Convolution Primitives in HyperFun

Images by Brian Wyvill

HyperFun Library

Primitives

Algebraic primitives:

hfSphere, hfEllipsoid, hfCylinder, hfEllCylinder, hfEllCone, hfTorus, hfSuperel, hfBlock

Skeletal objects:

hfBlobby, hfMetaball,

hfSoft

Convolution objects:

hfConvPoint, hfConvLine, hfConvArc, hfConvTriangle, hfConvCurve, hfConvMesh

Procedural objects:

hfNoiseG

Operations

hfScale,

hfShift,

hfRotate,

hfTwist,

hfStretch,

hfTaper

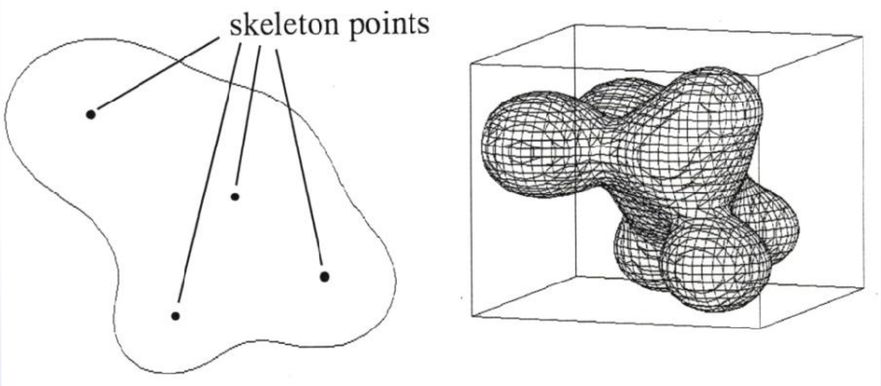
hfBlendUni,

hfBlendInt

Skeletal Surface Definition

$$F(P) - T = 0$$

with $$F(P) = \sum_{i=1}^{N} c_i F_i(r_i)$$



N is the number of skeletal elements,

F_i is the individual scalar field,
(*blending function*) of the i-th element,

r_i is the distance from P to the i-th element,

T is the *threshold* (or *level value*).

Convolution Integral

Defining function of a convolution primitive:

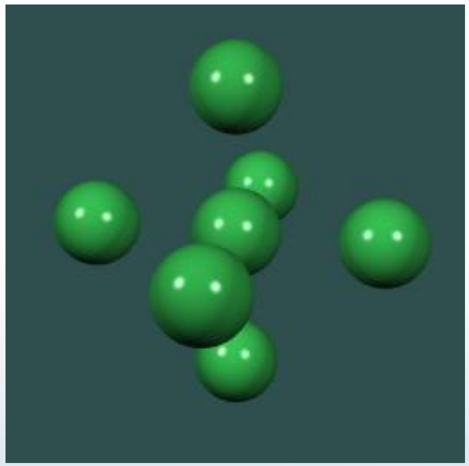
$$f(X) = \int_{R^3} s(P)h(X - P)dP$$

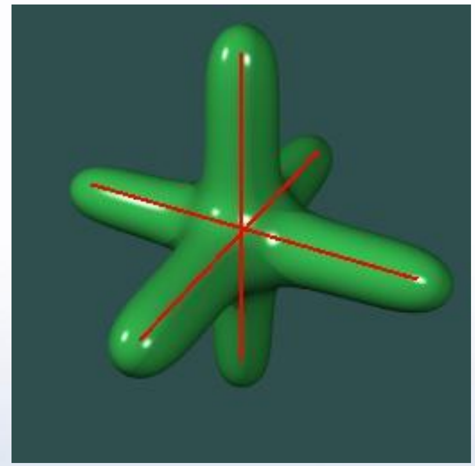
- *s(X)* is a predicate function defining geometry of the skeletal element
- *h(X)* is a convolution kernel

Usually integration requires heavy numerical calculations, but we use analytical solutions for integrals over several skeletal elements.

