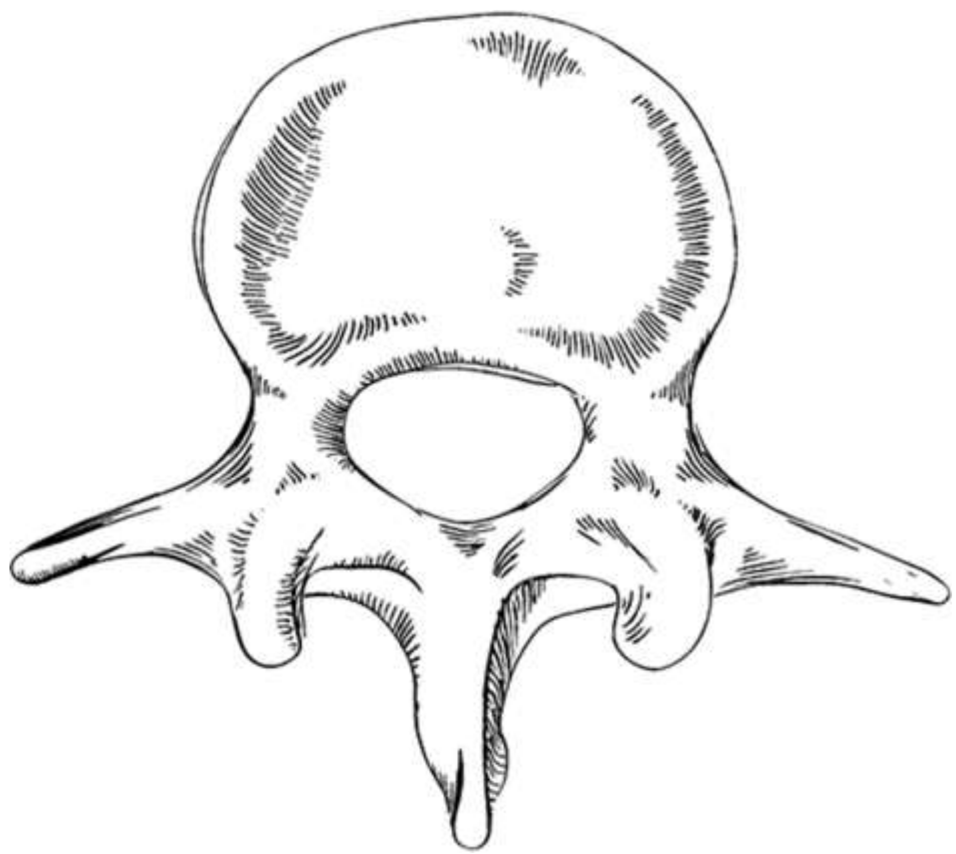
Interactive Example-based Hatching

Moritz Gerl and Tobias Isenberg

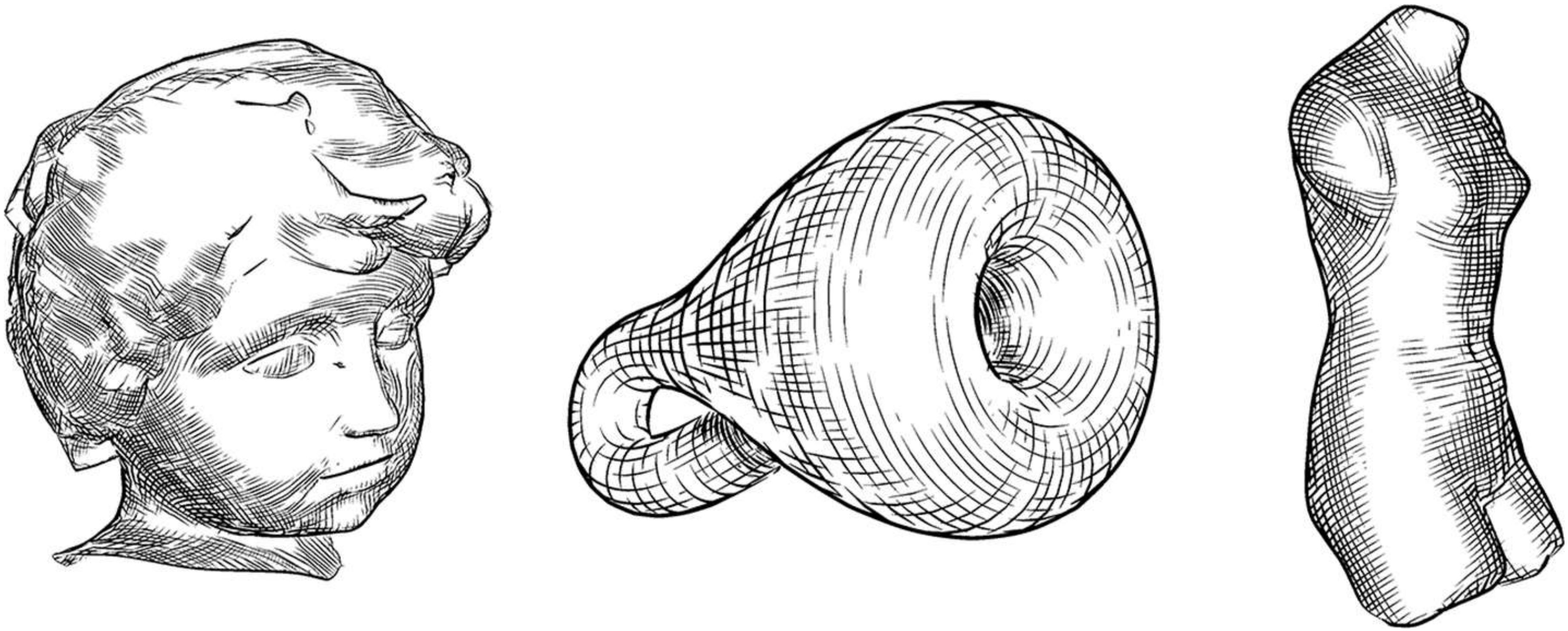


Expressive 2013 Journal Talks

(Computers & Graphics 37(1–2): 65–80, Feb.–Apr. 2013. doi> 10.1016/j.cag.2012.11.003)

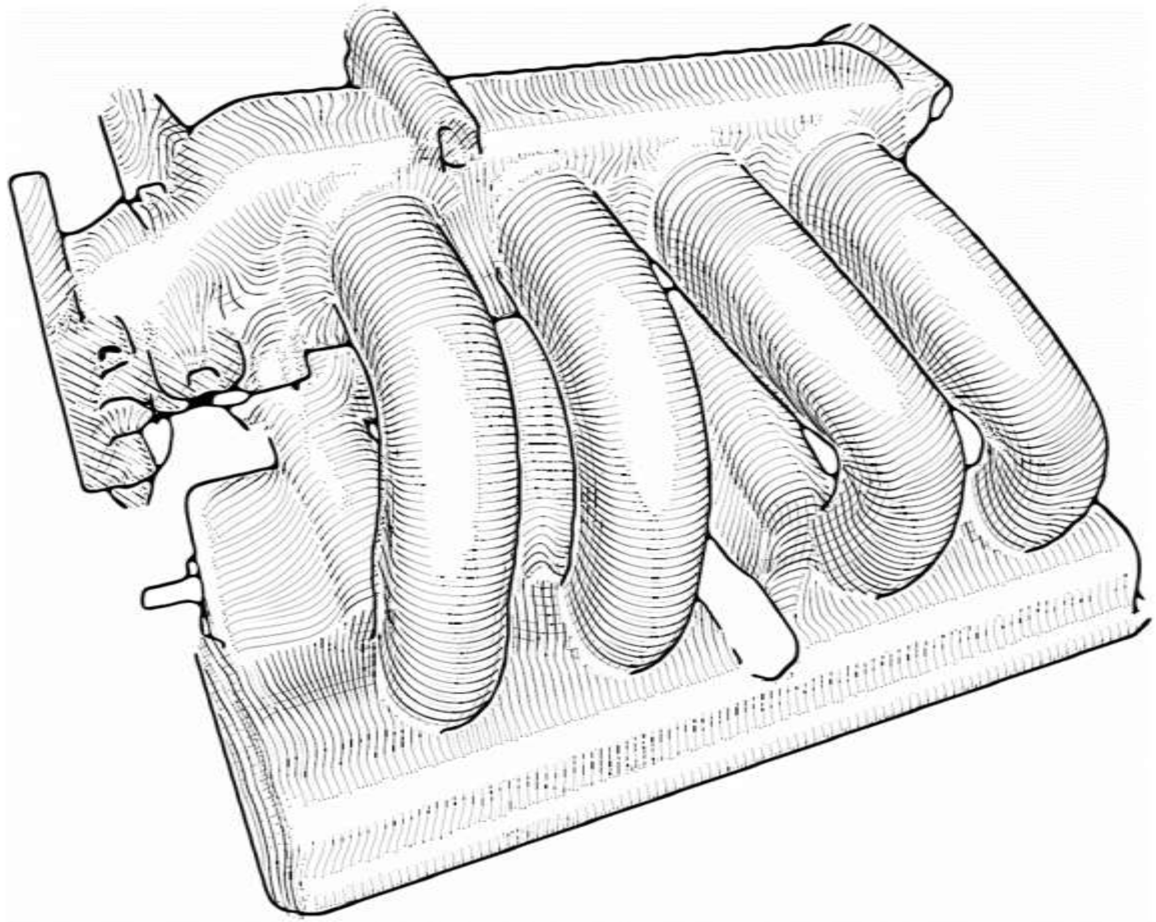
credits: Moritz Gerl

Motivation



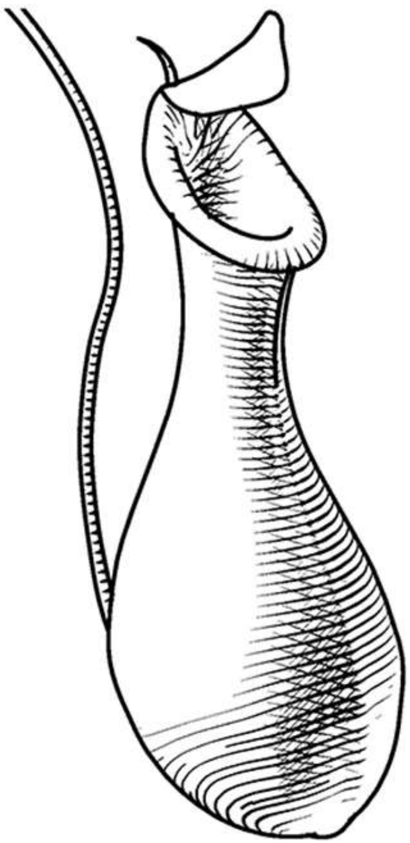
[Hertzmann and Zorin, 2000]

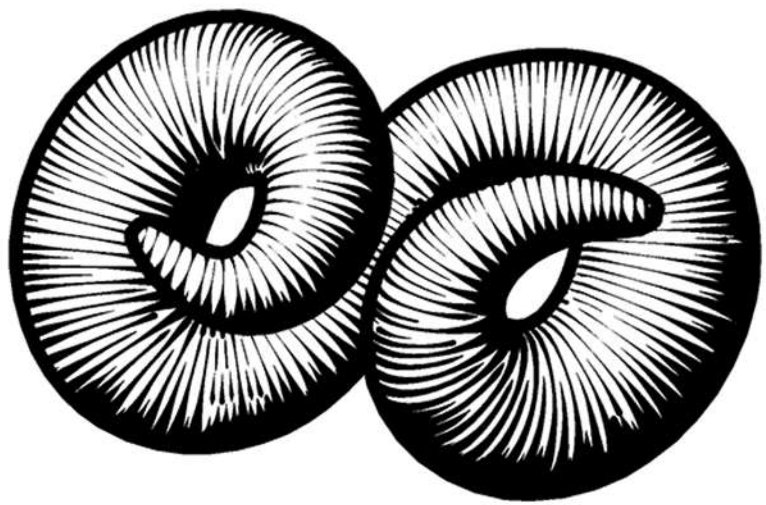
Motivation

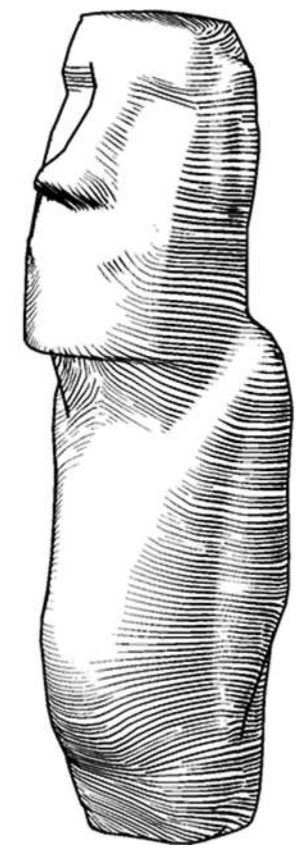


[Roessl and Kobbelt, 2000]

Motivation

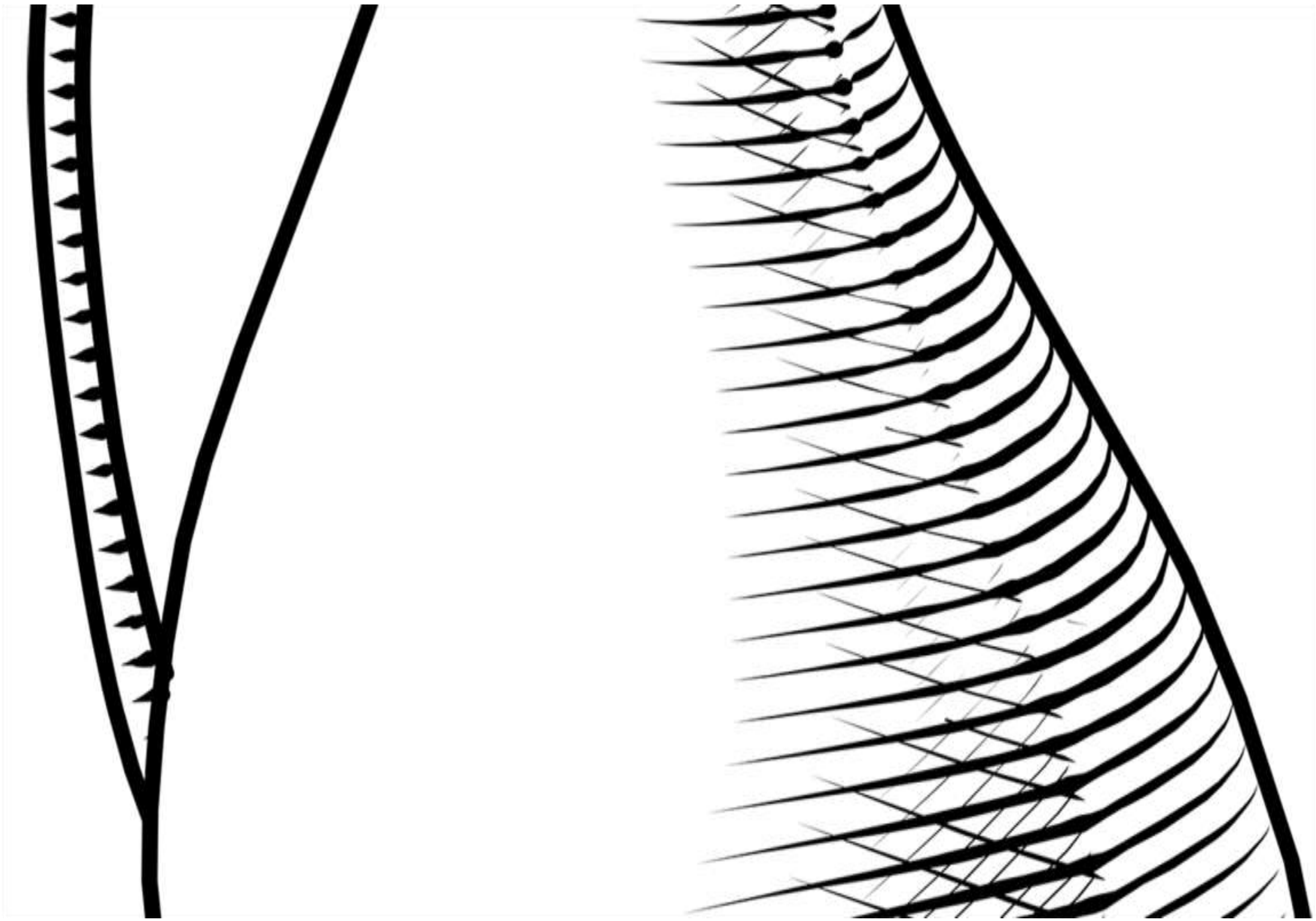






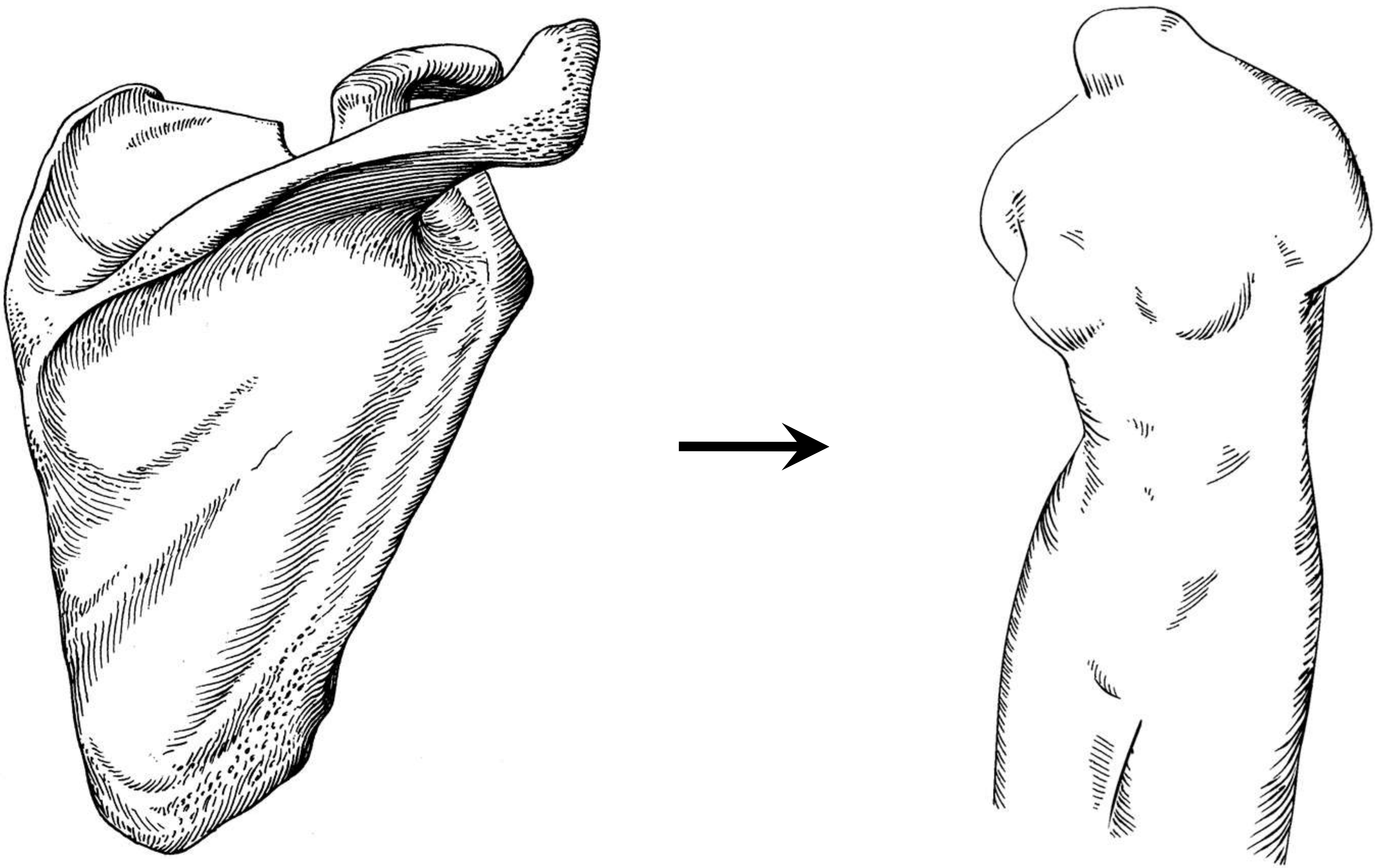
[Zander et al., 2004]

Motivation

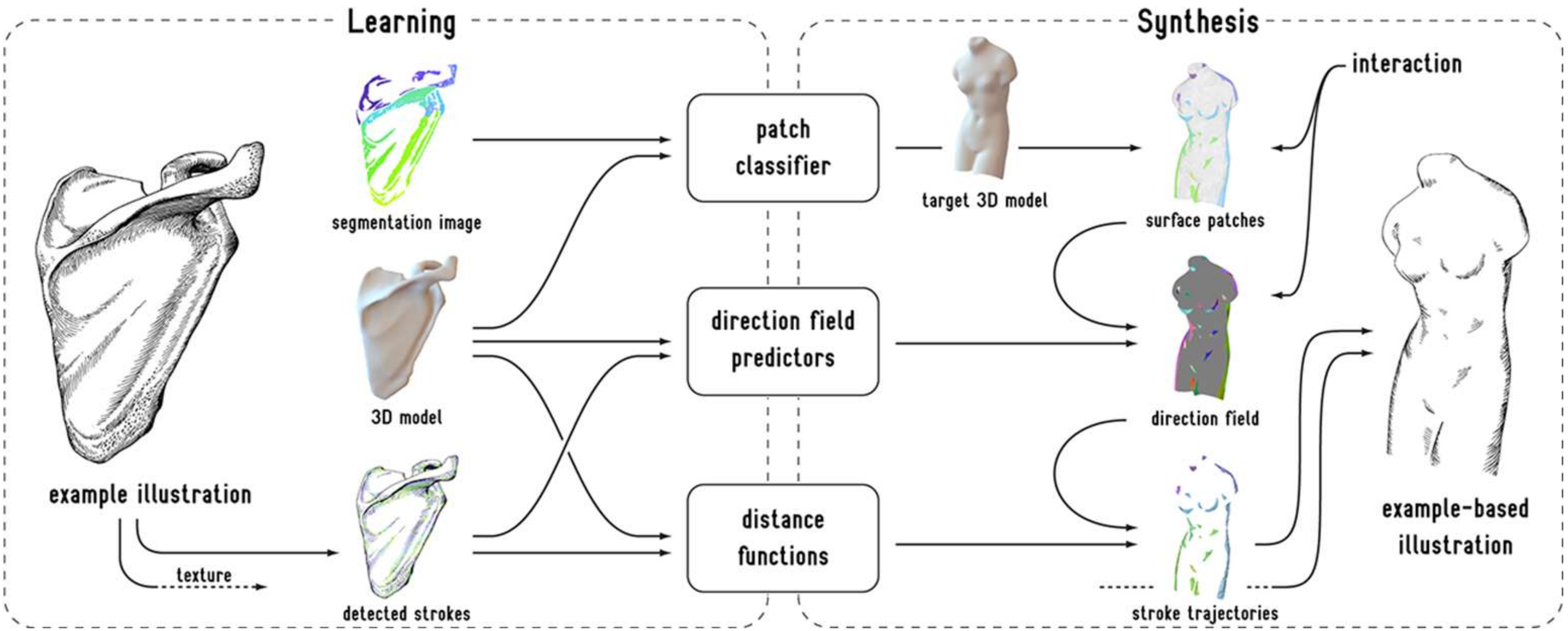


[Zander et al., 2004]

Example-based Hatching

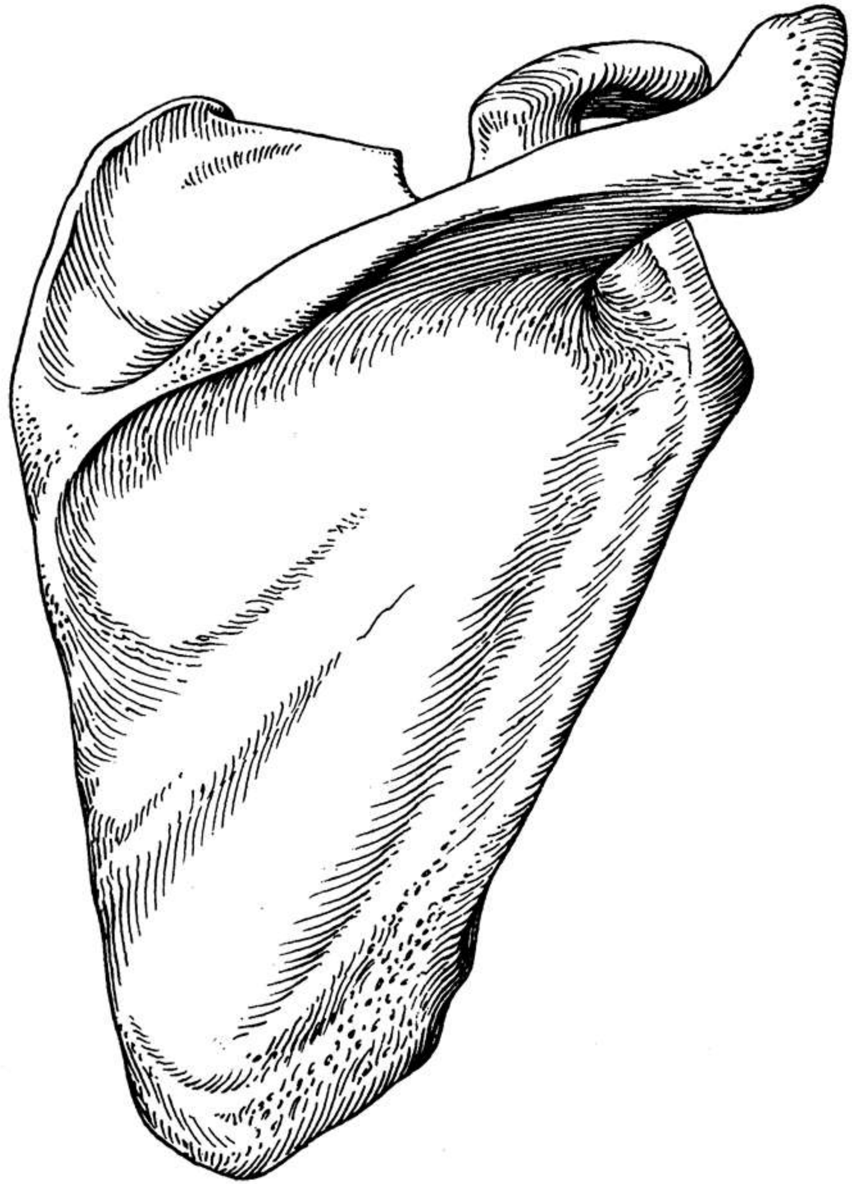


Overview



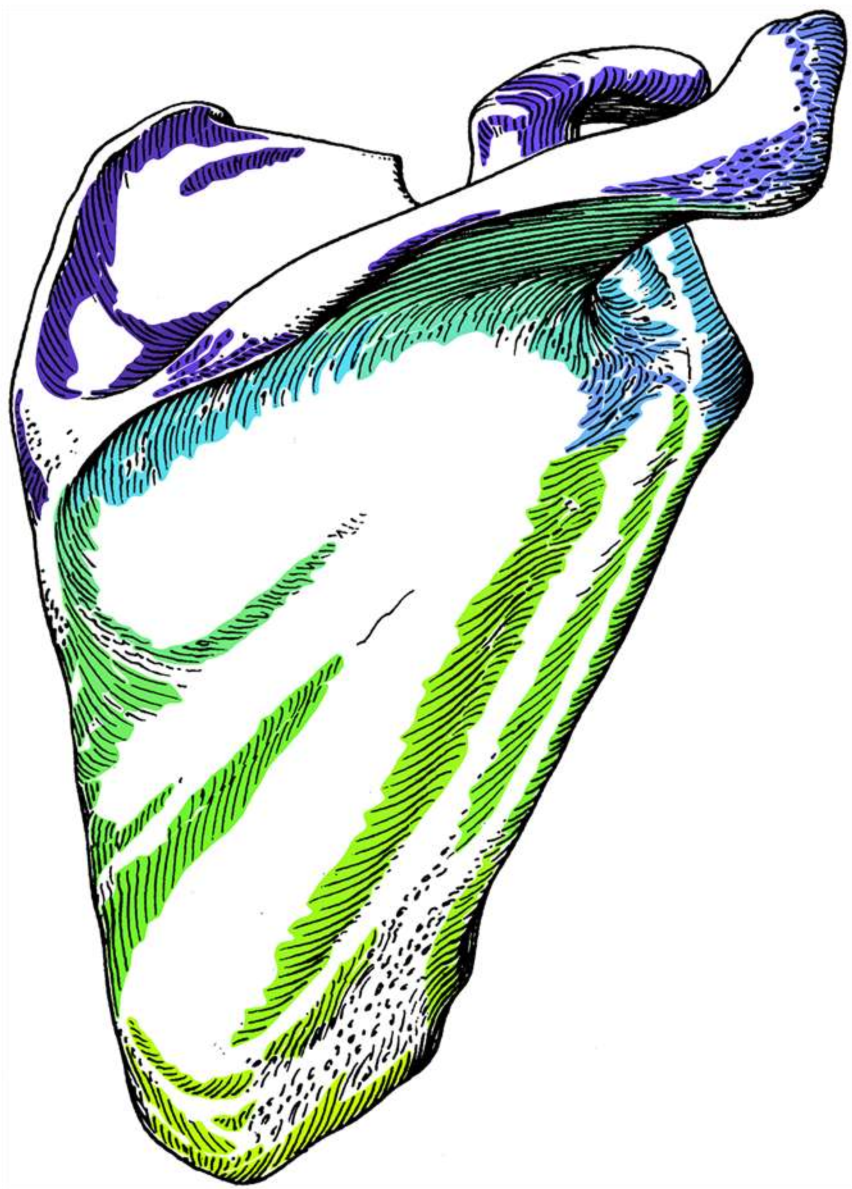
Learning
Synthesis
interaction
patch classifier
target 3D model
segmentation image
surface patches
3D model
direction field predictors
direction field
example illustration
distance functions
example-based illustration
texture
detected strokes
stroke trajectories

