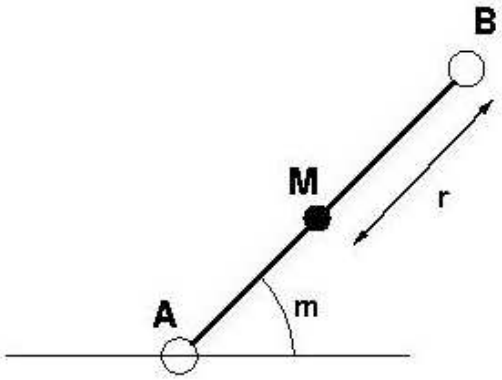
Chapter 1

Data Structures

In order to calculate the *Greedy Trinagulation* of a planar graph we start from the list of edges of the corrispondent *complete* graph, i.e. the graph with the maximum number of edegs $\big(N(N-1)/2\big)$. In addition we sort this list over the edge length in increasing order. Let's give a look at which characteristics of each link AB added or to be added to the GT link list we should look at. Each row



of the link list should look like this:

$$\{Nfrom, X_a, Y_a, X_M, Y_M, Nto, X_b, Y_b, r, m\}$$

Where:

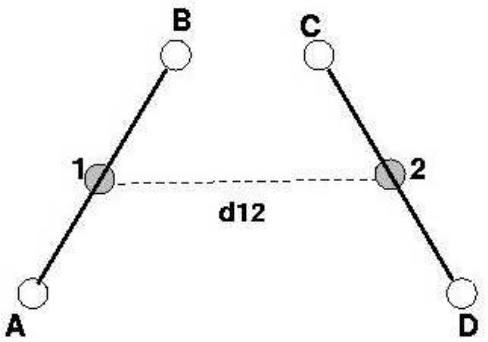
- Nfrom: ID of the starting node;
- X_a: X coordinate of the starting node;
- Y_a: Y coordinate of the starting node;
- X_M: X coordinate of the mean point of segment AB;
- Y_M: Y coordinate of the mean point of segment AB;
- Nto: ID of the arrival node;
- X_b: X coordinate of arrival node;
- Y_b: Y coordinate of arrival node;
- r: radius $\frac{d_{AB}}{2}$;
- m: angular coefficient of segment AB;

Chapter 2

Algorithm

Concerning the algorithm it is fundamentaly based on two levels of checks. Checks can be divided into two class: geometric and informatic. The geometrical check is easy to implement and allows to make a first filter over the list in order to select only the "worthful" cases avoiding an explosion of computational complexity. The second check, instead, is much complex and allows to individuate without any doubt the edeges belonging to GT.

Considered two edges, AB (belonging to GT) and CD (to check if it belongs to GT) and say m_1, m_2 the angular coefficients; r_1, r_2 the radiuses and 1, 2 the mean points of the segments, we have:



2.1 Step 1

If $d_{12} > (r_1 + r_2)$ then CD does not cross AB;
If $d_{12} \leq (r_1 + r_2)$ we must go to step 2;

2.2 Step 2

Supposing that the vertex of an edge are ordered such that:

$X_a < X_b$
$X_c < X_d$

Let's build the linear system associated with the equations of the straight lines which the segments belong to.

$$\begin{cases} y - y_1 = m_1(X - x_1) & X \in [X_a, X_b] \\ y - y_2 = m_2(X - x_2) & X \in [X_c, X_d] \end{cases}$$

$$y_1 + m_1(X - x_1) = y_1 + m_2(X - x_2)$$

$$y_1 - y_2 - m_1 x_1 + m_2 x_2 = (m_2 - m_1)X$$

$$\boxed{\mathbf{X} = \frac{y_1 - m_1 x_1 - (y_2 - m_2 x_2)}{m_2 - m_1}}$$

X is the projection on the X axis of the crossing point bewteen the two straight lines. Similarly one can obtain:

$$\boxed{\mathbf{Y} = \frac{m_2 y_1 - m_1 y_2 + m_1 m_2 (x_2 - x_1)}{m_2 - m_1}}$$

Which is the projection on the Y axis of the crossing point between the two straight lines. Now, given those two parameters, let us enumerate the controls made to check whether an edge belongs to GT or not.

2.2.1 Crossing point X

If $X \in ([X_a, X_b] \wedge [X_c, X_d])$, then the two edges will surely cross each other and the candidate edge can be purged from the GT list because, since we are on a planar graph, two edges cannot cross themselves without forming a node.

2.2.2 Coinciding Extrems

The check showed above is very powerful but wrong since if one of the two extrems coincide with one of the other two the edge will be purged from list even if it belongs to GT. For this reason we have to take into account during implementation the presence of such cases.

