Computational Photography Class Project (Part 1): Structured Light Depth Acquisition

The entire class project counts for 60% of your overall grade: 50% for Part 1, and 50% for Part 2. Part 1 of 2 is the same for all students, and is set out below. Part 2 of 2 will come later, and students will have choices about what to do, with what data, etc.

This part will take you through a minimal implementation of using structured light for 3D depth reconstruction, based on the techniques discussed in:

[1] **High-Accuracy Stereo Depth Maps Using Structured Light**
by Daniel Scharstein and Richard Szeliski
http://research.microsoft.com/pubs/75606/scharstein-cvpr03.pdf
and
[2] **Projector calibration for 3D scanning using virtual target images**
by Hafeez Anwar, Irfanud Din and Kang Park
(pdf available on Moodle, or at link below if you're on-campus)
http://link.springer.com/article/10.1007%2Fs12541-012-0017-3

Unlike [1], you will be expected to implement a minimal variant using only one projector and one camera. This will require the use of calibration techniques to discover the intrinsics of the camera and projector, and their relative pose to one another. This is discussed in [2].

On Moodle, you are provided with a **readme.txt** file, and image sets, where each consists of:

A. A set of gray code patterns, stored as images.
B. Calibration poses of a flat checker board, with projected gray patterns.
C. An object with the full sequence of gray codes projected onto it.
D. *Synthetic image sets only* Full camera matrices and camera parameters. These are stored in json format, and can be read by any text editor.

Both synthetic and a real image sets are provided. The **synthetic** sets are easier and the coursework requires that you start with them. The calibration images all have the camera and projector in the same location, but the plane is being moved - given three different plane locations you can calibrate the relationship between the camera and the projector. The images are paired up (0 and 1, 2 and 3, 4 and 5), so the first image of each pair (even number, where you have blank space) is a chequerboard projected onto the plane from the projector and the second is the chequerboard (odd number) affixed to the plane. Having two images was done for convenience - you will note that for the **real** data provided, both checkerboards appear in the same image, and we provided many more than three so you can get a more precise calibration.

The aim of the coursework is to build and demonstrate a system to determine the camera and projector projection matrices using (B) and the techniques in [2]. From there, estimate the per-pixel depth of an object as it appears to the camera, using (C) and the techniques in [1].

Using only the data provided will reduce your maximum possible score for Part 1 to 80/100. You are expected to gather your own data to get full marks (100% on Part 1). **However**, for just the data-capture of real objects, you can work in groups of up to three people. (Just name the others in your 3-person team when submitting your work.) Beyond the capture, your calibration and all other processing should be done alone, without the help of other students. Class TA's can certainly be consulted.

Your experiments should make use of binary gray codes rather than the sinusoids, where both are described in [1]. A further implementation of the sinusoidal method may be done for extra credit.
You are expected to:

<u>*For the Synthetic Image Sets only:*</u>
3D Reconstruction:
1. Decode the light patterns using image differencing to get (u, v) codes at each pixel.
2. Eliminate pixels whose gray code can not be determined reliably (see [1])
3. *Using the provided calibration matrices.* For each pixel remaining in the image, determine the unique depth that minimises the distance to the ray passing through the projector that is consistent with the (u, v) code.
4. Compute the depth-map for the provided datasets.
5. Visualise the generated 3D point cloud for each image set. Show both the depth image from the camera's perspective, and from an alternate view (as if looking from the side or from above).

Camera calibration:
1. Using the checkerboard patterns provided for each synthetic image set: Determine the projection matrices for the illumination source and the camera using the technique of [2].
2. Reproject the checkerboard patterns into the image plane using your projection matrices, and those provided. Compare the results and discuss the probable causes of errors.
3. Using your generated projection matrices, generate 3D point clouds, as in the previous section. Visualise your results by overlaying the two point clouds (plot each cloud in a different colour).
4. Discuss differences between the point clouds, and the effects of varying the calibration.

<u>*For the Real Image Sets and own data capture:*</u>
1. Acquire images under all gray code illuminations.
2. Decode the light patterns using image differencing to get (u, v) codes at each pixel.
3. Determine the projection matrices for the illumination source and the camera using the technique of [2].
4. Eliminate pixels whose gray code can not be determined reliably (see [1]).

5. For each pixel remaining in the image, determine the unique depth that minimises the distance to the ray, passing through the projector, that is consistent with the (u, v) code.
6. Compute this depth-map for at least two of the provided real-world datasets: real_tea.zip and real_crayon_dalek.zip, and for one (or more) datasets captured by you.
7. Visualise the generated 3D point cloud for each image set, as in previous sections.

Data Capture

To capture real-world data, you will be expected to set up a similar system using a computer, projector and camera. This will require the use of a darkened room with projector, such as MPEB 6.12, or any other classroom with controllable lighting. One or more digital cameras and projectors can be borrowed from Dave Twiselton from The Support Group on the 4th floor of MPEB, in case you don't have access to these already.

1. It is important to avoid shaking the camera or projector while generating results, as even heavy footsteps can break the data capture procedure.

2. Proper layout. Position the camera and the scanning object so that the camera's field of view can be nearly filled by the pattern you project. In the real world, distortions in the pattern generated by the projector are more pronounced around the edge of the projector, so please leave a safe gap between the edge of the object and the end of the pattern projected.

3. Proper image quality. Adjust the camera setting (focus, exposure time, lens etc.) to avoid fuzzy images, over-exposure, and camera bloom. Make sure that you have good quality images, and pay particular attention to the brightness on the boundaries of the pattern. Also, you will want to turn off some automatic functions, such as auto-focus.

Write-up

For this Part 1 of the class project, you must submit a report, along with your documented code. The report should describe the details of your steps, and show images, clearly depicting your results. Describe what worked, but also describe limitations and point out problems - either things you suspect as bugs, or problems endemic to the approach. You may use any programming language you like, but you should use only your own code, where possible. **In the event that you use code or a library from someone else, this must be documented in your final report. Failure to do so will be treated as plagiarism!** The final grade will depend on the quality of your write-up, system implementation, and the results you present. Your final submission must include all code in the zip file.

Late Policy

The department's standard late submission policy applies; late submissions will be marked down accordingly.

Deadline: The deadline will be announced on Moodle to all class participants.