Morphology

- Shape complexity
- Distance
- Dilation, contraction
- Fill, round, combos
- Mortar and tolerance zone
- Mason
- Tightening
- Transforms (distance, stability, ball)

Overarching objective

SIMPLIFICATION

Understand what it means to simplify a shape

Develop an explicit mathematical formalism

not defined by an algorithm, but by a mathematical formulation

independently of any particular domain or representation

MULTI-SCALE ANALYSIS

Explore the possibility of analyzing a shape, as it is increasingly simplified, so as to understand its morphological structure and identify/measure its features

identify features of interior, boundary, exterior

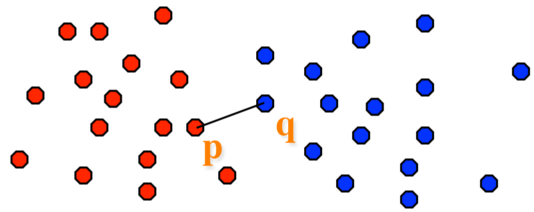
assess their resilience to simplification

Motivation

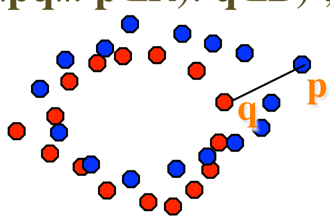
- a grown shape: primitive for many constructions and definitions
- minimum distance
  - Testing for clearance and design rules
  - Tracking possible collisions between moving shapes in animations
  - Translating objects to achieve desired contact
- maximum discrepancy (Hausdorff distance)
  - Testing how similar two shapes are for matching and statistics
  - Measuring the error between a shape and its approximation
  - Selecting the best next simplification operation

Distance/discrepancy between points clouds

- Consider a red cloud R and a blue cloud B of points (2D, 3D...)
- She picks one point p, I pick a point q of the other color
- Return the distance ||pq||
- Define D(p,S)≡Min(||pq||: q∈S)
- Minimum distance: *"We love each other"*
  - D(R,B) ≡ Min(D(p,B): p∈R)
    - D(R,B) ≡ Min(||**pq**||: **p**∈R, **q**∈B)

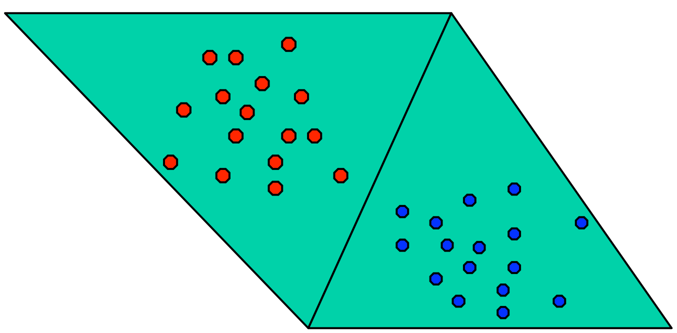


- Maximum discrepancy (Hausdorff distance): *"I love her, she hates me"*
  - H(R,B) ≡ Max(Max(D(p,B): p∈R) , Max(D(q,R): q∈B))
    - H(R,B) ≡ Max(Max(Min(||**pq**||: **p**∈R): **q**∈B) , Max(Min(||**pq**||: **q**∈B): **p**∈R))



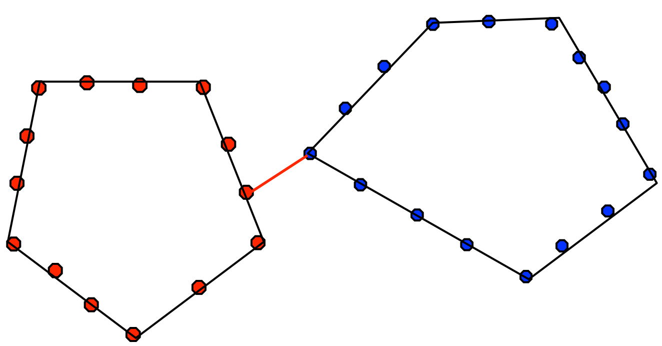
Approximate distance, discrepancy for T-meshes

- Approximate both T-meshes by dense clouds of points
- Compute distance or discrepancy between the two clouds
- Use them as approximations of the real ones
  - Trade-off between the accuracy of the approximating measures and the cost (number of pairs of points than must be considered)
- How to generate the clouds of points?



Approximate distance & discrepancy for curves

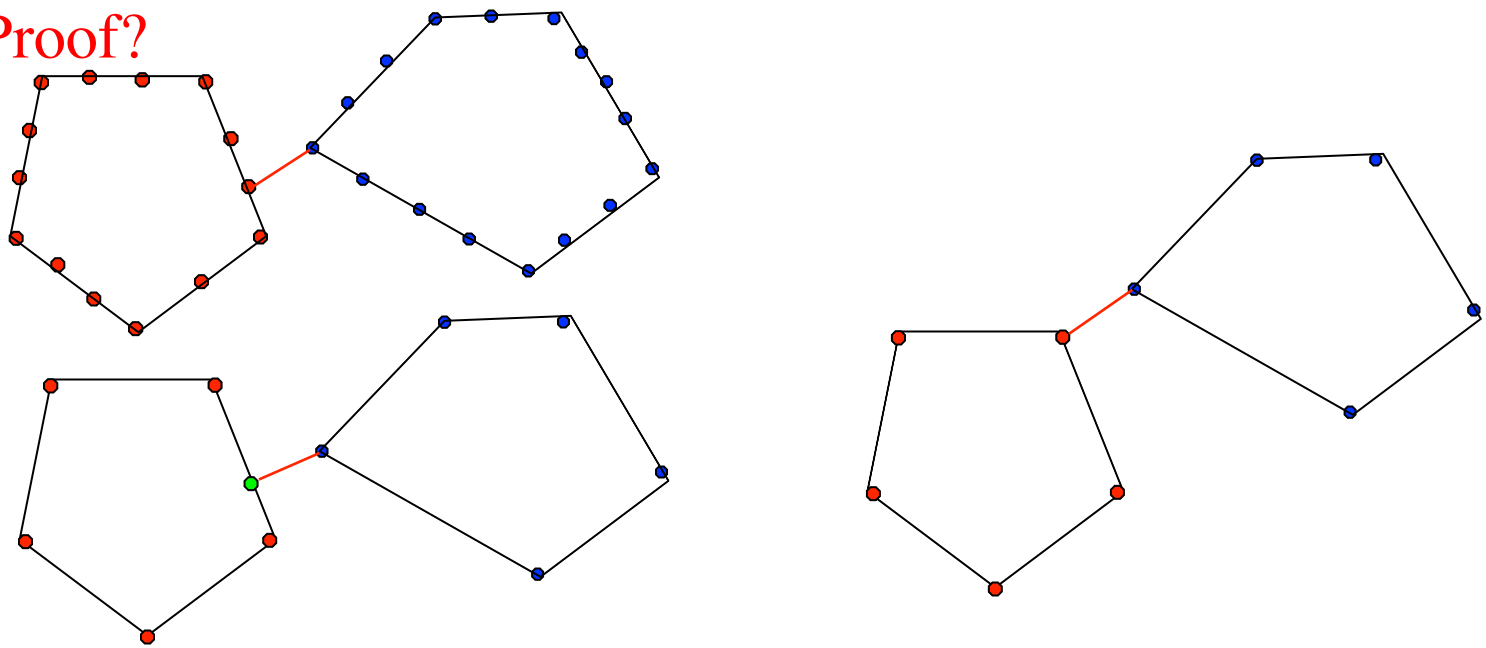
- Approximate both curves by dense series of points
- Compute distance / discrepancy between the two discrete sets
- Use them as approximations of the real ones
  - Trade-off between the accuracy of the approximating measures and the cost (number of pairs of points than must be considered)
- How to generate the points?
  - Want uniformly spaced points along each edge



Exact distance between two polygons

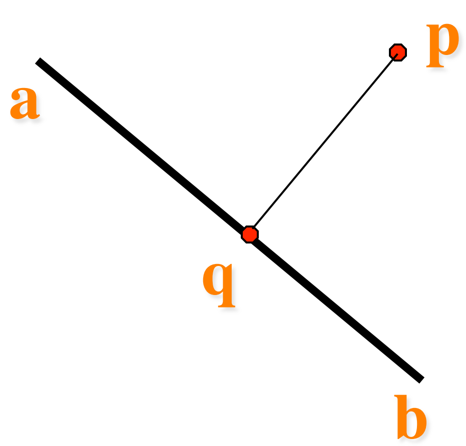
- Length of the shortest line that joins a vertex p of one shape (could be A or B, bust try both) with a point q of the other shape.
- Point q is either a vertex or the closest projection of p on an edge.

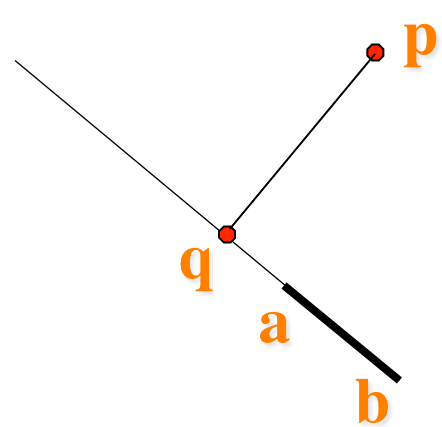
Proof?



How to compute closest projection on edge

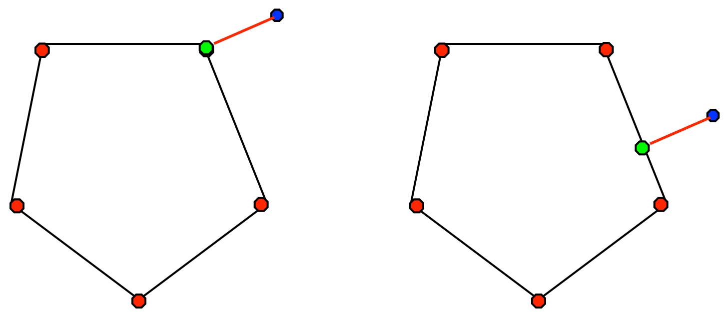
- Check that the projection falls between bounds
  - How to test that q lies between a and b?
    - Hint: dot product
  - How to compute ||pq||
    - Hint: Pythagorean Theorem (http://www.davis-inc.com/pythagor/proof2.html)





How to compute point/polygon distance

- Return minimum of distance to all vertices and of distance to projections that fall on edges



How to compute polygon/polygon distance

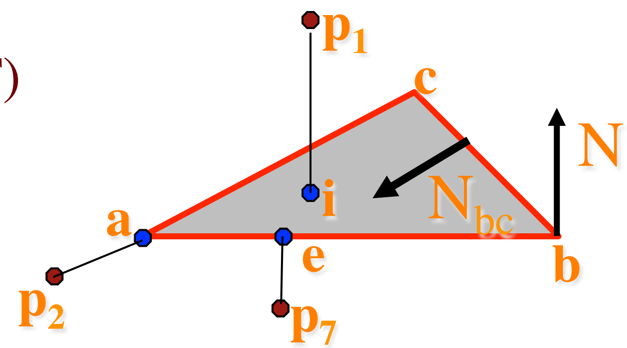
```
d:=100000;
For each vertex p of A do
    For each vertex q of B do
        if ||pq|| < d then d :=||pq||;
For each vertex p of A do
    For each edge ab of B do
        If p projects onto edge(a,b) then {
            d' :=distance from p to its projection on the edge;
            If d' <d then d:=d' ;}
For each vertex p of B do
    For each edge ab of A do
        If p projects onto edge(a,b) then {
        d' :=distance from p to its projection on the edge;
        If d' <d then d:=d' ;}
Return(d);
```

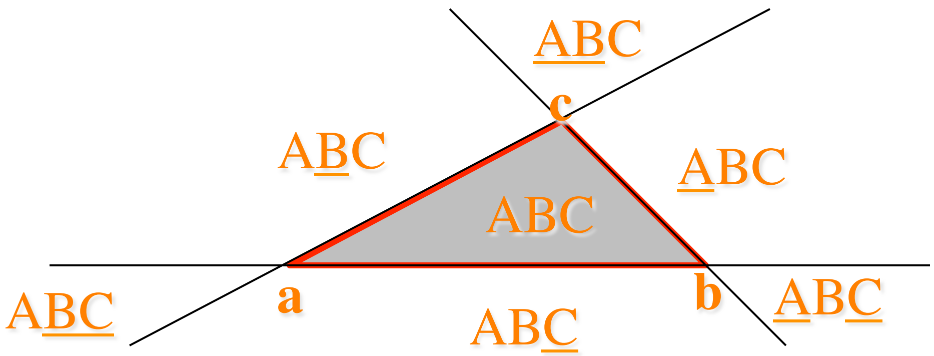
Exact point-triangle distance

T has vertices a, b, c

Algorithm for computing point-triangle distance D(p,T)

- A:= (bc•bc)(ba•bp) > bc•ba)(bc•bp)
- B:= (ca•ca)(cb•cp) > (ca•cb)(ca•cp)
- C:= (ab•ab)(ac•ap) > (ab•ac)(ab•ap)
- RETURN
  1. if A B C then ap•(ab×ac)/||ab×ac||
  2. if A $\underline{\text{B}}$ $\underline{\text{C}}$ then ||ap||
  3. if $\underline{\text{A}}$ B $\underline{\text{C}}$ then ||bp||
  4. if $\underline{\text{A}}$ $\underline{\text{B}}$ C then ||cp||
  5. if $\underline{\text{A}}$ B C then ||bc×bp||/||bc||
  6. if A $\underline{\text{B}}$ C then ||ca×cp||/||ca||
  7. if A B $\underline{\text{C}}$ then ||ab×ap||/||ab||





Justification

- N := bc×ba
- N_{bc} := N×bc = (bc×ba)×bc = (bc•bc)ba–(bc•ba)bc
  - using u×(v×w)=(u•w)v–(u•v)w
- A := N_{bc}•bp > 0

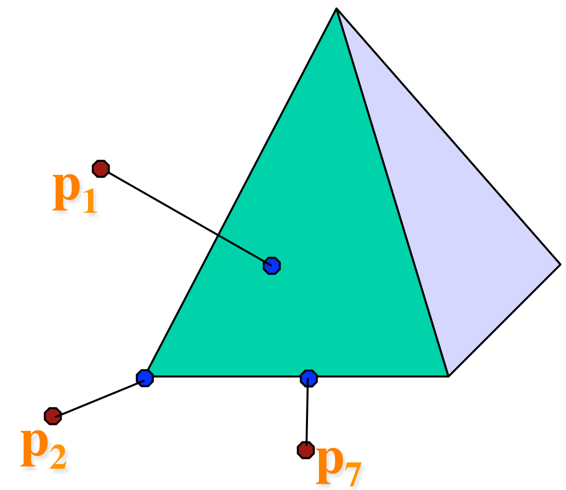
Cost: over 45 multiplications per triangle! Speed-ups?

Exact point-surface distance for T-meshes

- D(p,S) ≡ Min(D(p,T) : T is a triangle of S),using previous slide
- Speed-up
  - d:= min(D(p,v) : v is a vertex of S)
  - For each triangle T of S do
    - D(p,plane(T))<d then if normal projection of P on T then d:= D(p,plane(T))
  - For each edge E do
    - D(p,line(E))<d then if normal projection of P on E then d:= D(p,line(E))
- Further speed-up
  - Preprocessing
    - Compute a minimal sphere around each triangle
    - Group neighboring triangles and compute spheres around groups
    - Build tree of spherical containers by recursively merging groups
  - Query for a point p
    - Go down the tree, picking at each node the sphere whose center is closest to p
    - d:= distance to leaf triangle
    - Go down the tree again in depth first order, but only visit branches whose spheres are closer to p than d
      - At each visited leaf, update d

Distance between a point and a T-mesh

- The minimum distance may occur
  - In the middle of a face (p_1)
  - In the middle of an edge (p_7)
  - At a vertex (p_2)

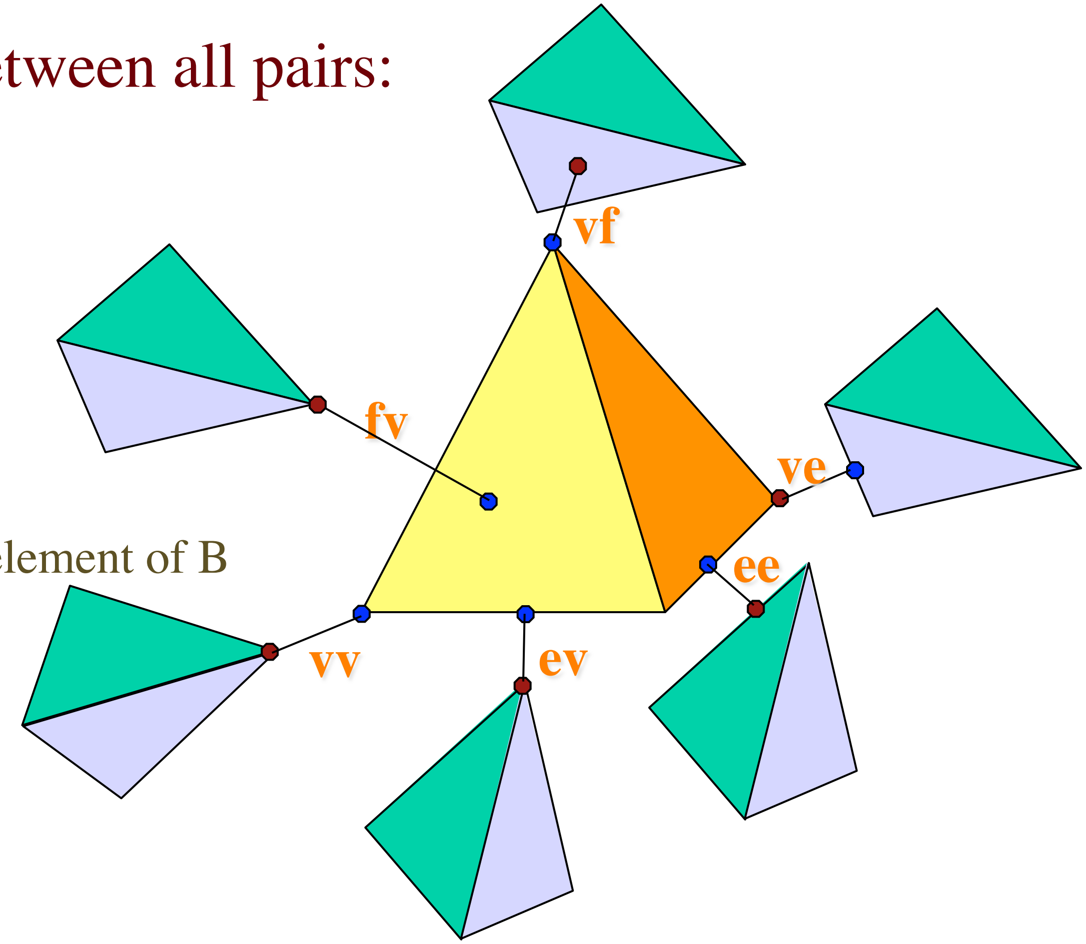


- The distance to the solid bounded by a T-mesh S is
  - 0 it p is inside the solid
    - use parity of the number of intersections of ray with S
  - D(p,S) otherwise

Distance between two T-meshes

- Minimum of distances between all pairs:
  - face/vertex
  - edge/vertex
  - vertex/vertex
  - vertex/edge
  - vertex/face
  - edge/edge

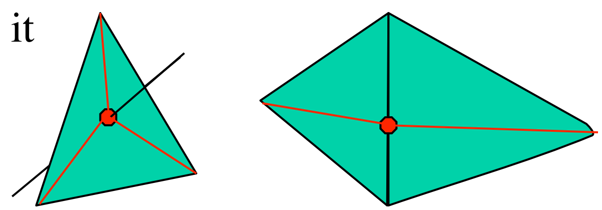
  of an element of A and an element of B



N:=(ab×cd) / ||ab×cd||
Dist:= N•ac
ee test ?

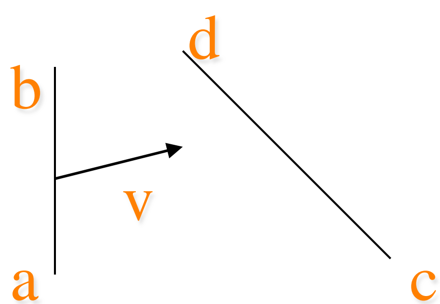
Intersection between T-meshes

- The triangle meshes A and B intersect if and only if the closure of an edge of one intersects the closure of a triangle of the other
    - The closure of an edge is the union of the edge with its vertices
    - The closure of a triangle is the union of the triangle with its boundary
  - These include the cases where an edge intersects or touches an edge or when a vertex touches a triangle
- How do we compute the intersection?
  - Assume A and B are represented by corner tables
  - Do all pairs of edge-triangle intersection tests
    - Each time you find an intersection, insert it in both meshes and subdivide the triangles to restore a valid triangulation and a correct corner table
    - With each vertex that you insert, associate the IDs of the 3 triangles involved
      - Two bounded by the edge and one intersected by it
  - Identify the intersection edges
    - They are new edges whose vertices share 2 IDs

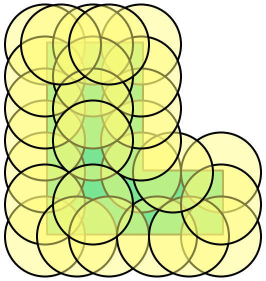


Collision prediction between T-meshes

- Assume A moves with constant velocity v and B is fixed
  - If both move, compute collision in local coordinate system of B
- Collision may occur between
  - A vertex p of A and (the closure of) a triangle T of B
    - **Intersect Ray(p,v) with T**
  - The closure of a triangle T of A with a vertex p of B
    - **Intersect Ray(p,-v) with T**
  - An edge (a,b) of A with an edge (c,d) of B
    - **Check when the volume of tetrahedron (a+tv,b+tv,c,d) becomes zero**
      - solve (cd×(ca+tv))•(cb+tv)=0 for t
      - (cd×(ca+tv))•cb + (cd×(ca+tv))•tv)=0
      - (cd×ca+t(cd×v))•cb + (cd×ca+t(cd×v))•tv)=0
      - (cd×ca)•cb +t(cd×v)•cb + (cd×ca)•tv +t^2(cd×v)•v = 0
      - (cd×ca)•cb +t(cd×v)•cb + (cd×ca)•tv = 0, because (cd×v)•v = 0
      - t = (ca×cd)•cb / ((cd×v)•cb - (cd×v)•ca)
      - **t = (ca×cd)•cb / (ab×cd)•v**
    - **Make sure that, at that time, the two edges intersect**
      - **Not just the lines**



Growing a shape by a distance r



- $S\uparrow r \equiv \{p : r \geq D(p,S)\}$
  - $(S\uparrow r) \cup$ {points whose distance from S is not exceeding r}
- Equivalent definition of the "grown S"
  - $S\uparrow r \equiv \cup \{ \text{ball}(p,r) : p \in S\}$
    - Replace each point of S by a ball of radius r and union them
- May be used to provide equivalent definitions of distance
    - *"I love her, she loves me"*
  - $D(A,B) \equiv \text{Min}(\|pq\| : p \in A, q \in B)$
  - $D(A,B) \equiv \text{Max}(r : (A\uparrow r) \cap B = \emptyset)$
    - *largest amount by which one can grow before it interferes with the other*
- May be used to provide equivalent definitions of discrepancy (Hausdorff)
    - *"I love her, she hates me"*
  - $H(A,B) \equiv \text{Max}(\text{Max}(\text{Min}(\|pq\|: p \in R): q \in B), \text{Max}(\text{Min}(\|pq\|: q \in B): p \in R))$
  - $H(A,B) \equiv \text{Min}(r : A \subset (B\uparrow r) \text{ and } B \subset (A\uparrow r))$
    - *smallest amount by which each must grow to contain the original of the other*

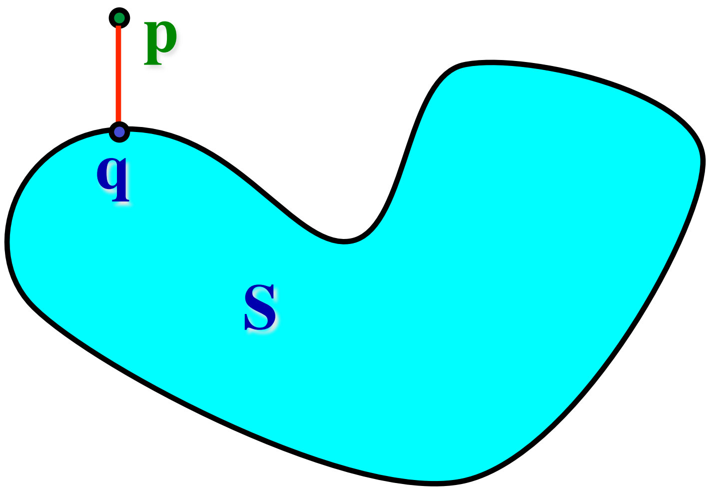
Distance and closest projection

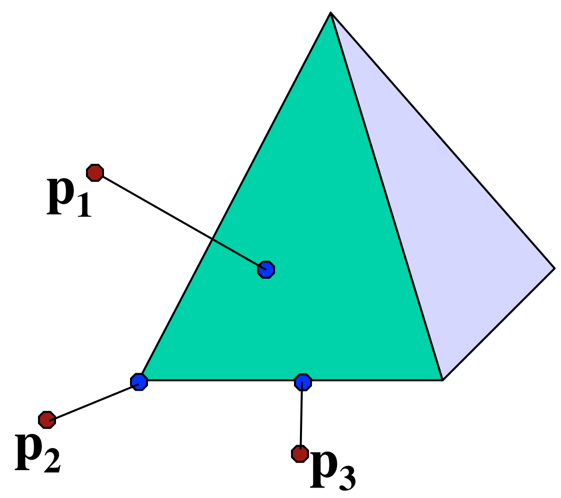
$D_{Min}(\mathbf{p},\mathbf{S})$ is the **minimum distance** from point **p** to set **S**.

