Weighted Perspective-n-Point Solver in Python
Project Management and Software Development for Medical Applications

General Info
Contact Person: Matthias Grimm
Contact Email: grimmma@in.tum.de

Project Abstract
This project works on implementing a weighted Perspective-n-Point solver in Python. It is as simple as that ☺.

Background and Motivation
The Perspective-n-Point (PnP) Problem deals with computing the pose of a camera, given the camera matrix (intrinsic parameters), a set of n 3D point coordinates and their corresponding 2D pixel locations. This is used a lot in augmented reality and computer vision, for example for computing a rigid registration between a 2D image and a 3D model. In a medical context, the 2D image could be an X-ray and the 3D model a CT scan.

While there are many ways to solve this mathematical problem, the student will look at one promising one in particular. This is based on first computing a Direct Linear Transform (DLT) to compute a rough initial pose. Afterwards, this will be fed into a Levenberg-Marquardt Optimizer, which minimizes the reprojection error between the projected 3D landmarks (according to the current estimate of the pose) and the detected 2D pixel locations.

So far, the project is just a re-implementation of existing methods. However, then the students will work on an interesting current research topic, which is the inclusion of weights into the solver. These weights are used to put different emphasis on the points. This is important for real world settings, as the mathematical problem so far assumed that the 2D locations are perfect. In practice however, they are estimated using algorithms, which will contain errors. To overcome this problem, the points can be weighted differently.

Student’s Tasks Description
The students task is to implement a weighted Perspective-n-Point solver in Python. This solver should be initialized using a Direct Linear Transform. Then, this initialization should be used to initialize an iterative optimizer minimizing the reprojection error.

In a second stage, students should adapt their algorithm to allow for weights to be incorporated. These weights should allow to place a different emphasis on different points.

Please send the completed proposal to ardit.ramadani@tum.de, zl.jiang@tum.de, jennart.bastian@tum.de and tianyu.song@tum.de. Please note that this proposal will be evaluated by the BMC coordinators and will be assigned to a student only in case of acceptance.
Technical Prerequisites

Students should be familiar with Python. Furthermore, they should be familiar with Linear Algebra and Optimization. Ideally students are a bit familiar with the perspective-n-point problem (for example if they attended my lecture in the Medical Augmented Reality course).

References
