Implementation Details

Though they present a neat algorithm to compute the medial axis transform of arbitrary 2d shapes, [Joan-Arinyo et al.] do not adequately fill the reader in on implementation details of their Tracing Paths paper. Here, I'll elaborate on my implementation of the paper.

Terminology

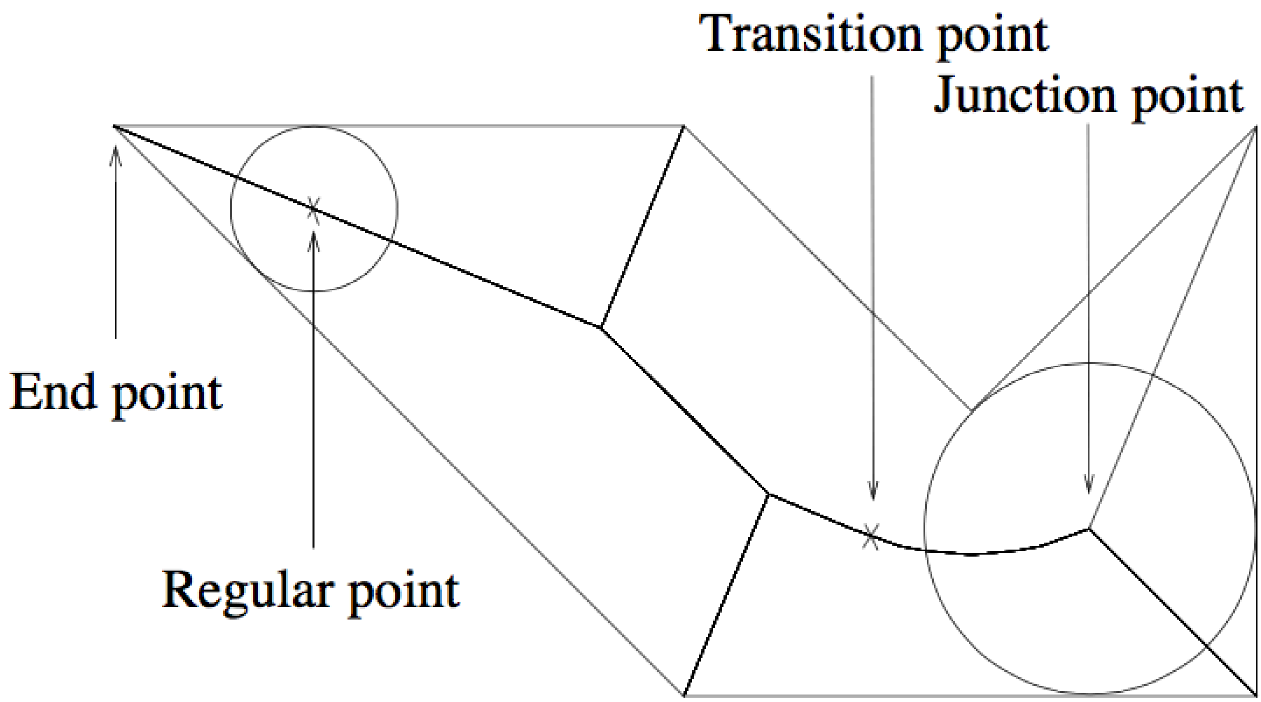


Figure 1: Taxonomy of medial axis points.

Boundary Element - An edge or a concave vertex in the boundary.
Governor - Boundary element to which a medial ball is tangent.

Junction Point - Center of the medial ball tangent to three or more governors.
End Point - Points on the medial axis with zero radius or a convex vertex of the polygon.
Regular Point - Center of the medial ball tangent to two governors.
Transition Point - Regular point where one or both of the governors change.
Key Point - A junction, end or transition point.

Path - Subset of the medial axis whose points are generated by the same set of governors. Bounded by two key points, one at each end.

