Image processing based **road surface variations detection and modelling for ADAS**

Problem Statement:

I am trying to device a way using image processing on images captured with a single camera, which could be one's phone camera placed on the dashboard/handle of a two wheeler to detect the potholes/depressions/elevations/obstacles, their approximate dimensions, including depth, and record their coordinates in the frame which can later be incorporated with GPS data for that route to help in ADAS (Advanced Driver Assistance Systems) and automatic driving. The motivation for this comes especially from the situations where the driver is unable to see the road surface such as at night due to the glare from other vehicles' lights or his own headlights stop working, etc. Also during floods, the driver is unable to see the road surface below and hence, a prior database of the coordinates and size (including depth) of the pit and road variations will help the driver to plan his trajectory of moving on the road for a smooth and more importantly safe drive.

Related Work done before:

Till now some papers have been published which talk about pothole detections like:

1. Pothole Detection with Image Processing and Spectral Clustering (I implemented this paper first.)
2. Automated Pothole Distress Assessment Using Asphalt Pavement Video Data
3. There was a survey of potholes and there classifications based on shapes, sizes, etc. (basically that supports that most of the ellipses can be circumscribed by a rough ellipse which I am using.)
4. Detecting potholes using simple image processing techniques and real-world footage System and Method for Detecting Potholes based on Video Data

Some interesting papers, one might want to experiment with:

1. Adaptive detection and localization of moving objects in image sequences
2. Segmentation and Recognition using Structure from Motion Point Clouds, ECCV 2008

However, all the current methods are limited mostly to single frames and cannot determine the severity of potholes. They also do not classify the nature of the road distress as to whether it is an elevation or a depression.

My approach:

The algorithm I used is giving fairly correct results for this classification on some test images.

My algorithm tries to do the following:

1- Segment out the road from the scene using the curve fitting technique. This detects both the asphalt roads and the muddy/sandy roads or paths.

I take the below 3 samples and plot their RGB values and use the curve fitting/regression app in Matlab to say that the pixels of the image which are close to the pixels in the below samples within a particular threshold constitute the road or the path and eliminate pixels of the other colors. This successfully eliminates all the non-road pixels. Additionally, this step will eliminate even those objects (obstacles) which are on the road.

On doing Convex Hull of all the non-eliminated pixels, the road section and the eliminated objects on the road also come in the region in the convex hull and can be easily termed as an unexpected obstacle with minimum computation to give alarm to the driver.
Also depending on whether the majority of road pixels resemble sample 1 cluster or sample 2 or 3 cluster, I classify the road in view as asphalt or muddy road and redo the curve fitting with only one of the below samples for more precise road segmentation.

2- After getting the road, I try to segment the smooth part of the road on the basis of entropy measure and variance in the pixel values over kernels of different sizes. This gives the part of the road with no depressions and bumps. The regions with rough road and variations in surface are high in entropy and the others less in entropy.

To add to this, I also tried finding the depressions and variations using expectancy of the road.

3- The Region of Interest thereafter if the part of the part with higher segmentation than an adaptive threshold and I get this using Convex Hull.
4- In the paper mentioned above, they use Spectral Clustering and Seed filling (which requires human input), I tried this and the results are in fig. 2.6 (using bwconncomponents)

A more useful technique I found was thresholding the image based on Otsu's method, which was also used in the same paper. However, in applying the Otsu's thresholding in some cases, a global threshold gives good results and in some cases the adaptive threshold gives good results therefore, for every image we choose which type of thresholding to apply depending on the variation in the illumination or the Value part of the HSV of the image. If greater than a certain threshold, adaptive thresholding is used or else global thresholding. Different thresholding may be required to be applied to different sections of an image too. This is very important since illumination invariance is very critical to the next step.

