Technische Universität München
Fakultät für Informatik
Prof. Dr. N. Navab

WiSe 2011/2012
Exercise Sheet 6
January 12, 2012

Exercises in 3D Computer Vision II

In these exercise we will read a report from Wojeciech Matusik (*Efficient View-Dependent Sampling of Visual Hulls*) corresponding to the work he did at MIT around 2000-2001 on image-based techniques for the computation of the visual hull. This report will allow us to go over the idea of shape-from-silhouette once more and discuss the related approaches. Furthermore, it will allow using notions of epipolar geometry that were introduced in the previous lectures.

Exercise 1 Shape From Silhouette: Background

a) What is the Visual Hull?

b) Cite the main limitation of Shape-From-Silhouette techniques, and two of its typical reconstruction artifacts.

c) Matusik mentions volumetric approaches to compute the visual hull. Recall the main steps of voxel carving, and why its naive implentation can be slow.

d) What is an octree? How can this structure help with the computation of the visual hull?

Exercise 2 Shape From Silhouette: A View Dependent Approach

a) How does view-dependent sampling differ from the volumetric approaches?

b) What do the authors claim as contribution?

c) The Wedge-cache described in Sec. 3 relies on the idea that some computation is redundant and that some of the computational complexity can be replaced by memory usage. Which sentence describes the redundant part of the computation?

d) For a given projective sampling camera, and one of the silhouette cameras, how is the Wedge-Cache structure defined (ignore the degenerate cases)?

e) What do the authors mean by *"lazy computation strategy"*?

f) How is the information from the different cameras merged?

g) Why is the recovery of surface normals important?

h) How is the normal computation performed?

Exercise 3 3D Reconstruction from Multiple Image Silhouettes

In this exercise, we will implement the simplest space carving algorithm to compute a 3D reconstruction from images taken with a multiple camera system. You will need to download the following files from the course webpage:

`pose4-cam*.ppm`	Original images from 6 cameras
`pose4-cam*-sil.ppm`	Silhouettes [0=occupied,255=empty]
`calib-mm00*.sca`	Calibration files
`vol3d.m`	Function to display a 3D occupation grid in MATLAB

a) **(H)** Load the projection matrices. Load the images and display them.

b) **(H)** Draw the projection of the global coordinate system in each image.

c) **(H)** Implement the 3D reconstruction algorithm outlined in Exercise 1. Give the bounding box and voxel size as input to the function. Assume that the volume is first entirely occupied and go through the volume elements (voxels) verifying which are empty. When a small voxel resolution is used, it is enough to check if the projection of the voxel's corners fall inside or outside the silhouette. Let the function return a binary 3-dimensional grid with binary values indicating the occupation.

d) **(H)** Draw the occupied volume. Use the `vol3d.m` function to display the result.