1. Border simplify.
2. Set an expected distance d1 for each two adjacent points. Start with the biggest curvature point, each patch no longer than d1, choose the maximum curvature point.
3. Check if this point satisfies the envelope condition or far enough. If violate any of these two conditions, find a point not violate. (also record the original point, because we may need it in (3)).
4. Check if any interior point is outside the border (Delaunay). If so, add a point in the related region. This point can be a local maximum curvature point, or a mid point of the corresponding region.

Why we choose interpolation over approximation? For approximation, it is hard to guess how many control points are required. To reach a tolerance it often requires several iterations. However, interpolation can avoid unnecessary iterations. The problem will be: for an unstructured mesh, how to use optimal number of control points to interpolating the surface, while avoiding wiggling.

The total structure:

1. Given a mesh. Parameterization (mean value coordinate?)
2. Knot clipping/merging (Constrains: 1. No flipping/inversion triangles during clipping; 2. Minimize some energy such as thin-plate energy to minimize the wiggling.) (or we can do: we first build a good/uniform grid, then decide which corners of the grid does the point’s parameter go) (or the energy can be related to the parameter speed)
3. Do we still need to add the missing points?
4. No, because the thin-plate energy or other kinds of energy can do that; (fair ahead)
5. Yes, in this sense we can use a Laplacian smoothing to “implicitly” build a quad-mesh. (fair later)
6. Interpolation in one direction. First decide how many control points are required according to the point distribution (the math condition). Then build the equivalent constrain, while minimizing the energy (if missing points are added, we use the least squared distance of the points; otherwise we use other kind of energy to fair the mesh, while guaranteeing the equivalent condition). Iterate to make the surface fair.
7. Interpolating the control points of step 4.

亮点：

1. 稀疏、散乱数据参数值给定条件下，给出了一个插值条件下的控制点数量计算方法和控制点插入位置确定方法。
2. 证明了无论是直接构建等式约束，还是先u后v构建，其结果是一样的？其求解自由度、控制点数量是一样的？
3. 给出了一个光顺化能量函数，其离散形式和解法。
4. 给出了一个节点矢量合并方法/能量函数，证明了保持拓扑结构的充分条件——三角网格没有flipping，并推导了节点数量和控制点数量的关系，证明了节点矢量合并后的