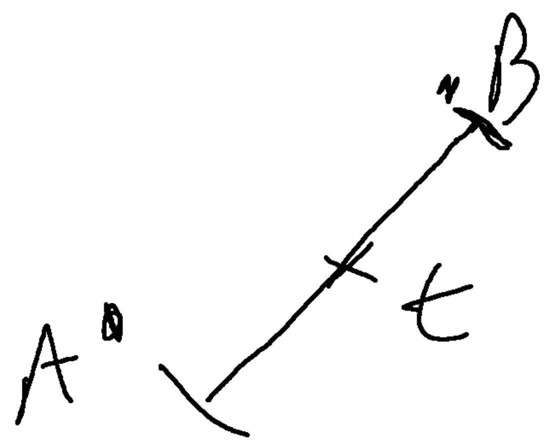
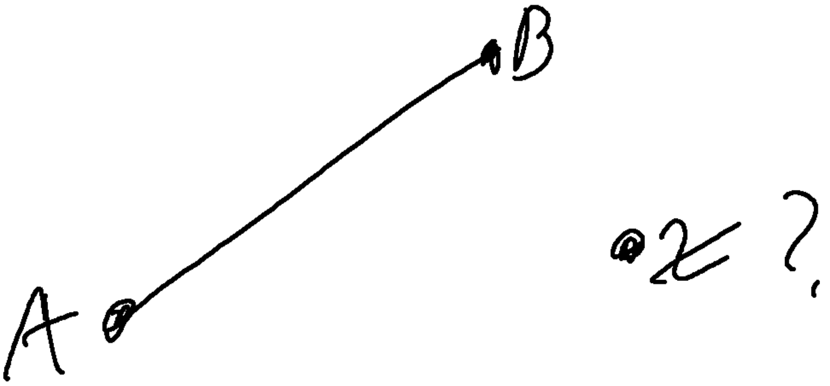
Given



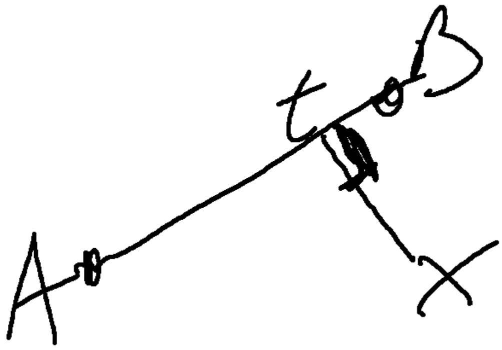
we can linearly interpolate the value as A transitions to B as t transitions from 0 (at A) to 1 (at B) whereby

$$x = \frac{(A \cdot t) + (B \cdot (1-t))}{2}$$

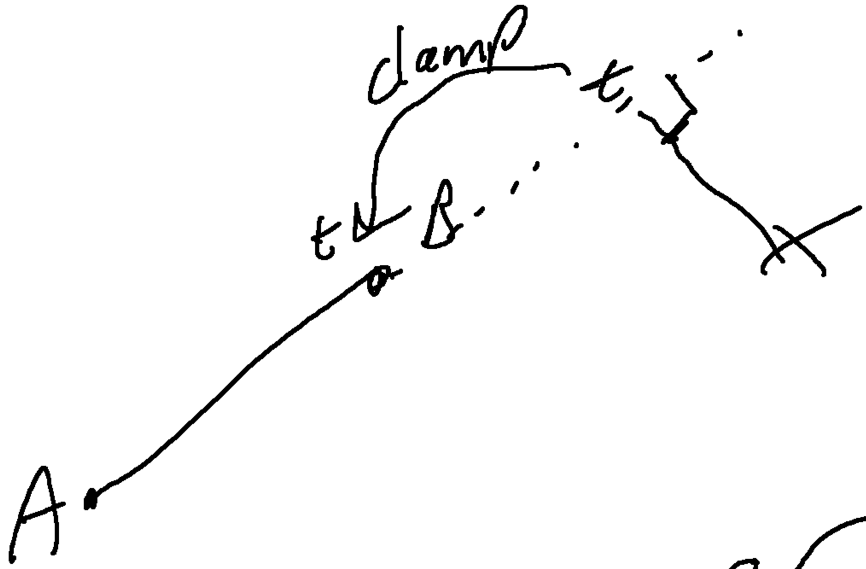
this essentially creates a weighted average. But what if our points x is now not on the line?



