Simulation

Special Issue on Driving Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
2016, Vol. 92(5) 407–415
© 2016 The Author(s)
DOI: 10.1177/0037549715627693
sim.sagepub.com
SAGE SCS

Changing lanes: generating steerable environments in video-based driving simulation

Michael Brogan[1], Charles Markham[2], Sean Commins[3] and Catherine Deegan[1]

Abstract

This paper describes an approach that allows for a steerable environment to be generated directly from a video for the purpose of integration with a video-based driving simulator. As the range of steering motion in a driving simulator is relatively limited, a pseudo-three-dimensional approach can be taken. This method requires only a single image sequence or video, acquired by any type of imaging system along a road. No three-dimensional, stereo or visual odometry data is acquired or calculated. An experiment is then presented, involving multiple lane change requests, where participants are asked to change from the left-hand lane to the right-hand lane and back again.

Keywords

Video-based, driving simulation, steering control

1. Introduction

The aim of this paper is to introduce a framework that allows for a steerable element to be introduced to a video sequence acquired by a low-cost witness camera. While this is the technical aim of the paper, the overarching aim of the research is to produce a methodology by which any person may acquire a video of a road in their local area and introduce the video into a driving simulator consisting of common equipment, such as a laptop and PC gaming steering wheel and pedals. The rationale behind this is to allow vulnerable members of the population, such as recovering stroke patients, the ability to drive around familiar environments that are presented with high-fidelity visuals. For example, a recovering stroke patient may be able to ‘drive’ the road to their local shopping center. Although graphical environments are becoming more and more realistic, a video sequence remains the most realistic visual representation of a real world environment.

The paper consists of seven sections: Section 1 presents an introduction to the research, and gives a structural breakdown of the remainder of the paper. Section 2 gives a background to the research. Section 3 presents the methodology of the approach developed, including the generation, interfacing and geometry of the simulator environment. Section 4 details the limitations of the environment. Section 5 describes the testing and results undertaken for this paper. Section 6 presents a discussion based upon the contents of the paper, and the paper concludes with Section 7, which describes the conclusions and future work of the research contained in the paper.

2. Background

Using video as a basis for a driving simulator’s visual cue stream is relatively rare, with the vast majority of simulators using a graphical-based environment. There are two primary reasons for this: the ease at which full three-dimensional graphical models can be generated, and the difficulty in introducing a form of interactive control over a pre-recorded video sequence. There are several existing video-based driving simulators, examples being Bredif,[1] Brogan et al.,[2] De Ceunynck et al.,[3] Heras et al.,[4] Ono et al.[5] and Sato et al.[6]

The simulator described by Ono et al.[5] uses a combination of photo-textures and graphical models to create a sense of high fidelity, but relies on graphical models for population and execution of scenarios. The image data

[1]Department of Engineering, Institute of Technology Blanchardstown, Ireland
[2]Department of Computer Science, Maynooth University, Ireland
[3]Department of Psychology, Maynooth University, Ireland

Corresponding author:
Michael Brogan, Department of Engineering, Institute of Technology Blanchardstown, Dublin 15, Ireland.
Email: michael.brogan@live.ie

acquired are used to texture the general environment in the far distance, but all near-distance features, such as roads, traffic and road boundaries, are graphics-based.

The simulator described by Bredif[1] utilizes data acquired from a high-cost mobile mapping system to generate a wire-frame environment. This environment is generated using data returned from a LiDAR laser-scanning device. Once generated, this skeletal structure can be textured using multi-view imagery acquired by the mapping system's cameras. This method produces a highly detailed photo-based environment that allows users to drive around a three-dimensional environment. As the generated environment is three-dimensional, the benefits associated with traditional graphical-based environments remain present; scenarios can be introduced, and full control over the driving simulator world is present. These systems have introduced photo-based environments to driving simulation, allowing the user to steer on or around textures, as opposed to through video sequences. The steering mechanisms are similar to previous graphical-based simulators, with the graphics having been replaced with photographic textures.

The driving simulator described by De Ceunynck et al.[3] introduces video sequences for the purpose of road sign evaluation A video of a road was acquired, and different road signage was introduced to the video via a four-step process: (1) an original 1920 × 1080 video sequence is optimized in terms of contrast and brightness; (2) a camera-tracking algorithm is applied to identify reference points in a video frame, using specialized three-dimensional software; (3) three-dimensional models are inserted into the video using these reference points; and (4) the models are rendered and masked, so they appear realistic in the resulting sequence.

Research has used non-steerable videos to demonstrate high correlations among video, graphical model and ground-truth speeds,[2] as well as augmenting different video sequences.[4] However, without the ability to steer, the full dynamic behavior of driving cannot be realized. Such a capacity would allow for driver speed, position and the effect of one on the other to be measured using a video-based visual cue stream.

