Advanced Vision Practical

Bob Fisher
School of Informatics

February 2019

Abstract

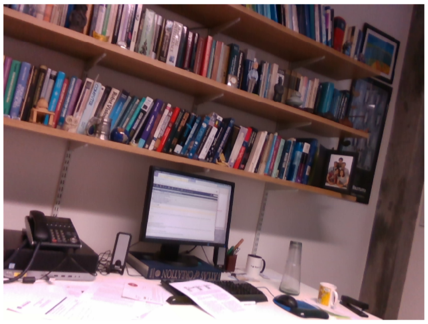
This describes the Advanced Vision assessed practical. The main goal is to reconstruct the inside of a portion of my office from a set of 3D point clouds acquired from an Intel RealSense depth sensor that is moved to scan different views. **There are 4 stages to this assignment, and you must do all 4 for full marks.** You must do this practical in teams of 2. The final stage is an assessed live demonstration of your practical on Friday morning, March 29.

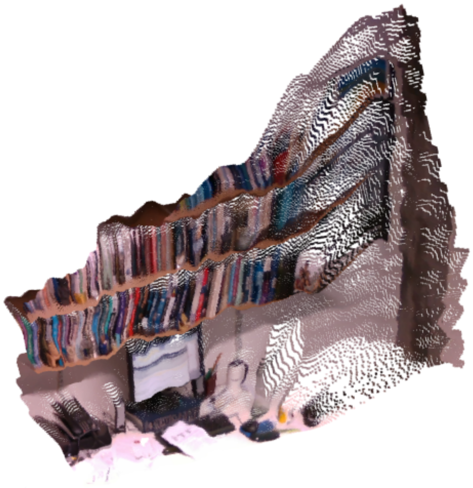
1 Task Background

The data for constructing the complete box comes from a set of 3D point clouds acquired by an Intel RealSense depth sensor. Each point cloud comes with XYZ values for the point positions and RGB values for the point colours. The data for each frame has its own coordinate system, with the camera at the origin, the X and Y coordinates horizontally and vertically, and Z going away from the camera.

The core of the assignment is to estimate how the 3D points captured in each frame can be transformed into points in one selected frame. Once all points are transformed, in theory one can see a complete view of the back end of my office.

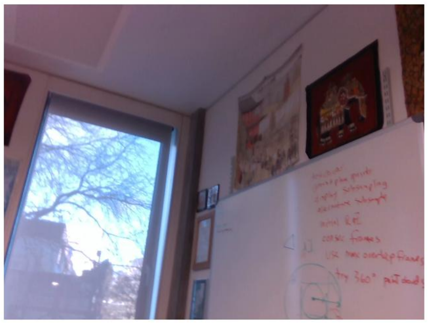
In the left image below, you can see an RGB image of the data from one frame. The right is a view of the 3D points (coloured with the surface colour).

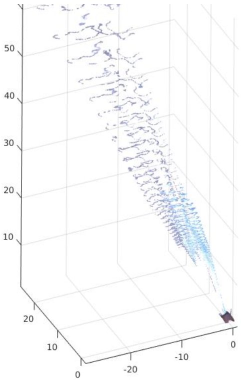




The left image below is a close up view of some of the point cloud, where you can see the individual points more clearly. The right image is the point cloud from frame 27. There is an undesirable person in this data - you have to remove the person and use the rest for room reconstruction.

The left image below shows a frame with the window visible. This transparent object causes problems. The right image shows the point cloud for this image. The little brown patch at the bottom right is the desired wall data. All of the other points are from the window area or are from outside and need to be removed.





2 Resources

The data (66 MB) for this practical can be found at:

`http://homepages.inf.ed.ac.uk/rbf/AVDATA/AV119DATA/office1.mat`

The data consists of N=40 frames of 3D point cloud data, where each frame is from a different viewpoint. The data is stored in a cell structure with N cells, each with 2 entries: X.Color which is the RGB value of each point and X.Location, which is the XYZ location of each point (and X is the cell index - see the sample code in `reading_points.m`. The original RGB or XYZ images are 480 rows by 640 columns by 3 channels but are encoded as a list of points(480*640,3). You can reshape one of the channel lists (*e.g.* the red channel) back to an image by `reshape(list(:,1), [640,480])`. The colour image shown at the above left was formed by reshaping the X.Color array.