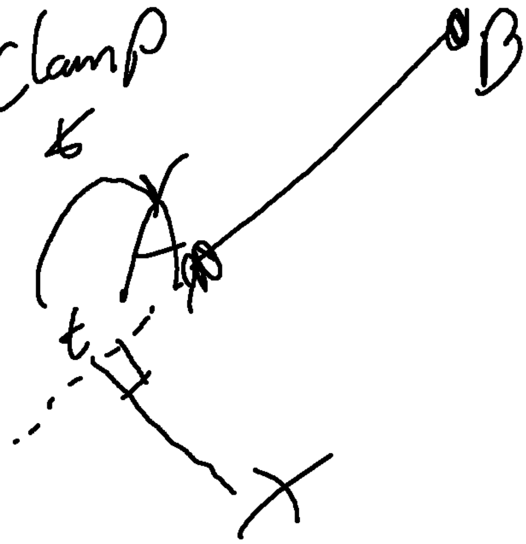
If we form a perpendicular from x to the line and clamp the value of t we can consider where in the interpolation x falls.



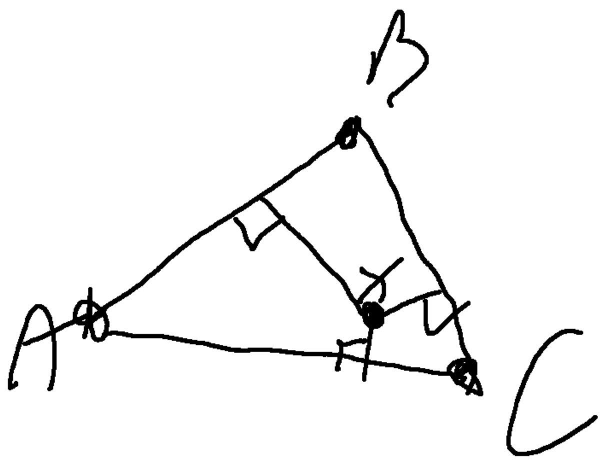
or



or

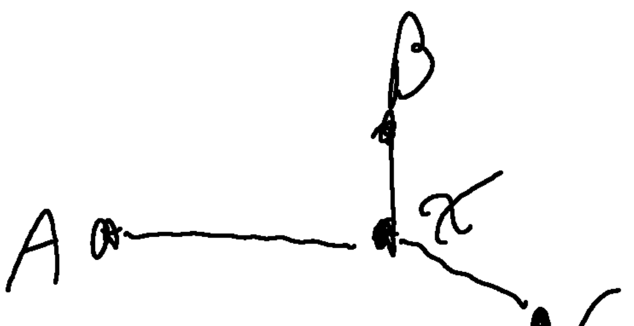


the problem is further complicated when we consider 3 points in our space



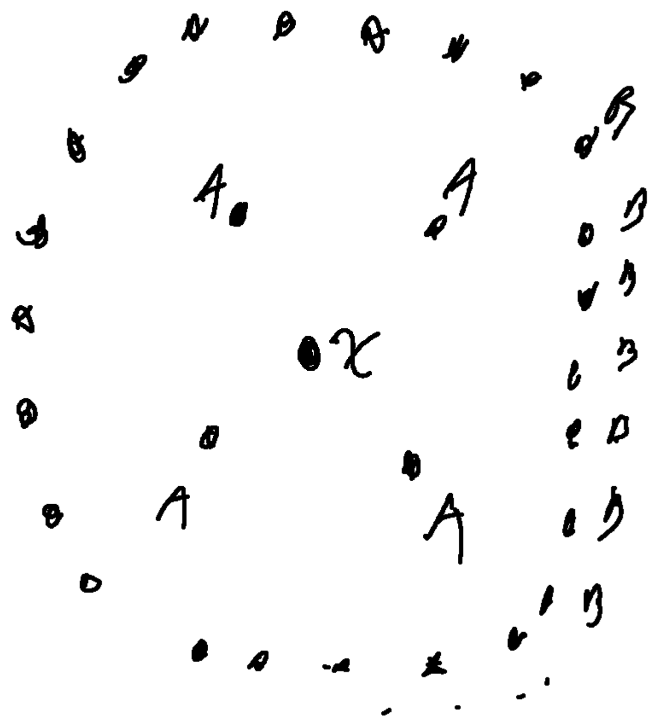
would we consider?

or?



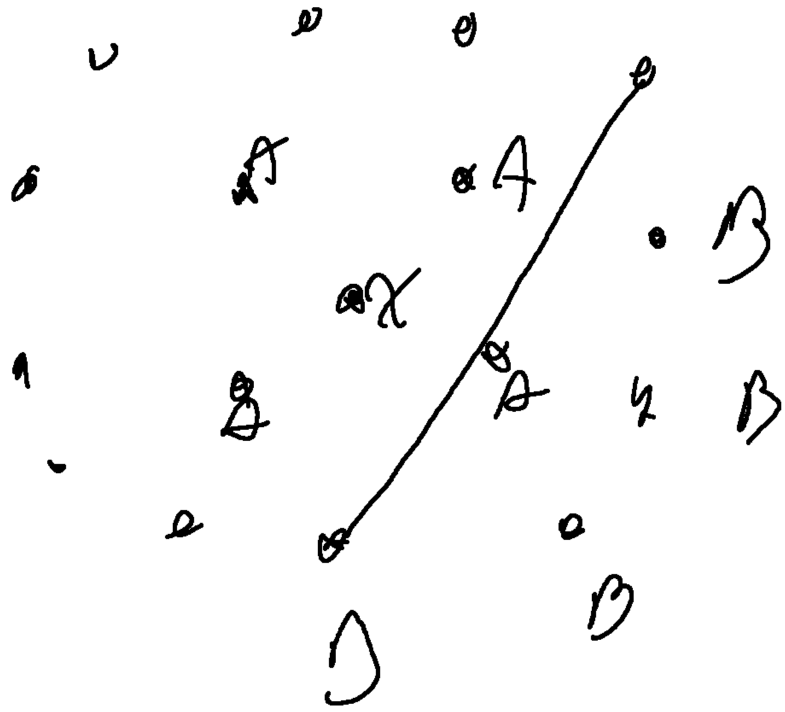
are these equivalent?
what about as $p \to \infty$
and triangular constraints don't hold?

further. What of the following situation:



Here if we consider an extremely dense region of B around a sparse region of A. Human intuition tells us that x is A, But if we combine the weights of all points surely the B's would overpower the A's. I suspect this kind of artifact is what I have been seeing.

kNN, seeks to undermine this artifact by saying just focus on the 3→4 closest points (4 in this case). But then k is blind to the entire surface, and even if not each region may have its own density dynamics, and herein is where we see neural networks take some advantage. Otherwise we need an extremely saturated dataset and a k value that speaks to that level of saturation; though at that points we are simply approaching a fuzzy identity function. So how do we overcome this double ray weight issue?!