Input to Learning



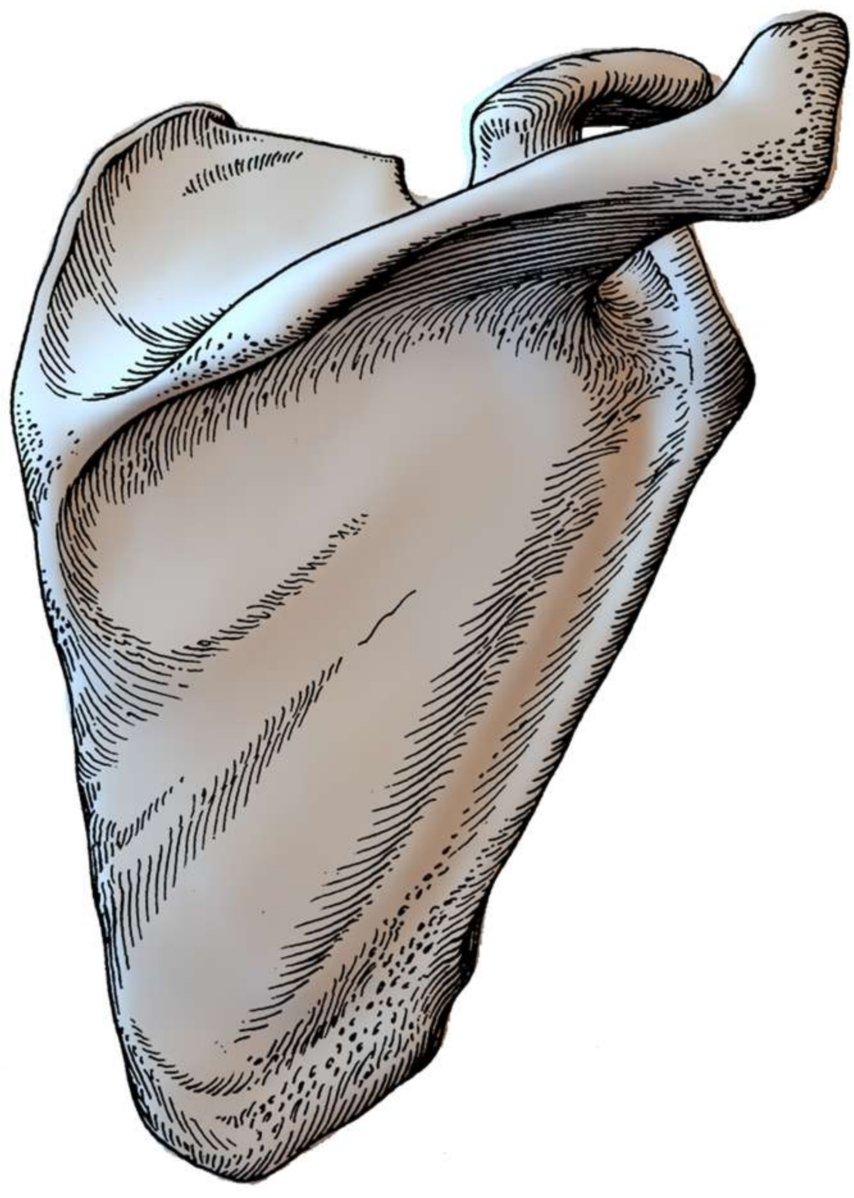
Input to Learning

Input to Learning



Input to Learning

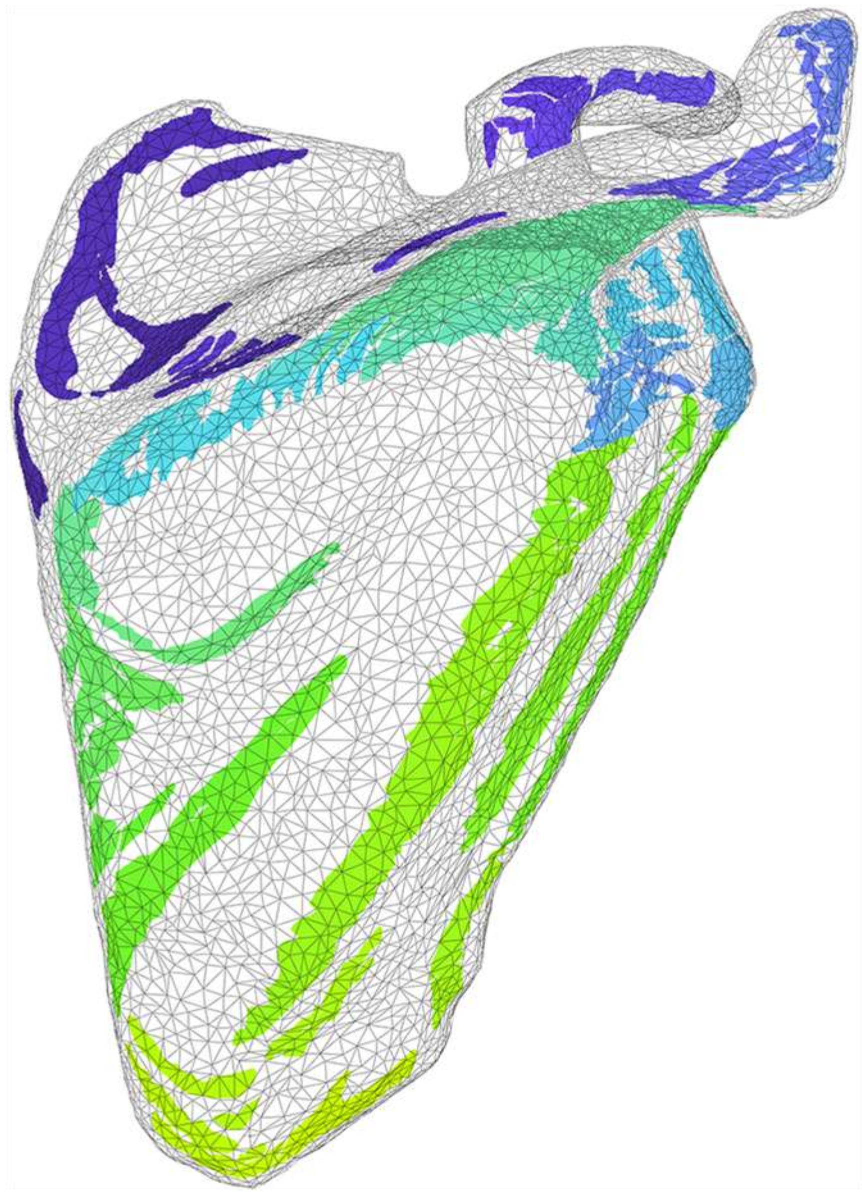
Input to Learning



Learning Regions

Learning Regions

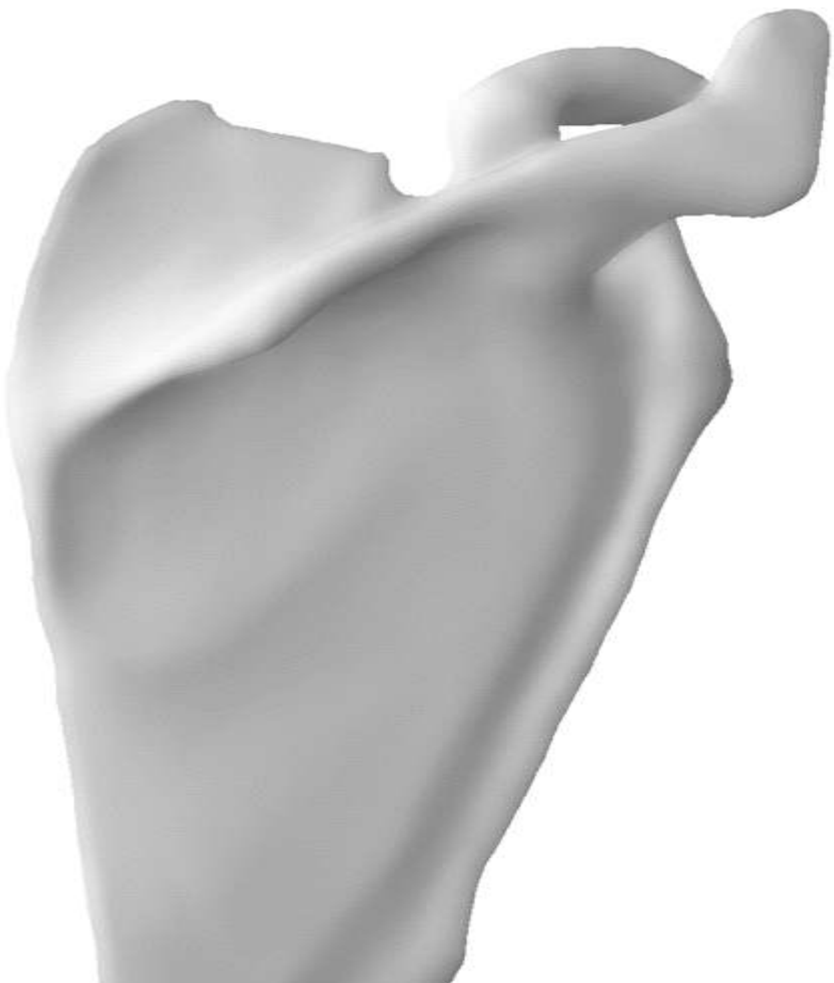
Learning Regions



Example Features (18 in total)

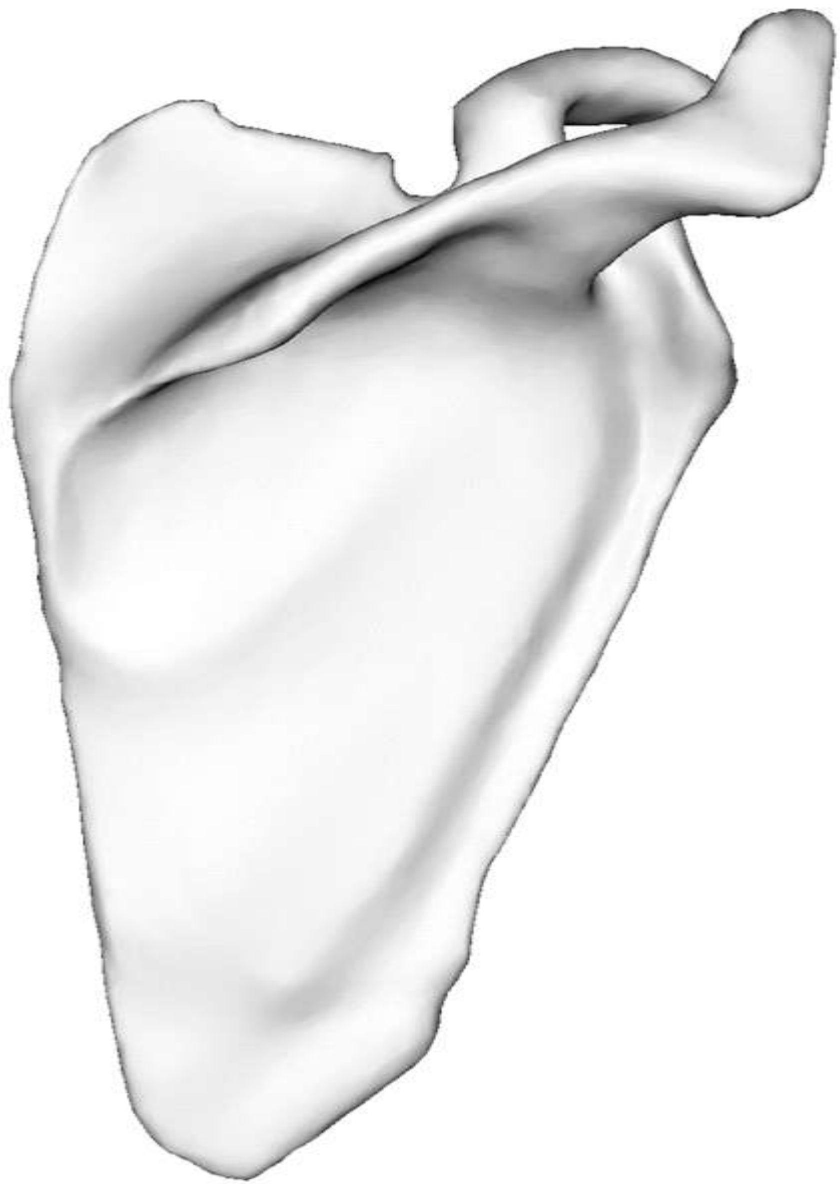
diffuse lighting

Example Features (18 in total)



ambient occlusion

Example Features (18 in total)



facing ratio

Example Features (18 in total)

curvature

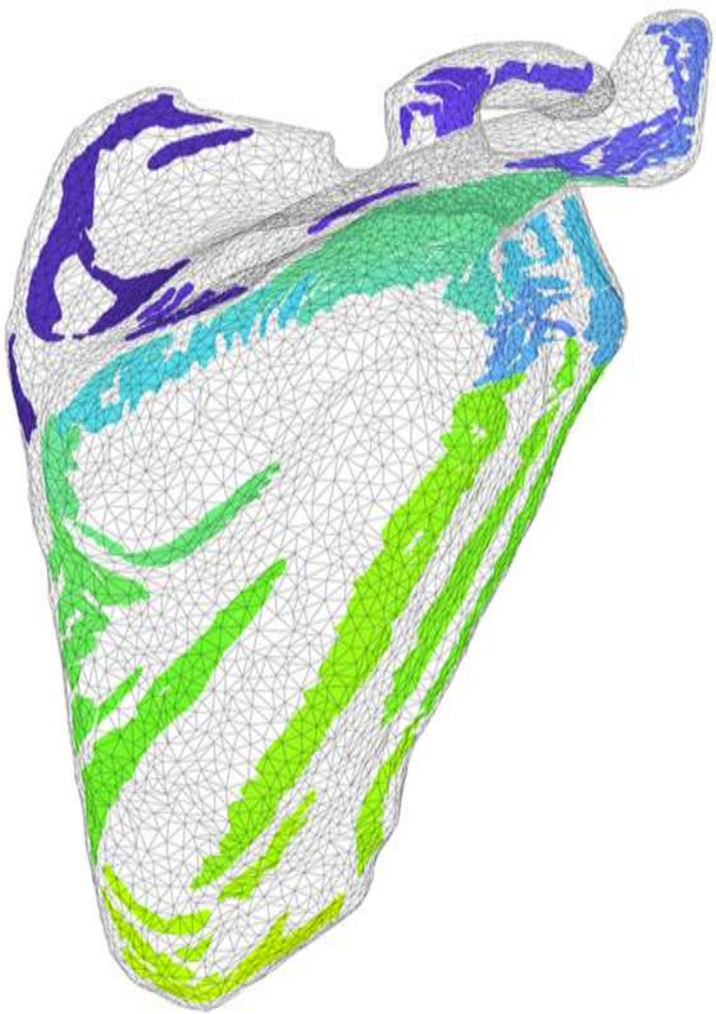
Example Features (18 in total)

normal

Example Features (18 in total)

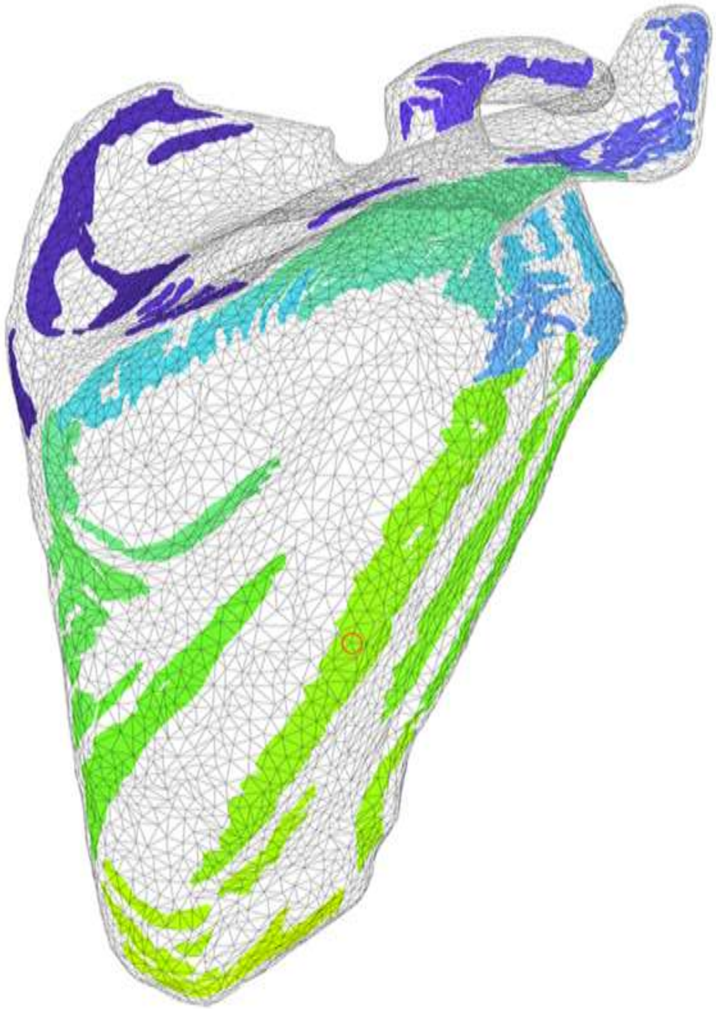
depth

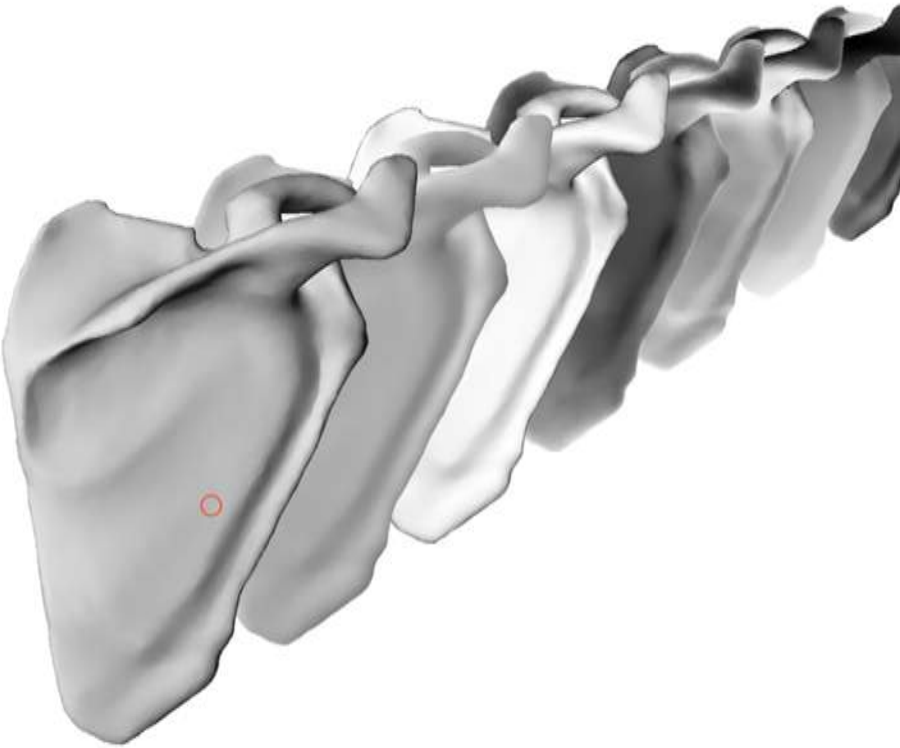
Training the Classifier



(Relevance Vector Machines, one-vs-one strategy for voting and Radial Basis Function Kernels)

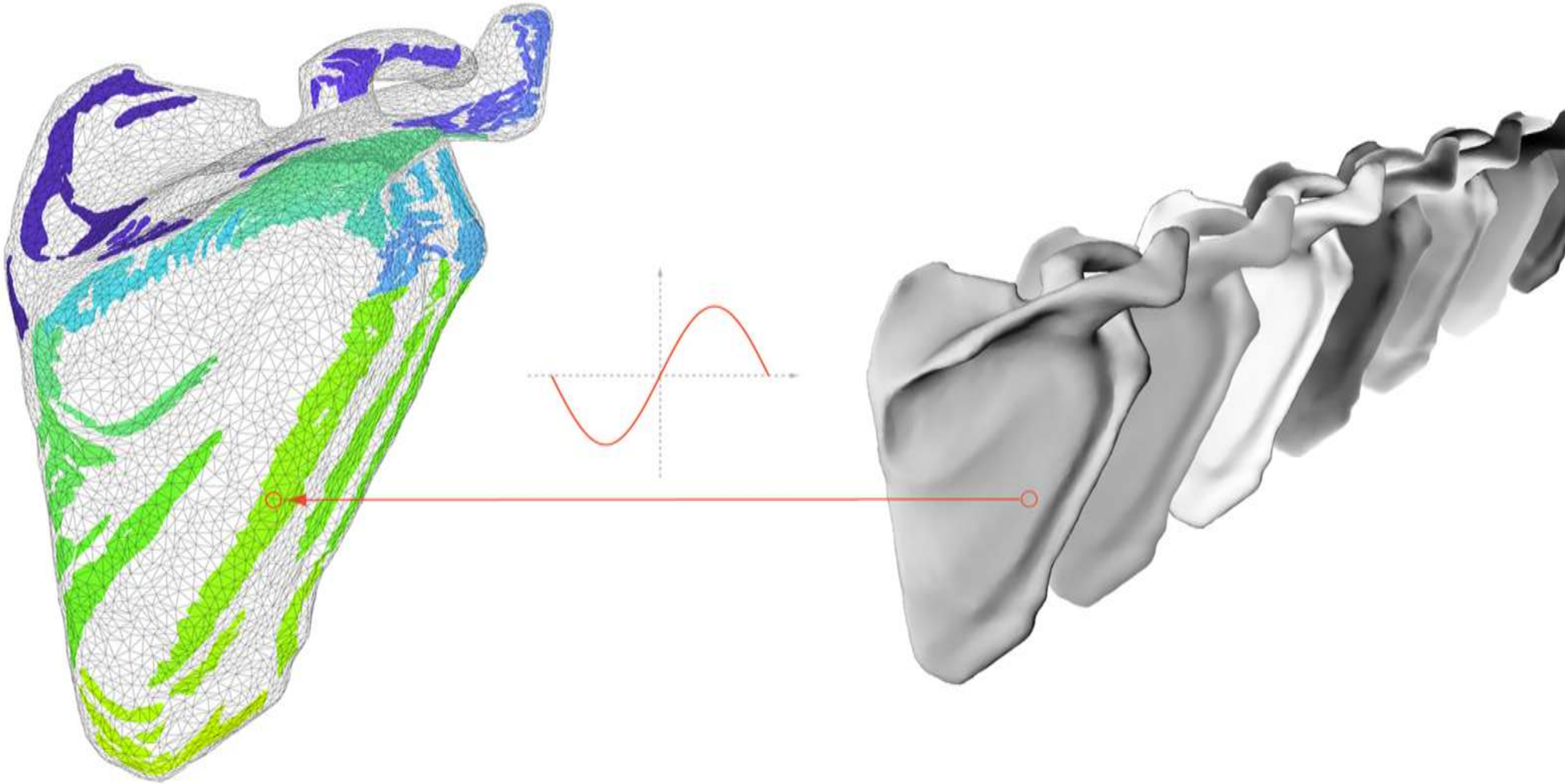
Training the Classifier





(Relevance Vector Machines, one-vs-one strategy for voting and Radial Basis Function Kernels)

