vector<LandmarkCollection<Eigen::Vector2f>> per\_frame\_landmarks;

多个视角的68个关键点

vector<float> pca\_shape\_coefficients;

多个视角公用一组Shape系数

vector<vector<float>> blendshape\_coefficients;

每个视角有单独的一组Expression系数

vector<core::Mesh> per\_frame\_meshes;

每个视角单独计算一个模型

vector<vector<Vector4f>> model\_points; // the points in the 3D shape model of all frames

vector<vector<int>> vertex\_indices; // their vertex indices of all frames

vector<vector<Vector2f>> image\_points; // the corresponding 2D landmark points of all frames

初始值是68个关键点和配置文件配置的对应点

# 计算一组仿射矩阵

# 每个视角计算一组Expression系数

使用最小二乘法求解现行方程 令

求解方程

# 迭代计算Shape系数和Expression系数

## 每个视角查找2D和3D对应点

每个视角单独分3个步骤查找2D和3D对应点，根据对应点重新估计仿射矩阵

## 拟合Shape系数

将所有视角的2D和3D对应点拼接成对应的矩阵，拟合Shape系数

## 每个视角计算一组Expression系数

使用公用的Shape系数每个视角但是计算Expression系数