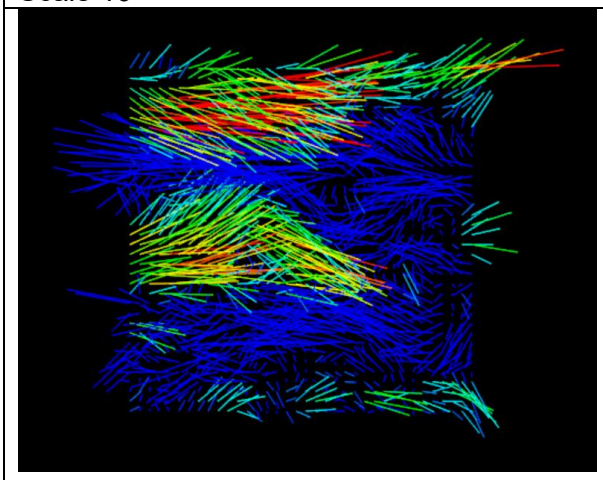
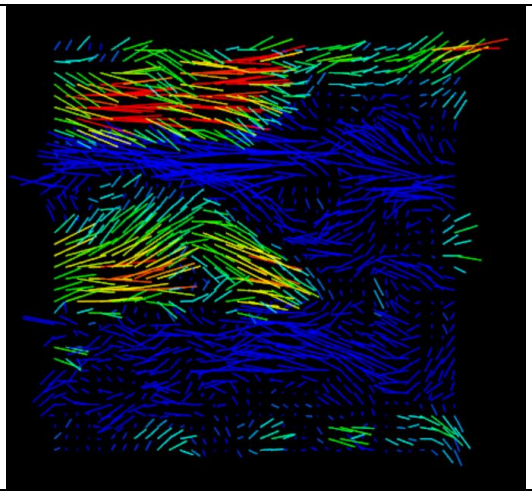
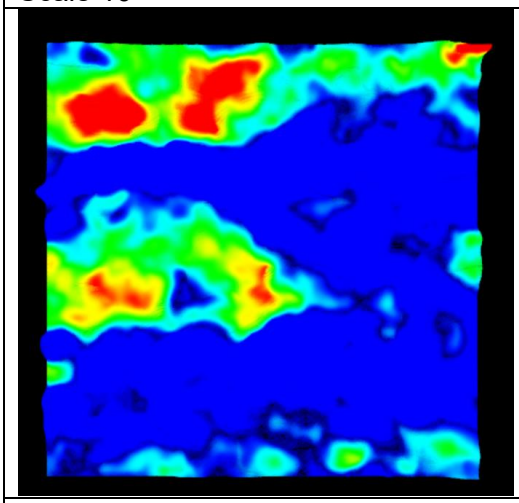
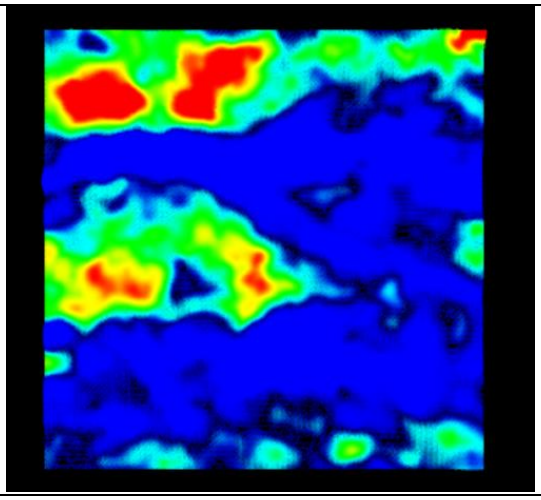
Flow Visualization Result