The **region of influence of a boundary element** is defined by its **half-lines.** For two consecutive edges on the boundary, the half-line is the bisector of the edges. For an edge - concave vertex pair, a half-line is the perpendicular to the edge. From an implementation perspective, each boundary element has two half-lines, a back half-line and a front half-line.

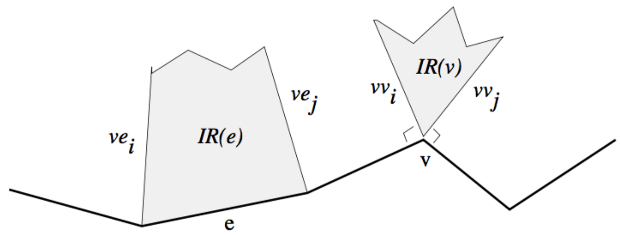


Figure 3: Influence regions of straight segments and concave vertices in a 2D polygonal domain boundary.

Orientation
The boundary elements must be stored in either a clockwise or anti clockwise order. The algorithm works by tracing the medial axis by a medial ball that moves from the initial key point (an end point) to the final key point, one key point at a time, in an X first search order such that the ball is always tangent to the governors.

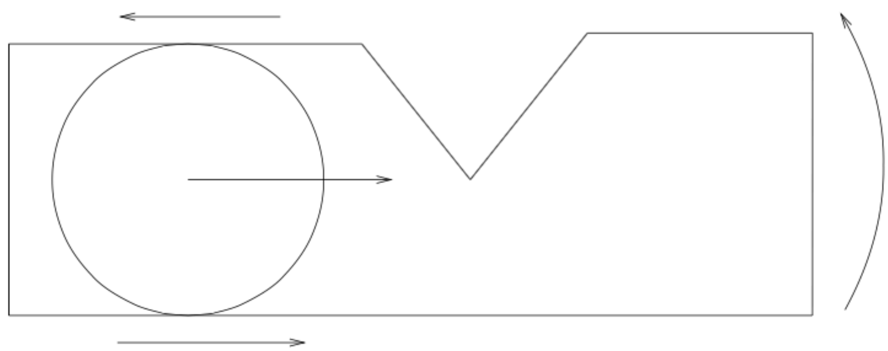


Figure 6: Governors orientation with respect to the disc motion.

The pseudo code from the paper is pasted below.

```
algorithm MAbyTracingPaths
  InitializeFirstPath (d, g1, g2)
  PushPath (d, g1, g2)
  while not EmptyPath() do
    PopPath (d, g1, g2)
    TracePath (d, g1, g2, InterferenceList)
    if not EndDisc(d) then
        InitializeNewPaths (d, g1, g2, InterferenceList, NewPathList)
        for path in NewPathList do
          PushPath (d, g1, g2)
        endfor
    endif
  endwhile
endalgorithm
```

Here, d is the disc centered on the medial axis and tangent to the properly oriented governors g1 and g2.

InitializeFirstPath
Any convex vertex serves as the starting key point of the medial axis while the edges its incident on serve as the properly oriented governors. That is, assuming a anti clockwise orientation, governor 1 lies before the key point and governor 2 after.

TracePath
Given an initial key point and a pair of corresponding governors, the next key point must be determined. To do so, first a key point candidate is computed and then it is checked whether its valid. If not, a new valid key point is computed. Computations are carried out depending on the type of governors of the path to be traced.

Edge-Edge Case
When the two governors are edges, the medial axis is the bisector l_m to the edges. If e_i is the first governor with h_{i1} as its first half-line and e_j the second governor with h_{j2} as its second half-line, the next key point cannot extend beyond point mi or mj where h_{i1} and h_{j2} respectively intersect l_m. The candidate key point is the point closest to the initial key point in the path. The medial ball radius is the perpendicular distance of the candidate key point to any of the governors.

