Max Planck Institute for Informatics & Saarland University
MPII D4 :: Computer Graphics
Geometric Modeling, Summer 2018, 108963
Rhaleb Zayer, Hans-Peter Seidel

max planck institut informatik

UNIVERSITÄT DES SAARLANDES

Assignment No. 06

Due: June 11, 2018

Task 6.1: Approximation of noisy input data **20 Points**

The following scenario is rather common in the industry and to some extent also in research.

You are an algorithm engineer at a larger graphics-oriented company. You work in a team with hardware and software engineers, artists, technicians, managers, and so on. Your product manager comes to you with the following problem:

> *The hardware guys developed this new device which records line-type data. We know it's supposed to be a smooth curve, but the data is noisy and obviously not editable in a convenient manner. So, our artists cannot work with that data directly. We need you to develop an algorithm that converts the noisy input data into something that our artists can use for geometric modeling. I'll send an email with the details later.*

In her email, your product manager reiterates the task: Develop an algorithm that approximates a (noisy) polyline with a smooth curve. She also gives a couple of side constraints, some of them are even conflicting:

- The smooth curve needs to be editable by artists. This means, it needs to be one of the following: a Bézier spline or a B-spline.
- The smooth curve needs to be at least C^1-continuous everywhere!
- The splines must have cubic degree, since the company has already nice tools to work with cubic splines.
- The smooth curve needs to be editable by artists *in a nice way:* fewer control points are easier to handle for the artists.
- The smooth curve shall approximate the input *as closely as possible:* more control points can provide better approximation.

Your work will be given to software engineers, who will incorporate your algorithm into the production code. You are asked to provide the following to the software engineers:

a) *(10 points)* A description of your algorithm in *one* page as a PDF file (you are free to design the page and to use multiple columns if needed). To keep it concise, this description may refer to proofs and mathematical properties given in the lecture slides. In detail:

   - One page.
   - PDF file.
   - **Do not write your names anywhere on that page or in that file.** We will have an anonymous peer reviewing in the following exercise.
   - Bring the file to the interview.

b) *(10 points)* A fully working implementation in matlab. Your code needs to be able to do the following:

- Load a text file with the coordinates of the polyline. Some example data is given. Some other data will be loaded during the interview.
- Compute everything that is necessary to create the smooth curve as desired by the product manager. You are welcome to add parameters to the user interface, if your algorithm needs them.
- Show the input polyline and the output smooth curve in different colors.
- Show the control points and the control polygon that define your smooth curve.

Note: After submission, you are requested to peer review the results of your fellow students. You will be given all PDF files from (a), and have to pick the best, the second best, and the third best solution; excluding your own, of course. We will assemble this into a global ranking, which we will announce in the lecture and/or tutorial. Most importantly, the first three solutions in the global ranking will get 6/4/2 bonus points.

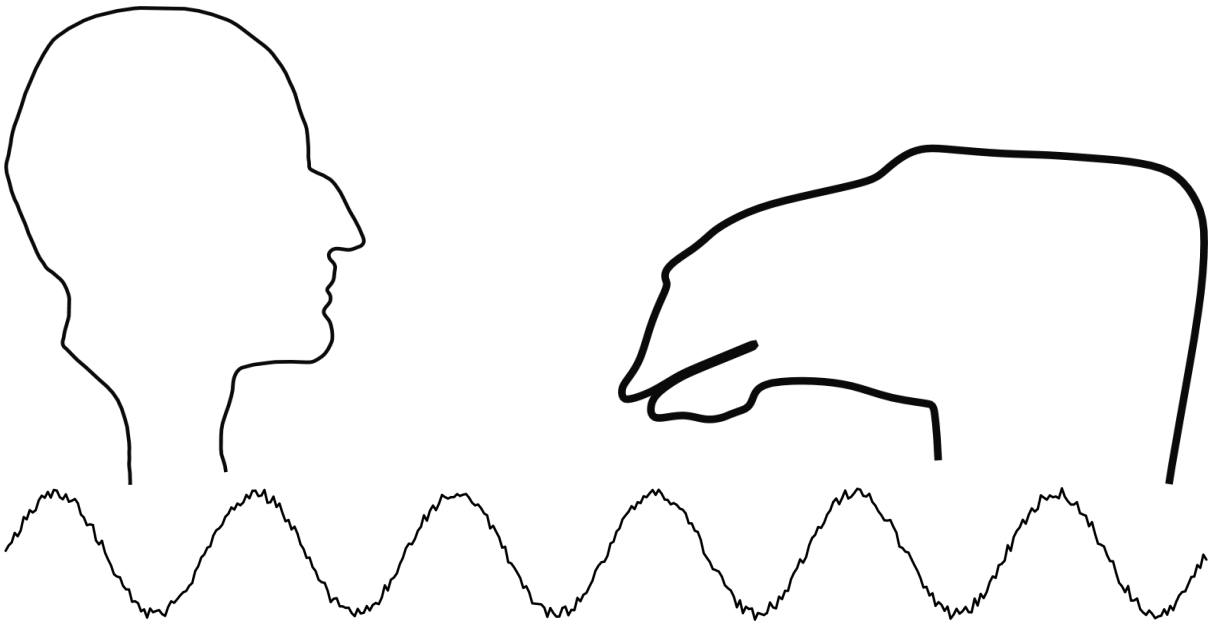

Figure 1: Some of the example data sets given for testing: the silhouette of the Max Planck statue, a randomized sine curve, and an intersection curve through a mesh representing a camel.