Computational Photography and Capture
Project Part I

Theo Turner

March 2018

This document provides an overview of the methods used for and results obtained from the titular assessment. The aim of this task was to achieve three-dimensional depth reconstruction through the use of structured light. MATLAB was used to implement the algorithm and a subset of results is included in this document. For full results, please see the submitted directory. For the own data capture section, the author partnered with Boyang Pan.

1 Three-dimensional Reconstruction

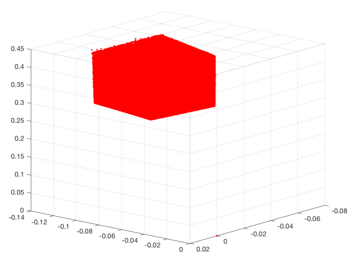
Given a set of images in which a sequence of bands of light have been projected on an object, the algorithm decodes the light patterns by means of image differencing. This requires splitting the image set into pairs and computing the image representing the absolute pixel difference between these, and is described in [1]. Looping through all pairs, a matrix of size *frame width* x *frame height* x *half the total number of frames* is produced, with the first half of the last dimension giving u-codes and the second half v-codes. Considering each pixel of this matrix in turn, if the pixel difference is found to be greater than zero, it is marked as a 1, otherwise 0. The result is a binary value of length equal to half the total number of frames per pixel, known as the gray code.

In order to save space and make the output size more manageable, the binary codes are converted to decimals. This allows the final (u, v)-code matrix to have dimension *frame width* x *frame height* x *2*, with the layers of the last dimensions being the separate u and v codes.

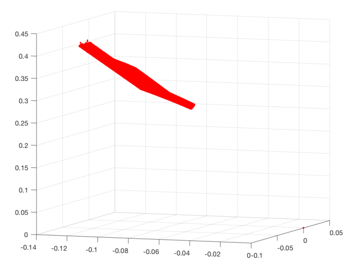
In addition, to eliminate pixels whose gray code can not be determined reliably, a simple method of setting a maximum allowable level of 'background noise' is used. For each set of differences at each pixel location, if the sum of these differences across all pairs is less than a threshold value (found by trial-and-error and specific to each input), then the pixel is deemed unreliable and its gray code is set to -1 (appropriate given decimals cannot be negative).

Using provided intrinsic and extrinsic parameters for the camera and projector, the depth map of each object is then computed. For each reliable pixel in the image, the unique depth that minimises the distance to the ray passing through the projector (that is consistent with the (u, v)-code) is found. The method for this is described in Simon Prince's *Computer Vision: Models, Learning and Inference* [2]. Given N calibrated cameras in known positions, viewing the same three-dimensional point w and knowing the corresponding projections in a set of images x, the position of the point in the world is established. This is done by looping through each pixel, converting each position to normalised camera co-ordinates, computing linear constraints and finding the least squares solution for the parameters. This solution is then computed from the camera's perspective to find the depth map.

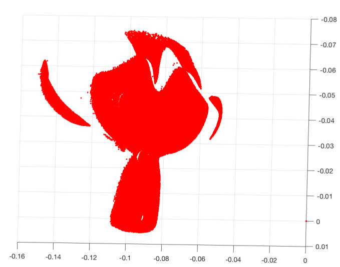
The depth maps can be visualised as point clouds.



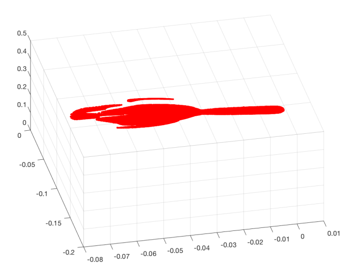
(a) Depth map of a synthetic cube from the perspective of the camera. There are a small number of errors on the rear edges.



(b) The same depth map from a side-on perspective. Notice it is not a complete cube as the camera cannot observe the rear faces.



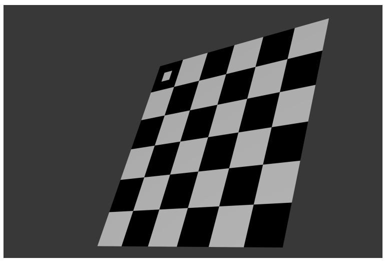
(c) Depth map of a synthetic monkey face.



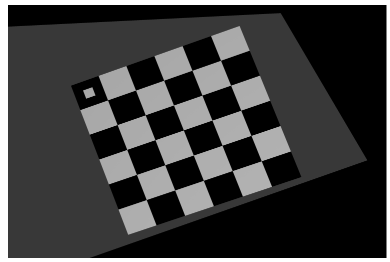
(d) The same depth map from a side-on perspective.

2 Camera Calibration

Using provided checkerboard patterns, the projection matrices for the illumination source and the camera are found using the method described in *Projector Calibration for 3D Scanning Using Virtual Target Images* [3]. The user is asked to select the corners of the checkerboard and the best homography between these points and the known projected points (found by the position of the top-left corner of the board and its known dimensions) is computed using a method described in Gabriel Brostow's Machine Vision course at University College London [4]. This homography is expressed as a two-dimensional projection and the frame is transformed such that the reprojected view appears as the projector sees the world.



