CIS 572: Streamline Development via Autoencoders

Algorithmic Development

Steven Walton

May 9, 2019

1 Project Description

In this project we will investigate the use of autoencoders to develop streamlines. Inspired by FlowNet In this project I will use this idea to build a variation upon this method that has an input of streamlines and the decoder outputs only the relevant pathlines.

1.1 Background

There is an open problem in flow visualization about how to make good pathline visualizations. While it is simple to create a large number of them, this can be overwhelming for the domain scientists and can actually hide important information, either by occlusion or distraction. Figure 1 shows an example image that has too many streamlines. Conversely, having too few streamlines means that a feature may be missed. Additionally even with the same number of streamlines, not picking the right ones can lead to a different interpretation of an image. Figure 2 shows an example where both images have the same number of streamlines but show different relevant features. In Image A we can see that the blue regions have redundant features, lines that are really similar. In Image B we see sinks and sources that were not as apparent in Image A. This demonstrates only some of the challenges in generating good visualizations for streamlines.

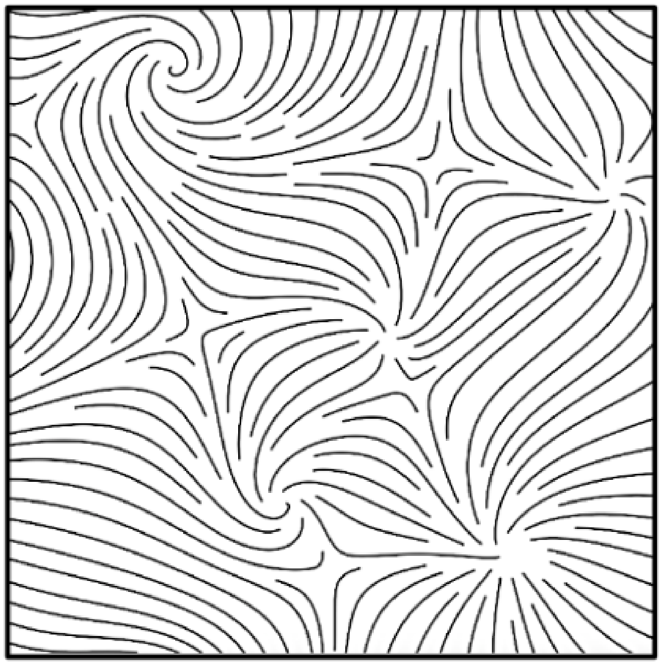

Figure 1: Example of many too many streamlines

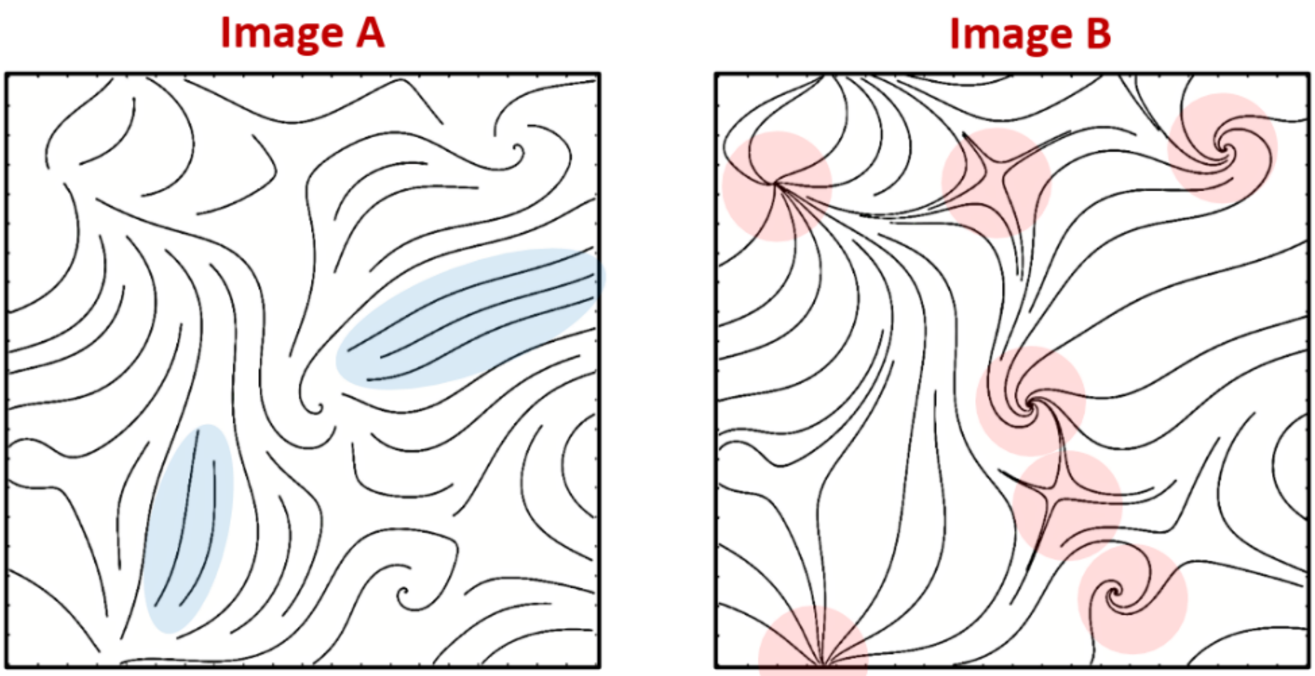

Figure 2: Both images have 50 streamlines but show different information

2 Work Plan