1. **HedgeHog Approach**

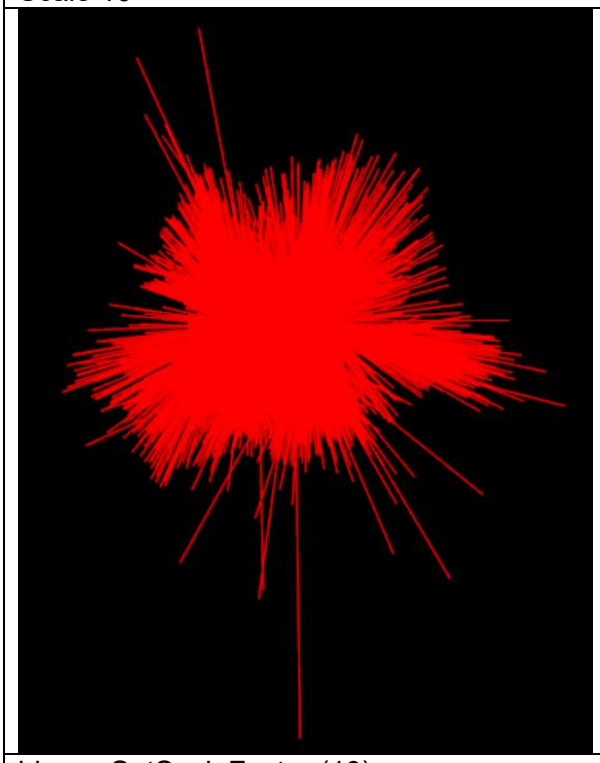
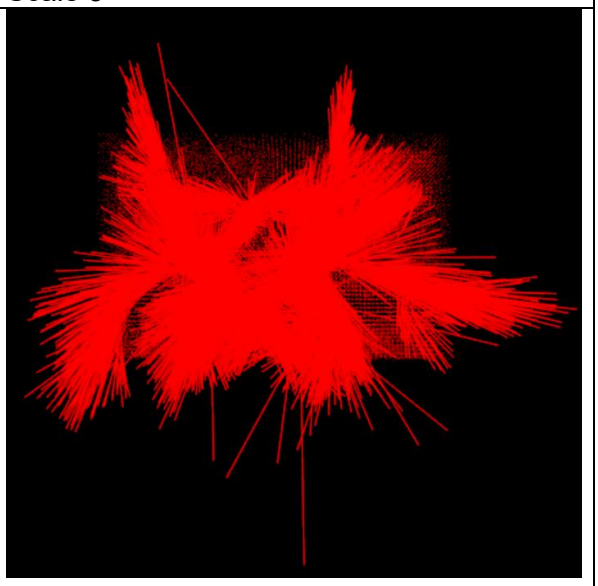
- testData1.vtk

Scale 10	Scale 5
	
hhog->SetScaleFactor(10);	hhog->SetScaleFactor(5);

- **testData2.vtk**

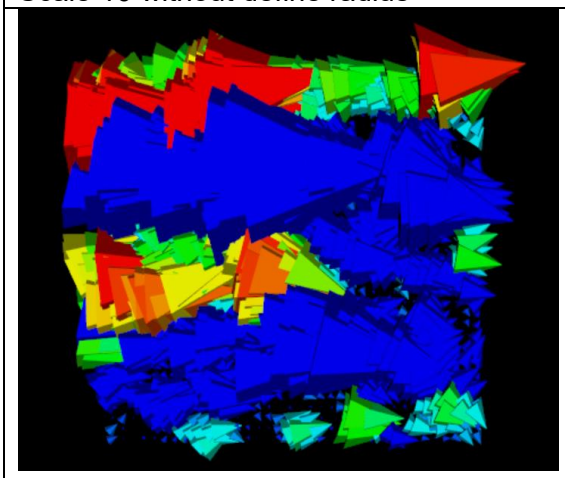
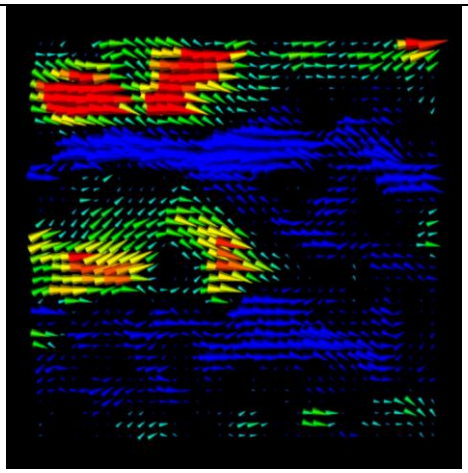
Scale 10	Scale 3
	
hhog->SetScaleFactor(10);	hhog->SetScaleFactor(3);