The **closest projection** $C_S(\mathbf{p}) = \{ \mathbf{q} : \|\mathbf{pq}\| = D_{Min}(\mathbf{p},\mathbf{S}) \}$

If the boundary of **S** is smooth, vector **qp** is orthogonal to **S** at **q**.

If not, it is in the normal fan at **q**





Hausdorff distance

- The Hausdorff distance $D_{Haus}(A,B)$ between two sets A and B is the **minimum** r such that $A \subset B\uparrow r$ and $B \subset A\uparrow r$

  Where $A\uparrow r$ (also denoted A^r) is the dilation of A by a distance r

  I love her and want to be close to her.

  She hates me and want to keep me away.

  She picks a set (A or B).

  I have to stay on the other one.

  I try to get as close as possible.

  She finds a spot that is the furthest from my set.

  We are separated by $D_{Haus}(A,B)$

Polygon/polygon maximum discrepancy

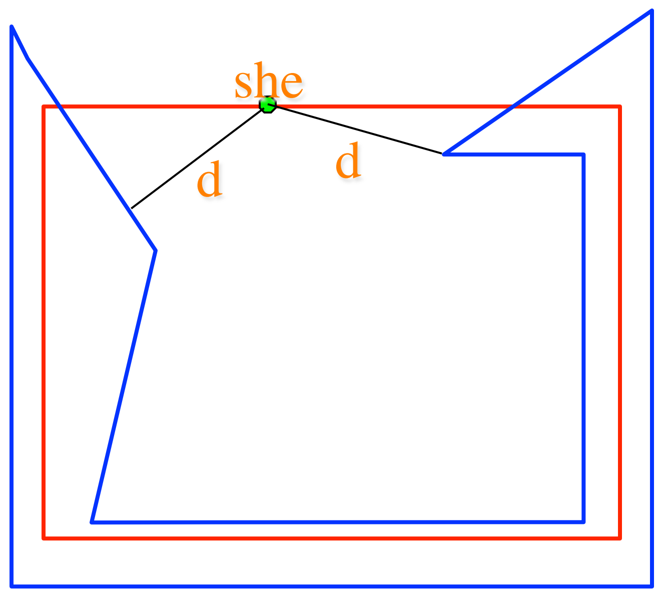
d:=0;

For each vertex p of A do if Dist(p,B) > d then d := Dist(p,B)

For each vertex p of B do if Dist(p,A) > d then d := Dist(p,A)

...

Return (d)



Is this sufficient ?

Complete the algorithm!

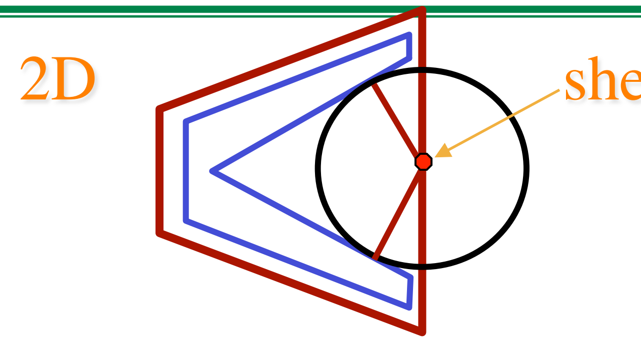
Hausdorff distance

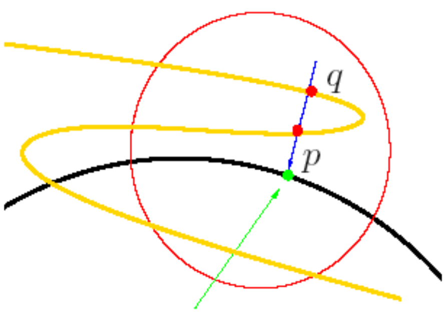
It is expensive to compute

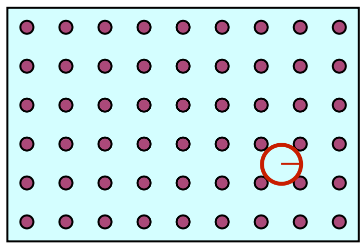
  even in 2D

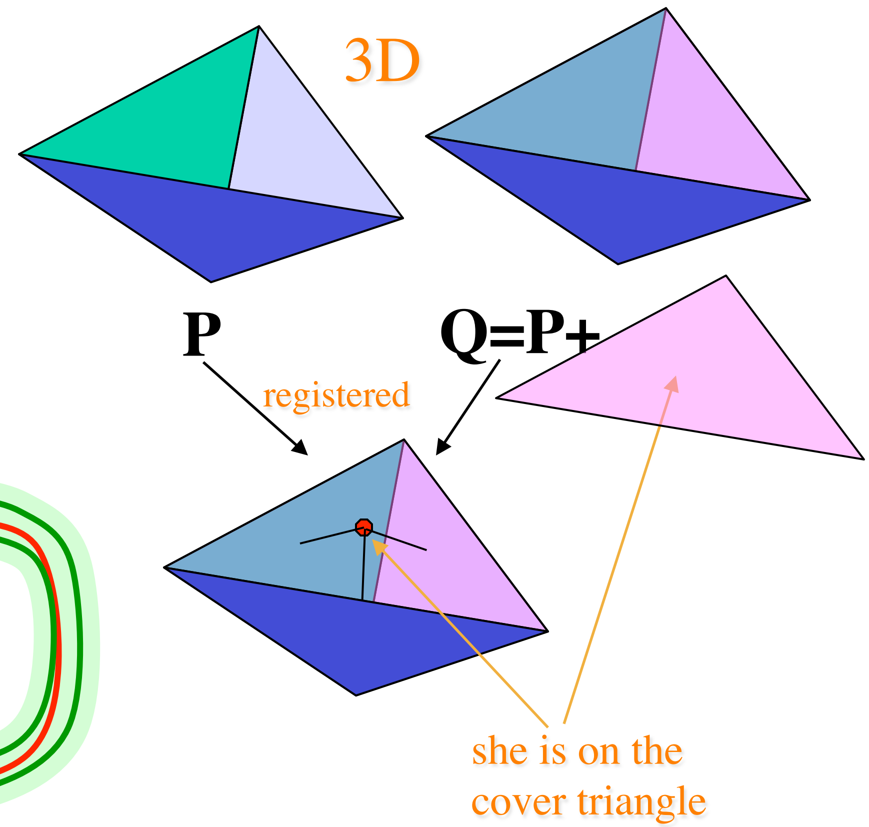
It is a poor measure of discrepancy

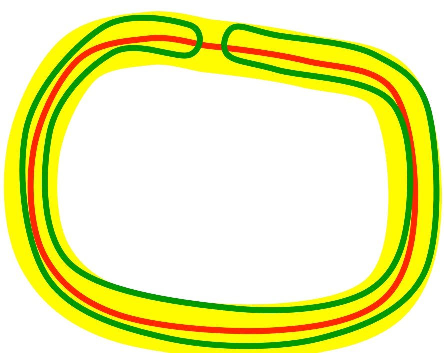
  it is orientation and topology insensitive

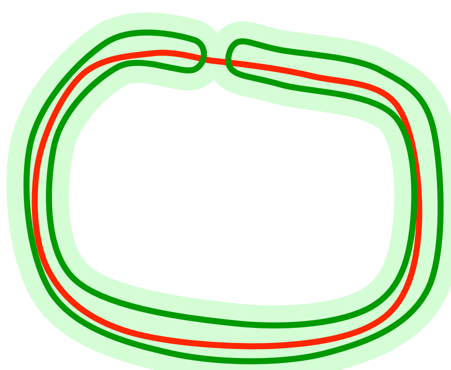






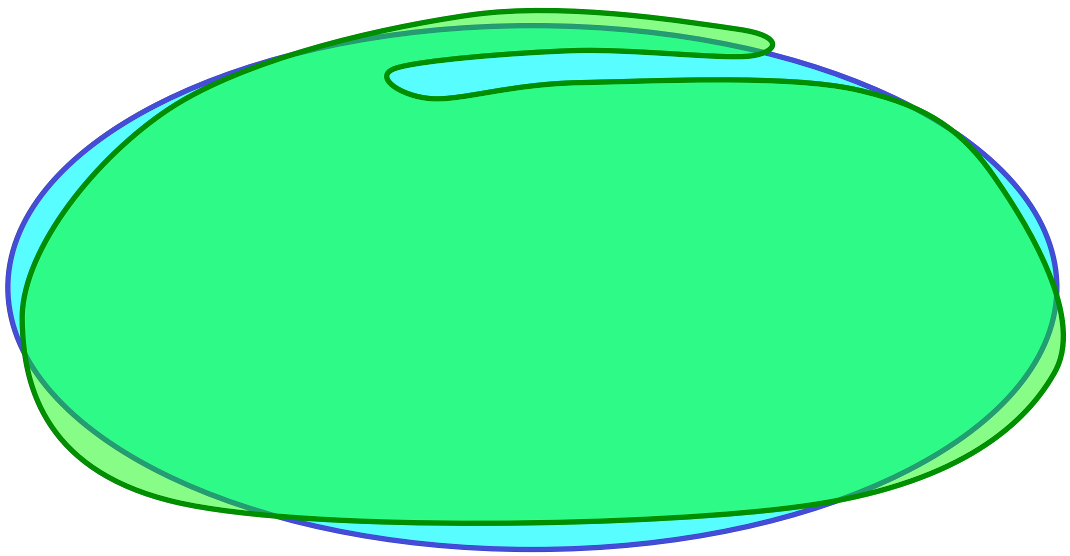






Discrepancy Measures

- How can we measure the discrepancy between two shapes?
  - Local measure? If so, at each point of each shape or of space?
  - Global measures? If so, max, average?



Parametric and Fréchet distance

Parameterized curves: **P**(t) and **Q**(s) for s and t $\in[0,1]$

Map φ: $[0,1]\rightarrow[0,1]$

Parametric discrepancy:

$D_{\varphi}(\mathbf{P},\mathbf{Q}) = \max \| \mathbf{P}(t)-\mathbf{Q}(\varphi(t)) \| \ : \ t\in[0,1]$

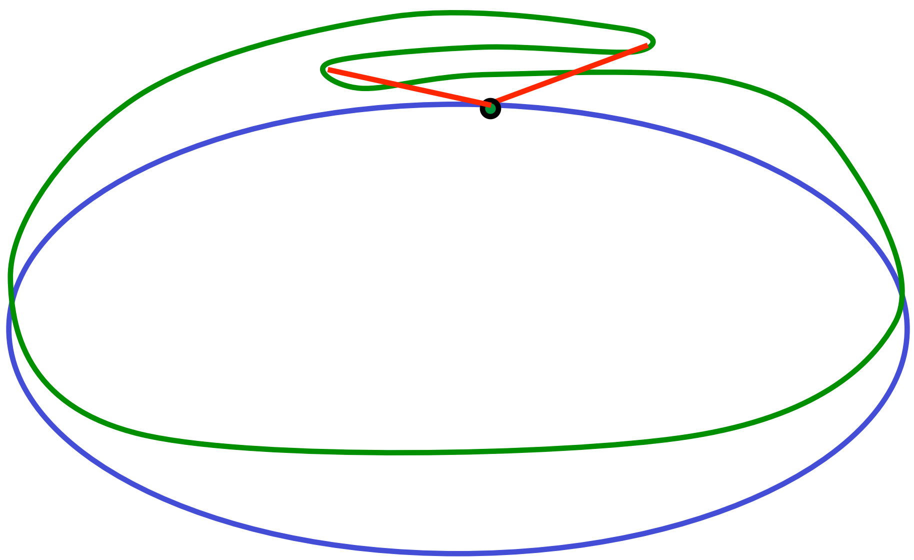
Fréchet distance:

$D_{Fre}(\mathbf{P},\mathbf{Q}) = \min D_{\varphi}(\mathbf{P},\mathbf{Q}) : \forall\varphi$

Man walks forward on P.

Dog walks forward on Q.

Length of shortest leash?



Examples of questions

- Definition of a grown set
- The definitions of distance and discrepancy
  - Metaphor, min/max, in terms of grown sets
- Their computation for clouds of points (sampling surfaces)
- The computation of point/triangle and point/T-mesh distance
- The prediction of collision for T-meshes under linear motion
- The volume change resulting form the collapse of a single edge
- The Hausdorff error resulting form the collapse of a single edge
- The formulation of the quadric error
- Its use for locating the optimal vertex and the formula

Example problem

Consider two triangle meshes A and B

A' = cloud of points on A

B' = cloud of points on B

For each one of the following statements, if it is always true, then justify it, otherwise, provide the correct version (by changing the inequality sign) and justify it.

1. $D(A',B') \leq D(A,B)$
2. $H(A',B') \leq H(A,B)$

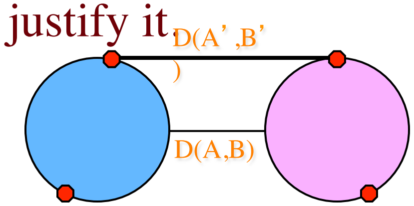
Solution

Consider two triangle meshes A and B

A' = cloud of points on A

B' = cloud of points on B

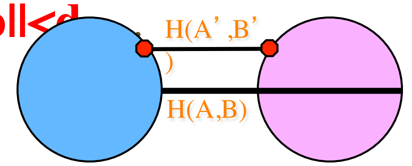
For each one of the following statements, if it is always true, then justify it, otherwise, provide the correct version (by changing the inequality sign) and justify it.



1. $D(A',B') \le D(A,B)$ is wrong (counterexample)

   $D(A,B) \le D(A',B')$ is correct

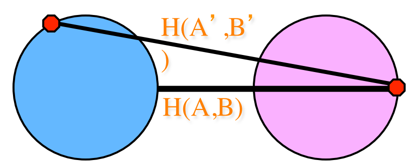
   - Prove that $D(A,B) > D(A',B')$ is impossible
   - Assume $D(A,B)=d$, then $\forall \mathbf{a} \in A, \forall \mathbf{b} \in B$, $\mathbf{d \le \|ab\|}$
   - Suppose $D(A',B')<d$, then $\exists \mathbf{a} \in A' \subset A, \exists \mathbf{b} \in B' \subset B$, $\mathbf{\|ab\|<d}$



2. $H(A',B') \le H(A,B)$ is wrong (counterexample)

   - Pick a single sample on each set

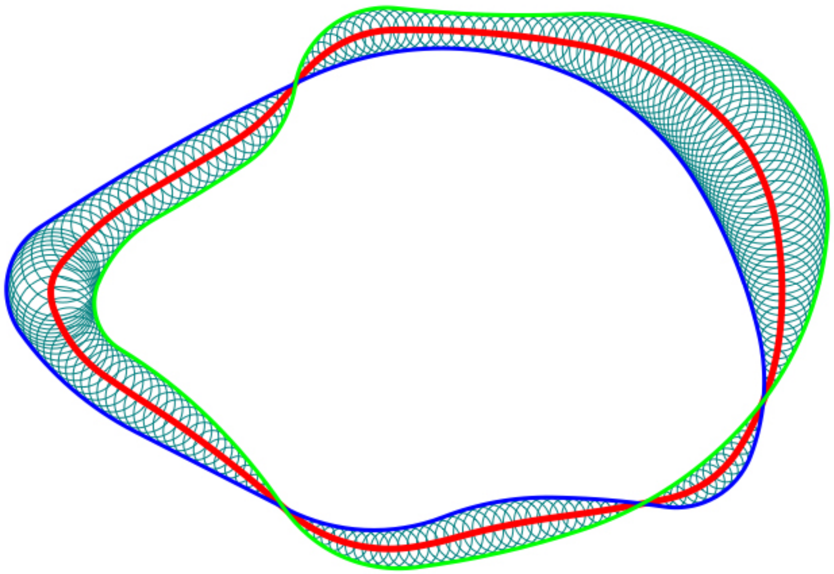
   $H(A,B) \le H(A',B')$ is also wrong (counterexample)

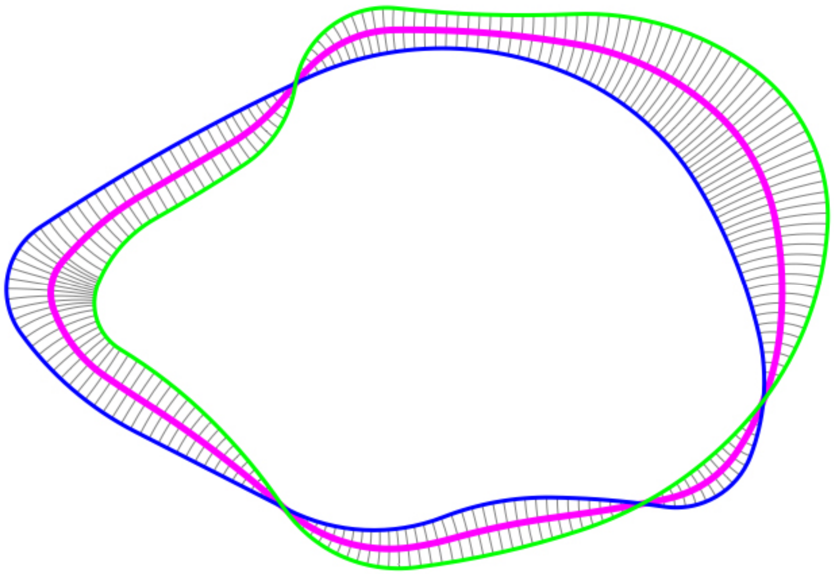


Ball distance ✷new✷

$D_{Ball}(\mathbf{P},\mathbf{Q})$ = Diameter of largest ball that fits in the **gap** and touches both shapes

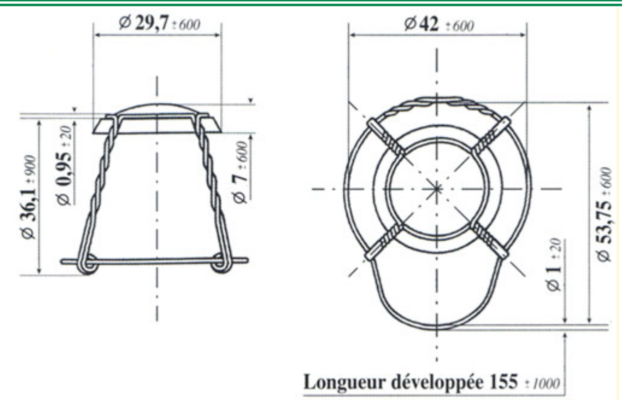
- Good measure if shapes are **b-compatible** (single point contacts)
  - Takes distance and orientation into account.
- Can be extended to incompatible shapes



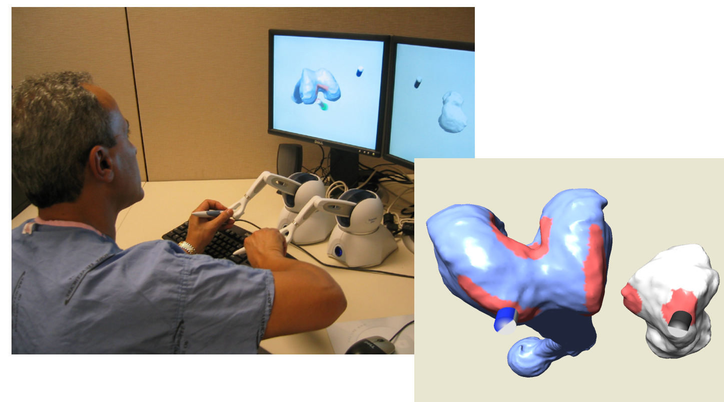


Applications

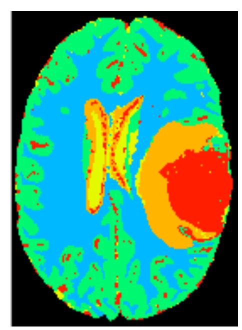
- Compare manufactured shape to nominal design



- Compare shape of weak cartilage to graft



- Compare stages in evolution of tumor or lake

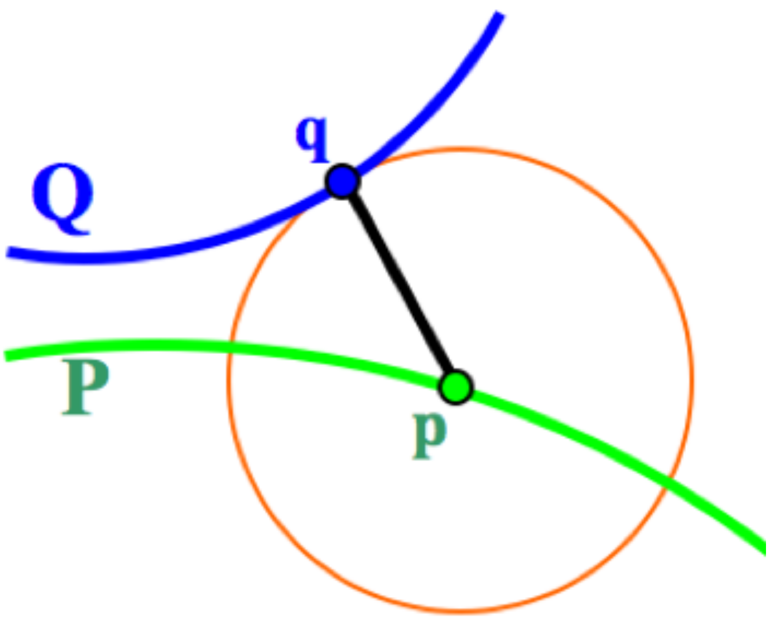


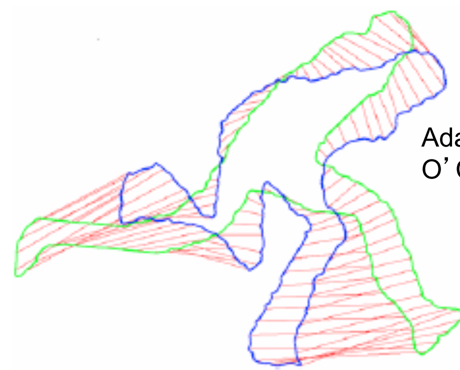
Old "Closest Point" c-map & c-morph

Most of prior work (e.g.: ICP) was based on the ***closest point map*** *or on intrinsic shape segmentation and matching (salient features, skeleton, parameterization…)*

$C_{P \to Q}$ maps **p∈P** to the set of its closest points **q∈Q**

c-morph moves **p** towards **q** at constant speed





Adamek,
O' Connor, Jones

Improving upon c-map and c-morph

- The c-map has drawbacks and limitations
- The new b-map addresses or reduces them

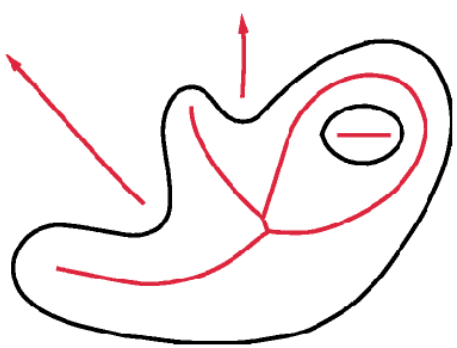
Cut locus

The **cut locus**, C(S), of S is the set of points with at least two closest points on bS.