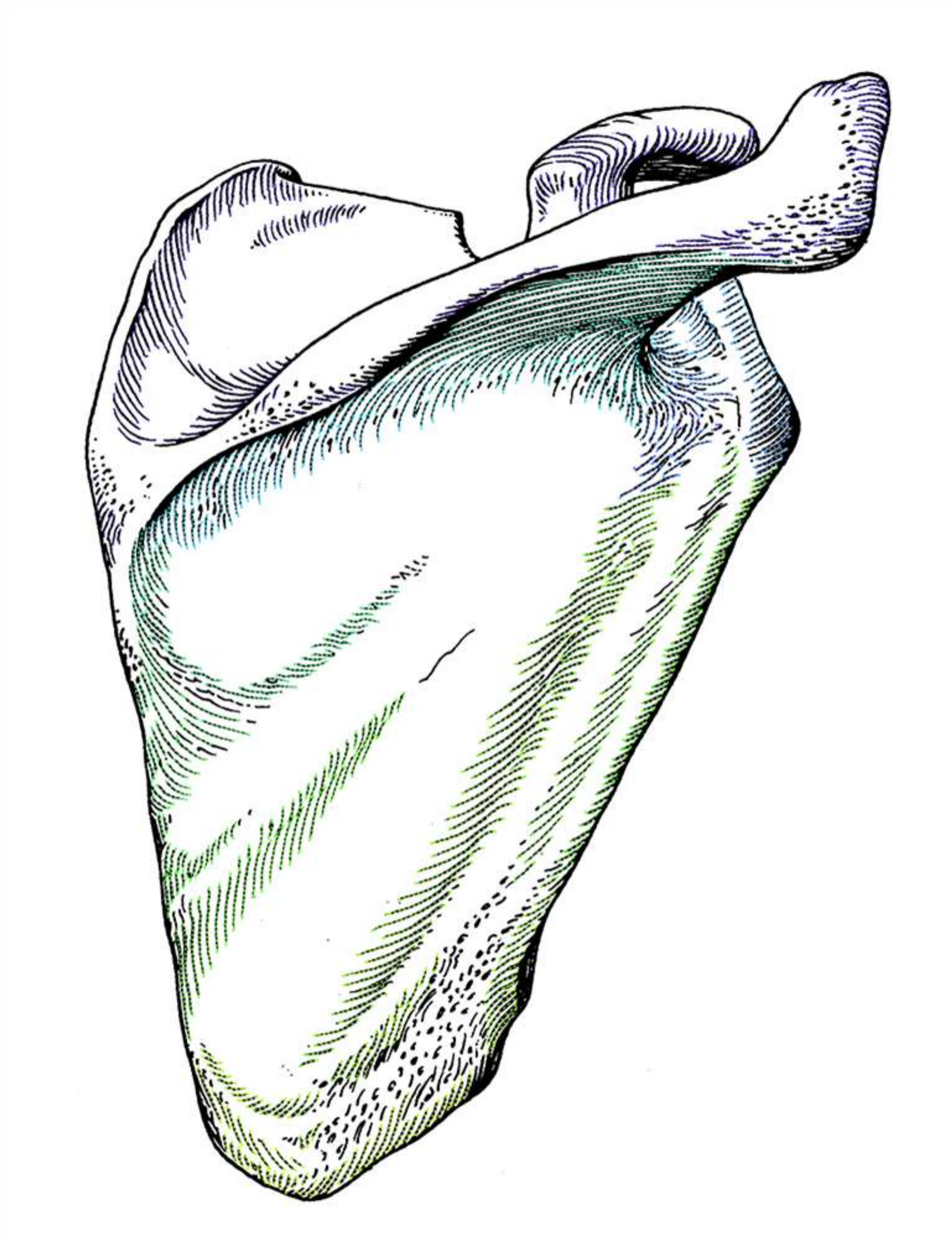
Training the Classifier



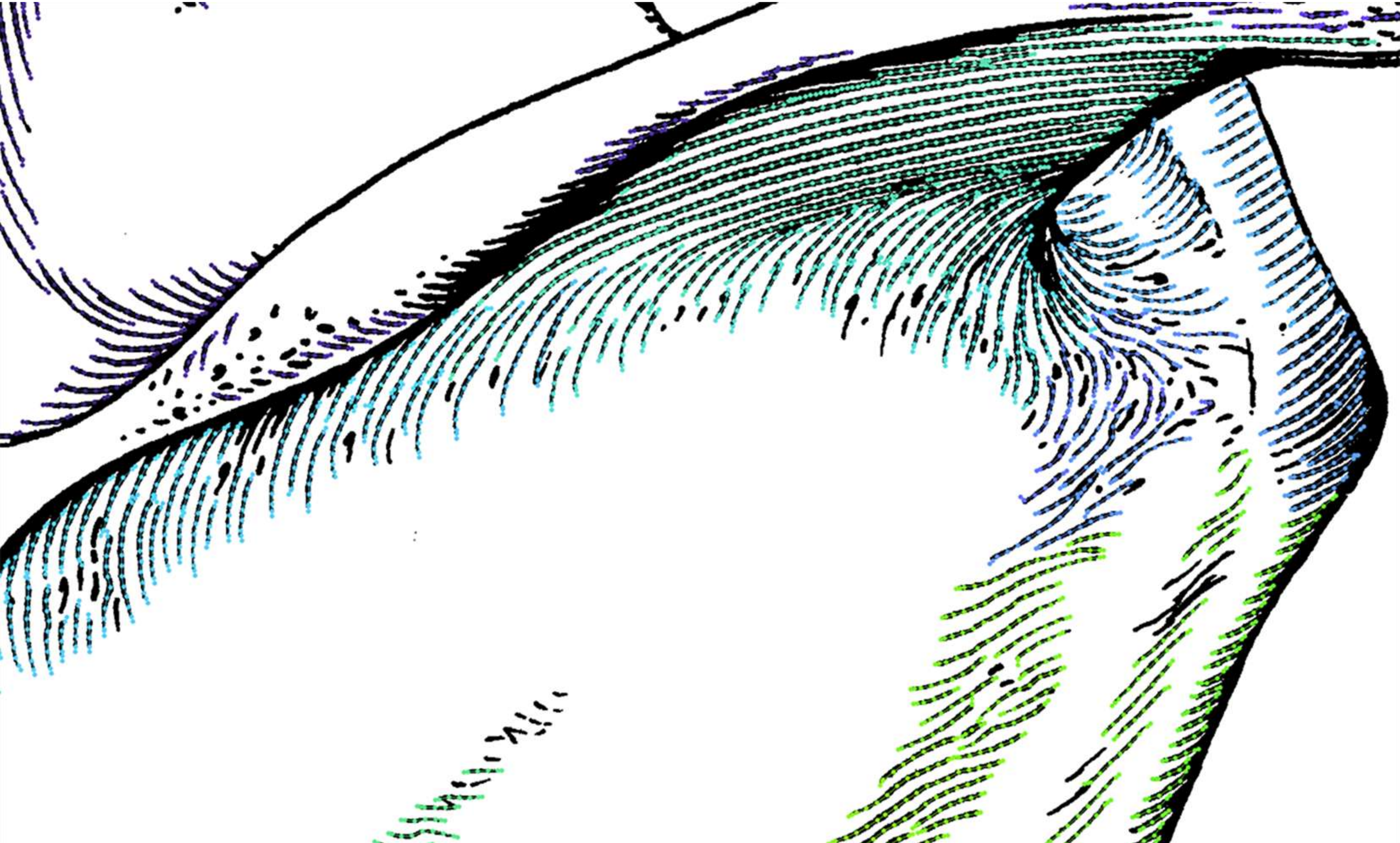
(Relevance Vector Machines, one-vs-one strategy for voting and Radial Basis Function Kernels)

Regions – Synthesis

Learning Directions: Stroke Detection

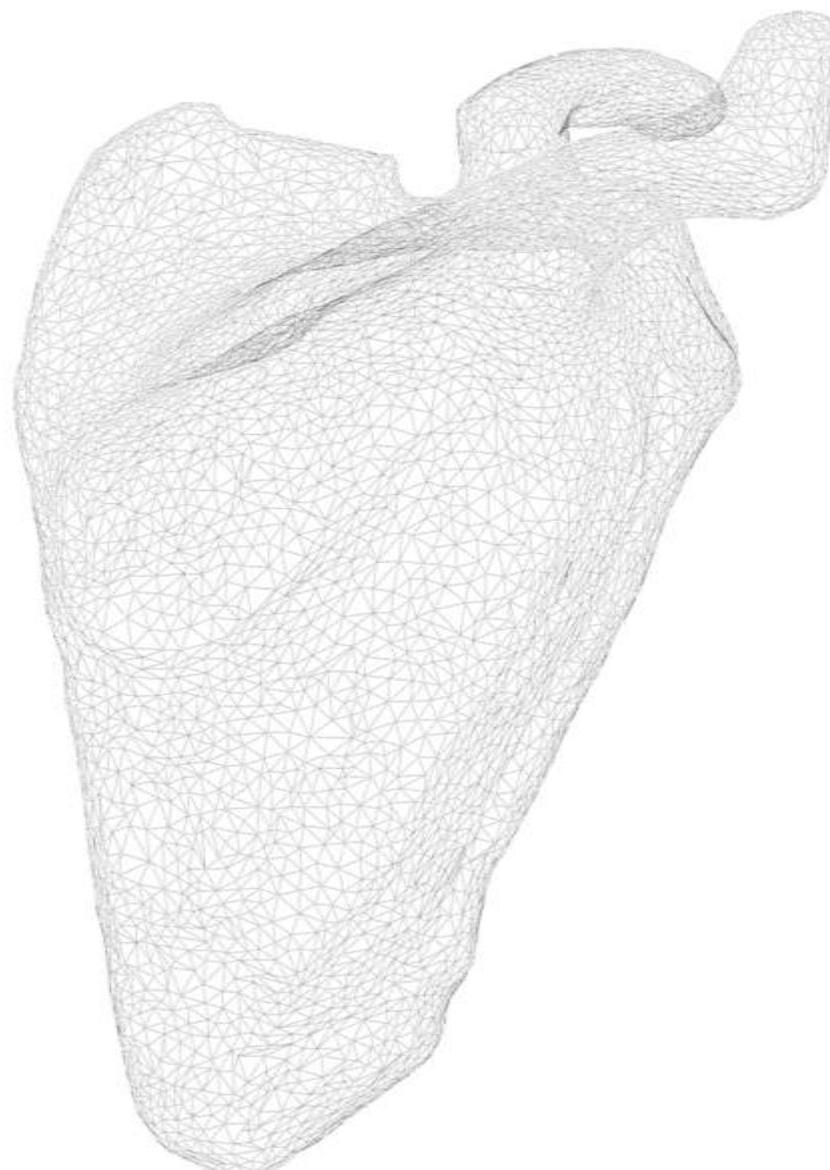


Learning Directions: Stroke Detection



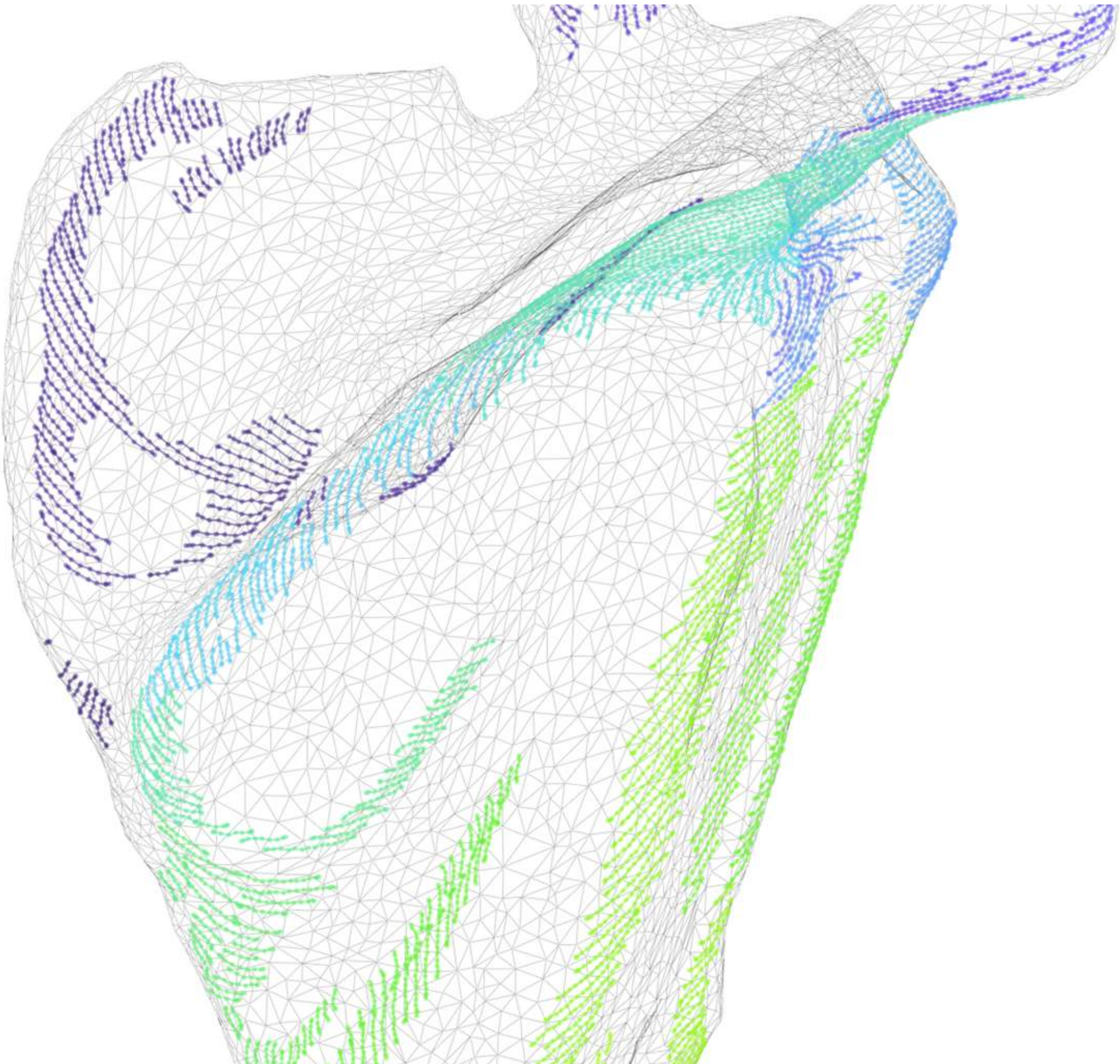
Learning Directions: Stroke Projection

Learning Directions: Stroke Projection

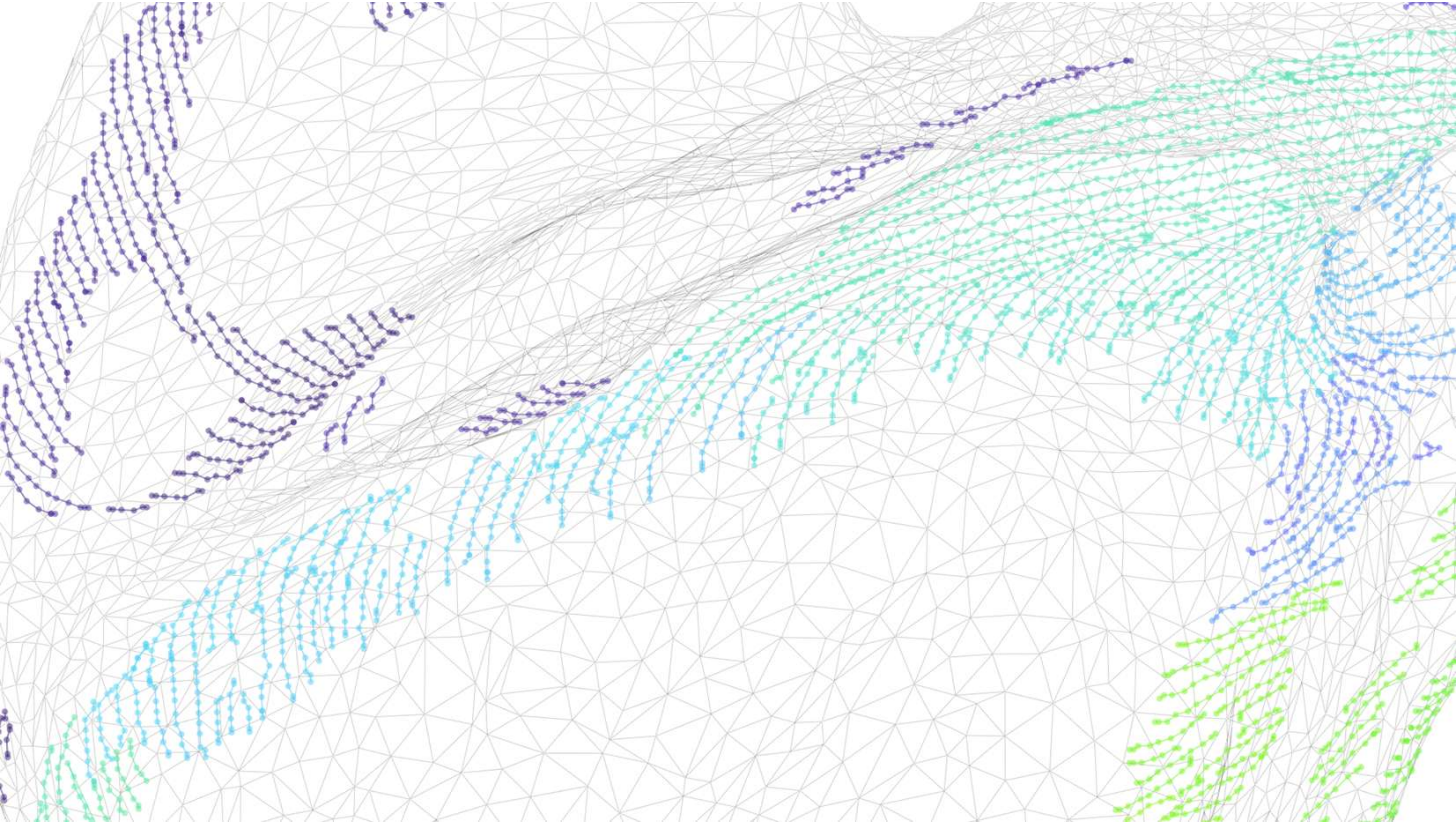


Learning Directions: Stroke Projection

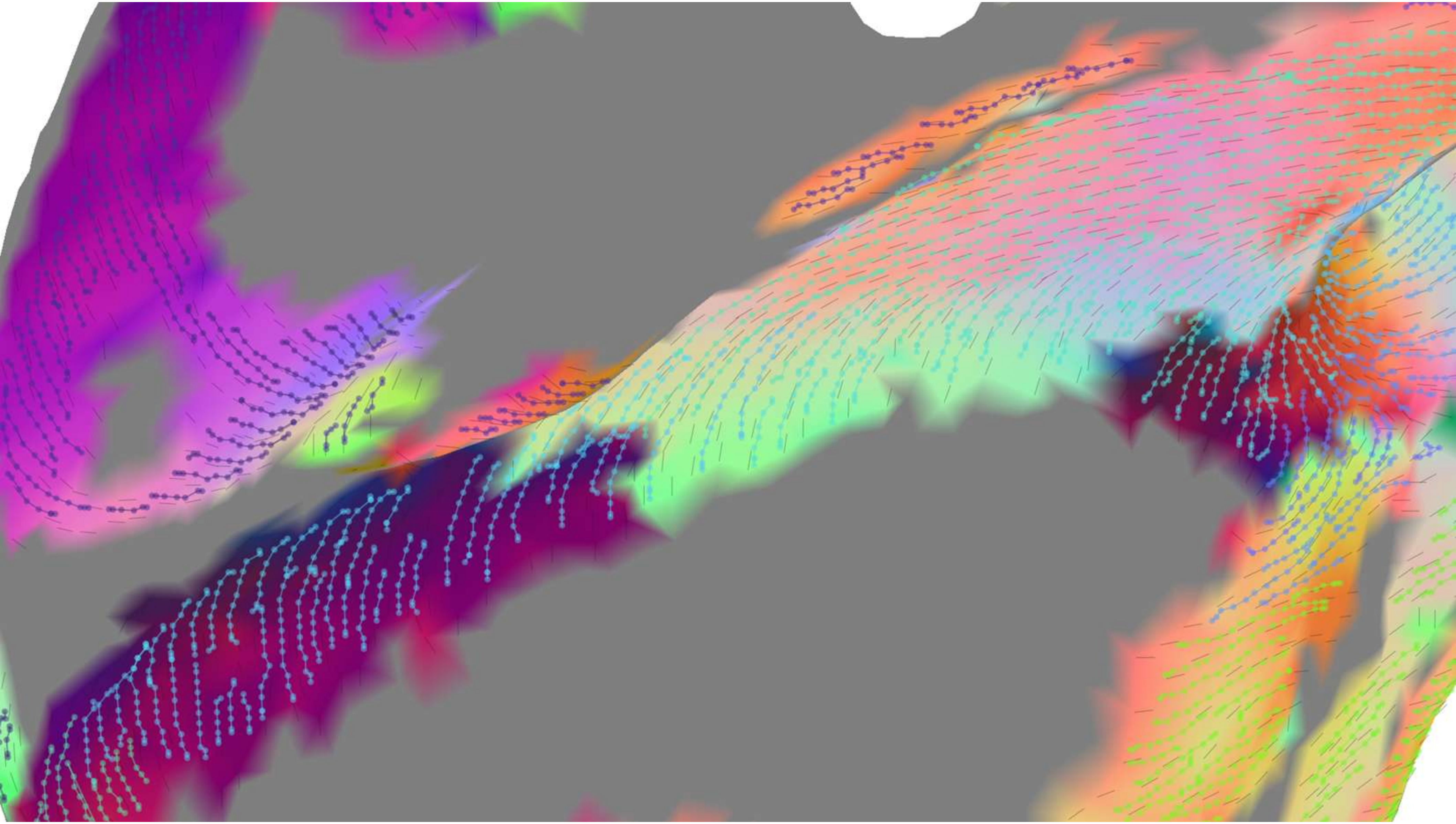
Learning Directions: Stroke Projection



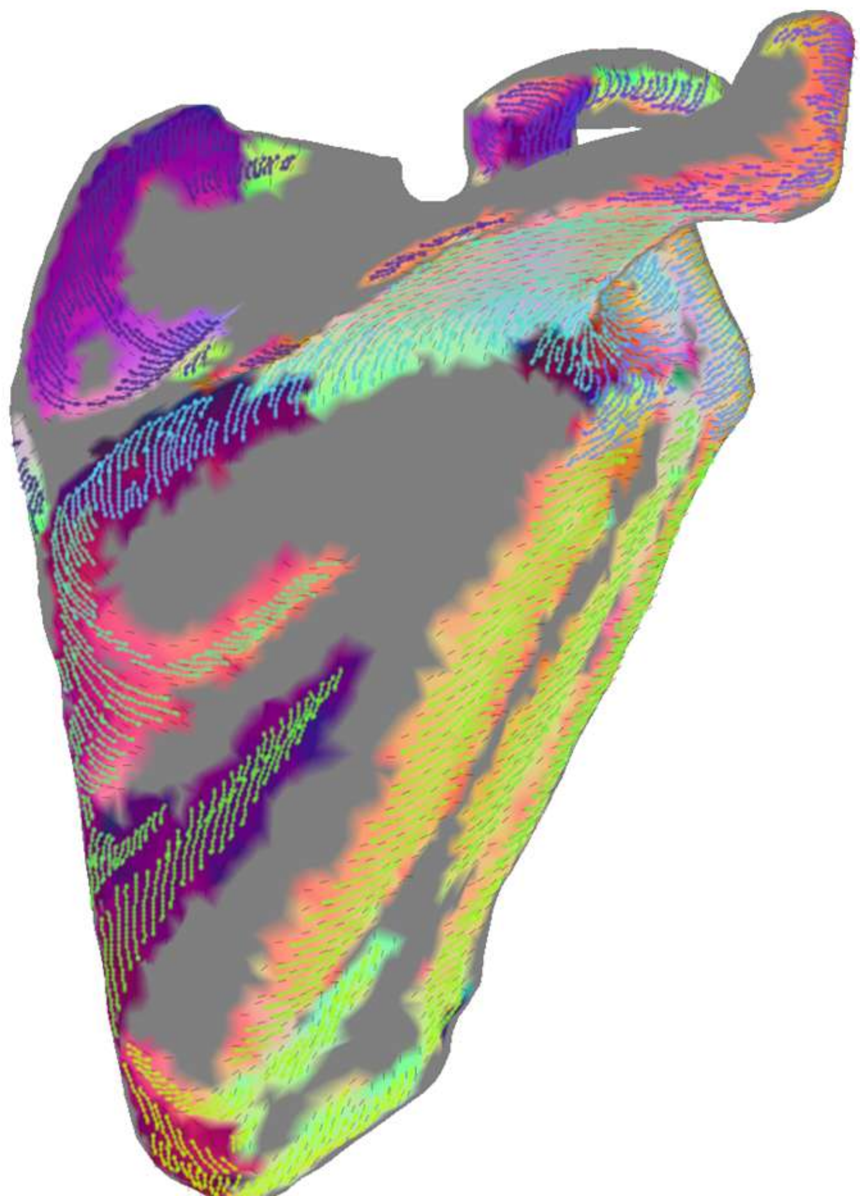
Learning Directions: Stroke Projection



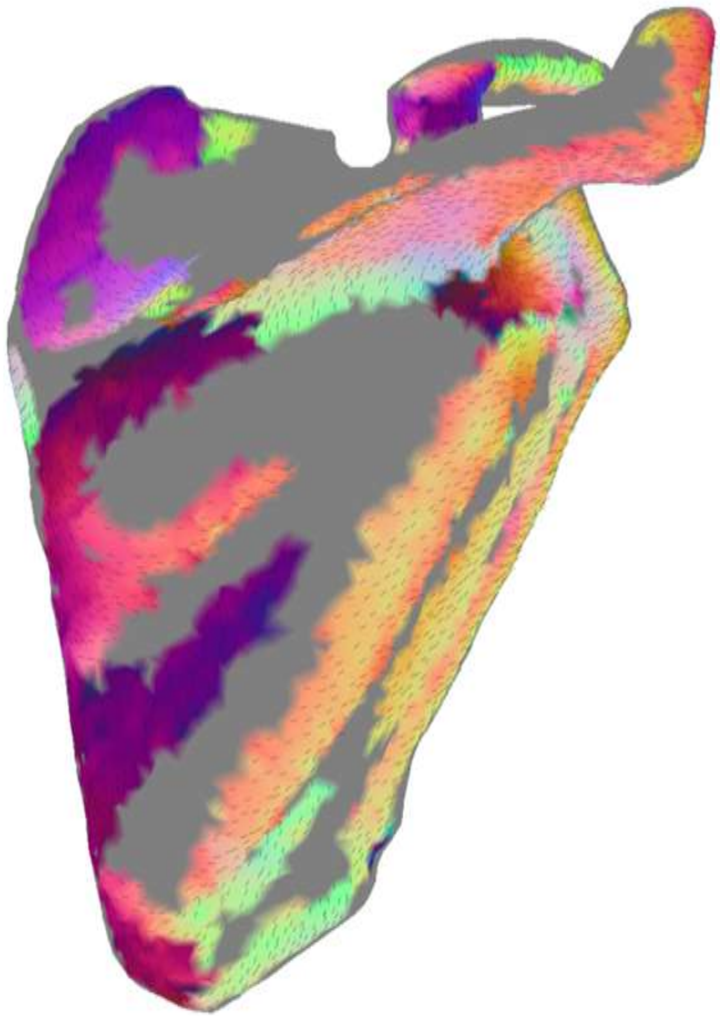
Learning Directions: Derive Direction Field