This paper describes a method by which a pseudo-three-dimensional photo-realistic video-based model can be generated to allow for a steerable environment to be generated without the need for stereo images, synchronization, calibration, correspondence or three-dimensional reconstruction. However, the absences of these introduce some constraints that will be discussed, including the introduction of distortions around road boundaries, based upon lack of sufficient data.

3. Methodology

For the purposes of developing and testing the technique described in this paper, an off-the-shelf Mio MiVue 388 witness camera was mounted on the internal side of the windshield of a standard vehicle, and a two-minute video of a road was acquired. This video was in high-definition format (1920 × 1080 resolution with a frame rate of 29.97 fps).[7]

The selected route was the link road between the M3 motorway and R147 rural road in Kells, County Meath, Ireland. It is shown in Figure 1 alongside an example of the road scene as acquired by the camera.

For the purposes of testing the technique described in this paper, a Microsoft Windows-based notebook with 15.6-inch, 1366 × 768-resolution widescreen display was used, with a Thrustmaster gaming steering wheel.[8] An example of an alternate view being generated using this technique is shown in Figure 2.

3.1. Environment generation

Generation of a full three-dimensional environment requires the acquisition of multi-view geometry of a scene, commonly using two camera views (stereo) or multiple camera views. The disparity in the resultant images can be used to generate a dense stereo description of the scene, called a depth map. This map is a grayscale description of the depth of a scene, where the lower the pixel intensity, the further the feature lies from the reference camera, and the higher the pixel intensity, the closer the feature lies to the reference camera.[9,10]

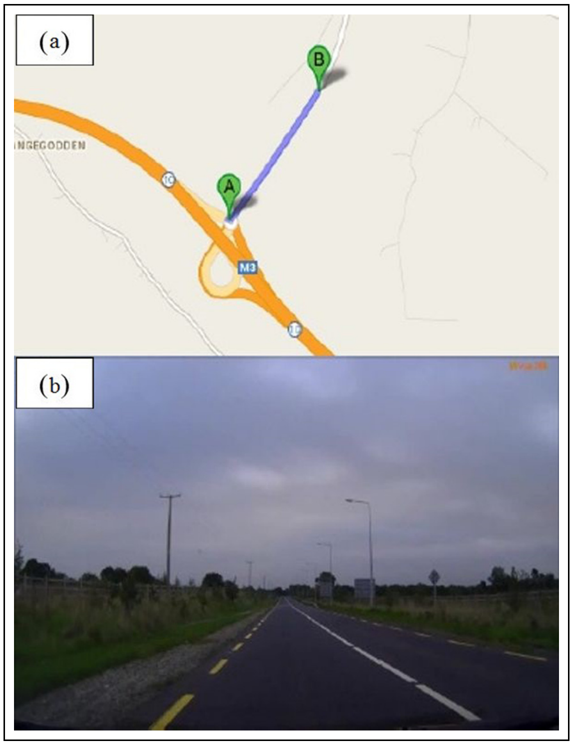

Figure 1. Selected data acquisition route (R147).

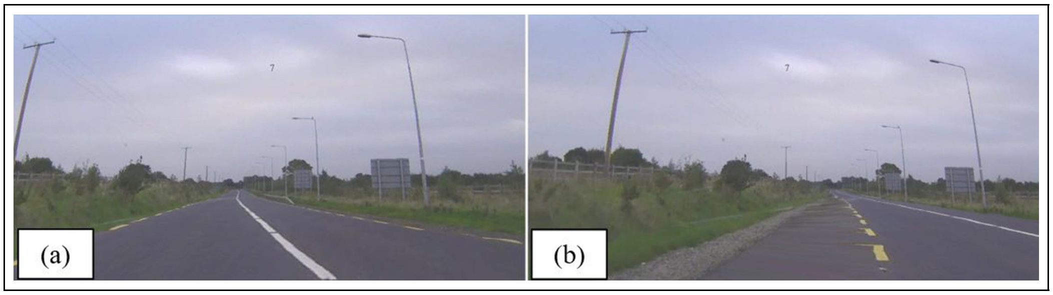

Figure 2. (a) A viewpoint of the original image. (b) A change in perspective by turning the steering wheel left.

Generation of genuine stereo depth maps requires the acquisition of stereo images, and is also dependent on a number of post-acquisition stages. This relies on the recovery of the three-dimensional transformation between the stereo cameras, in terms of the relative rotation and translation, and a series of parameters describing the distortions introduced to the images by the radial nature of the lenses. These parameters are estimated during a camera calibration process and post-processing, and are used to present the acquired images as undistorted and fronto-parallel. Matching corresponding features in each left and right image allow the disparity of the features to be found, allowing the calculation of a depth value, as disparity is a function of depth.[9–11]

In summary, to generate genuine depth maps, synchronized and calibrated stereo cameras are required, with a calibration accuracy that allows for a desired level of depth map quality. This allows depth to be assigned to all features that can be matched across each stereo image pair. In an environment where the geometry of the scene remains relatively constant, such as the road scene video acquired by the system on the test route, a dense stereo depth map can, instead of being generated using stereo-view or multi-view image sequences, be estimated using the horizon line. This allows for pseudo-depth perception to be inferred onto a single image.