- C(S) may be decomposed into
  - The **interior cut**, Ci(S): the part in S, which is M(S)
  - The **exterior cut**, Ce(S): the part in the complement S' of S

    Note that Ce(S)=Ci(S')

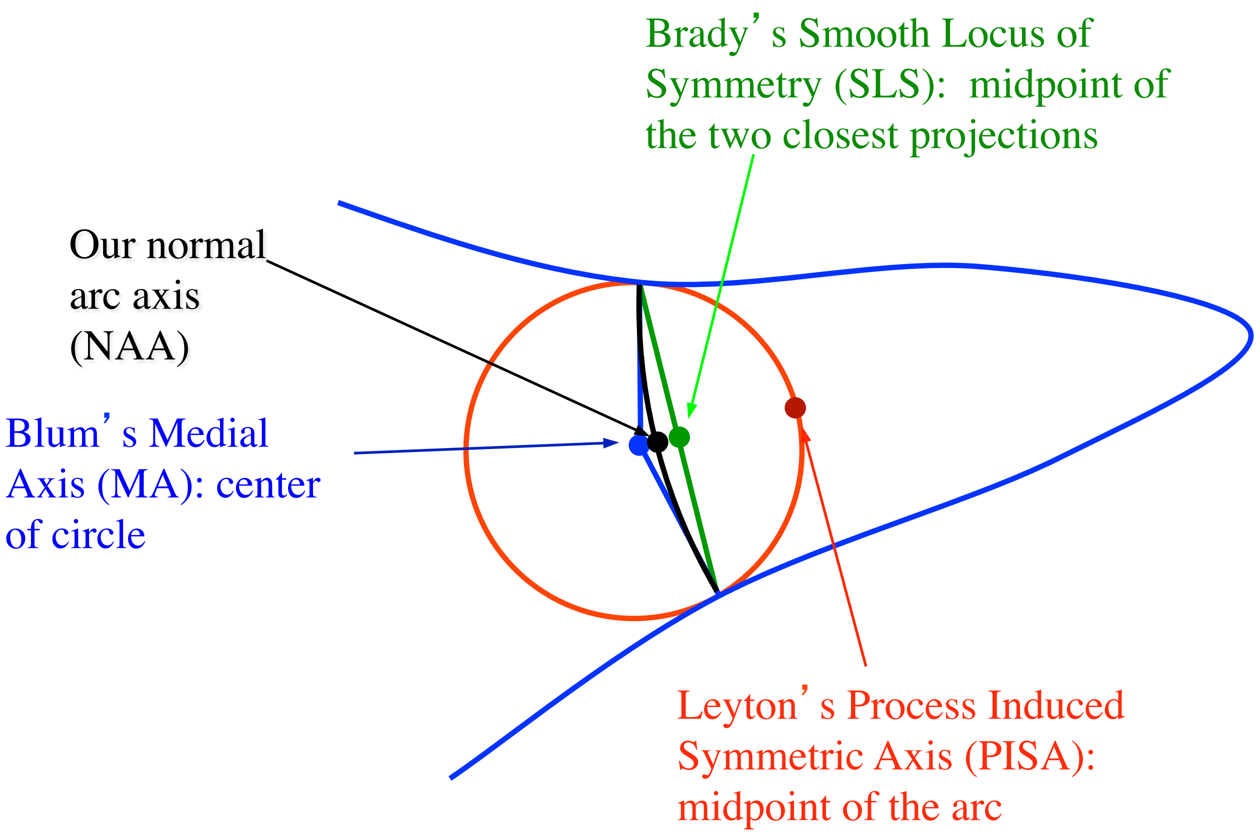
- Media Axis Transform MAT(S)
  - To each point p of Ci(S) we associate its distance r(p) from bS



http://www.lems.brown.edu/vision/Presentations/Wolter/

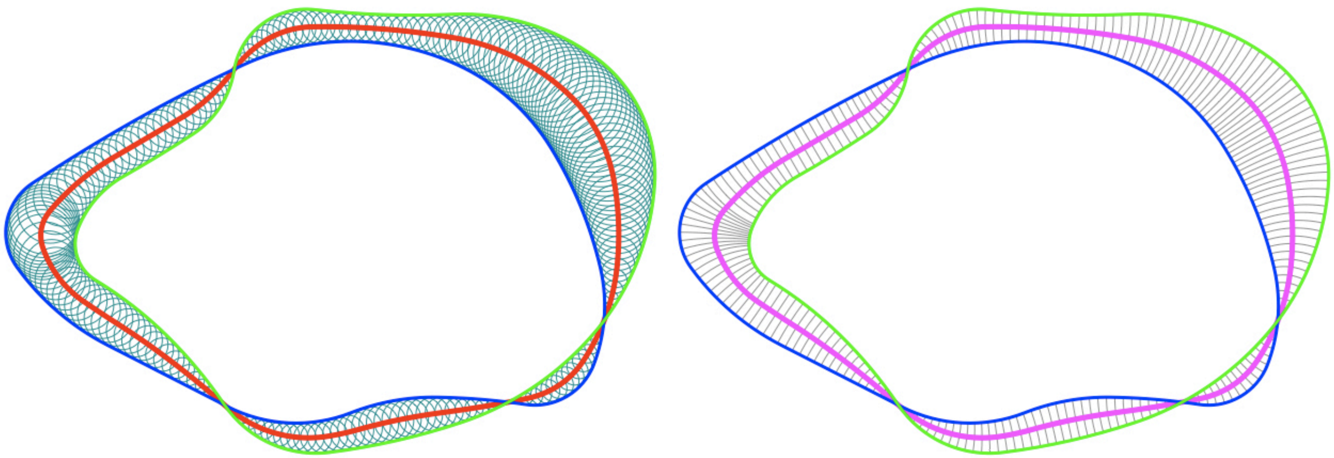
Variation of Medial Axis

Blum's Medial Axis (center of tangent disk)



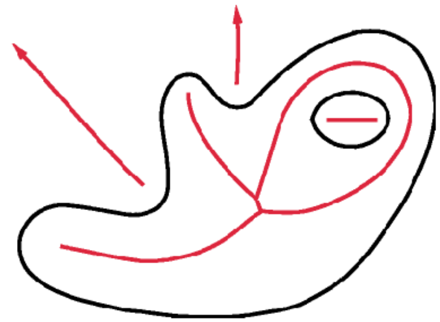
Median M and half-arc curve H

- The median **M** is the set of centers of the spheres of T_{PQ} if $h<f$
  - M has their topology
  - **M** is C^k if P and Q are C^k
- **H** is the locus of mid-arc curves, it is very close to **M**

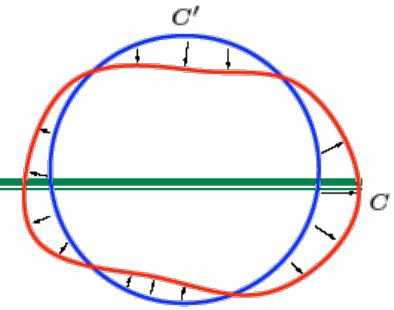


Local and Least Feature Size (lfs and LFS)

- The **Least Feature Size**, LFS(S), of a set S is the minimum of r(p) for all points p in C(S)
  - It is the distance between bS and C(S)
  - S is r-regular if LFS(S)=r
- The **local feature size**, lfs(p), of a point p with respect to a set S is the radius of the *largest* open ball that contains it and does not intersect bS:
  - lfs(p)=max r(q) for all q of C(S) such that $p \in B_{r(q)}(q)$
    - Extends Rupports definition of lfs to continuous sets
    - Corresponds to our definition of **r-regularity** at p

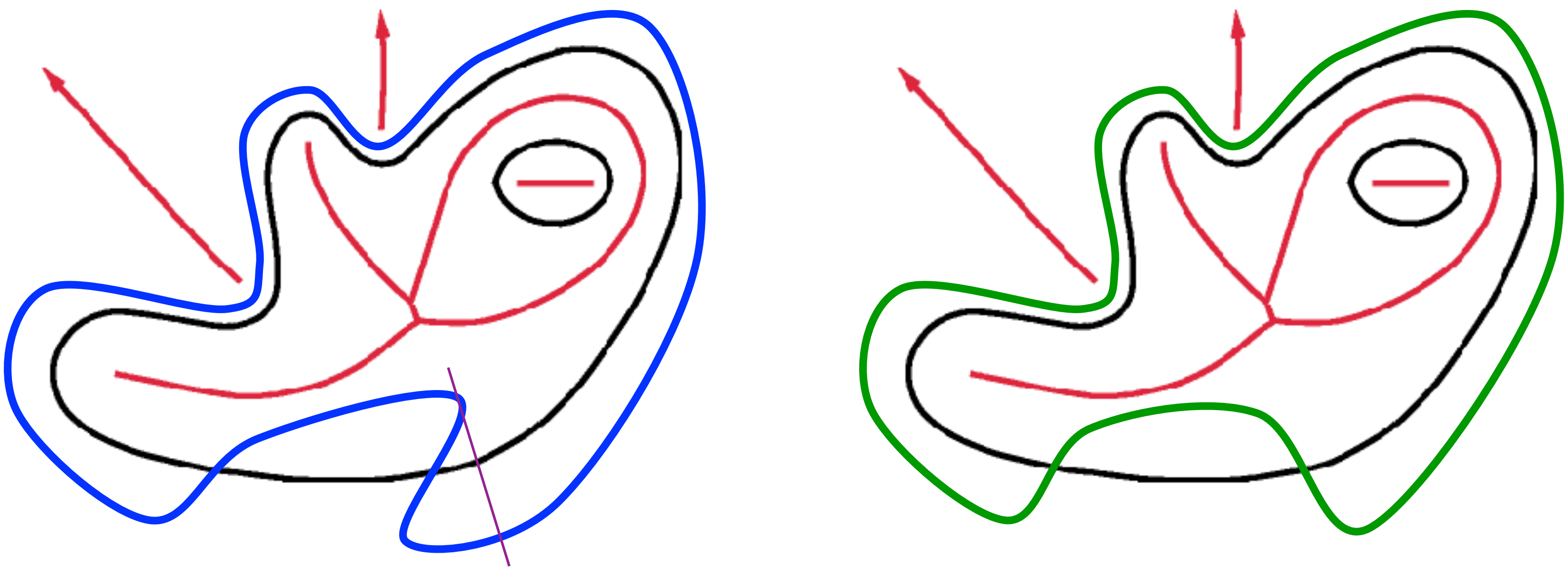


Normal Offsets of a manifold



- When is a set B the normal offset of set A?
  - When B may be obtained by displacing each point A(t) of the boundary of A along the normal N(t) by $h(t) \in [-h_i(t)\ , h_e(t)]$.
  - Hence, we must have Ci(A)⊂B and Ce(A)⊂B'

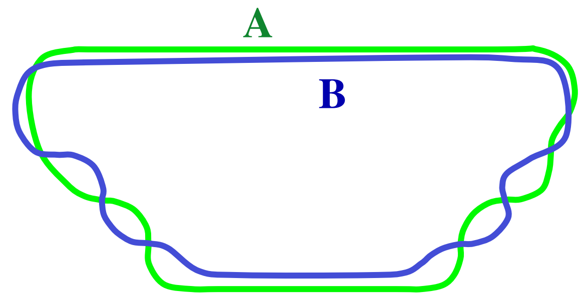
    and $N_B(s)$ cannot be orthogonal to B(s)–CP(B(s),bA)



C-compatible curves

Two manifolds A and B are c-compatible if

$$D_{Haus}(A,B) < c \max(LFS(A), LFS(B)), \text{ with } c = 2-\sqrt{2} \approx 0.58$$



$h<(2-\sqrt{2})f \;\Rightarrow\;$ P and Q are c-compatible

- h = Hausdorff distance $D_{Haus}(A,B)$ between **A** and **B**.
- h = smallest r such that $\mathbf{A} \subset \mathbf{B}^r$ and $\mathbf{B} \subset \mathbf{A}^r$,

  where $\mathbf{X}^r$ is the set of points at distance r or less from **X**.
- $f = \min(\, \mathrm{mfs}(\mathbf{A}) \,,\, \mathrm{mfs}(\mathbf{B})\,)$

  minimum feature size mfs(**X**) **=** largest r such that $\mathbf{X}=\mathbf{F}_r(\mathbf{X})$

  where $\mathbf{F}_r(\mathbf{X})$ is the set not reachable by open r-ball not intersecting **X**

C-map theorem

Theorem:

When A and B are **c-compatible,**

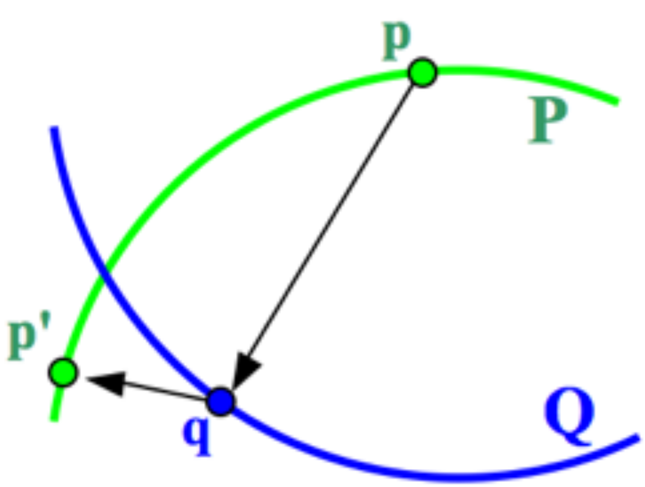
each is a **normal offset** of the other.

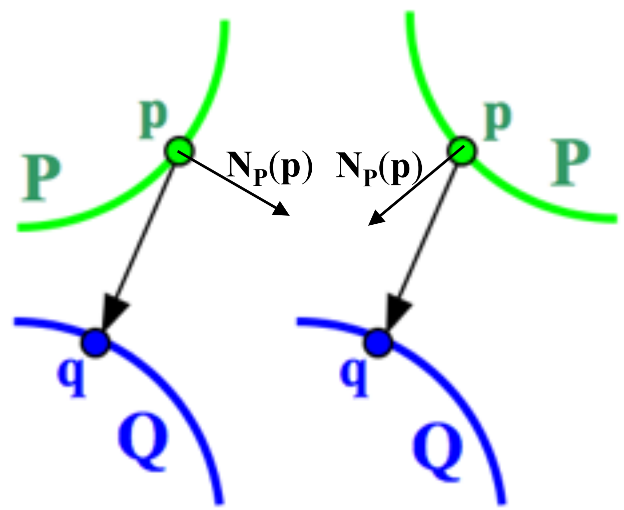
The definition of "c-compatible" and this theorem are true for smooth curves in 2D, surfaces in 3D, and hypersurfaces in nD!

C-map drawbacks

- Not symmetric: $C_{\mathbf{Q}\to\mathbf{P}}(C_{\mathbf{P}\to\mathbf{Q}}(\mathbf{p}))\neq\mathbf{p}$
- Insensitive to orientation: same **q**, $\forall \mathbf{N}_{\mathbf{P}}(\mathbf{p})$
- Unnecessary distortion
- Not all pairs are c-compatible
  - Would like a more forgiving condition than

    $H(A,B) < c \max(LFS(A), LFS(B))$, with $c = 2-\sqrt{2}$

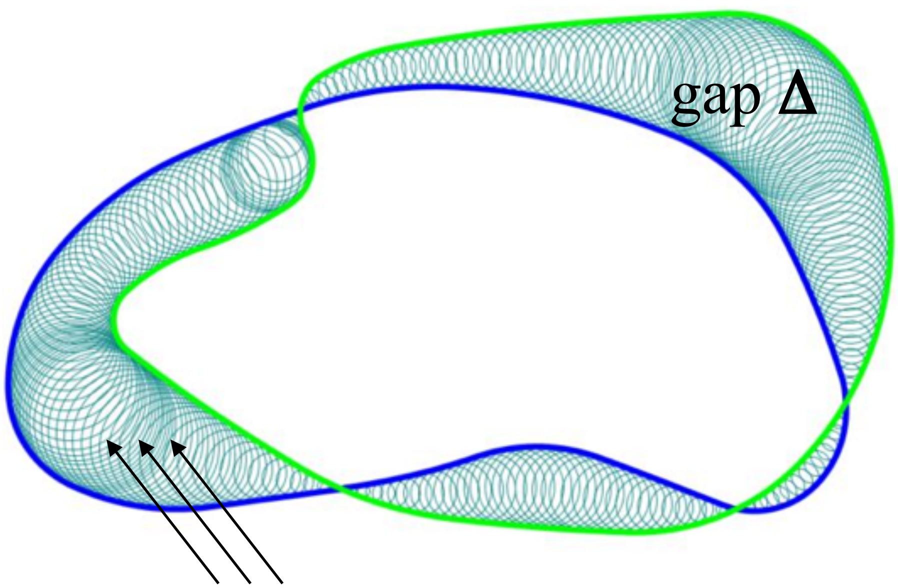


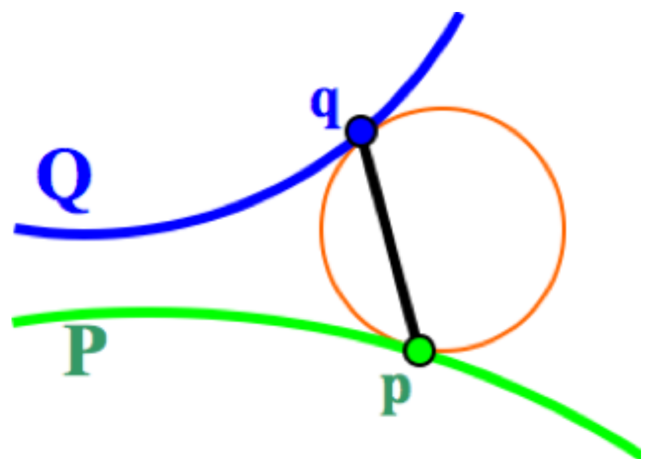


Our "Tangent Ball" **b-map**

- **Δ= (iP⊕iQ)** ∪ **(P∩Q)** is the **gap** between the two shapes
- T_{PQ} is the set of **disks/balls** in **Δ** that touch both **P** and **Q**
- $B_{P\leftrightarrow Q}$ maps **S∩P** to **S∩Q** for each disk **S** of T_{PQ}

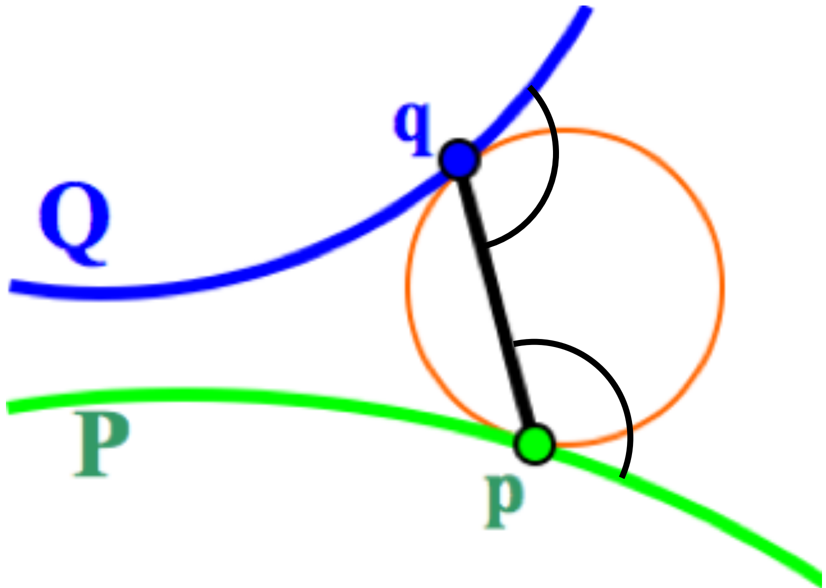
  Definition works in 2D (disks) and 3D (balls)





B-map advantages

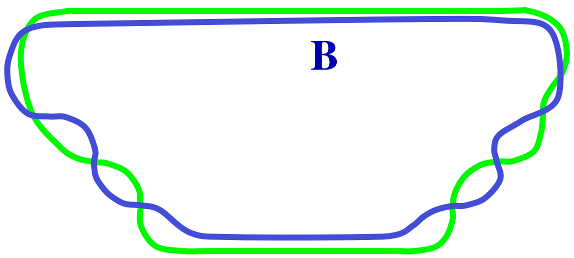
- $B_{\mathbf{P}\leftrightarrow\mathbf{Q}}$ is symmetric: $B_{\mathbf{Q}\leftrightarrow\mathbf{P}}(B_{\mathbf{P}\leftrightarrow\mathbf{Q}}(\mathbf{p}))=\mathbf{p}$
- $B_{\mathbf{P}\leftrightarrow\mathbf{Q}}$ is sensitive to the orientation of both shapes



B-compatible curves

Two curves A and B are **b-compatible** if

$$H(A,B) < \max(LFS(A), LFS(B))$$



- $h < f \;\Rightarrow\;$ P and Q are b-compatible

  h = **Hausdorff distance** between **A** and **B**.

  h = smallest r such that $\mathbf{A} \subset \mathbf{B}^r$ and $\mathbf{B} \subset \mathbf{A}^r$,

  where $\mathbf{X}^r$ is the set of points at distance r or less from **X**.

