AP6

Murmani Akhaladze
TarnNished

1. Introduction

For AP6 I decided to reconstruct a 3D mercury thermometer bulb using only a single 2D image of it. The idea is very similar to AP5, but now instead of drawing letters I am working with an actual physical object that has axial symmetry. Because the thermometer bulb is basically rotationally symmetric around the vertical axis, we can detect its side profile in the image and then rotate that curve in 3D to rebuild the whole shape.

The main goal of the assignment was to load a real image, detect the edges of the bulb, approximate the outline with a spline, and then revolve this spline around the axis to produce a 3D surface. After that we also have to compute the volume using the standard solid-of-revolution formula. So the workflow is half computer vision, half numerical modelling, which honestly made the project more fun than I expected.

2. Model & Approach

To start, I needed a clean 2D image of a thermometer where the bulb is clearly visible and the object is vertically aligned. The first image I tried didn't work well because the whole thermometer was tiny inside the picture, so I cropped the photo and kept only the bulb area. With the cropped image, Canny edge detection gave a really nice and clean outline for the bottom part of the thermometer.

Once I obtained the binary edges, I extracted the radius of the bulb for every horizontal row. Basically for each z coordinate (which is vertical height), I scan to the right to find the first edge pixel. The distance from the axis of symmetry to that pixel becomes the radius $r(z)$. After collecting all these (z, r) points, I fit a natural cubic spline that represents a smooth profile curve of the bulb.

The 3D reconstruction part is quite straightforward: the spline is rotated around the z axis, creating a full surface of revolution. This is done by sampling the spline on a dense grid and then computing the rotated coordinates (x, y, z) using $x = r\cos\theta$ and $y = r\sin\theta$.

3. Experiments

I tried a couple of different smoothing levels and Canny thresholds to see how sensitive the reconstruction is. If the thresholds are too high, edges disappear and the spline becomes too flat. If they are too low, the code picks up reflections inside the glass tube which messes up the profile. The cropped image I used in the final version actually worked almost immediately without much tuning.

Then I tested different spline resolutions (more or less points in the z direction). Using too few samples makes the 3D mesh look a bit blocky, and using too many doesn't change much besides slowing down plotting a little. The profile curve itself stays stable regardless of sampling, which is expected because the natural spline smooths the raw points.

The volume calculation is done numerically with the formula

$$V = \int \pi r(z)^2 \, dz,$$

which is basically the classical method for solids of revolution. The result is in pixel units, since the picture doesn't have any real-world scale, but it still demonstrates the method works.

4. Conclusions

This AP helped me understand how parametric splines can be used not only for "drawing" shapes like in AP5, but also for real 3D reconstruction tasks. Even from a single 2D image you can rebuild a full 3D object, as long as it has some symmetry. The thermometer bulb was a good choice because the edges are clean and the geometry is smooth.

The workflow I used — crop the image, detect edges, extract the profile, fit a spline, revolve it, compute volume — turned out to be a pretty general recipe for a whole category of similar problems. In the future I could try other symmetric objects, maybe even more complex ones, and the same approach should still work with small modifications.

Overall, the experiment confirmed that spline-based reconstruction is accurate, easy to implement, and strongly depends on good image preprocessing. If the initial image is clean enough, the rest of the pipeline works almost automatically.