There are also 2 Matlab support functions:

`http://homepages.inf.ed.ac.uk/rbf/AVDATA/AV119DATA/imag2d.m`
`http://homepages.inf.ed.ac.uk/rbf/AVDATA/AV119DATA/reading_points.m`

The function `imag2d` displays a list of colour points as a colour image as in the left image above. The function `reading_points` shows loading and displaying the set of point clouds, as in the right image above.

3 Task Details

The overall task for this assignment is to fuse the individual XYZ point clouds to create as much of a complete 3D room model as possible. To do this you need to write a set of programs that can:

1. Extract the relevant data from each point cloud.
2. Estimate the reference transformation linking each consecutive pair of frames.
3. Fuse all of the points into a single 3D coordinate system.
4. Build as complete a 3D model as possible and characterise the surfaces.

Each of these is described in more detail below.

3.1 Extract the relevant data from each point cloud

One key problem is finding the relevant data to use for fusing. By looking at the point clouds shown above, we can see that there are several issues: 1) irrelevant points outside the window, 2) a person in one frame, 3) defective depth data, probably due to specular reflections (for example, see the big spike next to me above), 4) deformed data near the edges of the image, and 5) noisy ripples in the data.

Here are some suggestions on how to deal with each of these:

1. From the 3D plot, we can see that these points are much further away than the room data. The camera was usually within 2-3 metres of the room wall, so thresholding the Z coordinate should work.
2. Sometimes colour can be used to identify items to be removed. In this example, my skin colour is reasonably distinctive, but my shirt and trousers are not. One could also use my position in front of the background, but that analysis is a bit too complex for this assignment. So, one could manually choose a bounding box in the 2D image and remove all 3D points from within that box.
3. There are often 3D points at the edges of foreground objects that are in the empty space between the foreground and background. These are called flying pixels. You can see some next to me in the image above. There is also the spike in the image next to me. One method for removing some of these points is: for each 3D point, count the number of other 3D points that are within a given distance of the initial point. If the number is too small, then we decide that the first point is noise, and eliminate it. Choose the distance and number thresholds is by trying different values and seeing the effect.
4. Because there is a one-to-one relation between image pixels and 3D points (see `imag2d.m`), we can identify the points that come from the edges of the image, and eliminate these.
5. We'll accept the noisy ripples for this assignment.

You should keep track of which 3D points to keep and which to ignore. Draw a binary image for each frame showing which pixels are kept (white) and which are ignored (black).

See drill lab exercises 4 and 5. These demonstrate how to read, write, and interpret the point-cloud data and the relation between a point-cloud and its 2D image.

3.2 Estimating the Reference Frame Transformation Between Consecutive Frames

Because the data frames were captured consecutively, there are a large number of points in common to each frame. The 3D points will not match exactly because of noise, camera motion, and sampling variation, but they should be close enough. Given a good set of corresponding points, one can use the algorithm given in Section 2.1 of this paper to estimate the translation and rotation that links the two point clouds:

http://homepages.inf.ed.ac.uk/rbf/MY_DAI_OLD_FTP/rp765.pdf

This algorithm needs a minimum of 3 points, but, since the data is quite noisy, more points are better, eg. 10-20+ would be preferable.

Maybe there is a Matlab equivalent.

The key issue is how to identify the corresponding points. Because there is a registered intensity image, one can detect SIFT (see course videos) points (or other types of image points). Using only points from the allowed binary mask areas, one can match points from the two images using the SIFT descriptors. Once you have matched the colour image points, this also gives you matched 3D points.

One possible approach to estimating the rotation and translation is to use all matched SIFT points in the algorithm given above.

If this is too erroneous, e.g. if there there are some incorrect matches, then you could try RANSAC (see course videos). You need 3 pairs of matched points to estimate the rotation and translation. With the rotation and translation, you can transform the 3D points corresponding to the 2D SIFT points from one point cloud to the other. One then counts the number of SIFT points (use the 3D coordinates) that match. If there are enough, then we accept these as valid matched points and estimate the rotation and translation using these.