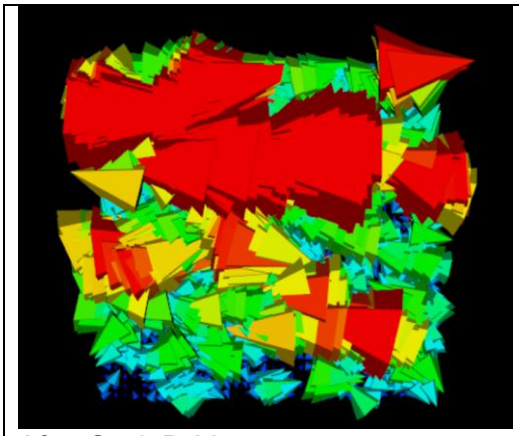
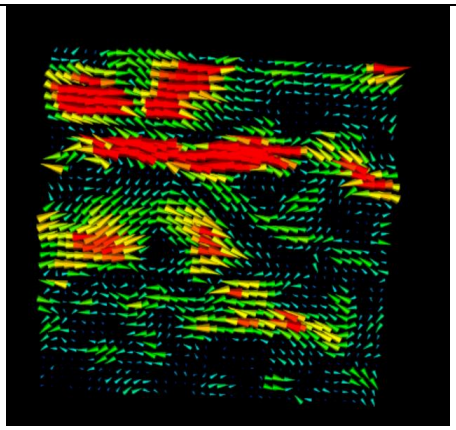
- **carotid.vtk**

Scale 10	Scale 3
	
hhog->SetScaleFactor (10);	hhog->SetScaleFactor (3);

2. **Glyph3D Approach**

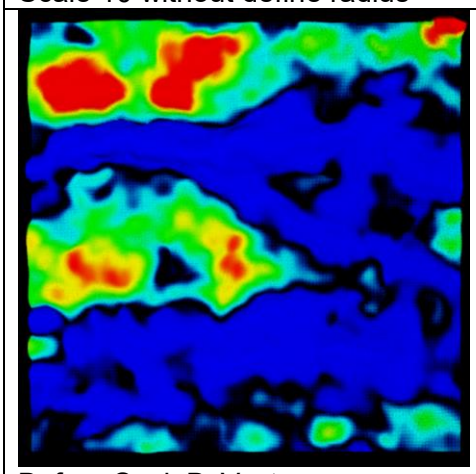
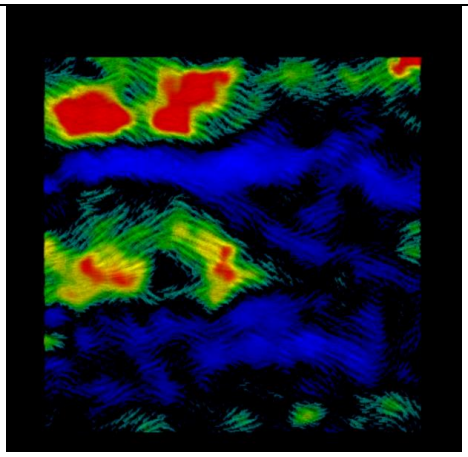
- **testData1.vtk**

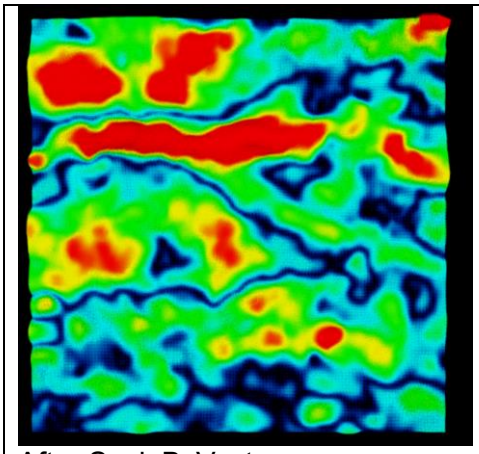
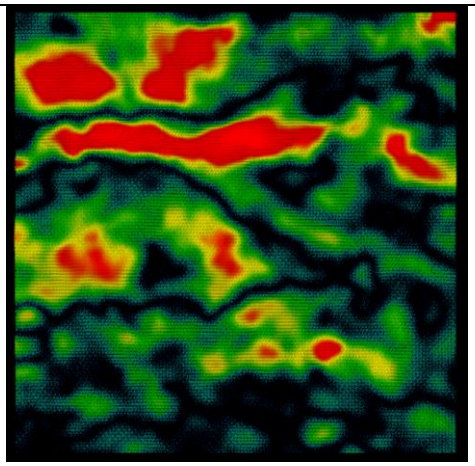
Scale 10 without define radius	Scale 5 with radius of 0.2
	
