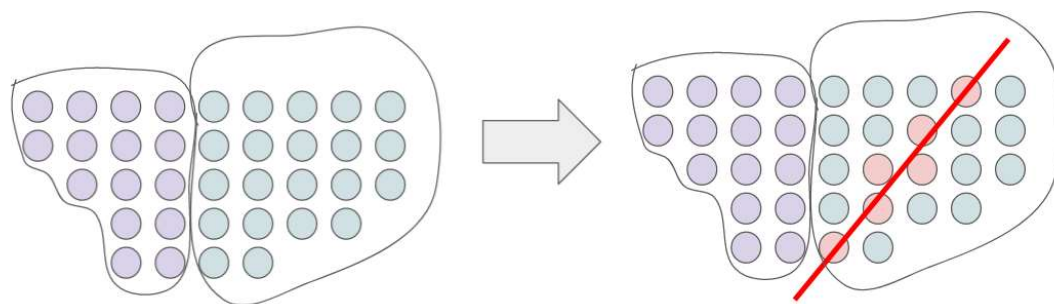


一、切割过程总览

1. 碰撞检测：识别出来哪些粒子在这一帧中需要进行“切割处理”。这一步是只读的，也就是说它不会修改任何粒子信息、簇信息。

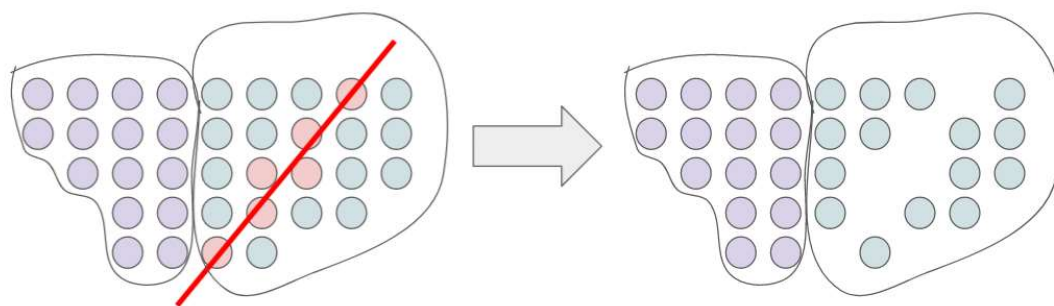
- 目前我们使用的碰撞检测方法是基于距离的，就是说当粒子离切割工具足够近的时候，它就需要被进行“切割处理”。一种可能的更优方案是计算体素应变，当应变特别大时就对其顶点进行“切割处理”。

碰撞检测



2. 切割处理：处理在上一步中识别出来的粒子。这一步只修改粒子信息。
 - 目前是直接删除这些粒子。一种可能的更优方案是将粒子分裂成两个。

切割处理

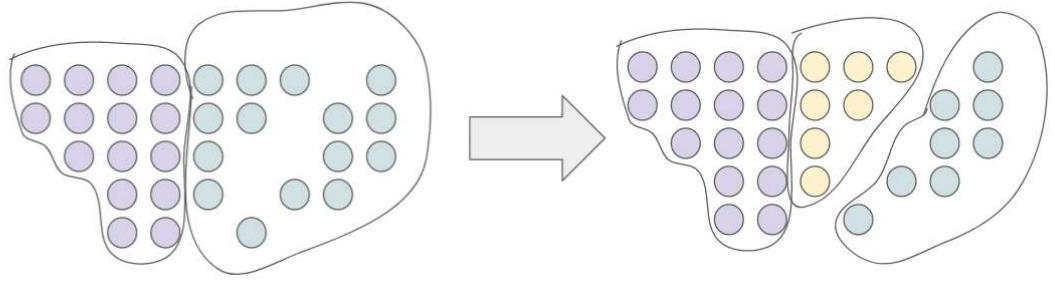


3. 新簇重构：因为在第二步修改了粒子信息，所以需要对应地修改簇信息。这一步只修改簇信息。
 - 本文档的内容就是这部分。

二、新簇重构算法总览

1. 称上一步“切割处理”中删除了的粒子为“被切割粒子”，并称这些粒子所在的簇为“被切割簇”（一般会有很多被切割簇，每个被切割粒子都会属于很多被切割簇）
2. 搜索被切割簇的极大连通子图集合，称这些子图为“待合并新簇”

