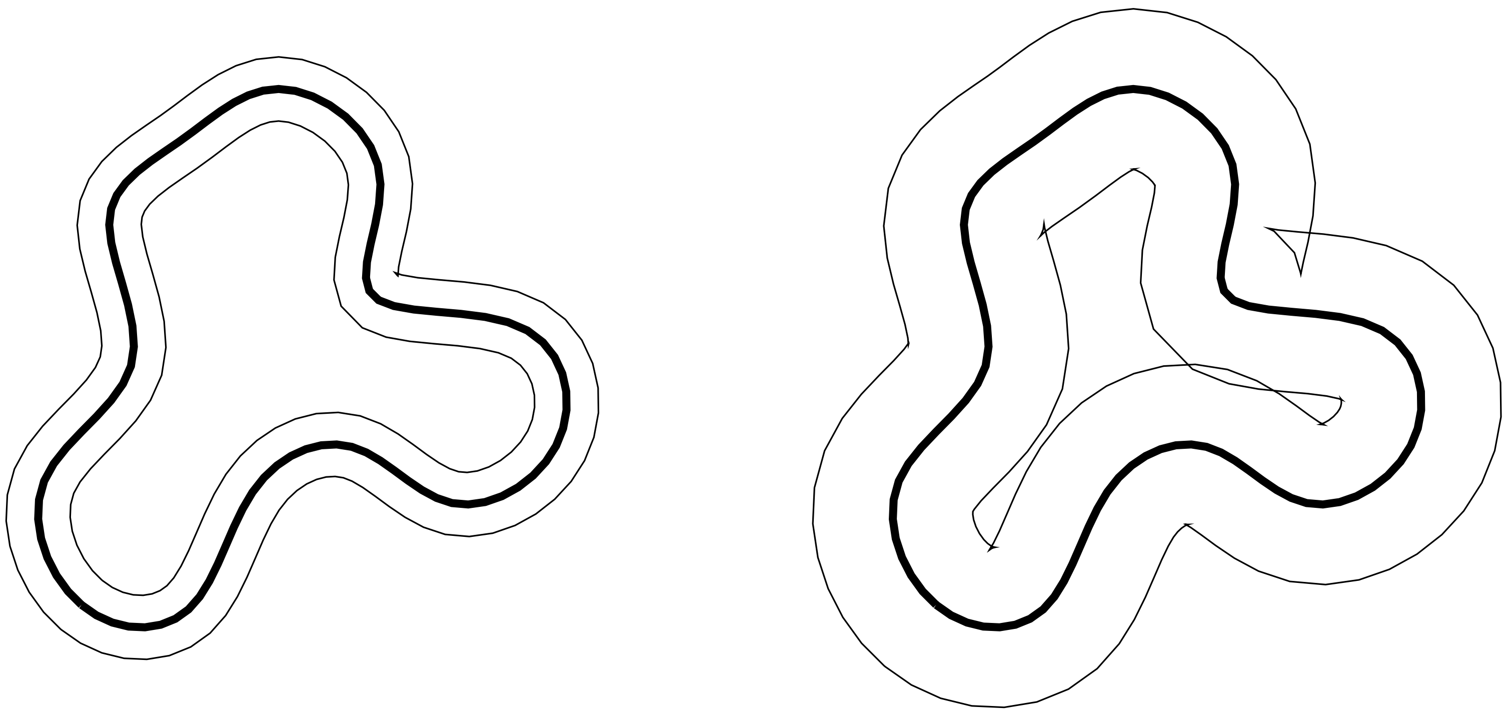
$$\mathcal{O}(t) = \gamma(t) \pm rN(t)$$

$$\gamma(t) = (x(t), y(t))$$

$$N(t) = \frac{1}{\|\gamma'(t)\|}(-y'(t), x'(t))$$

$$\|\gamma'(t)\| = \sqrt{x'(t)^2 + y'(t)^2}$$

Local formulation does not always work



Works well only when offset radius r is small. But how small?