By its nature, the line at infinity lies at an infinite distance from the acquiring camera. As the line at infinity is represented in images by the horizon, the horizon must first be identified in the single image. Once this is achieved, a grayscale gradient is generated, beginning at zero pixel intensity, representing zero, or minimum disparity, on the horizon, and increasing gradually to maximum pixel intensity to the bottom of the image, representing maximum disparity. The primary feature that requires a sense of depth is that of the road, and, as by its nature, the road will begin nearest the camera, and the edges of the road are parallel, will meet at the horizon.[9] This produces a mask whereby the depth map is masked so that a depth map describing the road only is estimated. An example of a depth map generated using this approach is shown in Figure 3.

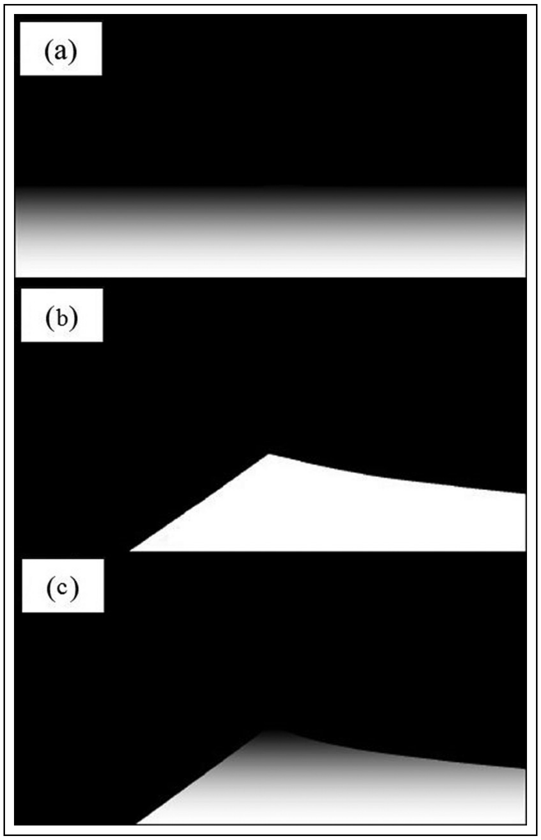

Figure 3. (a) Horizon-defined depth map: at this stage all features below the horizon line can be assigned a depth. (b) Road area mask: identifies the road area in the image. (c) Masking (a) with (b) allows only the road to be assigned a depth.

This allows a grayscale gradient to be applied to the corresponding road area, beginning at zero-pixel intensity on the horizon, and increasing gradually to maximum pixel intensity when the area closest to the acquiring camera is reached.

Once the depth map has been estimated, and the geometry of future images in the video sequence is approximately the same, the same depth map can be used for each image. The next step that must be undertaken is to infer depth perception on a video frame using this depth map.

This is done by layering the original red–green–blue (RGB) video frame over the depth map, essentially creating an RGB-depth (RGB-D) image from the original RGB image and the estimated depth map. Each value in the depth map is assigned a corresponding distance within a virtual graphical environment, and the virtual camera is situated at a constant distance from the displayed pseudo-three-dimensional image. Lack of sufficient data is evidenced in the pseudo-three-dimensional images as warping and distortion of areas close to the road edge, where the depth map was primarily estimated. This occurs where the depth map boundary does not map the road segment to the road plane, causing distortions between the road and non-road planes. The technique described in this paper was developed in the Open GL Utility Toolkit (GLUT) graphical development environment with OpenCV support.[12] Each pixel in the original image was assigned a three-dimensional polygon, with each polygon being assigned in terms of an x, y and z coordinate. The x and y coordinates are taken from the pixel location in the image, and the z coordinate is taken from the pseudo-depth map. A zero intensity value in the depth map will therefore correspond to a z coordinate of zero in the three-dimensional graphical environment.

3.2. Interfacing environment

To allow for steering around the three-dimensional image, an arcball camera view is linked to the driving simulator steering wheel position, thereby allowing for the view of the road scene to change based on the user's input. The implementation of steering within a video sequence is completed by sequencing the video frames, such that upon any pressure applied to the driving simulator's acceleration pedal, the next frame of the video is loaded, textured and displayed at the correct viewpoint, based upon the current orientation of the steering wheel.

The arcball camera used by the GLUT implementation is defined in terms of three three-dimensional vectors: the Camera Position vector, C, the Camera Look At vector, L, and the Camera View vector, V, and allows a virtual camera to orbit a fixed plane. The L vector is fixed on the central point of the image and describes the point around which the camera rotates. The V vector describes the direction of the virtual camera and remains fixed, as although the image and perspective of the scene may change, the planes onto which the images are being projected remain fixed relative to the direction of the camera.