5- Wherever there is any depression or bump (I am considering only speed breakers here, which span across almost the entire road width, otherwise it gives lots of false positives) a certain intensity change occurs which is shown by thresholding. In the ROI, I try to relate this intensity change to potholes and fit ellipses to that. I found that fitting ellipses gives fairly good results and is less computationally intensive than finding exact boundaries of the depressions (using bwboundaries command in Matlab)
6- In fitting the ellipses, I try to fit 2 ellipses to each depression, one ellipse, the Red one, is the ellipse which tries to circumscribe the boundary of the depression from the plane of the road (Assumption: the road excluding the depressions is a straight plane) and inside this ellipse, another ellipse, the Blue one, which circumscribes the lower part (plane) of the depression. This is more clear in fig. 1.4
7- The usefulness is that the distance between the inner ellipse (blue one) from the plane of the road (roughly circumscribed by the red ellipse) which is represented by the dark region in fig 1.2 can be used to approximate the depth of the depression. All the white region in the image I take as a straight plane (it may be inclined).
   In general, all the variations of the road surface can be roughly modelled on a mesh taking into account the thresholding and the occlusions.
8- In some cases, the inner ellipse is not found, when the depression:
   i. has a smooth paraboloid kind of shape
   ii. is a bump
   iii. is a speed breaker

   but if it is found then it is most certainly a pit.
9- In the logic of relating the ellipses in the threshold image to the road variation classification, I observe that the variation is:
   i. A depression if the circumscribing ellipse, the dark region is observed more on the convex inner side (not reliable though, will show the images where it failed)

ii. A bump or elevation if the darkness is more on the concave inner side.

10- On the basis of occlusion, the variation is:
   i. A depression if the pit's internal texture is occluded on the inner concave side
   ii. A bump if the variation's texture is occluded on the convex side
   I decide whether the variation is occluded or not based on by convolving an expectancy kernel on the region around the ROI (I don't know what it is called)

11- The depth of the variation is directly proportional to the dark region width in between the blue ellipse boundary and the plane of the road. (I don't know how to explain this here but I have a reason behind it) (Also, I actually wanted to find 2 ellipses and then take the depth proportional to the distance between the inner ellipse and outer ellipse, but the outer ellipse is not always coming correct for all images so now this)

Taking this approximation of depth and taking the variation boundary to be lying on the plane of the road i.e. X=h where h is the height of the camera center from the plane of the road given by the Y-Z plane, where Y is the direction along the camera's horizontal axis and +ve Z the direction away from the camera on the line joining the camera center and its principal point, I project all the image points on a 3D mesh using the Camera Matrix M (M consists of 2 matrices, the intrinsic matrix and the extrinsic matrix, we know the intrinsic matrix since I have done the calibration of the camera, and which is required to only be done once for any camera, better still it may be pre-calibrated and for the extrinsic parameters i.e. Rotation matrix R and translation matrix (t), we need only t since we are considering single images and I have written the code for that. This is also a one time job. Knowing the camera matrix now, I can easily find out the perimeter (the approximate diameter or major/minor axis) of the depression/elevation. I tried this, the 2nd image set is what I took on my own at a road in the campus, the estimated length of the pit by the code came as 965.11 mm while the actual dimension of the bigger pit in the picture is around slightly more than a meter (along its longer side).

If the depression is segmented correctly, this step will mostly always give correct results.

1- Now, I project my image on the 3D coordinates using the camera matrix and Matlab inbuilt functions taking the entire road as a plane and varying the Y component of only the depression/elevation to recreate the approximate road surface as a mesh in Matlab. So, using a single image, we would have a rough mesh of the road in hand.

As this will be done on a video, I am using Mean-Shift algorithm to track the depression once detected in a frame in the subsequent frames of the video (lots of good codes for it are available on net) and will simultaneously try to find new potholes in the frames minus the regions which are already detected and are being tracked. Tracking is working fine. (I recorded a very bad and shaky video but it tracked the pothole well for some time). I am using Mean-Shift because it is a Kernel-based method and paper 2 explains with reasoning why kernel based methods (out of the 3 methods they told) are better esp. in our case. But, we may have to enlarge the tracked ROI as the pothole when coming closer to the vehicle will appear bigger in each frame.

