Prof. Dr. Michael Krone
Marco Schäfer
Marcel Bok
Big Data Visual Analytics in Life Sciences
Department of Computer Science & IBMI
Faculty of Science

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Scientific Visualization

Assignment 10

SS/2021

Due: July 29th, 2021, 9:00 am

Code skeleton:

We supply you with a basic code template to use for this assignment but you are free to use the code you wrote for assignment one as a basis for this task. Our supplied code will contain code snippets assisting you in solving the tasks. So be sure to look at the supplied code if you choose to use your own code as a basis for this assignment.

1 Vector Field Visualization (10pts)

In this last Assignment you are tasked to generate and visualize a two-dimensional vector field.

1.1 Generate Vector Field Data Texture (3 pts)

Using the `sampleVectorField()` function from `utility.js` to calculate a vector field in the appropriate data structure and generate a `THREE.DataTexture()` from it. The function `sampleVectorField()` accepts three arguments, an x- and a y-coordinate as well as a time point. Both x and y, are in the domain $[-2.0, 2.0]$, while t is in the domain $[0.0, 4.0]$. For this task, use a fixed `t` of 1.0.

In order to check if your vector field is correct map the vector components to colors and attach a figure. The vectors have two components. Use them for red and green and set blue to zero. The result should look like Figure 1.

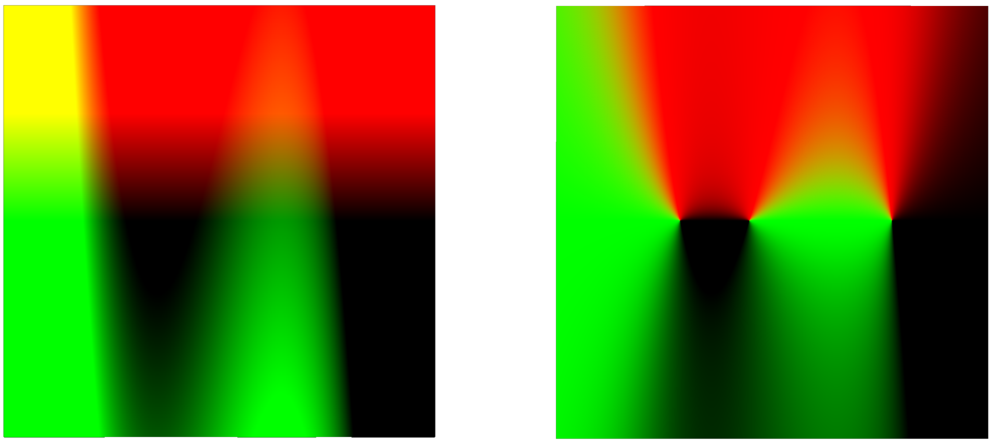

Figure 1: Vector values mapped to fragment colors at t=1.0 (left: vector values; right normalized vector values).

1.2 Visualize the Vector Field using Arrows (4 pts)

Now you asked to visualize the vectors as arrows (triangles). Align the triangles to the directions of the vectors. Draw one triangle-arrow per grid cell as shown in Figure 2.

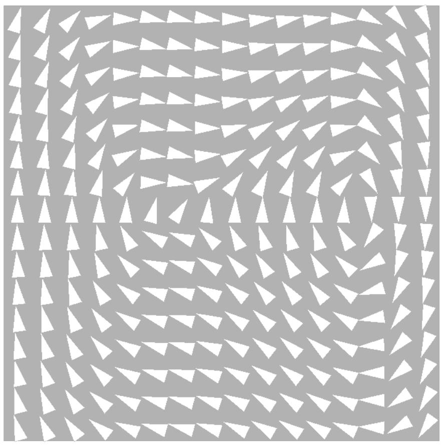

Figure 2: Arrows (triangles) visualizing the vector direction at t=1.0

After that, visualize the magnitude (length) of the vectors using a color scale from white to red. Normalize your magnitude values to make good use of the color range (e.g., map the minimum and maximum magnitude values to white and red, respectively). After this you should get the result shown in Figure 3.

