GPS Data Interpolation: Bezier Vs. Biarcs for Tracing Vehicle Trajectory.

Rahul Vishen[1], Marius C. Silaghi[1], and Joerg Denzinger[2]

[1] Florida Institute of Technology, Melbourne, Florida, USA.
rvishen2009@my.fit.edu, msilaghi@fit.edu
[2] University of Calgary, Calgary, Alberta, Canada.
denzinge@cpsc.ucalgary.ca

Abstract. Our target is a driving simulator application that is designed to generate a simulation environment that is, in fact, a recreation of a prerecorded driving experience. The simulator does not just replicate the original scenery, but allows the users to maneuver within a recorded environment. The idea is to video-record the environment while driving a vehicle, recording the precise geolocation of each frame. During simulation, the precise geolocation of each frame is important for generating smooth transition from a viewpoint to another.
The assumption is that the recorded frames are tagged with their precise geolocations. However, in practice this is not true due to difference in sampling rates of the hardware systems involved. Cameras record images at a higher frequency in comparison to the GPS (Global Positioning System) data from a GPS unit. To tag frames with relatively accurate geolocations we present two interpolation techniques that trace the trajectory of a recoding vehicle using the GPS data. We then compare the effectiveness of the two techniques by drawing a comparison with respect to the ground truth based on error registered in position and orientation.

Keywords: GPS, interpolation, driving simulation, Bézier curves, Biarc, trace trajectory

1 Introduction

Our target application is a driving simulator that is an attempt to move from a realistic experience towards a real one by creating a *driving circuit* using real world images that are recorded during an actual driving session [13, 4, 2, 10]. Most simulators available today create a realistic environment through graphically designed virtual circuits. The designers attempt to represent the real environment to the best of their abilities. However, it is difficult to include the smallest of details that may still be of some importance. Visual cues are one of the factors that play an important role in the validation of driving simulations [12, 17, 15, 22]. Studies suggest a direct correlation between the quality of the display environment and validation of simulation [26, 16, 30, 22]. The driving circuits in our target application are previously recorded driving sequences where users can maneuver through as they would in the real life. A driving circuit is generated using a setup of several synchronized cameras mounted on a surface vehicle. During simulation, for the system to produce a smooth transition from a viewpoint to another knowing the precise geolocation of each frame is critical.

Common interpolation techniques focus on optimizing length, or on guaranteeing smoothness. However, the constraints of the interpolation problem we face greatly depend on the process of recording a driving circuit. Through interpolation of GPS data we attempt to trace the trajectory that may have been followed by the recording vehicle. In our studied simulator the recording vehicle is a four wheeled automobile which is subject to a limit on the maximum supported acceleration. For example, we know that a human being in such a vehicle can only sustain an acceleration of some $4.0g$ [32]. Therefore we have to be able to guarantee an upper bound on the trajectory curvature.

In Section 2 we give an overview of the work relevant to our problem. Sections 3.1 & 3.2 , present a technique that uses piecewise cubic Bézier curves based and biarc based interpolations techniques, respectively. In Section 4 we evaluate the techniques with respect to the ground truth.

2 Related Work

Previous attempts to build such simulators are presented in [13, 4, 2, 10]. A type of approach is used in the Aspen Movie Map [23], where four cameras are placed at a 90° angle interval on a circular disk. However, users experiencing simulation in the virtual environment do not have the ability to change their viewpoint. An improvement to this work is suggested in QuicTime VR [7]. The cameras capture an outward cylindrical projection [28] of the view around the vehicle, creating a cylindrical environment. We use a similar approach to recording a driving circuit. Circuit images are recorded while the vehicle is driven following a desired route. During simulation the recorded cylindrical projections allow the system to generate a virtual world where the users have the ability to look around. However, the users can only look around from a fixed point inside a cylindrical environment. To be able to navigate around in an environment, a solution using omni-directional cameras is suggested in [8, 29, 14, 33]. The common assumption is that the recorded frames are tagged with their precise geolocations [31]. Although the subproblem addressed in [33] does not require a GPS to allow transitions within a recorded viewpoint, for the solution to work with the full system, each recorded projection needs to be tagged with its precise geolocation. For example [14, 29], suggest using a GPS sensor to get precise geolocation for the outdoor image recordings. However, in practice this is not easy due to difference in sampling rates of the hardware systems involved. Cameras record images at a higher frequency in comparison to the data from a GPS unit. This makes an impediment for tagging every image projection with its precises geolocation. As a result, between any pair of consecutive GPS tagged cylindrical frames we have a set of projections that are not tagged with their precise geolocation.

Trajectory Compression: Essentially the problem we face here is of tracing the trajectory of a vehicle using only the GPS data, and represent the trace with a continuous curve. A relevant work is trajectory compression, a.k.a, trajectory encoding [20, 21, 36]. The problem of compression is to reduce the GPS data size by removing some recorded measurements. In order to reconstruct the original trajectory one uses some sort of interpolation techniques. In [36], a parametric cubic function is proposed that obtains a spline between any two spatiotemporal data points. Each spatiotemporal data point contains position and the time at which the data point is recorded. The data point also has the recorded velocity. A similar approach is used in [21] to encode a trajectory. However, the techniques compress the data maintaining trajectory within a given accuracy bound. Extension to this is presented in [20] where an attempt is made to improve accuracy using clothoids. Clothoids are curves where the curvature varies linearly with its length [24, 35].

Path Planning: Another related work is that of path planning [11, 34, 9, 1]. Curvature constrained path planning, in presence of obstacles, is studied in [1]. The algorithm determines an obstacle free path from a point A to point B, for a point robot with a bound on maximal curvature. In [9], navigation algorithms are suggested that guide a robot to visit a set of waypoints while adhering to corridor constraints. The algorithms use piecewise-Bézier-curve to represent a path between a pair of consecutive waypoints. The path segments are joined in a manner such that the result is a C^2 class continuous curvature curve.[3] Bézier curves are also used in [34] path smoothing. In addition, a bisection method based approach is suggested to subdivide Bézier curve segments in order to respect maximal curvature constraints. On the other hand a different approach is used in [11] exploring B-splines for real-time path smoothing.

Bézier Curves: Given a set of points, $(P_0, P_1, ..., P_N)$, parametric definition of a Bézier curve, of degree N, is given by [34]:

$$P(t) = \sum_{i=0}^{N} P_i \binom{N}{i} t^i (1-t)^{N-i}, t \in [0, 1] \tag{1}$$

P_0 and P_N are the end points of the curve, the remaining intermediary points are the control points that do not necessarily fall on the curve. From interpolation perspective the major drawback of Bézier curves is that they only approximate the control points rather than pass through them, an important property for interpolation. This can be addressed by taking a piecewise approach as seen in [34, 9]. Using the piecewise approach a curve is constructed by concatenating several Bézier segments. Although the resulting curve is a C^0 continuous curve, higher degree of continuity can be achieved by manipulation of control points as suggested in [9].

[3] In terms of parametric continuity, a C^n class curve is a curve whose first through n^{th} derivatives are continuous [3].

Biarc Curves: Also know as piecewise-circular curves, Biarcs were introduced by Bolton in 1975 [6]. Biarcs have been extensively used for geometric modeling [5] [18] [25]. In certain application such as geometric modeling for computer aided manufacturing, machine tools are better suited to move along a circular path or a straight line [25]. A biarc is a curve consisting of two circular arcs that

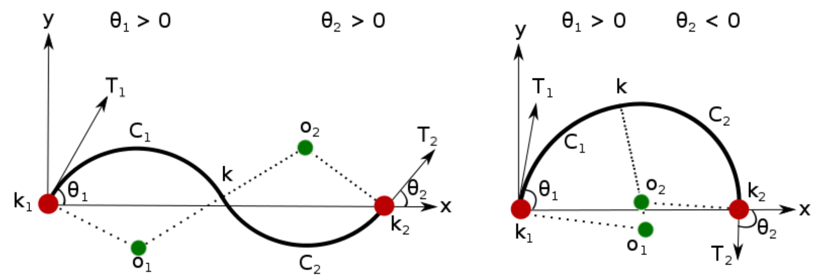


Fig. 1: S- and C-shaped biarcs and their tangents.

meet at a point where they share a common tangent vector. The shape of a biarc between two end points depends on the tangent vectors defined at these points and the vector connecting the end points. Biarcs have been classified into two categories, C-shaped, and S-shaped (See Figure 1). Figure 1 shows examples of the biarc curve fitting given two points along with their direction tangents. The biarc in this figure consists of circular arcs, C_1 and C_2, joining two end points, k_1 and k_2, which have associated tangent vectors, $\overrightarrow{T_1}$ and $\overrightarrow{T_2}$ respectively. The circular arcs C_1 and C_2 share a common tangent vector at point k. As shown in the figure, if θ_1 and θ_2 have the same sign, we have an S-shaped curve (Figure 1, left image). If the signs are opposite, we have a C-shaped curve (Figure 1, right image). Given the points k_1 and k_2 with their associated tangent vectors, $\overrightarrow{T_1}$ and $\overrightarrow{T_2}$ respectively, the problem of biarc curve fitting is to find the positions for the centers O_1 and O_2 of each circular arc C_1 and C_2 respectively. Versions of bi-arc algorithms are described in [27, 19].

Unlike the trajectory compression problem, for our target driving simulator application we need higher positional and orientation accuracy. On the other we share two of the most common constraints found in the path planning problems - bounded curvature, and continuous curvature path. In this paper we propose two techniques that, given accurate GPS data, can be used to interpolate GPS coordinates for the previously untagged image projections. First, we present a piecewise cubic Bézier based interpolation technique where for each Bézier piece the placement of each of the two control points is a function of vehicle velocity as recorded by the GPS. Second, we present a piecewise Biarc based interpolation technique that, given recorded direction vectors for two GPS coordinates, guarantees a continuous curve with the minimal curvature by concatenating two arcs. We evaluate the results based on ground truth.

3 Interpolation Techniques

We are given a sequence of images, $(I_0, I_1, ..., I_Z)$, captured by a recording vehicle. Each image I_i is tagged with a unique GPS data record, $G_i = \langle P_i, V_i, t_i \rangle$. $P_i = \langle x_i, y_i \rangle$ is a GPS point (coordinate), where x_i is the longitude and y_i is the latitude, and V_i is the recorded velocity at time t_i. Due to movement and to difference between the data sampling frequencies of the camera and of the GPS sensor, some of the images are not tagged with GPS coordinates acquired within the last frame period. As a result, between a pair of images (I_l, I_n), $l < n$ that are tagged with their accurate GPS points P_l and P_n, we have a sequence of images, $(I_{l+1}, ..., I_m, ..., I_{n-1})$, with $P_m = P_l$, where P_m is the GPS point recorded for image I_m. The assumption here is that the GPS sensor has no positional error. The problem is to trace the trajectory of the recording vehicle in order to deduce the geolocations for the previously untagged images.

3.1 Bézier Based Trajectory Interpolation

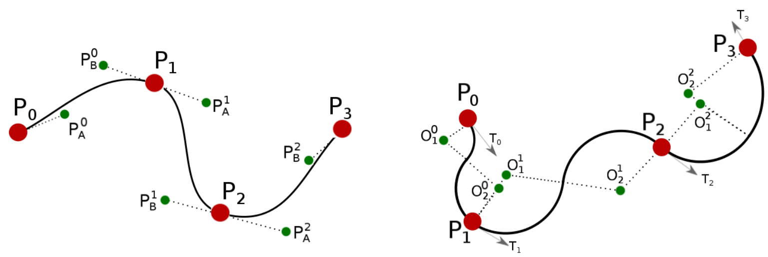
We take the piecewise approach to tracing the trajectory. Each pair of consecutive GPS points, (P_i, P_{i+1}), for $i = 0, 1, ..., N-1$, forms a single segment of the entire trajectory. Each segment, S_i, for $i = 1, 2, ..., N$, is a cubic Bézier curve that is defined by a sequence of four control points $(P_i, P_A^i, P_B^i, P_{i+1})$. Here P_i and P_{i+1} are a pair of consecutive GPS points, P_A^i and P_B^i are the intermediary control points. For each segment S_i, we compute the control points, P_A^i and P_B^i, such that when all the segments $(S_1, S_2, ..., S_N)$ are joined at points $(P_1, P_2, ..., P_{N-1})$ respectively, the trajectory is a C^1 class continuous curve. Each point P_i, except P_0 and P_N, has two associated control points P_B^{i-1} and P_A^i that are defined on a line parallel to $|P_{i-1}P_{i+1}|$ passing through P_i (See Figure 2a). This allows the concatenated trajectory to be a C^1 continuous curve. A similar approach is suggested in [9]. P_0 and P_N are the data end points and have only one associated control point, P_A^0 and P_B^{N-1} respectively. P_A^0 is placed on the line segment joining P_0 and P_1, and P_B^{N-1} is placed on the line segment joining P_{N-1} and P_N. The $||P_iP_B^{i-1}||$ and $||P_iP_A^i||$ is some fraction, f_i, of $||P_{i-1}P_i||$ and $||P_iP_{i+1}||$ respectively. The function is defined as:

$$f_i = F(s_i) = 0.5 \frac{s_i^2}{s_i^2 + x} \tag{2}$$

The value of f_i is a parametrized function of speed, s_i recorded at P_i that can be tuned through the value of x, where $x >= 0$.

3.2 Biarc Based Trajectory Interpolation

Similar to the approach used in Bézier based technique, a trajectory is constructed by concatenating several segments. Each segment, $(S_i \mid i = 1, 2, ..., N)$,



(a) Bézier trajectory (b) Biarc trajectory

Fig. 2: Trajectory examples

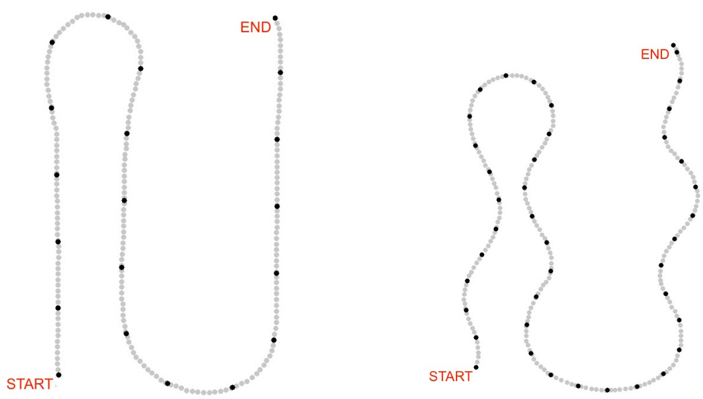
is a biarc curve that joins P_{i-1} and P_i. The algorithm we use for our experiments with biarc curves is described in [19]. Given a pair of consecutive GPS points P_{i-1} and P_i along with their respective direction vectors $\overrightarrow{T_{i-1}}$ and $\overrightarrow{T_i}$, the algorithm finds a biarc with minimal curvature difference between the two arcs. $\overrightarrow{T_i}$ is a shared direction vector between segments S_i and S_{i+1}, thus making the complete trajectory a C^1 continuous curve. The direction vector for each point P_i, except for P_0 and P_N, is set parallel to the direction vector $\overrightarrow{P_{i-1}P_{i+1}}$. For P_0 the direction vector is parallel to $\overrightarrow{P_0P_1}$, and for P_N the direction vector is parallel to $\overrightarrow{P_{N-1}P_N}$. Using the points and their corresponding direction vectors an intial piecewise biarc based trace of the trajectory is generated. We perform a hill-climbing search for a local optima while updating the direction vectors by $\pm\Delta$ radians ($\pm\Delta$ is reduced/halved on convergence up to a minimum value). An optima is reached when the global maximum curvature cannot be minimized. Each search iteration attempts to minimize the curvature difference between two consecutive biarc segments of a the trajectory.

4 Experiments

For evaluation of the proposed interpolation techniques we require highly accurate GPS data. High accuracy GPS units are expensive. Instead, we use an alternate approach to record geolocations that factors out GPS errors and the need to account for such errors. A recording vehicle marks the road surface while following a path at a constant speed of approximately 8 Kmph (5 mph). The marking represents the true trajectory of the vehicle on the Earth's surface, we call it the *ground truth* (See Figure 3). The ground truth is captured using a high altitude camera. Since we know the geographic location where the ground truth is recorded, using a GIS (Geographic Information System) tool, Google Earth, we are able to obtain a digitized version of the ground truth. Figure 4 shows the digitized version of the ground truth markings seen in Figure 3. The digitized version of the ground truth is a sequence of geolocations (latitude and longitude pair) with a sampling interval of 50 centimeters (See Figure 4). We want to be able to compare the two techniques with respect to their ability to reconstruct the trajectories in absence of the complete ground truth. To do so, a subset of the digitized ground truth, dark colored points in Figure 4, is used to simulate accurate input GPS points for our proposed interpolation techniques. This subset of GPS points can be sampled at some defined interval distance. Figure 4 gives two examples of the recorded ground truth and GPS points sampled at different intervals.

(a) (b)

Fig. 3: Ground truth marked by a recording vehicle.

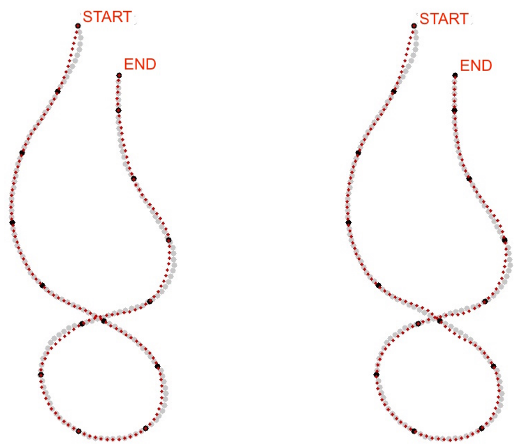


(a) Simulated GPS input sampled at 6 m.

(b) Simulated GPS input sampled at 3 m.

Fig. 4: Ground truth (light colored) and sample GPS input (dark colored).

The reconstructed trajectories are matched against the ground truth for accuracy in terms of position and orientation. Figure 5 shows an example of trajectory reconstruction using Bézier and biarc based techniques for a common input. The



(a) Bézier. (b) Biarc.

Fig. 5: Example of reconstructed trajectories (red colored).

experiments are conducted using 296 different ground truths which are sampled at various interval distances to simulate GPS input sets for our techniques. We compare the root mean square error (positional and orientation) registered for all the available ground truths when GPS points are sampled at different interval distances. Figure 6 shows the graph representing the average RMS (root mean square) errors registered in all the reconstructed trajectories with respect to position and orientation. To compute position and orientation error we compare the corresponding sample points from the ground truth and the reconstructed trajectory. The sample points are sampled at the distance of 0.5 meter. Position error for a single trajectory represents the average RMS displacement measured between the corresponding sample points from the ground truth and its reconstructed trajectory. Orientation error is the error in the bearing of the recording vehicle. Bearing of a point on the ground truth is the direction of movement registered by a recording vehicle at that point. In Section 3.1, we discuss the tuning parameter for Bézier technique (See Equation (2)). The parameter is used to fine tune reconstruction of trajectory when using the Bézier technique. Figure 7 shows the average errors registered at different values for the tuning parameter.

5 Conclusion

For our target application, the recording based driving simulator, we present in this paper two interpolation techniques addressing the need of tagging images with their accurate geolocation coordinates. The images and GPS data are

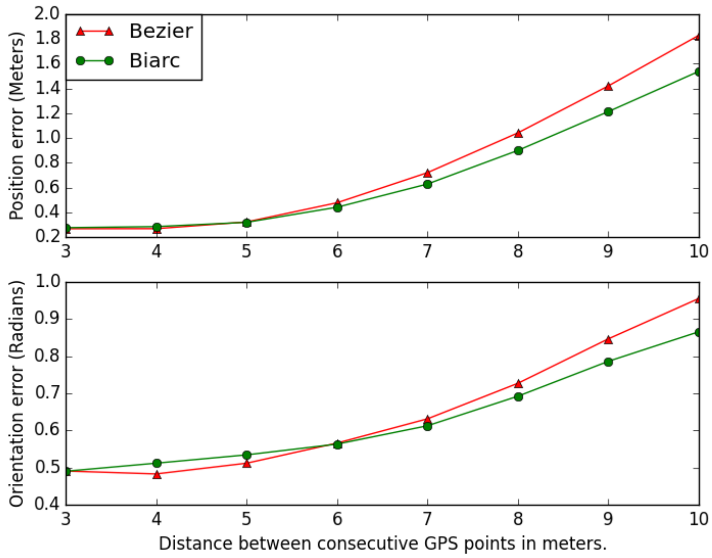


Fig. 6: Bézier and Biarc average RMS error.

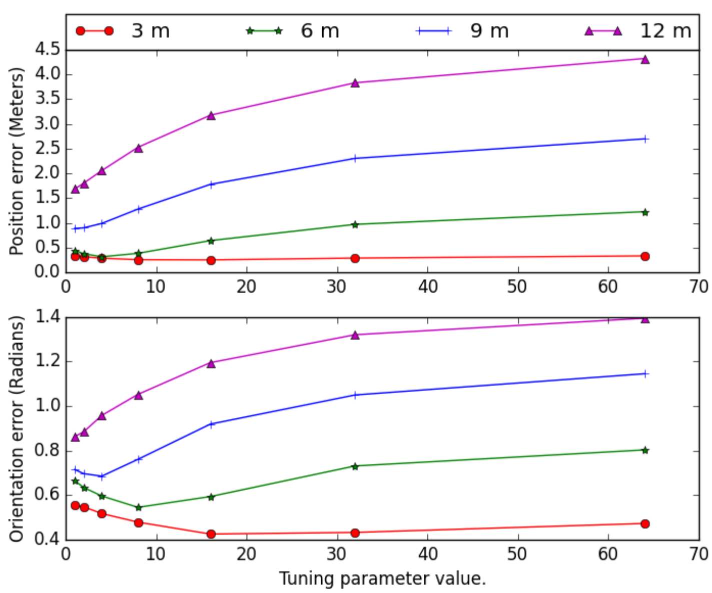


Fig. 7: Bézier tuning parameter

recorded by a vehicle while moving along a defined path. Due to differences in sampling rate of the hardware systems involved, some images are not tagged with their accurate geolocation. The two techniques, one based on Bézier curves and the other based on biarc curves, reconstruct the trajectories of the recording vehicle using the available GPS data, assumed accurate. The reconstructed trajectories are compared with respect to the ground truth to measure errors in terms of positional and orientation accuracy. Ground truth of a trajectory is the digitized version of the actual trajectory marked by the recording vehicle on the earth's surface, the digitized version is extracted using a GIS tool. In absence of an accurate GPS device we choose to sample input GPS data from the ground truth. For a single trajectory the input data is sampled at various interval distances to evaluate the techniques presented in this paper.

The results in Section 4 are averaged over 296 different ground truth trajectories. Our experiments show that the biarc based technique outperforms the Bézier curve based technique in terms of both positional and orientation accuracy. The Bézier based technique performs better in terms of positional accuracy for interval distances lower than 6 meters. However, as the distance between two accurately tagged GPS data points increases, the errors in reconstruction of the original trajectory with the Bézier based approach grow faster in comparison to the biarc based approach. Another drawback of the Bézier based techniques is that it requires tuning in order to improve control point positioning. The biarc based technique presented here has no such requirement.

References

1. Agarwal, P.K., Biedl, T., Lazard, S., Robbins, S., Suri, S., Whitesides, S.: Curvature-constrained shortest paths in a convex polygon. SIAM Journal on Computing 31(6), 1814–1851 (2002)
2. Allard, J.C., Deslypper, C., Saunier, C.: Method and device for training in the operation of moving vehicles (Jun 14 1988), uS Patent 4,750,888
3. Barsky, B.A., DeRose, T.D.: Geometric continuity of parametric curves. Computer Science Division, University of California (1984)
4. Blanton, K.A., Finlay, W.M., Sinclair, M.J., Tumblin, J.E.: Method and apparatus for reproducing video images to simulate movement within a multi-dimensional space (Jun 21 1988), uS Patent 4,752,836
5. Boissonnat, J.D., Cazals, F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. In: Proceedings of the sixteenth annual symposium on Computational geometry. pp. 223–232. ACM (2000)
6. Bolton, K.: Biarc curves. Computer-Aided Design 7(2), 89–92 (1975)
7. Chen, S.E.: Quicktime vr: An image-based approach to virtual environment navigation. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp. 29–38. ACM (1995)
8. Chen, S.E., Williams, L.: View interpolation for image synthesis. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. pp. 279–288. ACM (1993)
9. Choi, J.W., Curry, R., Elkaim, G.: Piecewise bezier curves path planning with continuous curvature constraint for autonomous driving. Machine Learning and Systems Engineering pp. 31–45 (2010)

10. Deslypper, C.: Method for reading a recorded moving scene, in particular on a videodisk, and application of said method to driving simulators (Jul 3 1990), uS Patent 4,939,587
11. Elbanhawi, M., Simic, M., Jazar, R.N.: Continuous path smoothing for car-like robots using b-spline curves. Journal of Intelligent & Robotic Systems pp. 1–34 (2015)
12. Engström, J., Johansson, E., Östlund, J.: Effects of visual and cognitive load in real and simulated motorway driving. Transportation Research Part F: Traffic Psychology and Behaviour 8(2), 97–120 (2005)
13. Foerst, R.: Driving simulator (May 17 1983), uS Patent 4,383,827
14. Ikeuchi, K., Sakauchi, M., Kawasaki, H., Sato, I.: Constructing virtual cities by using panoramic images. International Journal of Computer Vision 58(3), 237–247 (2004)
15. Jamson, H.: Driving simulation validity: issues of field of view and resolution. In: Proceedings of the driving simulation conference. pp. 57–64 (2000)
16. Kaptein, N.A., Theeuwes, J., Van Der Horst, R.: Driving simulator validity: Some considerations. Transportation Research Record: Journal of the Transportation Research Board 1550(1), 30–36 (1996)
17. Kemeny, A., Panerai, F.: Evaluating perception in driving simulation experiments. Trends in cognitive sciences 7(1), 31–37 (2003)
18. Koc, B., Ma, Y., Lee, Y.: Smoothing stl files by max-fit biarc curves for rapid prototyping. Rapid Prototyping Journal 6(3), 186–205 (2000)
19. Koc, B., Lee, Y.S., Ma, Y.: Max-fit biarc fitting to stl models for rapid prototyping processes. Proceedings of the sixth ACM symposium on Solid modeling and applications (2000)
20. Koegel, M., Baselt, D., Mauve, M., Scheuermann, B.: A comparison of vehicular trajectory encoding techniques. In: Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean. pp. 87–94. IEEE (2011)
21. Koegel, M., Kiess, W., Kerper, M., Mauve, M.: Compact vehicular trajectory encoding. In: Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd. pp. 1–5. IEEE (2011)
22. Levine, O.H., Mourant, R.R.: Effect of visual display parameters on driving performance in a virtual environments driving simulator. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. vol. 40, pp. 1136–1140. SAGE Publications (1996)
23. Lippman, A.: Movie-maps: An application of the optical videodisc to computer graphics. In: ACM SIGGRAPH Computer Graphics. vol. 14, pp. 32–42. ACM (1980)
24. Makino, H.: Clothoidal interpolation—a new tool for high-speed continuous path control. CIRP Annals-Manufacturing Technology 37(1), 25–28 (1988)
25. Moreton, D., Parkinson, D., Wu, W.: The application of a biarc technique in cnc machining. Computer-Aided Engineering Journal 8(2), 54–60 (1991)
26. Mullen, N., Charlton, J., Devlin, A., Bedard, M.: Simulator validity: Behaviors observed on the simulator and on the road (2011)
27. Rossignac, J.R., Requicha, A.A.G.: Piecewise-circular curves for geometric modeling. IBM Journal of Research Development, Vol.31(3) (1987)
28. Snyder, J.P.: Map Projections — Working Manual, U.S. Geological Survey Professional Paper 1395, pp. 37–47. United States Government Printing Office, Washington, D.C. (1987)

29. Takahashi, T., Kawasaki, H., Ikeuchi, K., Sakauchi, M.: Arbitrary view position and direction rendering for large-scale scenes. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. vol. 2, pp. 296–303. IEEE (2000)
30. Thiffault, P., Bergeron, J.: Monotony of road environment and driver fatigue: a simulator study. Accident Analysis & Prevention 35(3), 381–391 (2003)
31. Tomite, K., Yamazawa, K., Yokoya, N.: Arbitrary viewpoint rendering from multiple omnidirectional images for interactive walkthroughs. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. vol. 3, pp. 987–990. IEEE (2002)
32. Voshell, M.: High acceleration and the human body. http://csel.eng.ohio-state.edu/voshell/gforce.pdf (2004)
33. Wither, J., Tsai, Y.T., Azuma, R.: Indirect augmented reality. Computers & Graphics 35(4), 810–822 (2011)
34. Yang, K., Jung, D., Sukkarieh, S.: Continuous curvature path-smoothing algorithm using cubic b zier spiral curves for non-holonomic robots. Advanced Robotics 27(4), 247–258 (2013)
35. Yao, Z., Joneja, A.: Path generation for high speed machining using spiral curves. Computer-Aided Design & Applications 4, 191–198 (2007)
36. Yu, B., Kim, S.H., Bailey, T., Gamboa, R.: Curve-based representation of moving object trajectory. Proceedings of the International Database Engineering and Application Symposium (2004)