

# HW on GCV course by Aleksandr Nevarko

aleksandr.nevarko@skoltech.edu

March 2021

Homework on course Geometric Computer Vision contained 6 TODO's. Here I will describe my solutions for them. My solution is available on *Github*: <https://github.com/AlexanderNevarko/gcv-hw1>.

1. `get_view` function:

- `pose_i` contains the instance of the `CameraPose` class for transformation of points between camera and world coordinate frames.
- `imaging_i` contains `RaycastingImaging` instance to transform from picture uv coordinates to 3d camera frame coordinates.
- `points_i` - is the result of transforming from picture distance points to 3d points in world coordinate frame.

2. reprojection:

Using `CameraPose` instance (`pose_i`), that knows the extrinsic parameters of the camera of depth map `i`, we transform `j`-th 3d points to `i`-th camera coordinate frame.

3. `cKDTree`:

- `cKDTree` indexes the pixel points coordinates of the `i`-th depth map in some way, that after that it is easy to find distances to them.
- We then query the `kdtree` with `j`-th points to find nearest neighbours and distances to them from the `i`-th picture space.

Next TODO's were completed, but commented then (you can still see my solution). I commented them, because the following **python** *for* cycle was very slow and ineffective.

4. lines 104-107:

- `distances_to_nearest` contains the distance for each point from `j`-th space to the closest point of `i`-th space. I used the calculated `distance_2d`, that was returned from the `kdtree` query to transform it from 2d distance to 3d and not to redo calculations.

- `interp_mask` is the mask, that shows what points have a neighbour, that is not further, than distance interpolation threshold.
5. lines 126-127:
- I decided to take away from the *for* cycle everything except interpolation, so now *points\_from\_j\_nns* has shape (*n\_points*, *nn\_set\_size*, 3 - dimensionality).
6. *for* cycle (128-138):
- The cycle runs only over *True* cells in *interp\_mask*.
  - Interpolator is created with coordinates of neighbours of each interpolatable point from j-th space.
  - Distances values are calculated via interpolator for each reprojected from the j-th to i-th space point.