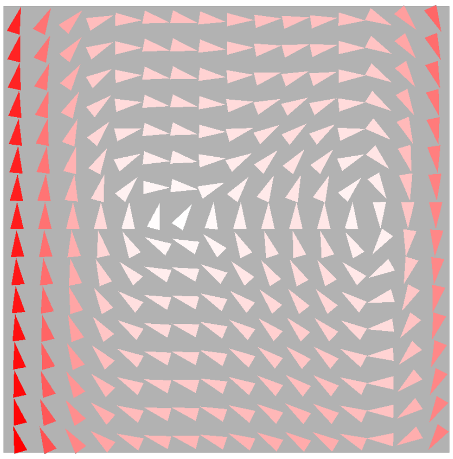

Figure 3: Arrows (triangles) visualizing the vector direction and vector magnitude at t=1.0

1.3 Line Integral Convolution (LIC) (3 pts)

Use the supplied noise texture and the vector field calculated in the previous task to generate a LIC visualization with an average-convolution. To calculate the average-convolution, accumulate the texture values by walking some steps forward in vector direction along the streamline and some steps backward. Use a proper step size utilizing the dimension of the texture. To achieve the same results as in Figure 4 you should create a 128x128 texture for you vector field and make 20 steps forwards and backwards.

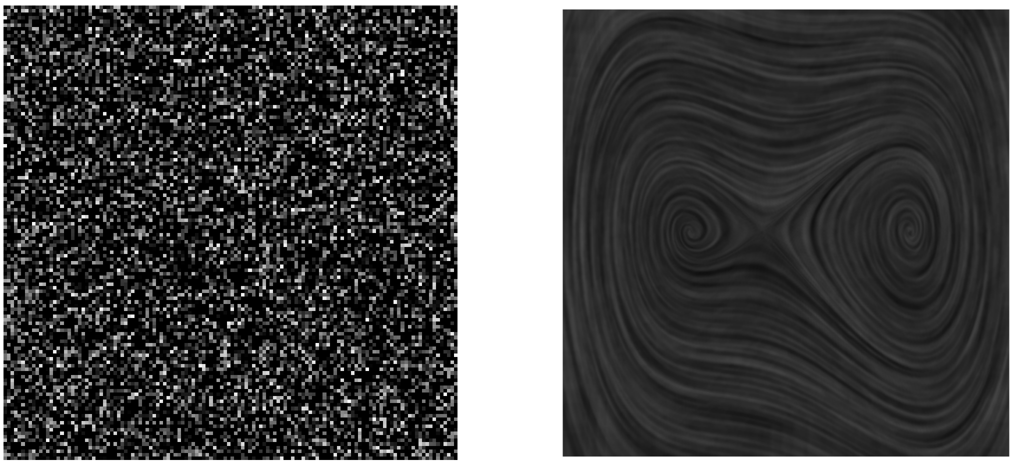

Figure 4: A 128x128 noise texture and the line integral convolution at t=1.0

1.4 BONUS: Animate the LIC (1 pt)

In the previous task you used LIC to visualize a single temporal slice. Animate the LIC by varying the temporal component of the `sampleVectorField()` function. You should loop the animation (i.e., start from the beginning once the maximum time is reached).

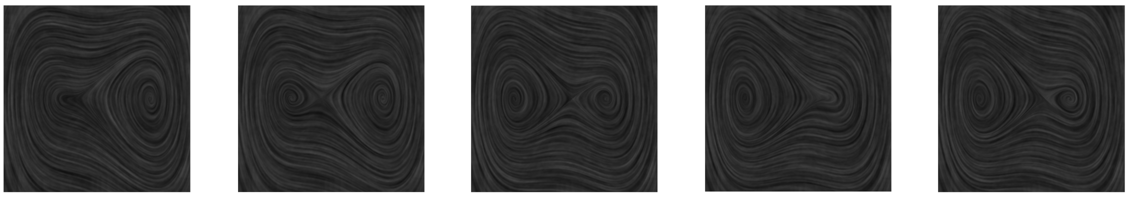

Figure 5: The line integral convolution at t=0.0 to t=4.0 over t= 1.0, 2.0, 3.0.

1.5 BONUS: Implement the "Double Gyre" (1 pt)

Implement the *Double Gyre* from `https://cgl.ethz.ch/research/visualization/data.php` where you also can find the *Forced-Damped Duffing Oscillator*, which was used for the LIC.

Handing in

For the Hand-in, write the names of both group members at the top of all code files and PDF documents. Create a ZIP archive of your project folder and **PLEASE EXCLUDE** the folder **NODE_MODULES** as well as the **main.js** file from your archive. Name it with the last and first name of each group member, and the assignment number (e.g., *schaefer_marco___bok_marcel_assignment10.zip*) and upload it to Assignment_10.