The C vector describes the location of the camera on the arcball orbit in three-dimensional space. However, this can be reduced to a two-dimensional vector, as the y coordinate remains of a fixed value, as the virtual camera is representative of a driver's viewpoint, so relative motion along the y-axis can be disregarded. The C vector can therefore be reduced to two coordinates along the x- and z-axes: C_x and C_z. These two coordinates can then be used to update the camera position, and therefore the participant's perspective with respect to the angle of the steering wheel rotation. This update can be performed by a single rotation around the arcball camera's y-axis. A driver has two degrees of freedom, ignoring the limited vertical movement that a vehicle may experience when being driven over speed bumps, for example. A driver may move left or right (first degree of freedom) and forward or backward (second degree of freedom).

The second degree of freedom is addressed in the driving simulator by advancing, or regressing, through the video sequence. The first degree of freedom, however, is the one that governs the driver's perspective, that is, the one that allows a driver to change lanes. This degree of freedom is represented in the video-based driving simulator as the movement around the arcball camera's y-axis. Rotations around the arcball camera's other two axes are not required, as the rotations around these axes can be either discounted or accounted for using other aspects of the simulator environment.

Using the video-based environment, the relative translation of the camera can therefore be reduced from three dimensions to two dimensions, and the rotation reduced from three dimensions to one dimension. The camera position is described by the C vector, as shown in Equation (1):

$$C = [\, C_x \quad C_y \quad C_z \,]^T \tag{1}$$

The normalized steering wheel angle is shown in Equation (2):

$$\alpha = 180.0\left[0.5 - \left(\frac{v}{65,536}\right)\right] \tag{2}$$

where v is the current 16-bit steering wheel value.

The rotation matrix describing a rotation through α degrees around the y-axis, as defined by the steering wheel orientation, is shown in Equation (3):

$$R_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \tag{3}$$

The full transformation required to update the C vector position is shown in Equation (4):

$$C = R_y(\alpha)C \tag{4}$$

The angle around which the view is oriented is not the raw angular value from the steering wheel, but is instead a steering ratio value, as is reflected in real cars. The steering ratio chosen was 12:1, indicating that a turn of 12 degrees from the steering wheel is required to turn the vehicle wheels by a single degree. As the road on which

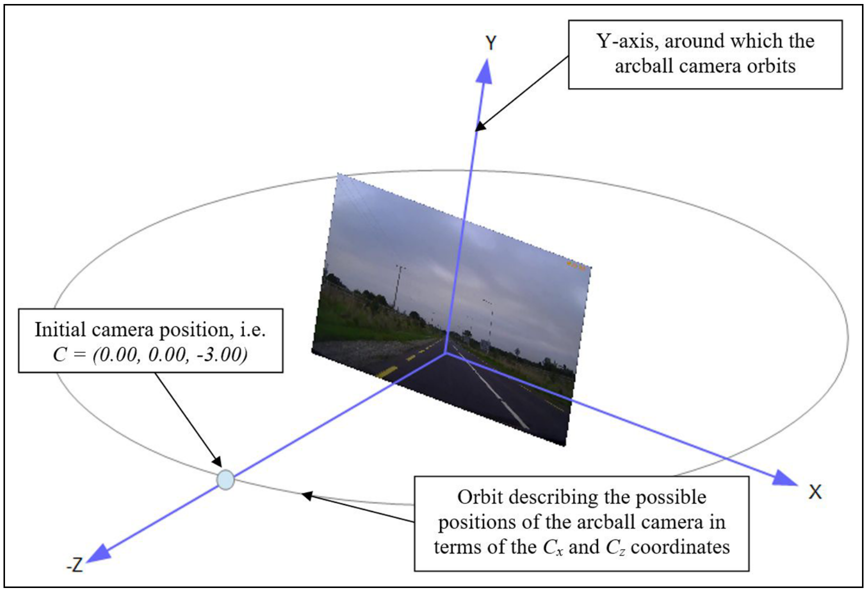
Y
Y-axis, around which the arcball camera orbits
Initial camera position, i.e. $C = (0.00, 0.00, -3.00)$
X
-Z
Orbit describing the possible positions of the arcball camera in terms of the C_x and C_z coordinates

Figure 4. Possible virtual camera positions in the driving simulator environment.

the video was acquired is a straight road, no angular correction data is required. If no steering control is introduced by a participant, the participant will travel the simulated road in a fashion similar to how the video was acquired.

3.3. *Environmental geometry*

The generated environment consists of a local world three-dimensional coordinate system, which is viewed using an arcball camera, centered on the central horizon point. Two two-dimensional planes are used to generate the pseudo-three-dimensional environment: the non-road plane, π_N, and the road plane, π_R. π_N will contain all non-road features and π_R will contain the road only.