  $f = \min(\, \mathrm{mfs}(\mathbf{A}),\ \mathrm{mfs}(\mathbf{B})\,)$

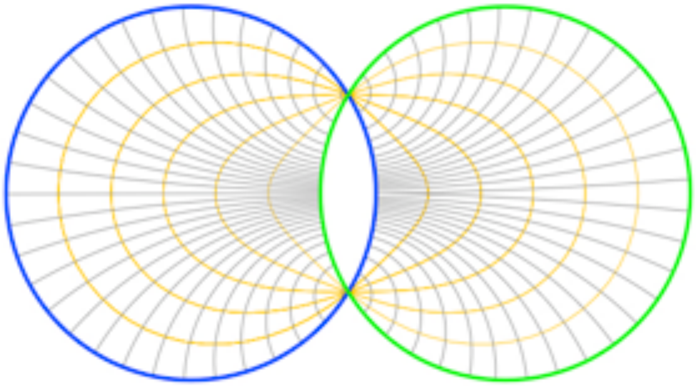
  minimum feature size mfs(**X**) = largest r such that $\mathbf{X} = \mathbf{F}_r(\mathbf{X})$

  where $\mathbf{F}_r(\mathbf{X})$ is the set not reachable by open r-ball not intersecting **X**

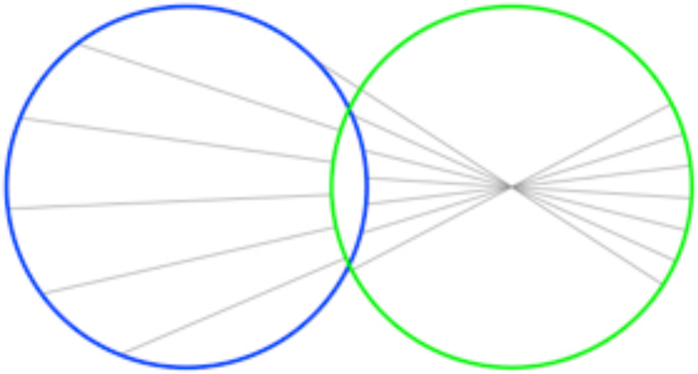
- *$h < f$ is less constraining than: $h < 0.58\, f$* $\;\Rightarrow\;$ **P** and **Q** are c-compatible

Comparing compatibility conditions

- **P** and **Q** are **c-compatible** when both $C_{\mathbf{P}\to\mathbf{Q}}$ and $C_{\mathbf{Q}\to\mathbf{P}}$ are homeomorphisms
- c-compatibility implies b-compatibility
- **P** and **Q** are **b-compatible** when $B_{\mathbf{P}\leftrightarrow\mathbf{Q}}$ is a homeomorphism



b-compatible



not c-compatible

Complexity measures for 2D shapes

Many measures of complexity are useful in different contexts:

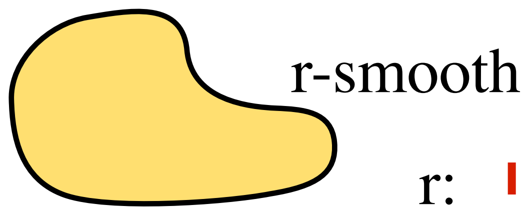
- **Transmission**: compressed file size
  - **Kolmogorov**: Length of data and program
- **Processing**: Number of bounding elements
- **Algebraic**: Polynomial degree of bounding curves
- **Visibility**: Number of guards needed
- **Graphics**: Number of intersections with "random" ray

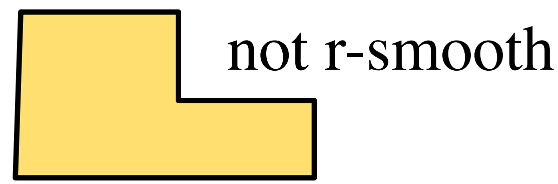
We focus primarily on morphological ones:

- **Sharpness**: curvature, high frequency energy
- **Regularity**: Minimal feature size
- **Topology**: Number of components, holes (, genus in 3D)
- **Tightness**: Perimeter length / surface area

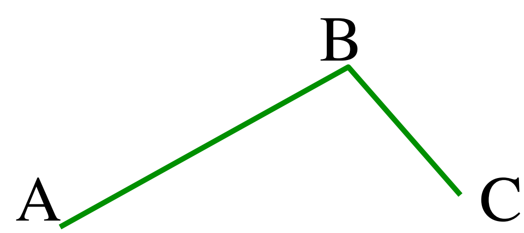
r-smoothness

- A shape S is r-smooth if the curvature of every point B in its boundary bS exceeds r





- How to check for r-smoothness at B?
  – For C_2 curves: compare radius of curvature to r
  – For **polygons**: estimate radius of curvature
    • $R=v^2/(a\times v)_z$, where v=AC/2 and a=BC–AB

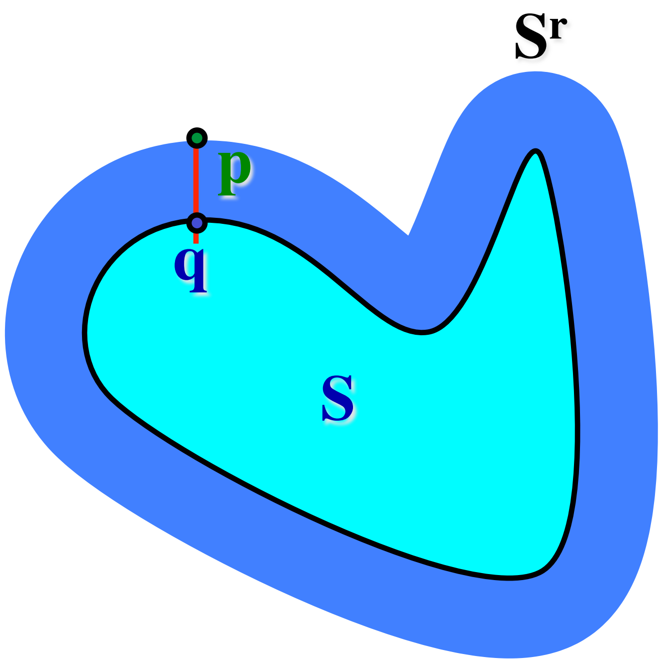


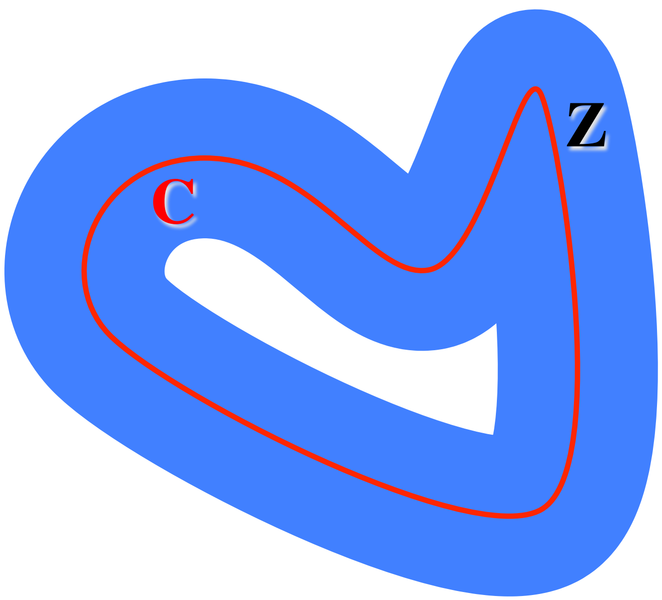
Dilation S^r and tolerence zone Z

The **dilation** $\mathbf{S}^r$ of **S** (also denoted $\mathbf{S}\uparrow r$) = { **p** : $D_{Min}(\mathbf{p},\mathbf{S})$ |≤ r }

= r-offset = Minkowski sum with a ball of radius r

The dilation of a curve **C** defines a **tolerance zone** Z around it

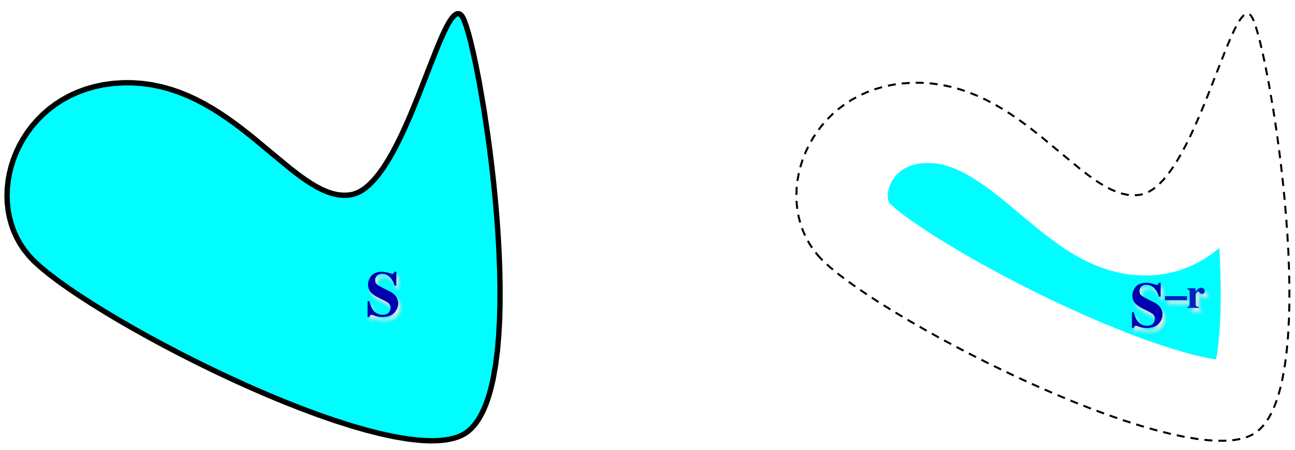




Contraction S^{-r}

The **contraction** of **S**, denoted $\mathbf{S}^{-r}$ and also $S\downarrow r$, is $\mathbf{S} - (b\mathbf{S})^r$

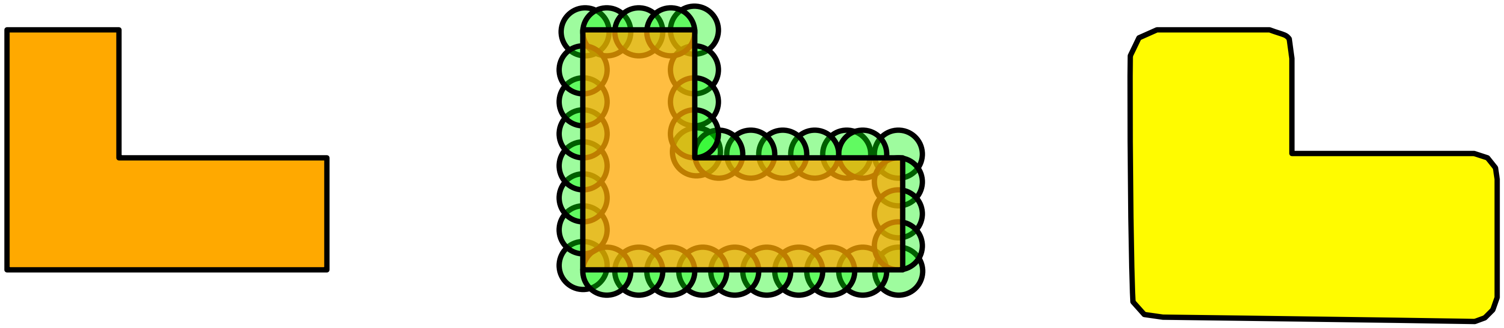
Remove all points within distance r from the boundary b**S**



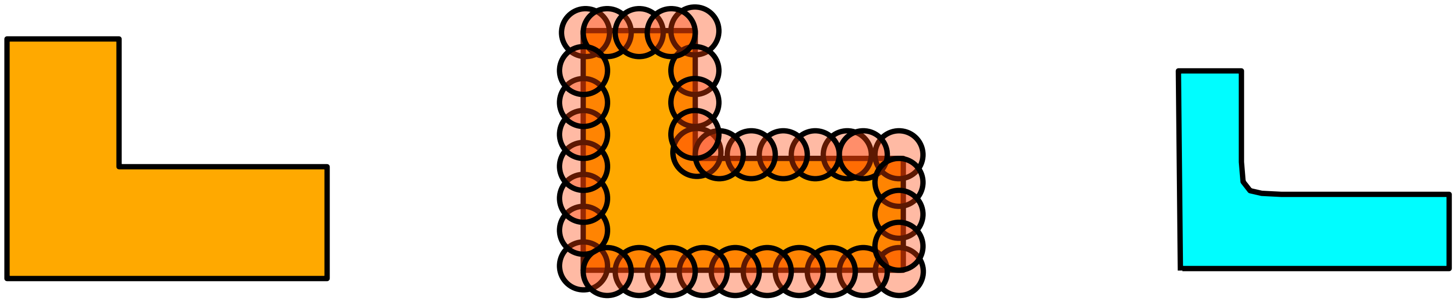
The contraction of curves and isolated points is empty.

Growing and shrinking

- To grow a shape by r means to add all points within distance r
- The grown version of S will be denoted S^r or $S\uparrow r$
- It is the union of all balls $B_r(c)$ of radius r whose center c is in S

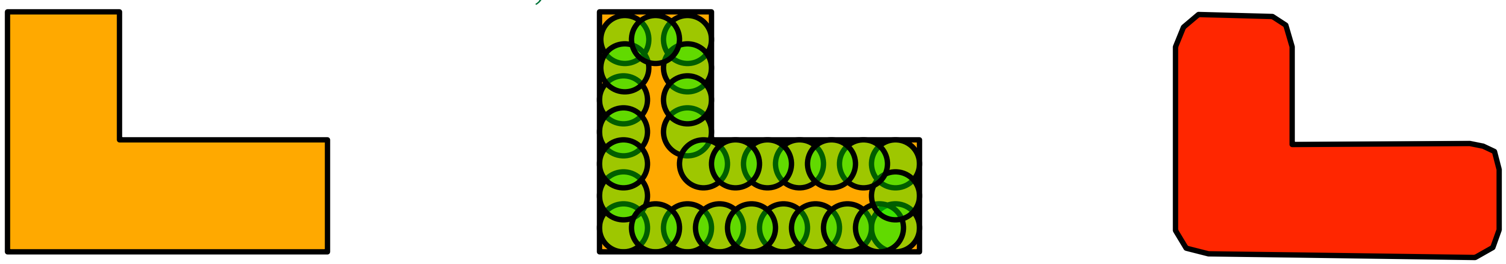


- To shrink a shape by r means to subtract all points within distance r from its boundary
- The shrunken version of S will be denoted S^{-r} or $S\downarrow r$
- It is the space that cannot be reached by $B_r(c)$, when c is out of S

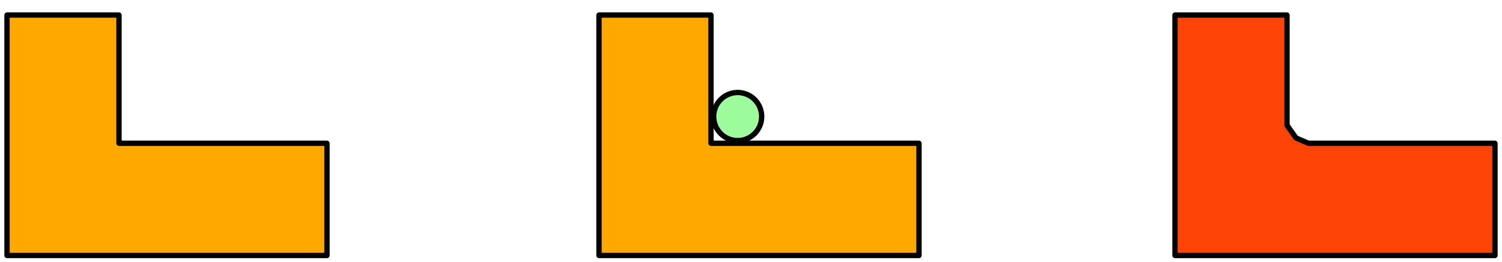


Rounding and Filleting

- Rounding $R_r(S) = (\mathbf{S}\!\downarrow\!\mathbf{r})\!\uparrow\!\mathbf{r}$, **"opening"**
  - Union of all balls B_r in S
  - Rounds convex corners, removes hair

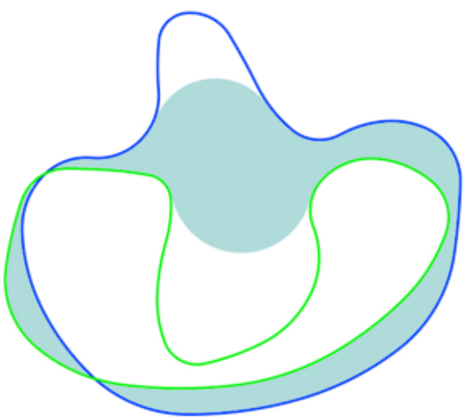


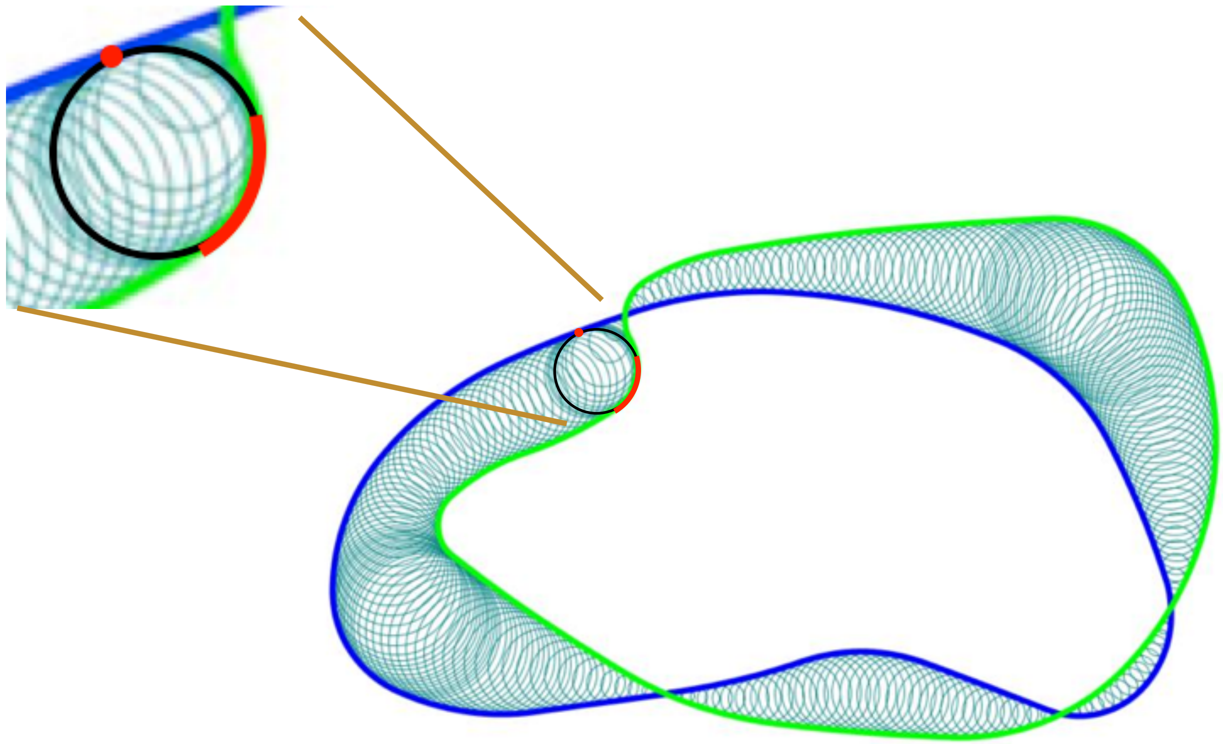
- Filleting $F_r(S) = (\mathbf{S}\!\uparrow\!\mathbf{r})\!\downarrow\!\mathbf{r}$, **"closing"**
  - Region not reachable by B_r that does not interfere with S
  - Rounds concave corners, fills cracks



The canal C in the gap Δ

- The **canal** C is the union of all spheres of T_{PQ}
- P and Q are **b-compatible** $\Rightarrow$ C=Δ
- (C=Δ) && (not b-compatible) $\Rightarrow$ **quasi-compatible**





Definition of relative rounding

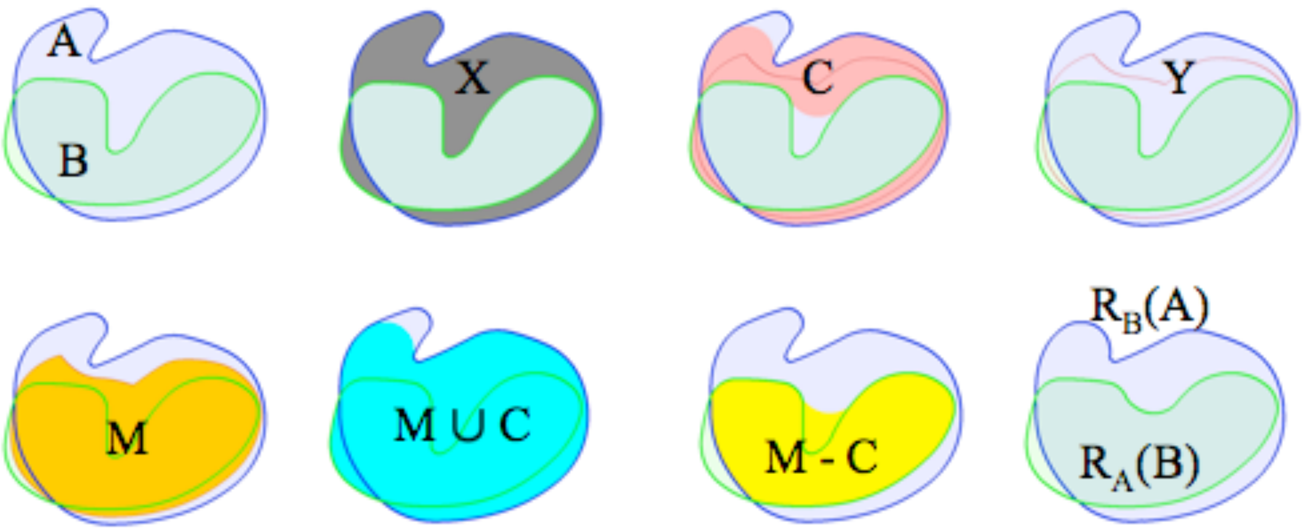
- The relative rounding of A with respect to B is

$$R_B(A)=A(M+C)+(M–C)$$

Moat X=(A–iB)+(B–iA) is the extended symetric difference

Canal C = union of maximal balls in X that intersect bA and bB

Mean M is set of points closer to iAiB than to !(A+B)

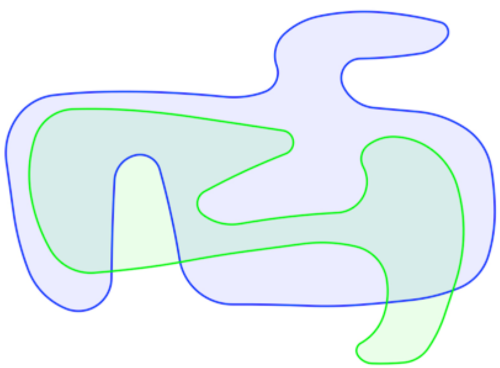


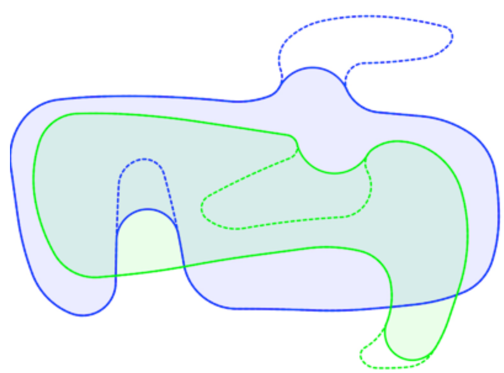
Relative b-roundings **P'** and **Q'**

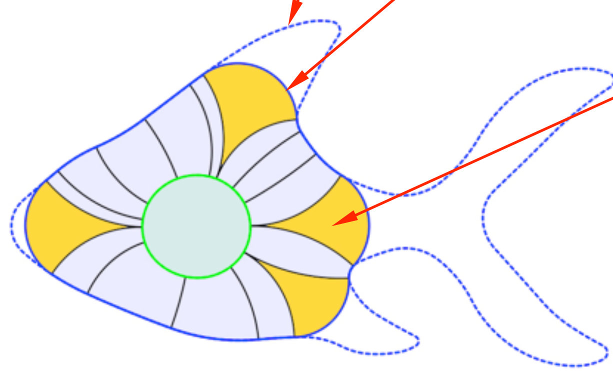
Modified lacing identifies **incompatible features** that cannot be reached by any sphere of T_{PQ}

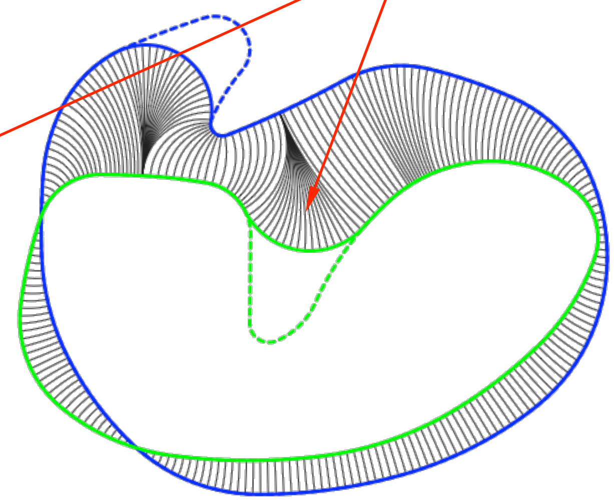
It replaces each one of them by a circular **cap**

The b-morph becomes one-to-many or many-to-one in the **fans**









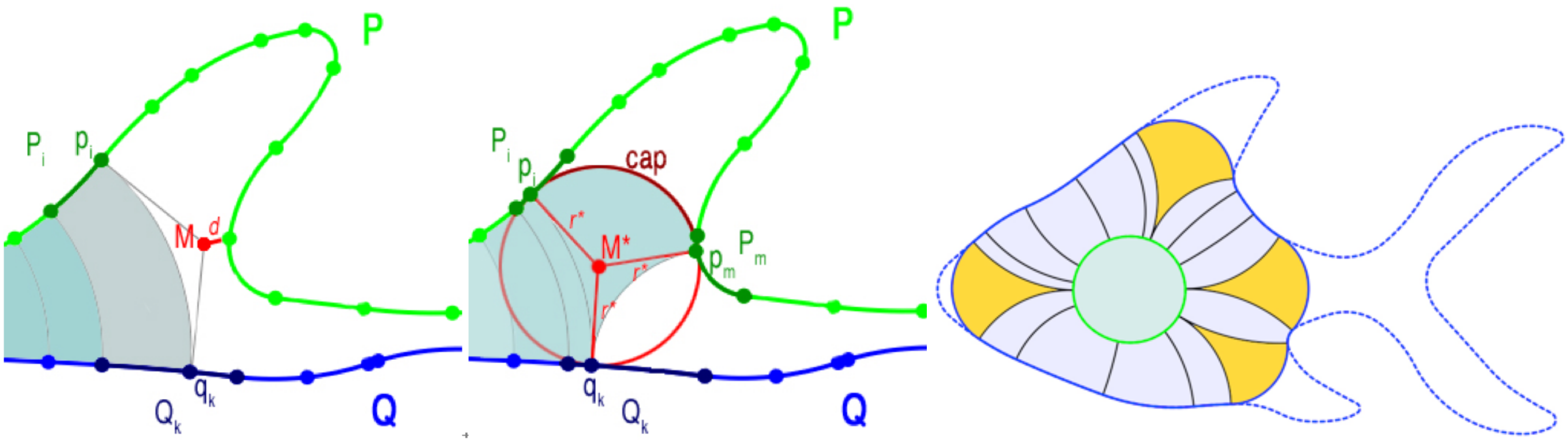
P' is a **tightening** of **P** in (**Δ–C**)∪(**P**∩**Q**), obtained by tightening the incompatible features, until they snap to the corresponding caps.

b-rounding via modified lacing

if Dist(**m**,**Q**)<r, find center **m*** and radius **r*** of a circle **K** in Δ that is tangent to $\mathbf{P}_i$, to $\mathbf{Q}_k$, and to another edge $\mathbf{P}_m$ of **P**

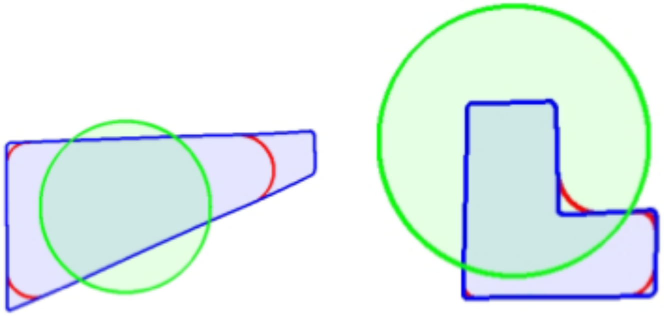
(the first one found along P)

The portion of **K** between $\mathbf{p}_i$ and $\mathbf{p}_m$ is the **cap** and replaces the corresponding incompatible portion of **P**.