Skeletal elements

Point

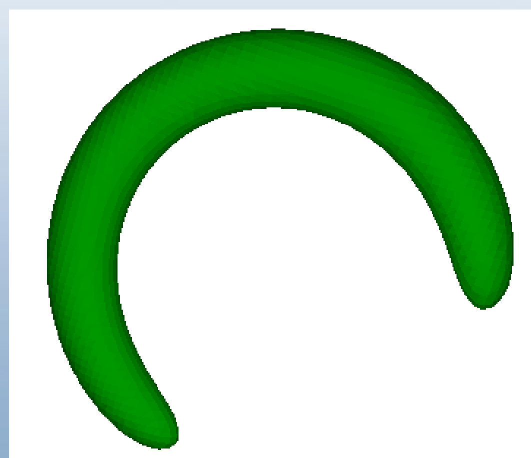


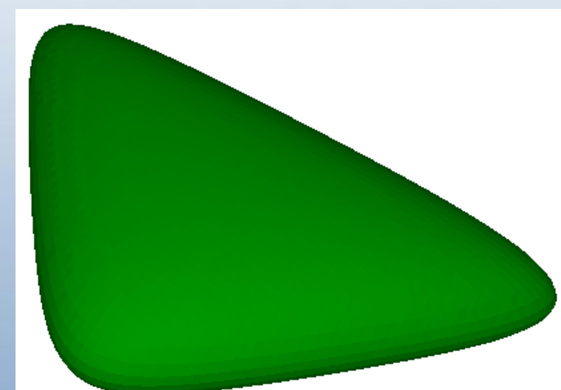


Line

Images by Yuichiro Goto

Arc





Triangle

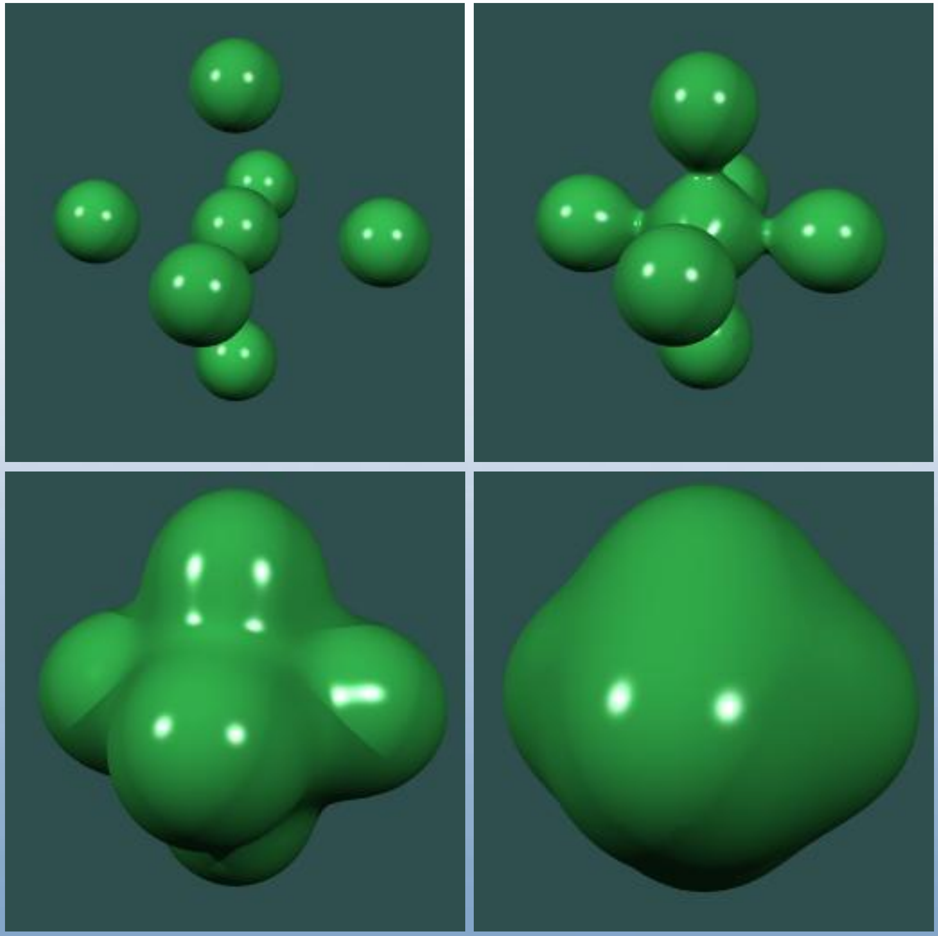
Convolution primitive: Skeletal Points

hfConvPoint(x,vect,S,T)

- **x** – given point coordinates for the function evaluation;
- **vect** – linear array of skeleton points' coordinates organized as (x_1,y_1,z_1, x_2,y_2,z_2, …);
- **S** - array of inverse kernel width parameters for each skeletal point; smaller S_i means bigger i-th component;
- **T** - threshold value for the entire model; smaller **T** means entire expanded surface; bigger **T** means entire contracted surface.

