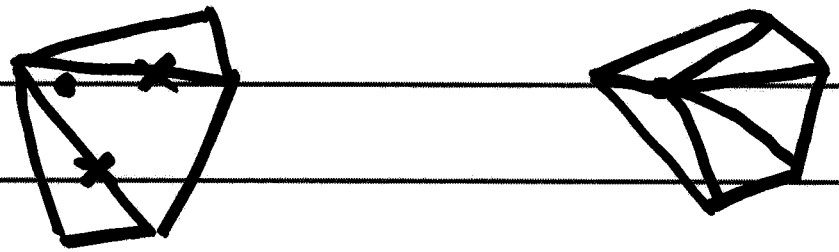
4. make union of these Δ's : form insertion polygon



5. Destroy all internal edges, and connect the point with vertices
go back to 2 (pick a point)

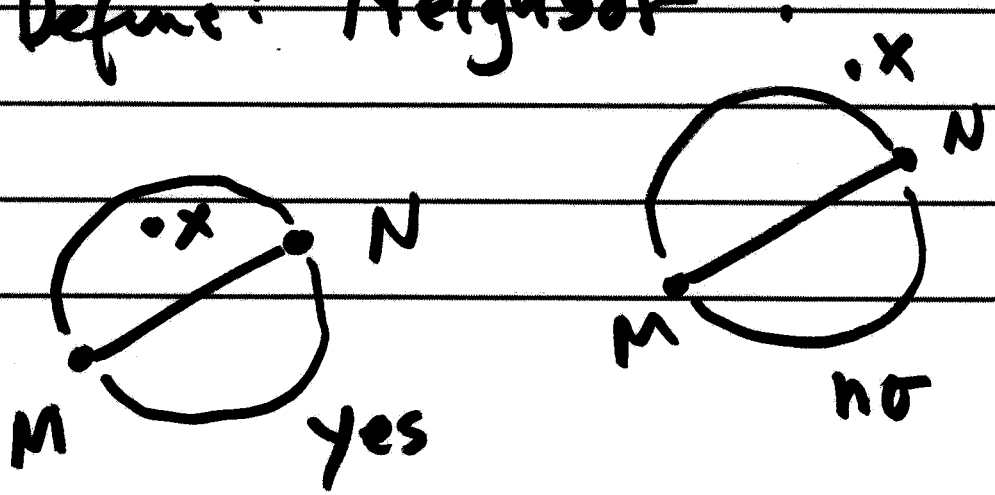
6. eliminate any Δ with vertex at any of 3 fictitious points.

Constrained Delaunay Triangulation (Breaklines)

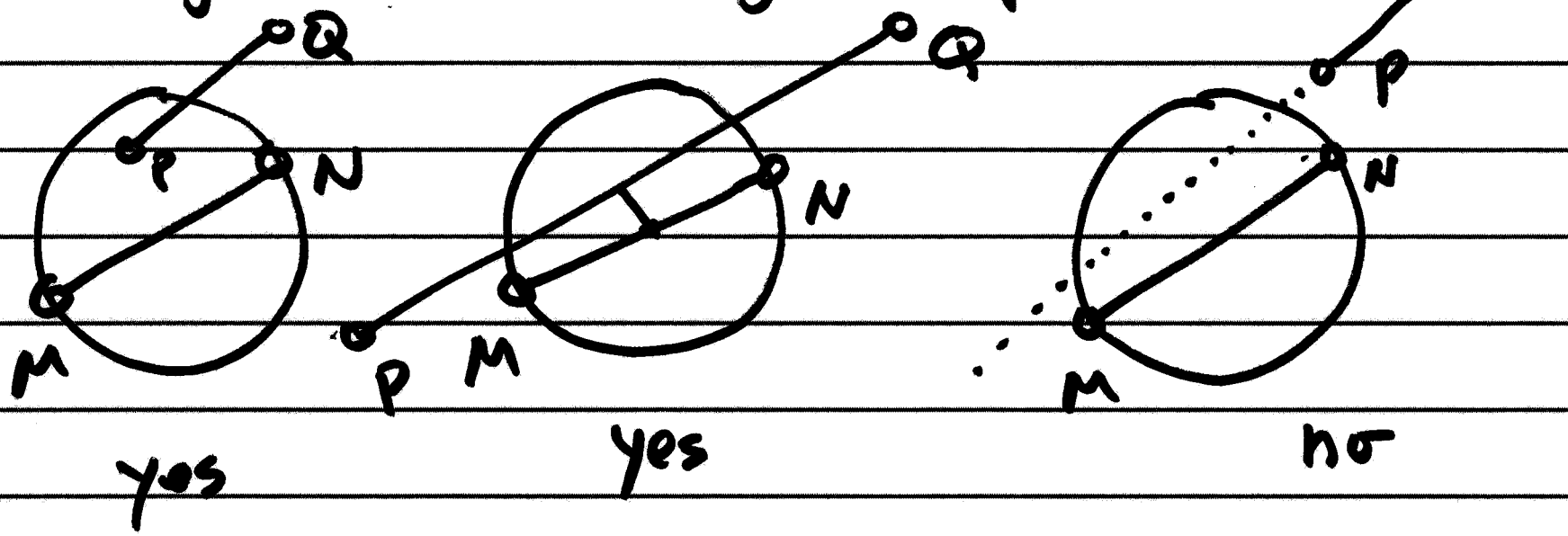
Reference: Fangeras, Three Dimensional Computer Vision, MIT Press '93 pp. 438-471

