Overview documentation, trivariate

Vibeke Skytt

June 17, 2010

1 Data Structures

Figure 1 shows the main geometric classes in GoTools and how they are divided between the modules. The focus of this document is the trivariate module and in particular the class SplineVolume.

2 B-spline Volumes

A B-spline volume is represented in SplineVolume in the GoTools module trivariate.

The volume is defined by the formula

$$\mathbf{V}(u,v,w) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{h=1}^{n_3} \mathbf{p}_{i,j,h} B_{i,k_1,\mathbf{u}}(u) B_{j,k_2,\mathbf{v}}(v) B_{h,k_3,\mathbf{w}}(w)$$

with control points $\mathbf{p}_{i,j,h}$ and three variables (or parameters) u, v and w. A basis function of a B-spline volume is a product of three basis functions of B-spline curves (B-splines).

The following is a list of the components of the representation:

dim : The dimension of the underlying Euclidean space.

n_1 : The number of vertices with respect to the first parameter.

n_2 : The number of vertices with respect to the second parameter.

n_3 : The number of vertices with respect to the third parameter.

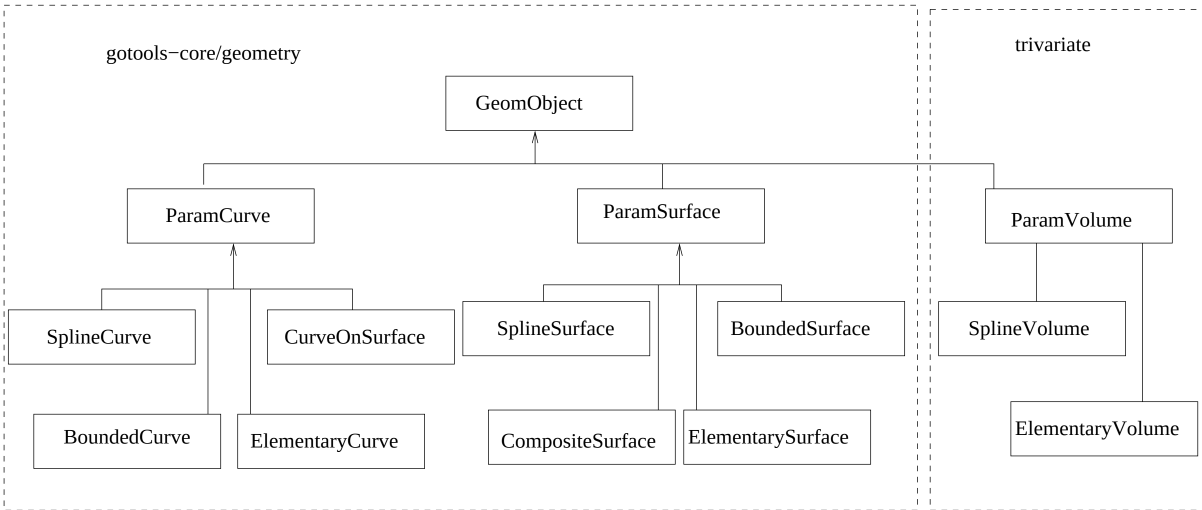

Figure 1: Simplified overview of the geometry class hierarchy

k_1 : The order of the B-splines in the first parameter.

k_2 : The order of the B-splines in the second parameter.

k_3 : The order of the B-splines in the third parameter.

$\mathbf{u}$: The knot vector of the B-splines with respect to the first parameter, $\mathbf{u} = (u_1, u_2, \ldots, u_{n_1+k_1})$.

$\mathbf{v}$: The knot vector of the B-splines with respect to the second parameter, $\mathbf{v} = (v_1, v_2, \ldots, v_{n_2+k_2})$.

$\mathbf{w}$: The knot vector of the B-splines with respect to the third parameter, $\mathbf{w} = (w_1, w_2, \ldots, w_{n_3+k_3})$.

$\mathbf{p}$: The control points of the B-spline volume, $c_{d,i,j,h}$, $d = 1, \ldots, dim$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$, $h = 1, \ldots, n_3$. When $dim = 3$, we have $\mathbf{p} =$ $(x_{1,1,1}, y_{1,1,1}, z_{1,1,1}, x_{2,1,1}, y_{2,1,1}, z_{2,1,1}, \ldots, x_{n_1,1,1}, y_{n_1,1,1}, z_{n_1,1,1}, \ldots, x_{n_1,n_2,1}, y_{n_1,n_2,1}, z_{n_1,n_2,1}, \ldots x_{n_1,}$

The data of the B-spline volume must fulfill the following requirements:

- All knot vectors must be non-decreasing.
- The number of vertices must be greater than or equal to the order with respect to all three parameters: $n_1 \geq k_1$, $n_2 \geq k_2$ and $n_3 \geq k_3$.

