Assignment 3 (Practical)

Parts of this assignment will be automatically graded, to receive full points you will need to complete the functions listed in the following directory structure:

- assignment3
  - matrices
    - ___init___.py
    - differential_coordinates.py
      - `triangle_gradient()`
      - `build_gradient_matrix()`
      - `rotation_component()`
      - `build_mass_matrices()`
      - `build_cotangent_matrix()`
    - test.py
  - deformation
    - ___init___.py
    - deform.py
      - `gradient_deform()`
      - `constrained_gradient_deform()`
    - test.py
  - extension
    - ___init___.py
    - *(You can add more files as needed!)*

In each file, the necessary function stubs are already present, with some documentation explaining their purpose and expected behavior. More details are provided in the assignment packet itself, as well as some useful implementation hints. *Make sure not to modify these function signatures, as they are expected by the grading script.*

If you're not sure where to start, look for the `# TODO` comments!

You may modify any other functions, and writing your own helper methods will likely make the tasks easier.

Unit Tests

For part 1 (matrices), we will also be looking for high-quality unit tests. You should consider some invariants you know will hold for a correctly constructed G, M, M_v, or S matrix. It might also be helpful to work out the expected values by hand for a very small mesh.

It's a good idea to set up your unit tests early on in development; part 2 is much easier if you can be certain your gradient and mass matrices are correct!

Using the UI

Part 2 (deformation) includes some UI code which makes use of the functions in deform.py.

Global Deformation

The Mesh Gradient Deformation operator is accessible only in Object mode, and affects the whole mesh (Fig. 1).

This operator allows you to set up a transformation matrix A and then invokes your `gradient_deform()` function to reshape the currently selected mesh, as shown in Fig. 2.

Local (constrained) Deformation

The Constrained Gradient Deformation operator is accessible only while in Edit mode, because it requires you to select individual faces on the object. It can be accessed from the Mesh dropdown (Fig. 3).

When the constrained version of the operator is invoked, selected faces are modified using your transformation matrix A, but other parts of the mesh are left unchanged, as shown in Fig. 4.

Extension

Part 3 of this assignment is a fully custom extension. We have provided a small amount of boilerplate in `__init__.py` and `extension/__init__.py`, but this is a very open-ended task. You should implement your own Operators or Panels to complete some task, a few ideas include:

- a brushes tool for smoothing surfaces using Laplace Coordinates
- a shape editing tool that allows users to specify constraints on vertex positions
- a mesh smoothing tool that offers implicit Laplace smoothing
- a tool for hole filling in meshes
- a tool that computes geodesic distances using the heat method

But this is far from an exhaustive list, and we encourage you to come up with something you think would be interesting! Feel free to adapt UI code from this assignment or previous parts of the course, or take inspirations from Blender's provided templates.