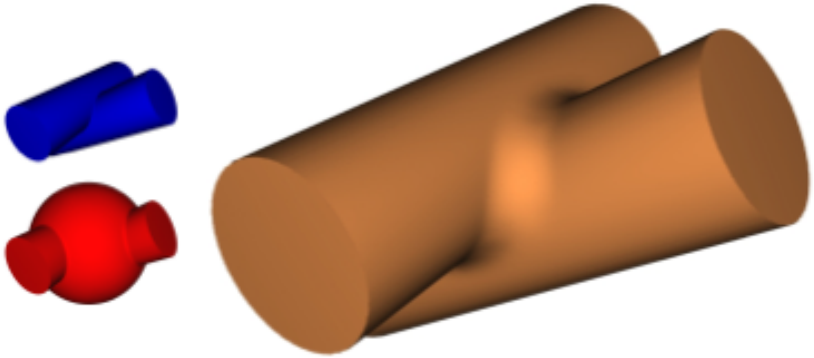


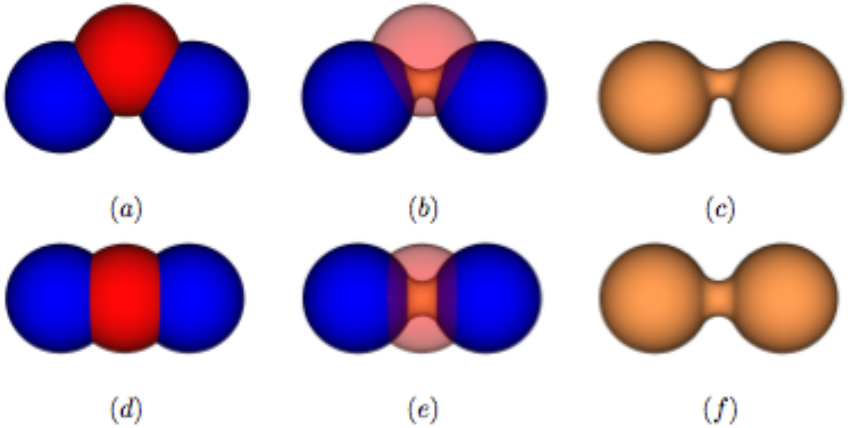
Examples of relative rounding

- Use green shape to control
- what is rounded and by how much

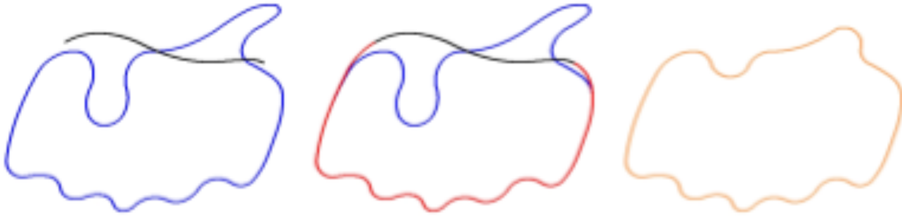


- Works the same in 3D





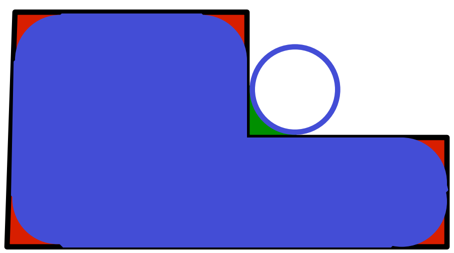
- Can use local control shape



r-regularity

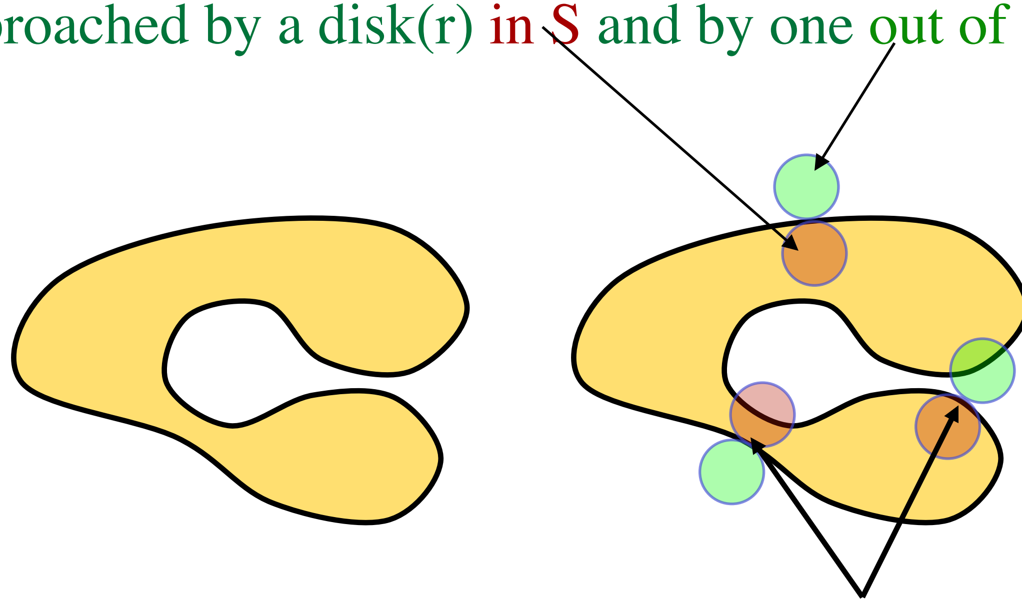
A shape S is r-regular if $S=F_r(S)=R_r(S)$

- $F_r(S) = S{\uparrow}r{\downarrow}r$, r-Fillet (closing) = area not reachable by r-disks out of S
- $R_r(S)= S{\downarrow}r{\uparrow}r$, r-Rounding (opening) = area reachable by r-disks in S
- Each point of bS can be approached by a disk(r) in S and by one out of S



Original L is not r-regular

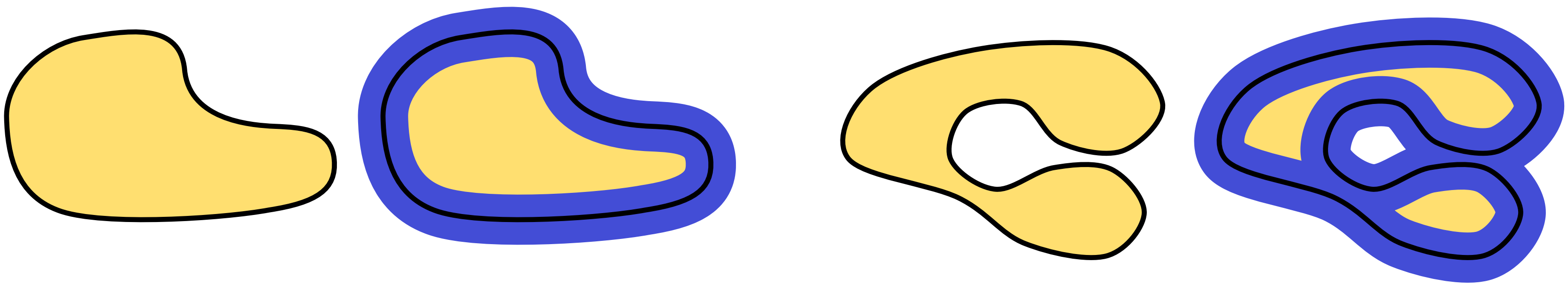
Removing the red and adding the green makes it r-regular



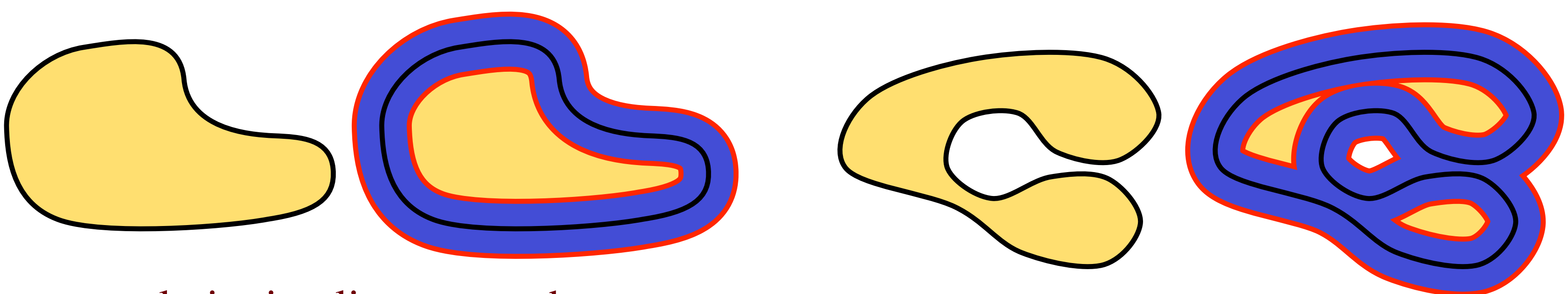
This shape is not r-regular

Properties of r-regular shapes

- Boundary is resilient to thickening by r
  - bS can be recovered from its rendering as a curve of thickness 2r



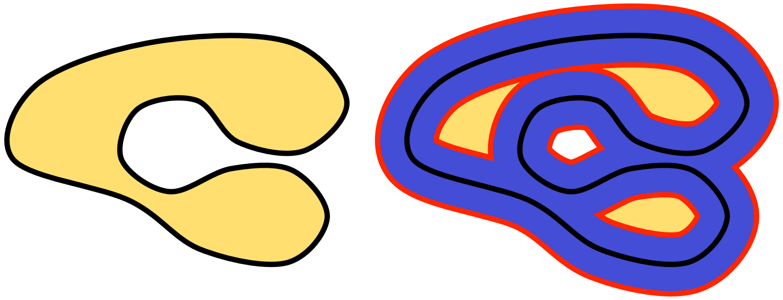
- One-to-one mapping from boundary to its two offsets by r
  - The boundary of S↑r (resp. S↓r) may be obtained by offsetting each point of bS along the outward (resp. inward) normal. No need to trim.



- r-regularity implies r-smoothness

When is a point of bS regular?

- How to check for r-regularity at B?
  - Check whether offset points is at distance r from bS
    - Dist(B±rN,bS)<r ?

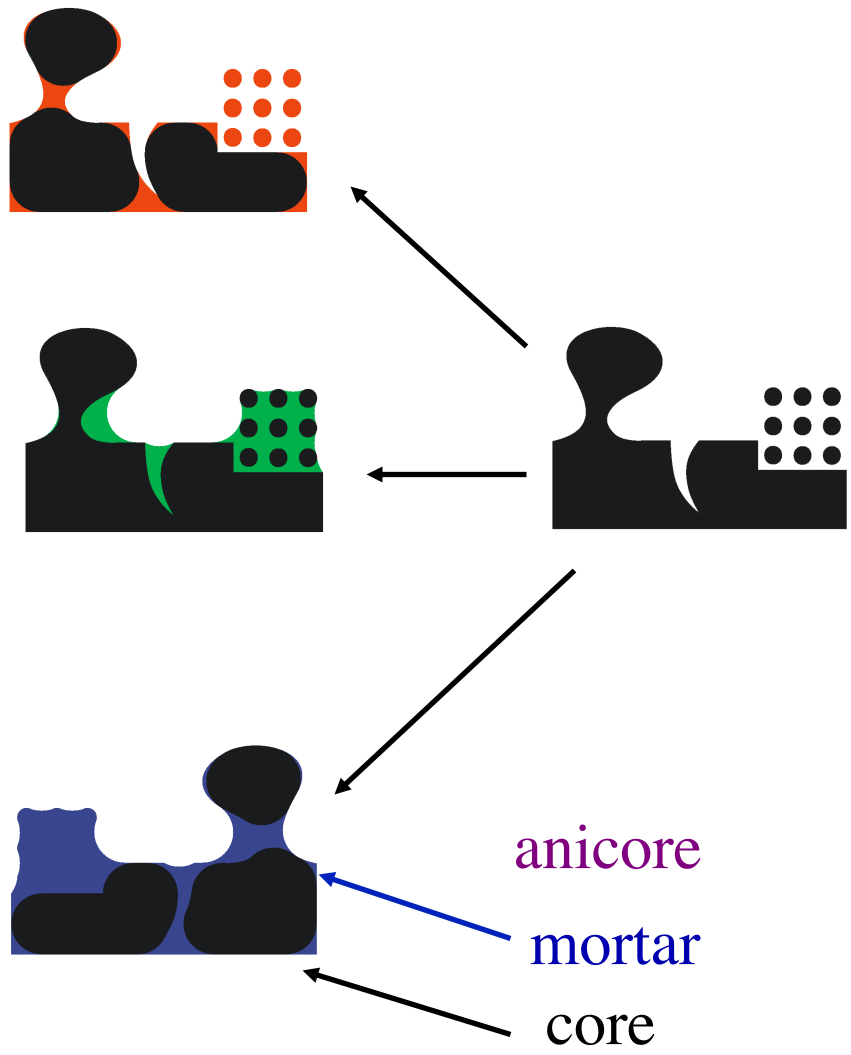


  - Want a set-theoretic definition?

- B is r-regular if it does not lie in the mortar $M_r(S)$
    - The **mortar** is the set of all points that are not r-regular
    - It is defined in the next slide

Core, Fill, Anticore, Mortar

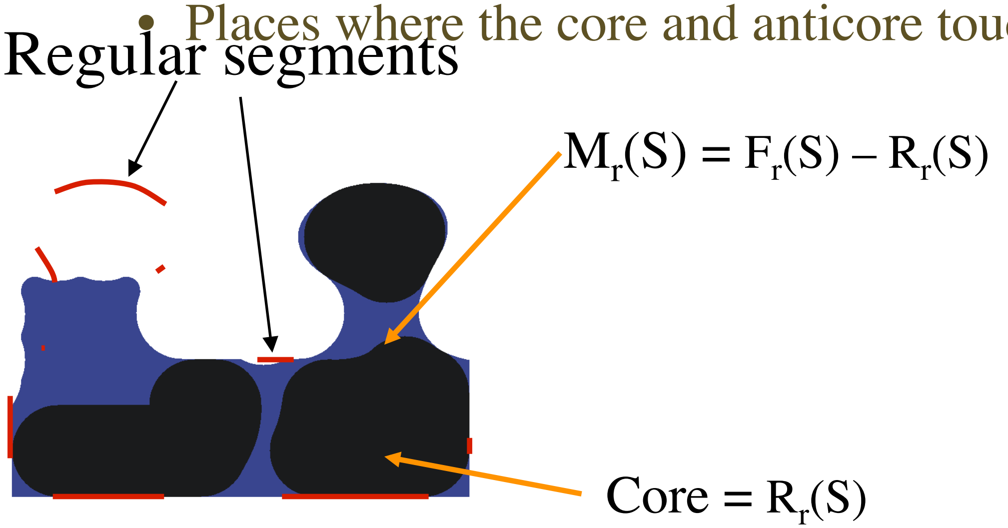
- Core: $R_r(S)$
  - $\cup B_r \subset S$
  - Removes the red
- Fill: $F_r(S)$
  - $(\cup B_r \subset S')'$
  - Adds the green
- Anticore: $(F_r(S))'$
  - $\cup B_r \subset S'$
  - Not black or green
- Mortar: $M_r(S)=F_r(S)-R_r(S)$
  - Green plus red
  - It is also $F_r(bS)$

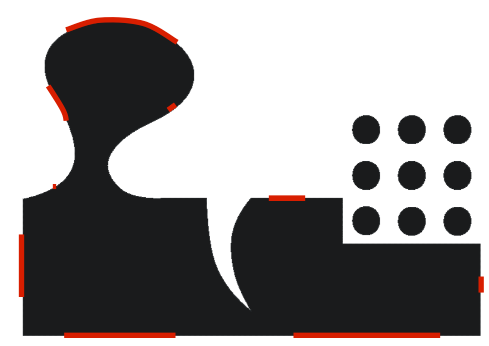


Using the Mortar to decompose bS

Given r, the **regular segments** of bS are defined as the connected components of r-regular points of bS.

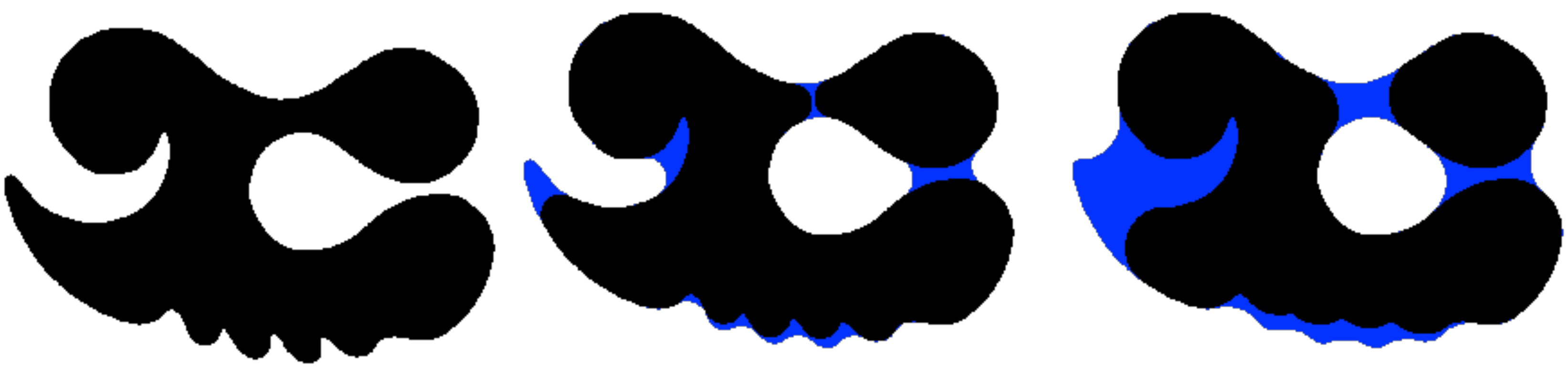
- Th1: Regular segment = connected component of bS–$M_r(S)$
  - Places where the core and anticore touch





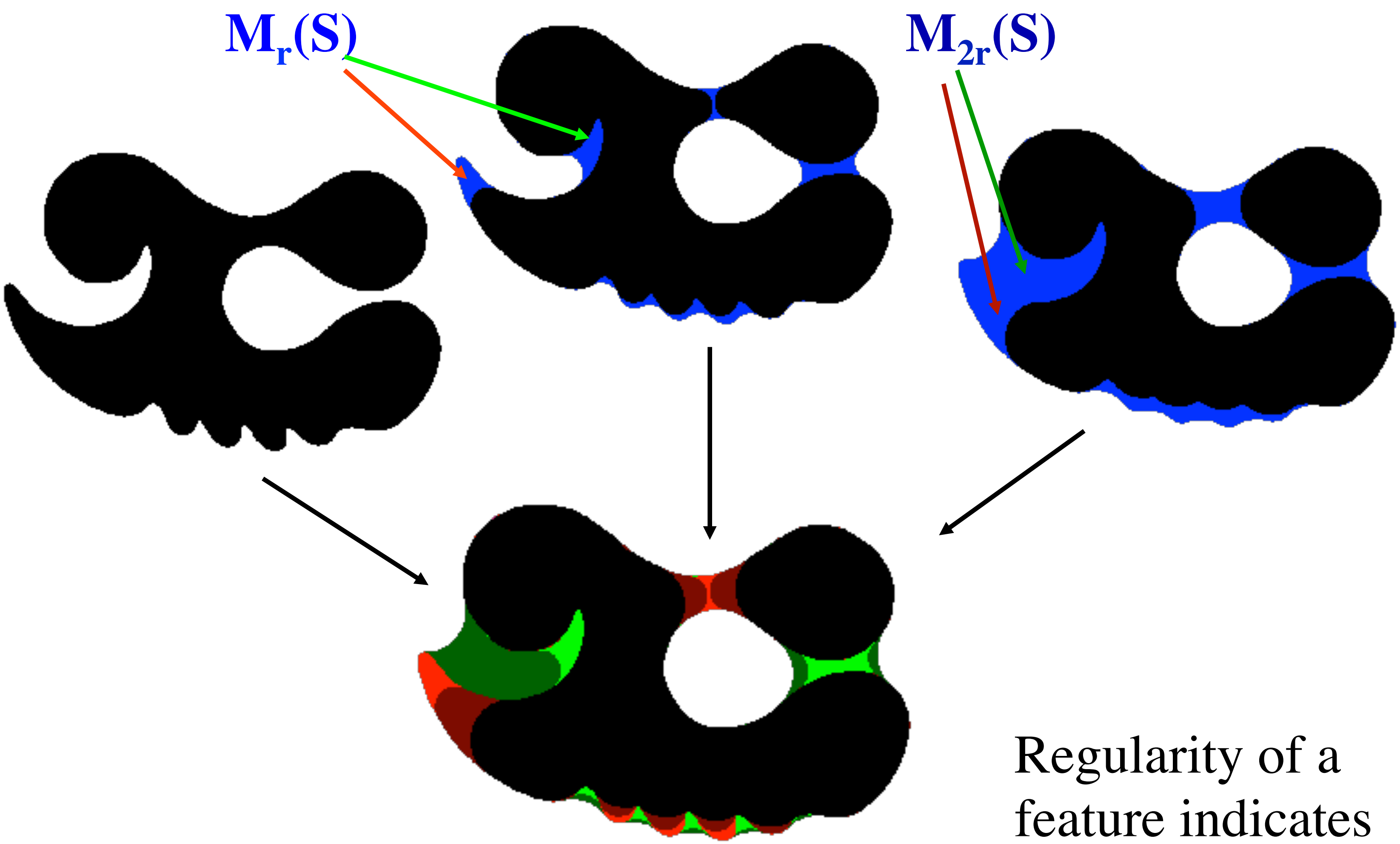
Properties of the mortar

- All points of the mortar are closer than r to the boundary bS
  – Restricting the effect of **simplification** to the **mortar** will ensure that we do not modify the shape in places **far from its boundary**
- The mortar excludes r-regular regions
  – Restricting the effect of **simplification** to the **mortar** will ensure that **regular portions** of the boundary are **not affected**



S $\qquad$ $M_r(S)$ $\qquad$ $M_{2r}(S)$

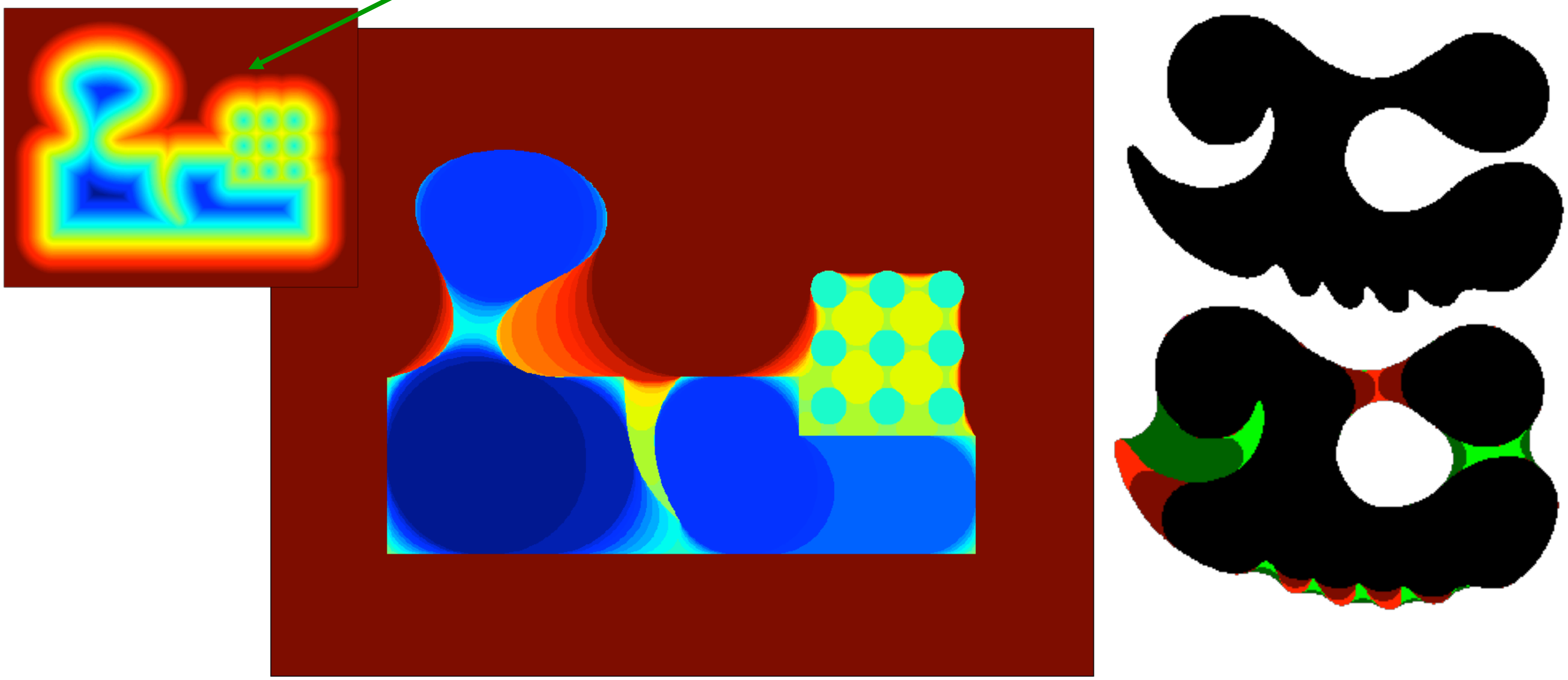
Mortar for multi-resolution analysis of space



Regularity of a feature indicates its “thickness”

Analyzing the regularity of space

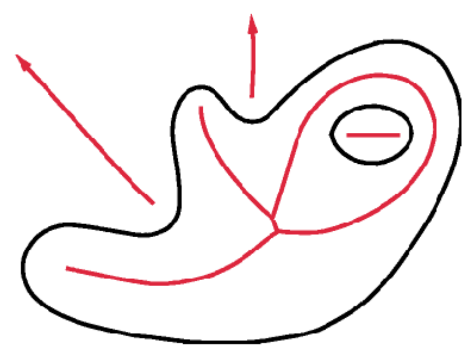
- The regularity of a point p with respect to a set S is defined as the minimum r for which $p \in M_r(S)$
  - Points close to sharp features or constrictions are less regular
  - Different from signed distance field



Medial Axis (Transform)

- **The** medial axis **M(S) of a set S is the locus of centers of disks in S that touch the boundary bS of S in at least 2 points. (Blum)**
- **The** cut locus **C(S) of S is the set of points with at least two closest points on bS.**
  - C(S) may be decomposed into
    - The interior cut, Ci(S): the part in S, which is M(S)
    - The exterior cut Ce(S): the part in the complement S' of S
    - Ce(S)=Ci(S')
- **Media Axis Transform MAT(S)**
  - To each point p of Ci(S)

    we associate its distance r(p) from bS
  - S is the union of disks $B_{r(p)}$(p) for all p in C
    - S' is the union of $B_{r(p)}$(p) for all p in Ce(S)



http://www.lems.brown.edu/vision/Presentations/Wolter/

Height fields in shape

- Distance Transform

  Height at P = distance from P to bS

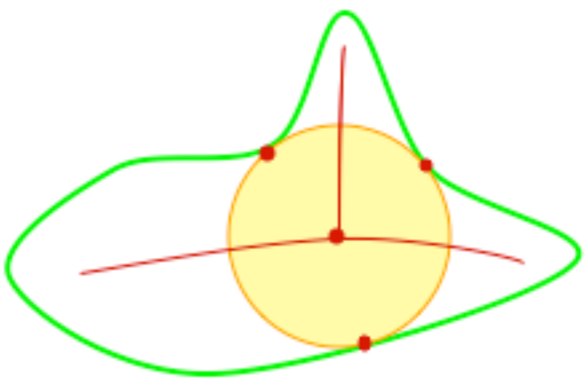
- Stability Transform

  Height at P = radius of largest disk in S that contains P

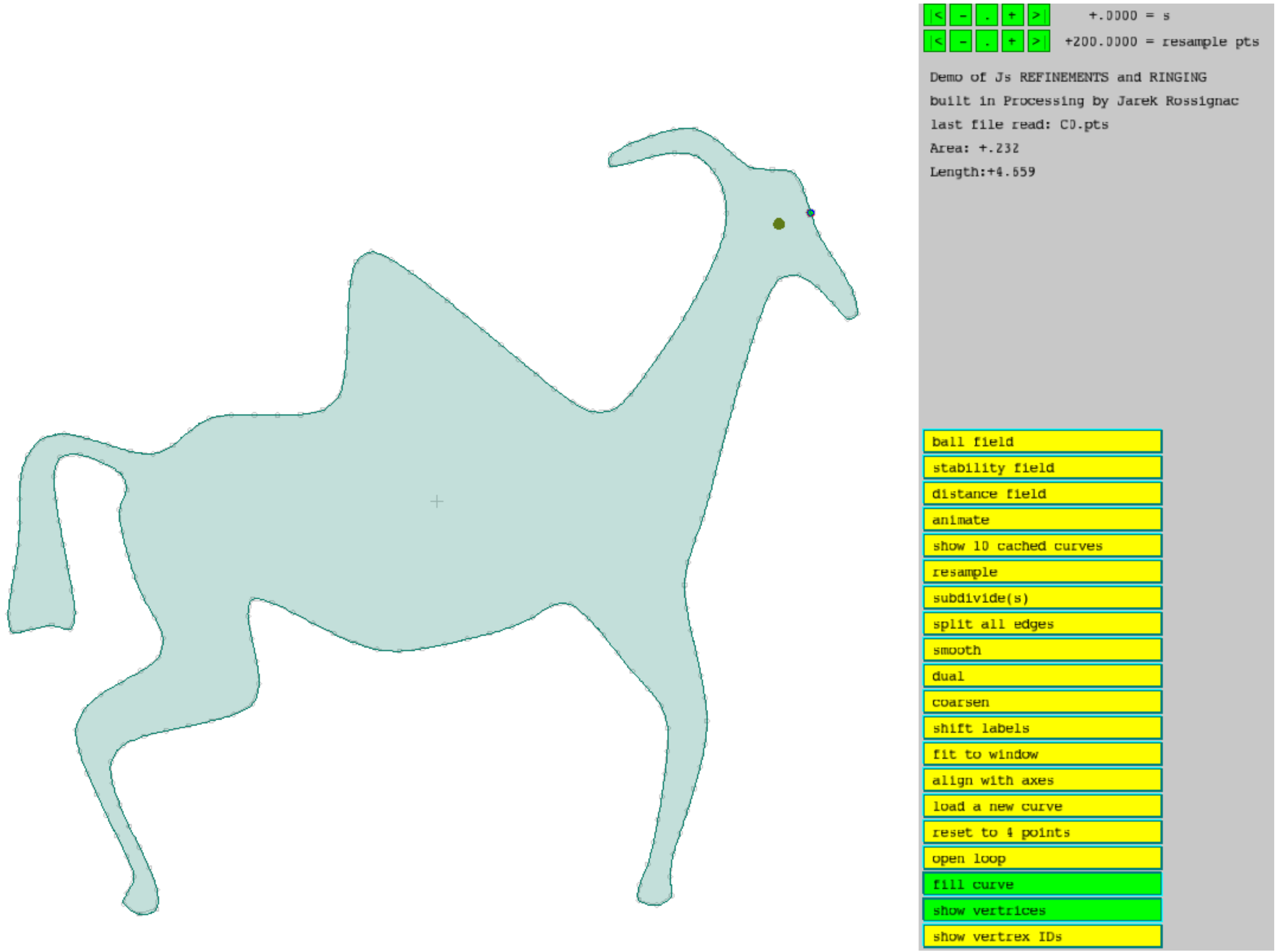
- **Ball Transform *NEW***

  Height at P = height at P of largest disk in S covering P

  Union of all semi-spheres with base disk in S

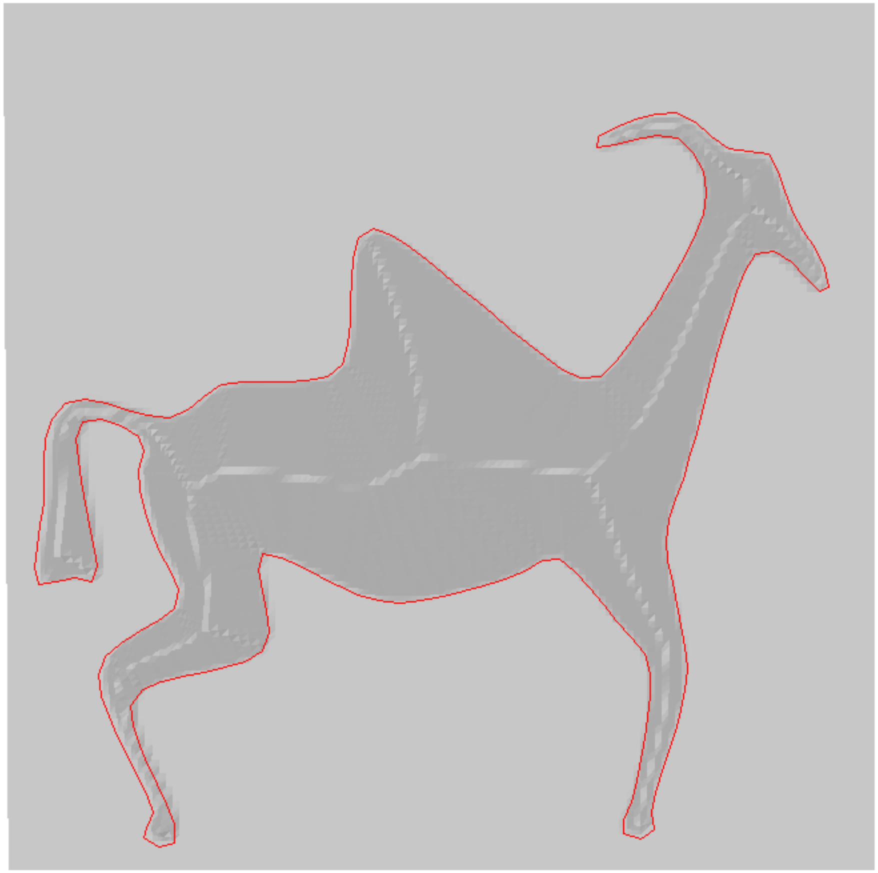


Curve

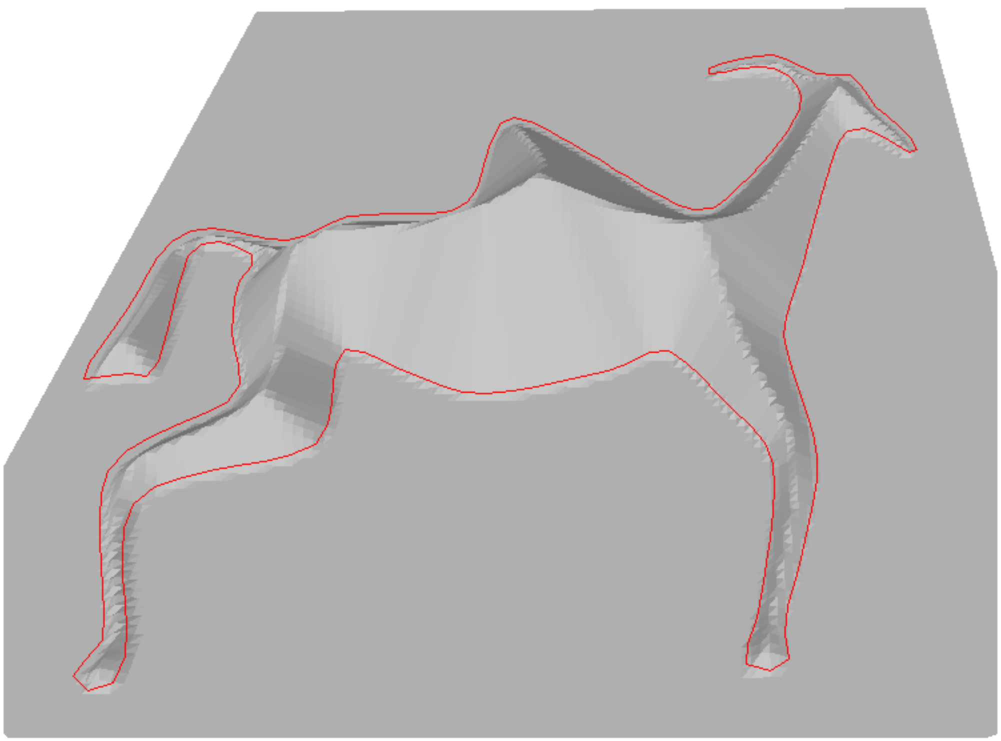


+.0000 = s
+200.0000 = resample pts
Demo of Js REFINEMENTS and RINGING
built in Processing by Jarek Rossignac
last file read: C0.pts
Area: +.232
Length:+4.659
ball field
stability field
distance field
animate
show 10 cached curves
resample
subdivide(s)
split all edges
smooth
dual
coarsen
shift labels
fit to window
align with axes
load a new curve
reset to 4 points
open loop
fill curve
show vertrices
show vertrex IDs

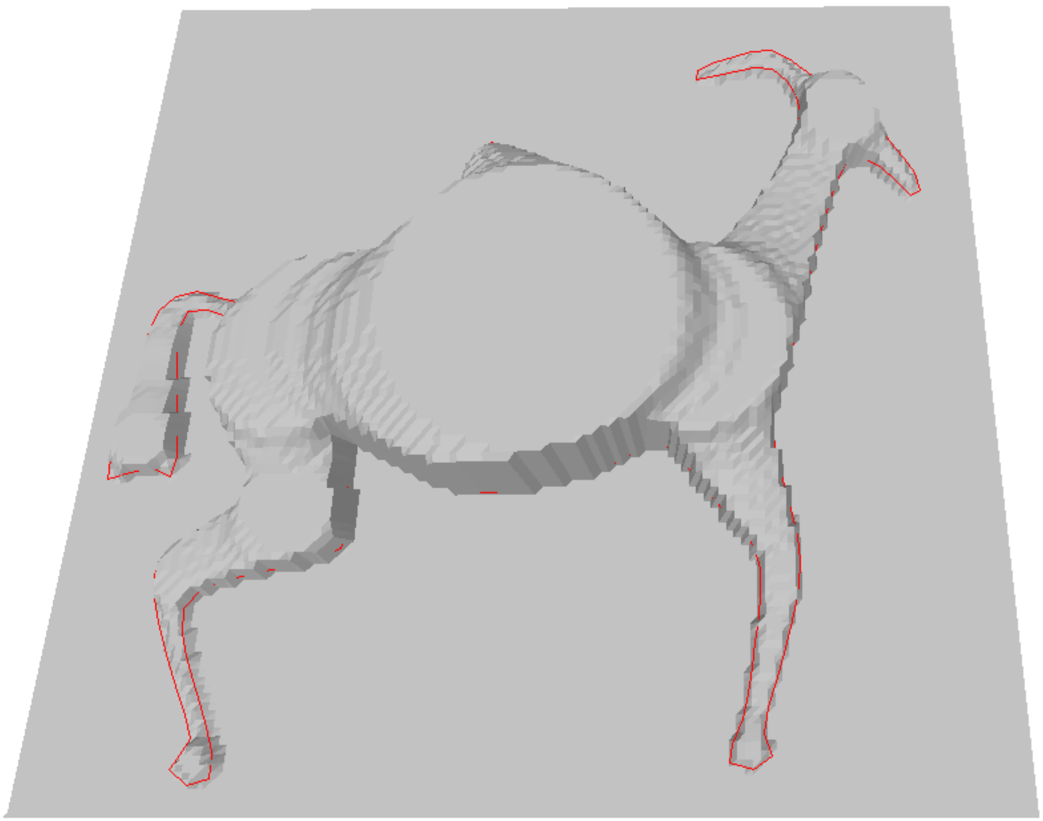
Distance Transform



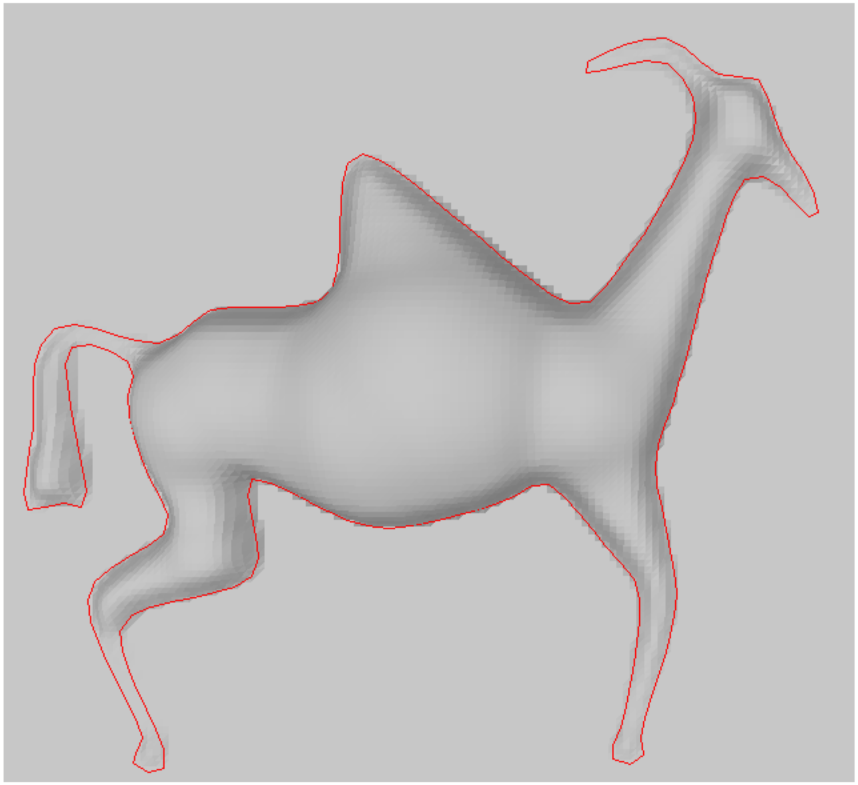
Distance Transform



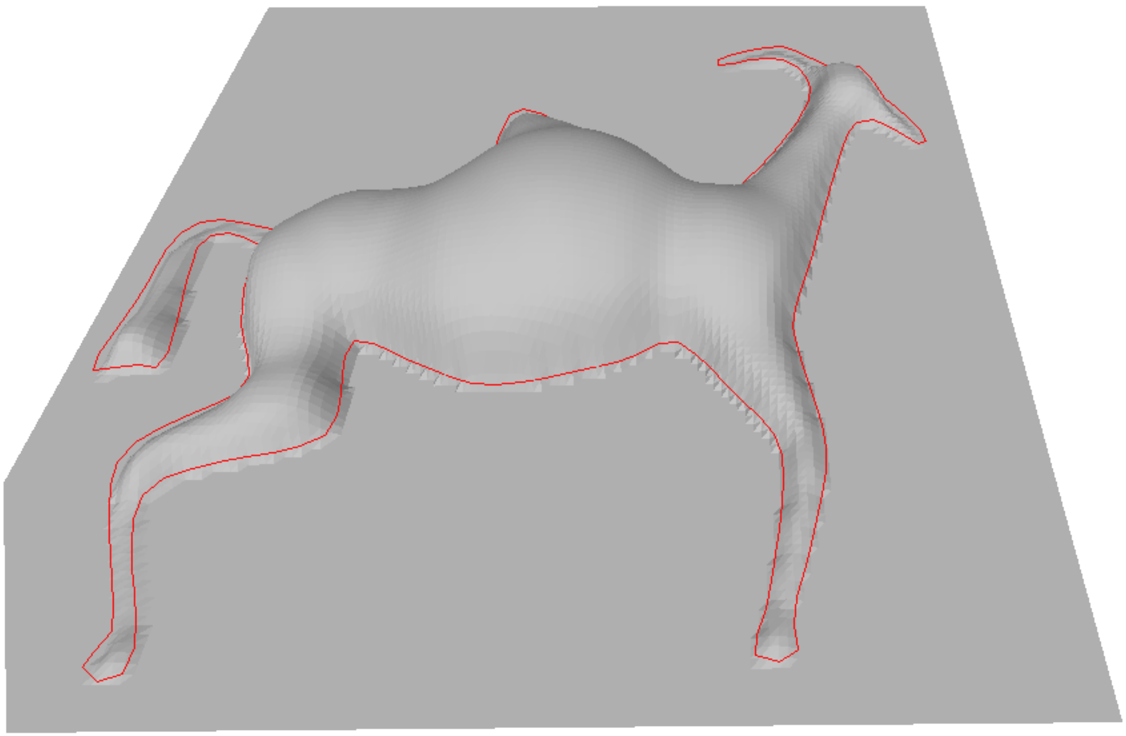
Stability Transform



Ball Transform

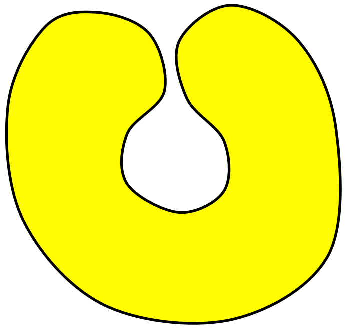


Ball Transform



Increasing the regularity of S

- Can we make S r-regular by removing from Ci(S) all points p for which r(p)?
  - This would shrink S, but not necessarily make it r-regular
  - It may not fill cracks and thin passages

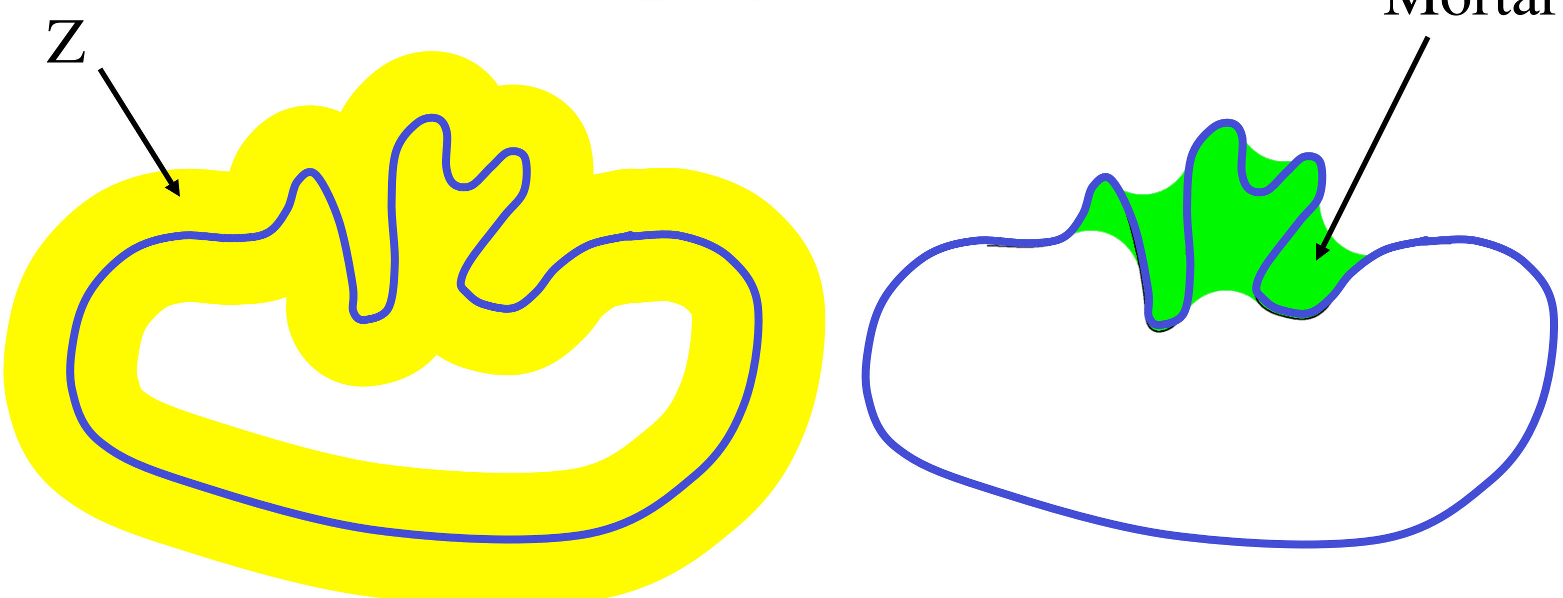


- Can we make S r-regular by removing from Ci(S) and from Ce(S) all points p for which r(p)?
  - The union of all $B_{r(p)}(p)$ for $p \in C(S)$ **will not cover all space**. Its complement is the **mortar** (if we assume that $B_{r(p)}(p)$ is open)
  - What is the new boundary of S? We want to change it in the mortar, so as to preserve regularity and minimize other measures (area change, perimeter)

Restricting simplification to the mortar

- We want to restrict all changes to a tight zone around the boundary bS of the (solid) shape S
  - Tolerence zone $Z=(bS)^{r}$
  - Mortar $M_{r}(S)=((bS)^{r})^{-r}=Z^{-r}$

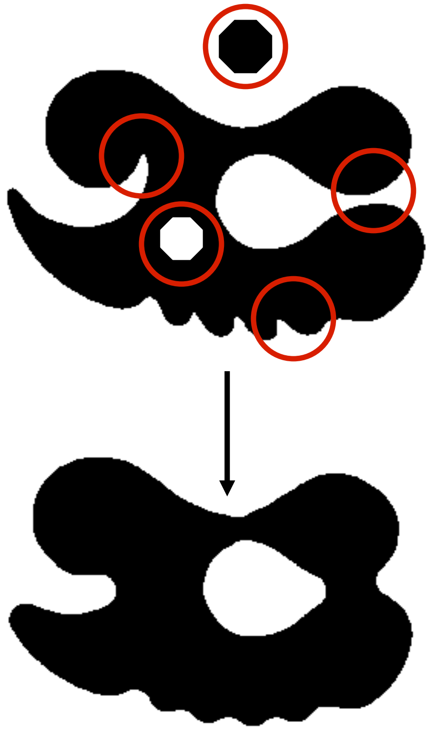
Simplify in the mortar.



What do we **simplify** and how much?

Simplification replaces a shape by a <u>simpler</u> one

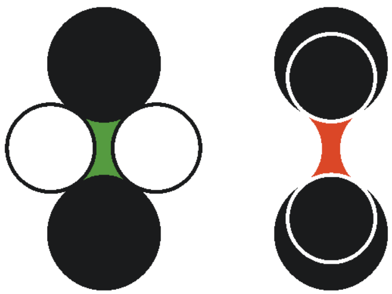
- What do we mean by "simpler"?
  - Informally, we want to
  - – Remove details
    - Reduce sharpness and wiggles
    - Eliminate small components and holes
    - Hence increase the smallest feature size
  - – Shorten (tighten) the perimeter
    - While minimizing local **changes in density**
      - – Ratio of interior points per square unit



- How much do we want to simplify
  - – We want to be able to use a geometric measure, r, to specify which of the details should be simplified and how
  - – What does the measure mean? What is the size of a detail?

Morphological Simplifications

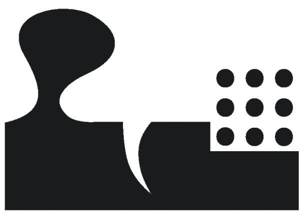
- **Fill** (closing) fills in creases and concave corners
- **Round** (opening) removes convex corners and branches
- Fill and round operations may be combined to produce more symmetric filters that tend to remove both concave and convex features ([Rossignac85])
- $F_r(R_r(S))$ and $R_r(F_r(S))$ combinations tend to:
  - **Simplify topology**: Eliminate small holes and components
  - **Smooth** the shape almost everywhere
  - **Regularize** almost everywhere
  - Increase roundness (by **reducing perimeter**)
- However they
  - Do not guarantee r-regularity or r-smoothness
  - Tend to increase or to decrease the density



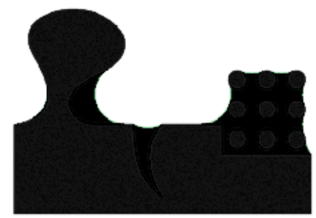
Neither $F_r(R_r(S))$ nor $R_r(F_r(S))$ will make this set r-regular

Fill F_r and Round R_r

- The r-filleting (or simply **fill**) $F_r(\mathbf{S}) = (\mathbf{S}^r)^{-r}$
  - Morphological closing with a ball of radius r [Serra 82]
  - $(S\uparrow r)\downarrow r$ [Rossignac 85]

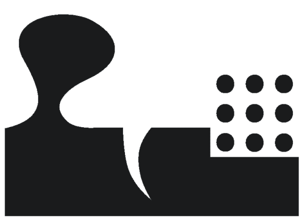


F_r adds green



add area not reachable by r-disks out of S

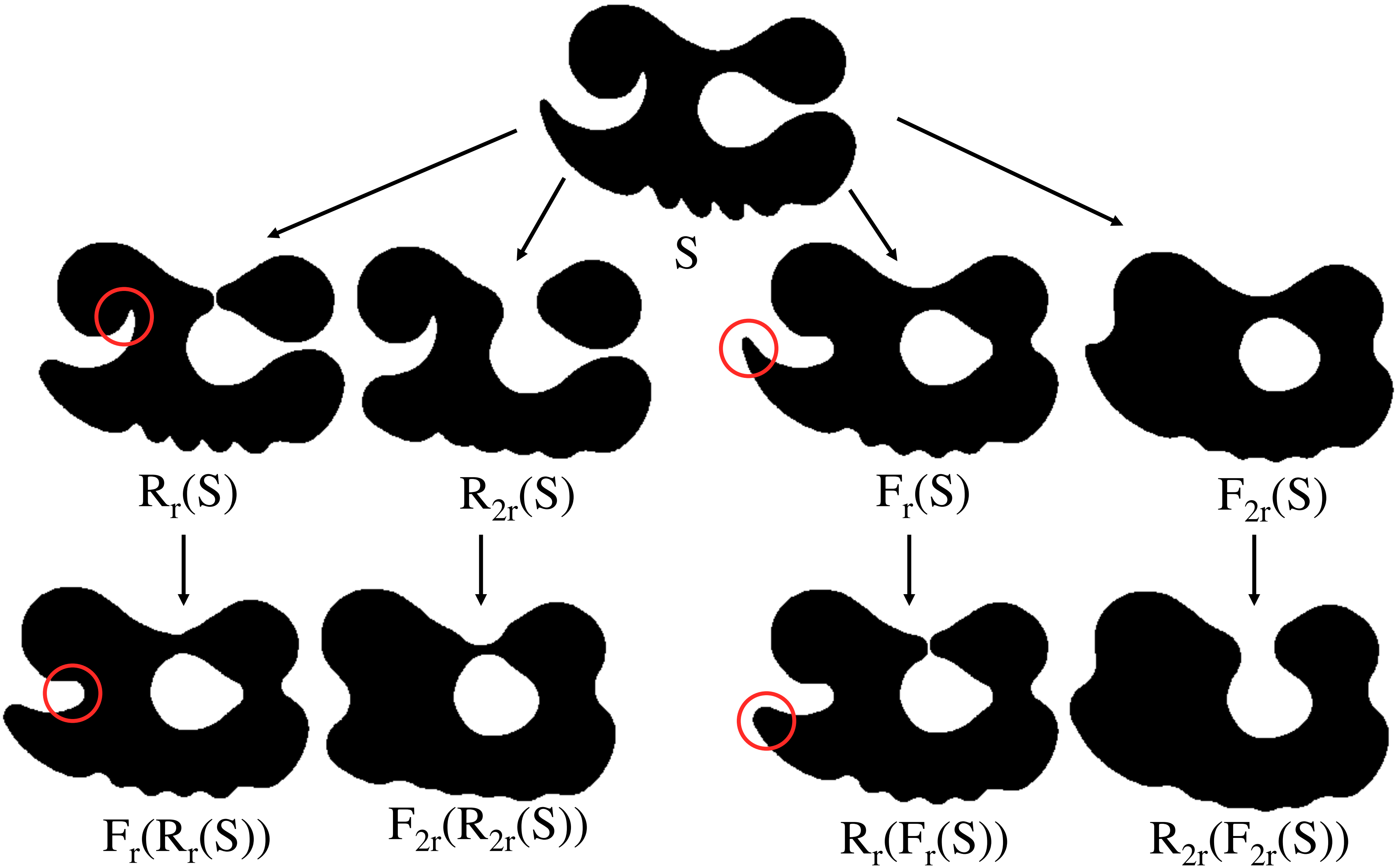
- The r-rounding (or simply **Round**) $R_r(\mathbf{S}) = (\mathbf{S}^r)^{-r}$
  - Morphological opening with a ball of radius r [Serra 82]
  - $(S\downarrow r)\uparrow r$ [Rossignac 85]



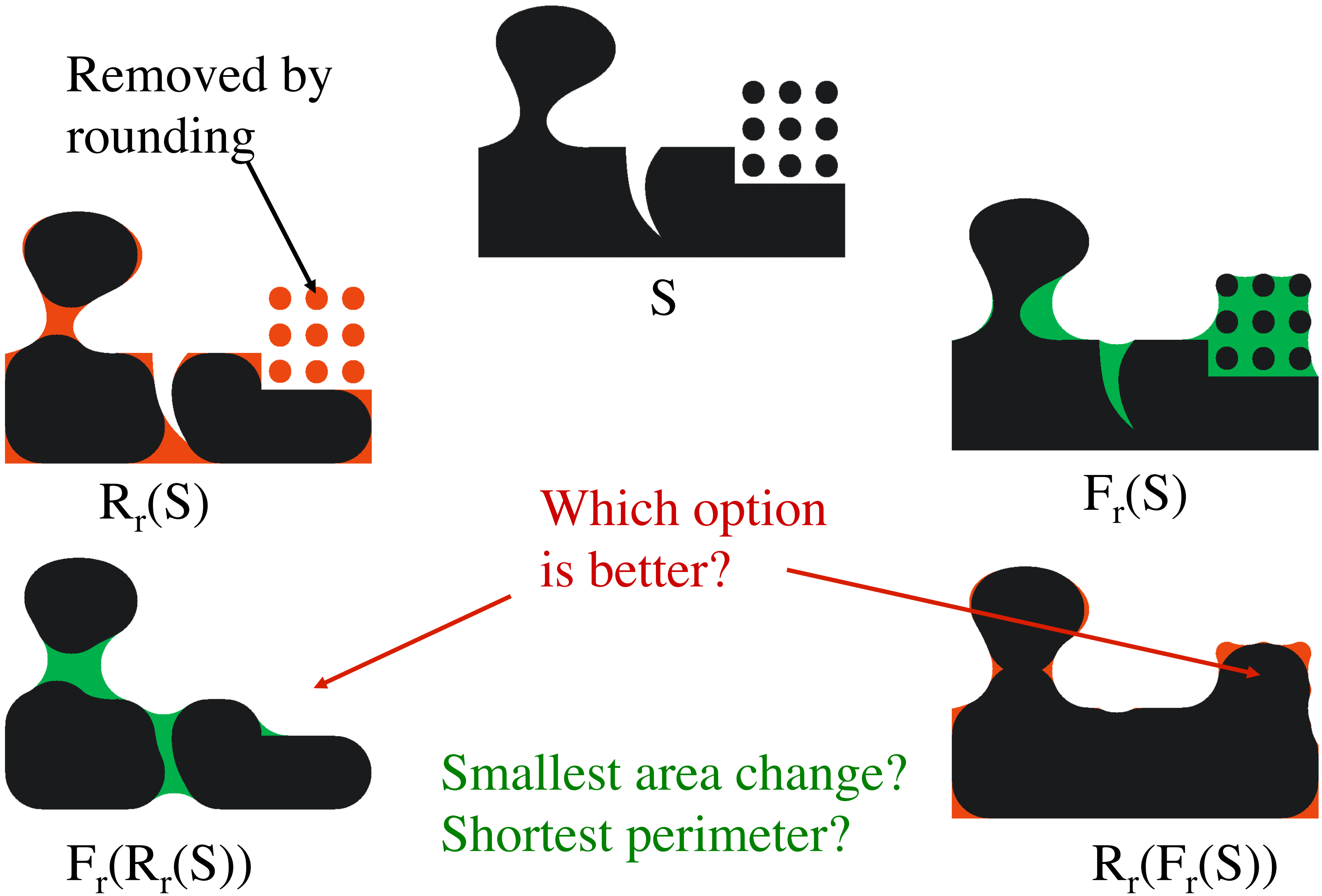
R_r removes orange

remove area not reachable by r-disks in S

Fill & Round combos



Which is better: FR or RF?

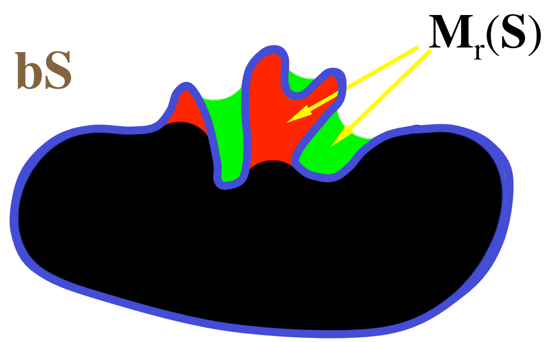


Mortar M_r

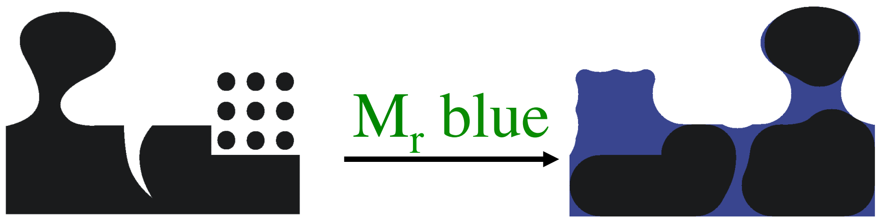
$M_r(S)$ = not reachable by open r-balls disjoint from bS

Molecular surface [Connolly 83]

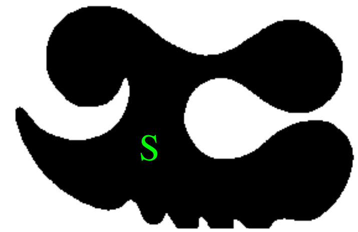
α-hull(A) for α=1/r [Edelsbrunner 93]

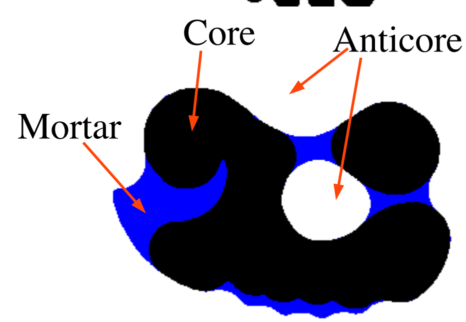


$M_r(S) = F_r(bS)$

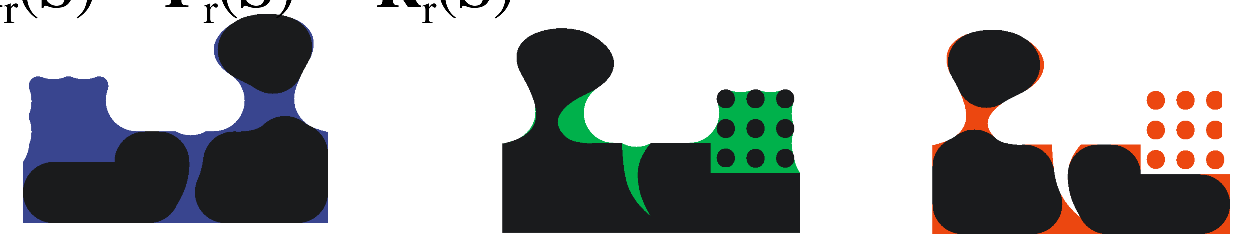


mortar





$M_r(S) = F_r(S) - R_r(S)$



blue = green + orange

core, $R_r(S)$

mortar, $M_r(S)$

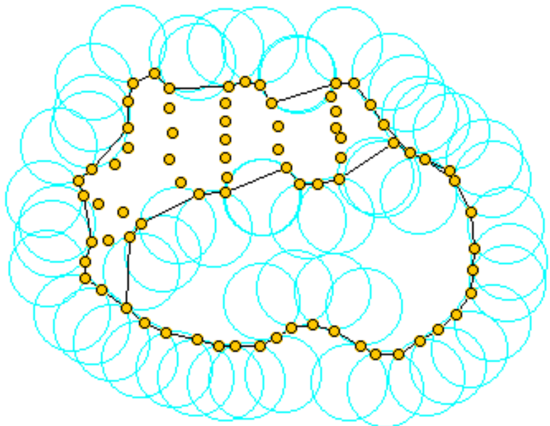
aniticore, $!F_r(S)$

Relation to α-shapes

- $F_\alpha(S)$ generalizes the notion of α-hull to continuous sets
  - Not just isolated points

- $M_\alpha(S)$ generalizes the notion of α-shape to continuous sets

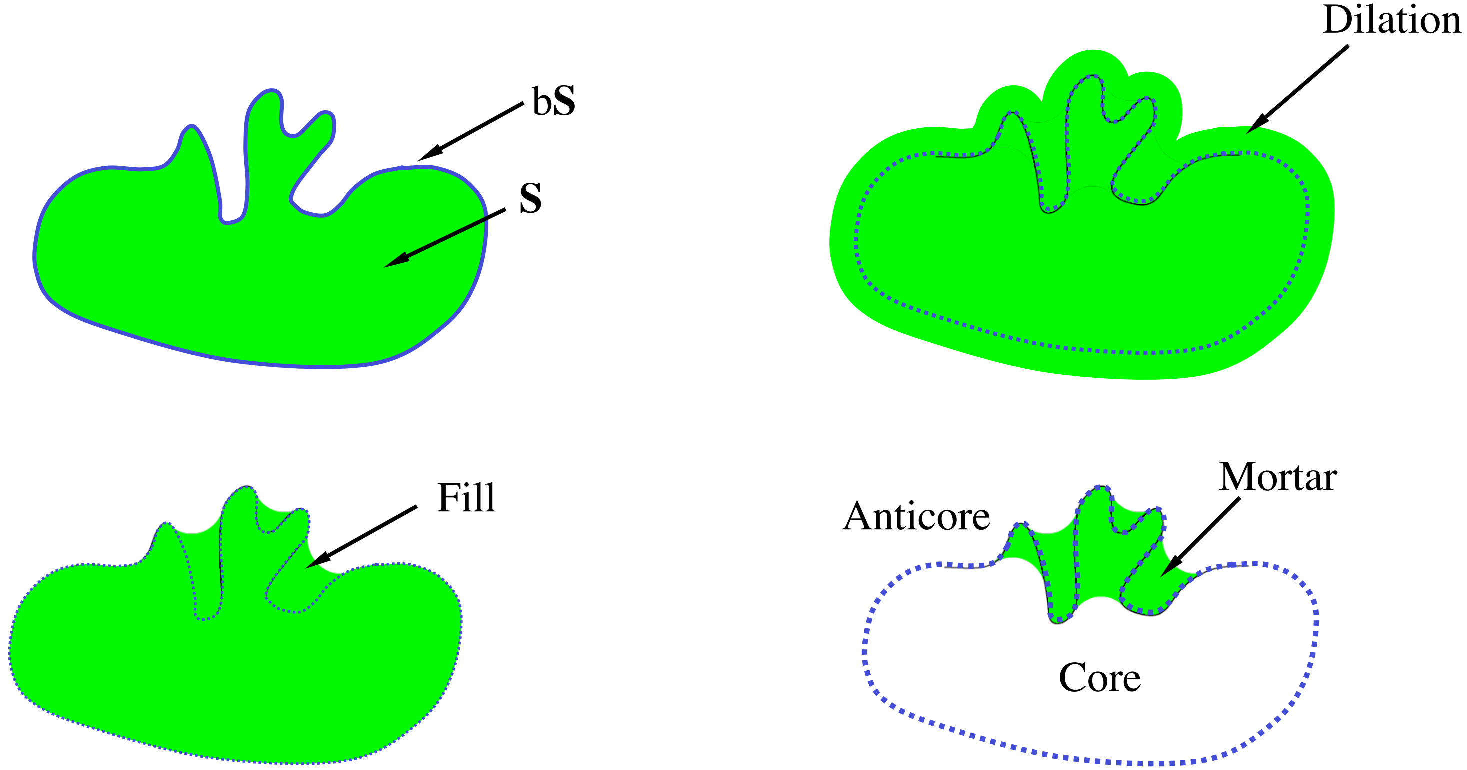
- An α-shape is a polygonal approximation of $M_\alpha(S)$
  - The result is a (possibly smooth) curved shape and not a polygon



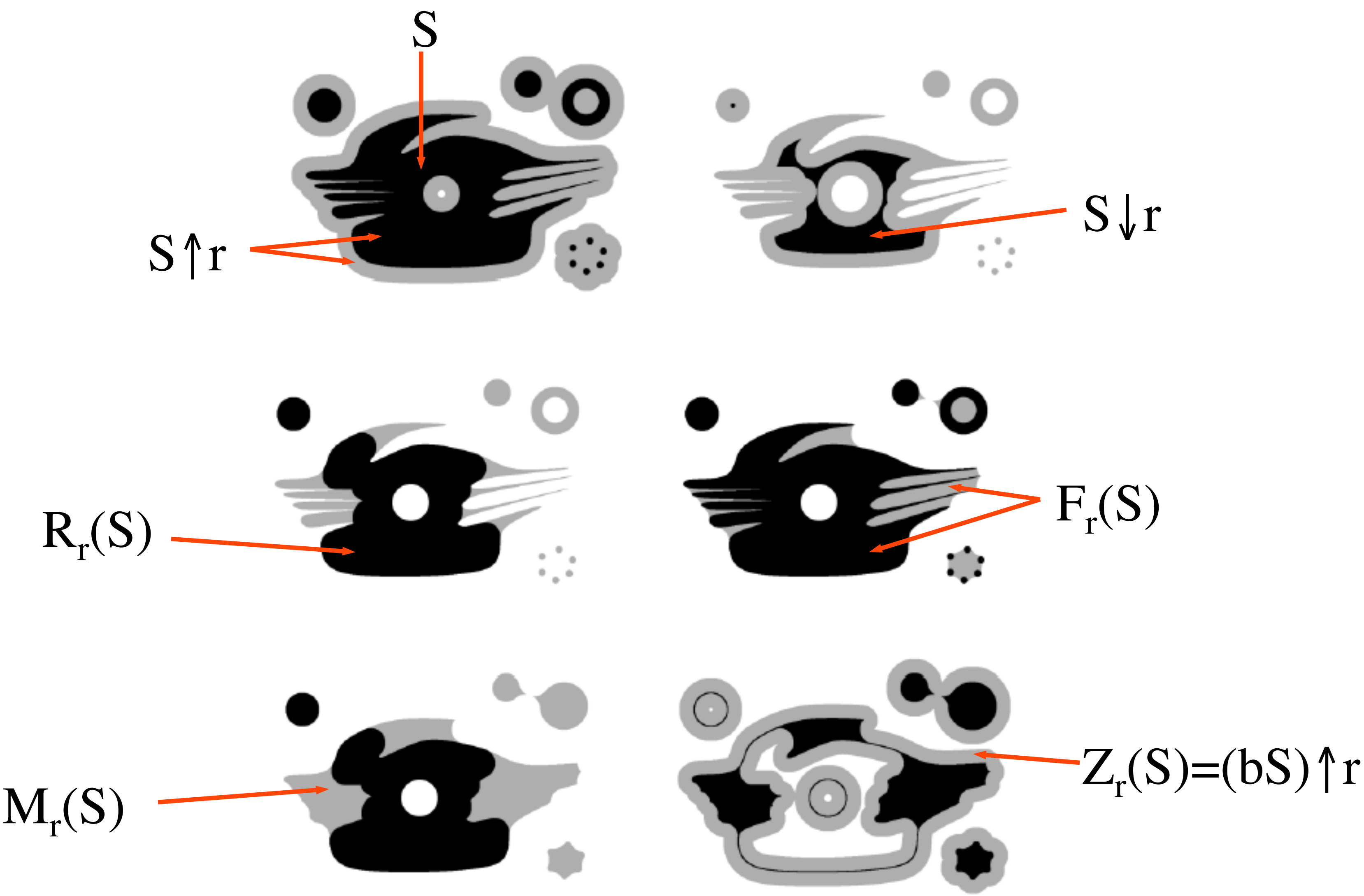
Morphological operators (summary)

r-ball : open ball of radius r

- **Dilation**, $\mathbf{S}^r$: union of r-balls with centers in **S**
- **Fill**, $F_r(\mathbf{S})$: not reachable by r-balls disjoint from **S**
- **Mortar**, $M_r(\mathbf{S})$: not reachable by r-balls disjoint from b**S**



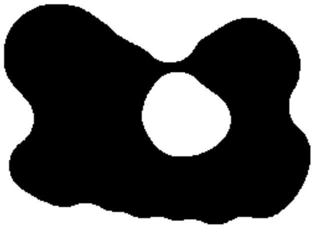
Review



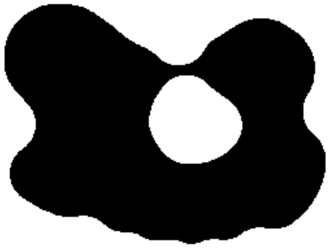
The Mason filter

For each connected component M of $M_r(S)$ replace $M \cap S$ by $M \cap F_r(R_r(S))$ or by $M \cap R_r(F_r(S))$, whichever best **preserves the shape** (minimize area change in M)

S



$F_r(R_r(S))$



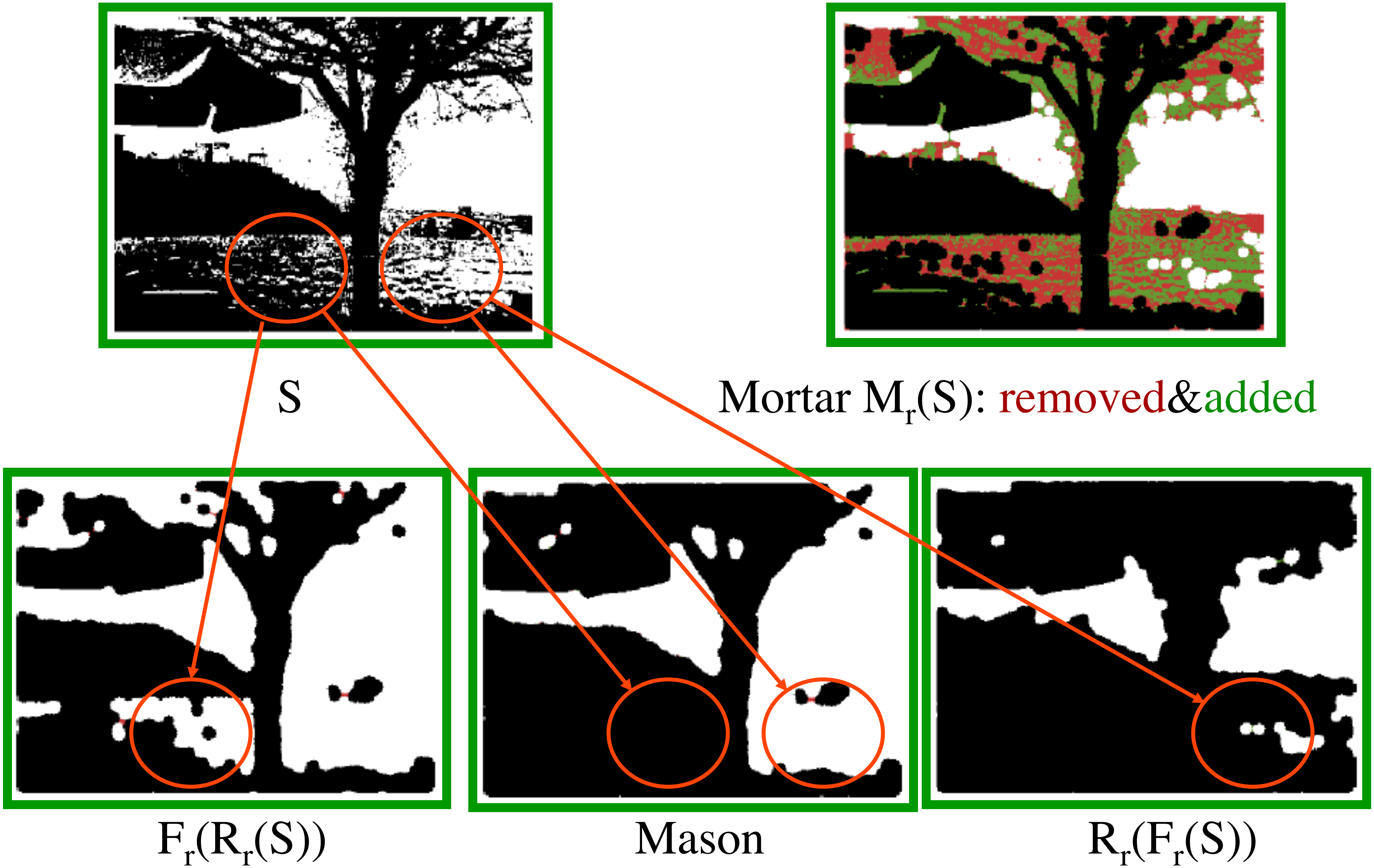
Mason

$R_r(F_r(S))$

Preserves density (average area) better than a global $F_r(R_r(S))$ or $R_r(F_r(S))$, but does not guarantee smoothness nor minimality of perimeter

"Mason: Morphological Simplification", J. Williams, J. Rossignac. Graphical Models, 67(4)285:303, 2005.

Mason in Granada



3D Mason

Can we improve on Mason?

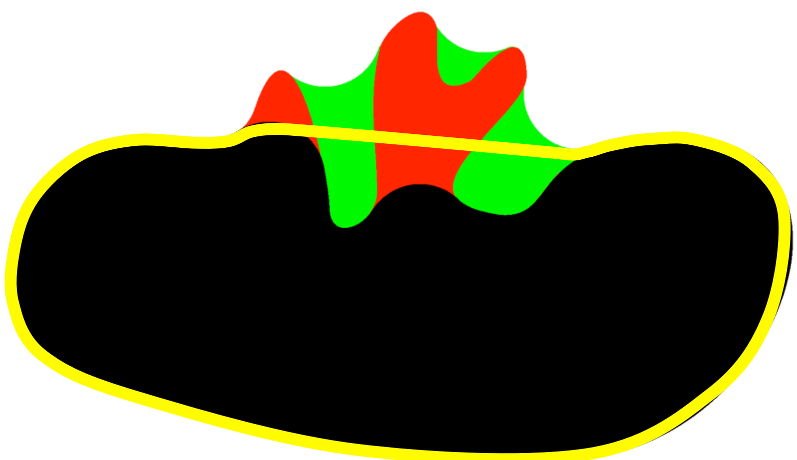
- Want to ensure r-smoothness
- Want to minimize perimeter

- Willing to give up some r-regularity
- Willing to give up some density preservation

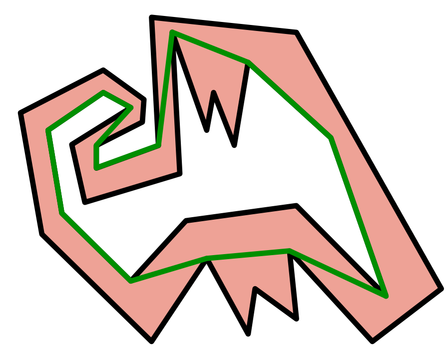
Tightening is better than Mason

A tightening, $T_r(S)$, may be obtained by tightening b**S** in $\mathbf{M}_r(\mathbf{S})$

- 2D: The shortest of all *homotopic* curves in $M_r(A)$
- 3D: Minimal area surface in Mortar

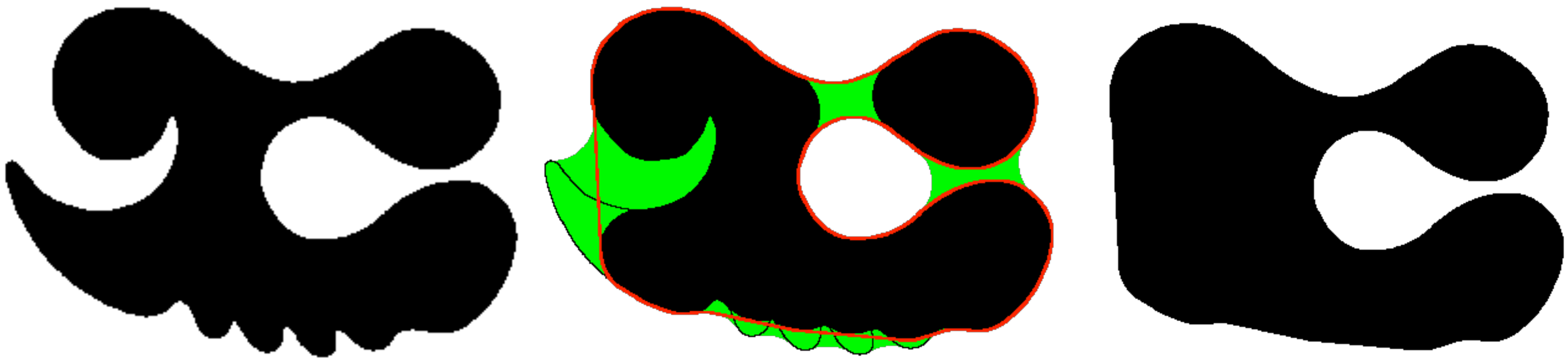


Related to the Tight Hull [Sklansk&Kibler76 , Robinson97, Mitchell04], but the hull is the mortar!

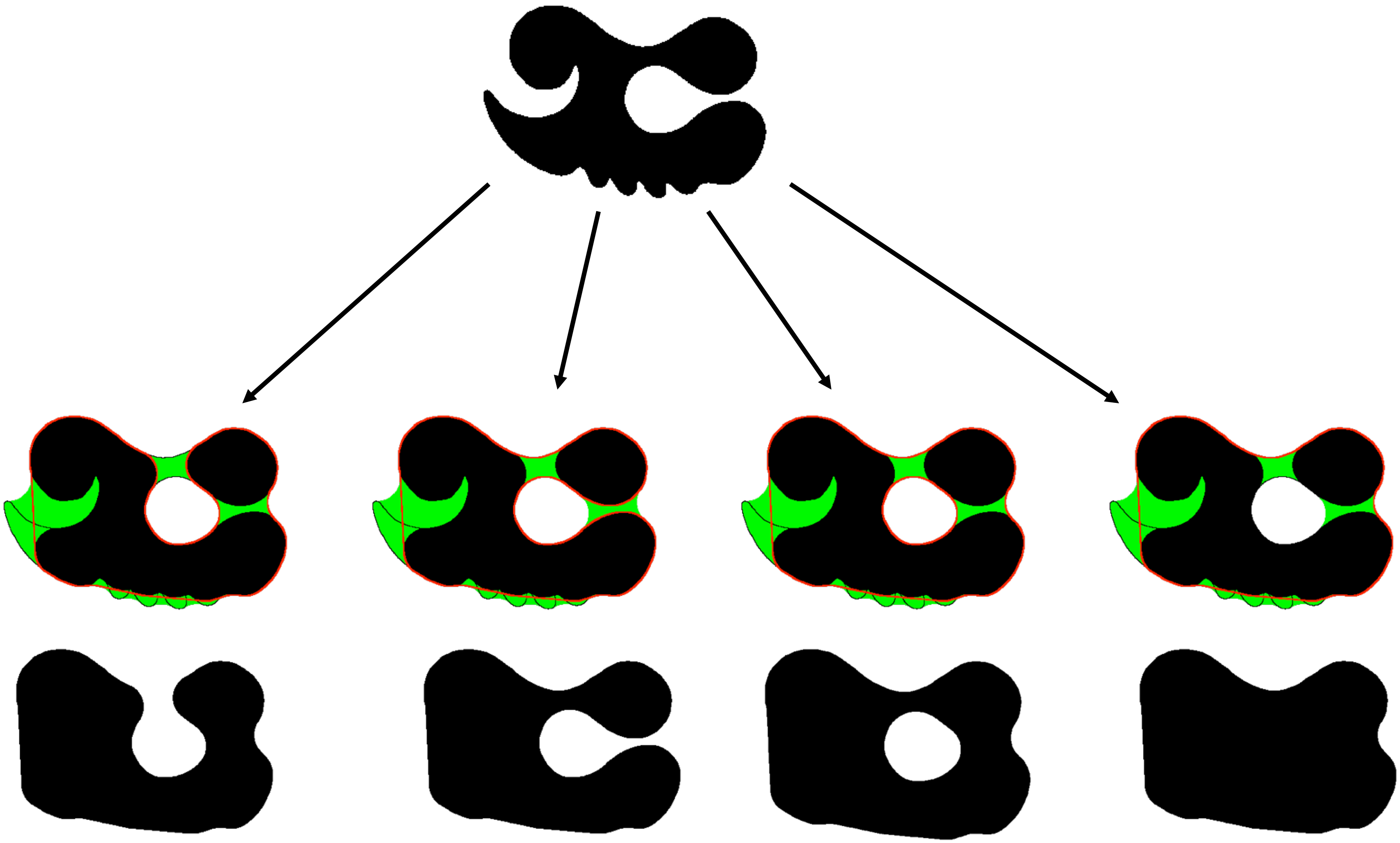


Tightening: Curvature-Limiting Morphological Simplification. J. Williams & J. Rossignac. Sketch in the *ACM Symposium on Solid and Physical Modeling* (SPM). pp. 107-112, June 2005.

Example of tightening



Topological choices of tightening



Different topology

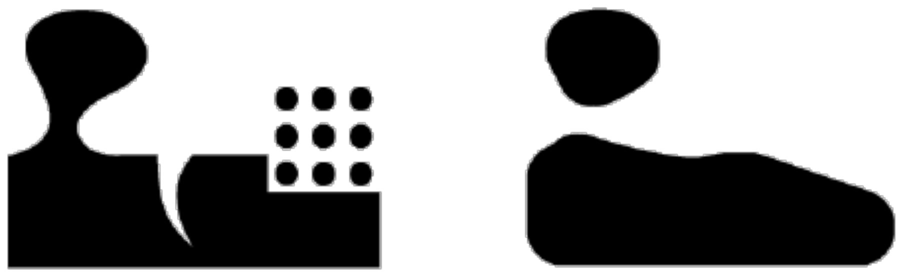
Invalid

Tightening with curvature flow

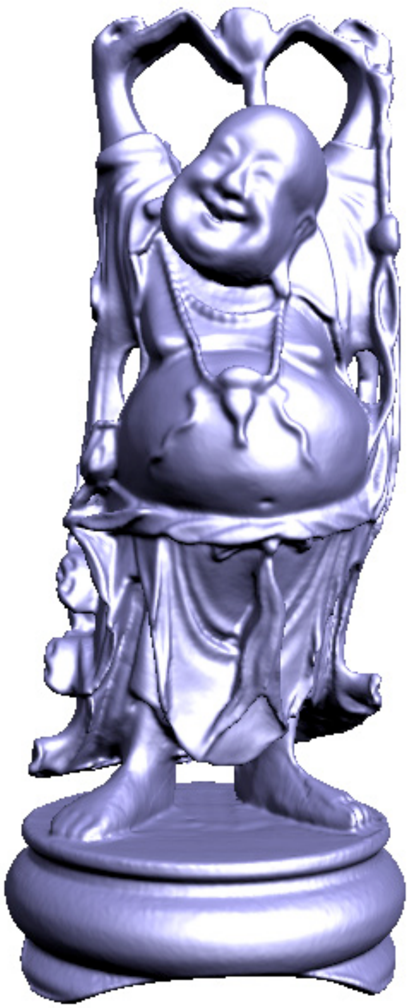
We implemented the computation of the **Mortar** on discrete grids using fast **distance field** computation. (60 Sec)

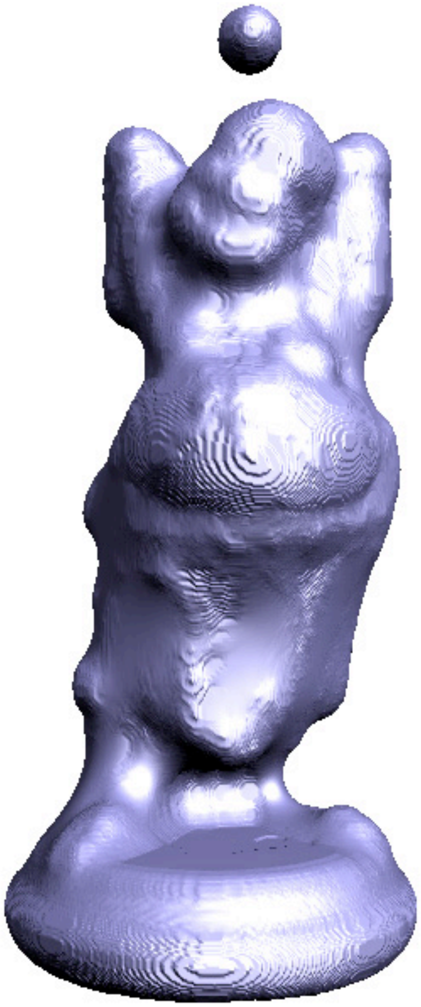
1102x832 pixels

We implemented **Tightening** as a **level-set curvature flow** constrained to the Mortar. (90 sec)



3D tightening



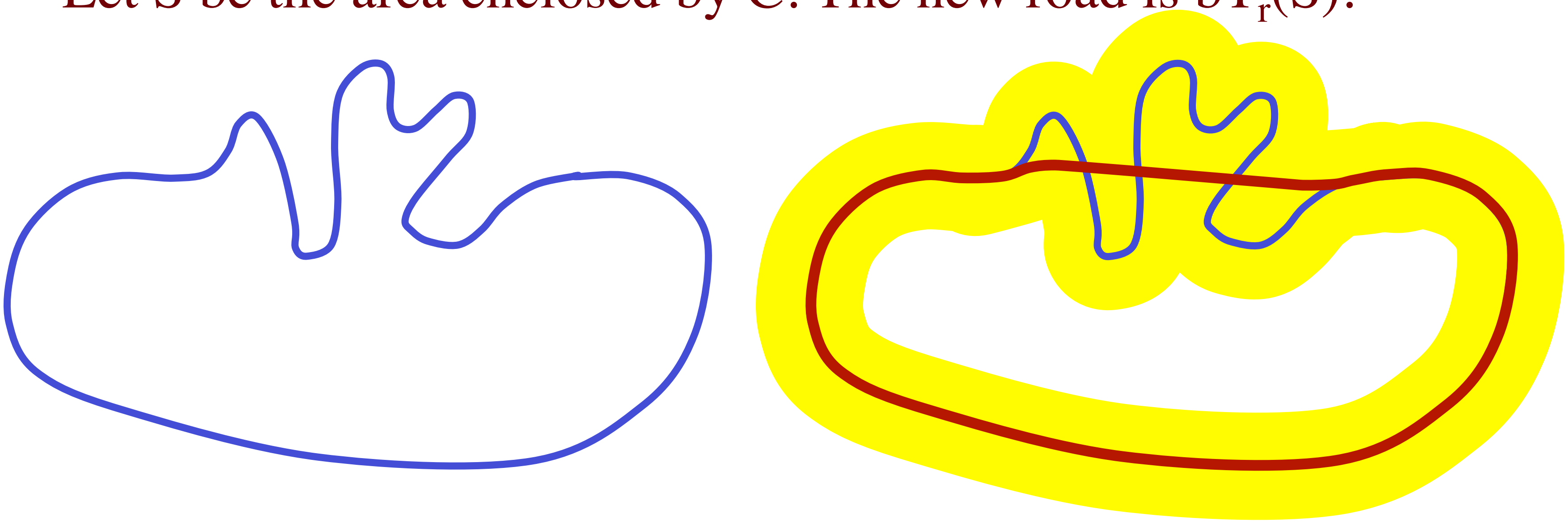


The road-tightening problem

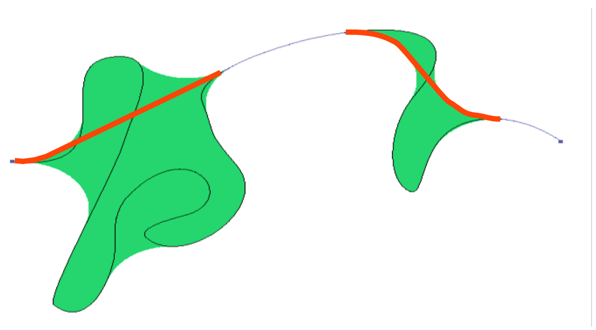
- By law, the state must own all land located at a distance less then or equal to r from a state road.
- The states owns an old road C and wants to make it r-smoother.
  - Wand the radius of curvature to be > r everywhere
- Can it do so without purchasing any new land?
  - For simplicity, assume first that C is a manifold closed loop.

The road-tightening solution

- Let S be the area enclosed by C. The new road is $bT_r(S)$.



We can extend this result to open road segments.



Comparing morphological simplifications