This results in π_N being coincident with the XY_{World} plane, and π_R being non-parallel with the XZ_{World} plane, and creates the pseudo-three-dimensional environment that enables steerable video to be generated. The offset between the world coordinate system origin and the intersection of π_R with π_N is dependent on the position of the horizon line in the acquired video. The arcball radius is set as -3.0 units and the arcball camera is given an initial coordinate of (0.0, 0.0, -3.0). This results in the initial camera view being in the center of the left-hand lane. This is shown in Figure 4.

π_R is divided into 256 subsections, along the Z_{World} axis, each relating to a depth map grayscale intensity. This allows for the depth map to define which pixel intensities are to be mapped to the corresponding π_R Z_{World} axis line. The parameters of the arcball camera are defined in terms of its rotation and translation with respect to a fixed point in the viewing scene. That is, the projection of the world coordinate scene is defined in terms of three parameters: the fixed point, the camera's rotation around this point, and the camera's translation from this point. The fixed point can be described in terms of a 3-vector, with the position of the camera being described in terms of the C vector, with C_x and C_y describing the position of the camera relative to the surface of the viewing arc and the C_z coordinate describing the radius of the viewing arc from the point of focus. The camera's orbit around the y-axis is shown in Figure 5. Each change in the value of α updates the position of the C_x and C_z coordinates.

When $\alpha = 0^{\circ}$, the virtual camera plane will be parallel to π_N. In this case the original image will be displayed, with no redundancy visible. Lack of sufficient data in the viewed image plane is dependent on the value of α, and can be quantified in terms of the percentage of the road plane that increases with the increase in α.

The orientation of the steering wheel is relayed to the PC as a 16-bit value, giving a range of 0–65,535, with a 0 value representing the wheel at the leftmost position and the 65,535 value representing the wheel at the rightmost position. The value is normalized in the range of $\pm 90^{\circ}$, representing the full range across 180°. In the case where the arcball camera position is neutral (i.e., it lies in the same position relative to the imaged road as the acquiring camera), the projected scene will be the same as that of the original video. As the arcball camera's rotation changes, the viewpoint of the road scene will change also, resulting in a steerable scene.

4. Limitations of the approach

As the technique described in this paper depends solely upon the information contained in the single video

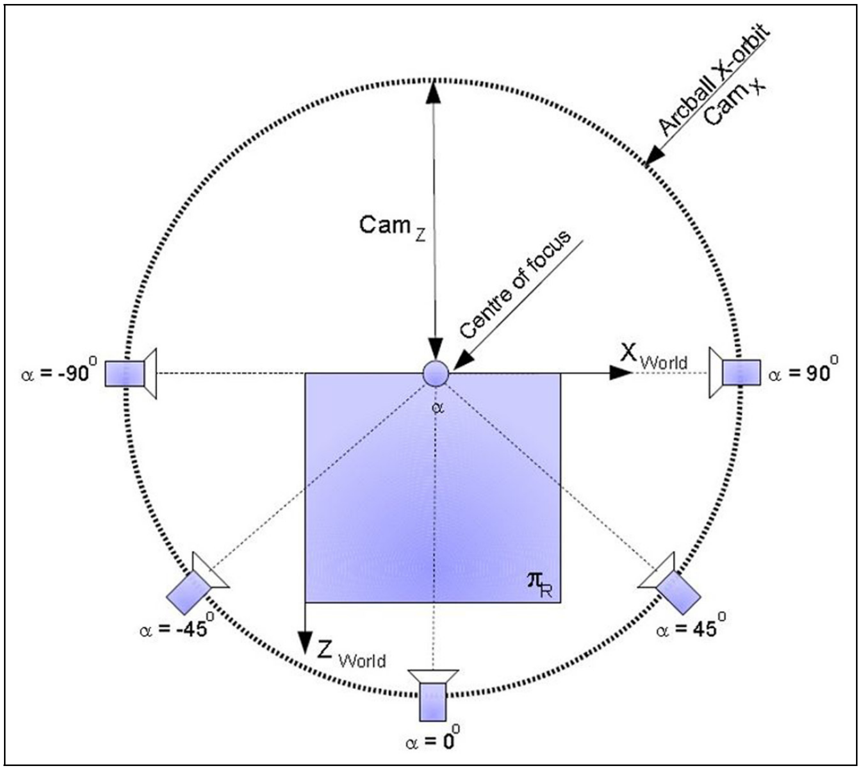

Figure 5. Orbit of the camera's *x* and *z* coordinates, showing the movement of the camera relative to the *XZ* plane, described by the angle α.

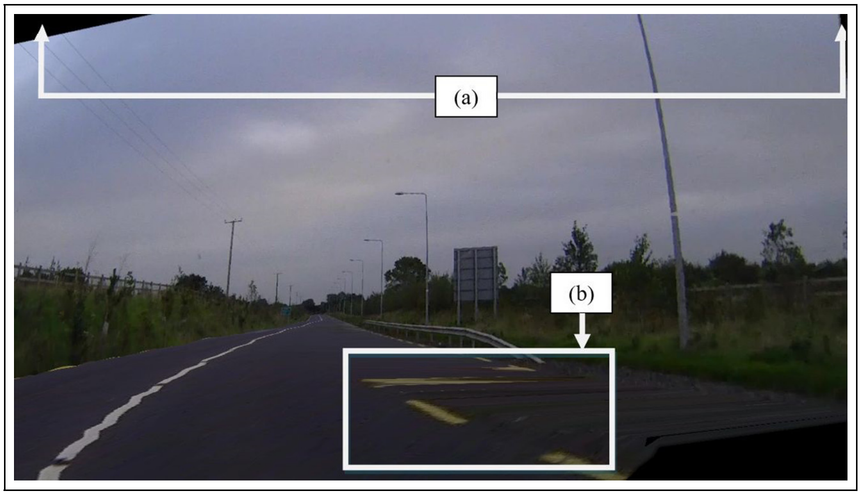

Figure 6. (a) Non-textured areas in the simulator environment caused by lack of sufficient data from the images. (b) Distortions introduced by incorrect road plane segmentation.

sequence, any areas that are not viewable will be evidenced as non-textured areas in the simulator environment. An example of this is shown in Figure 6. Incorrect road boundary estimations, due to the use of the depth map estimated using the first frame, may also cause distortions.

As the video sequence was obtained in one direction only, the ability to perform a U-turn and drive in the opposite direction is also lacking, although this could be addressed by the acquisition of a video of the road in the opposite direction. Road boundary distortions are evidenced where the estimated depth map either under- or

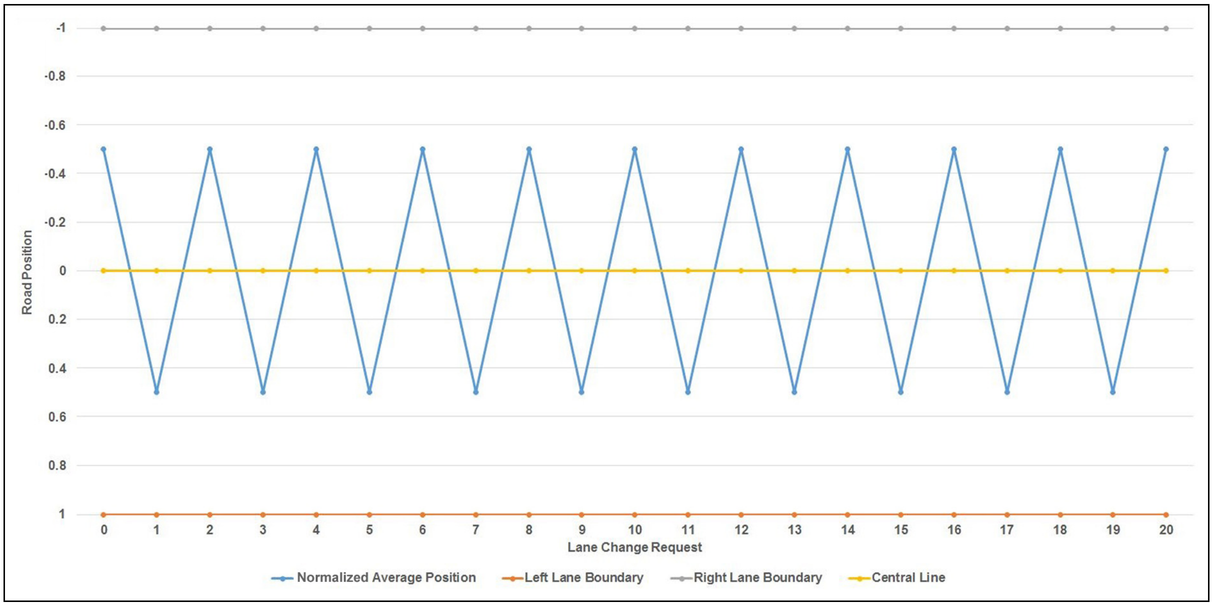

Figure 7. Normalized average lane position (22 lane changes per participant with 10 participants).

over-segments the non-road and road areas. This issue can be addressed by identifying the road area in each image, instead of relying on the single depth map, derived from the first frame of the video sequence.

No method for angular correction is presented, as the road examined is a straight road; however, such a correction may be introduced by generating a differential offset sign based upon the GPS data acquired by the witness camera, thereby forcing a participant to turn the steering wheel to ensure that the image plane remains approximately parallel to the virtual camera plane.

5. Testing and results

A lane change request experiment, where participants were requested to change lanes on the expiration of an onscreen timer, was conducted. This timer counted from zero seconds to nine seconds and then reset. Participants were instructed to change lanes upon the reset of the timer. Restrictions imposed on text-based displays using GLUT resulted in the timer being somewhat difficult to observe in terms of both size and color. Despite these limitations however, 186 of 220 lane-change requests were completed successfully. The 10 participant data set had an average success rate of 87%, ranging from 64% to 100% per participant, with a mean of 18.6 lane changes and standard deviation of 2.67.

The normalized average position of the 10 participants was calculated, with the threshold set at a camera position of 0.0 representing the center line. The middle of the left-hand lane was set as -0.5, and the middle of the right-hand lane set as $+0.5$. If the average position was between -1.0 and 0.0, this indicated that the camera position was left of the center line and set as -0.5. This was interpreted as the driver being in the left-hand lane, and if the average position was between 0 and $+0.5$, then this was interpreted as the driver being in the right-hand lane. This is shown in Figure 7.

Each participant began the experiment in the left-hand lane, and the camera position was recorded for each participant for the duration of the experiment. The timer resets alongside the individual responses to the lane change requests are shown in Figure 8. It can be seen that upon the expiration of the timer, each participant invoked some form of steering response.

6. Discussion

The approaches described in this paper allow for a video of a road to be acquired, and then integrated with a driving simulator, allowing a steerable element to be introduced, allowing for the perspective of the video to be changed while maintaining a realistic level of three-dimensional perception. The two main advantages of this approach are (1) the ease of which a video sequence may be acquired—a high-fidelity visual representation of a driver's local environment may be generated with ease, by using a low-cost video camera and (2) the environment is an exact recreation of the real world environment at the instance of video acquisition. These attributes may be of great

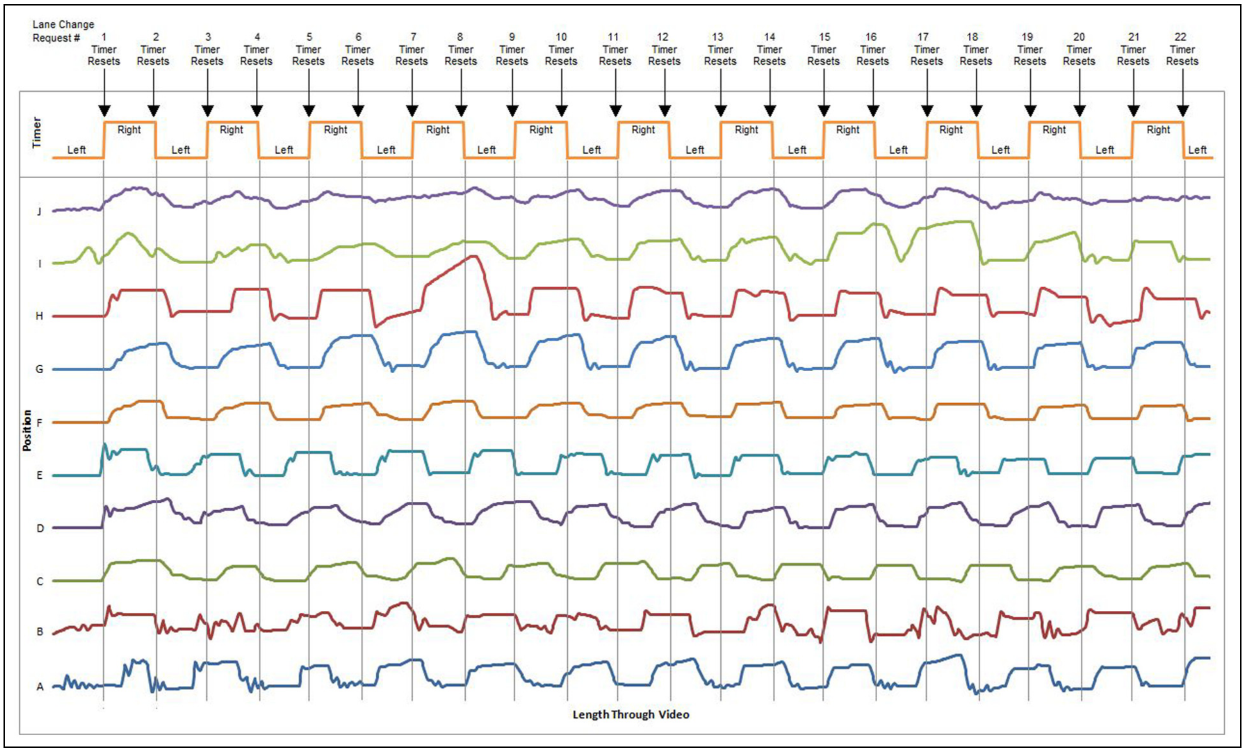

Figure 8. Participant lateral position when requested to change lanes.

importance when considering the training or re-evaluation of vulnerable members of the community, such as stroke patients.

The approach, as described in this paper, allows a video sequence as acquired by an off-the-shelf witness camera to be integrated with a basic driving simulator. An alternate approach, using a high-cost mapping system, as described by Brogan et al.,[13] introduces a video sequence using roads with bends. While the method described in the present paper requires only a single video sequence, the method described by Brogan et al.[13] requires a full mapping system. However, the angular offset signal generation method described by Brogan et al.[13] may be adapted for use in this present work.

A limitation of the present approach is inherent in the video acquisition tool used; a standard video camera may allow for a limited visual display to be used, for example, the laptop screen used in the present paper's experiment. An extended version of the approach may be to use three synchronized cameras: one recording the view of the road from the windshield and another two recording the view of the road from the passenger and driver side window. Image stitching among the three videos may be performed to increase the peripheral range of the resultant video, thereby allowing it to be used on an 180° screen, although this extension would require more expensive video acquisition systems, and issues with synchronizing video sequences from independent video cameras are regarded as non-trivial, especially when being acquired from a platform moving at standard road speeds.[14]

7. Conclusions and future work

This paper has shown that a single video sequence, acquired by a standard witness camera, can be adapted for use in a video-based driving simulator by estimating a single depth map that represents a road with an assumed constant geometry. Experimental testing of the approach showed that the average driver response to the change lane instruction was 87%.

Future work will consist of extending the existing approach such that a depth map can be estimated for each frame in the video sequence using the parallelism of the road edges and the horizon line. This will enable the approach to be implemented using any road geometry, whether it is constant or changing. This will also remove, or greatly reduce, the distortion issues associated with the incorrect segmentation of the road plane from the non-road plane, as highlighted in Figure 6(b). A comparison of this technique against the genuine depth maps generated using the stereo image data of the original mapping system will then be undertaken to compare and contrast the two methods. Behavioral testing of drivers will be the focus of further testing using the driving simulator once the full video steering component has been integrated with it.

Acknowledgements

The authors are grateful for the support of the Irish Research Council (IRC) for their continuing support, and also to the participants for their time.

This paper is an extended version of the paper entitled 'Steerable Video: Generating Video-Based Environments for Driving Simulation' that was presented at the Driving Simulation Conference Europe 2014 in Paris, France.

Funding

This work was supported by the Irish Research Council (IRC).

References

1. Bredif M. Image-based rendering of LOD1 3D city models for traffic-augmented immersive street-view navigation. In: *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, Antalya, Turkey, 2013.
2. Brogan M, Kaneswaran D, Commins S, et al. Automatic generation and population of graphics-based driving simulator: use of mobile mapping data for behavioral testing of drivers. *Transp Res Rec* 2014; 2432: 95–102.
3. De Ceunynck T, Arien C, Brijs K, et al. Proactive evaluation of traffic signs using a traffic sign simulator. In: *annual meeting of the Transportation Research Board*, Washington, DC, 2014.
4. Heras A, Breckon T and Tirovic M. Video re-sampling and content re-targeting for realistic driving incident simulation. In: *proceedings of the 8th European conference on visual media production*, 2011.
5. Ono S, Ogawara K, Kagesawa M, et al. A photo-realistic driving simulation system for mixed-reality traffic experiment space. In: *IEEE symposium on intelligent vehicles symposium*, Las Vegas, Nevada, 2005.
6. Sato R, Ono S, Kawasaki H, et al. Real-time image-based rendering system for virtual city based on image compression technique and Eigen texture method. In: *19th international conference on pattern recognition (ICPR)*, Tampa, Florida, 2008.
7. Mio. http://eu.mio.com/en_gb/mivue-388.htm (accessed 31 August 2014).
8. Thrustmaster. http://www.thrustmaster.com (accessed 31 August 2014).
9. Hartley R and Zisserman A. *Multiple view geometry in computer vision*. 2nd ed. New York, NY: Cambridge University Press, 2003.
10. Trucco E and Verri A. *Introductory techniques for 3-D computer vision*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 1998.
11. Bradski G and Kaehler A. *Learning OpenCV: computer vision with the OpenCV library*. Sebastopol, CA: O'Reilly Media Inc., 2008.
12. FreeGLUT. http://freeglut.sourceforge.net/ (accessed 31 August 2014).
13. Brogan M, Markham C, Commins S, et al. Dynamic perspective corrections for steering in video-based driving simulation using stereo depth maps. *Transp Res Rec* 2015; 2518: paper no. 15–1229.
14. Haughey S, Brogan M, Murray S, et al. Sensor integration for mobile mapping and feature analysis. In: *proceedings of the road transport information and control (RTIC) conference*, London, England, 2010.

Author biographies

Michael Brogan, MEng, is a software development engineer at Reflective Measurement Systems in Dublin, Ireland, and is completing his PhD thesis at the Department of Engineering in the Institute of Technology Blanchardstown, Dublin, Ireland, where he worked previously as an assistant lecturer.

Charles Markham, PhD, is a senior lecturer at the Department of Computer Science in Maynooth University, County Kildare, Ireland, and a principal investigator with the SimRG research group.

Sean Commins, PhD, is a senior lecturer at the Department of Psychology in Maynooth University, County Kildare, Ireland, and a principal investigator with the SimRG research group.

Catherine Deegan, PhD, is a senior lecturer at the Department of Engineering in the Institute of Technology Blanchardstown, Dublin, Ireland, and a principal investigator with the SimRG research group.