搜索极大连通子图

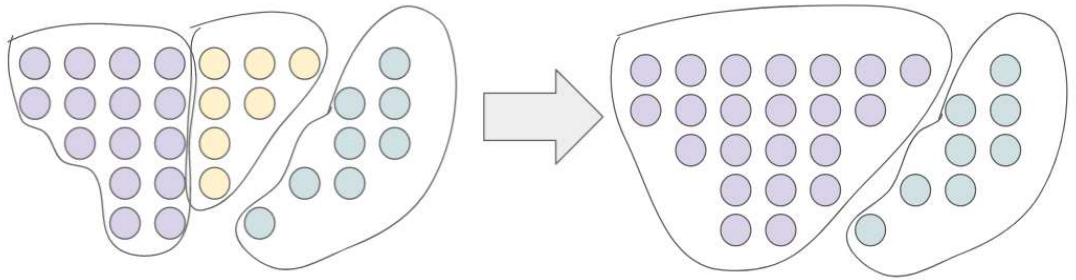


。

3. 对每一个待合并新簇：

1. 搜索与之相邻的一个粒子，随机选择该粒子所在的一个簇（与当前在处理中的待合并新簇不同的一个簇），称其为“邻居簇”
 - 若不存在与之相邻的粒子，说明该簇被从物体本体上切了下来，则直接将该待合并新簇作为一个独立的新簇存下来，转至下一个待合并新簇。
2. 将当前在处理中的待合并新簇合并进邻居簇中

合并进邻居簇



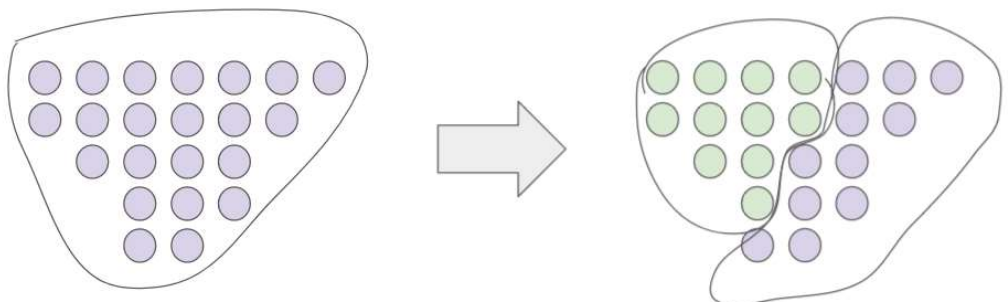
。

- 图中黄色新簇被合并进紫色邻居簇中，而青色新簇因为不与任何粒子相邻，所以被作为一个独立的新簇存了下来。后面我们不再讨论青色新簇。

4. 对上一步中处理过的每一个邻居簇：

1. 检查其粒子数，如果大于设定阈值，说明该簇过大、需要分裂；否则转至下一个邻居簇。
2. 分裂该邻居簇，得到两个新的小簇
3. 对小簇再次检查其粒子数，如果大于设定阈值，则继续分裂