Convolution primitive: Skeletal Points

Smaller S_i means bigger i-th component



S = 1.0

T = 0.1

S = 0.75

T = 0.1

S = 0.5

T = 0.1

S = 0.35

T = 0.1

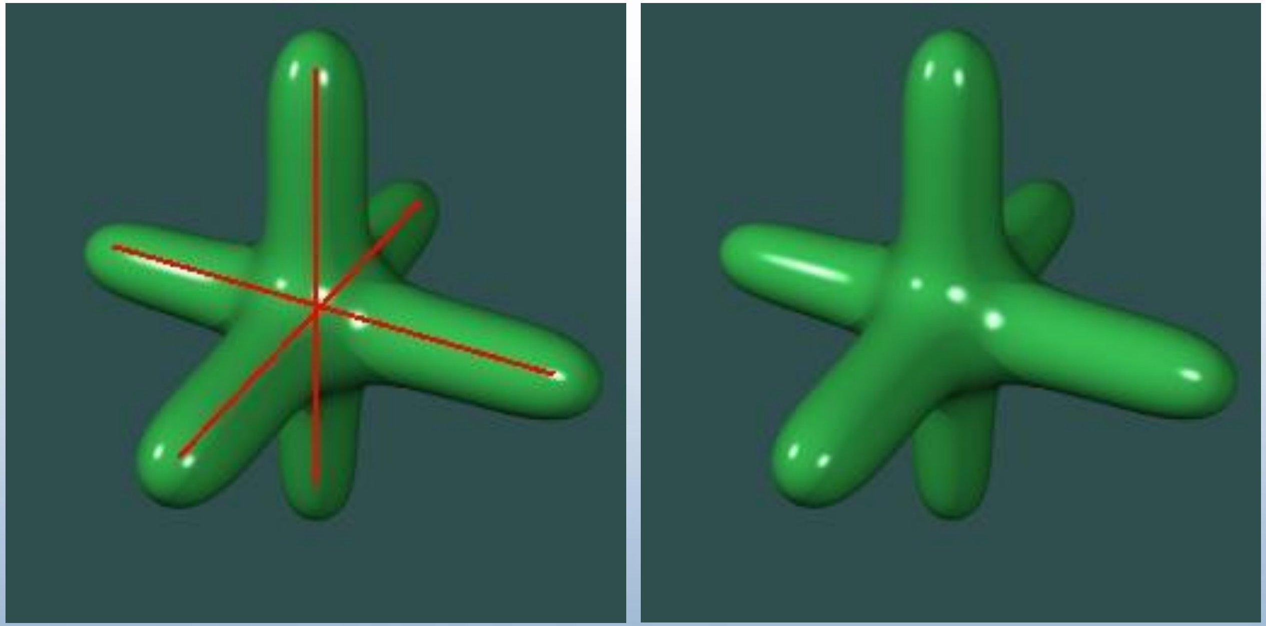
Convolution primitive: Skeletal Lines

hfConvLine(x,begin,end,S,T)

- **x** – given point coordinates for the function evaluation;
- **begin** – linear array of beginning points' coordinates of line segments, organized as $(x_{b1},y_{b1},z_{b1}, x_{b2},y_{b2},z_{b2}, \ldots)$;
- **end** – array of ending points' coordinates of line segments, organized as $(x_{e1},y_{e1},z_{e1}, x_{e2},y_{e2},z_{e2}, \ldots)$;
- **S** - array of inverse kernel width parameters for each skeletal line segment; smaller S_i means bigger i-th component;
- **T** - threshold value for the entire model; smaller **T** means entire expanded surface; bigger **T** means entire contracted surface.

Convolution primitive: Skeletal Lines

Convolution primitive defined by three line segments.

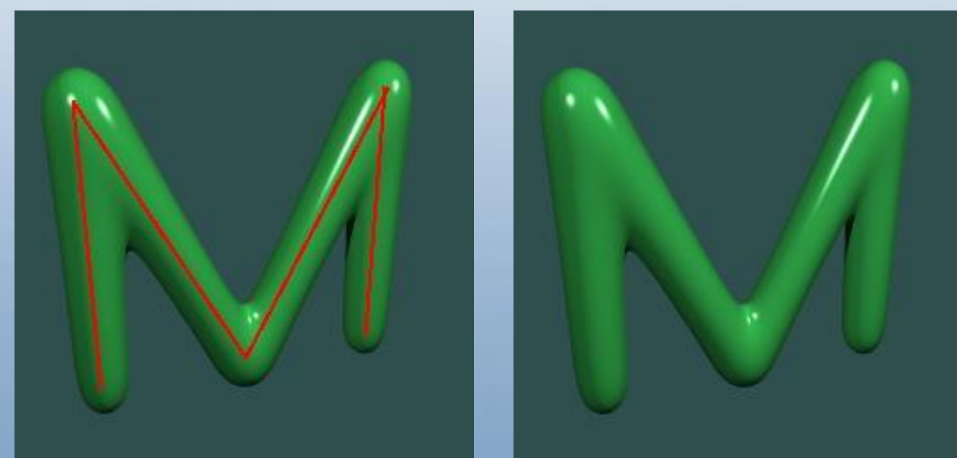


Convolution primitive: Skeletal Curve

hfConvCurve(x,vect,S,T)