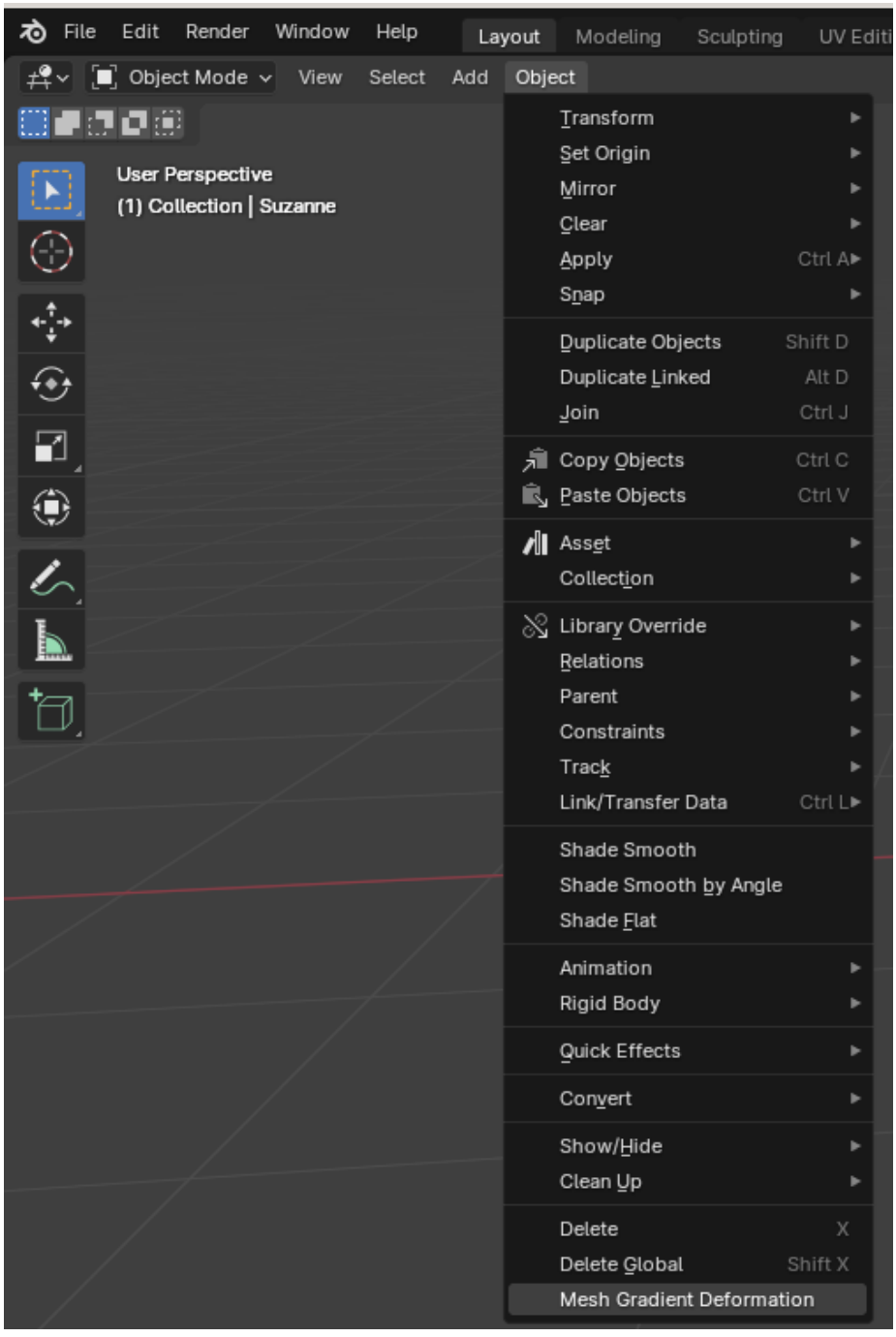

Figure 1: The Mesh Gradient Deformation operator (bottom right) can be used when in Object mode (top left).

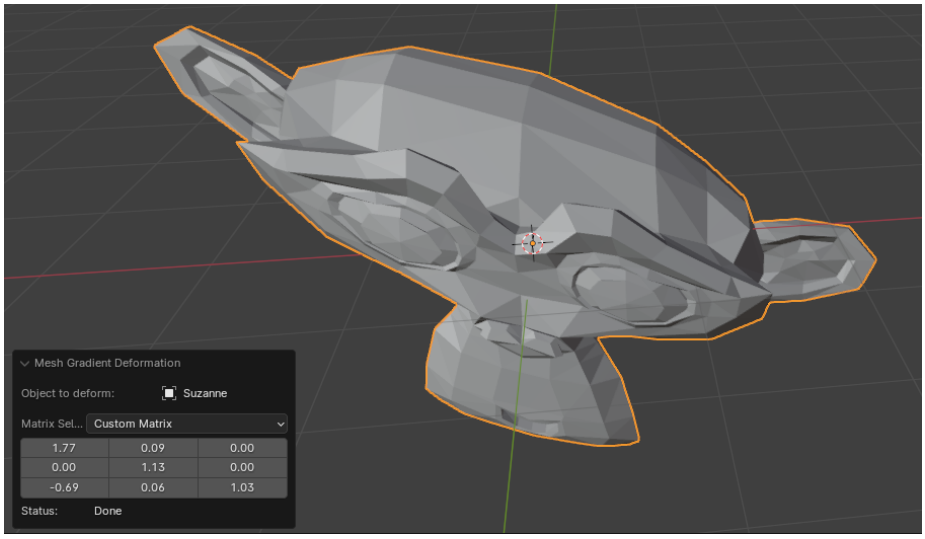

Figure 2: Suzanne the monkey, stretched and skewed by the global gradient deformation operator.

