CS 532: Homework Assignment 4
Due: November 27 , 6:00pm

Enrique Dunn
Department of Computer Science
Stevens Institute of Technology
edunn@stevens.edu

Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment. I will assume that, as participants in a graduate course, you will be taking the responsibility of making sure that you personally understand the solution to any work arising from collaboration.

Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.

Submission Format. Electronic submission on Canvas is mandatory. Submit in a zip file, **a single pdf file containing**:

- the source code,
- brief of explanations of what was done,
- images and screenshots of the model.

Also include the final ply model.

Problem 1. Download the *dancer* dataset from the course web page. The zip file contains images, silhouettes and projection matrices for a single time instant of a dynamic scene. A simple ply file containing 8 points is also included. (Finally, it contains a Matlab script that may be helpful, but is not necessary for completing this assignment.)

The goal is to obtain a voxel-based reconstruction of the scene using the provided silhouettes as inputs.

Part 1. Define a voxel grid with x ranging from -2.5 m to 2.5 m, y from -3 to 3 m and z from 0 to 2.5 m. Set the size of each voxel so that the total number of voxels fits comfortably in the memory of your computer. You can start at lower resolution initially to accelerate development. You may also chose to make the initial volume tighter, but this is strictly optional. In this part, the goal is to estimate which voxels are occupied and which are free space.

Part 2. The goal of this part is to identify surface points that should be included in the output 3D model. You can represent faces of voxels by one point each. The output should be in ASCII ply format. Use the included ply file as a sample for how to create such as file. The number after `element vertex` is the total number of vertices and has to be correct for the model to be displayed correctly. Each vertex is represented by three floating point numbers for the coordinates and three unsigned characters for the RGB colors. In this part, it is acceptable to have all points be of the same color. Models can be visualized using Meshlab, which is available at http://meshlab.sourceforge.net/ and works on Windows, MacOSX and Linux.

Part 3. Try to color the model by selecting RGB values from the images. Using one image per point is perfectly acceptable, but the criteria for choosing it have to be described in the report. It will be helpful to determine a vector for each vertex that points outward from the occupied volume to the cameras.

Acknowledgement. The input images, silhouettes and calibration parameters have been generated and made available by the Morpheo team at INRIA, Grenoble, France.