The properties of the representation of a B-spline volume are similar to the properties of the representation of a B-spline curve or surface. The control points $\mathbf{p}_{i,j,h}$ form a *control net*. The control net has similar properties to the control polygon of a B-spline curve, described in the module gotools-core. A B-spline volume has three knot vectors, one for each parameter.

2.1 The Basis Functions

A basis function of a B-spline volume is the product of three basis functions corresponding to B-spline curves,

$$B_{i,k_1,\mathbf{u}}(u)B_{j,k_2,\mathbf{v}}(v)B_{h,k_3,\mathbf{w}}(w).$$

Its support is the box $[u_i, u_{i+k_1}] \times [v_j, v_{j+k_2}] \times [w_h, w_{h+k_3}]$.

2.2 NURBS Volumes

A NURBS (Non-Uniform Rational B-Spline) volume is a generalization of a B-spline volume,

$$\mathbf{V}(u,v,w) = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{r=1}^{n_3} h_{i,j,r}\mathbf{p}_{i,j,r}B_{i,k_1,\mathbf{u}}(u)B_{j,k_2,\mathbf{v}}(v)B_{r,k_3,\mathbf{w}}(w)}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{r=1}^{n_3} h_{i,j,r}B_{i,k_1,\mathbf{u}}(u)B_{j,k_2,\mathbf{v}}(v)B_{r,k_3,\mathbf{w}}(w)}.$$

In addition to the data of a B-spline surface, the NURBS surface has a weights $h_{i,j,r}$. NURBS volumes can be used to exactly represent volumes that have common 'analytic' surfaces such as spheres, cylinders, tori, and cones as boundary surfaces. A disadvantage is that NURBS volume depend nonlinearly on their weights, making some calculations less efficient.

The representation of a NURBS volume is the same as for a B-spline volume except that it also includes

h : The weights of the NURBS volume, $h_{i,j,r}$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$, $r = 1, \ldots, n_3$, so $\mathbf{h} = (h_{1,1,1}, h_{2,1,1}, \ldots, h_{n_1,1,1}, h_{1,2,1}, \ldots, h_{n_1,n_2,1}, \ldots h_{n_1,n_2,n_3})$.

The weights are required to be strictly positive: $h_{i,j,r} > 0$.

The NURBS volume is represented by SplineVolume. As for the curve and surface cases, the constructor expects the coefficients to be multiplied with the weights.

2.3 Spline Volume Functionality

The functionality of a spline volume to a large extend corresponds to the functionality of a spline surface. Important functionality is:

- A NURBS volume is able to make a copy of itself
- Compute the bounding box of the volume
- Evaluation and grid evaluation
- Grid evaluation of basis functions
- Compute the derivative volume corresponding to a volume
- Closest point computation
- Fetch a sub volume of a given volume
- Fetch information related to the spline spaces
- Swap and reverse parameter directions in a volume
- Fetch the control polygon of the volume
- Fetch all weights of a NURBS volume
- Insert knots into the spline spaces of the volume and adapt the volume description accordingly
- Increase the polynomial degree of the volume in one parameter direction
- Fetch a constant parameter surface from the volume
- Fetch all boundary surfaces surrounding a volume
- Check for periodicity and degeneracy

3 Construction Methods for SplineVolume

The following methods exist for construction of a spline volume. The corresponding GoTools class names are given in brackets.

- Sweep a NURBS surface along a NURBS curve (SweepVolumeCreator).
- Rotational sweep of a NURBS surface (SweepVolumeCreator).
- Lofting to interpolate a number of NURBS surfaces (LoftVolumeCreator).
- Interpolate 6 boundary surface to create a volume using a Coons patch approach (CoonsPatchVolumeGen). This functionality does apply only to non-rational spline surfaces.
- Represent an elementary volume as a spline volume. The volume types that are handled are:
  - Sphere
  - Cylinder
  - Cone
  - Parallelepiped
  - Torus

A spline volume may have a well behaved outer boundary, but a bad distribution of coefficients in the inner. This is in particular the case if the volume is constructed by a Coons patch approach. The positioning of the internal coefficients may be improved by smoothing. The coefficients at the boundaries are kept fixed and the coefficients in the inner are redistributed by solving a minimization problem. The smoothing is performed in the class SmoothVolume.