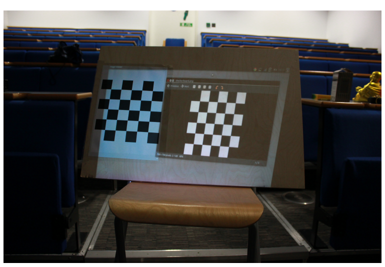
(e) Synthetic checkerboard before reprojection.



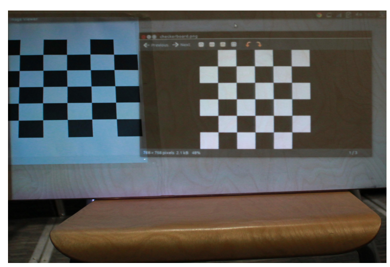
(f) The same checkerboard after reprojection.

The most likely cause of error in the case of synthetic data is the user not selecting the exact corners of the checkerboard.

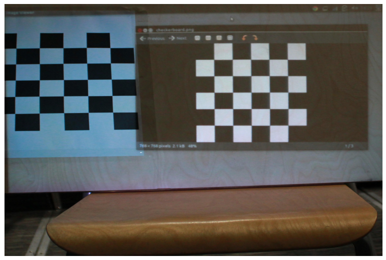
Alternative camera and projector calibration has been performed using Jean-Yves Bouguet's Camera Calibration Toolbox for MATLAB. Results from calibrations have been saved in new .matrices files.



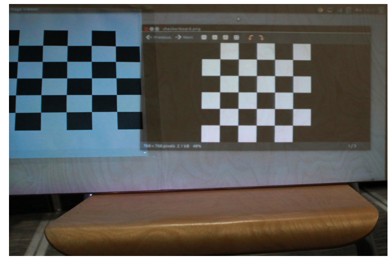
(g) Real checkerboard before reprojection.



(h) The same checkerboard after reprojection.

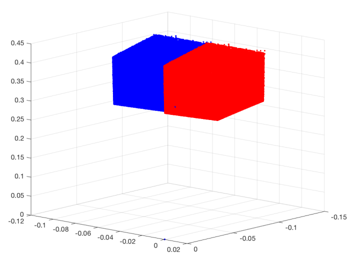


(i) The same reprojected checkerboard as above (for side-by-side comparison).

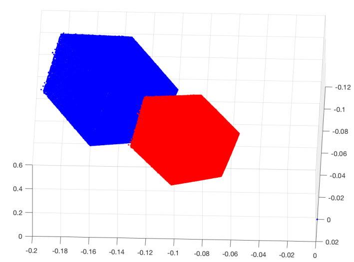


(j) The same checkerboard after re-projection using alternative calibration matrices.

The most likely cause of errors in this case is inaccurate calibration, probably as a result of inexact user input. The differences in calibration can also be expressed using depth map point clouds. Poor lighting conditions (including colour) may also lead to projection errors.



(k) The depth maps for the synthetic cube produced by different calibrations. The provided synthetic matrices produce the red cloud, the alternative ones produce the blue one.

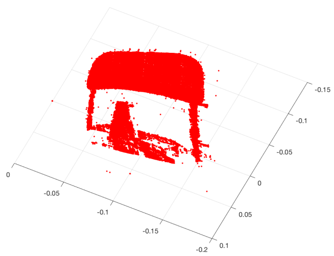


(l) The same two point clouds with the alternative extrinsic matrix having its difference from the synthetic one scaled up.

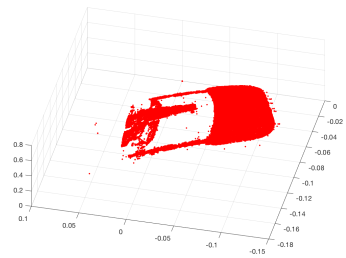
The fact that the point clouds are spatially close and in fact overlap suggests that the alternative calibration is a reasonable result. Adjusting the intrinsic matrix shifts the optical center and focal length, whereas adjusting the extrinsic matrix shifts rotation and translation. Note that extrinsic parameters are computed using intrinsic ones, so changing the latter also has an effect on the former.

3 Real Data, Own Calibration and Own Data

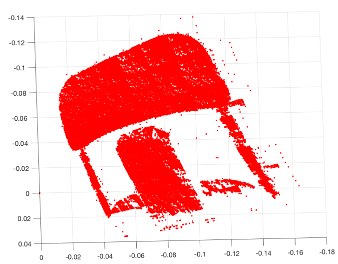
Until now only depth maps for synthetic data have been shown. Using the exact same methodology, depth maps for a number of real objects have been computed.