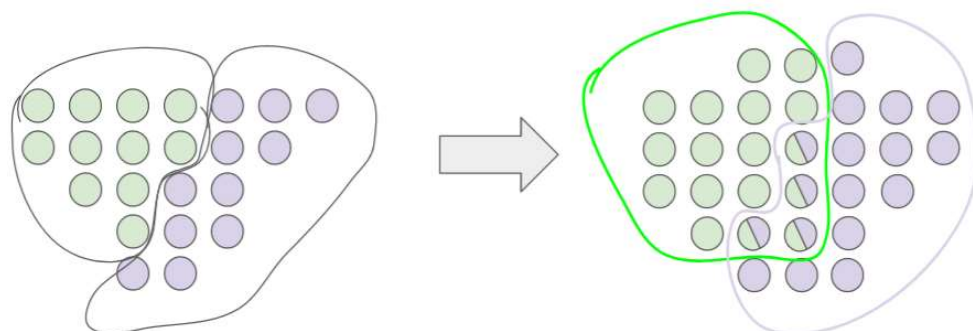
大簇分裂



。

5. 对上一步中得到的所有小簇进行拓展，使其包含若干相邻粒子

拓展新簇



。

- 。图中一半青色、一半紫色的点为同时属于青色小簇和紫色小簇的点。

关于为何使用这种基于拓扑结构的方法进行新簇重构，而不使用几何方法，理由我写在附录A里了。

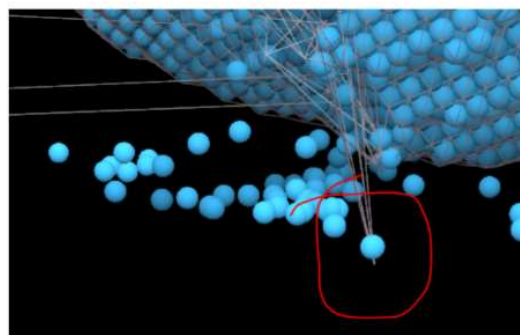
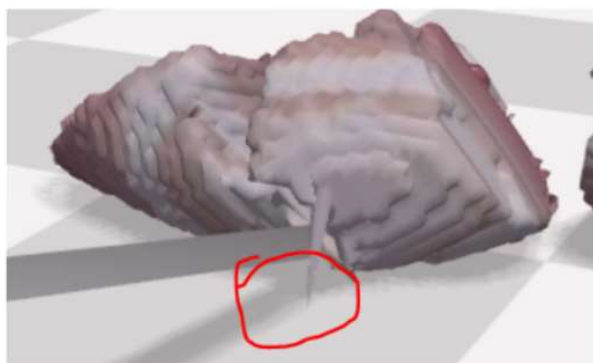
第二步搜索极大连通子图集合只是BFS而已，在后文中不会被讨论。

第三步合并至邻居簇中，随机寻找到的邻居簇可能本身就是“被切割簇”，但这不会构成问题，因为合并进去的粒子只是会增大之后第二步时找到的某个极大连通子图而已，并不会产生坏的影响。

第四步分裂大簇将在后文中详细讨论。

第五步小簇拓展也是BFS，随机向外拓展若干个粒子，从而与其相邻粒子建立更紧密的联系，避免变成孤立簇。

- 孤立簇指的是虽然在拓扑结构上与其他粒子相邻，但是却没有任何一个簇能够将这个簇和相邻的粒子建立起联系。视觉上的效果就是连在物体上的某一块儿突然拉了好长一条、掉了下来。如图：



三、一些数学定义

为了避免语义的混淆，此处解释一些数学定义。

一个簇为一张无向图。若一个粒子属于一个簇，则表示该粒子为该图中的一个顶点。若两粒子在拓扑结构上是相邻的（即，它们之间存在一条体素边），则在在该图中，它们之间存在一条边。

一个粒子可以到达另一个粒子是指这两个粒子之间存在路径。

连通图指的是该图中任意两顶点之间互相可到达。一张图的极大连通子图为一张无法再拓展的连通子图，即加入任何顶点进这张子图都会导致它不再是连通图。

一个簇的相邻粒子指的是该粒子与簇中的某一粒子之间存在一条边，即在拓扑结构上相邻。

一个粒子可以同时属于多个簇。

四、簇的连通性

我们始终假设一个簇是一张连通图。

- 除了“被切割簇”，这类簇因为部分粒子被删去，可能已经失去了连通性。

这一要求源自我们基于拓扑结构来划分新簇，而不是使用几何方法。如果一个簇被切割前本就不是连通图，那它被切割后划分出来的极大连通子图们并不一定就是因为切割而形成的新簇了，这会导致额外的切面。

算法过程中我们会利用该性质，并且也会始终维护该性质。

五、分裂大簇

注意，下文中所有的计算都是针对粒子的rest position的，就是物体发生形变前各粒子的坐标。之所以使用rest position，是因为簇表示的是刚体约束，如果我们使用current position的话，就会导致物体根据当前形变更新刚体约束，结果就是物体形变得越来越夸张。