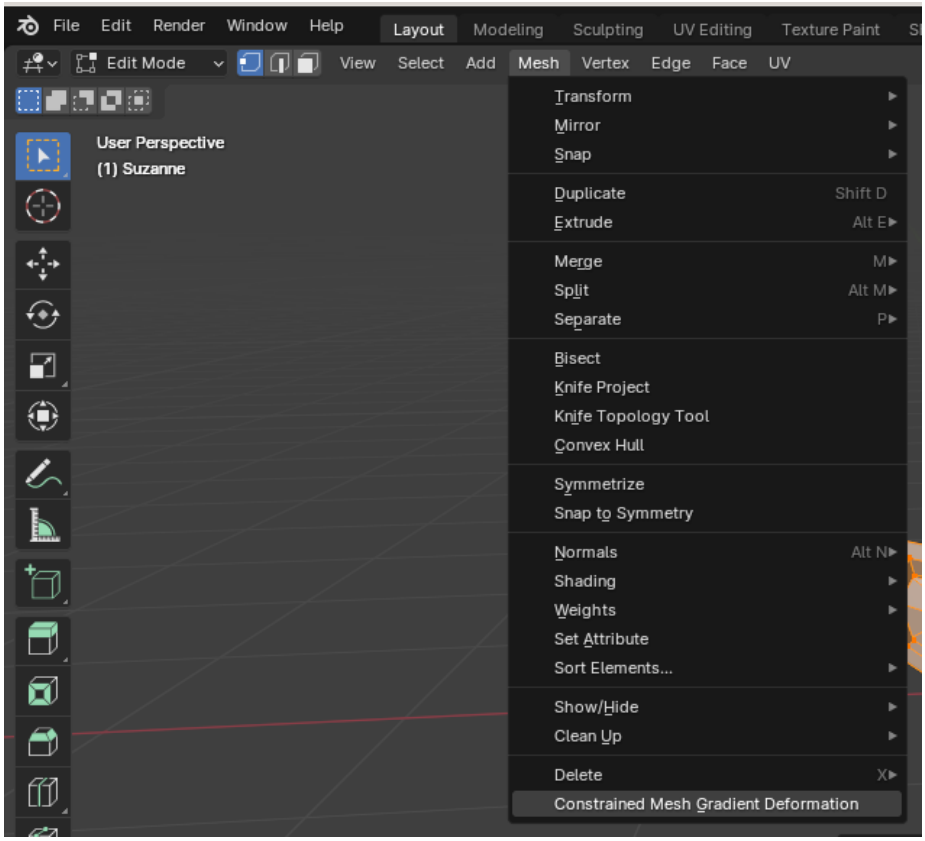

Figure 3: The Constrained Gradient Deformation operator (bottom right) can be used when in Edit mode (top left).

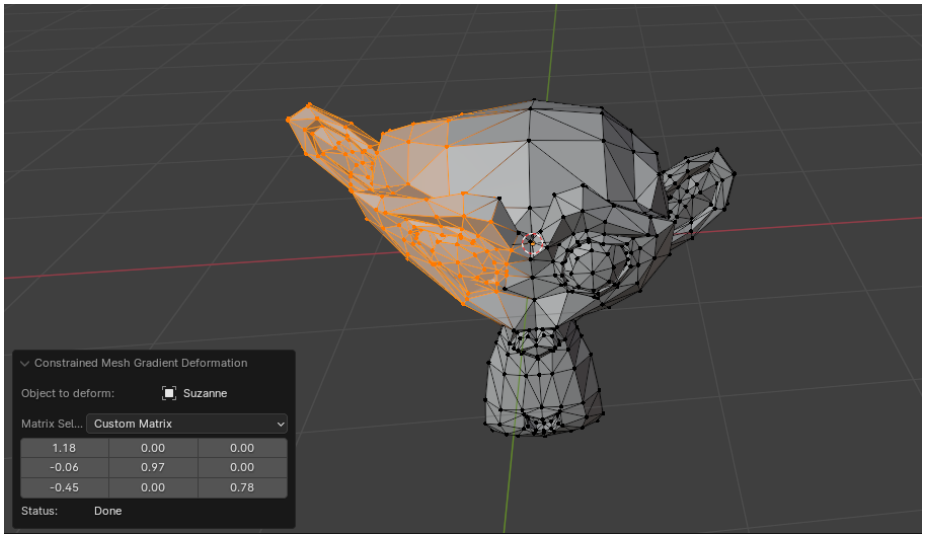

Figure 4: Suzanne the monkey's ear, stretched and skewed by the constrained gradient deformation operator.