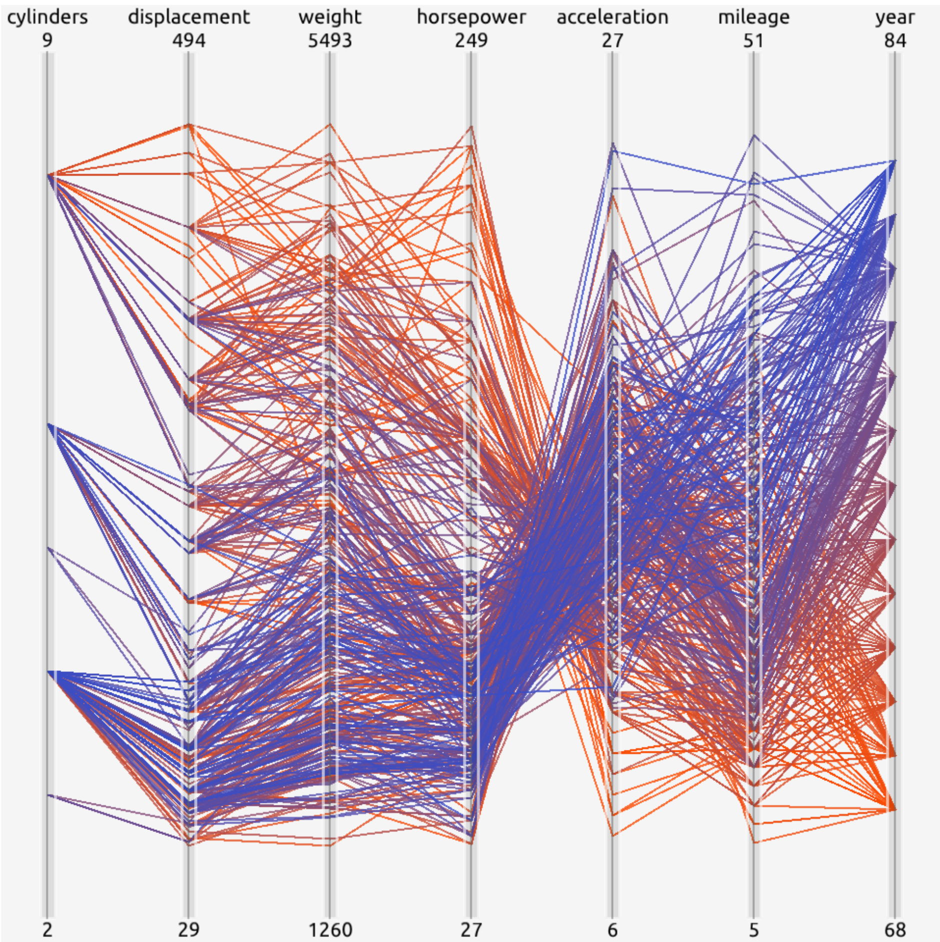
Lab5 Report

叶茂 2200017852

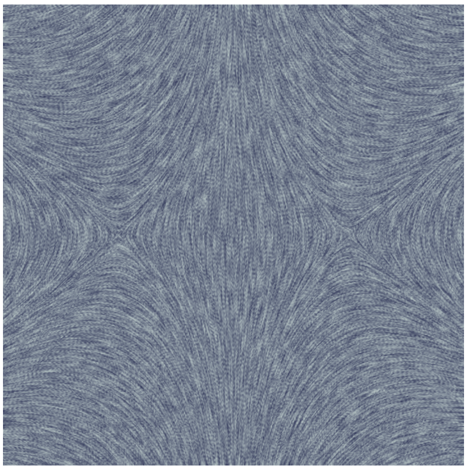
1 Parallel Coordinates Visualization

在CoordinateStates类中先定义成员变量data、构造函数、Update函数和Paint函数。由于暂不实现鼠标交互功能，因此Update函数始终返回false。在Paint函数中，首先绘制背景，随后对于每组数据，算出每个数据在标识轴上的占比，以此确定在画布上的百分比坐标，随后在每个数据点间绘制直线，并利用一个计数变量实现颜色的渐变。最后绘制标识轴方框以及相关文字和数据范围即可。在PaintParallelCoordinates函数中，使用传入的参数初始化上述状态类，若无鼠标交互和强制绘制则返回false选择不绘制；否则进行绘制并返回true。