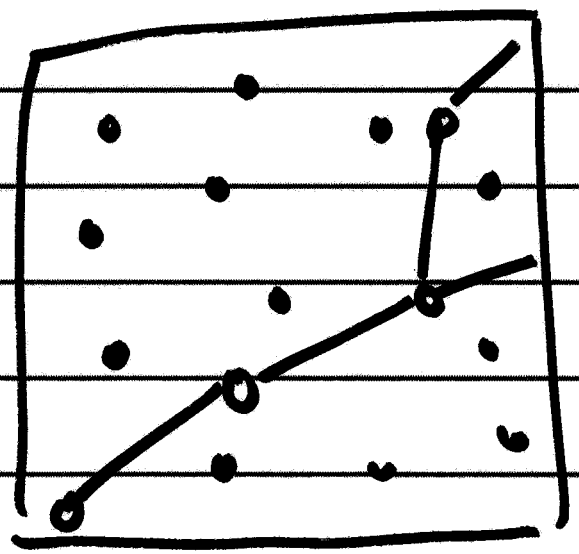
Given: 1. Mass points X: $X_1, X_2, X_3, \ldots X_n$
2. Break line segments S: $S_1, S_2, \ldots S_M$

Define: Neighbor?



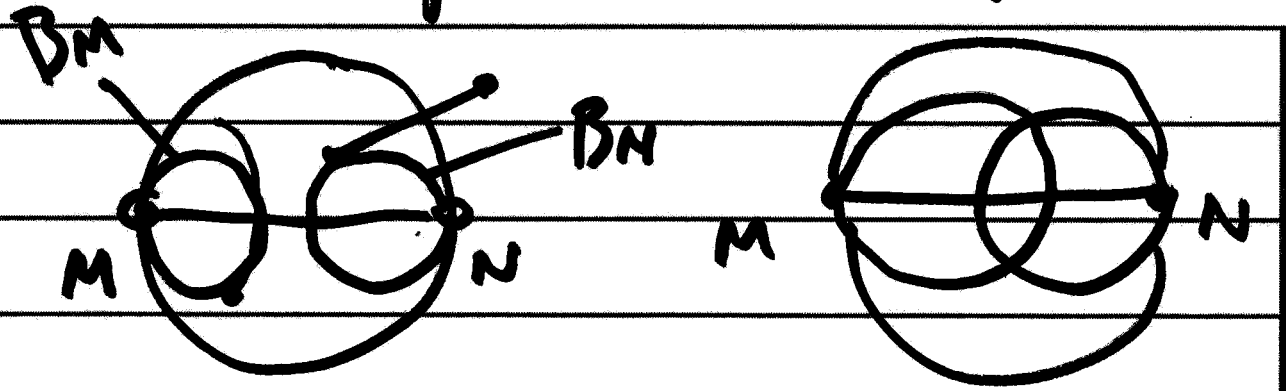
is segment PQ a neighbor of MN?





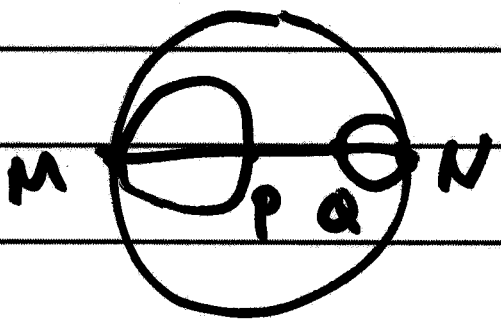
1. Initialization: compute neighbors of each segment insert all segments of S into OPEN, initialize CLOSED to NIL

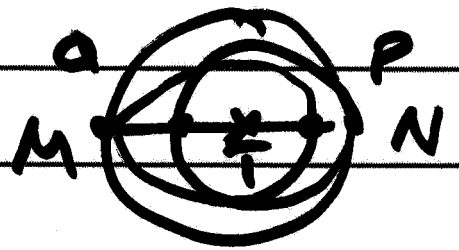
2. While OPEN not NIL, execute through Step 6. Take first segment MN of list if no neighbors, move to CLOSED, go to next segment, if neighbors then compute ball B_M, B_N



2. IF $B_M \wedge B_N = 0$, go to 3, otherwise 4

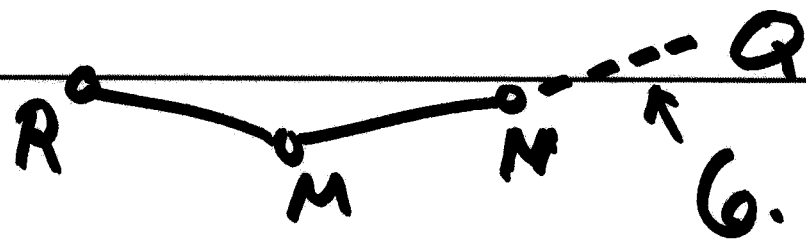
3. $B_M \wedge B_N = 0$, insert MP, NQ into CLOSED
insert PQ into OPEN, go to 5





4. $B_M \wedge B_N \neq 0$, Let T be midpoint of PQ,
add MT, NT to CLOSED

5. if MR adjacent to MN



7. Compute via Watson a triangulation of collection mass points + subdivided segment endpoints.

8. done.