Learning Directions: Derive Direction Field

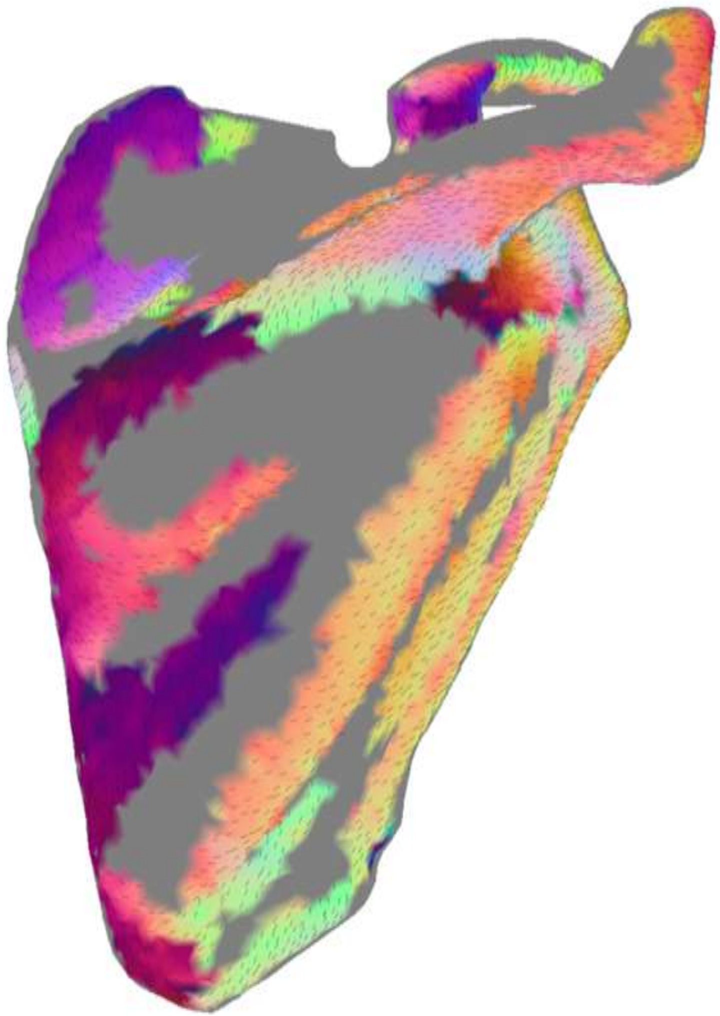


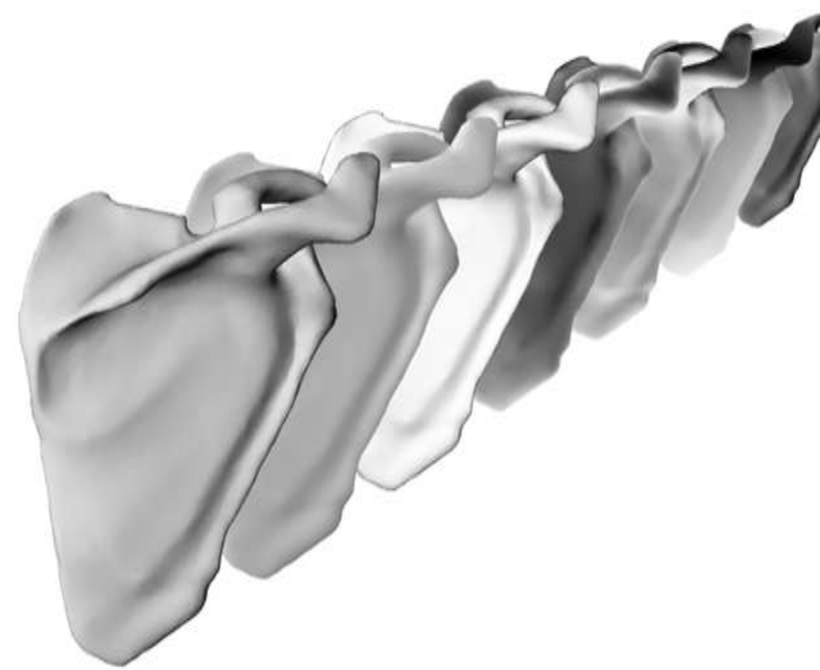
Learning Directions: Learn Function



(Kernel Ridge Regression using Radial Basis Function Kernels)

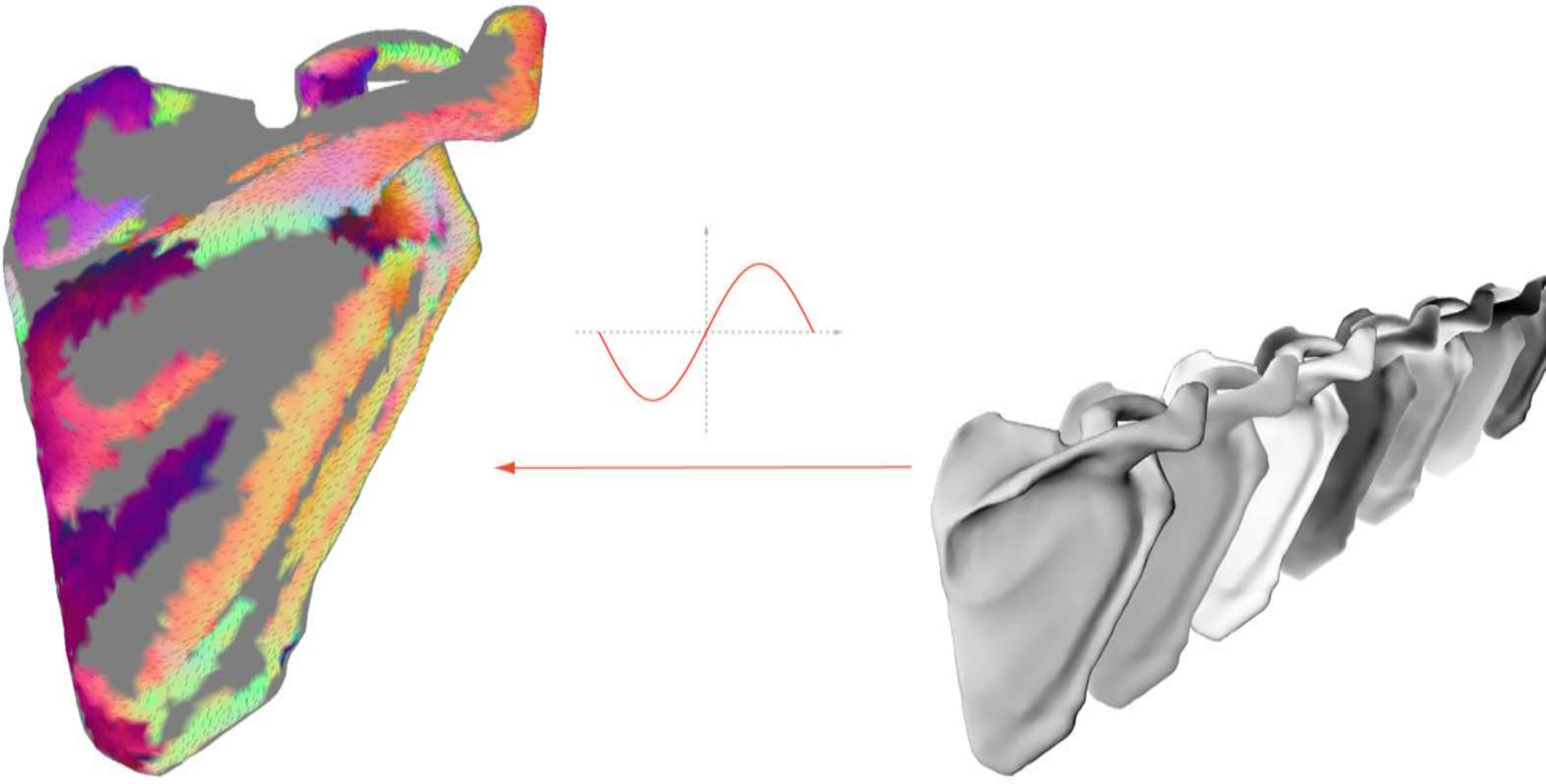
Learning Directions: Learn Function





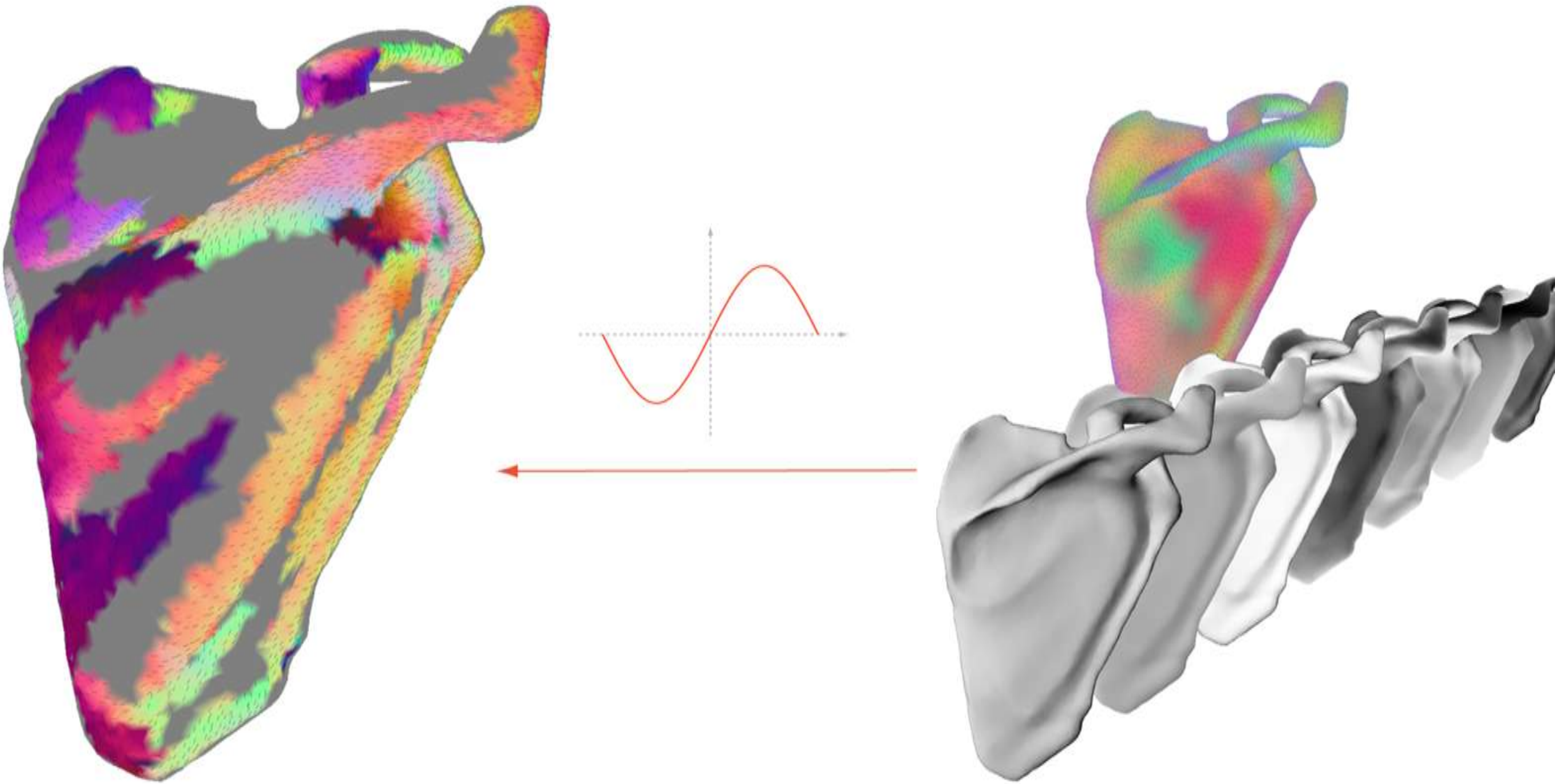
(Kernel Ridge Regression using Radial Basis Function Kernels)

Learning Directions: Learn Function



(Kernel Ridge Regression using Radial Basis Function Kernels)

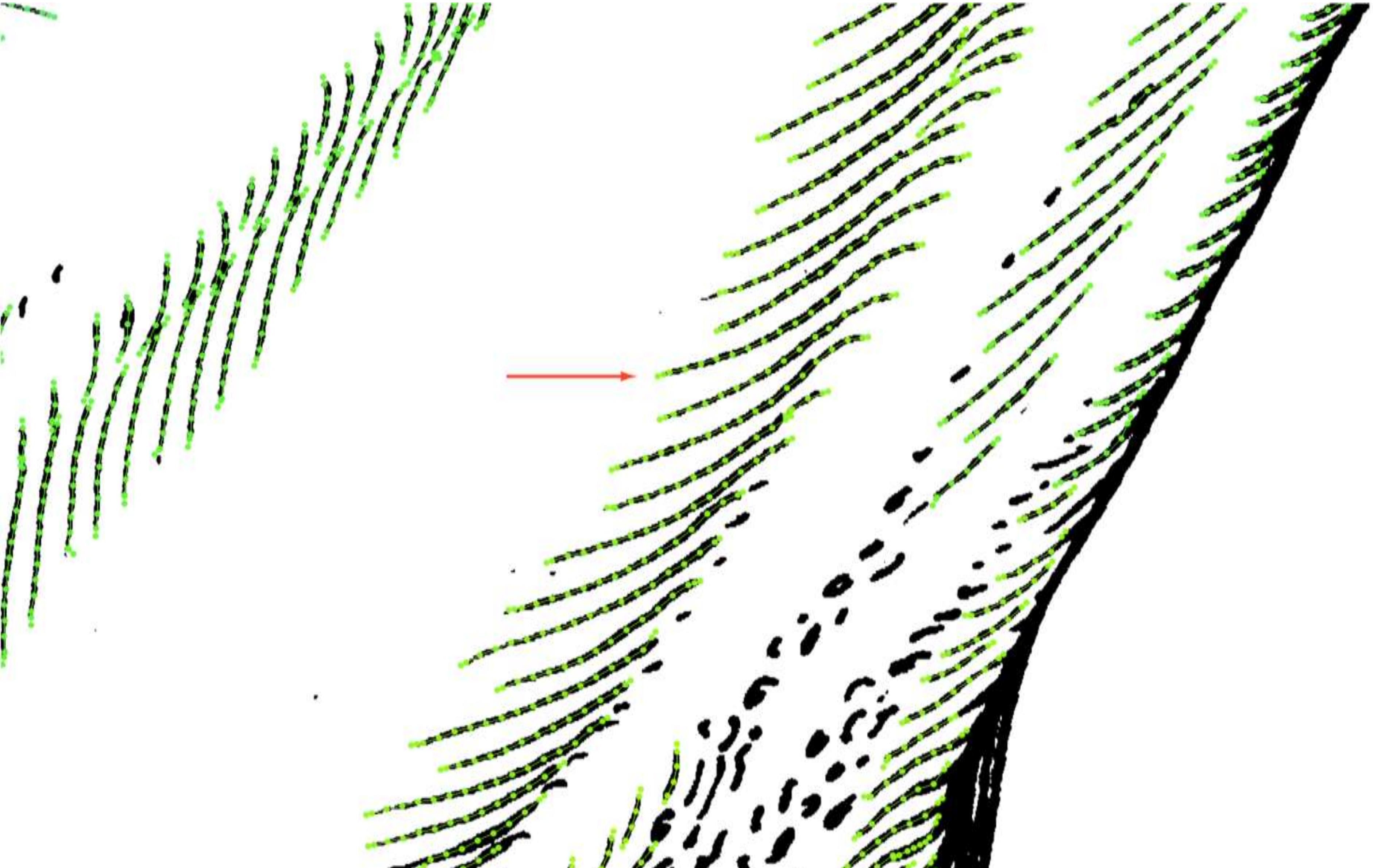
Learning Directions: Learn Function



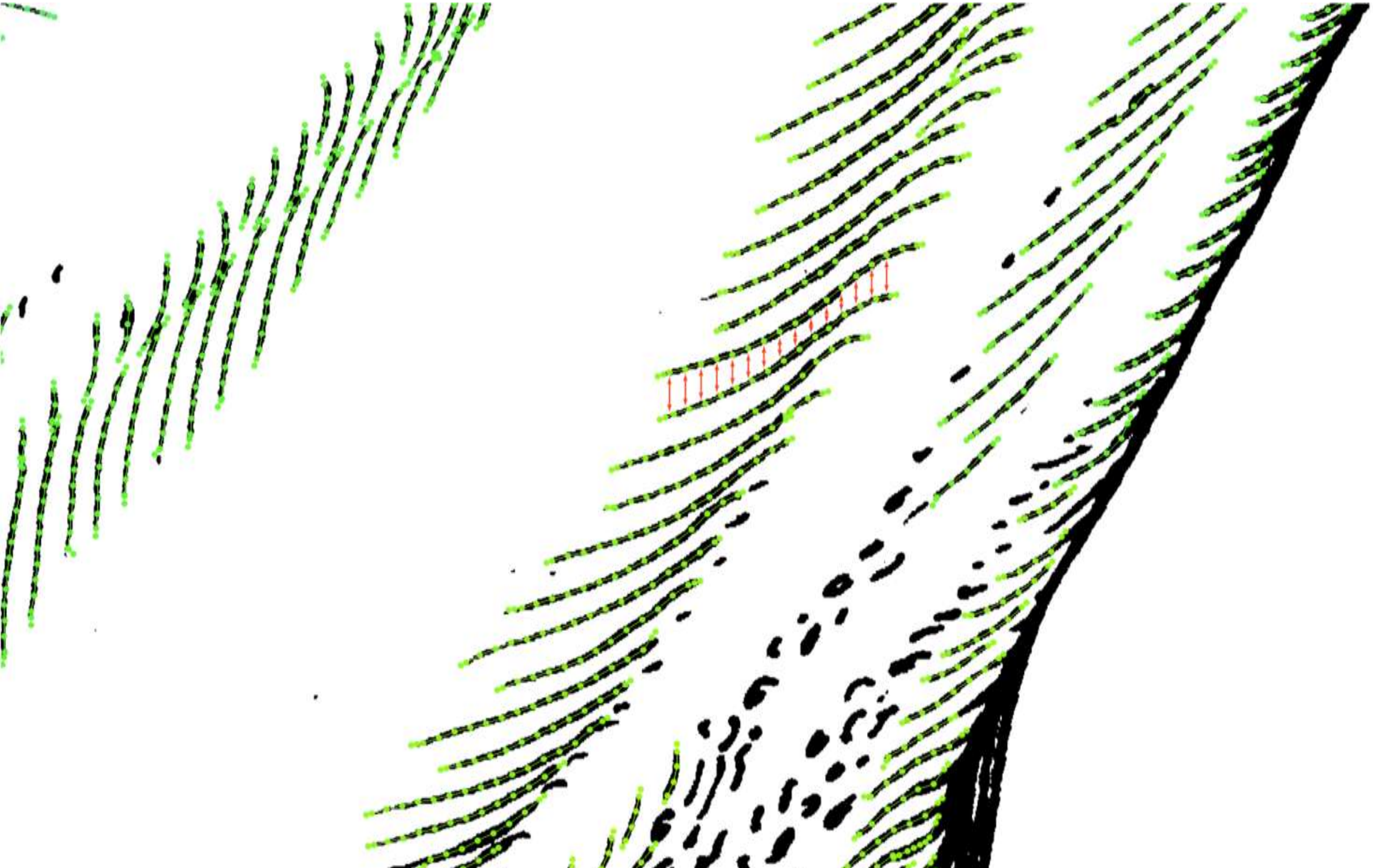
(Kernel Ridge Regression using Radial Basis Function Kernels)

Directions – Synthesis

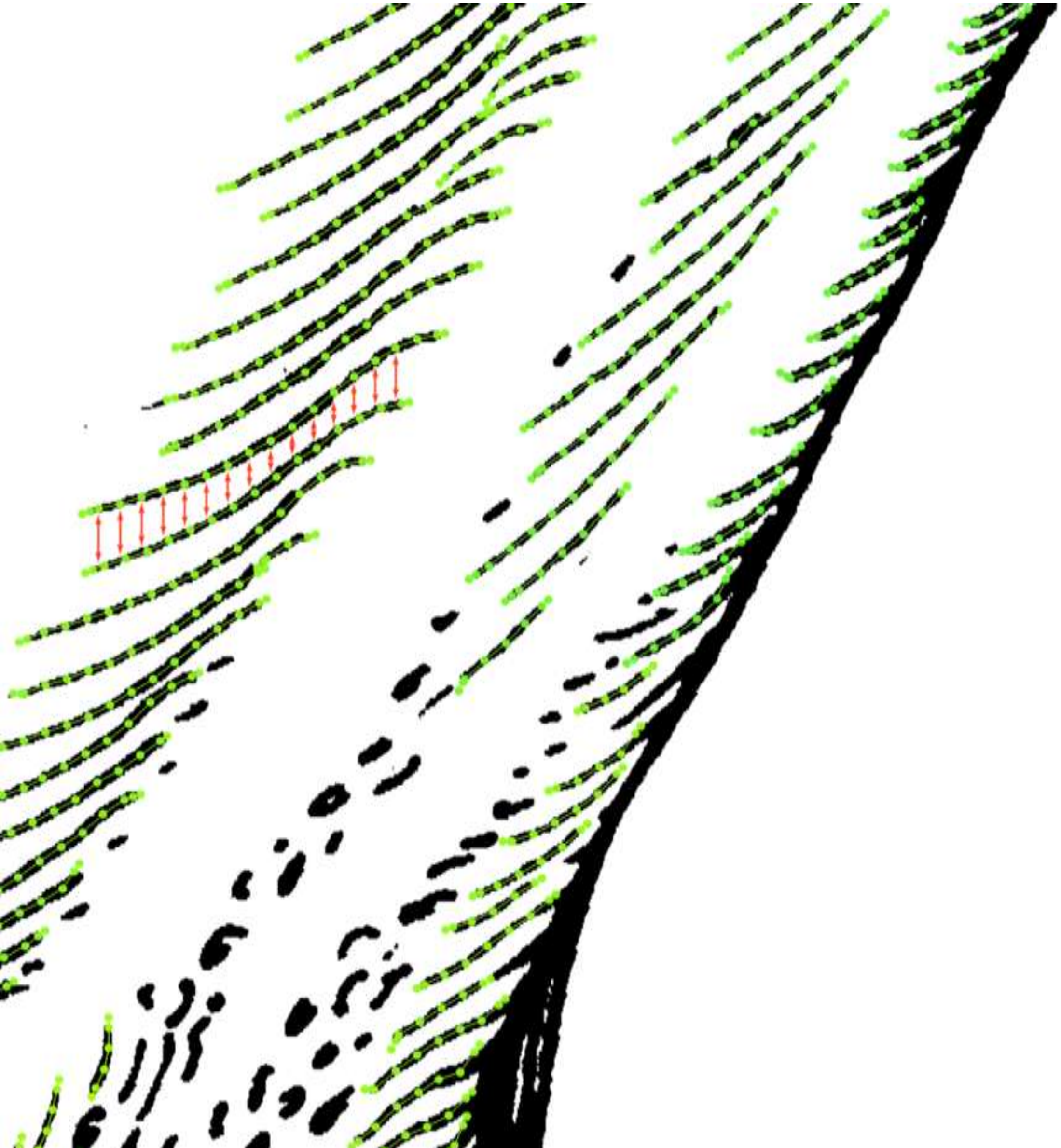
Distances – Learning



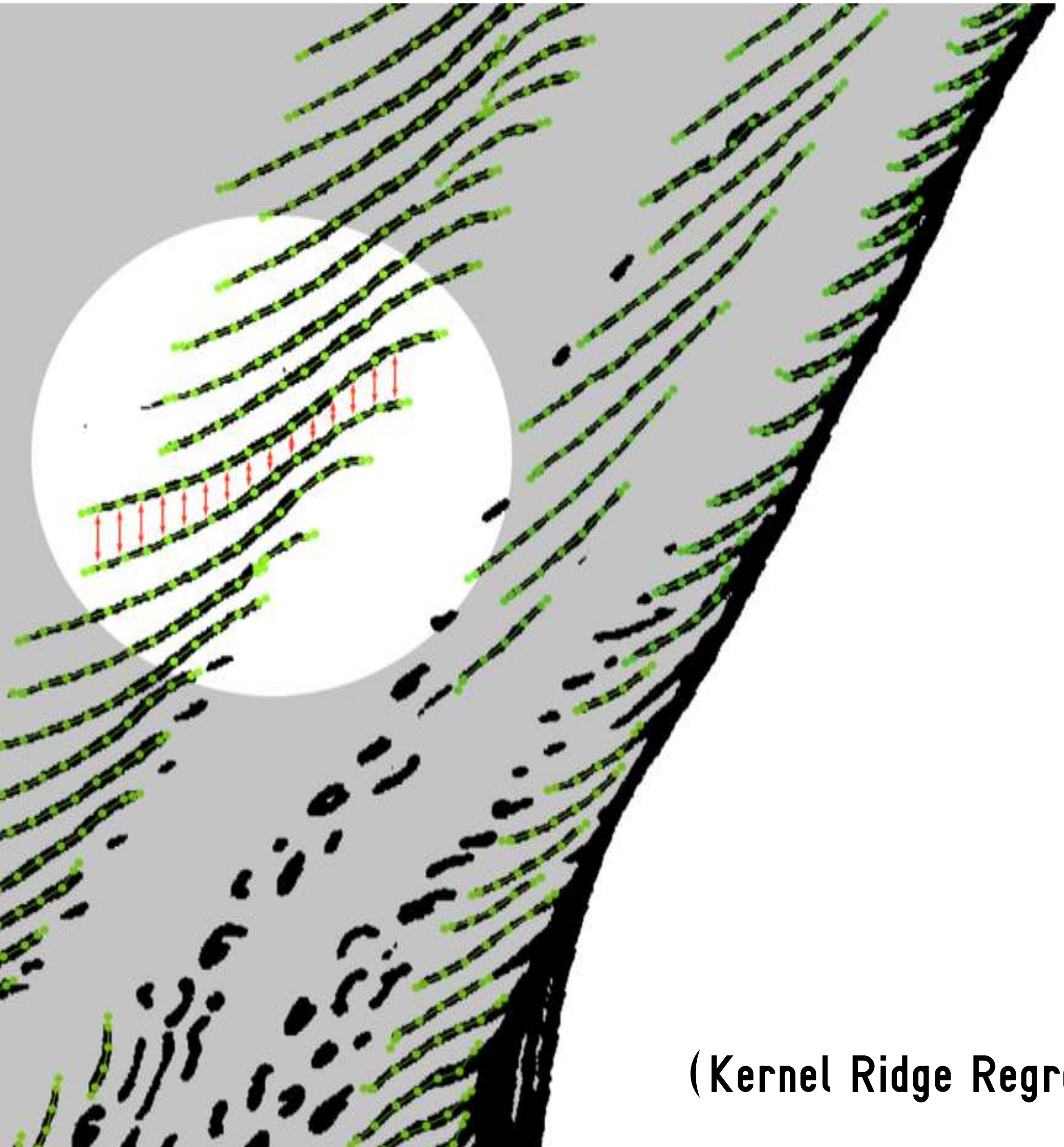
Distances – Learning

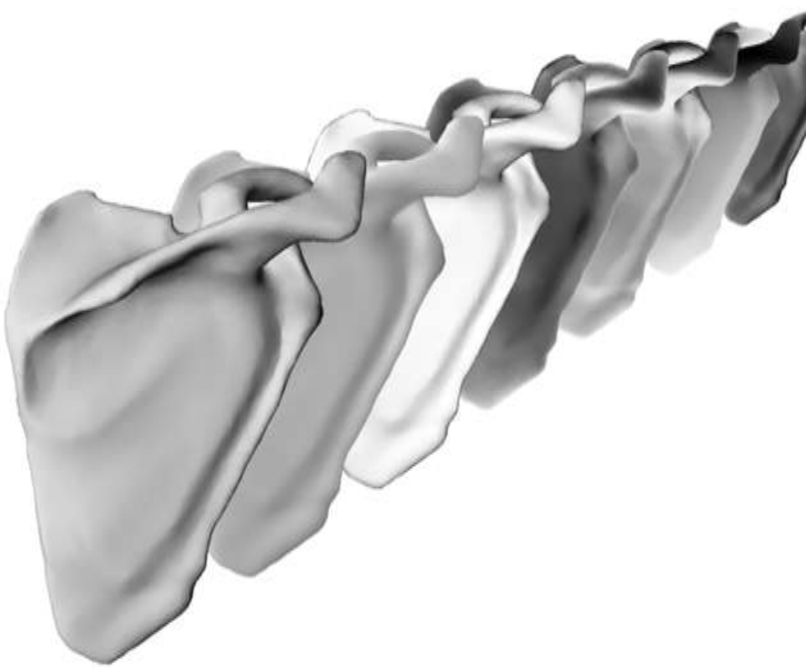


Distances – Learning



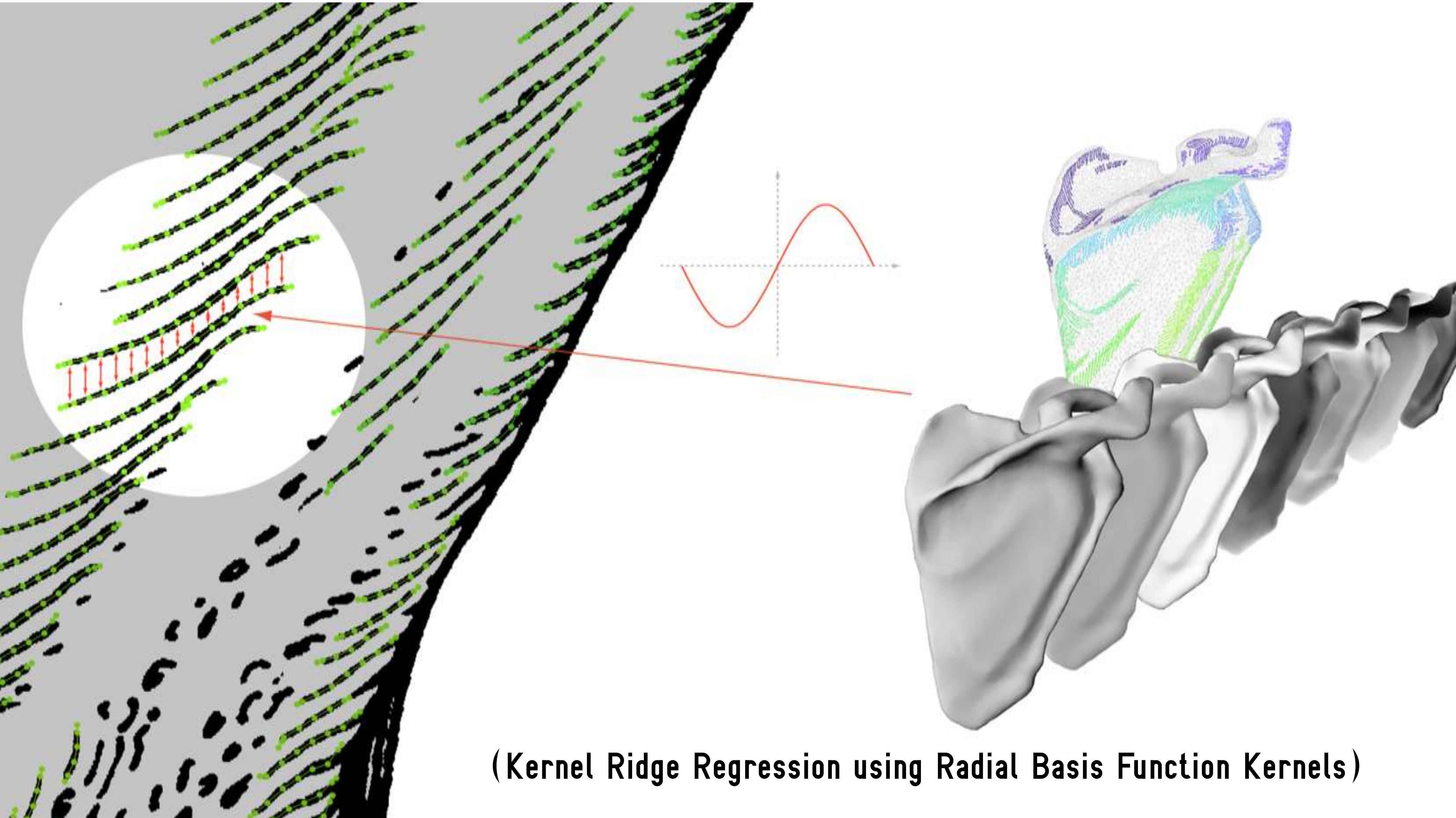
Distances – Learning





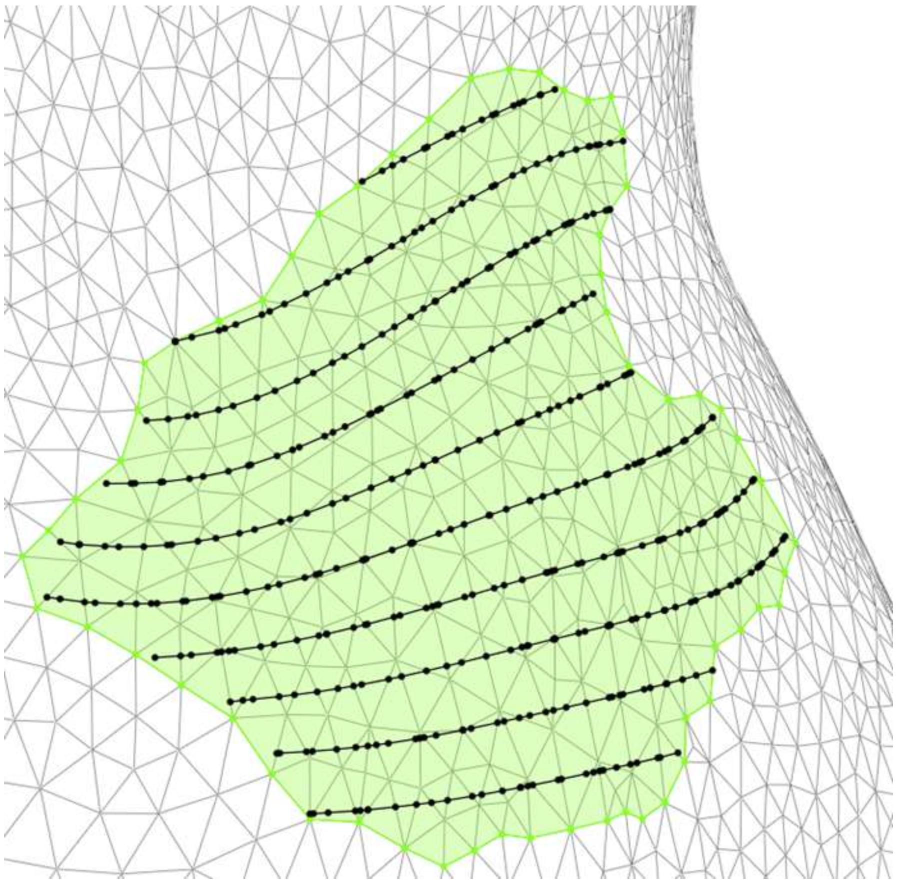
(Kernel Ridge Regression using Radial Basis Function Kernels)

Distances – Learning

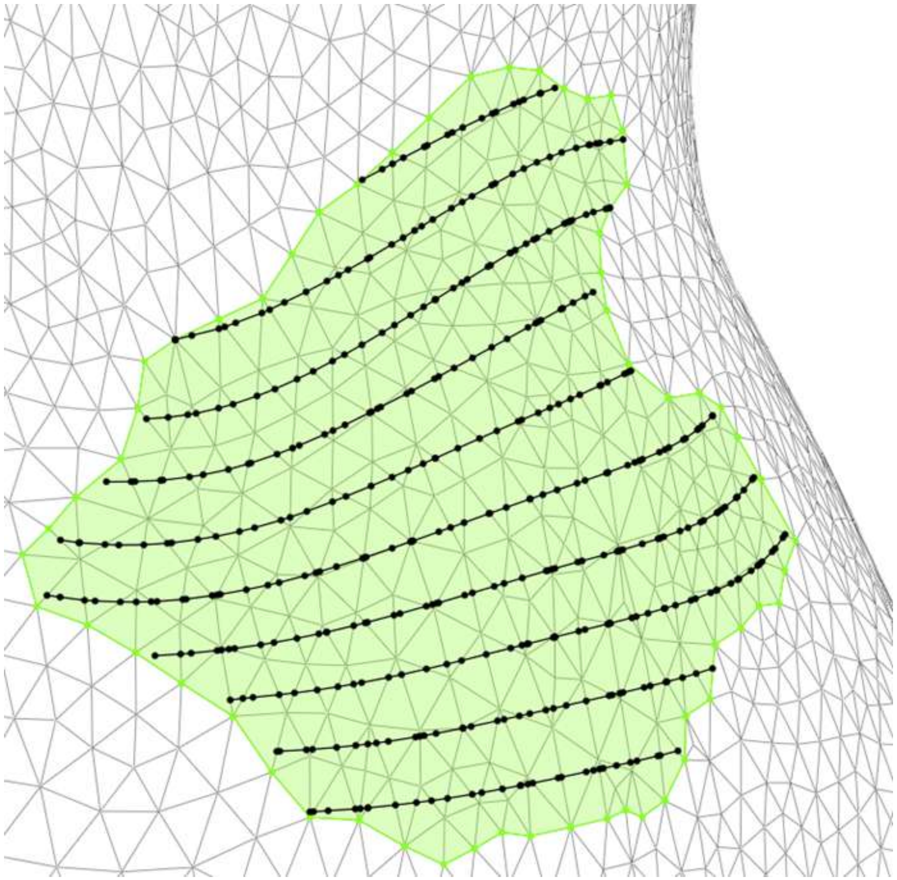


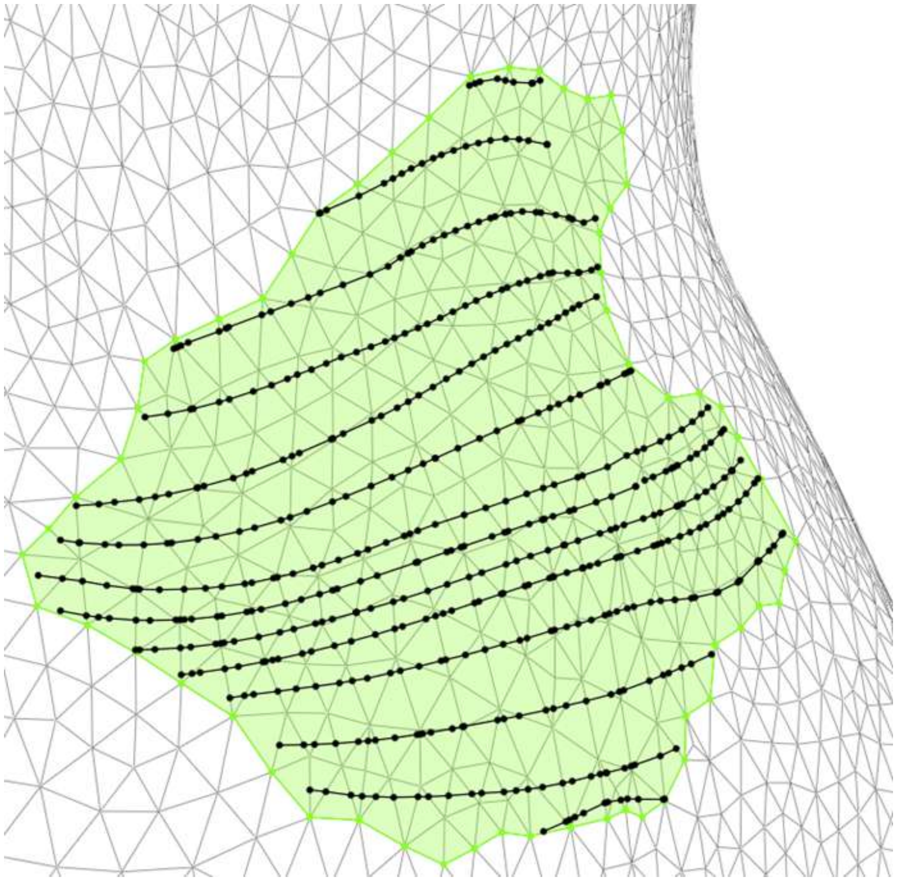
(Kernel Ridge Regression using Radial Basis Function Kernels)

Distances – Synthesis

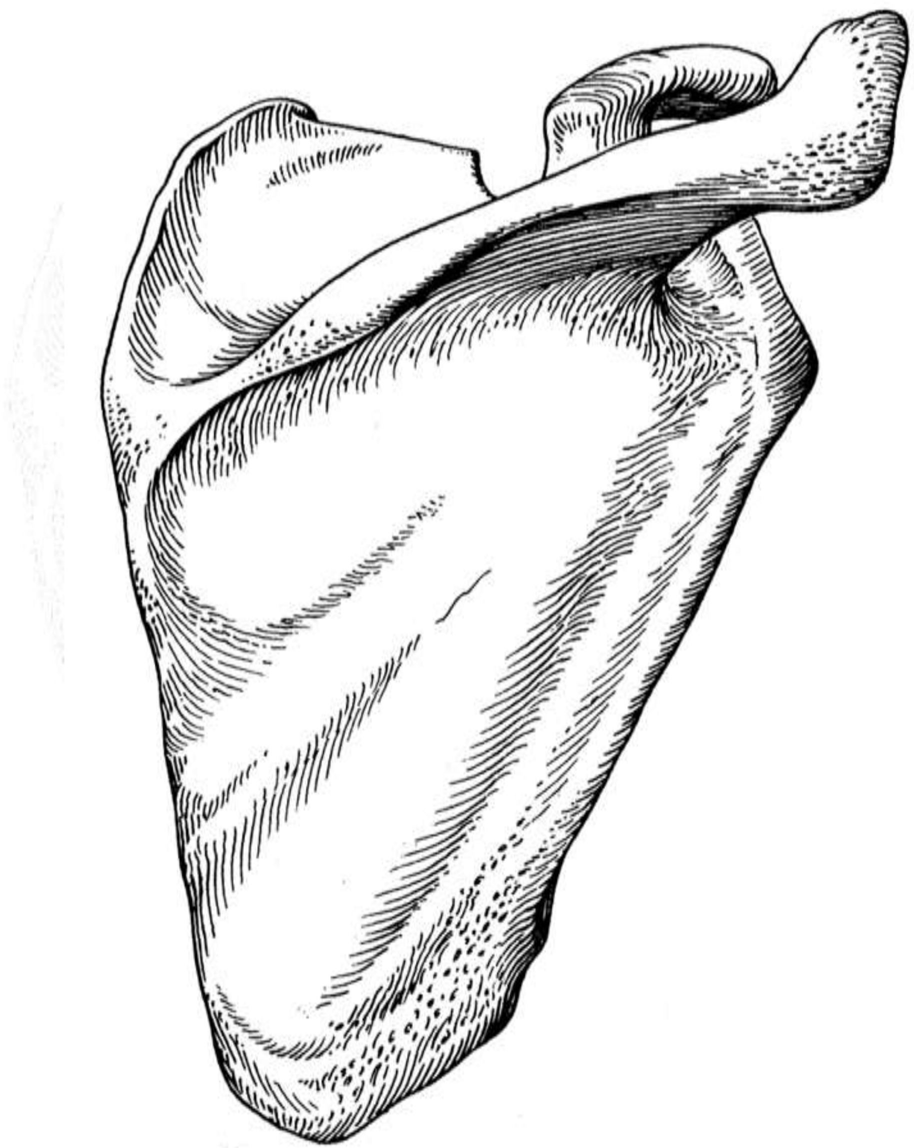


Distances – Synthesis

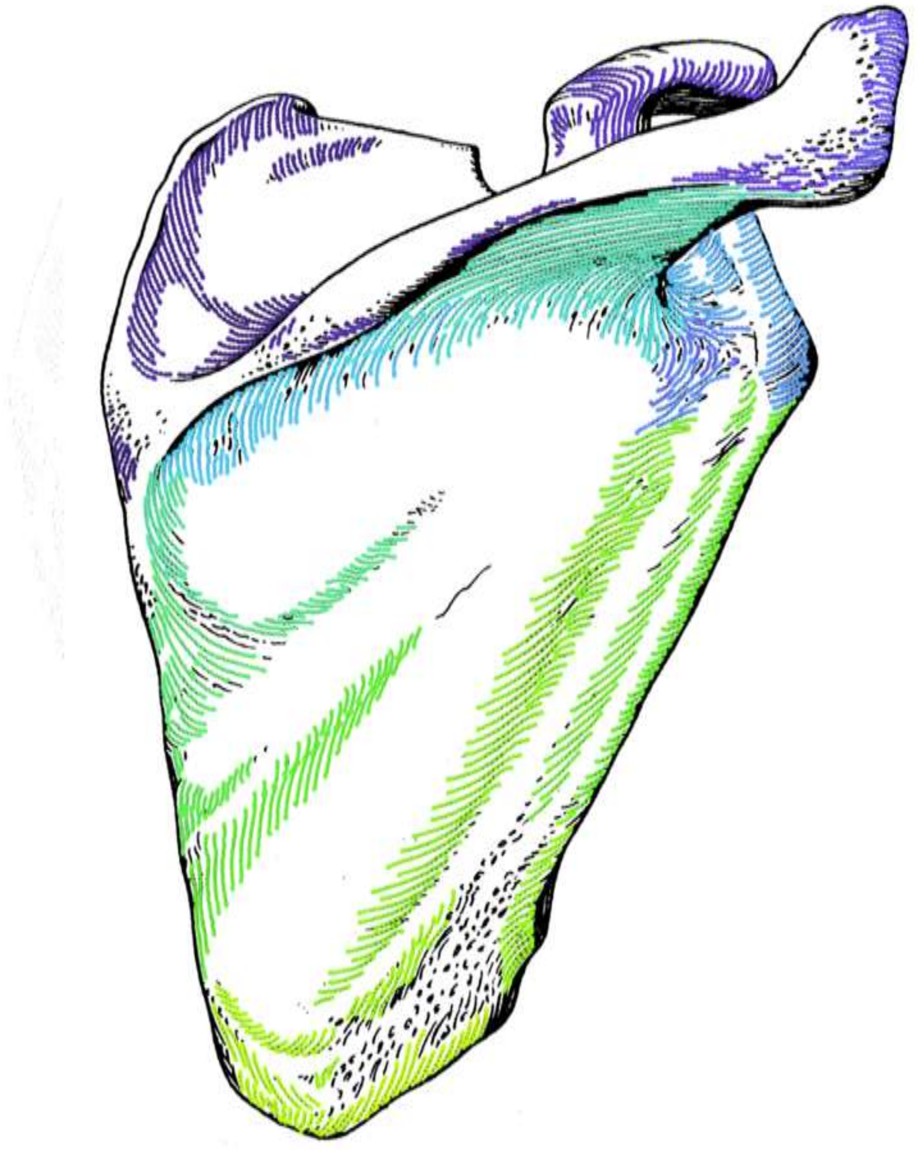




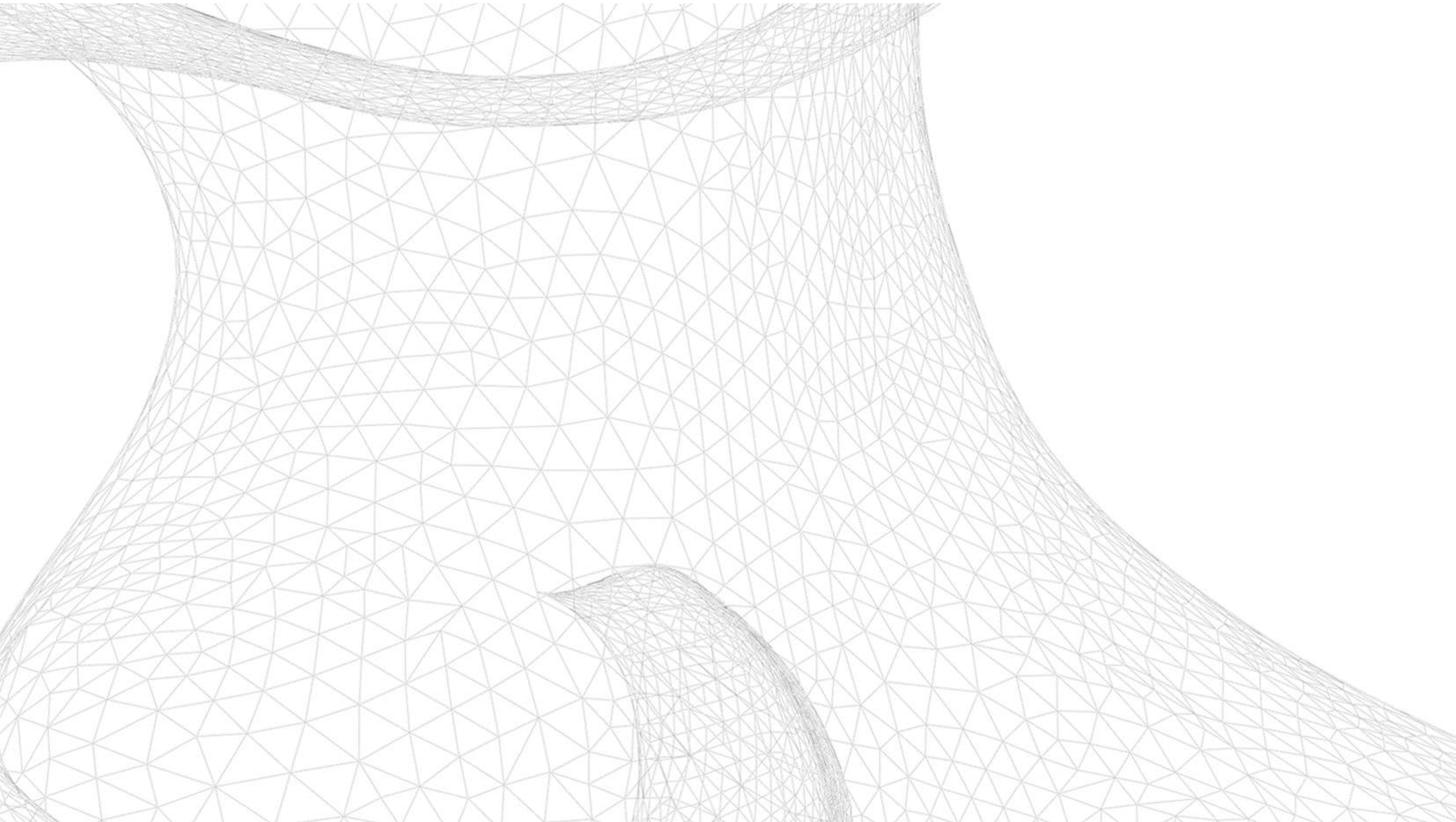
Stroke Rendering



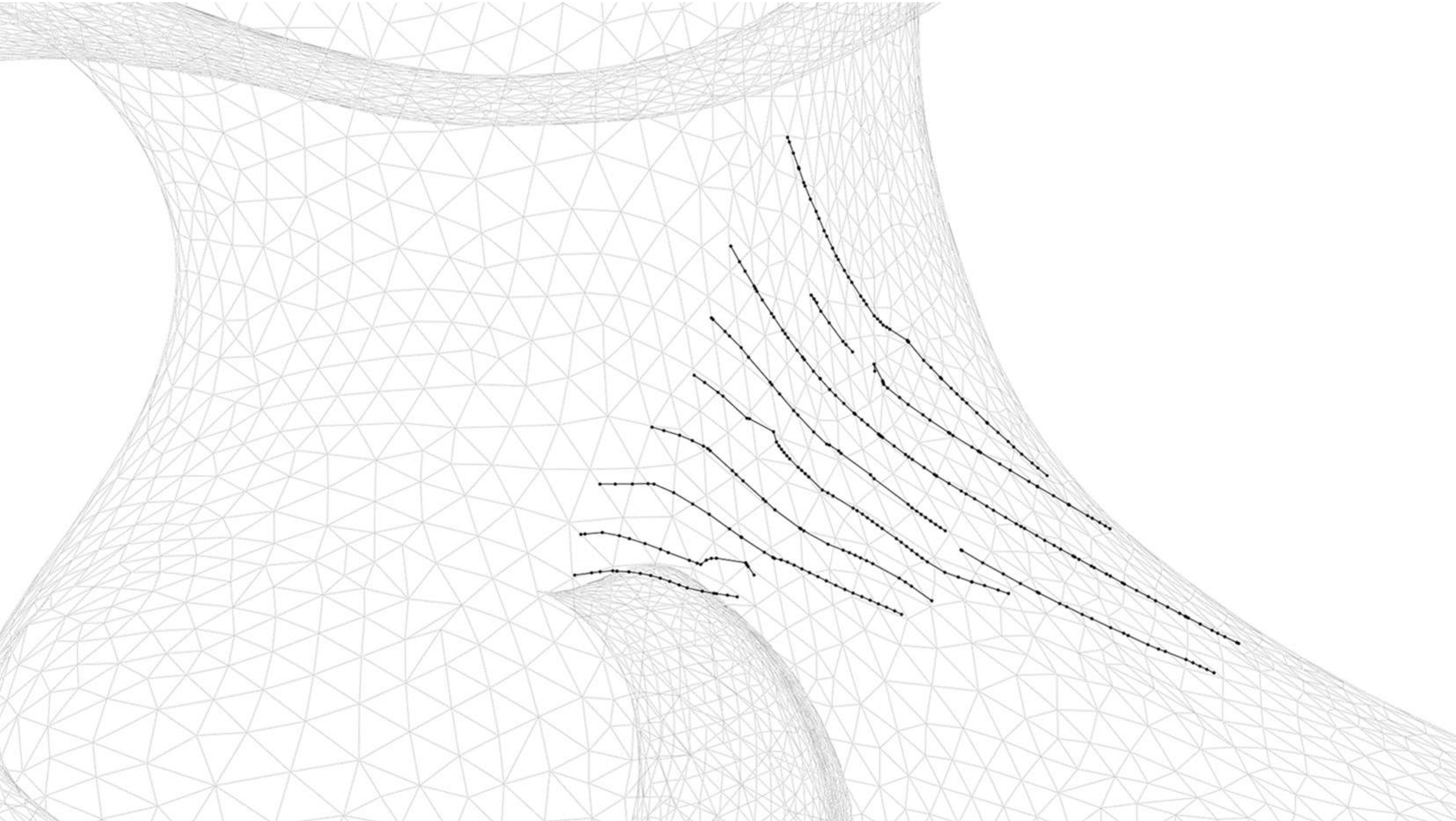
Stroke Rendering



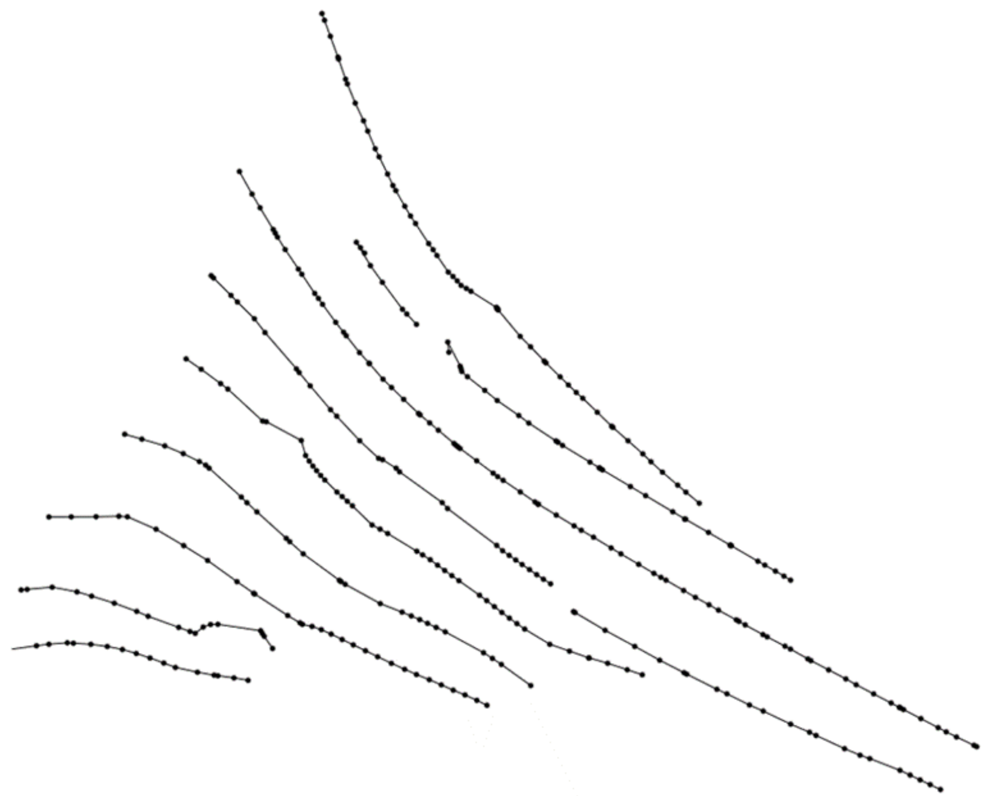
Stroke Rendering



Stroke Rendering

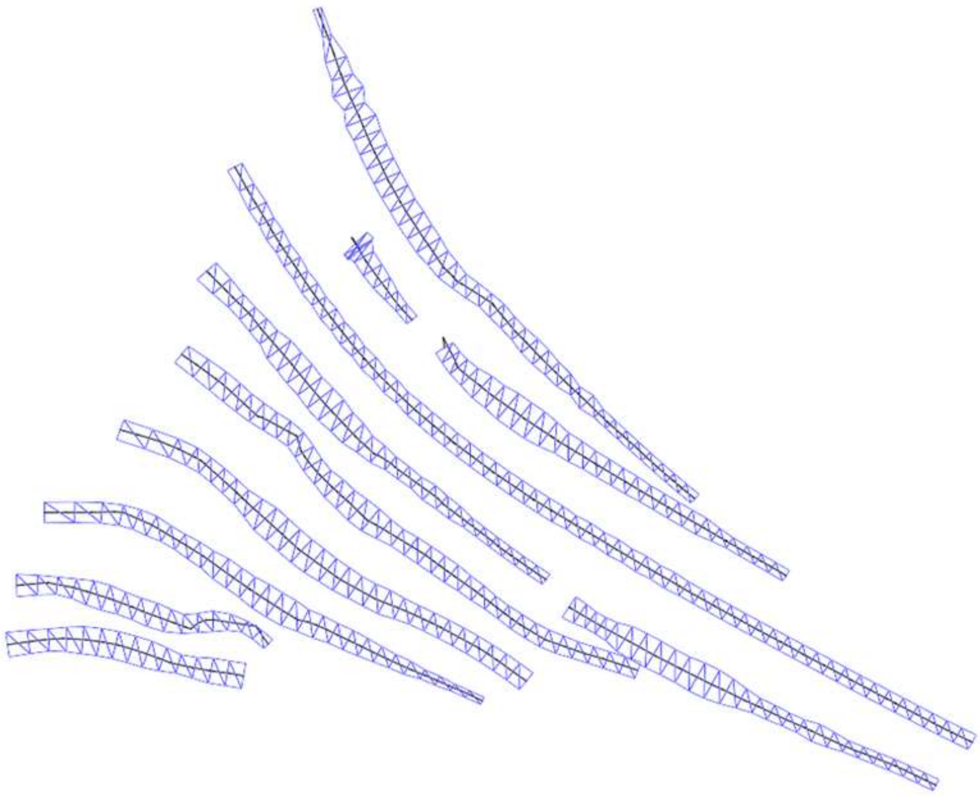


Stroke Rendering

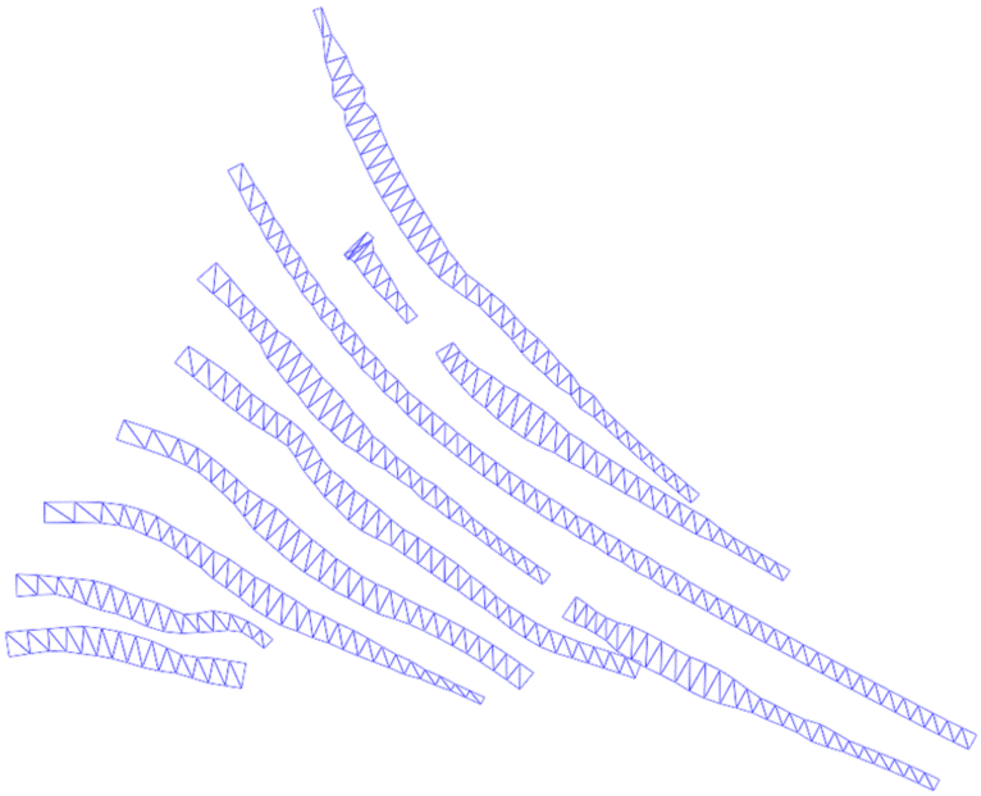


Stroke Rendering

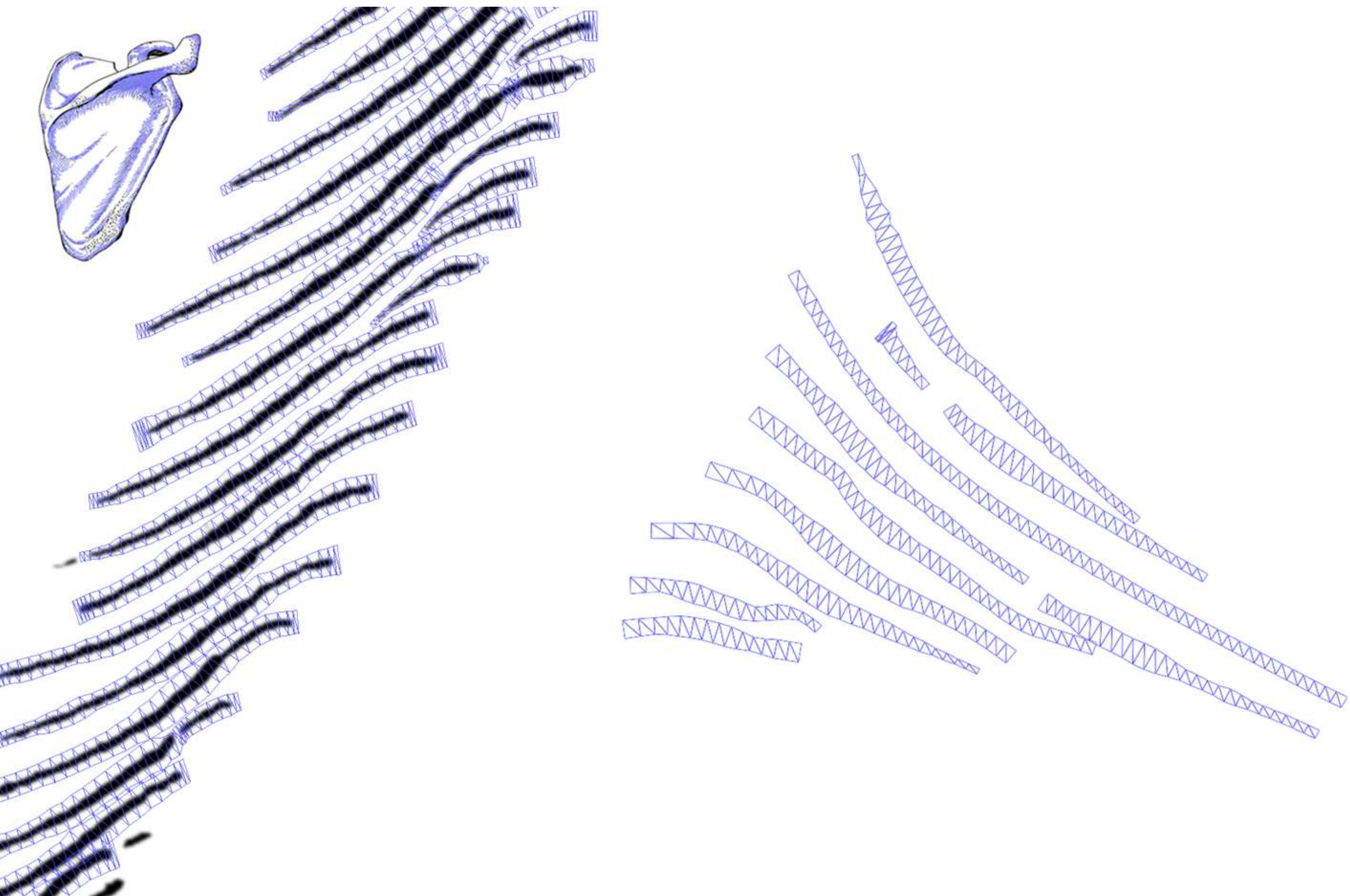
Stroke Rendering



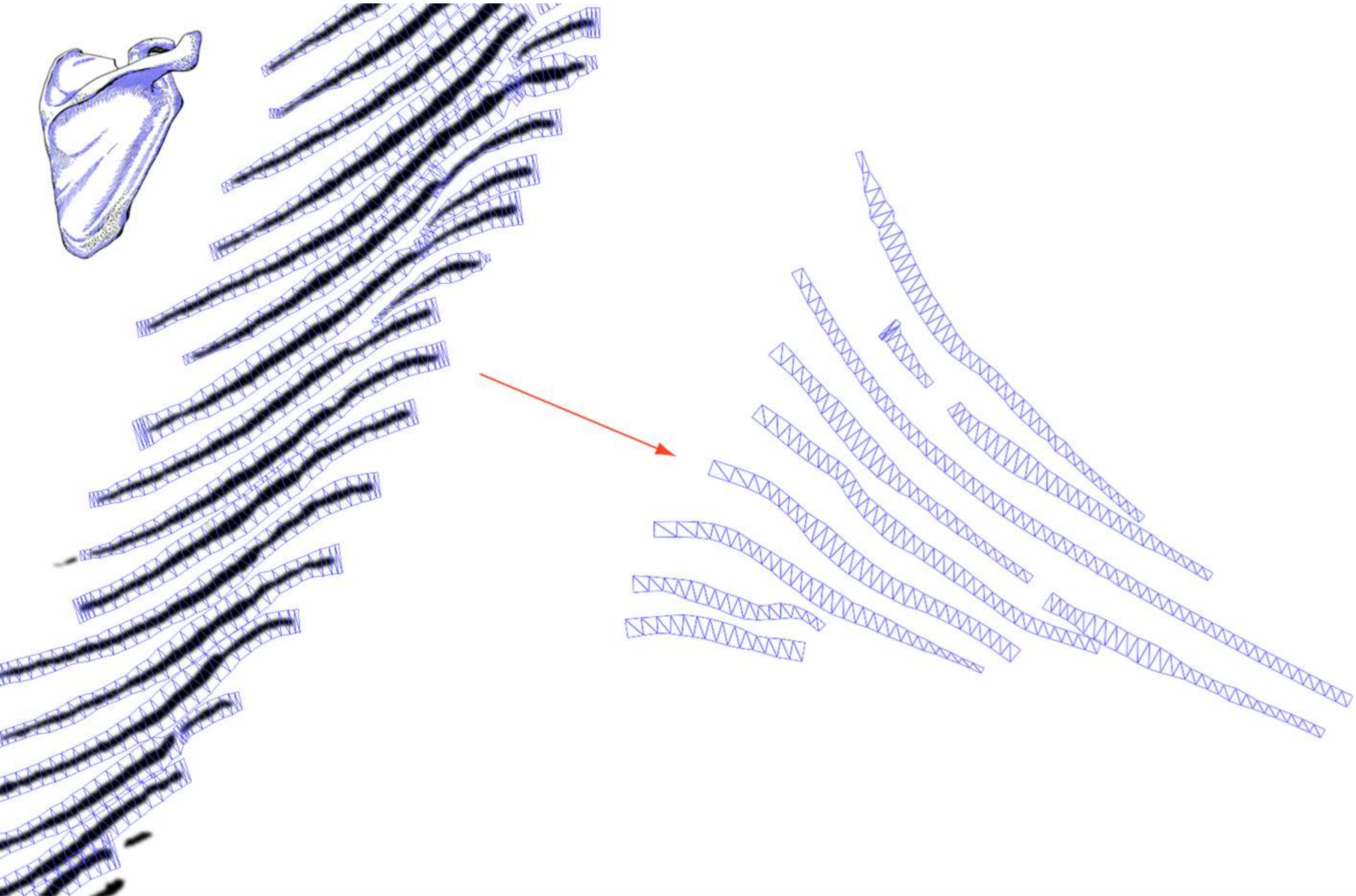
Stroke Rendering



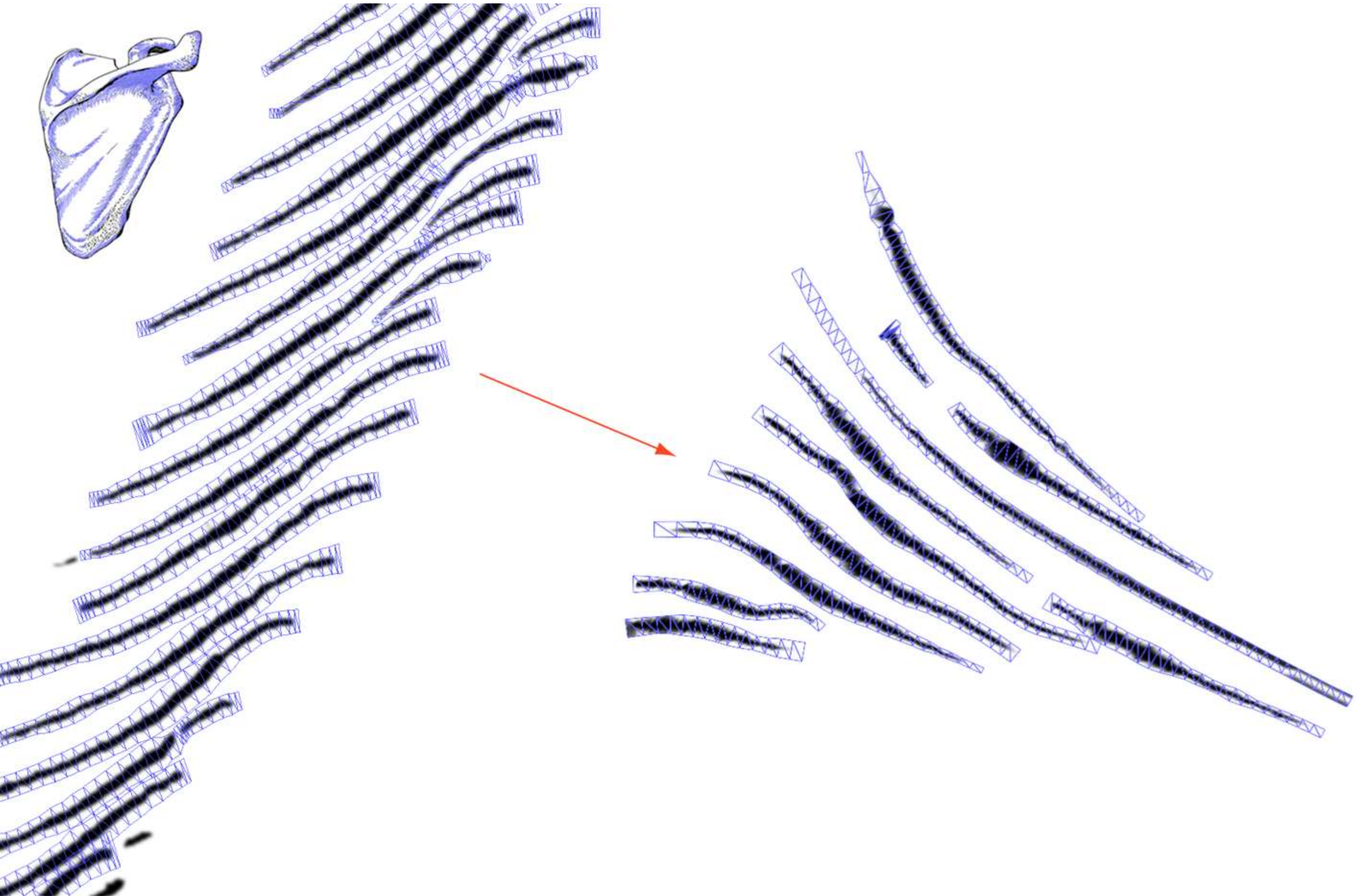
Stroke Rendering



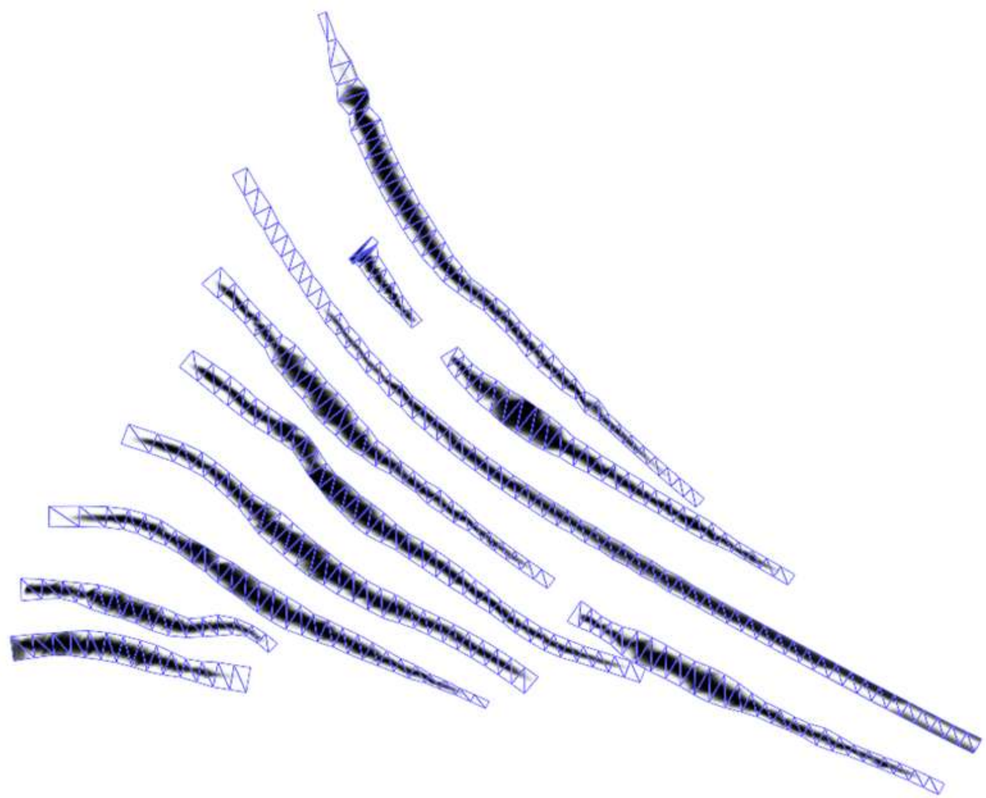
Stroke Rendering



Stroke Rendering

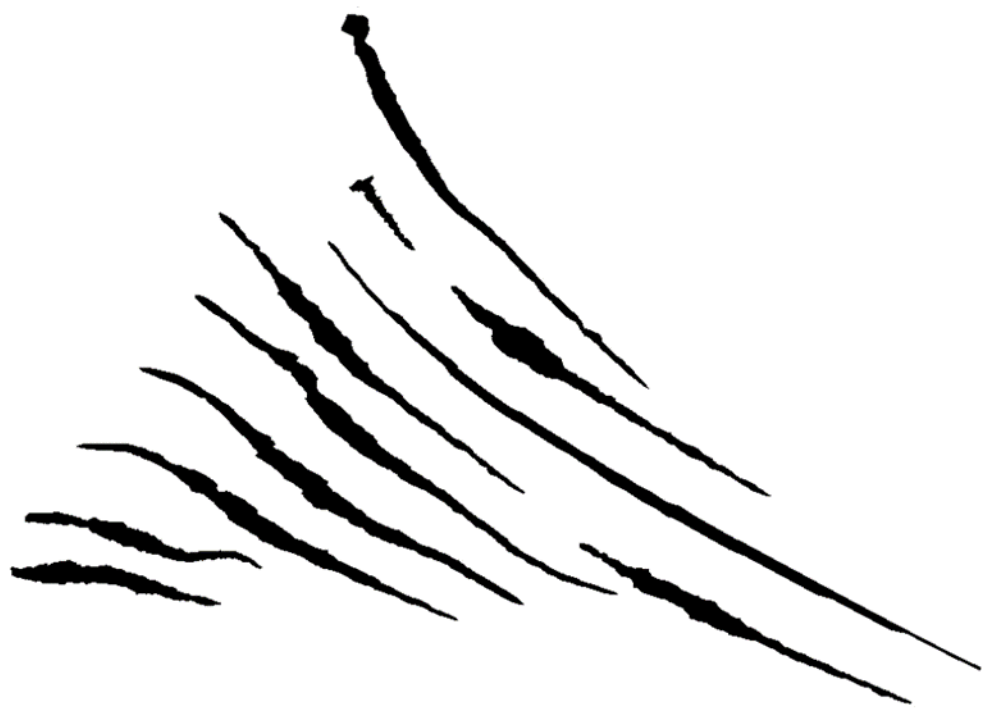


Stroke Rendering



Stroke Rendering

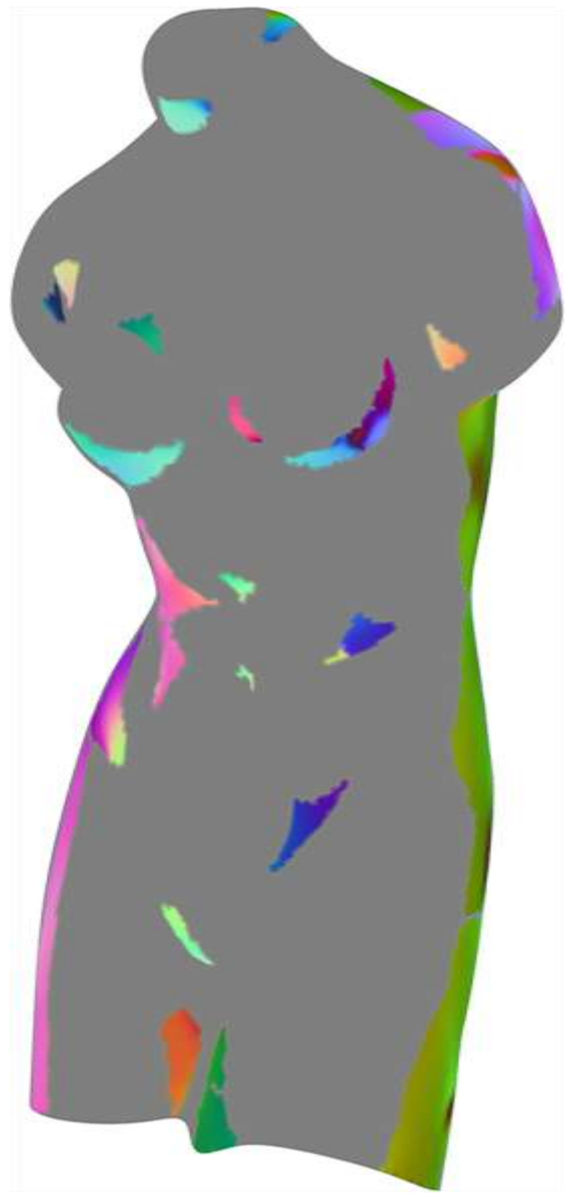
Stroke Rendering



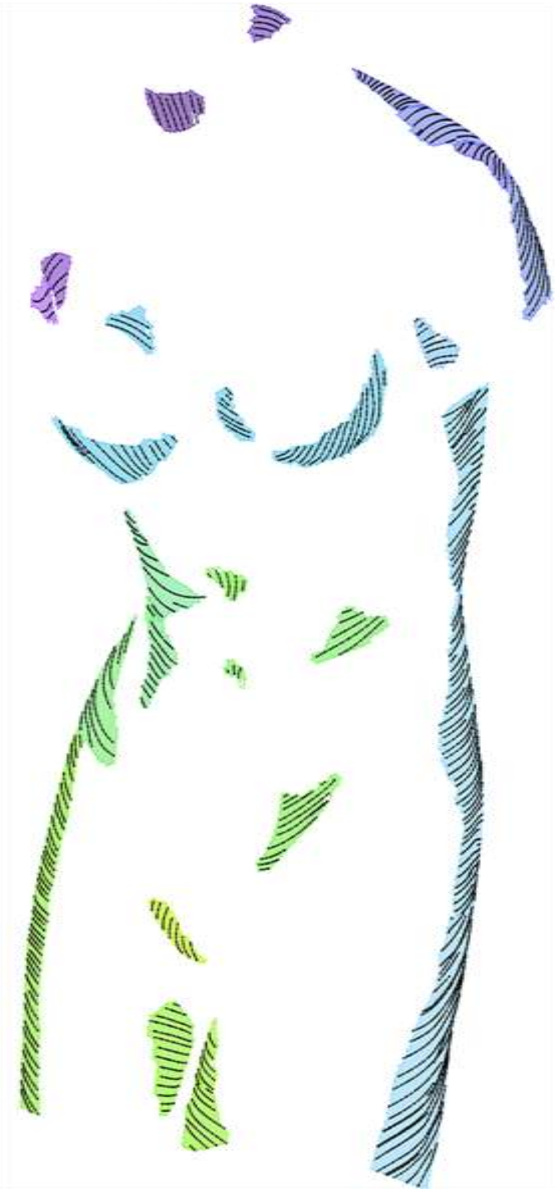
Synthesis - Summary

Synthesis - Summary

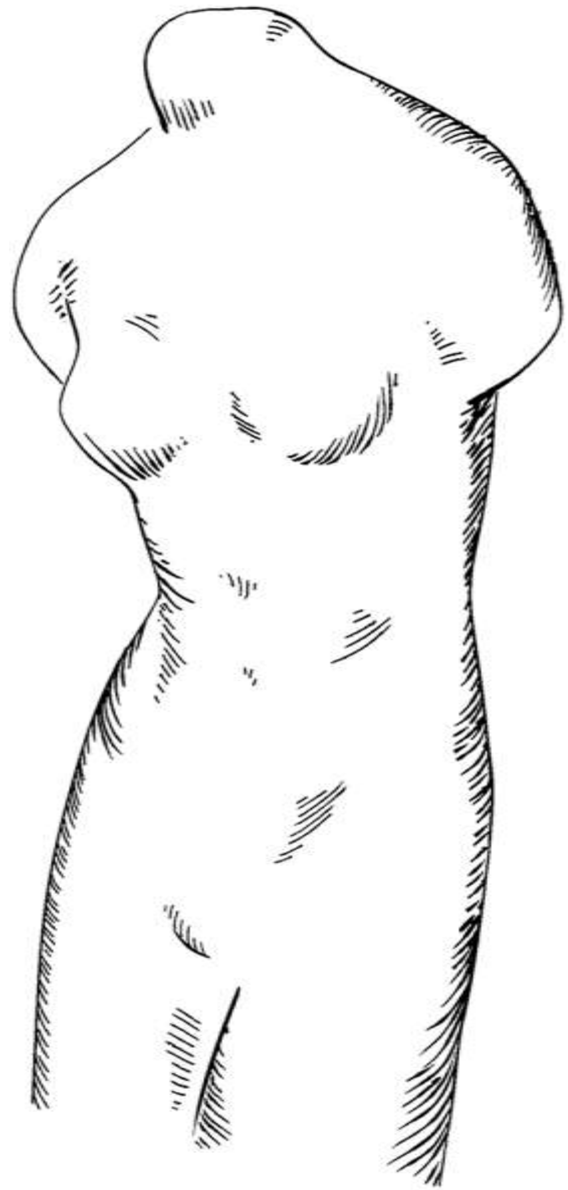
Synthesis - Summary



Synthesis - Summary

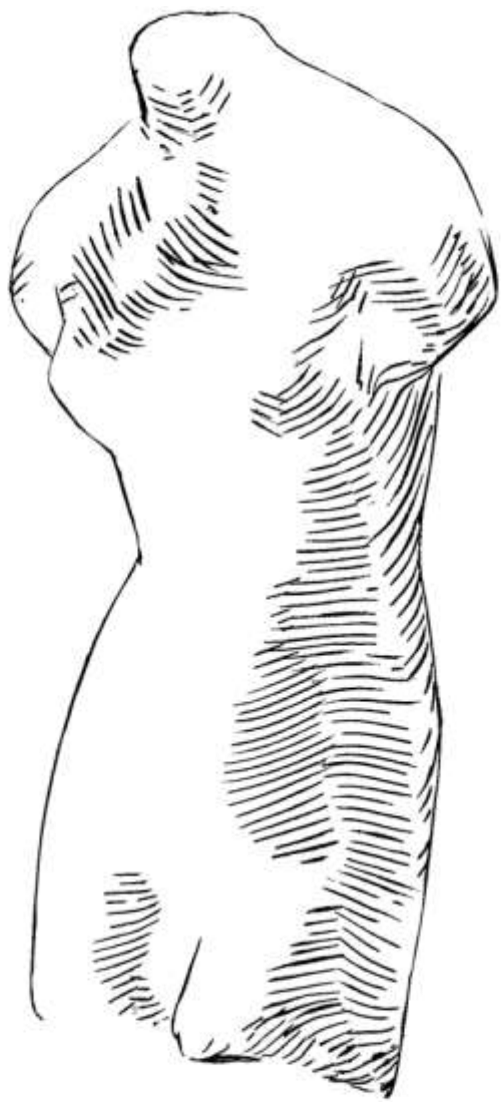


Synthesis - Summary



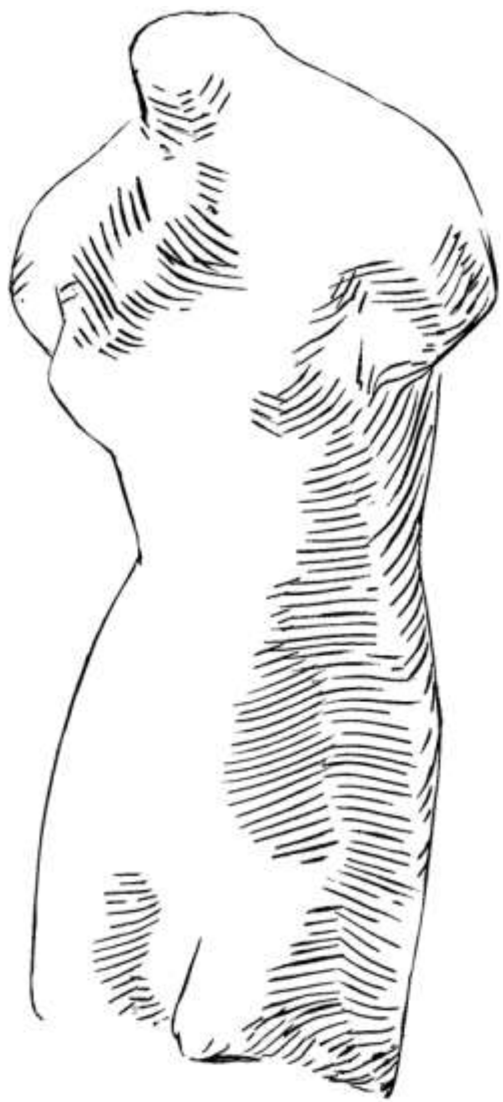
Automatic vs. Semi-automatic

Automatic vs. Semi-automatic

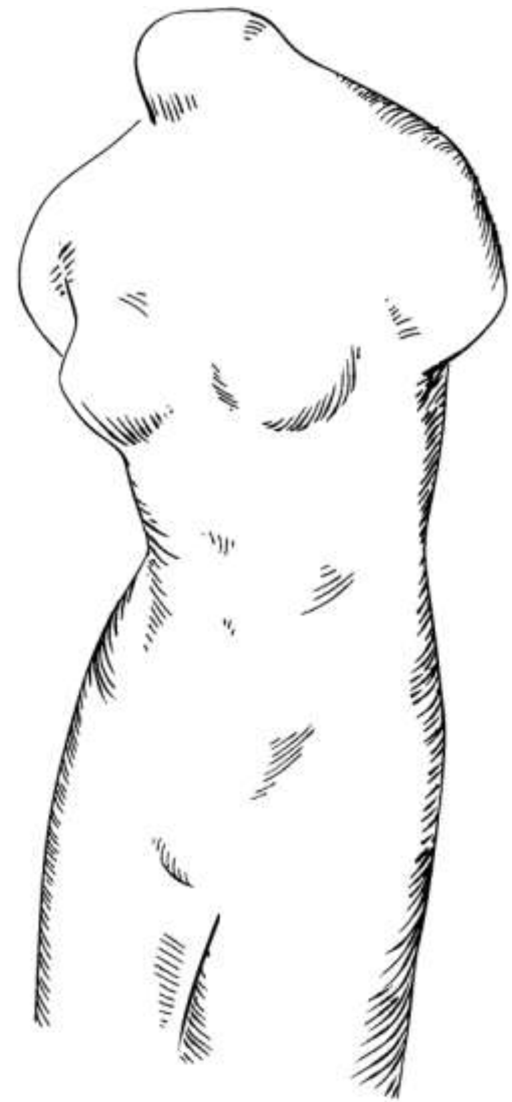


fully automatic

Automatic vs. Semi-automatic



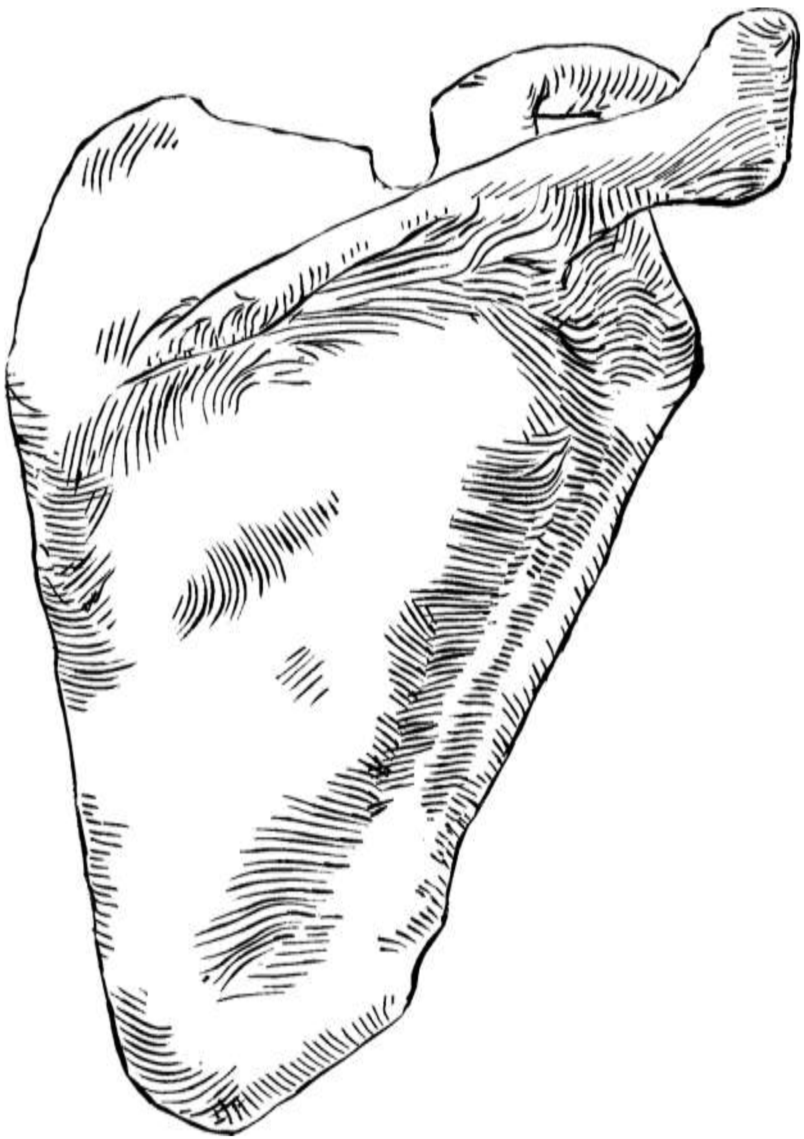
fully automatic



semi-automatic, human input

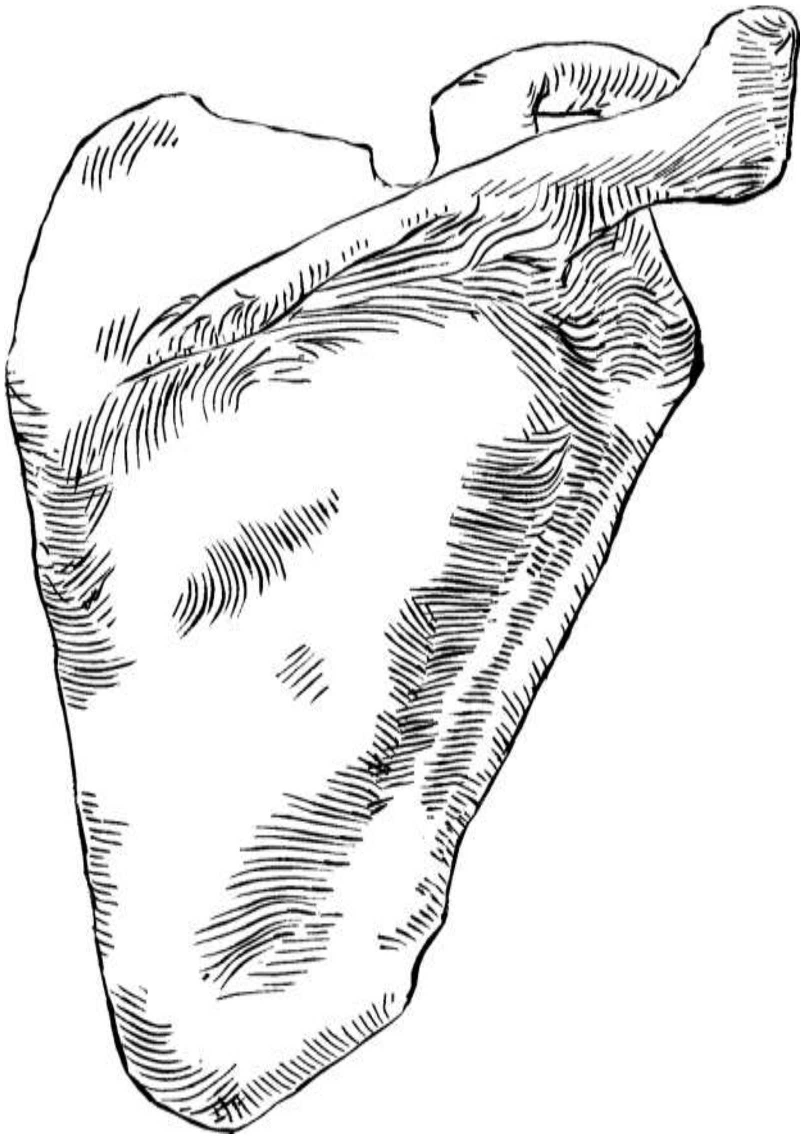
Automatic vs. Semi-automatic

Automatic vs. Semi-automatic

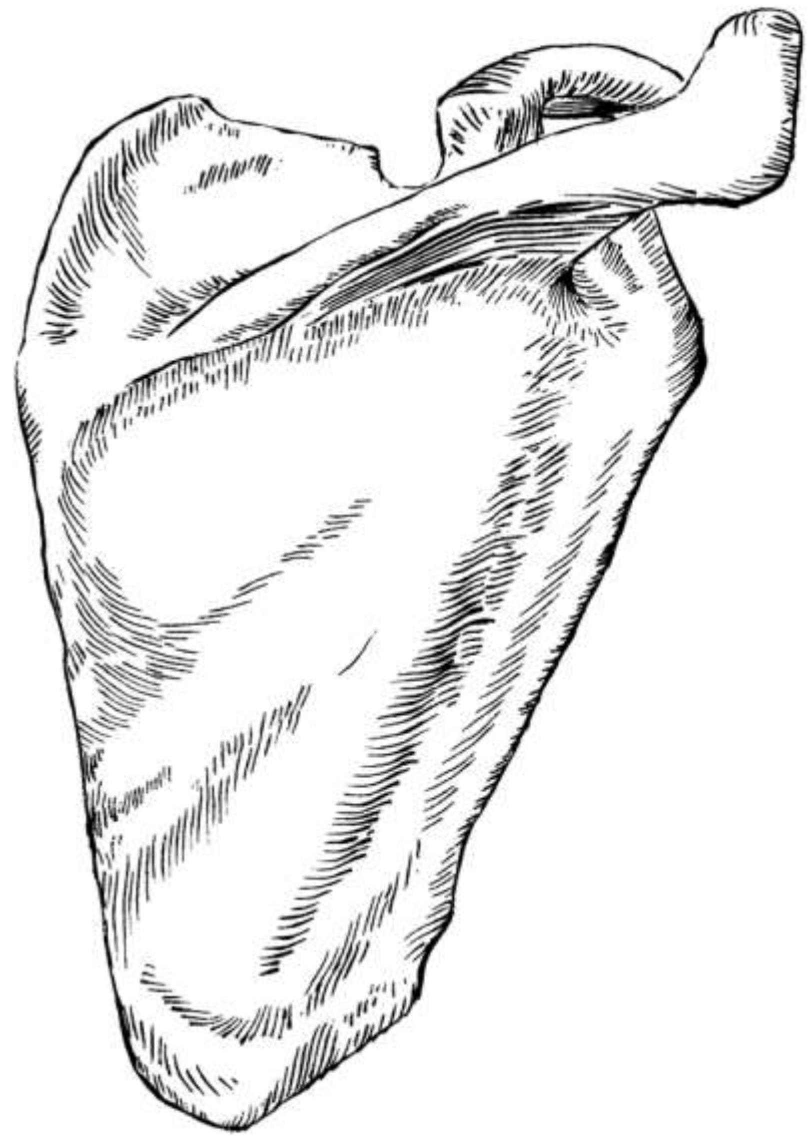


fully automatic

Automatic vs. Semi-automatic



fully automatic

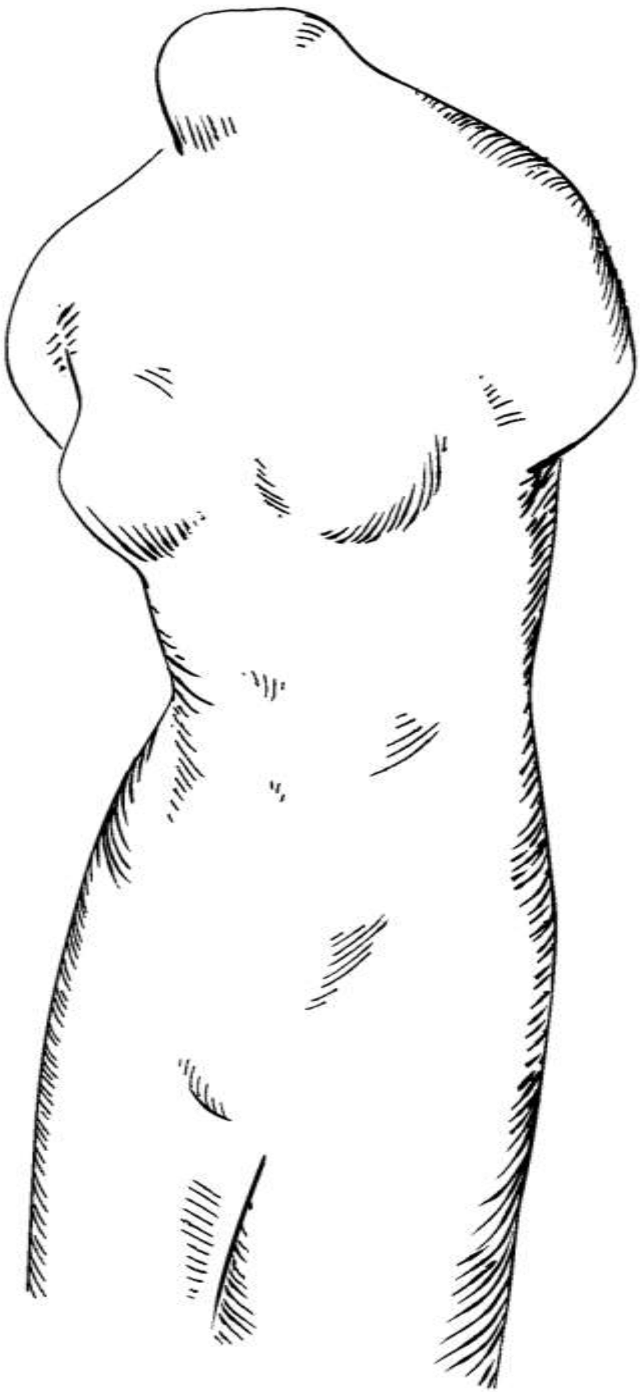


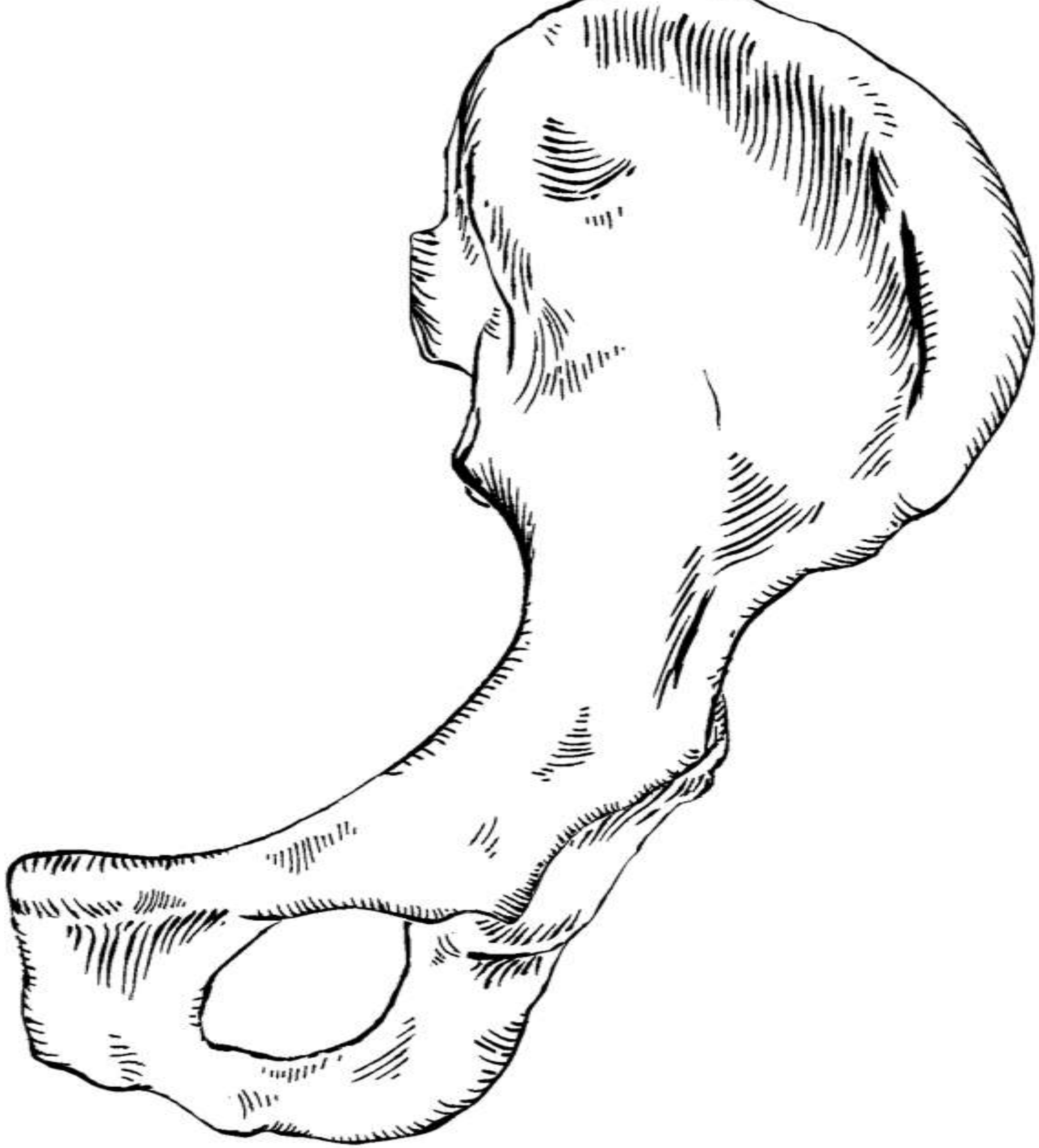
semi-automatic, human input

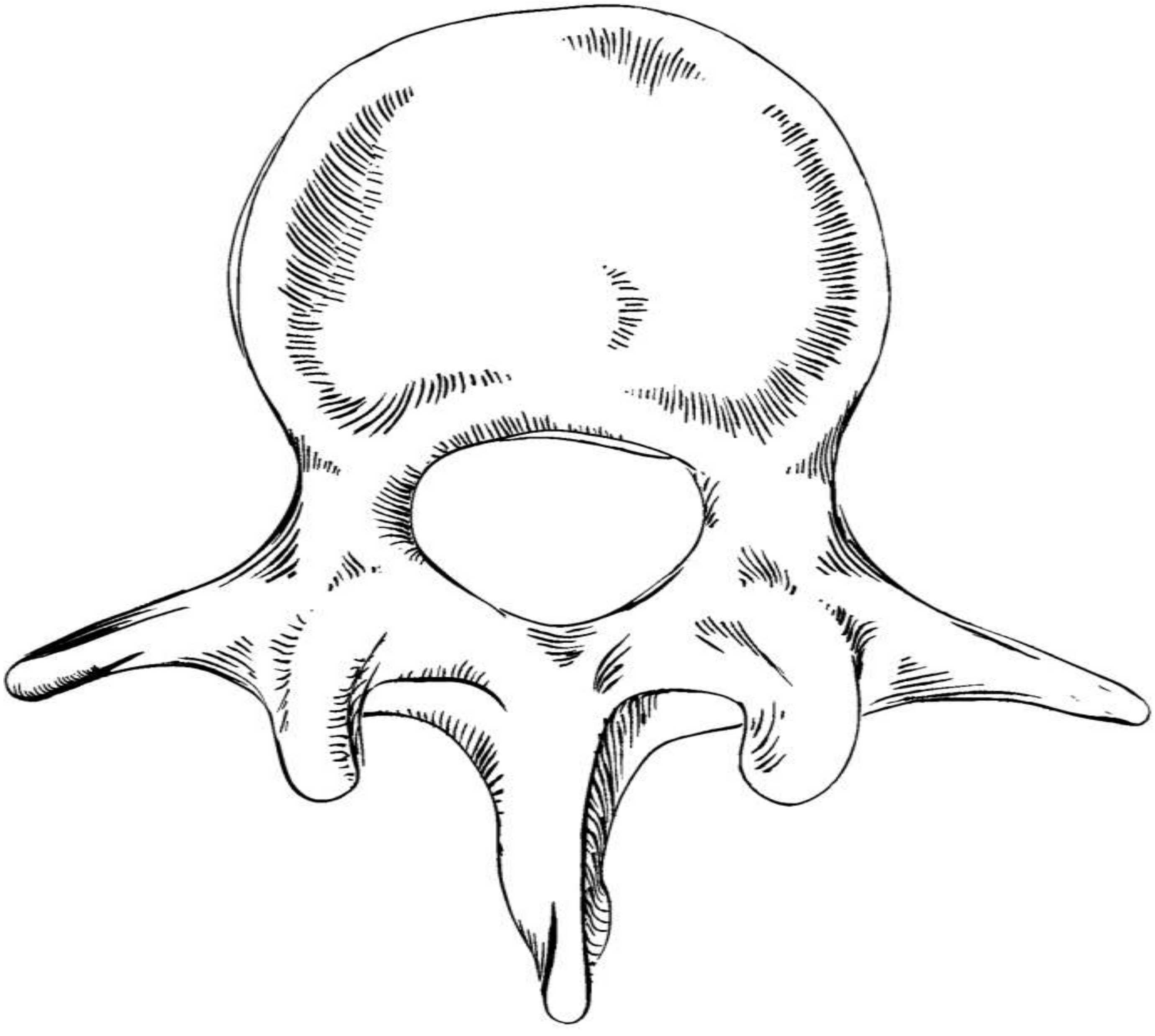
Interaction

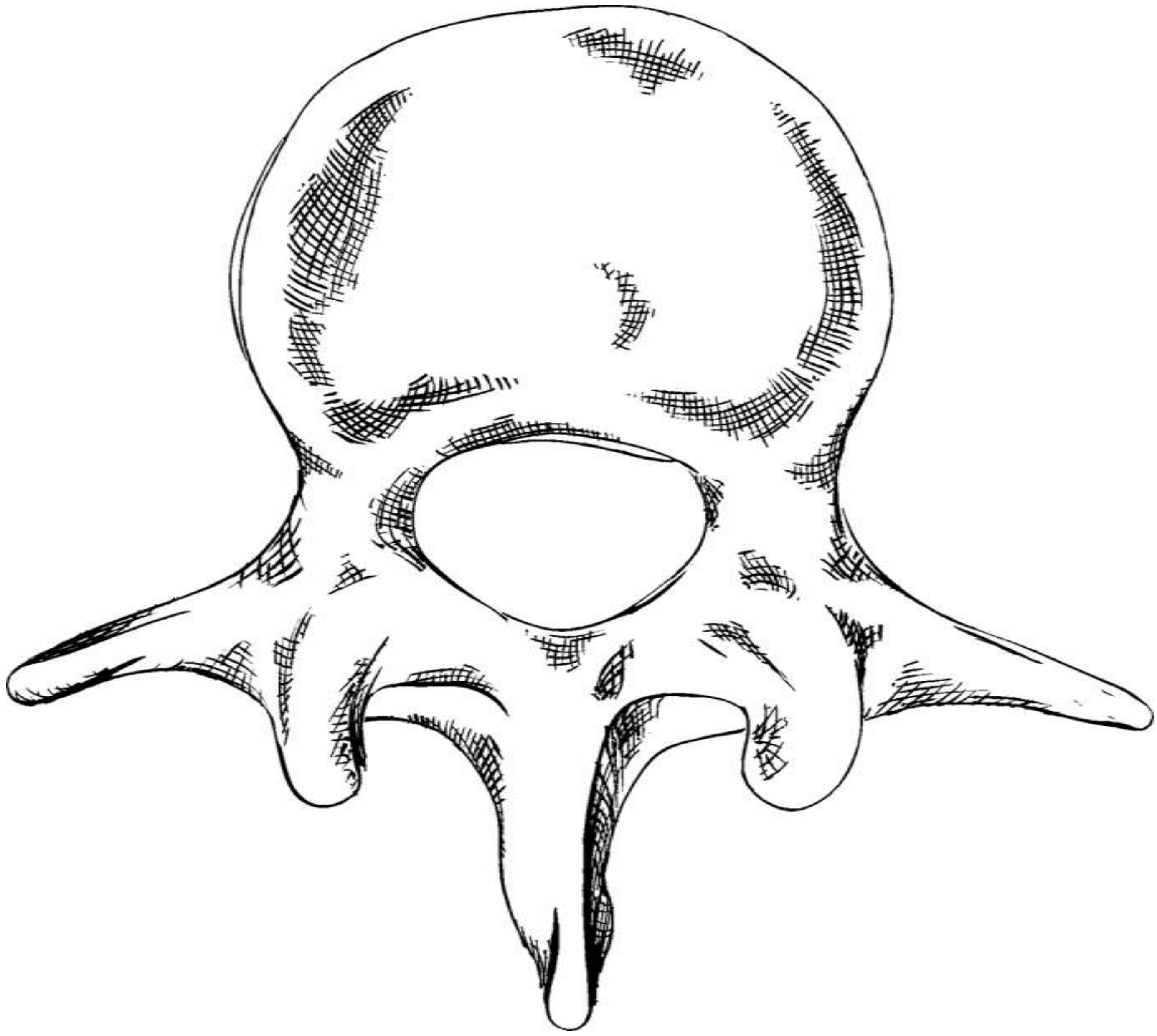
interaction

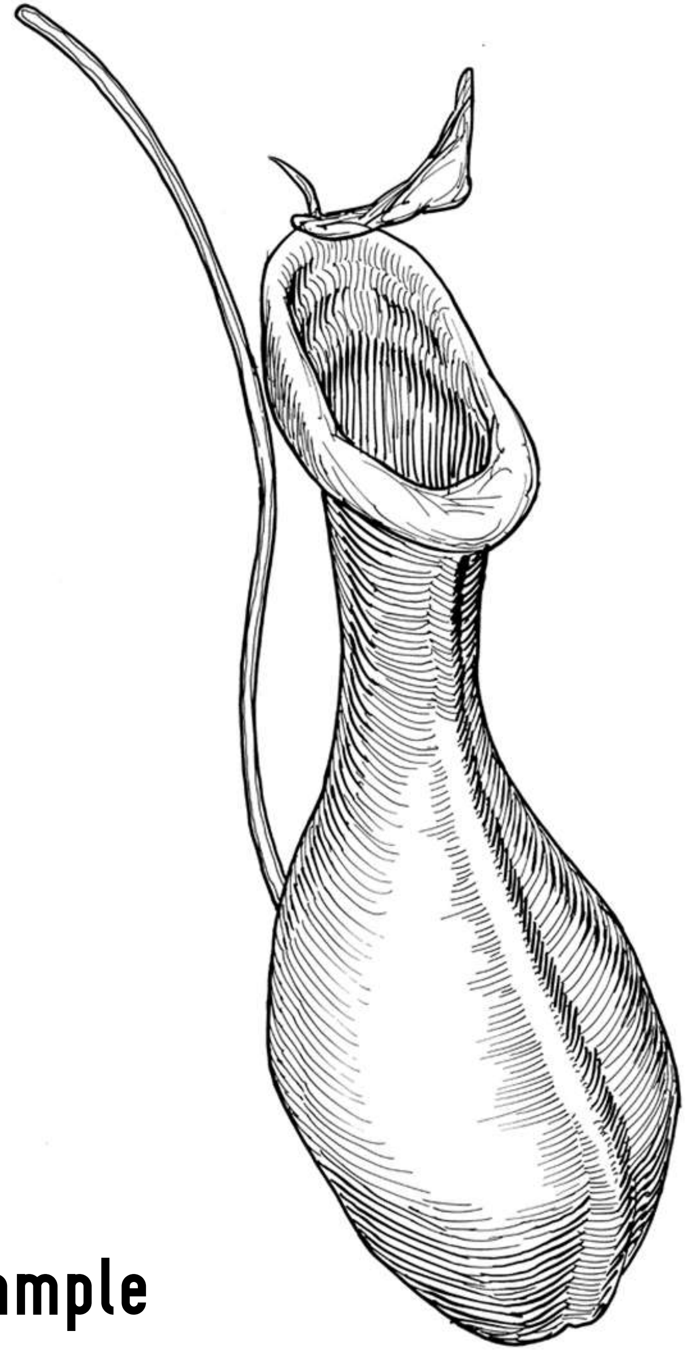
results



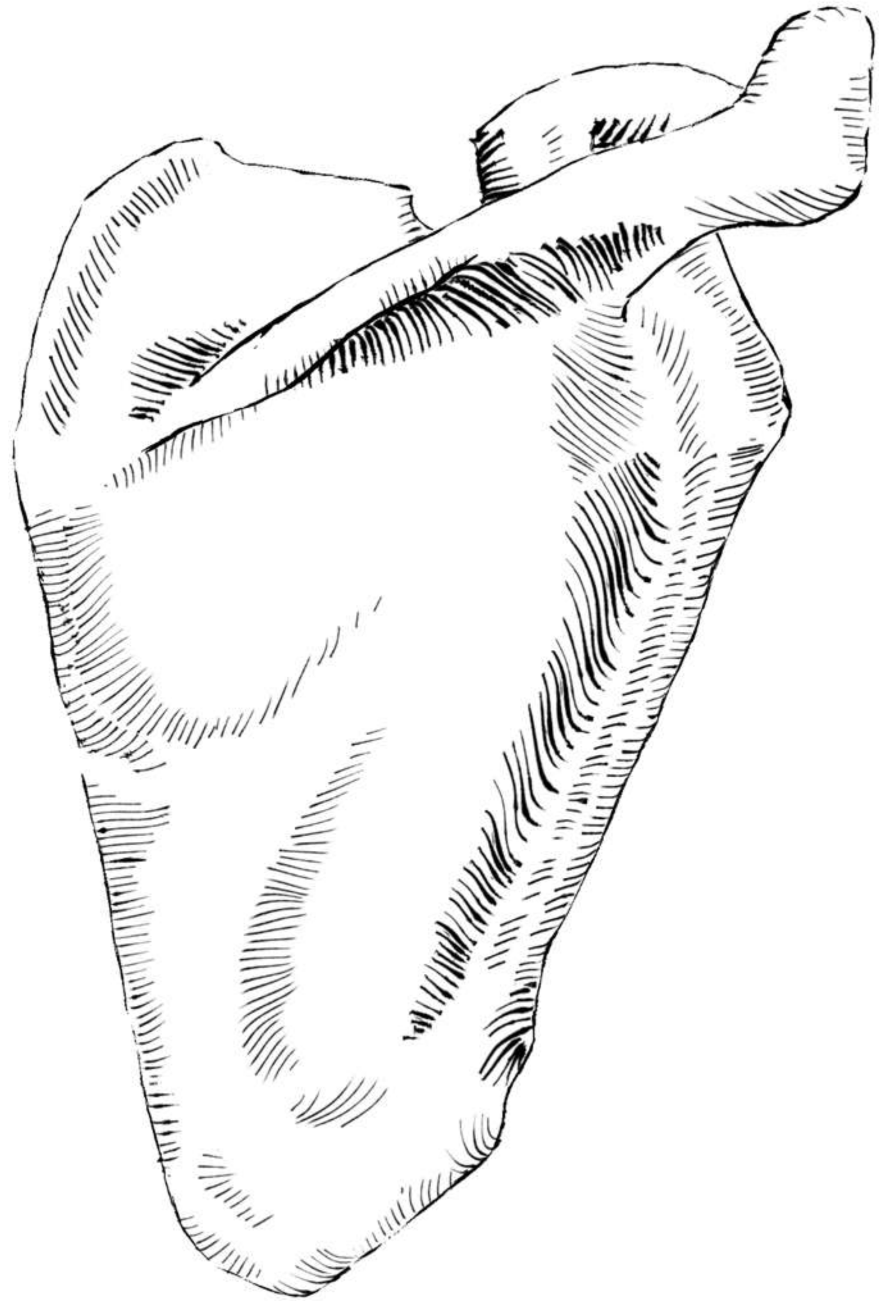


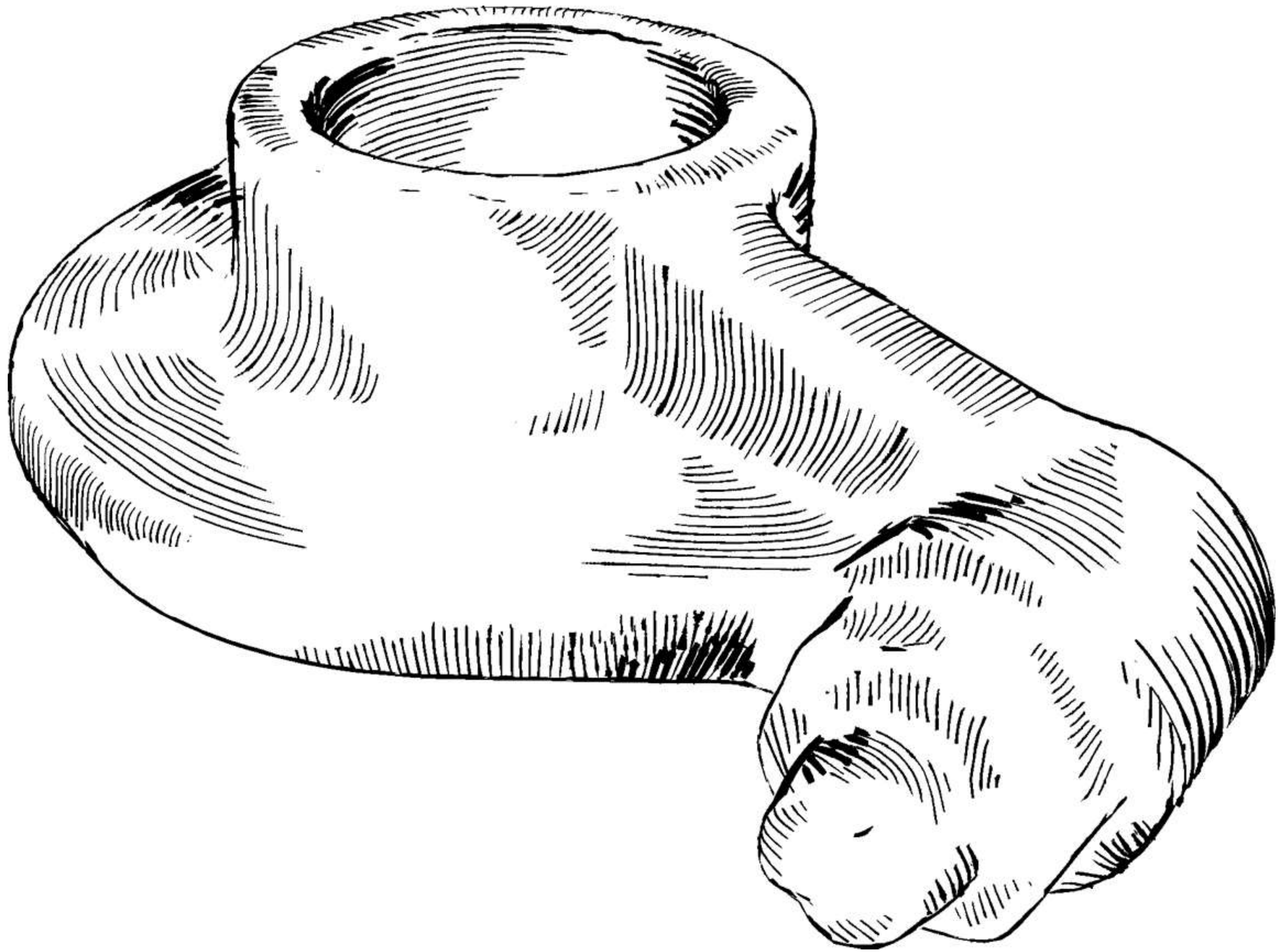


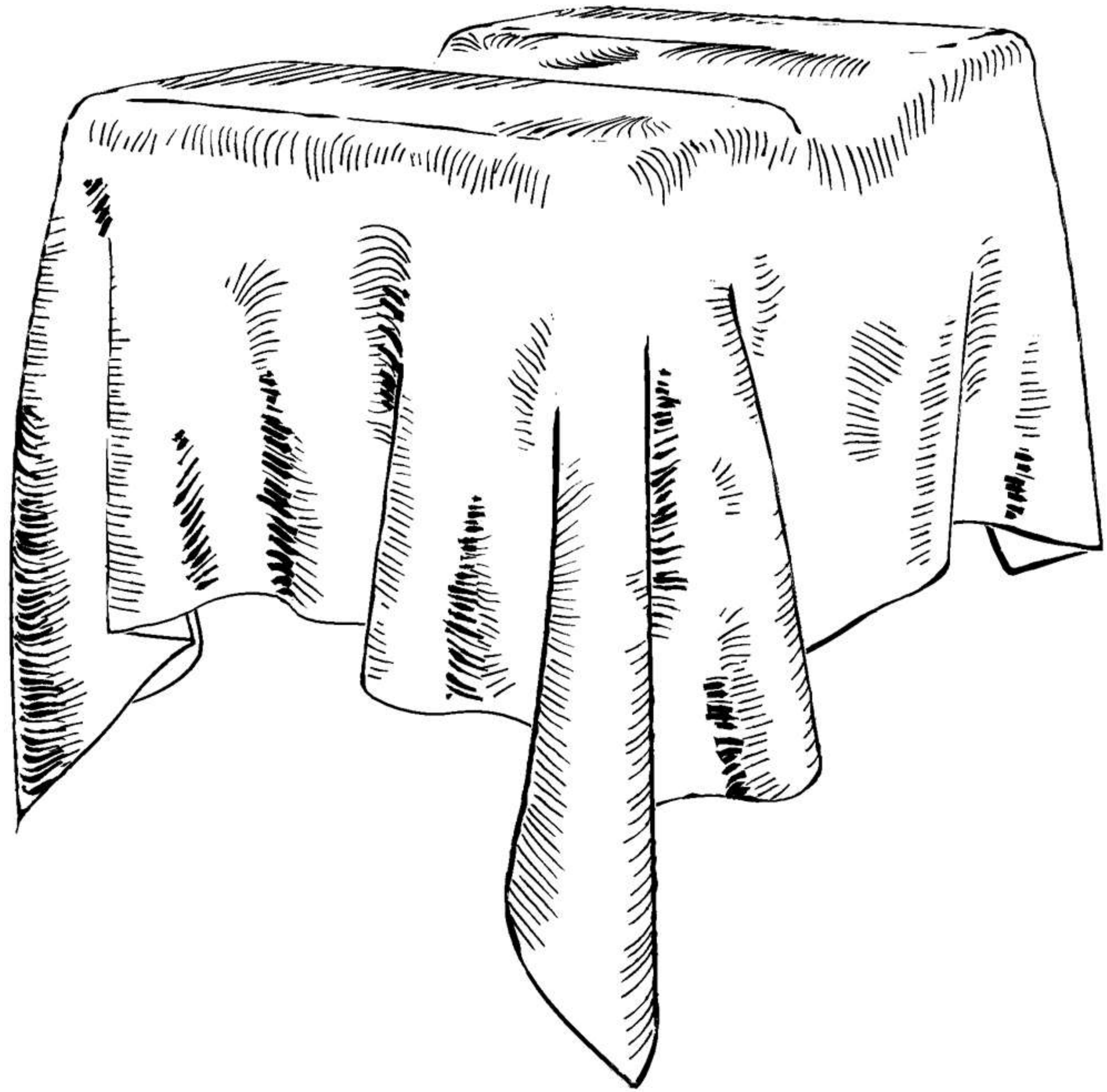




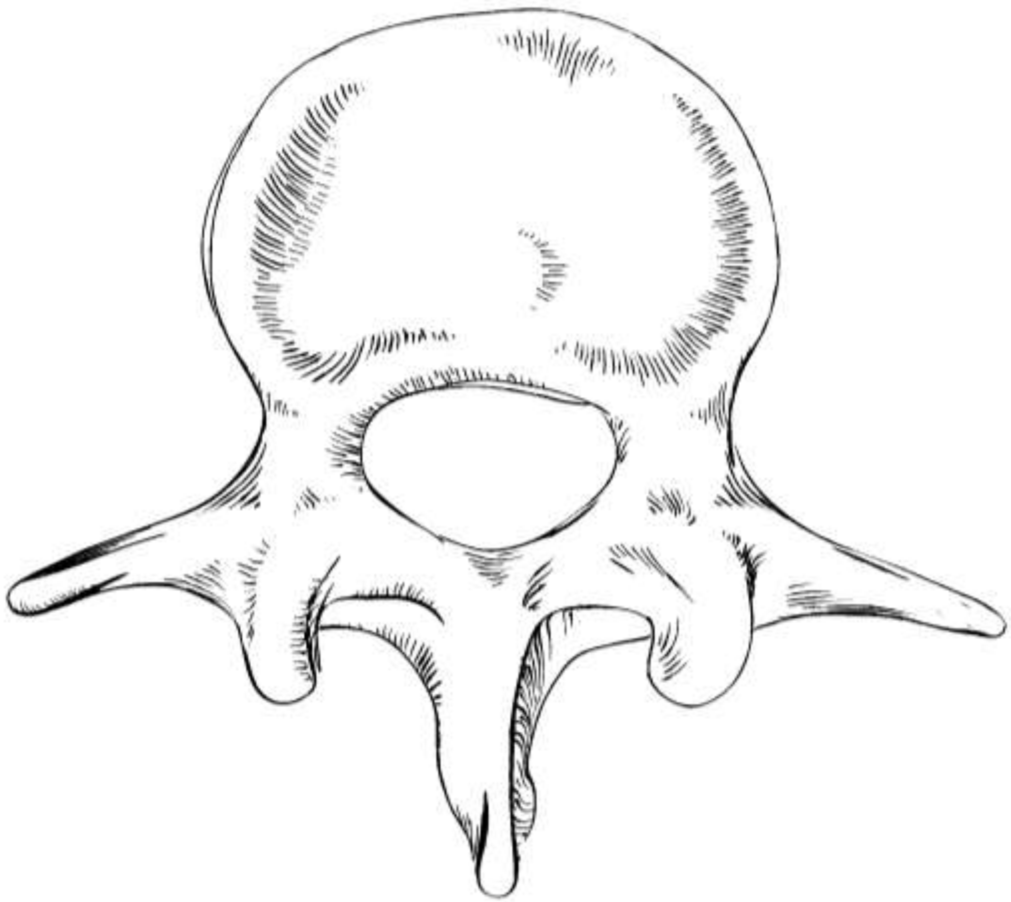
different example

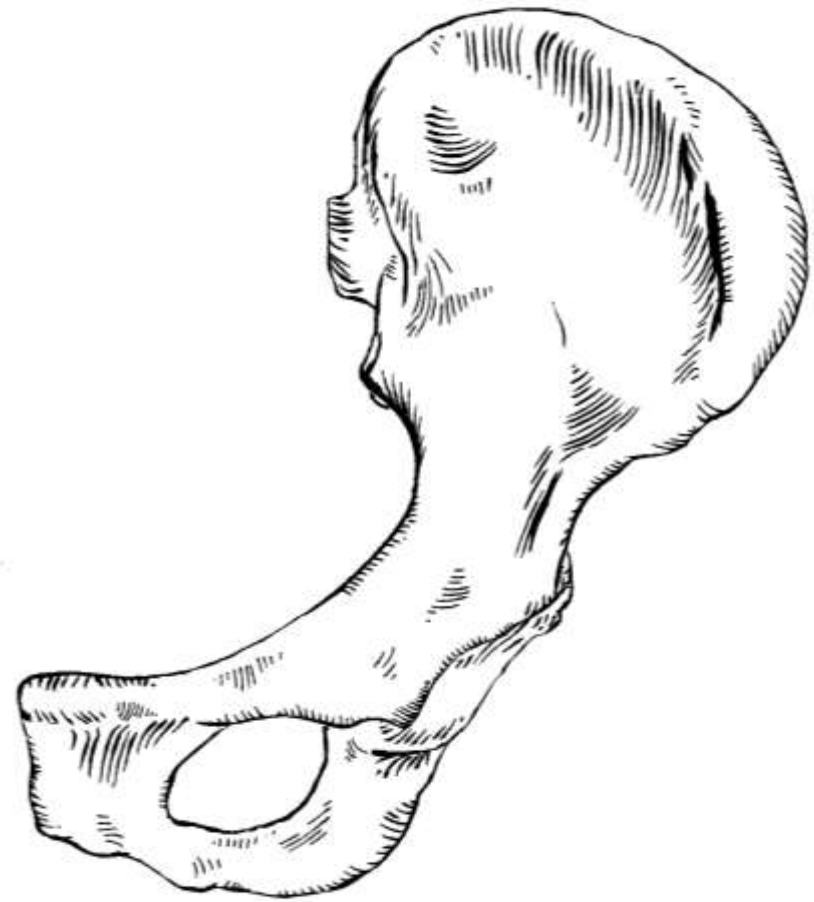






Results





see two hq examples in the Expressive Visual Showcase

Animation

Limitations

- Fully automatic style transfer is inaccurate
- Interaction is required
- Interaction time of 5-10 hours

→ improve machine learning

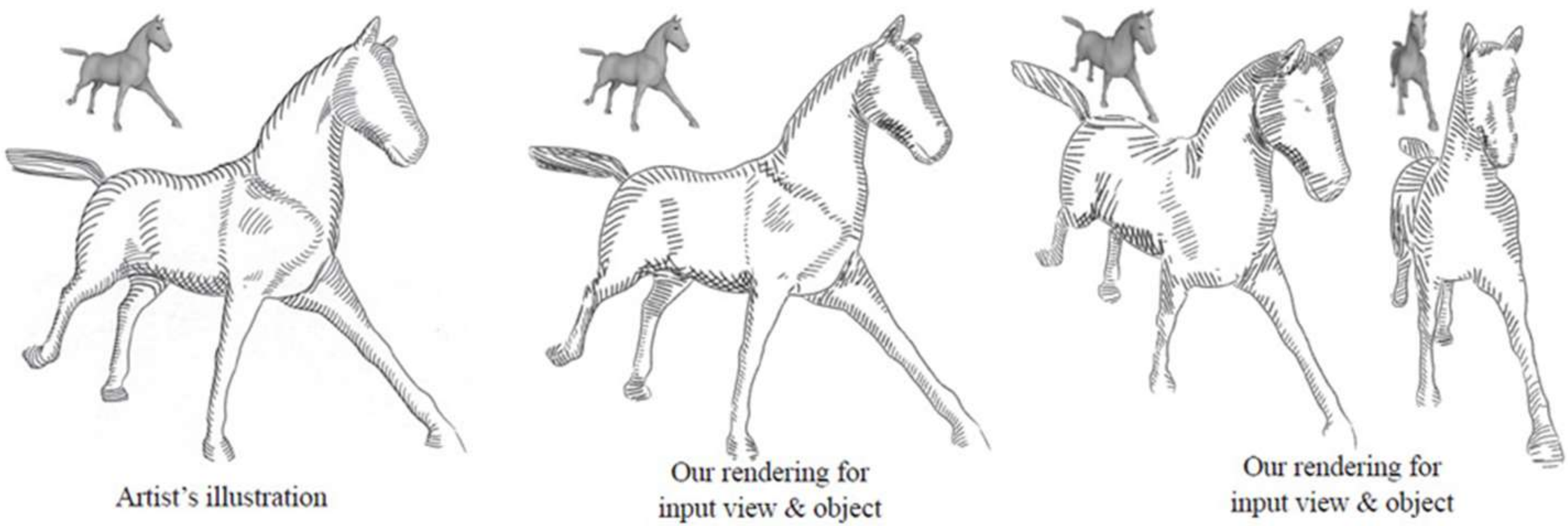
Ideas for Improvement

- Use a larger set of features
- Use more advanced machine learning methods
- Use more examples for learning
- Extend interaction capabilities

Comparison to [Kalogerakis et al. 2012]

"Learning Hatching for

Pen-and-Ink Illustration of Surfaces"



Comparison to [Kalogerakis et al. 2012]

our technique	[Kalogerakis et al. 2012]
analytic representations of drawing elements	pixel-based
interactive	static result
faster synthesis (20 sec)	slower synthesis (30 min)

Comparison to [Kalogerakis et al. 2012]

our technique	[Kalogerakis et al. 2012]
less accurate style transfer	accurate style transfer
limited # of features (18)	many features (1204)
fixed viewpoint	arbitrary viewpoint
interactive input possible	automatic

Conclusion

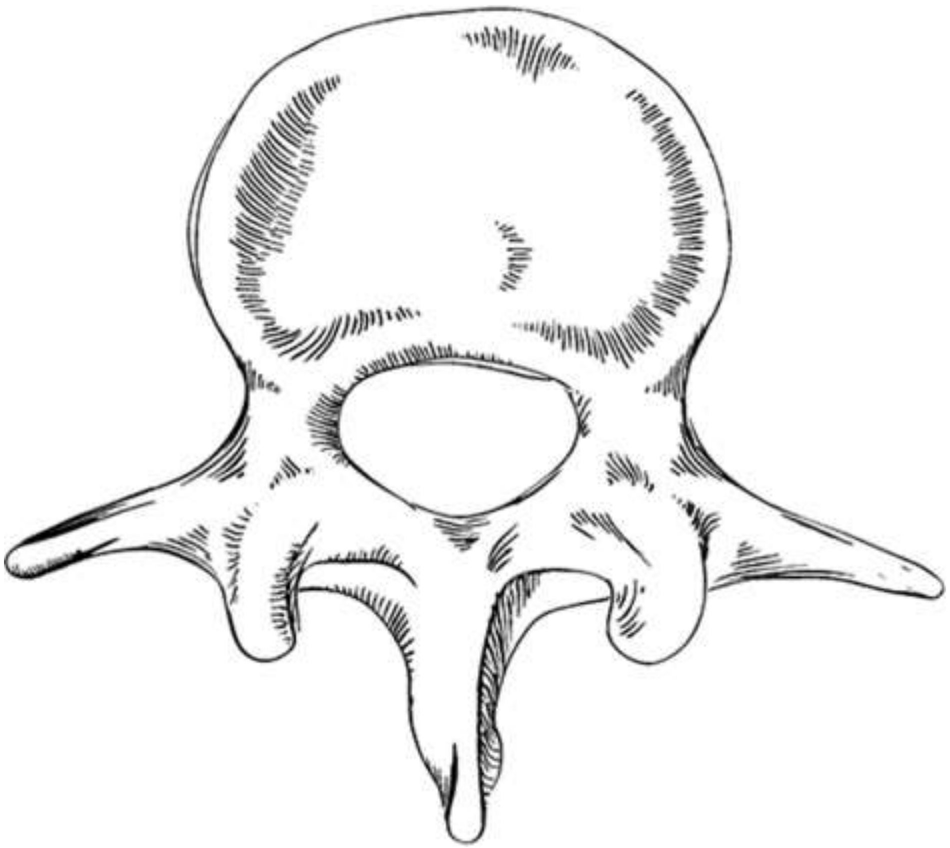
- Combine interactive and example-based functionalities
- Twofold influence of human virtuosity on the rendering result (interactive and through the examples)

→ improvement of aesthetic quality

Conclusion

- Learning and reproduction of a complex drawing style
- Improved automatic style transfer would be beneficial
- Interactivity is attractive for creatives as well as for people with no background in hatching

thanks – questions?



http://tobias.isenberg.cc/VideosAndDemos/Gerl2013IEH