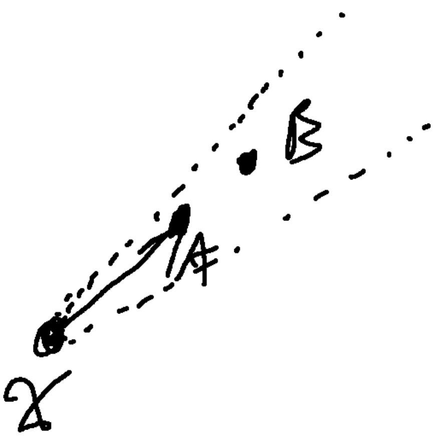
if we consider every pair of points and exclude any perpendicular to x whereby x falls outside the extents of the segment, we still have a preponderous of B-B projections that x falls under the influence of....

Even if we try a normalized distance or any kind of reciprocal falloff function we run into an issue if

the B ring is tight to the A Ring.
when distance is our only consideration
A and B became basically equals
and with for more B's their weight
wins out.

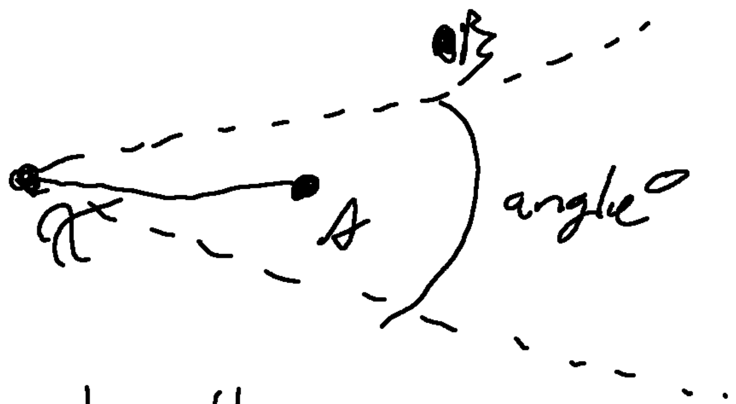
So if we need an adjunct to distance
its almost like a kind of obstruction
or shadowing?

consider



In this case B would be ignored as
A is obstructing or eclipsing the vision
of B as projected onto x.

However now we require same concept of cone of obstruction.

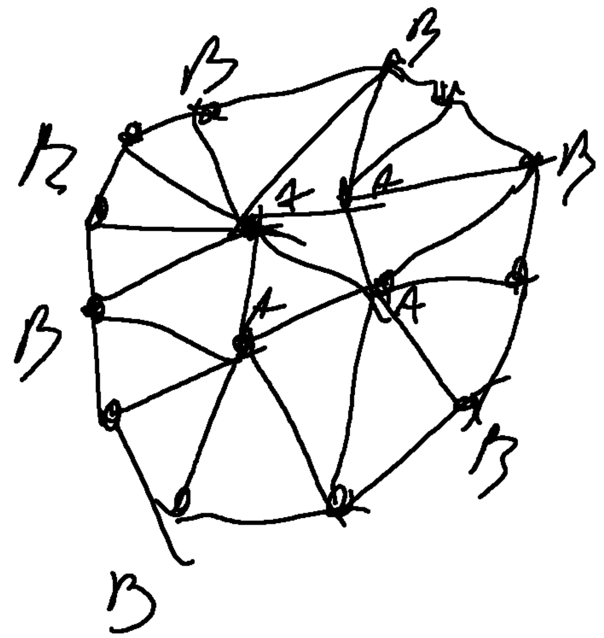


If we sort all neighbors from near to far, and select one at a time providing a angle of obstruction....

But we run into a similar kind situation where we cant have a one size fits all.

It appears we are actually looking for a sort of hulled map? of.... triangles?

→



this triangulation is greedy or an optimization in its own and basically is equivalent to $K=2$ with linear interpolation.

---

hmm maybe a better idea might be a decay rate, so

$K1 = 1.0$
$K2 = 0.5$
$K3 = 0.25$
$K4 = 0.125$
... etc.

they might rectify some issues but more likely would just be a novelty.