Before ScaleByVector	Before ScaleByVector

 After ScaleByVector	 After ScaleByVector

vtkGlyph3D* glyh = vtkGlyph3D::New();
glyh->SetInputConnection(reader->GetOutputPort());
vtkConeSource *cone = vtkConeSource::New();
glyh->SetSourceConnection(cone->GetOutputPort());
glyh->SetScaleModeToScaleByVector();
glyh->SetScaleFactor(10);
	vtkGlyph3D* glyh = vtkGlyph3D::New();
	glyh->SetInputConnection(reader->GetOutputPort());
	vtkConeSource *cone = vtkConeSource::New();
	glyh->SetSourceConnection(cone->GetOutputPort());
	glyh->SetScaleModeToScaleByVector();
	glyh->SetScaleFactor(5);

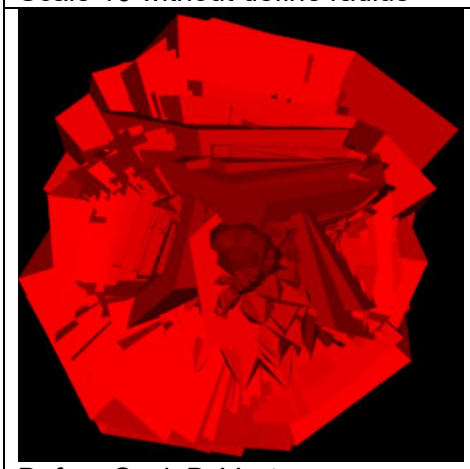
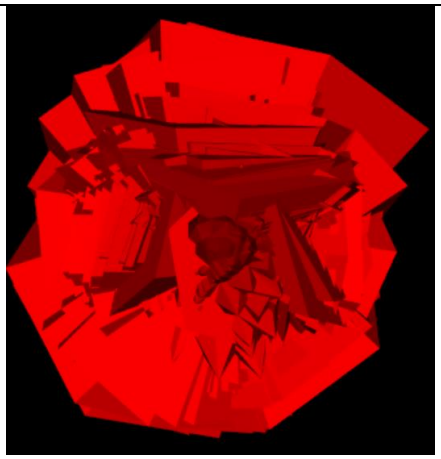
- **testData2.vtk**

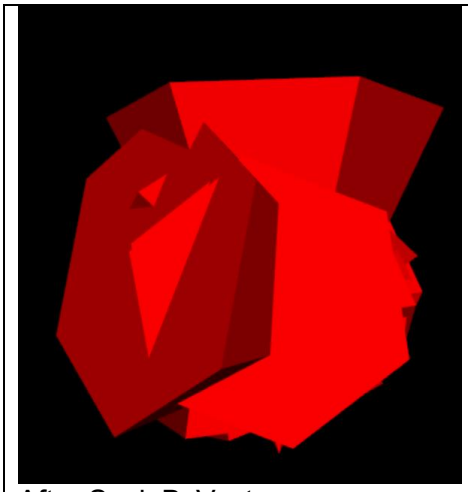
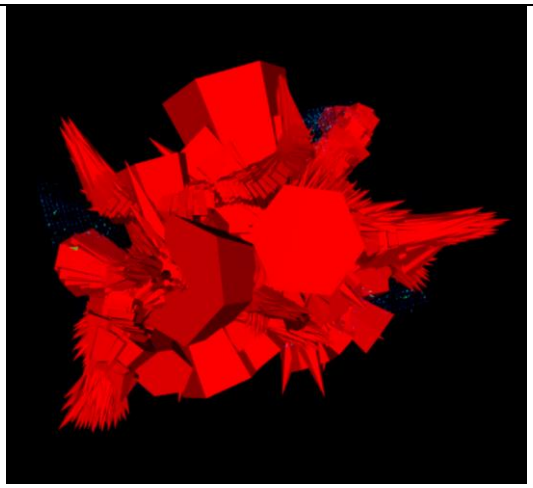
Scale 10 without define radius	Scale 3 with radius = 0.2
 Before ScaleByVector	 Before ScaleByVector