- 在具体实现中，因为我们的粒子都是沿一个规则三维网格采样得到的，所以其rest position等价于其在该三维网格中的拓扑坐标。

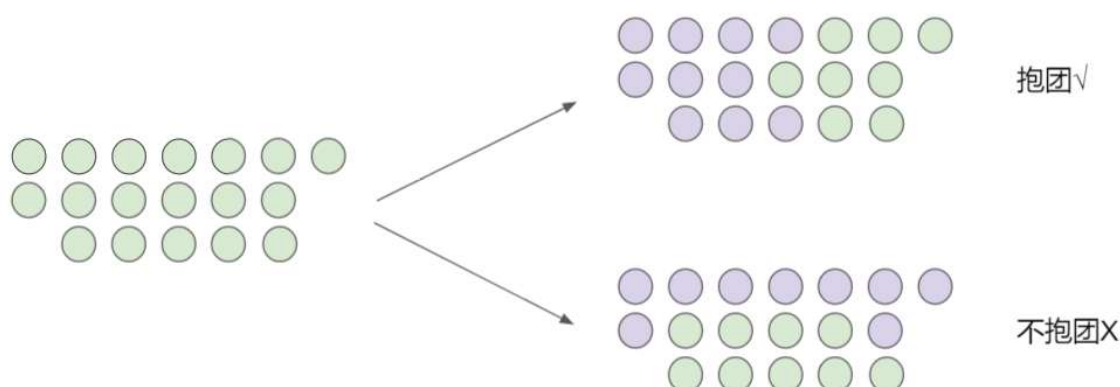
目标

之所以要分裂一个大簇是因为如果簇不停合并的话，簇的数量会越来越来少，结果就是物体呈现出的刚性越来越强。我们希望簇的数量、簇中粒子的平均数量尽量保持不变，从而使得物体刚性也保持不变。

这也指出大簇分裂算法的一个目标：分裂出来的新簇在粒子数量上应该尽量接近。

另一方面，因为我们在构建柔体时的簇采样算法是基于距离的聚类方法，所以我们希望分裂出来的簇也尽量接近该聚类方法的性质，也就是说：新簇内的粒子应尽量抱团。

- 举个例子：



最后，新簇还应为连通图，这一要求是因为我们假设所有簇都是连通图。

总结来说，大簇分类算法的目标：

- 新簇的粒子数量接近
- 新簇各自抱团
- 新簇为连通图

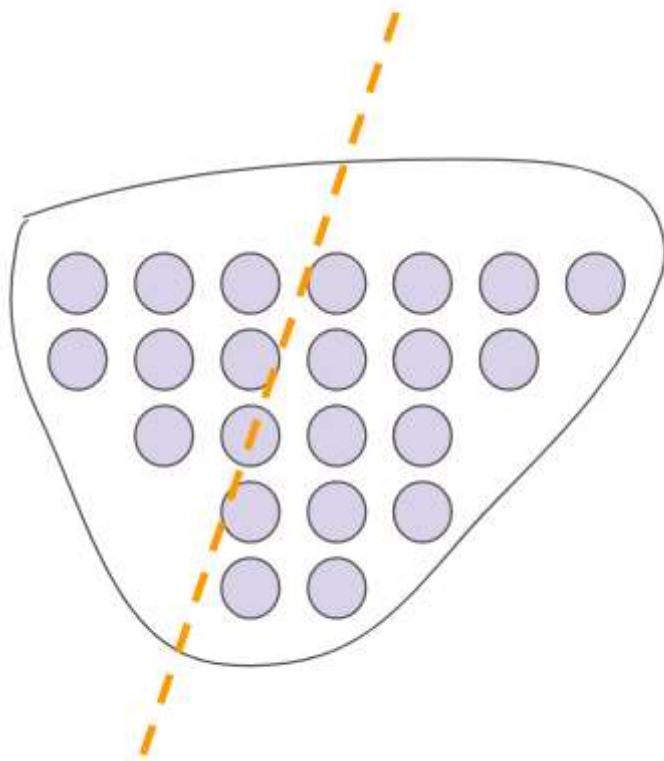
第一步：几何划分

因为新簇应为连通图，很容易想到的就是把原簇给一切为二、劈成两半。为此就要找到一个二维平面来劈开原簇。

这种情况下，各自抱团的目标可以解释为“粒子落到这个平面上的投影点尽可能紧凑”，也就是让它们在两个维度上抱团（TODO: 可能可以直接要求它们在三个维度上抱团，那样就是基于距离的聚类了，不过我这里是就这么解释成两个维度了）。