As may be inferred from the background, we can think of features as an encoding. This suggests that autoencoding might be a useful tactic to generate good images. Being that I am attempting to develop a new method, several different things will have to be attempted and there is a high risk that a good encoder is not completed by the time the class ends, though this is research my advisor is interested in and would like to pursue if these trials show promise.

The first attempt will be to try to use an autoencoder on generated streamlines and try to reduce these streamlines to only relevant ones (an open problem). If success can be achieved here it appears that the next logical step is to test it against a variational autoencoder. My inclination is that the probabilistic nature of the variational encoder may help escape local minima and reveal better features. It is hopeful that I will be able to achieve this by the end of the class.

As a continuation, likely beyond the scope of this class, I would like to find a way to generate streamlines using an autoencoder directly from vector fields. If this can be done, then a preprocessing step can be removed and this will be a big step forward for use in applications. I believe this may be possible because we can imagine that these streamlines are an encoding of the vector fields themselves. Figure 3 shows two streamlines overlaying a vector field. We can see that the streamlines are showing how the vectors point.

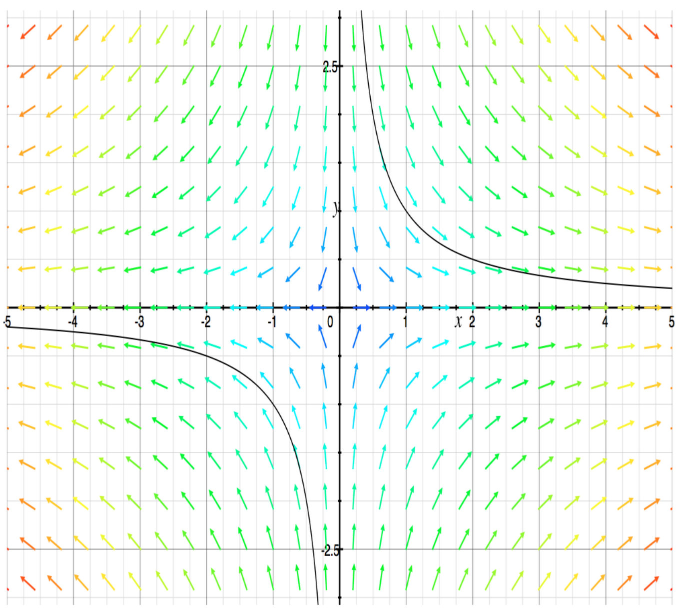

Figure 3: Two streamlines overlaid on a vector field

It may be possible to think of streamlines as an encoding of the vector directions, excluding their magnitude. Scientists already calculate out these vector fields, and need this for further calculations. Therefore it may be possible to teach an autoencoder to represent the vector field as streamlines.