This rotation and translation estimation depends on a few points and may be noisy because of the errors in the point positions. One can potentially reduce the alignment error by using the Iterative Closest Point (ICP) algorithm. Given the initial registration from the steps above, ICP can refine the registration. Use the Matlab `pcregrigid` function for one iteration on the initially registered point clouds.

For each consecutive pair of images to be registered, display:

- The SIFT points overlaid on each colour image, with matched points in green and unmatched points in red (or another pair colours if you are red/green colourblind).
- The 3D points from the first image transformed into the coordinate system of the second 3D point set. Display both point sets together.

3.3 Fuse all of the points into a single 3D coordinate system

Given the reference frame transformation (rotation + translation) between each consecutive pair of frames, you can then map all of the points into the initial frame. Ie, if $T_{t,t+1}$ maps points in frame $t+1$ to frame t, then you can also map points from frame $t+2$ to frame t, because $T_{t,t+2} = T_{t,t+1} * T_{t+1,t+2}$. And so on, until you have all of the points from any frame t to frame 1.

Once you have transformed the points, you could: a) just concatenate the sets, or b) use the Matlab command `pcmerge` (which averages points to reduce the number of points).

This means that you can get a complete point cloud model of the end of the office.

After registering each pair of point clouds, display the fully merged set (i.e. frame 1+2, 1+2+3, 1+2+3+4, etc).

3.4 Evaluate the quality of the final model

In the lectures, we saw how to fit a 3D plane to a set of 3D points. Here, the points $\{(x_i, y_i, z_i)'\}$ will need to be augmented with a final 1 value to become $\{\vec{d_i}\} = \{(x_i, y_i, z_i, 1)'\}$. Assume that the least square plane fit solution was the 4 vector $\vec{p}$. In other words, each of the data points $\vec{d_i}$ approximately satisfies $\vec{p}\,'\vec{d_i} \doteq 0$.

In theory, the left and right walls of the room are parallel to each other. Similarly, the wall at the end of the room should be perpendicular to the side walls.

Given the final registered points, use some heuristics to select the registered points from the left, right and end walls. Remove the ceiling points.

Fit a plane to each of the three subsets. Extract the surface normals and compute the angles between each pair of planes. Using the plane fits, compute the separation between the left and right walls, using a plane point near the centre-of-mass of the wall points.

4 Your Report

Each team writes a single report that describes:

- The algorithms that you used for each stage of the process.
- How well the algorithms performed on the supplied test data. Show the statistical results and images requested above.
- Show some example images of each processing stage, including some original images and the detected walls.
- Show some examples of any unsuccessful frame fusions.
- Discussion on performance: successes and failures, causes of failures and potential remedies.
- As an appendix, add the code that your team wrote. Do not include code that was downloaded from the AV or IVR or other web sites, but include a statement about what code and where it came from.

5 Other Comments

The assignment is estimated to take 20 hours coding/test and 5 hours report writing per person, resulting in a 10 page report plus the code appendix.

You must do this assignment in teams of 2. You must find your partner and email Bob Fisher (rbf@inf.ed.ac.uk) the name of your partner.

A single, joint, report is to be submitted. Split the work so that each partner can do most work independently (*i.e.* share the work rather than duplicate it).

You can use the lecture example code from:
`http://www.inf.ed.ac.uk/teaching/courses/av/MATLAB/`

If you find a limit on the use of Matlab Image Processing Toolbox licenses, some alternative MATLAB functions are at:
`http://www.inf.ed.ac.uk/teaching/courses/av/MATLAB/UTILITIES/`

6 Assignment Submissions and Deadlines

There are 4 stages to this assignment, with 3 written submissions. For each stage, you submit only 1 PDF file for both of members of the team. It does not matter which of you submits the PDF file.

1. The first submission is your draft practical report, containing all of the details as described above. This submission counts for 0% of your course marks. **If you do not submit a full draft report, then your team and your report will not participate in the second stage (and thus you will lose 5%).**

Submit your report in PDF online by 4pm Monday March 18. The online submission line on the School's DICE system is:

```
submit av cw1 FILE1.pdf
```

where `FILE1.pdf` is the name of your PDF file.

Because other groups depend on this submission, no late submissions are allowed for the first part. If you submit late, then you get a 0% for parts 1 and 2.