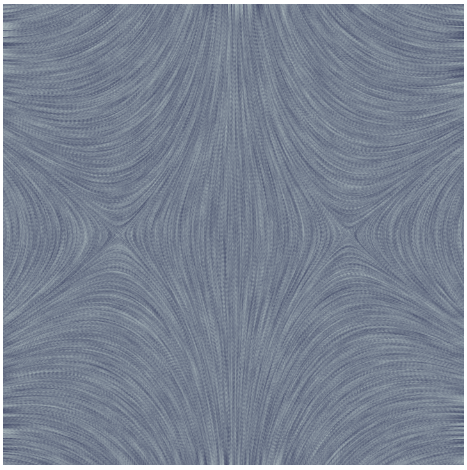


2 Flow Visualization

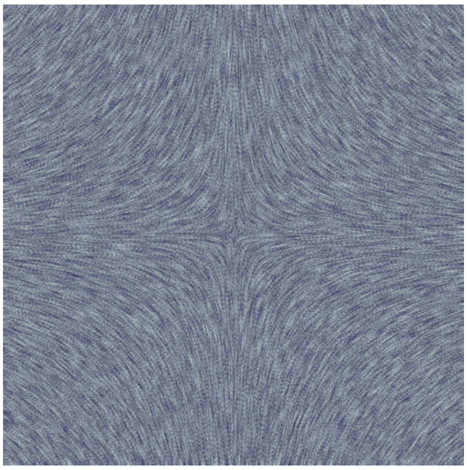
首先初始化`output`保证输出图像尺寸与输入流场尺寸一致。根据`LIC`算法的思想，对于输出图像的每个像素点，我们需要在流场中找到对应点并获取速度，在前向插值过程中。每次向着速度方向前进一小步，获取`noise`图像中对应颜色并乘以权重值；在后向插值则是沿着速度的反方向每次前进一小步。前向和后向分别`step`步，将所得结果 (包含当前点) 累加并用权重总和归一化就得到了输出图像对应像素的值。需要注意坐标范围防止越界。我使用了两种不同的权重函数: 一是仿照`tutorial`中给的代码样例使用 cos^2 量级的函数；二是使用高斯函数作为权重函数。



