Geometric Data Processing (CS4540)
TU Delft, EEMCS, Department of Intelligent Systems
K. Hildebrandt

2024-2025 Q. 4
Due: 20-05-2025

# Practical Assignment 1

(Mesh and Surface Analysis and Surface Registration)

---

## Task 1 (Mesh and Surface Analysis)

Create Blender panel add-ons for mesh analysis that compute

1. the genus of the surface,
2. the volume enclosed by a surface (without boundary loops),
3. the number of connected components of the mesh, and
4. the number of boundary loops.

## Task 2 (Rigid Registration)

Create a Blender operator add-on that offers a basic Iterative Closest Point (ICP) registration of two user-selected meshes $P$ and $Q$ using the point-to-point and point-to-plane distances.
As discussed in the lectures, the basic ICP algorithm iterates the following steps until convergence or stopped:

1. Select a random set of $n$ vertices $\{p_1, p_2, ..., p_n\} \subset P$.
2. For every $p_i$, find the closest vertex $q_i$ in $Q$. A brute-force search will work, but we recommend using `scipy`'s `KDTree`, which makes the nearest-neighbor lookup more efficient.
3. Compute the median distance of the set of pairs $S = \{(p_i, q_i) | 1 \leq i \leq n\}$. Remove all pairs $\{p_i, q_i\}$ whose distance is greater than $k$ times the median distance from $S$.
4. Compute the optimal rigid transformation for the set $S$ (from which the pairs have been removed in Step 2) and transform the mesh $P$.

Extend the basic implementation by offering the user to select the point-to-plane distance instead of the point-to-point distance in step 4.

## Task 3 (Variants and Experiments)

Implement variants of the basic ICP algorithm. Examples of extensions you can consider are: an alternative sampling method such as farthest-point sampling or normal-space sampling, including additional information such as normal vectors when finding the matching points, an alternative culling scheme (step 3), or computing not only a rigid transformation but also scaling.
Evaluate the ICP algorithm by performing experiments that examine parameter settings and the variants you implemented. Examples of parameters to examine are: $n$, the number of points sampled in step 1, and $k$, which is used in step 3. Examples of variants you can investigate are: the variants you implemented for this task, using the point-to-point distance versus using the point-to-plane distance, and using a kD-tree versus using a brute-force search to find the closest points.
Write a report that describes and illustrates

- the variants of the basic ICP algorithms you implemented (please also specify the Python files) and how they can be used in Blender,
- the experiments you conducted and the conclusions you draw.

## Implementation Hints

- We provide boiler-plate code for both tasks in the zipped file `assignment-1-distribution.zip`. Upon decompressing this file, you can find more in-depth information about the code's structure, which functions to complete, and the test to perform in the `README.pdf` files. Also, consult the tutorials' slides for additional information about the assignments' and code structure.
- For Task 1, we recommend the implementation order `genus` → `volume` → `components` → `boundaries` as each comes with progressively less guidance.
- Each task has its own directory. Add your tests in the `test_task.py` file within each directory. Implement unit tests with synthetic data for the functions you create to prevent failing due to improperly handled corner cases. The file `mesh_properties.csv` provides some expected values for different objects provided in the data directory. Use these to validate your methods' correctness.
- Many of the provided functions will be automatically tested, these are marked with a comment. **Do not change the signatures of these provided functions or the directory's structure.**

## Policy on use of Generative AI

Generative AI tools may not be used for Tasks 1 and 2.
For Task 3, any type of generative AI tool may be used for brainstorming/idea generation, writing quality enhancement (improve readability and language of the work), and general formatting of references. The use must be declared and explained in the report. For other purposes including coding, generative AI tools may be used for Tasks 3, the use must be declared and explained in detail in the report. It must be clear what work can be attributed to the group and what to the AI. Only your own contribution is evaluated, not the work of AI.

The undeclared use of AI tools, as well as incomplete or misleading explanations of the use of AI tools anywhere in the course, is fraud.

## Required deliverables on Brightspace

- For Tasks 1 and 2, complete the provided boilerplate with your implementation and upload the zipped directory (in the same directory structure as it was provided). Include also the implemented variants (Task 3) in the zip file.
- The report should be one PDF file.

Deadline: May 20, 20:00.