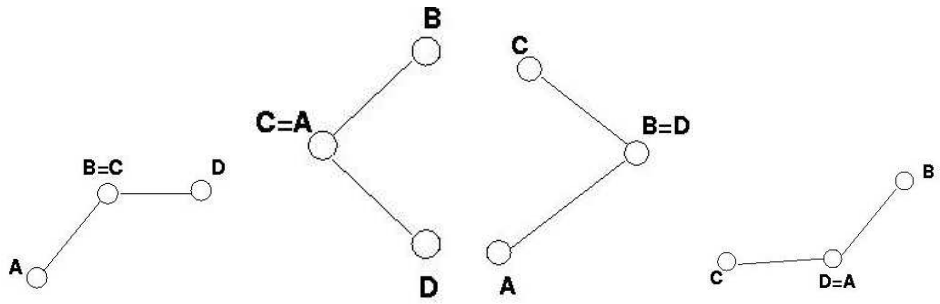


Figure 2.1: Edges with coinciding extrems.

Parallel edges

Moreover we have to take into account "parallel edges", i.e. edges which superpose over existing ones. This can be done putting another check.

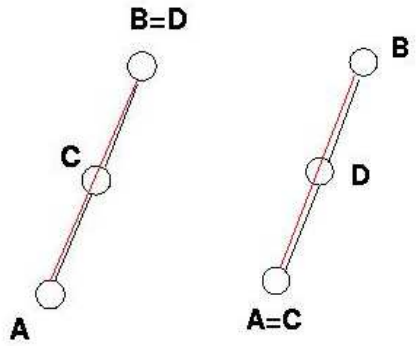


Figure 2.2: Edges superposing over themselves. The red edges are the superposing ones.

2.2.3 Intersections with edges parallels to an axis

Unfortunately the check based only on the parameter X (or eventually Y) is not enough to eliminate all the edges not belonging to GT. In particular, it has been observed that when one or more edges are parallel to an axis the algorithm shows some bugs that can be eliminated adding another check of the

same kind of the one used on X but, this time, applicated on the other parameter that is Y. In this way we can avoi certain peculiar configurations that can be encountered during simulation.

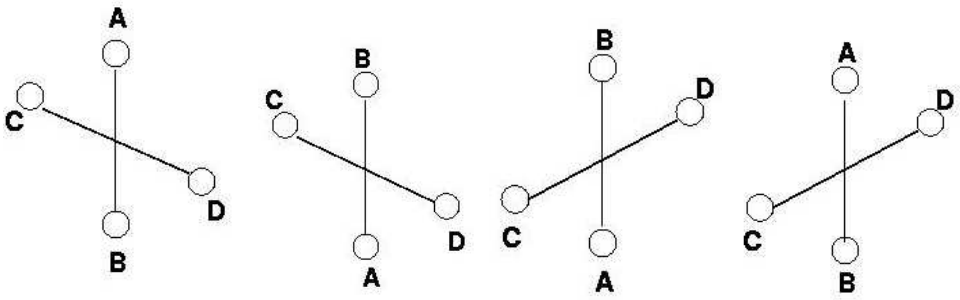


Figure 2.3: Edges parallels to Y axis.

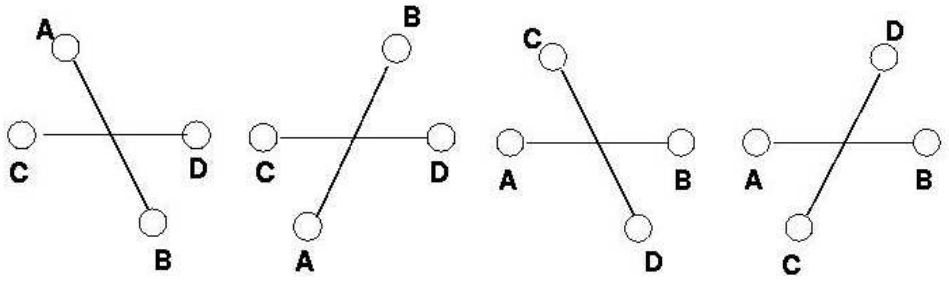


Figure 2.4: Edges parallels to X axis.

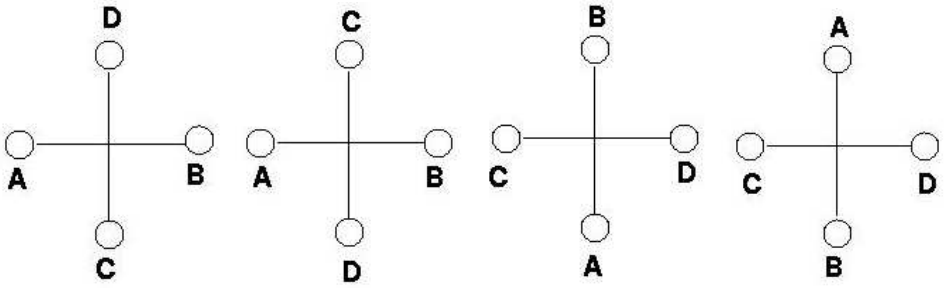


Figure 2.5: Perpendicular edges.