Need *trimming* step [Farouki and Neff 1990].

Our approach:

Robust approximations with interval arithmetic. No trimming.

Range analysis

Range analysis is the study of the global behavior of real functions based on estimates for their set of values.

Given $f\colon \Omega \subseteq \mathbf{R}^n \to \mathbf{R}$, range analysis provides *inclusion function* for f:

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

Range estimates are useful for global optimization:

$$F(X) \supseteq f(X) \Rightarrow \min F(X) \le \min f(X)$$

$$r \le \min F(X) \Rightarrow r \le f(x) \text{ for \textit{all} points } x \in X$$

$$r \ge \max F(X) \Rightarrow r \ge f(x) \text{ for \textit{all} points } x \in X$$

Interval arithmetic is the natural computational tool for range analysis.

Interval arithmetic

- Quantities represented by intervals:

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

- Primitive operations:

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

- 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.

Range analysis and global geometric processing

- Recursive exploration of domain Ω.
- Discard subregions X of Ω when we can *prove* that X does not contain any part of the offset $\mathcal{O}$ (proof uses range estimates!).

```
explore(X):
    if X does not contain a part of O then
        discard X
    elseif X is small enough then
        output X
    else
        divide X into smaller pieces X_i
        for each i, explore(X_i)
```

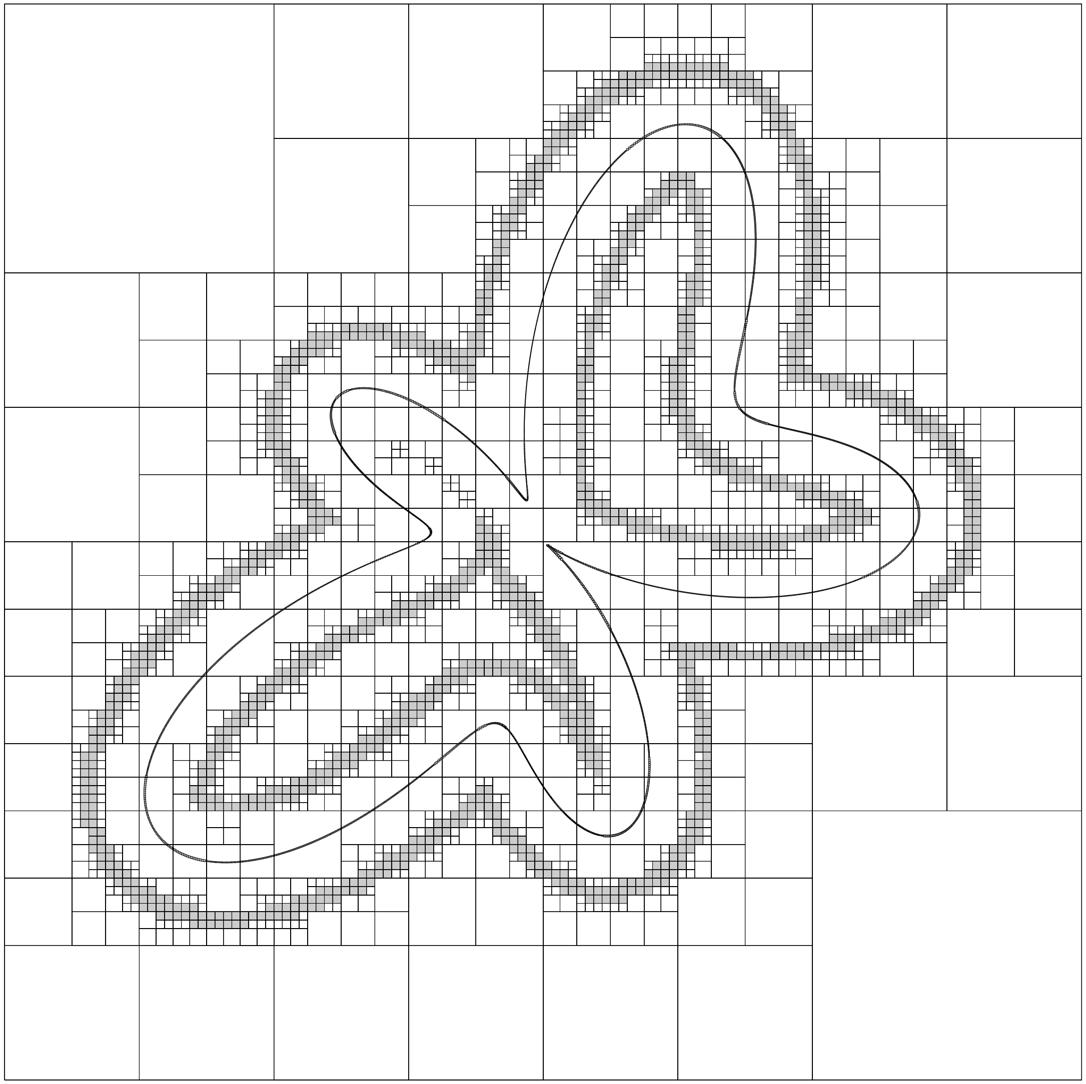
Start with explore(Ω).

Generate quadtree decomposition of Ω when Ω is a rectangle.

Compute *guaranteed* approximation of $\mathcal{O}$.

Crucial step: testing whether X is *empty*.

Robust adaptive approximation of offset – quadtree decomposition



Robust approximation of offsets – testing for emptiness

```
test(X, T, r):
    if max D(X, T) < r then
        return true
    if min D(X, T) > r then
        return false
    if diam G(T) < ε then
        all ← false
        return false
    else
        bisect T into T1 and T2
        return test(X, T1, r) ∨ test(X, T2, r)

empty(X, r):
    all ← true
    return test(X, I, r) ∨ all
```

Like interval global optimization, but needs not *find* global minimum.

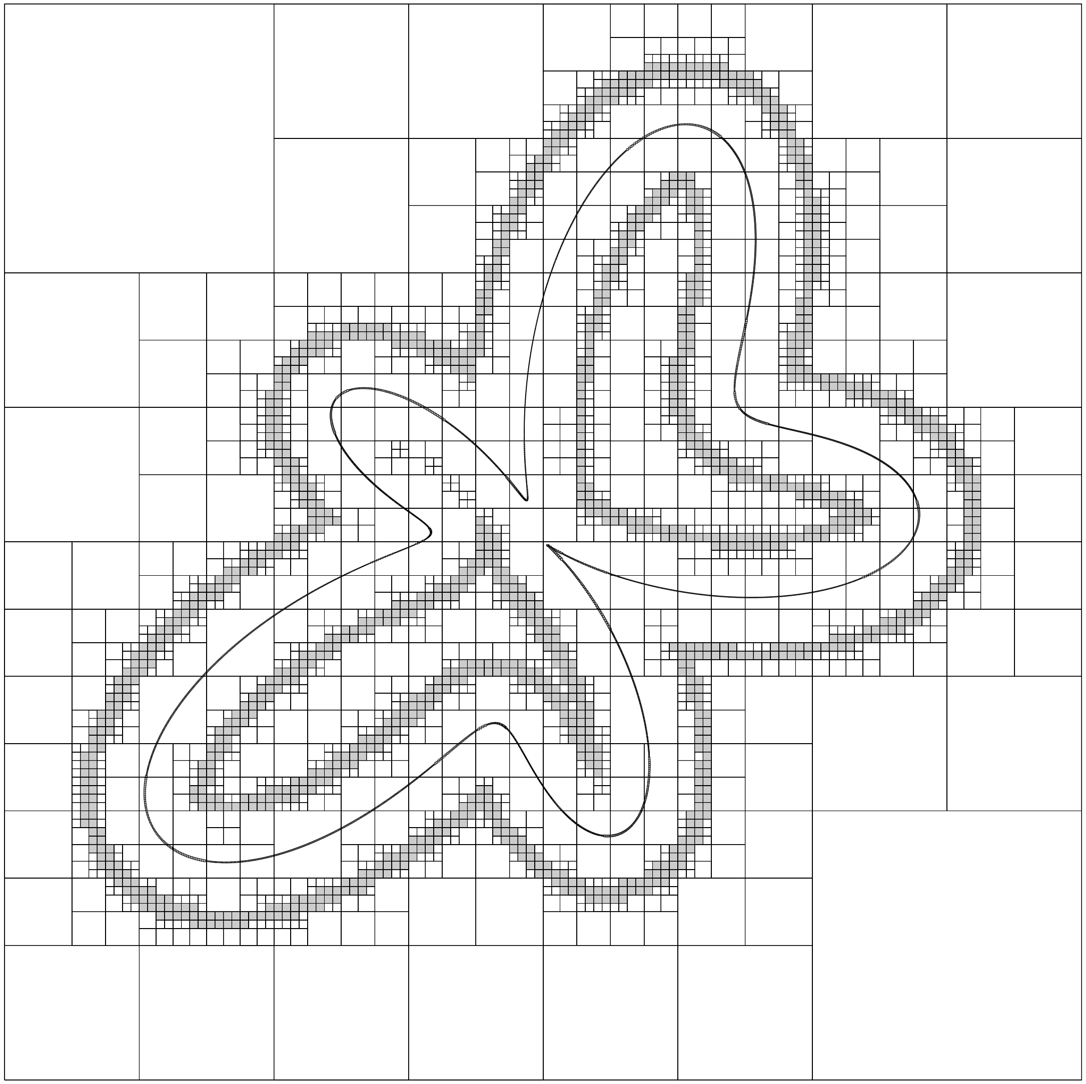
Robust approximation of offsets – main algorithm

```
explore(X):
    if empty(X, r) then
        discard X
    elseif diam(X) < ε then
        output X
    else
        divide X into four equal pieces X_i
        for each i, explore(X_i)
```

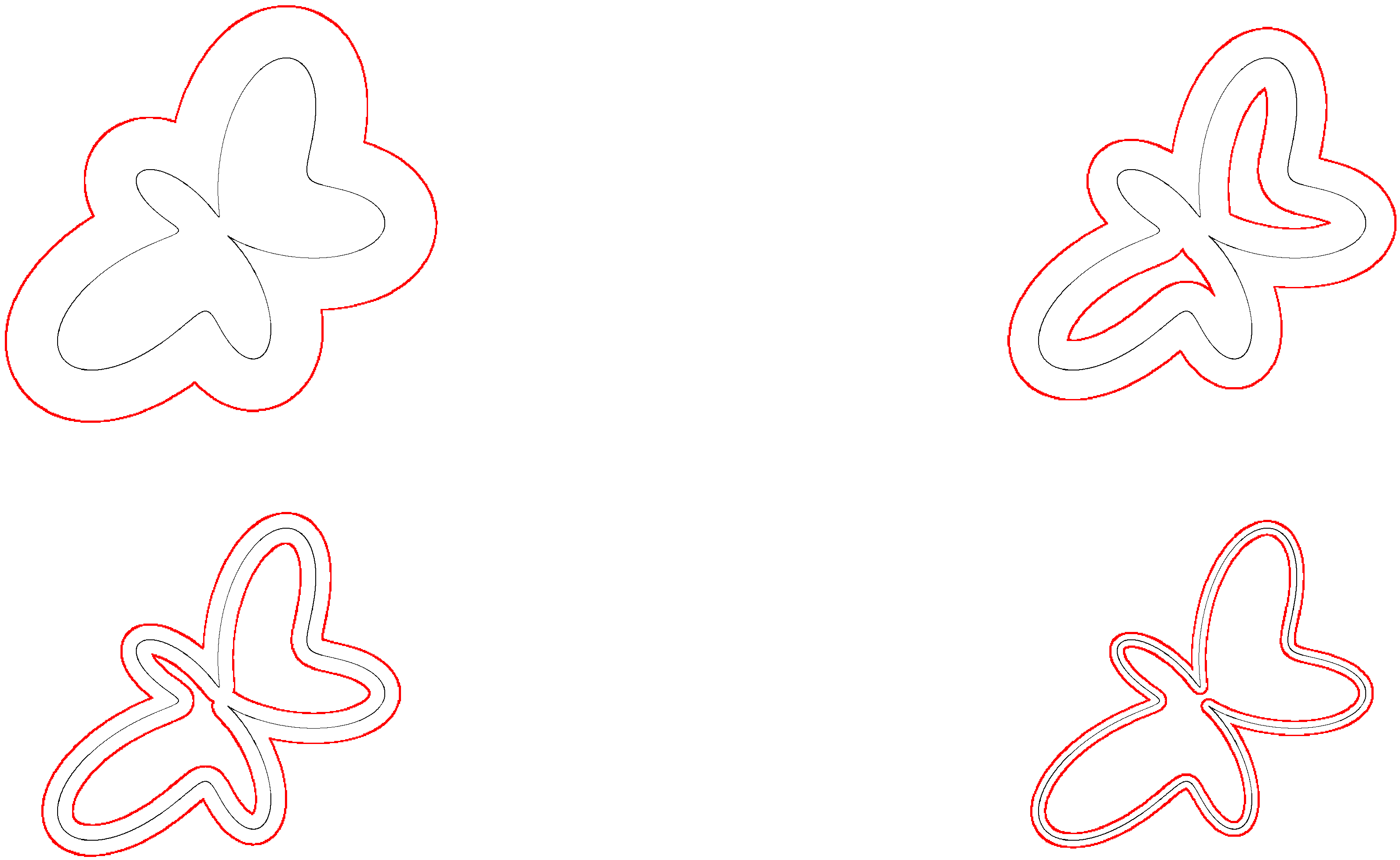
- Start with explore(Ω).
- Perform adaptive exploration of Ω.
- Quickly discard empty subregions.
- Work harder near $\mathcal{O}$.

Cache trees store evaluations of $G(T)$.

Robust adaptive approximation of offset



Offsets of decreasing radius



Cache trees

- Interval estimates $D(X, T)$ and $G(T)$ may dominate cost.
- $G(T)$ used twice in test(X, T, r) but $G(T)$ does not depend on X.
- Cache values of $G(T)$ in a binary tree and re-use them.
- Root of tree corresponds to $T = I$.
- Each node contains $G(T)$ and pointer to children nodes, corresponding to T_1 and T_2, the two halves of T.
- Reduce overestimation by updating estimates from the bottom up.

The cache tree is a dynamic adaptive representation of γ on I: it summarizes the behavior of γ at various resolution scales, and gets locally refined as needed when X varies.

- Approximation in previous figure needed 220089 evaluations of G, but cache contained 218618 of these; only 1471 fresh evaluations were required (less than 1%).

Point/curve bisectors

The bisector of a point p_0 and plane curve Γ:

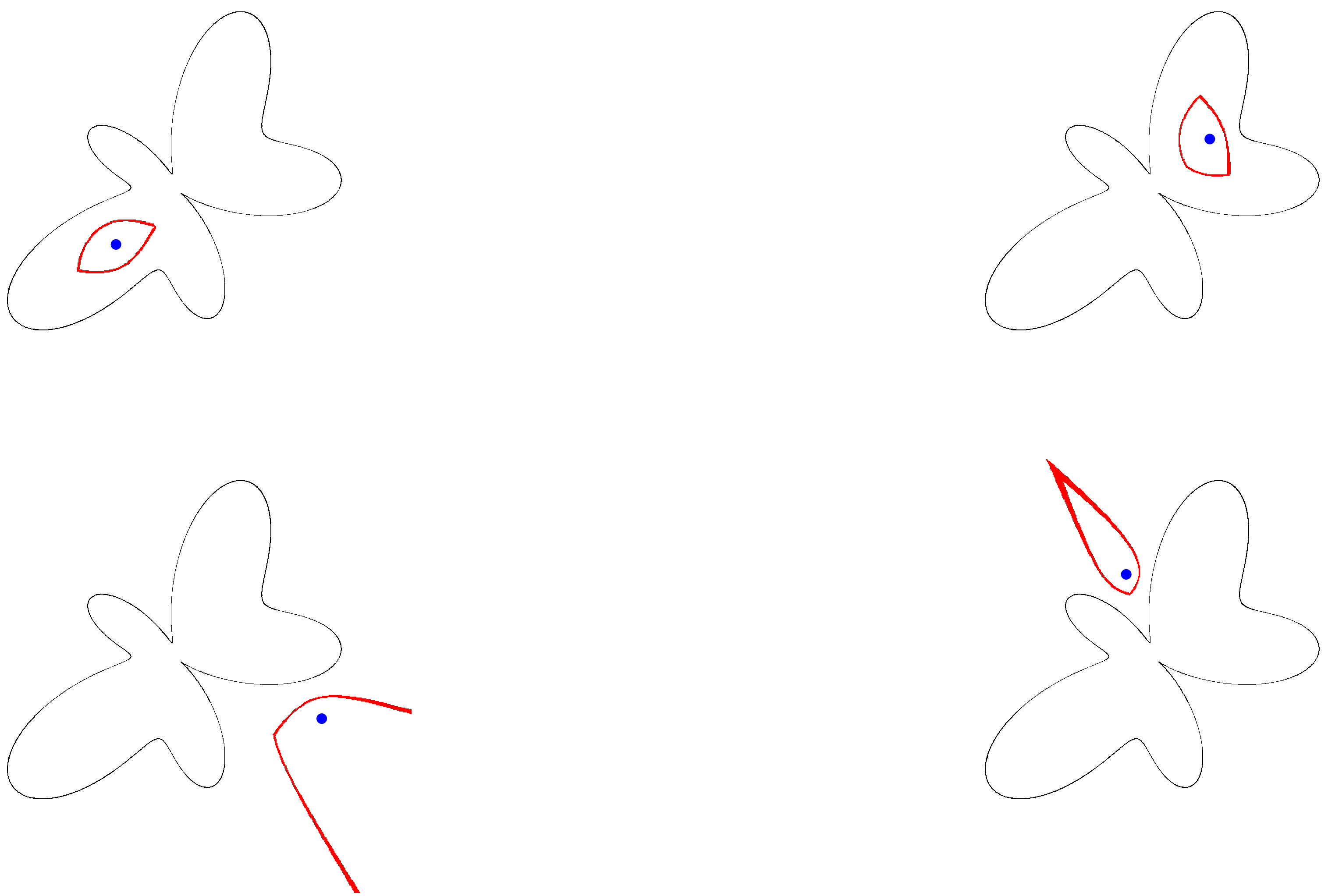
$$\mathcal{B} = \{p \in \mathbf{R}^2 : d(p, \Gamma) = d(p, p_0)\}$$

Trivial to modify offset algorithm to compute bisectors:

test(X, T, r):
  if $\max D(X, T) < \min D(X, p_0)$ then
    return true
  if $\min D(X, T) > \max D(X, p_0)$ then
    return false
  $\ldots$

No overestimation in $D(X, p_0)$, because variables x and y occur only once in $d(p, p_0)$.

Examples of point/curve bisectors



Conclusion

- Robust adaptive approximation of offsets and bisectors.
- Works for non-smooth curves: no normal vector required.
- Works even for discontinuous curves: just need inclusion functions for each piece.
- No need for trimming.
- Need to reconstruct curves from box approximation.
- Uses cache trees for speed.
- Future work
  - ◇ Curve/curve bisectors and medial axes.
  - ◇ Adaptively sampled distance fields (ADFs) for parametric curves.
  - ◇ Use affine arithmetic to reduce overestimation.
  - ◇ Surfaces?