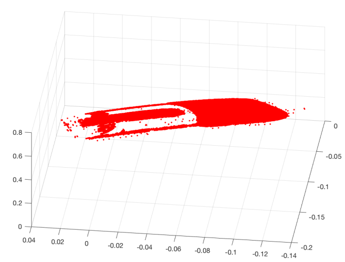
(m) The depth map for a Dalek figurine and a set of crayons placed on a chair.



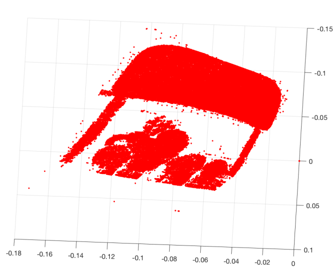
(n) The same depth map from a side-on view.



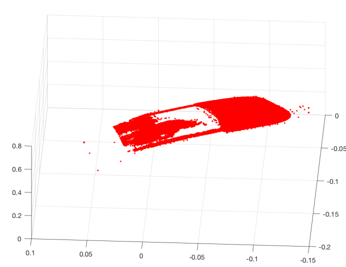
(o) The depth map for a box of tea placed on a chair.



(p) The same depth map from a side-on view.



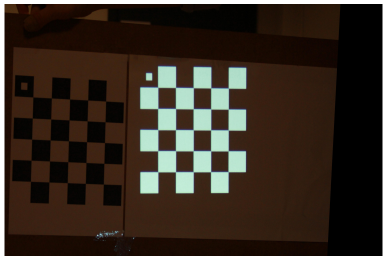
(q) The depth map for a teacup and a set of balls placed on a chair.



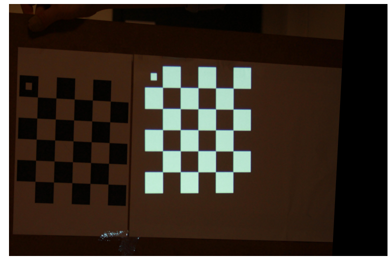
(r) The same depth map from a side-on view.

The algorithm has difficulty identifying the depth of more reflective objects, most evident in the last example. In this case, the acceptable background noise level had to be reduced to capture acceptable depth information.

Calibration using an alternative checkerboard setup has also been performed. The algorithm was applied in exactly the same way to this data.

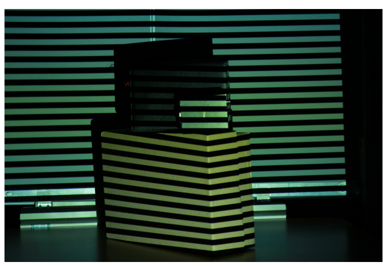


(s) A frame of own captured data before reprojection.

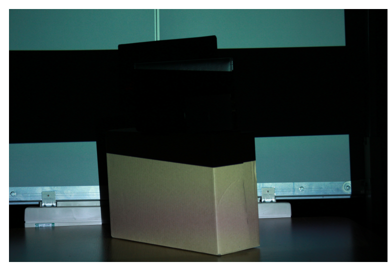


(t) The same frame after reprojection.

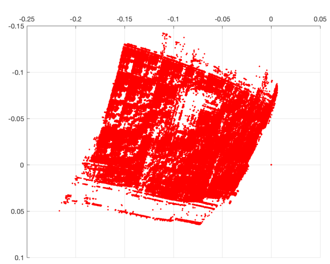
Own data capture was also performed. In this case, a flat, vertical surface was placed close behind the objects in the depth map in the hopes of achieving interesting results. The objects used were a black book and black box placed on top of a larger brown box.



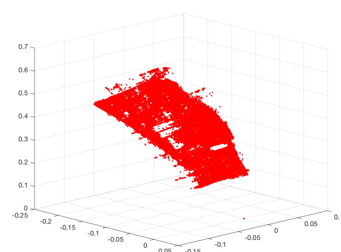
(u) The objects used.



(v) A particularly problematic light pattern. Notice how the book fades into the dark part of the pattern.



(w) Depth map of the data.



(x) The same depth map from a side-on view.

The shadows cast on the vertical flat surface, as well as the book, are difficult to distinguish from the dark parts of the light pattern. As a result, the depth maps are erratic. However, with appropriate acceptable background noise level adjustment, the shadows and book are marked as unreliable pixels and not plotted.

References

[1] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, vol. I. Madison, WI: IEEE Computer Society, June 2003, pp. 195–202. [Online]. Available: https://www.microsoft.com/en-us/research/publication/high-accuracy-stereo-depth-maps-using-structured-light/

[2] S. J. Prince, *Computer vision: models, learning, and inference.* Cambridge University Press, 2012.

[3] H. Anwar, I. Din, and K. Park, "Projector calibration for 3d scanning using virtual target images," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 1, pp. 125–131, Jan 2012. [Online]. Available: https://doi.org/10.1007/s12541-012-0017-3

[4] B. G., *Machine Vision.* University College London, 2017.