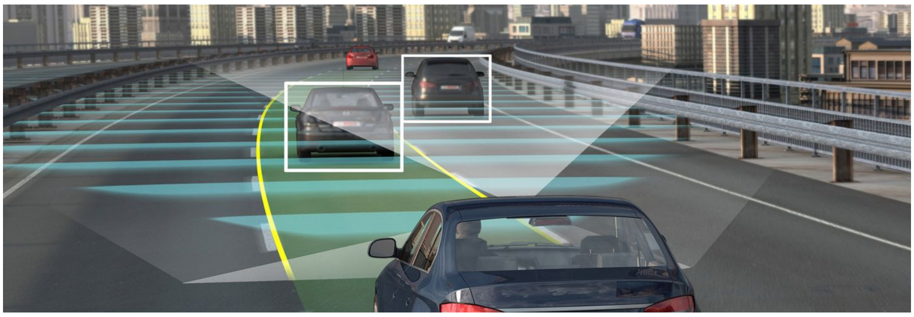
Autonomous Vehicles: Road Surface Region Detection in 3D Stereo



Background

Autonomous road vehicles and advanced driver assistance systems are fast becoming a reality. Computer Vision is increasingly being used to allow such vehicles to understand the road environment around them based on imagery from on-board forward facing cameras.

In this assignment we are dealing with the automatic detection of the unoccupied road surface immediately in front of the vehicle within stereo video imagery from an on-board forward facing stereo camera. This can be performed by finding the 3D plane corresponding to the depth (disparity) information recovered from an existing stereo vision algorithm. Knowledge of the road surface directly in front of the vehicle, in terms of both its bounds and objects that are present within that region assist both automatic forward motion planning and collision avoidance.

The low cost and high granularity (i.e. full-scene) 3D information available from stereo vision means that detection of the road surface and obstacles in and around the vehicle can be detected more readily than with radar or lidar (laser) sensing technologies.

As such, we have a set of still image pairs (left and right) extracted from on-board forward facing stereo video footage under varying illumination conditions and road surface environments. Your task is to design and prototype a computer vision system to detect the 3D orientation and bounds of the road surface directly from of the vehicle imagery based on shape detection using the RANdom SAmple and Consensus (RANSAC) algorithm outlined in lectures. You will develop this prototype system using Python with the OpenCV library and the techniques covered in the module.

This is a real-world task, comprising a real-world image set. As such, this is an open-ended challenge type task to which a perfect solution that works perfectly over all the images in the provided data set may not be possible.

Task Specification – *Road Surface Region Detection*

You are required develop to road surface detection system that correctly detects the 3D planar orientation and bounds (i.e. edges) of any free (unoccupied) road surface region immediately in-front of the vehicle in which the autonomous vehicle needs to operate (e.g. for staying in the correct lane / staying on the road itself / avoiding obstacles / automatic braking).

In constructing your solution you may wish to consider the following aspects of your design:

- exploring the optimization of the provided stereo vision algorithm in use and its operation and hows its performance under varying illumination conditions could perhaps be improved using the HSV or other colour spaces covered in CS SM L2 - Image Processing (these paper also presents an interesting research approach [1,2], others solutions exist also for illumination invariant colour spaces; use these as a starting point for your search).

- selection of a region of interest, possibly adaptively, within the image (including possibly areas of road, pavement, other or not) that represents the region directly in-front of the vehicle and how to deal with the problem of missing disparity (depth) values in that region.

- Calculating the equation of a plane from 3 points in 3D: http://mathworld.wolfram.com/Plane.html
  [*Further hint:* for this assignment this can be done in full projected floating-point 3D space *(X,Y, Z)*or in integer image space *(x,y,disparity) – see provided hints python file*]

Your solution must use a RANdom SAmple and Consensus (RANSAC) approach to perform the detection of the 3D plane in front of the vehicle (when and where possible). For the avoidance of doubt, no credit will be given for a 2D solution based on the built-in *Hough Transform or Douglas-Pecker contour* detection in OpenCV or that does not recover the 3D parameters of the plane.

Additionally, some example images may not have significant noise-free disparity (depth) available in front of the vehicle or the road region may be partially occluded by other objects (people, vehicles etc.). The road surface itself will change in terrain type, illumination conditions and road markings – ideally your solution should be able to cope with all of these. Road edges may or may not be marked by line markings in the colour image. All examples will contain a clear front facing view of the road in front of the vehicle only – your system should report all appropriate road surface plane instances it can detect *recognising this may not be possible for all cases within the data set provided.*

As this is only a prototype – efficiency of your approach is less important than performance.

Additional Program Specifications

Additionally, to facilitate easy testing, your prototype program **must** meet the following **functional requirements:**

- Your program must contain an obvious variable setting in the top of the main code file that allows a directory containing images to be specified. e.g.

  ```
  master_path_to_dataset = "TTBB-durham-02-10-17-sub10"
  ```

  from which it will cycle through each stereo pair in turn processing it for road surface plane detection and display it. A basic example (*stereo_disparity.py*) for cycling through the data set of images and computing the stereo disparity is provided on DUO.

- When the road surface plane are detected within a stereo image it must display a red polygon on the left (colour) image highlighting where the road plane has been detected as shown in Figure 1 (see the drawing examples in the OpenCV Python Lab exercises).