 After ScaleByVector	 After ScaleByVector

vtkGlyph3D* glyh = vtkGlyph3D::New();
glyh->SetInputConnection(reader->GetOutputPort());
vtkConeSource *cone = vtkConeSource::New();
glyh->SetSourceConnection(cone->GetOutputPort());
glyh->SetScaleModeToScaleByVector();
glyh->SetScaleFactor(10);
	vtkGlyph3D* glyh = vtkGlyph3D::New();
	glyh->SetInputConnection(reader->GetOutputPort());
	vtkConeSource *cone = vtkConeSource::New();
	glyh->SetSourceConnection(cone->GetOutputPort());
	glyh->SetScaleModeToScaleByVector();
	glyh->SetScaleFactor(3);

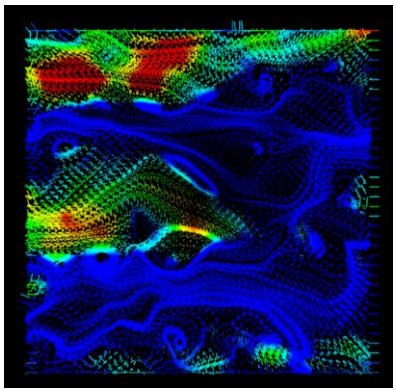
- **carotid.vtk**

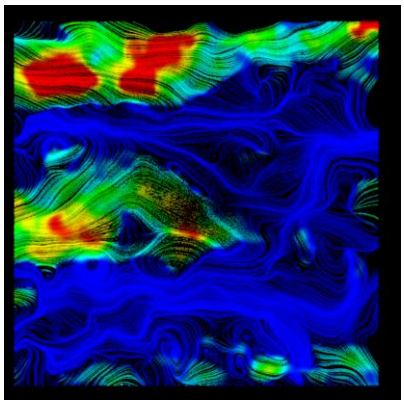
Scale 10 without define radius	Scale 3 with radius = 0.2
 Before ScaleByVector	 Before ScaleByVector

 After ScaleByVector	 After ScaleByVector

vtkGlyph3D* glyh = vtkGlyph3D::New();
glyh->SetInputConnection(reader->GetOutputPort());
vtkConeSource *cone = vtkConeSource::New();
glyh->SetSourceConnection(cone->GetOutputPort());
glyh->SetScaleModeToScaleByVector();
glyh->SetScaleFactor(10);
	vtkGlyph3D* glyh = vtkGlyph3D::New();
	glyh->SetInputConnection(reader->GetOutputPort());
	vtkConeSource *cone = vtkConeSource::New();
	glyh->SetSourceConnection(cone->GetOutputPort());
	glyh->SetScaleModeToScaleByVector();
	glyh->SetScaleFactor(3);

3. Streamline applied to HedgeHog Approach

TESTDATA1.VTK
 
	int xSize = 36;
	// Number of points in x-direction
	int ySize = 36;
	// Number of points in y-direction
	int zSize = 2;
	// Number of points in z-direction
	int maxX = 36;
	int minX = 0;
	int maxY = 36;
	int minY = 0;
	int maxZ = 0;
	int minZ = 0;
	streamLine->SetStartPosition(10.1, 10.1, 2.5);

TESTDATA2.VTK
 
	int xSize = 50;
	// Number of points in x-direction
	int ySize = 50;
	// Number of points in y-direction
	int zSize = 2;
	// Number of points in z-direction
	int maxX = 357;
	int minX = 0;
	int maxY = 357;
	int minY = 0;
	int maxZ = 0;
	int minZ = 0;
	streamLine->SetStartPosition(10.1, 10.1, 2.5);

CAROTID.VTK
 
	int xSize = 46;
	// Number of points in x-direction
	int ySize = 29;
	// Number of points in y-direction
	int zSize = 25;
	// Number of points in z-direction
	int maxX = 76;
	int minX = 0;
	int maxY = 49;
	int minY = 0;
	int maxZ = 45;
	int minZ = 0;
	streamLine->SetStartPosition(10.1, 10.1, 2.5);

Prepared by:

Tee Ng Zikang	A20EC0161
Goh Chai Seng	B20EC3004
Abraham Loh Tze Lung	A20EC0001
Zhao Xin	A20EC4053