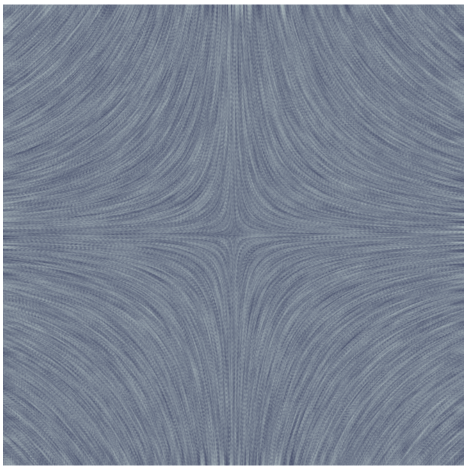
(a) step=10 with cos^2 function



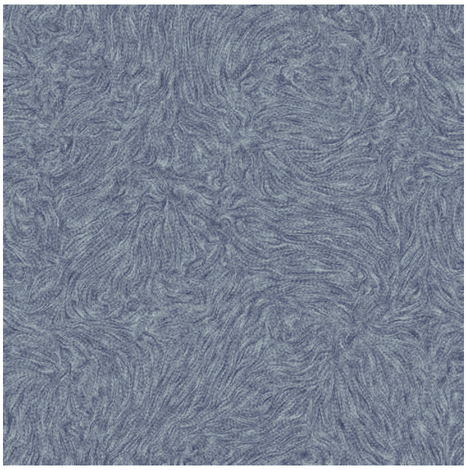
(b) step=30 with cos^2 function



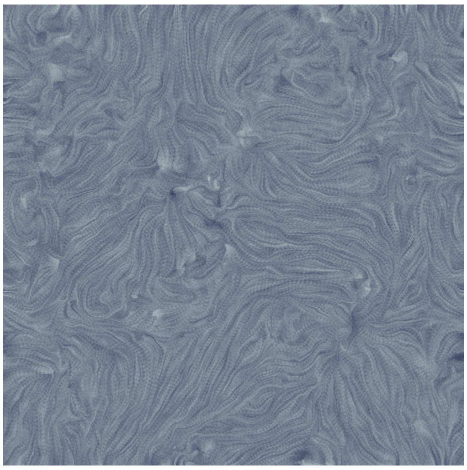
(a) step=10 with cos^2 function



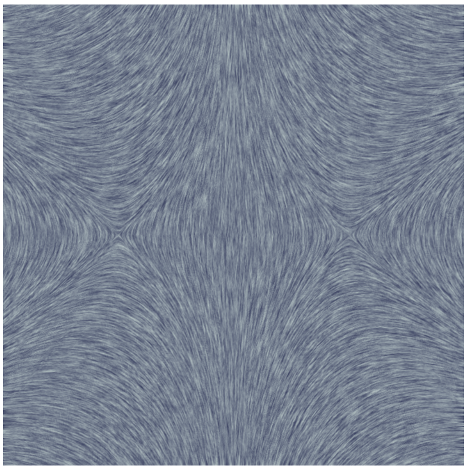
(b) step=30 with cos^2 function



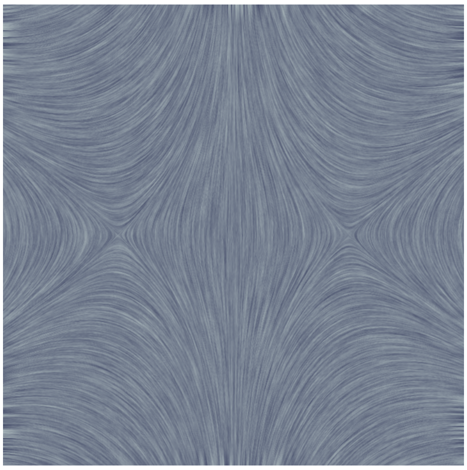
(a) step=10 with cos^2 function



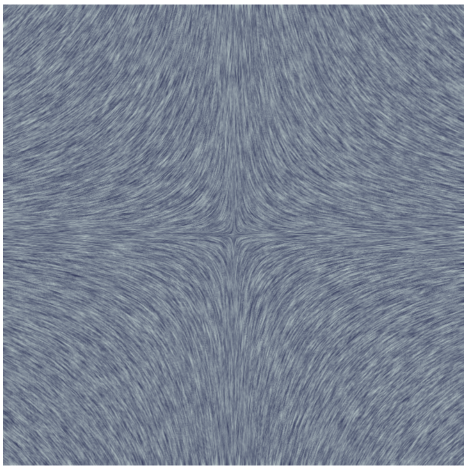
(b) step=30 with cos^2 function



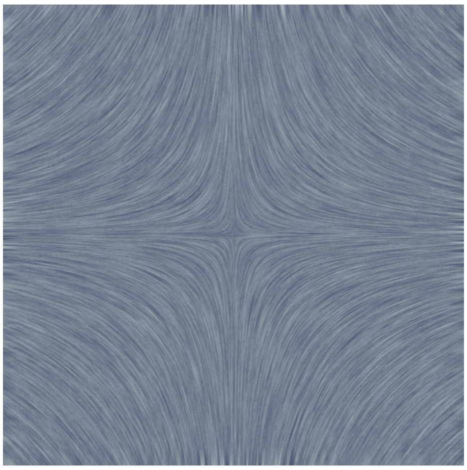
(a) step=10 with gaussian function



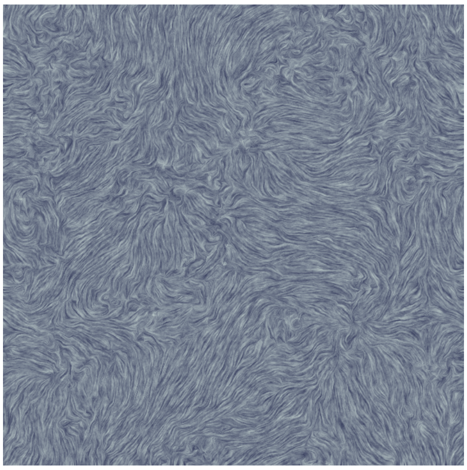
(b) step=30 with gaussian function



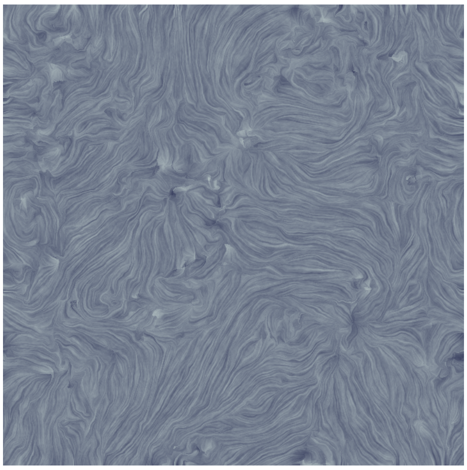
(a) step=10 with gaussian function



(b) step=30 with gaussian function



(a) step=10 with gaussian function



(b) step=30 with gaussian function