- Furthermore, for each image file it encounters in the directory listing it must display the following to standard output:

  ```
  filename_L.png
  filename_R.png : road surface normal (a, b, c)
  ```

Figure 1: Example polygon outline of the detected road plane region (shown with varying levels of polygon complexity) drawn on the left (colour) image.

where "filename" is the current image filename and *(a, b, c)* are the normalized surface normal coefficients of the road plane that has been detected. When no road plane region can be detected output a zero vector. *Your final program must run through all the files as a "batch" without requiring a user key press or similar.*

- For the purposes of this assignment when a road has either curved road edges or other complexities due to the road configuration (e.g. junctions, roundabouts, road type, occlusions) report and display the road boundaries as far as possible using a polygon or an alternative pixel-wise boundary.

  You may use any heuristics you wish to aid/filter/adjust your approach but RANSAC must be central to the detection you perform.

- Your program must compile and work with **OpenCV 3.3** on the lab PCs.

Sample Data & Example Software

Figure 2: Example left (colour), right (greyscale, rectified) and corresponding disparity calculated using the example python code provided for the assignment.

The sample data provided is a set of 1449 sequential still image stereo pairs extracted from on-board stereo camera video footage (see example – Figure 2). These images have been rectified based on the camera calibration and you do not need to perform stereo calibration yourself.

- The full set of images is available as a single ZIP file from DUO.

  TTBB-durham-02-10-17-sub10.zip

  Be aware that this data set is still large! (~2Gb, this is the nature of this business).

A set of example python scripts are also provided as follows:

- *stereo_disparity.py* – cycles through the dataset and calculates the disparity from the left and right stereo images provided.
- *stereo_to_3d.py* – projects a single example stereo pair to a 3D, write a point cloud of this data to file for reference and shows an example back-projection from 3D to the 2D image.
- *planar_fitting_hints.py* – a set of hints on how to efficiently use some of the advanced library features of Python to compute the coefficients of a plane from a given set of 3D points.

These examples are available via DUO and from the assignment github repository – https://github.com/tobybreckon/stereo-disparity

Marks

The marks for this assignment will be awarded as follows:

- Overall design and implementation of your solution including aspects of:
  - any image pre-filtering performed (or similar first stage processing)
  - choice of region of interest extraction methodology
  - effective use of RANSAC for 3D planar surface detection — 30%
- General performance on detection of planar orientation and bounds in the imagery ** (taking into account accuracy, false detection, missed detection, failures etc.) — 30%
- Clear, well documented program source code — 10%
- Report:
  - Discussion / detail of solution design and choices made — 5%
  - *Statistical* evidence of the performance of system at the task — 5%
- Additional credit will be given for one or more of the following:
  - plotting of the planar normal direction direction glyph / vector in the image
  - automatically adjusting the parameters initial region of interest extraction or image pre-filtering based on some form of preliminary analysis image
  - the successful use of any heuristics to speed up processing times or reduce false detections (including at the stage of colour pre-filtering)
  - automatically detecting and highlighting obstacles that rise above the road surface plane (vehicles, pedestrians, bollards etc.) as they appear directly in front of the vehicle s

  (for any of the above up to a maximum, dependent on quality) — 20%

Total : 100%

*[** as supporting evidence for this part you may optional submit a video file of your system in operation over a sample of the data – make sure your video shows both the colour and disparity images. This can be constructed using OpenCV directly or any tool of your choice. File size must be less than 10Mb in size, video format in use must playback in the VLC tool – www.vlc.org]*

Submission :

You must submit the following:

- Full program **source code** for your solution to the above task as a **working python script** meeting the above ***"additional program specifications"*** for testing.

- **Optional video file** showing general performance on some of the example data (see above)
- **Report (max. 750 words!, remember - it can always be shorter than this)** detailing your approach to the problem and the success of your solution in the task specified. Provide any illustrative images (as many as you feel necessary) of the intermediate results of the system you produce *(overlays, results of processing stages etc.) Remember that any titles, captions, tables, references, and graphs do not count towards the total word count of the report.*

  Summarise the success of your system in detecting the planar road regions in the data set. Submit this as a PDF (not in any other format).

Make it clear in the initial comments of your source code how to run your Python script. Your executable must run on one of the lab based PCs (either on Windows via OpenCV 3.3 or Linux via OpenCV 3.1/3.3) – ensure compatibility before submission.

Plagiarism: You must not plagiarise your work. You may use program source code from the provided course examples, the OpenCV library itself or any other source BUT this usage must be acknowledged in the comments of your submitted file. Automated software tools (e.g. https://theory.stanford.edu/~aiken/moss/) may be used to initially detect cases of potential source code plagiarism in this practical exercise which will include automatic comparison against code from previous year groups. Attempts to hide plagiarism by simply changing comments/variable names will be detected.

You should have been made aware of the Durham University policy on plagiarism. Anyone unclear on this must consult the course lecturer prior to submission of this practical.

To submit your work create a directory named as your username (e.g. *cxfh123*). Place all required files in this directory using, **ZIP** compress/archive this entire directory structure(not rar or .z7 or anything else as this breaks the automated extract/test tools) and submit it via DUO *(late submissions will be penalised following school policy).*

Submission Deadline :– 2pm (UK time) on 1st December 2017

[1] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination Invariant Imaging: Applications in Robust Vision-based Localisation, Mapping and Classification for Autonomous Vehicles," in Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

[2] J. M. Alvarez, A. Lopez and R. Baldrich, "Illuminant-invariant model-based road segmentation," Intelligent Vehicles Symposium, 2008 IEEE, Eindhoven, 2008, pp. 1175-1180.