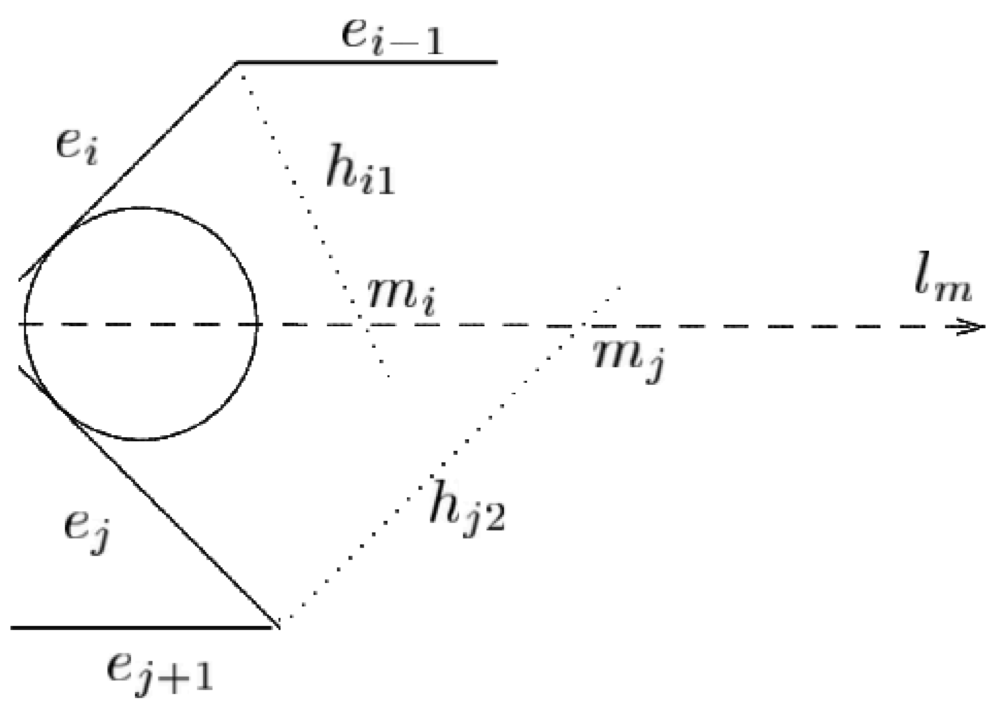


Figure 7: Edge-Edge case.

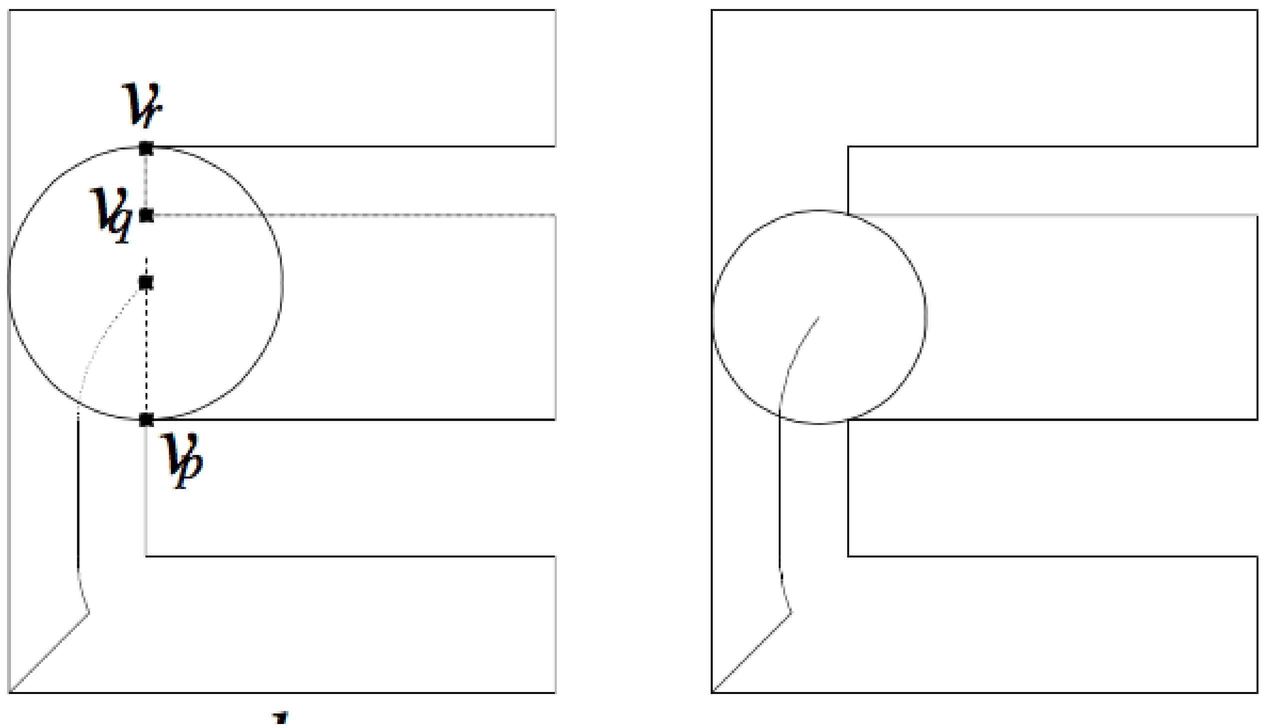
If there is any boundary element that lies within the medial ball centered on the candidate key point, then the candidate key point is not valid. In that case, the concave vertex closest to the initial key point that is not a governor and lies within the medial ball is selected to determine the new key point. Given two edges tangent to a circle and a concave vertex that lies on the circle, an easy way to determine the actual key point/center of the medial ball is to reduce this case to that of three tangential edges to a circle, from which it is easy to determine the center.

To determine the tangent edge that lies on the concave vertex, the following routine can be used.

```
Vector2d h = (h1 + h2)/2;
h.normalize();
h = Vector2d(-h.y(),h.x());

Edge e(p, p + l);
```

Here, h1 is the back half-line of the concave vertex and h2 the front. After averaging them, we get a vector whose perpendicular lies along the tangent of the tangential edge. This perpendicular is used to determine the second vertex of the edge.



Concave Vertex-Edge Case
As illustrated in the diagram below, the medial axis path is an arc of a parabola for a concave vertex - edge governor pair. This arc can be computed by equating the distance of concave vertex (h, k) to a point (x, y) on the parabola to the distance of a point the edge to the same point on the parabola.

$(ax + by + c)^2 / (a^2 + b^2) = (x-h)^2 + (y-k)^2$

This equation can be solved for y in terms of x on wolfram. The expression is complicated, and I'm still working out how to simplify it to the standard equation of a quadratic $y = ax^2 + bx + c$ for reasons I'll explain soon.

As in the Edge-Edge case, the back half-line of the first governor and front half-line of the second governor intersect the parabola. While both of them do not necessarily intersect the parabola, I believe one of them always does. The intersection closer to the initial key point again serves as the candidate for the next key point. If the equation of the parabola were available in the form $y = ax^2 + bx + c$, then determining the line parabola intersection is easy, just equate

y_parabola = y_line
=> $ax^2 + bx + c = mx + x$

and solve for x. However, without this form, the only way I could think of find the intersection point was through the brute force approach - incrementing through values of x till y_parabola = y_line. This is both inefficient and inaccurate.