2- Now in the subsequent frames, in the tracked ROI, new image points which were earlier occluded will appear. Depending on whether the new points are appearing on the concave inner side or convex side of the ellipse, we can further assure whether it is a depression or an elevation. And we have to plot these points further in the same 3D plot.
   (Additionally what can be done is, we after projecting the newly appearing points on the mesh, we reproject the 3D mesh to an image and try to match that image with the actual road frame and check how closely they match, if they vary beyond a certain threshold, we discard the previous mesh and make a new mesh entirely or do RANSAC to retain only some pixels in the ROI which tend to match the pothole well (I have a few ideas but did not try them, will be very happy if I can discuss them with you and you can help with them. The problem I think here will be to

decide which image pixels and new and which are old since I want to only plot the new ones to the old mesh.)

12- Therefore, by the time the vehicle passes across a pothole, a fair enough mesh should be constructed.

Practical problems: The steps till 6 (maybe even 7) are computationally not intensive, can be done in real time and are good enough to assist the driver to navigate on the smoothest path on the road and even saving the desired route gradients and give immediate results but plotting takes time. The steps involving creating meshes and iteratively improving the mesh takes a lot of time, even tracking is not as fast as a vehicle moves.

13- Thinking of it as a product, we may also have to do real-time velocity estimation and estimate the distance and positions of the potholes along the path (if we are not taking the speedometer data from the vehicle or say, in case for a person walking who does not know its speed) to play the saved routes and show potholes in the next run or when he cannot see the road. I found some papers for that in particular VELOCITY ESTIMATION OF A MOBILE MAPPING VEHICLE USING FILTERED MONOCULAR OPTICAL FLOW. I have implemented optical flow before (codes are available) but not for the roads, will try to do that before coming back. Also, if we use optical flow, the flow vectors around the depressions and elevations will give confused vectors than for a smooth road, but I am not sure will try.

I tried doing the same with LSD-SLAM by TUM group for the road surface reconstruction, it faired badly, gives no variation and it is too slow, certainly not applicable here and hence this approach is much better.

RESULTS:

Fig. 1.1 to fig 1.4

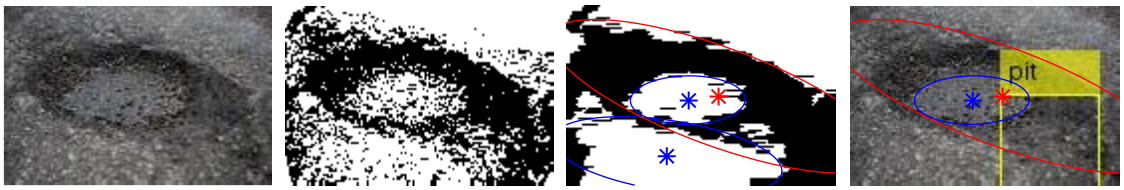
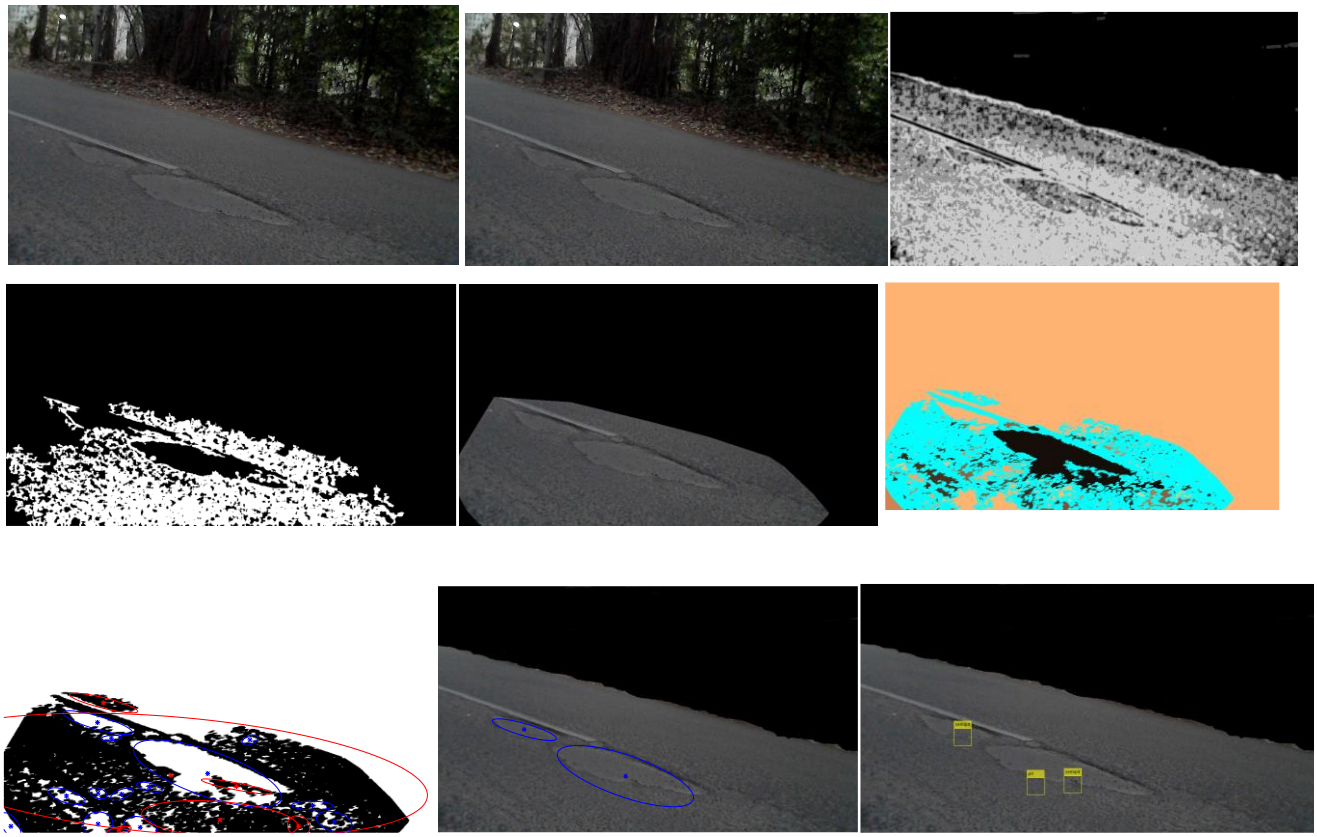


Fig. 2.1 to 2.9



Actual size- around a meter long Estimated size- 965.11mm

Fig.3.1 to 3.5

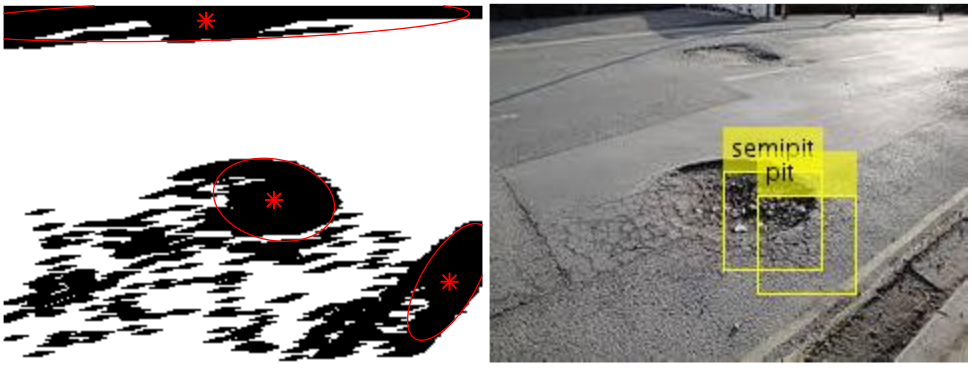


Fig. 4.1 to 4.5