故而首先就是要找到“能使投影点尽可能紧凑的投影平面”，这一部分的详细内容我记录在附录B中。

计算最紧投影平面



在找到这么一个投影平面后，接下来要划分出两个新簇，使得这两新簇之间的‘切面’尽量贴近这个投影平面。这个问题可以抽象成：

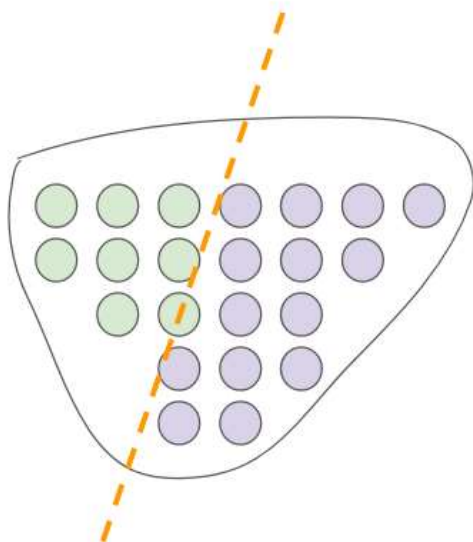
- 给定一个连通图 G 和平面法向量 n ，求 G 的一个连通子图 S ，使得 $M = G - S$ 也为连通子图，且 S 和 M 之间的‘切面’尽量接近一个平面，且该平面法向量尽量接近 n 。

在这第一步“几何划分”中，我们先解决‘切面’接近平面、平面法向量接近 n 的问题， S 和 M 的连通性留至下一步“拓扑修补”中再解决。

流程：

1. 随机选择簇中一点，设其为 v ，过它以 n 为法向量作投影平面
2. 设当前集合 S 和边缘集合 N ，初始时 $S=\{v\}$ ， N 中为 v 的邻居。
3. 当 N 中存在到投影平面的距离值为负数的点时（总有一面的距离值为负的，不过不一定有点在那一面就是了）：
 - 设该点为 x ，将 x 加入 S ，移出 N ，并将 x 未被访问过的邻居加入 N
 - 重复直到循环条件不满足

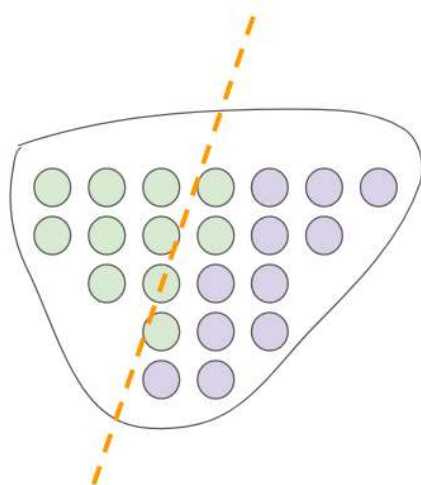
随机选择一点，沿平面的某一面搜索点集



◦

4. 若 $|S|+1 < |G|/2$:
 - 从 N 中找到最小距离值的元素 x ，将其移出 N ，加入 S ，并将 x 未被访问过的邻居加入 N
 - 重复直到循环条件不满足

拓展该点集，直至该点集大小接近总体的一半



◦

5. 若 $|S| > |G|/2 + 1$:
 - 反转平面法向量，则原本距离值为负数的点现在变为正数，原本为正数的点现在变为负数。
 - 跳转回第3步
 - 注：这一步不能直接将顶点移出 S ，因为那可能会破坏 S 的连通性。

此时 S 必定连通，但 $G-S$ 不一定连通(TODO:也可能其实必定连通？这部分有待考究)，故而需要对 $G-S$ 进行一些修补。

第二步：拓扑修补

- 对 $G-S$ 搜索其极大连通子图集合 H ，即 H 中的每一项都是 $G-S$ 的一个极大连通子图
- 考虑任意极大连通子图 $e \in H$:
 - 因为 e 极大，所以 e 不可到达 H 中其他任何子图。
 - 若 e 不可到达 S ，则因为 H 和 S 包含了所有顶点，故而 e 不可到达图中任何顶点。但原图为连通图，故而 e 应可以到达所有顶点，导出矛盾，因此 e 必定可以到达 S 。
 - 而因为 e 不可到达 H 中其他任何子图，但可到达 S ，且 G 中不包含 S 和 H 以外的任何顶点，所以 e 必定可直接到达 S 。直接到达的意思是 e 和 S 中各自存在一个顶点，这两个顶点之间有边相连。
 - 由上述证明可知， $\forall e \in H: e \text{ 可直接到达 } S \rightarrow S+e \text{ 为连通图}$
- 我们重复选择 H 中顶点数最少的子图 e ，将其加入 S ，最后得到 H 中最大的子图 M

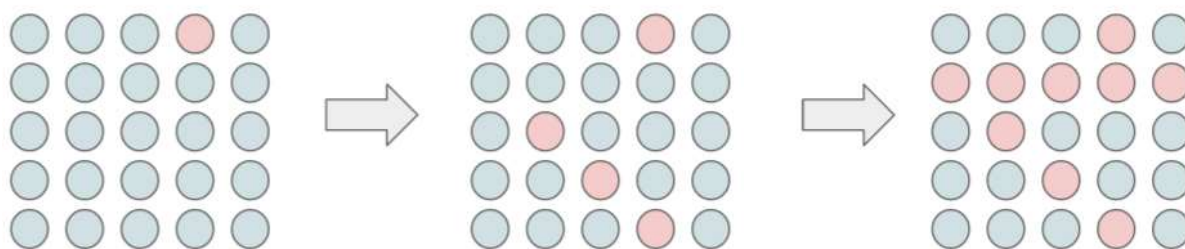
此时 S, M 都为连通图，且 $S+M=G, S \cap M=\emptyset$ 。

附录A：使用拓扑方法划分新簇的理由

1. 划分新簇时使用几何方法时的切面问题

如果切面刚好是个完美的平面的话，那划分的过程非常简单，只要简单地代入平面公式求个值就可以知道粒子在切面的哪一边了。但实际情况下，切面的形状可能会非常复杂。

例如：



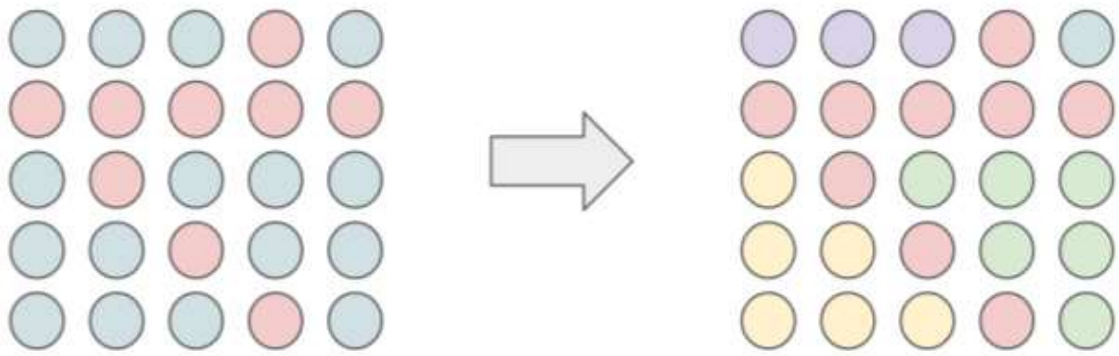
- 第一刀只砍了最上面的一个粒子，没有把簇切割成两半。
- 第二刀也只砍了最下面的三个粒子，也没有把簇切割成两半。
- 第三刀横向地砍了一行粒子，这才把簇切割开来，但是最终形成切面非常复杂，显然不是一个平面，结果簇被划分成了四块。

因为切面的复杂性，如果使用几何方法来判定哪些粒子该被划分进哪个新簇的话，过程会非常复杂：哪怕每一刀的切面都肯定是一个平面，最后划分簇时需要考虑的也会是由多个平面构成的不连续复杂曲面。

所以划分新簇时不应使用几何方法，而应该使用与切面复杂性无关的方法，

2. 划分新簇的连通子图方法

例如对上一节中的新簇分别用不同颜色着色：



- 这一方法的优点是它的划分效果与切面的复杂度无关，哪怕切面是不规则曲面也不会增加该方法的开销。
- 该方法的缺点是它对切割的第一步“碰撞检测”的要求很高，需要保证新簇之间一定互相不可到达才行。

附录B：寻找投影平面使得投影点尽可能紧凑

也发布到了知乎上

一、输入&问题

设点集 $S_x=\{x_1, x_2, \dots, x_n\}$ ，其中 $x_i = \begin{bmatrix} \alpha_i^1 \\ \alpha_i^2 \\ \vdots \\ \alpha_i^k \end{bmatrix}$

设向量 $X=\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$ ， $\alpha_j = \begin{bmatrix} \alpha_j^1 \\ \alpha_j^2 \\ \vdots \\ \alpha_j^n \end{bmatrix}$

也就是说 α_i^j 为点 i 的第 j 维， α_j 为由所有点第 j 维构成的行向量。

假设 $\sum_{i=1}^n x_i = 0$ ，即点集 S 的均值点为坐标原点。

我们的目标是在这个 k 维空间里找到一个 t 维“平面”（ $t < k$ ），将所有点投影到这个平面上后，投影点尽可能密集。

二、分析&目标

设投影后的点构成点集 $S_y = \{y_1, y_2, \dots, y_n\}$ ，其中 $y_i = \begin{bmatrix} \beta_i^1 \\ \beta_i^2 \\ \vdots \\ \beta_i^t \end{bmatrix}$

设向量 $Y=\begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$ ， $\beta_j = \begin{bmatrix} \beta_j^1 \\ \beta_j^2 \\ \vdots \\ \beta_j^n \end{bmatrix}$

设 t 维平面的基向量为 p_1, p_2, \dots, p_t ，其中 p 为 k 维列向量，则有由 k 维空间的基到 t 维平面的基的过渡矩阵 $P = \begin{bmatrix} p_1 & p_2 & \dots & p_t \end{bmatrix}$ ，满足 $P \underset{i}{y} = \underset{i}{x}$ 。

则有

$$\begin{aligned} Y &= [y_1 \ y_2 \ \dots \ y_n] \quad \&= [P^{-1}x_1 \ P^{-1}x_2 \ \dots \ P^{-1}x_n] \quad \&= P^{-1} \\ [x_1 \ x_2 \ \dots \ x_n] \quad \&= P^{-1}X \end{aligned}$$

因为 P 为绕坐标原点的旋转矩阵（任何过渡矩阵都是绕坐标原点的旋转矩阵），所以 $\overline{y} = \overline{x} = 0$ ，即点集 S_y 的均值点也为坐标原点。

我们要求投影平面，就是要求这个 P ，下面讨论 P 应满足的条件。

目标1: $\underset{i}{p}$ 线性无关

因为我们希望投影后确实是在一个 t 维“平面”上，而不是在 $t-1$ 维“线段”之类的上面。所以变换后的各维应该要线性无关。

- 例如，想象三维空间投影到二维平面上，如果构成二维平面的两个基向量相互平行的话，那其实是定义不出这个二维平面的，只能根据一个点和一个方向定义出一条直线。所以我们希望这两个基向量能线性无关。（事实上，线性无关是包含在基向量的定义中的）

所以 t 维平面的基应由 t 个线性无关的向量构成，也就是 p_1, p_2, \dots, p_t 两两线性无关。

目标2: $Cov(\underset{i}{\beta}, \underset{i}{\beta})$ 尽可能小

我们希望投影后的点尽可能“密集”。密集的意思就是各点相互之间离得尽量近一些，我们可以使用方差来描述这个概念。一组数据的方差指的是这组数据的离散程度，例如一组点的 x 坐标的方差描述的是它们在 x 轴上有多分散。

所以要想让投影后的点尽可能密集，就是要让它们在 t 各维度上的方差都尽可能小。

也就是

$$\min_{j=1,2,\dots,t} Cov(\beta_j, \beta_j) = \frac{1}{n-1} \sum_{i=1}^n (\beta_j^i - \overline{\beta_j})^2$$

其中 $\overline{\beta_j} = \frac{1}{n} \sum_{i=1}^n \beta_j^i = 0$ ，为各点在第 j 维上的均值。

目标3: $i \neq j, Cov(\underset{i}{\beta}, \underset{j}{\beta}) = 0$

$$\begin{aligned} \text{由 } PY &= X \text{ 得 } P \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_t \end{bmatrix} = X \\ X &\rightarrow \sum_{i=1}^t p_i \beta_i = X \end{aligned}$$

考虑 $t=2$ 的情况，也就是投影到二维平面上时：

$$p_1 \beta_1 + p_2 \beta_2 = X$$

若 $\beta_2 = h\beta_1 + d$ ，即两者呈线性关系，则：

$$(p_1 + hp_2) \beta_1 = X - p_2 d$$

注意到此时虽然 p_1, p_2 线性无关，但是所有点其实都沿 $p_1 + hp_2$ 这个方向分布（等号右边的 $X - p_2 d$ 为定值），也就是说原本的二维平面此时退化成了一维直线。