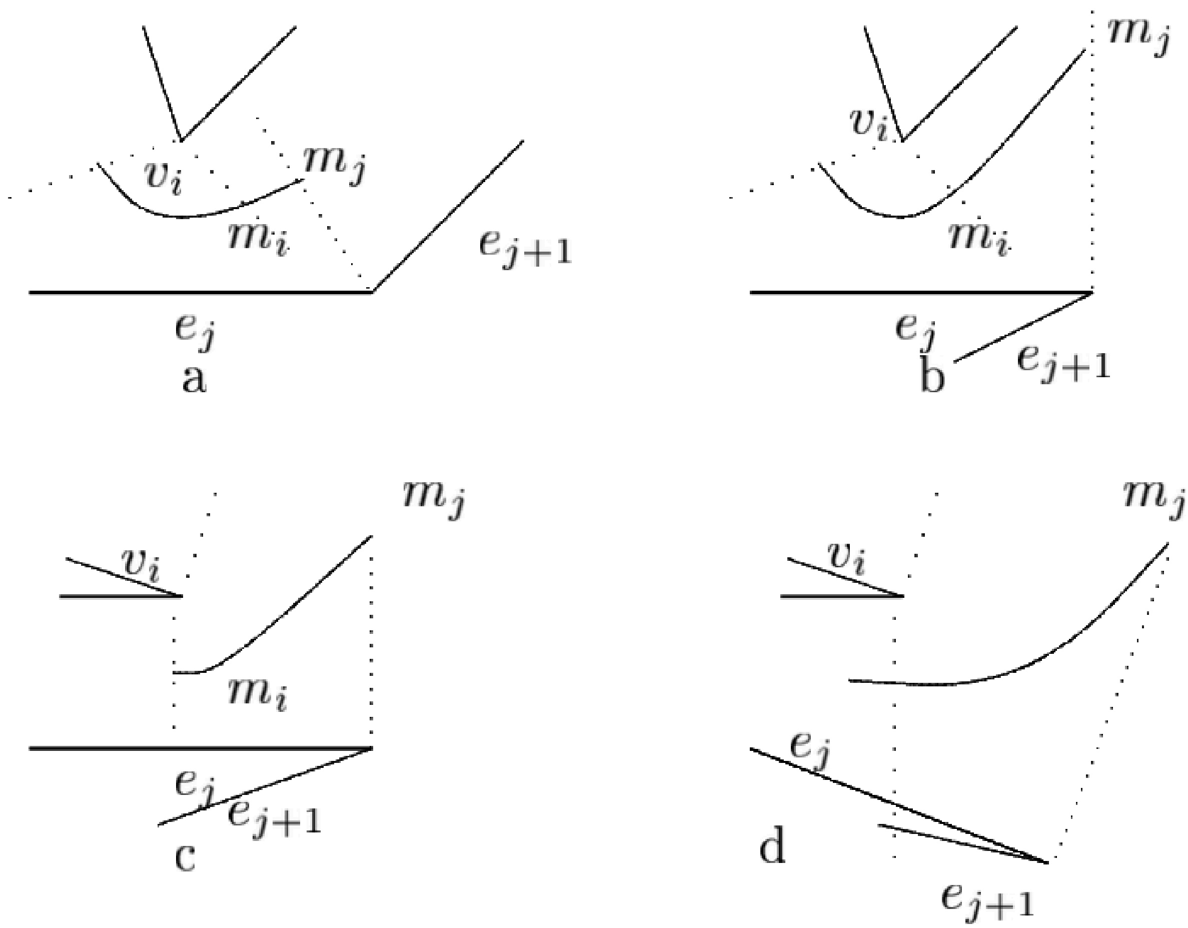


Figure 8: Concave Vertex-Edge case.

After determining the candidate key point, it needs to checked whether it is valid. Again, if there is any boundary element that lies within the medial ball centered on the candidate key point, then the candidate key point is not valid. In that case, the concave vertex closest to the initial key point that is not a governor and lies within the medial ball is selected to determine the new key point. Given one edge tangent to a circle and two concave vertices that lie on the circle, it is again possible to determine the actual key point/center of the medial ball by reducing this case to that of three tangential edges to a circle in the same manner as the one described above.

