

quadtree-py

原始步骤

- 计算点间斥力
- 计算线条的弹力
- 计算阻力
- 更新速度
- 迭代

优化版本

优化版本主要是对于计算点间斥力的优化，原始计算每个点间的斥力为枚举两点，这样的复杂度是 $O(n^2)$

通过Quadtree可以优化到 $O(n\log n)$

主要思想在于缩点，对于符合一定条件的点，可以在计算时视作一个点，这样可以减少枚举的复杂的

Quadtree的思想在于对图空间的分割，简单实现是与图像空间大小成正相关的。但是实际上采用中位点对空间进行分割，可以达到更好的效果，类似的实现可以试试 KDTree

我的实现

在以上，实现了原始步骤的基础上，对于点间斥力的计算采用quadtree进行优化。

每次迭代完成更新quadtree。quadtree分为能四分和只能左右二分与上下二分三种情况。