2. The second stage is the peer review. Your report will be given to another team for review, and you will be given another team's report to review. Instructions on what to give feedback on is given in Section 8 below. You are not marking their report; instead you are giving them criticisms that help them improve their report and results. A good quality report gets a maximum of 5% of your final course mark.

   Submit your report (in PDF) on the other team's report online by 4pm Thursday March 21. The online submission line on the School's DICE system is:

   ```
   submit av cw2 FILE2.pdf
   ```

   where `FILE2.pdf` is the name of your PDF file.

   Assuming that the other team has done their report, your team should receive a feedback report later on Thursday March 21.

3. The third stage is your revised final PDF report. This could be the same as the draft report from stage 1, but I hope that the stage 2 review will have given you suggestions for improvements.

   Submit your report (in PDF) on the other team's report online by 4pm Thursday March 28. The online submission line on the School's DICE system is:

   ```
   submit av cw3 FILE3.pdf
   ```

   where `FILE3.pdf` is the name of your PDF file.

   The final report will be marked as follows:

Issue	Percentage
1. Clear description of sensible algorithms used	40%
2. Performance on the data set	40%
3. Clear Matlab code	10%
4. Discussion of result quality and causes of any failures	10%

   This submission counts for 20% of your course marks.

4. The final stage is an assessed live demonstration of your practical on Friday morning, March 29. This counts for 5% of your course marks. The demonstration will be in AT 5.08 between 9:00 and 13:00. You will get an assigned 30 minute slot for demonstrating your practical on a new, but similar, dataset.

7 Live Demonstration

There will also be a demonstration session assigned between 9:00-13:00 on Friday March 29, where you will have to demonstrate your code on a new dataset. The demonstrations will be in the usual lab: AT 5.08 South.

We'll email you about the schedule, but each team will have 30 minutes to demonstrate their practical.

A new but similar dataset will be available for demonstrating your results. You will need your matlab program to show, for each frame (first 10 frames only, and ignoring any frame that includes me):

1. The original image and point cloud.
2. The cleaned point cloud.
3. A binary image showing which pixels are kept (white) and which are ignored (black).
4. The current frame with SIFT points (red)
5. The current and previous frame with SIFT points (green=matched, red=unmatched)
6. The registered point clouds from the previous and current frame
7. All registered point clouds after each new frame is added

Put a pause after each frame.

8 Guidance on Stage 2 Feedback

You are expected to give a **maximum** of one page of feedback. You can use word or LaTeX. Any reasonable margins and font sizes are allowed.

The feedback should be helpful. I.e., if you see something that could be fixed, suggest how it can be fixed. Things to possibly comment include:

1. Does the report have a clear introduction?
2. Is it clear what methods were used, and how these methods were implemented?
3. Were the methods evaluated? How were the methods evaluated?
4. Were the expected results presented? Were the results good? Do you see situations where the methods failed? Do you have suggestions for how to improve the methods or results?
5. Did the report include the new code written? Was it clear what the code did and how?
6. Did the discussion of the results address the successes, failures and possible improvements?

9 Publication of Solutions

We will not publish a solution set of code. You may make public your solution **but only 2 weeks after the submission date**. Making the solutions public before then will create suspicions about why you made them public.

Good Scholarly Practice:

Please remember the University requirement as regards all assessed work. Details about this can be found at:

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

10 Plagiarism Avoidance Advice

You are expected to write the document in your own words. Short quotations (with proper, explicit attribution) are allowed, but the bulk of the submission should be your own work. Use proper citation style for all citations, whether traditional paper resources or web-based materials.

If you use small amounts of code from another student or the web, you must acknowledge the original source and make clear what portions of the code were yours and what were obtained elsewhere. You can ignore this condition for the AV lecture examples, which can be used freely.

The school has a robust policy on plagiarism that can be viewed here: http://web.inf.ed.ac.uk/infweb/admin/policies/guidelines-plagiarism.

The school uses various techniques to detect plagiarism, included automated tools and comparison against on-line repositories. *Remember: a weak assignment is not a ruined career (and may not reduce your final average more than 1%), but getting caught at plagiarism could ruin it.*

11 Late coursework policy

Because other groups depend on the first stage submission, no late submissions are allowed for the first part. If you submit late, then you get a 0% for parts 1 and 2.

Otherwise, see: http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests