# A Parametric Model for Oriented, Navigable Surfaces in Virtual Environments

Hongling Wang*
Center for Statistical Genetics Research
University of Iowa

Joseph K. Kearney†
Department of Computer Science
University of Iowa

## Abstract

This paper introduces a parametric model of oriented, navigable surfaces in virtual environments. An oriented, navigable surface in 3D space is modeled as a ribbon which has a central axis and can be twisted around the central axis. The axis is represented with a cubic spline curve and is approximately arc-length parameterized. The profile of the ribbon along the axis is described by a slant function. With the representation of the ribbon axis and the ribbon profile, a point on the ribbon surface can be expressed as a function of two parameters: the arc length along the axis and the offset from the axis. The parametric model of a ribbon naturally forms a local ribbon coordinate system which provides a frame of reference for behavior code. In order to provide an uniform, continuous frame of reference for an agent to navigate through connected ribbons, we unite the ribbons to form a ribbon called a path.

**CR Categories:** I.6.3 [SIMULATION AND MODELING]: Applications; K.7.m [SIMULATION AND MODELING]: Simulation Support Systems—Environments

**Keywords:** virtual environments, surface modeling, navigation

## 1 Introduction

Ground surfaces are a vital component of virtual environment models. The ground establishes a frame of reference for determining the perceptual "up" direction and provides a platform and visible support structure for buildings and objects. Terrain is most commonly represented by either a height field or a patchwork of polygons. While height fields and polygons accurately represent the geometry of ground surfaces, they provide a poor foundation on which to build the programs that control the movements of autonomous agents that give life to the virtual world. More structured representations are needed to support navigation of synthetic vehicles and pedestrians.

Many of the surfaces on which we move have a natural orientation that defines a preferred direction of movement. For example, roads have a ribbon-like shape that is cross-sectionally divided into lanes. For the most part, vehicles travel within their lanes moving in parallel streams running tangent to local orientation of the road. Roads have complex shapes that bend and curve with the contour of the underlying terrain and twist on banked turns (see Figure 1). In this paper, we present a parametric representation for oriented surfaces based on mathematical ribbons. By making local orientation explicit in the parameterization, we greatly simplify computations for motion guidance. We also explain how to piece together surface fragments to construct continuous composite ribbons.

We represent a ribbon by a 3D space curve. This curve acts as a central axis or spine for this section of road with fixed width. In addition, we define a slant function to represent how the ribbon surface twists around its axis in 3D space. Both the central axis and the slant of the ribbon are functions parameterized by arc length of the central axis. With the central axis, the slant function, and the width of a ribbon specified, the geometry of the ribbon is fixed. The arc-length parameter of the axis and the slant of the ribbon can be seen as a longitudinal measure for a position on the ribbon. The offset of a position on a ribbon from the axis of the ribbon is a lateral measure of the position. The longitudinal measure and lateral measure together can uniquely define a position on a ribbon. With these two measures, a parametric model for the ribbon is uniquely defined to represent a road surface in 3D space. This parametric model forms a frame of reference for behavior code. However, this reference is interrupted when an autonomous agent transfers from one road to an adjacent road. In order to provide a continuous frame of reference for an agent to navigate through connected roads, ribbons representing adjacent roads are united to form a composite ribbon on which the steering behaviors are constructed.

Figure 1: An example road surface

*e-mail: Hongling-wang@uiowa.edu
†e-mail:kearney@cs.uiowa.edu


VRCIA 2006, Hong Kong, 14–17 June 2006.
© 2006 ACM 1-59593-324-7/06/0006 $5.00

## 2 Related work

Road geometry is often modeled using sections of analytical curves in driving simulators [Artz 1995] [Evans 1995] [Carles and Espie 1999] [Papelis and Bahauddin 1995] [Donikian 1997]. A representative work in analytical roadway modeling is by Artz et al. [Artz 1995]. This work presented an analytical road segment database that was implemented at the Ford Driving Simulator. This road segment database is used by vehicle dynamics computations to keep track of which segment each tire is on and to query the database for height, surface normal, and any other information needed.

Artz [Artz 1995] defines a local coordinate system for each road segment. He divides road segments into straight segments, curve segments, and spiral segments. He designs different algorithms for mapping between local coordinates and world coordinates. A straight segment connects two points in the world by a straight line when viewed from above. A simple rectangular coordinate system is constructed for the straight segment with one of the two points at the origin and the local $x$ axis passing through the second point at a positive value. The width of the road is then defined in the local $y$ dimension using a right-handed system with $z$ up. Road parameters such as crown are stored as functions of local $x$ and $y$. A curve segment appears as a circular arc with fixed radius $R$ and arc length $Q$ when viewed from above. For curve segments, a cylindrical coordinate system is defined in a right-handed system, with the origin at the center of the circle that includes the arc, one of the end points at $(R,0)$ and the other at $(R,Q)$ where $Q$ is positive. The road parameters such as crown are again stored as functions of $r$ and $q$ where $R-W/2 \leq r \leq R+W/2$ (W is the horizontal width of the road surface) and $0 \leq q \leq Q$. Transition spirals are used to connect straight segments to curve segments. Curvature of a transition spiral changes linearly with road distance from 0 at the beginning end to a maximum value $\frac{1}{R}$ at the ending point, where $R$ is the radius of the curve segment to which it connects. For transition spirals, a rectangular coordinate system is used to define the spiral and two other measures called local road pseudo-coordinates are used to define the road parameters. The rectangular coordinate system has its origin at the beginning of the spiral where the curvature is zero. The positive $x$ direction is tangent to the spiral at this point. Positive $z$ is up and $y$ is defined accordingly in a right-handed system. The two other measures are $x'$, the distance along the spiral, and $y'$, the shortest distance to the spiral. An important contribution of this work is that they emphasized the importance of having a local coordinate system for behavior control. A disadvantage of this approach is that the local coordinate systems are different for different kinds of road segments.

Donikian and colleagues developed the Virtual Urban Environment Modeling System (VUEMS) [Donikian 1997] for modeling urban road networks for traffic simulation. Their model connects multiple levels of representation of roads by assembling geometric, topological and semantic data in an integrated database. Road geometry is based on a parametric curve that represents an axial line. Roads and intersections are classified according to structure into 11 different categories. In contrast, we've focused on generic representations that can model a wide variety of road and intersection configurations.

Geographers have examined techniques to extract information about roads from aerial imagery. For example, Koutaki and Uchimura use active contours called ribbon snakes to model road boundaries on maps [Koutaki and Uchimura 2004].

In [Willemsen et al. 2003], we presented a real-time database modeling complex networks of intersecting roads and walkways in virtual urban environments. Our representation of a road network is based on a network of interconnected ribbons. Each ribbon represents a section of a road. The strength of our approach is that we provided a uniform representation of ribbons to model different kinds of road segments in 3D space. As a result, we can build uniform local coordinate systems for different road segments, which provide a basis for us to unite connected, different road segments to form a frame of reference across the boundaries of these road sections for behavior code. While this paper explains arc-length parameterization of the central axis of a ribbon and mapping computations between Cartesian and ribbon coordinates, it didn't give the parametric model for ribbons representing the general navigable surfaces in 3D space, which may be curved both horizontally and vertically and twisted. In [Wang et al. 2005], we described how our ribbon based model of a road network introduced in [Willemsen et al. 2003] benefits behavior control of virtual vehicles. We believe the reason why our ribbon based model of a road network greatly simplifies our behavior control is that we have a good geometric model for ribbons which is used to model oriented, navigable surfaces such as roads. In this paper, we describe this geometric model for ribbons and how the ribbons based on this geometric model are united to form a composite ribbon as an uniform, continuous frame of reference for behavior code. The contributions of this paper are a parametric model for ribbons representing the general navigable surfaces in 3D space and the construction of a composite ribbon as a continuous frame of reference for behavior code from adjacent ribbons defined by the parametric model. In contrast to other virtual environment systems [Donikian 1997] [Artz 1995], which have different mathematical models for navigable surfaces in different shapes such as straight roads, circular roads and spiral roads, we use one parametric model to represent navigable surfaces in different shapes, which may be curved both vertically and horizontally and twisted,

## 3 A parametric model for navigable surfaces

We use a 3-dimensional space curve to define the central axis of a road ribbon. Parametric cubic splines are the curves of choice to define space curves for the purpose of motion control. They are widely used in computer animation and virtual environments to define motion paths [Willemsen et al. 2003]. As the parameter variable ranges over the interval of definition, the computed position traces a smooth curve in space. To model the central axis of a road ribbon, we sample a set of points on the central axis, $(x_0,y_0,z_0),(x_1,y_1,z_1),\ldots,(x_n,y_n,z_n)$, where $(x_0,y_0,z_0)$ is the start point of the central axis of the road and $(x_n,y_n,z_n)$ is the ending point of the central axis of the road. We further assume we have a beginning tangent vector $(tan0_x,tan0_y,tan0_z)$ and an ending tangent vector $(tan1_x,tan1_y,tan1_z)$.

Using cubic spline interpolation, we get a parametric representation of a cubic spline curve $Q(t)=(x(t),y(t),z(t))$,

$$\begin{aligned} x(t) &= a_{x,i}(t-t_i)^3+b_{x,i}(t-t_i)^2+c_{x,i}(t-t_i)+d_{x,i} \\ y(t) &= a_{y,i}(t-t_i)^3+b_{y,i}(t-t_i)^2+c_{y,i}(t-t_i)+d_{y,i} \\ z(t) &= a_{z,i}(t-t_i)^3+b_{z,i}(t-t_i)^2+c_{z,i}(t-t_i)+d_{z,i} \end{aligned} \quad (1)$$

where $t$ is from $t_0$ to $t_n$, $n$ is the number of spline segments, and $\{t_0, t_1, t_2, \cdots, t_n\}$ are the break points, $\frac{dx}{dt}|_{t=t_0}=tan0_x$, $\frac{dy}{dt}|_{t=t_0}=tan0_y$, $\frac{dz}{dt}|_{t=t_0}=tan0_z$, $\frac{dx}{dt}|_{t=t_n}=tan1_x$, $\frac{dy}{dt}|_{t=t_n}=tan1_y$, $\frac{dz}{dt}|_{t=t_n}=tan1_z$. The values of $x(t)$, $y(t)$, $z(t)$, $\frac{dx}{dt}$, $\frac{dy}{dt}$, $\frac{dz}{dt}$, $\frac{d^2x}{dt^2}$, $\frac{d^2y}{dt^2}$ and $\frac{d^2z}{dt^2}$ are continuous on $[t_0,t_n]$.

With the spline functions defined for the central axis of a road ribbon, naively, one might compute a position on the central axis using the parameter variable directly. However, more often we locate a new position based on a known position plus the distance from the known position to the new position. For example, a popular way to steer a vehicle on a road in virtual environments is to have the vehicle aim for a position called a pursuit point which is one lookahead distance away from the current position of the vehicle. Here, we need to compute the position of the pursuit point by the current position plus the lookahead distance. Let's assume the vehicle runs on the central axis of the road. The pursuit point can be computed from the parameter value that corresponds to the current position of the vehicle plus a parameter interval that corresponds to the arc length of the road axis equal to the lookahead distance. However, this parameter interval is very expensive to compute in real-time applications.

In addition, we often use arc length between two positions on the central axis of a road to consider their relationship. For example, we determine if two vehicles running on the central axis of a road are too close to each other from the arc length of the axis between their positions. The arc length that corresponds to the parameter interval is very expensive to compute in real-time applications.

The proceeding analysis tells us it is very hard to derive an arc-length interval and the corresponding parameter interval on a road axis from each other because the parameter variable and curve length are not, in general, linearly related [Farouki and Sakkalis 1991] ( For example, see the left graph in Figure 2.) However, this problem will be gone if we have the central axis of the road arc-length parameterized. Therefore, we want the central axis of the road to be parameterized by arc length.

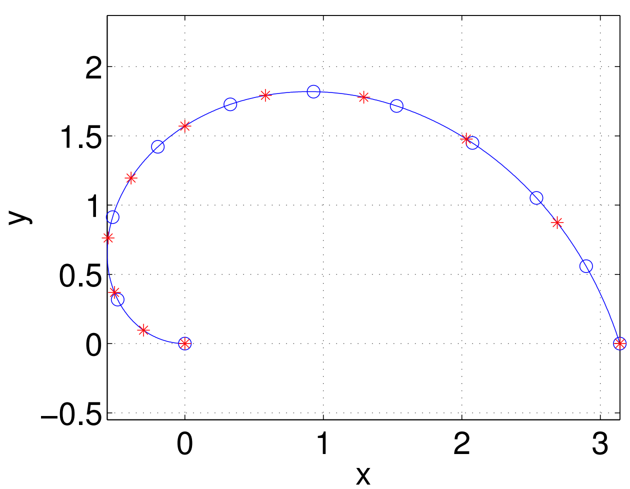


Figure 2: A spiral curve with the points at constant parameter interval ('*') and constant arc-length interval ('o')

For the cubic spline curve defined in formula 1, the first step to compute the arc-length parameterization is to compute arc length $s$ as a function of parameter $t$, $s = A(t)$. The function $A(t)$ is an integral shown in formula 2,

$$A(t) = \int_{t_0}^{t} \left((x'(t))^2 + (y'(t))^2 + (z'(t))^2\right)^{1/2} dt. \tag{2}$$

In general, the integral for a cubic spline curve cannot be computed analytically. Therefore, the arc-length parameterization for cubic spline curves cannot be expressed as a combination of elementary functions and must be evaluated numerically.

Our method computes an approximately arc-length parameterized curve to approximate the central axis of a ribbon defined in formula 1. The approximation curve is computed in three steps [Wang et al. 2002a]. First, the arc lengths of all the cubic segments in the input spline curve, $Q(t)$, are computed and summed to determine the arc length $L$ of $Q(t)$. The second step is to find $m+1$ points equally spaced along $Q(t)$, $(\tilde{x}_0, \tilde{y}_0, \tilde{z}_0)$, $(\tilde{x}_1, \tilde{y}_1, \tilde{z}_1)$, ..., $(\tilde{x}_m, \tilde{y}_m, \tilde{z}_m)$. The third step is to compute a new spline curve using the equally spaced points as knots. We reparameterize the spline curve by interpolating $[(s_0, \tilde{x}_0), (s_1, \tilde{x}_1), ..., (s_m, \tilde{x}_m)]$, $[(s_0, \tilde{y}_0), (s_1, \tilde{y}_1), ..., (s_m, \tilde{y}_m)]$ and $[(s_0, \tilde{z}_0), (s_1, \tilde{z}_1), ..., (s_m, \tilde{z}_m)]$. In this interpolation, we interpolate $x$, $y$ and $z$ to arc length $s$ and get the cubic spline functions in formula 3 [Atkinson 2002] [Wang et al. 2002a],

$$\begin{aligned}
\tilde{x}(s) &= \tilde{a}_{x,i}(s-s_i)^3 + \tilde{b}_{x,i}(s-s_i)^2 + \tilde{c}_{x,i}(s-s_i) + \tilde{d}_{x,i} \\
\tilde{y}(s) &= \tilde{a}_{y,i}(s-s_i)^3 + \tilde{b}_{y,i}(s-s_i)^2 + \tilde{c}_{y,i}(s-s_i) + \tilde{d}_{y,i} \\
\tilde{z}(s) &= \tilde{a}_{z,i}(s-s_i)^3 + \tilde{b}_{z,i}(s-s_i)^2 + \tilde{c}_{z,i}(s-s_i) + \tilde{d}_{z,i},
\end{aligned} \tag{3}$$

where $s \in [s_i, s_{i+1}]$, $i = 0, 1, 2, ..., m-1$, and the values for $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ are of class $C^2$ on $[0, L]$. The tangent vectors of the derived curve at the beginning point and the ending point are set to be equal to the normalized tangent vectors of the original curve at the beginning point and the ending point, respectively. The result is an approximately arc-length parameterized piecewise spline curve divided into $m$ cubic segments.

When we sample the points $(x_0, y_0, z_0)$, $(x_1, y_1, z_1)$, ..., and $(x_n, y_n, z_n)$ used in formula 1 in order to model the central axis of a road, we can also sample the normal of the road surface at each of these points in order to model the road. However, the normal at each of these points must be perpendicular to the central axis. As a result, we have only one degree of freedom when we sample the normal of the road surface at a point on the central axis. In order to quantify this degree of freedom, we draw the plane which is perpendicular to the tangent of the central axis at this point. We assume the road surface is not crowned. The plane intersects the road surface at a cross sectional line. Two unit vectors $v_1$ and $v_2$ are drawn on the plane starting from the point on the central axis. Vector $v_1$ points to the left side along the horizontal direction seen in the direction of the tangent of the central axis. Vector $v_2$ is along the cross sectional line to the left side seen in the same direction. This degree of freedom can be quantified by the angle between vector $v_1$ and vector $v_2$, which defines the slant of the road surface. Instead of directly sampling the normal of the road surface at the sampled point on the central axis of the road, we sample the slant angle, $\theta$, which is shown in figure 3. For each of the sampled points on the central axis of the road, we sample such an angle. Without loss of generality, we assume angle $\theta$ is positive if it is clockwise from vector $v_1$ to vector $v_2$ and negative otherwise. We get a set of slant-angle values $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$ at the sampled points. By cubic spline interpolation, we get a cubic function for the slant of the road surface,

$$\theta(t) = a_{\theta,i}(t-t_i)^3 + b_{\theta,i}(t-t_i)^2 + c_{\theta,i}(t-t_i) + d_{\theta,i}, \tag{4}$$

where $t$, $t_0$, $t_1$, $t_2$, $\cdots$, and $t_n$ have the same meaning as in equation 1, $\frac{d\theta}{dt}|_{t=t_0} = \theta_1 - \theta_0$, and $\frac{d\theta}{dt}|_{t=t_n} = \theta_n - \theta_{n-1}$.

For consistency between parameters of the slant and the central axis of a road, the slant of the road should be reparameterized with arc length. We have solved $\tilde{t}_0, \tilde{t}_1, ..., \tilde{t}_m$ that correspond to the equally spaced points $(\tilde{x}_0, \tilde{y}_0, \tilde{z}_0)$, $(\tilde{x}_1, \tilde{y}_1, \tilde{z}_1)$, ..., $(\tilde{x}_m, \tilde{y}_m, \tilde{z}_m)$ on the central axis of the ribbon when we compute arc-length parameterization. Using the cubic spline function in formula 4, we compute the slant of the road at these equally space points on the central axis of the

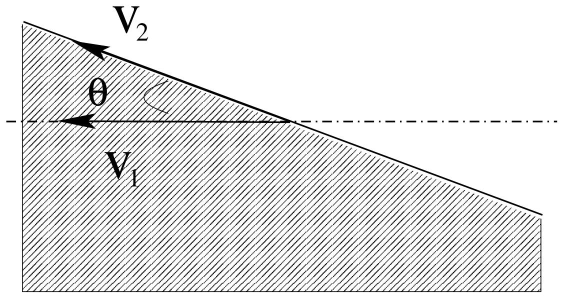


Figure 3: Two vectors $v_1$ and $v_2$ on the plane perpendicular to the central axis of a road seen in the direction of the tangent of the central axis

road ribbon, $\tilde{\theta}_0$,$\tilde{\theta}_1$,$\cdots$, $\tilde{\theta}_m$. We reparameterize the slant function in formula 4 by interpolating $[(s_0,\tilde{\theta}_0),(s_1,\tilde{\theta}_1),\cdots,(s_m,\tilde{\theta}_m)]$ and get the new slant function,

$$\tilde{\theta}(s) = \tilde{a}_{\theta,i}(s-s_i)^3 + \tilde{b}_{\theta,i}(s-s_i)^2 + \tilde{c}_{\theta,i}(s-s_i) + \tilde{d}_{\theta,i}, \quad (5)$$

where $s_0 = 0$, $s_1 = \tilde{l}$, $s_2 = 2\cdot\tilde{l}$, ..., $s_m = m\cdot\tilde{l}$, $\frac{d\tilde{\theta}}{ds}|_{s=s_0} = \frac{\tilde{\theta}_1-\tilde{\theta}_0}{\tilde{l}}$, $\frac{d\tilde{\theta}}{ds}|_{s=s_m} = \frac{\tilde{\theta}_m-\tilde{\theta}_{m-1}}{\tilde{l}}$, $\tilde{l} = \frac{L}{m}$, and $L$ is the arc length of the central axis of the road ribbon.

Unit vector $v_1$ in figure 3 is given by,

$$v_1 = \left(-\frac{\frac{d\tilde{y}}{ds}}{\sqrt{(\frac{d\tilde{x}}{ds})^2+(\frac{d\tilde{y}}{ds})^2}}, \frac{\frac{d\tilde{x}}{ds}}{\sqrt{(\frac{d\tilde{x}}{ds})^2+(\frac{d\tilde{y}}{ds})^2}}, 0\right), \quad (6)$$

where $\tilde{x}$ and $\tilde{y}$ are the spline functions of the central axis of the ribbon defined in formula 3. Vector $v_2$ in figure 3 is given by [Hill 2001],

$$v_2 = \begin{pmatrix} c+qu_x^2 & qu_yu_x - ru_z & qu_zu_x + ru_y \\ qu_xu_y + ru_z & c+qu_y^2 & qu_zu_y - ru_x \\ qu_xu_z - ru_y & qu_yu_z + ru_x & c+qu_z^2 \end{pmatrix} v_1^{'}, \quad (7)$$

where $c = cos(\theta)$, $r = sin(\theta)$, $q = 1-c$, $(u_x,u_y,u_z)$ are the components of the unit tangent vector of the central axis of the road, and $v_1^{'}$ is the transpose of $v_1$ computed in formula 6. If the widths of the left and right sides of the road are $w_l$ and $w_r$ respectively, the road surface can be expressed as the below parametric form,

$$p(s,w) = (\tilde{x}(s),\tilde{y}(s),\tilde{z}(s)) + w * v_2, \quad (8)$$

where $s$ is the parameter of arc length of the central axis of the road, $(\tilde{x}(s)$,$\tilde{y}(s)$,$\tilde{z}(s))$ are the spline functions of the central axis of the road defined in formula 3, $w$ is the parameter of lateral distance between a point on the road and the central axis of the road, and $w \in [-w_r, w_l]$. This parametric presentation enforces another constraint on the geometry of a ribbon that the boundaries of the ribbon on both ends must be perpendicular to the central axis. The boundaries at the start and ending ends are formed by ranging the parameter $w$ between $[-w_r, w_l]$ while the parameter of $s$ is kept at 0 and the arc length of the central axis respectively. In figure 5, we show a ribbon defined by the parametric model in formula 8 that models the road surface shown in figure 4.

## 4 A ribbon coordinate system based on the parametric model of navigable surfaces

The parametric model of navigable surfaces defined in formula 8 gives a natural coordinate system in which points on a ribbon are

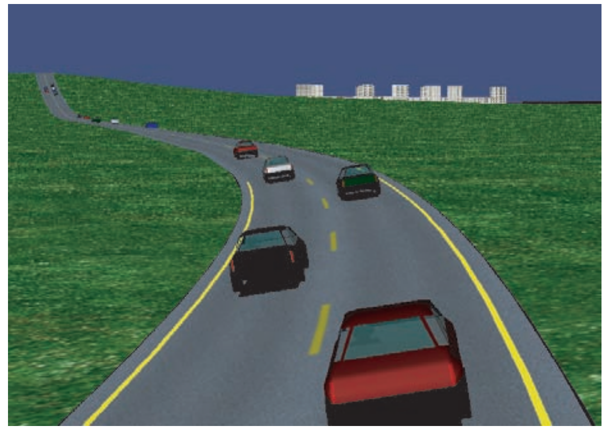

Figure 4: A road surface in a 3D scene

Figure 5: The underlined parametric model of the road surface in figure 4

expressed in coordinates of $(s,w)$ where $s$ and $w$ are the two parameters of the parametric model. However, in virtual environments we are interested in not only the positions on navigable surfaces, but also the positions that are locally around navigable surfaces. In order to express a position $p$ that is locally around a ribbon, we project the position perpendicularly to the ribbon surface. The projection $p_2$ can be expressed by the coordinates of $(s,w)$. Position $p$ can be expressed with two coordinates $(s,w)$, which determine position $p_2$, plus a coordinate of loft $L$ above or below the ribbon surface.

The ribbon establishes a curvilinear coordinate system in which 3-dimensional points are expressed in coordinates of distance along the spine, $D$, offset on the ribbon surface from the spine, $O$, and loft above or below the ribbon surface (positive if it is a displacement above the ribbon and negative if it is a displacement below the ribbon) as shown in figure 6. Figure 7 shows the ribbon coordinates of the position in figure 6 in the ribbon coordinate system.

Ribbons provide a natural basis for behavior code. A ribbon provides a frame of reference for interpreting the location of an object. The distance coordinate $D$ is a longitudinal measure of the location on road surface. The offset coordinate $O$ is a lateral measure of the location on road surface. The loft coordinate $L$ helps to define a position that is above or below road surface. Therefore, behavior code can control tracking based on ribbon coordinates. The ribbon also provides a frame of reference for interpreting the spatial relations

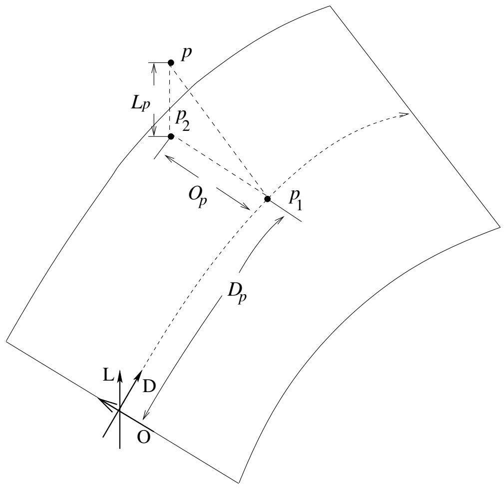


Figure 6: A position around a road in 3D space

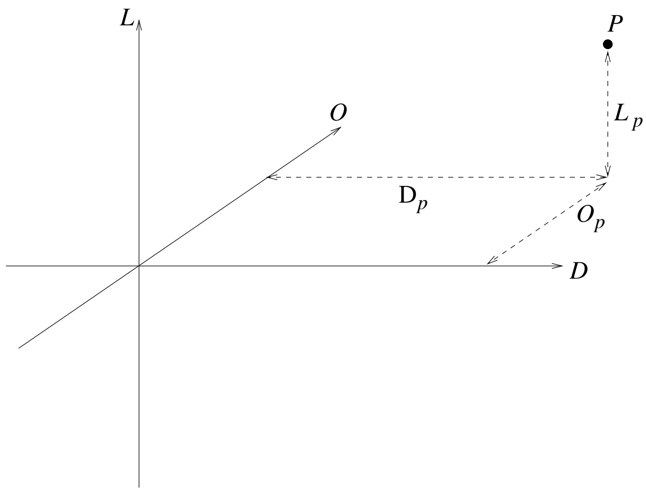


Figure 7: Ribbon coordinates of the position in figure 6

among nearby objects on the same road. By comparing its distance coordinate and those of other nearby objects, an object can determine what is ahead of it, what is behind it, and how far away these objects are from it. By comparing its offset coordinate and those of other nearby objects, an object can determine what is on its left side, what is on its right side, and if any of these objects forms a threat of collision with it. Therefore, behavior code can control interaction between objects based on their ribbon coordinates after they are projected into a same ribbon coordinate system.

Some care must be taken in interpreting the distance coordinate $D$. It roughly corresponds to a mile marker on the road. On multi-lane roads, vehicles may track the ribbon at different offsets from the central axis corresponding to different lanes. On curved roads, this means that $D$ will compress distances on outside lanes and expand distances on inside lanes (track runners are well aware of the different lengths inside and outside lanes.) The mile marker measure means that vehicles at the same value $D$ lie on a cross-section of the road. This is natural for determining important spatial relations such as who is to the left and who is to the right. In our experience, the small distortions in longitudinal distances are inconsequential for behavior controllers.

In real-time simulation, while some computations are most effectively implemented using ribbon coordinates, other computations are most effectively implemented using Cartesian coordinates. For example, behavior modules that track roads and avoid obstacles, are most easily expressed with object locations represented in ribbon coordinates. However, the dynamics code that computes object motions from control parameters set by object behaviors is most simply written in Cartesian coordinates. Because these computations are performed at very high frequency, it is essential to have efficient and robust code to map from ribbon coordinates to Cartesian coordinates and to compute the inverse mapping from Cartesian coordinates to local ribbon coordinates. In [Wang 2005], we defined efficient algorithms to compute the mapping between local ribbon coordinates $(D,O,L)$ and global Cartesian coordinates $(X,Y,Z)$.

## 5 Ribbon networks and composite ribbons

Roads join at intersections to form an interconnected network of ribbons. In general, there is no predominant direction of motion through an intersection and, hence, no natural orientation. Vehicles tracks criss-cross as they drive through an intersection. Reflecting this isotropism, we model an intersection as a non oriented surface with a polygonal boundary. To assure a smooth and continuous surface at (road, intersection) boundaries, we require that the terminating edge of the road be coincident with an edge of the intersection polygon and that the normal of the road surface be aligned with the normal of the intersection along their common boundary. This means that road axes must be perpendicular to the intersection boundary where they make contact with the intersection.

To guide agents across an intersection, we overlay the intersection with corridors that link incoming lanes to outgoing lanes. For example, in figure 8, a corridor $C_1$ connects a lane of road $R_1$ to a lane of road $R_4$. A corridor is a virtual one-lane road – internally "visible" to the synthetic agents, but not rendered on the image. Agents track corridors across the intersection surface. In addition, the corridors provide a frame of reference to interpret the locations and movements of other cars in the vicinity. This local frame of reference is critical for gap acceptance and collision avoidance behaviors. Thus, vehicles on or approaching the corridor to be traversed present an potential hazard for the vehicle intending to track the corridor through the intersection. Corridors are also used to specify right of way rules that regulate access to the intersection for safe traversal.

The change in coordinate systems at the boundary between one ribbon and another creates a bookkeeping challenge that can lead to awkward and complicated code in the programs that control vehicle behavior. For example, consider the vehicle on road $R_1$ in figure 8 as it approaches the intersection. To compute a pursuit point for tracking, it must take into account the section of ribbon it is traversing and the section of the ribbon corridor it plans to track through the intersection. This requires mapping of the two coordinate systems into a single, consistent frame of reference. The axes of the ribbons are offset which further complicates code and the axes are in opposite directions creating the potential for a sign error.

To relieve the programmer from the chore of tedious conversion code, we created a data object called a "path" that unites connected, aligned lanes on roads and corridors on intersections to form a logically continuous ribbon. A path has a single axis from the start to end. On a path, we build an uniform ribbon coordinate system that crosses the boundaries between roads and intersections.

A path [Willemsen 2000] [Willemsen et al. 2003] is defined as a

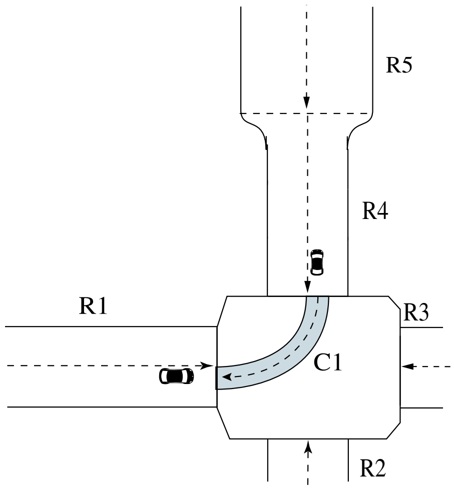


Figure 8: Connected ribbons with different widths, different orientations, and offset axes

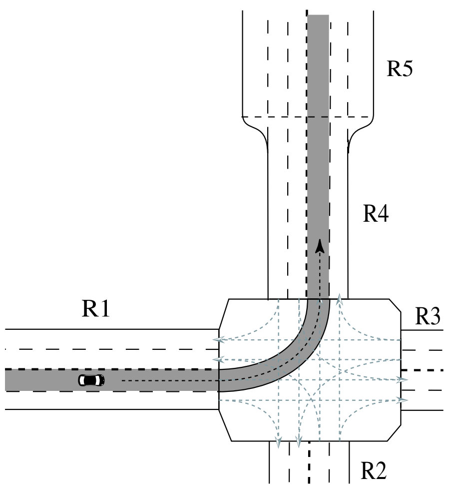


Figure 9: A path crosses an intersection and several roads

connected sequence of lanes on roads and corridors on intersections that an agent has traversed, is traversing, or will traverse in the near future. It is a one-lane ribbon overlaid on the road network (see figure 9). It is oriented and has a beginning and an end. Through the path, an agent can query the roads and intersections on the path, including the road or intersection the agent is currently traversing and the roads and intersections the agent has traversed or will traverse in the near future. The path of an agent is dynamically extended as the agent goes forward. It is an ephemeral data structure representing the immediate plan of action for the agent.

A path is composed of a sequence of strips from adjacent ribbons. The path data object performs conversions to map path queries into the local coordinates of the constituent ribbons. We define a local ribbon coordinate system $(D,O,L)$ for the path of an object, where $D$ is the distance on the central axis of the path between the current position of the object and the start position of the path, $O$ is the offset of the position from the central axis of the path, and $L$ is the loft of the position from the surface of the path. By the definition of ribbon coordinates in section 4, the distance coordinate $D$ is based on the arc length of the central axis of a path, which is the sum of the arc lengths of the central axes of the roads and the intersection corridors on the path.

Geometrically, a path is smooth and continuous. Shifts in the offset of the axes on the underlying ribbons may cause an abrupt change in the parameterization if the ribbons join on a curved section of roadway. For example, the axis of one ribbon may place a lane on the inside of a turn (compressing distances along the center line of the lane). If the axes of the next ribbon runs down the center of the lane, then there will be a discontinuous change in the parameterization on the join. In practice this has caused no problems for the behavior code. The compression is minimal on most roads and typically intersections do not occur on highly curved sections of roads.

The path provides a convenient, egocentric frame of reference that provides a good conceptual foundation for constructing steering behaviors.

# 6 Results and discussion

The model described in section 3 for navigable surfaces has been implemented in the Hank virtual environment software and rigorously tested in psychological studies investigating the behavior of children and adults riding a virtual bike on roadways populated with simulated vehicles [Plumert et al. 2004]. Based on this model and the ribbon coordinate system build on this model, we successfully developed complex steering behaviors [Wang et al. 2005] that control virtual vehicles to run on road networks of virtual environments.

The core problem in modeling oriented, navigable surfaces is to model its central axis. The quality of surface modeling is largely dictated by the quality of axis modeling. We defined a measure called match error to describe how well the arc-length parameterized curve we derived fits the original axis of a ribbon generated from the initial interpolation points. The shape of the derived curve approximately matches the shape of the input curve. We call the misfit of the derived curve from the input curve the match error. We defined another measure called arc-length parameterization error to describe how well the axis of a ribbon is arc-length parameterized. We call the deviation from arc-length parameterization the parameterization error. Both of the two errors can be decreased by increasing the number of segments in the derived curve. Experiments showed that the match error decreases about 10 times, and the arc-length parameterization error decreases more than 5 times for each doubling of the number of spline segments in the arc-length parameterized curve.

Common database computations are efficient including computing global Cartesian (X,Y,Z) coordinates, the surface normal, and the tangent and curvature of a road given local ribbon coordinates. The inverse mapping, from global Cartesian coordinates to local ribbon coordinates, is a nettlesome problem and frequent bottleneck for real-time simulators. This mapping is successfully solved in our navigable surface model. It is efficient and accurate in that it converges to the required accuracy in a small number of iterations. The key component in this mapping is the computation of the closest point on the central axis of the ribbon to a three-dimensional point expressed in Cartesian coordinates. We create a two-stage method [Wang et al. 2002b] which takes the complementary strength of two optimization methods to solve this problem. This method is efficient and extremely robust. Its robustness is demonstrated by rigorous testing over several years without failures.

The ribbon model for navigable surfaces presented in this paper provides a good basis on which we build steering behaviors. The steering behaviors we have built include tracking, following, intersection, and lane changing behaviors. Tracking behavior controls a virtual vehicle to keep its track and run smoothly with its desired speed on the road network. Following behavior controls a virtual vehicle to keep a safe distance behind its leader. Intersection behavior controls a virtual vehicle to respond properly to the control signals and other vehicles on or coming toward an intersection. Lane changing behavior controls a virtual vehicle to move laterally along road surfaces in order to be able to take turns on intersections which are necessary for achieving its long term strategical navigation goal.

The basic steering behaviors mentioned above are constructed in the following way with the support of the parametric model of road surfaces. The initial position of an object is able to be set by its ribbon coordinates on a ribbon. In simulation, we use ribbon coordinates of the object to determine the position on the ribbon of the pursuit point for tracking. Based on the current position, the current orientation and the pursuit point of the object, we compute the new position and orientation of the object on the ribbon. This is the basic tracking behavior. By the distance coordinate of an object on a road, we can determine how far the road end is ahead and introduce appropriate actions (for example, stopping) accordingly. This forms a basis of intersection behavior which controls an agent to navigate intersections safely. By projecting the leader into the ribbon coordinate system of an object, the agent can then determine the distance between the object and its leader from their ribbon coordinates. This distance is a measure the agent uses to compute an appropriate acceleration for following behavior which controls the object to keep a safe distance behind its leader. By projecting the nearby objects into the ribbon coordinate system of a vehicle who is planning lane change, the agent can determine if there is an appropriate gap in the target lane for the vehicle to move in safely. This is important for decision making in lane changing behavior which controls a vehicle to change lanes on roads safely. In [Wang et al. 2005], we described how these steering behaviors are realized based on our parametric model of roadways. We are now working to build more complex steering behaviors such as highway merging behavior and roundabout-intersection behavior.

## 7 Acknowledgement

This material is based on work supported through National Science Foundation grants CDA-9623614, INT-9724746, EIA-0130864, and IS-0002535.

## References

ARTZ, B. 1995. An analytical road segment terrain database for driving simulation. In *Proceedings of 1995 Driving Simulation Conference*, 274–284.

ATKINSON, K. 2002. Modelling a road using spline interpolation. Tech. rep., Reports on Computational Mathematics #145, Department of Mathematics, The University of Iowa.

CARLES, O., AND ESPIE, S. 1999. Database generation system for road applications. In *Proceedings of 1999 Driving Simulation Conference*, 87–103.

DONIKIAN, S. 1997. VUEMS: a virtual urban environment modeling system. *Computer Graphics International* (June), 84–92.

EVANS, D. F. 1995. *High-fidelity Roadway Modeling for Real-time Driving Simulation*. Master's thesis, The University of Iowa.

FAROUKI, R. T., AND SAKKALIS, T. 1991. Real rational curves are not 'unit speed'. *Computer Aided Geometric Design*, 8, 151–157.

HILL, F. 2001. *Computer Graphics Using OpenGL*. Prentice Hall.

KOUTAKI, G., AND UCHIMURA, K. 2004. Automatic road extraction based on cross detection in suburb. *Computer Graphics International* (April), 337–344.

PAPELIS, Y., AND BAHAUDDIN, S. 1995. Logical modeling of roadway environment to support real-time simulation of autonomous traffic. In *Proceedings of First Workshop on Simulation and Interaction in Virtual Environments*, 62–71.

PLUMERT, J. M., KEARNEY, J. K., AND CREMER, J. F. 2004. Children's perception of gap affordances: Bicycling across traffic-filled intersections in an immersive virtual environment. *Child Development 75*, 4, 1243–1253.

WANG, H., KEARNEY, J. K., AND ATKINSON, K. 2002. Arc-length parameterized spline curve for real-time simulation. In *Proceedings of 5th international conference on Curves and Surfaces*, 387–396.

WANG, H., KEARNEY, J. K., AND ATKINSON, K. 2002. Robust and efficient computation of the closest point on a spline curve. In *Proceedings of 5th international conference on Curves and Surfaces*, 397–406.

WANG, H., KEARNEY, J., CREMER, J., AND WILLEMSEN, P. 2005. Steering behaviors for autonomous vehicles in virtual environments. In *proceedings of IEEE Virtual Reality Conference*, 155–162.

WANG, H. 2005. *Efficient Roadway Modeling and Behavior Control for Real-time Simulation*. PhD thesis, The University of Iowa.

WILLEMSEN, P., KEARNEY, J. K., AND WANG, H. 2003. Ribbon networks for modeling navigable paths of autonomous agents in virtual urban environments. In *Proceedings of IEEE Virtual Reality Conference*, 79–86.

WILLEMSEN, P. J. 2000. *Behavior and Scenario Modeling For Real-Time Virtual Environment*. PhD thesis, The University of Iowa.