Concave Vertex-Concave Vertex Case

When the governors are two concave vertices, the medial axis is a straight segment perpendicular to the straight line joining the two concave vertices through its midpoint. If at least one of the half-lines h_{i1} or h_{j2} intersect the path, the candidate key point is determined in the same manner as in the above cases. It must also be checked whether its valid.

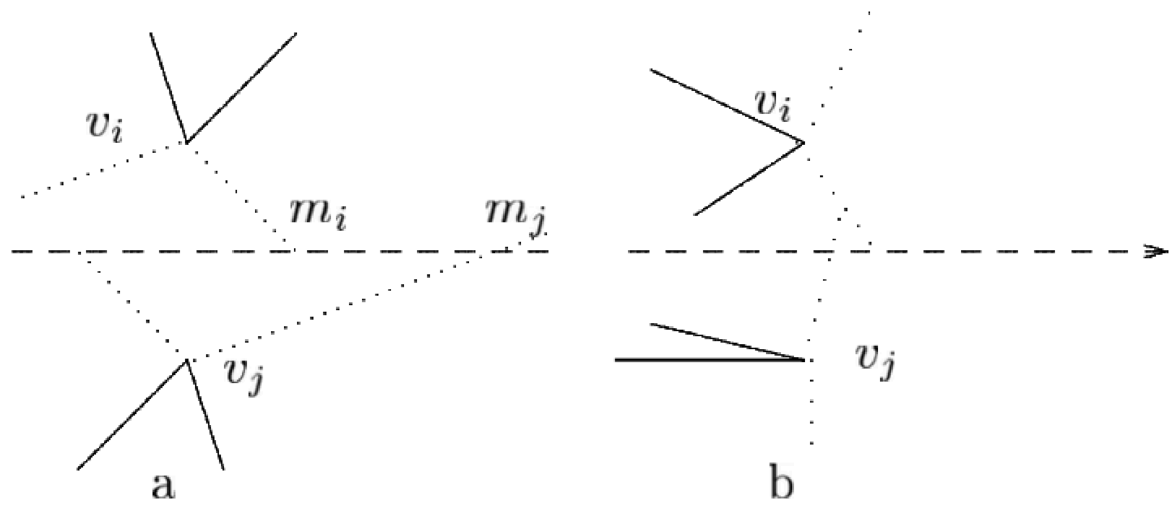


Figure 9: Concave vertex-Concave vertex case.

If neither h_{i1} or h_{j2} intersect the path, then a third boundary element closest to the previous key point candidate is required. In this case, we now have three concave vertices that lie on a circle, from which it is easy to determine the next key point/ center of the medial ball.

InitializeNewPaths
Once a path has been traced, new paths start at junction and transition points. A medial ball centered on a junction point is tangent to three or more boundary elements, and thus serves as the starting point for two or more medial paths. The boundary elements intersecting with the medial ball are sorted either in an anti clockwise or clockwise order with respect to the key point at the center of the medial path (the boundary elements don't actually need to be sorted if they are stored in an ordered array in the first place). The first and last elements in the set are the governors which defined the path leading to the junction point. This junction point then serves as the first key point of the next path of the medial axis with consecutive boundary elements serving as governors for that path.

A tricky implementation detail to remember is that when the medial ball is tangent to a concave vertex, then it also intersects the edges the concave vertex is incident on. As a first step, these edges are not included in the list of intersecting boundary elements.

Next, transition points must be accounted for. In all cases, there is at least one concave vertex involved in the list of intersecting boundary elements, which either becomes a governor in the current step or is replaced in the next by an adjacent edge. From the implementation mentioned above, the intersecting boundary elements no longer contain adjacent edges to a concave vertex. If the concave vertex is paired with the same boundary element a second time to form the new governors, then it should be replaced with the adjacent edge behind it or in front of it. To determine the direction of the transition, track the governors the concave vertex was paired with in the previous step. If the concave vertex was previously the first governor, it should be replaced with the edge in front of it and the point should transition forward. If the concave vertex was previously the second governor, it should be replaced with the edge behind it and the point should transition back.