而我们希望的是投影到一个二维平面上，所以我们希望避免这种退化，也就是希望 β_1, β_2 不能够相互线性表出，这两个维度上的样本值不具有线性相关性，使用皮尔逊相关系数来表示这一性质（注意，仅仅只是 $\beta_1 \neq h\beta_2 + d$ 是不够的，因为可能从里面去除几个点后，剩下的点又退化成为直线了，我们是希望所有点都不要退化成为直线）：

$$\rho_{\beta_1, \beta_2} = \frac{Cov(\beta_1, \beta_2)}{\sigma_{\beta_1} \sigma_{\beta_2}} = 0$$

$$Cov(\beta_1, \beta_2) = 0$$

即：

可以简单泛化到t维平面的情况，也就是当 $i \neq j$ 时：

$$Cov(\beta_i, \beta_j) = 0$$

三、推导&推论

推论1：协方差矩阵 Y_c 为对角矩阵

以点集 S 为样本集，以 $\beta_1, \beta_2, \dots, \beta_t$ 为随机变量，设其协方差矩阵为 Y_c ：

$Y_c = \begin{bmatrix} Cov(\beta_i, \beta_j) \end{bmatrix}, i \text{ 控制行}, j \text{ 控制列}$

则若要满足目标3：当 $i \neq j$ ， $Cov(\beta_i, \beta_j) = 0$ ， Y_c 就需要是一个对角矩阵：它只有主对角线上元素可以非零。

推导1： $Y_c = P_{-1}^T X_c P_c$

$Y_c = \begin{bmatrix} Cov(\beta_i, \beta_j) \end{bmatrix}, i \text{ 控制行}, j \text{ 控制列} \quad \& \& (\text{because } \overline{Cov(\beta_i, \beta_j)} = 0) \quad \& \& = \begin{bmatrix} \frac{1}{n-1} \sum (\beta_i - \bar{\beta}_i)(\beta_j - \bar{\beta}_j) \end{bmatrix} \quad \& \& = \frac{1}{n-1} \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_n \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_n \end{bmatrix}^T \quad \& \& = \frac{1}{n-1} Y Y^T$

同理， X_c 的协方差矩阵 $X_c = \frac{1}{n-1} X_c X_c^T$

然后，考虑过渡矩阵 P ，其满足 $P Y_c = X_c$ ，因此：

$$\begin{aligned} Y_c &= \frac{1}{n-1} Y Y^T \\ &= \frac{1}{n-1} (P_{-1}^T X_c) (P_{-1} X_c)^T \\ &\quad (\text{P is orthogonal matrix, so } P_{-1}^T = P) \\ &= \frac{1}{n-1} P_{-1}^T X_c X_c^T P \\ &= P_{-1}^T \left(\frac{1}{n-1} X_c X_c^T \right) P \\ &= P_{-1}^T X_c P \end{aligned}$$

故而要让 Y_c 为对角矩阵，就是要让 X_c 对角化。

推导2： X_c 相似对角化

注意到 X 为实对称矩阵，故而其必定可以相似对角化。

设 X 的特征值从小到大为 $\lambda_1, \lambda_2, \dots, \lambda_c$ （若存在多重特征值，则分开写成多个 λ ），对应的单位特征向量（列向量）为 v_1, v_2, \dots, v_c 。并设 $V = \begin{bmatrix} v_1 & v_2 & \dots & v_c \end{bmatrix}$, $\Sigma = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_c)$ ，则有：

$$V^{-1} X V = \Sigma$$

推论2: P 等于 V 的左上 $k \times t$ 子块， Y 等于 Σ 的左上 $t \times t$ 子块

首先检查目标1，我们需要 p_i 线性无关。而 v_i 为 X 的特征向量，所以必定两两线性无关，故而满足目标1。

然后检查目标2，我们希望 Y 对角线上的值尽可能小。因为我们取的是 Σ 的左上角子块，而特征值按照从小到大的顺序排列，所以 Y 在对角线上的值会尽量得小。TODO:关于这一点，我并没有严格证明，也没有逻辑推导，只是感觉差不多。

最后检查目标3，因为 Σ 是对角矩阵，所以其左上角子块也为对角矩阵，所以 Y 为对角矩阵。

四、结论

若 X 的均值点为原点，则通过相似对角化 X 的协方差矩阵 $X = \frac{1}{n} X X^T$ ，得到 $V^{-1} X V = \Sigma$ ，随后使 P 为 V 的左上角 $k \times t$ 子块，就能得到过渡矩阵 P ，利用它就能变换得到投影到 t 维“平面”上的点 $Y = P^{-1} X$ ：在该投影“平面”上，投影点尽可能地紧凑。

- 相似对角化 X 时，因为它为实对称矩阵，故而必定为 Hermitian Matrix，可以直接应用 eigen 里的 SelfAdjointEigenSolver 来计算。
- 其他的计算就都只是矩阵乘法了。