Advanced Vision Practical 3

Bob Fisher
School of Informatics

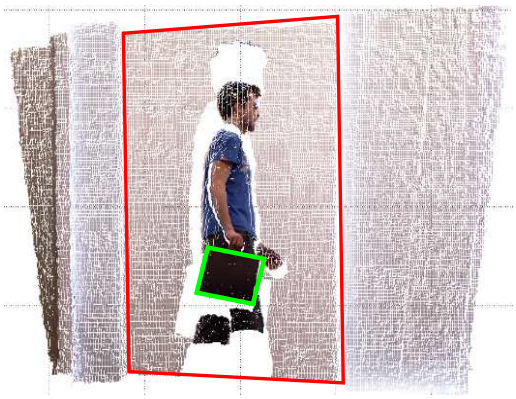
March 2012

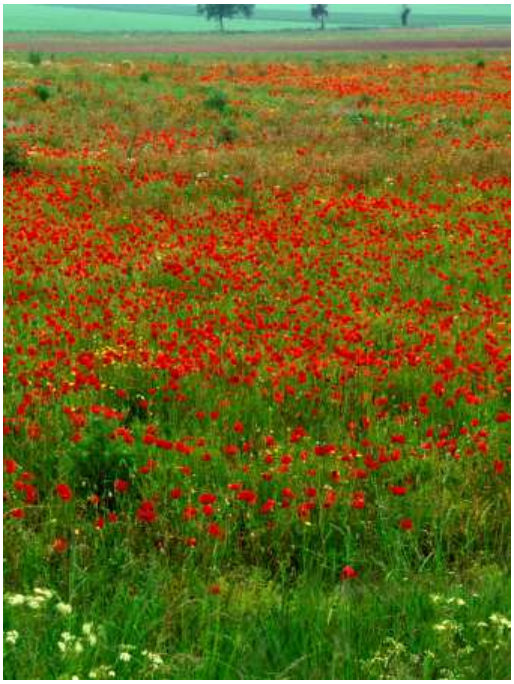
Abstract

This describes the third assignment for assessment on Advanced Vision. The main goal is to extract and overlay planes from a Kinect range image video. The assignment is due: **4pm Thursday 29 March**. You must do this practical in teams of 2, and submit 1 report only.

Task Background

At the left you can see one frame from a video captured by a Kinect sensor of a person walking in front of a wall.





You will notice the person is carrying a planar quadrilateral (outlined in green) and there is a planar quadrilateral (outlined in red) behind the person.

We have captured a short video of 36 frames using a Kinect sensor, so have a set of registered depth and colour data. Your goal is to make a new short video, using a little Hollywood magic, in which the scene is modified. In particular, you need to:

1. Extract the person from in front of the wall.
2. Overlay (using homographic transfer) the background wall (red outline) with the new scene image shown at the right above.
3. Extract the plane and boundary of the black quadrilateral (green outline) that the person is carrying.

4. Overlay (using homographic transfer) a new video on the quadrilateral.

5. Reconstruct a video from the combined images.

It is not necessary to write a computer program that does all steps automatically - some manual intervention, *e.g.* to identify the 4 points needed for the background image transfer, is OK.

At the URL:

`http://homepages.inf.ed.ac.uk/rbf/AV312DATA/bindermat.tar`

you can see the dataset needed for the practical. At the URL:

`http://homepages.inf.ed.ac.uk/rbf/AV312DATA/binder.avi`

you can see a video (125 Mb) of the practical.

There are 36 frames extracted from a Kinect video sequence and reformatted into a Matlab array, with one array for each frame.

When you load this file into Matlab (e.g. `load xyzrgb_frame_0001`) this results in an array (e.g. `xyzrgb_frame_0001`). The array is 307200 x 6 doubles, which you should rebuild as an 480x640x6 array. Data(:,:,1:3) are the RGB values of each pixel and Data(:,:,4:6) are the XYZ values of the corresponding 3D point.

What you have to do:

1. The camera is stationary. We are interested only in the central panel. So, you need to find the image pixel coordinates of the corners of that patch.

2. If you recall: in IVR we had a short example of image transfer. You need to transfer the image at:

   `http://homepages.inf.ed.ac.uk/rbf/AV312DATA/field.jpg`

   onto the central panel (see the right image above). Since the plane does not move you can determine the image homography points once manually.

3. Proper transfer will require you to only transfer to pixels that lie behind the walking person. To do this, you need to determine whether the pixel lies in the background plane or in front of the plane (ie. is part of the person). Find the least-square equation of the plane in 3D and then only transfer pixels that lie in the plane. Note that there are some pixels that may not have any range data in a given frame because of the shadow that is cast by the infrared structured light emitter. However, you have 36 frames, so should be able to figure out what the depth and colour of every background pixel is.

4. Notice that the walking person is carrying a rectangular plane. This is another opportunity for some augmented reality. Capture a 36 frame video of your choice and transfer it onto the rectangle. In this case, you are expected to locate the rectangle automatically. You can use: a) the colour (eg. some thresholding) and the fact that it's a plane (eg. some RANSAC and region growing) to locate it and its 4 corners. Transfer each different frame of your video onto the corresponding frame of the input video. Note that the rectangle is only visible for about frames 15-25.

5. Finally, assemble the individual frames into an avi or other video suitable for display, using this code:

```
function [  ] = images2vid(  )

numImages = 10;
filename{numImages} = {};
filename(:) = {'depth_grey_0011.tif'};

for i=1:numImages
 % generate an image from somewhere
 im = imread(filename{i});
```

```
 imshow(im);
 % get a movie frame (a snapshot of the current axis)
 M(i) = getframe(gcf);
end

% write movie object to disk
fps = 5;
movie2avi(M,'AV_movie.avi','FPS',fps);
```

Other Comments

1. As you will see, there may be foreground pixels without data. Don't show the background pixels through the walking person - use open/close to fill in the mask of the walking person.
2. You can use the lecture example code from IVR for the transfer. See `esthomog.m` and `remap.m` from: `http://www.inf.ed.ac.uk/teaching/courses/ivr/MATLAB/FLATPARTRECOG/`
3. Write a report that:
   - Describes the algorithms that you used for background detection and image transfer, foreground person detection, foreground plane detection and video transfer.
   - Shows images of the performance at each stage for several frames with the person at the centre of the image.
   - Discussion on performance: successes and failures. Causes of failures and potential remedies.
   - Your new code. Do not include any code that was downloaded from the IVR or AV web sites.

Assignment Submission

Submit your report in PDF online by 4 pm Thursday 29 March. The online submission line is:

```
submit av 3 FILE
```

where `FILE` is the name of your file.

There will be a live demonstration of your reconstructed video on Friday 30 March between 14:00 and 18:00. You will have to show the video and the frame generation process for a selected frame.

The assignment is estimated to take 15 hours work, resulting in a 5 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) the name of your partner. The partner must be someone different from your partner in Practical 2. A single, joint, report is to be submitted.

The assignment will be marked as follows:

Issue	Percentage
1. Clear description of algorithms used	30%
2. Performance on supplied image data	30%
3. Clear Matlab code	10%
4. Discussion of result quality and causes of any failures	10%
5. Live demonstration of video	20%

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://www.inf.ed.ac.uk/teaching/plagiarism.html`

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.*