43075-01 Probabilistic Shape Modelling

Lecturers
Dr. Marcel Lüthi ⟨marcel.luethi@unibas.ch⟩

Introduction	03. Mai 2022
Discussion	10. Mai 2022

Exercise 7 — Shape reconstruction

Project preparation

7.1. Theory and implementation in Scalismo

Work through Week 3 of the probabilistic fitting online course.
(`https://shapemodelling.cs.unibas.ch/probabilistic-fitting-course/`)

Make sure you understand both the theory, and its implementation in Scalismo. Write down any questions you have. We can discuss them on Tuesday.

7.2. Data

Download the fragments `bone-fragments.zip` from the Adam workspace:
`https://adam.unibas.ch/goto_adam_fold_1396379.html`.
The file contains 10 bone fragments of different size of 10 different femur bones.

7.3. Fitting a shape model to bone fragments

Week 3 of the online course discusses how to set up a method for shape model fitting using the Metropolis-Hastings algorithm. The tutorial addresses the case, where corresponding points on the shape model and the target are known. Your task is to adapt the code such that it can fit the given bone fragments. This can be achieved by implementing a new `Evaluator` for the likelihood, which can deal with points that are not in correspondence, by making the probability of observing a deformation dependent on the distance between points on the fragment and their closest points on the model, instead of the corresponding points. Use the fitting pipeline to obtain reconstructions of the given bone fragments.
Think about the following questions:

- How do you specify the prior distributions? How can you find out if the values you choose are reasonable?
- How do you need to adapt the proposals (and their parameters)?
- Visualize the reconstructions for some randomly selected samples. Does the variability in the posterior samples reflect the variability you would expect?

A note on the implementation

A likelihood evaluator that works with the full model rather than only the corresponding point, is computationally expensive. The more vertices the reference mesh of the model has, the heavier will be the computations. In Metropolis-Hastings sampling it is important to be able to quickly draw many samples. Scalismo therefore provides functionality to downsample meshes and models on the fly.
A model can be downsampled as follows:

```
val shapeModel: PointDistributionModel[_3D, TriangleMesh] = ???
val mesh : TriangleMesh[_3D] = shapeModel.reference
val downsampledMesh = mesh.operations.decimate(targetedNumberOfVertices = 500)
val downsampledShapeModel = shapeModel.newReference(
    newReference = downsampledMesh,
    interpolator = TriangleMeshInterpolator3D()
)
```

This downsampling operation can be performed directly in the evaluator and should greatly speed up the computation. Make sure that downsampling is only performed once when the evaluator is constructed and not in the method `logValue` of the evaluator. Meshes with around 500 points should approximate the surfaces well enough for computing the likelihood.