- **x** – given point coordinates;
- **vect** – linear array of skeleton curve points' coordinates organized as $(x_1,y_1,z_1, x_2,y_2,z_2, \ldots)$;
- **S** - array of inverse kernel width parameters;
- **T** - threshold.

Convolution surface with a skeleton curve defined by five points.

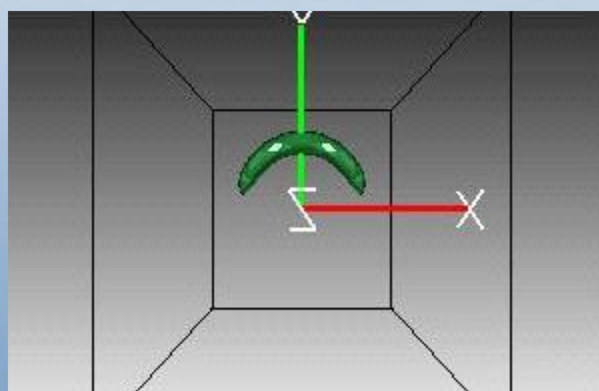


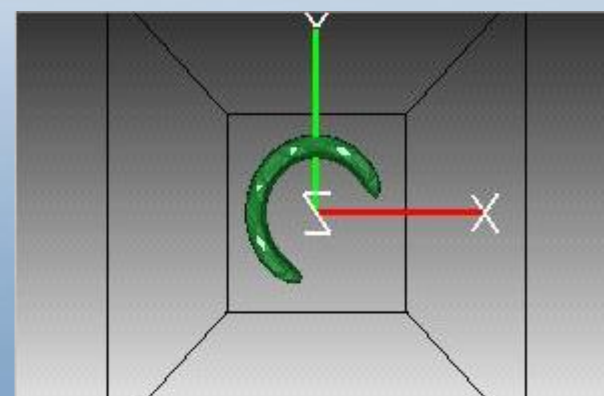
Convolution primitive: Skeletal Arcs

hfConvArc(x,center,radius,theta,axis,angle,S,T)

- **x** – given point coordinates
- **center** – coordinate array for centers of arcs
- **radius** – array of arcs' radii
- **theta** – array of arcs' angles measured from positive x-axis counter-clockwise, 360 degrees are used for the full circle)
- **axis** – array of vectors defining axis of rotation for each arc placed on a local plane parallel to the xy-plane
- **angle** – angles of rotation for arcs around axis of rotation
- **S** - array of inverse kernel width parameters
- **T** - threshold.

theta = 180





theta = 270

Convolution primitive: Skeletal Arcs

Convolution primitive defined by two skeletal arcs
- two full circles with theta = 360
- one rotated about x-axis

```
arcs(x[3], a[1]) {
.............
theta = [ 360.0, 360.0];
axis = [ 0.0, 0.0, 1.0,
         1.0, 0.0, 0.0];
angle = [ 0.0, 90.0];
s = [ 0.5, 0.5 ];
arcs = hfConvArc(x, center, radius, theta, axis,
        angle, s, 0.5);
}
```

Convolution primitive: Skeletal Triangles

hfConvTriangle(x,vect,S,T)

- **x** – given point coordinates;
- **vect** – coordinate array for vertices of triangles, 9 elements for each triangle organized as (x_1,y_1,z_1, x_2,y_2,z_2, x_3,y_3,z_3 …);
- **S** - array of inverse kernel width parameters;
- **T** - threshold.

Convolution surface with four skeleton triangles.

Convolution primitive: Skeletal Mesh

hfConvMesh(x,vertex,index,S,T)

- **x** – given point coordinates;
- **vertex** – coordinate array for vertices of connected triangles organized as (x_1,y_1,z_1, x_2,y_2,z_2, x_3,y_3,z_3 …);
- **index** – list of vertex indices, 3 per triangle organized as (i_1, i_2, i_3, …)
- **S** - array of inverse kernel width parameters;
- **T** - threshold.

```
vertex = [
-2.5, 0.0, 0.0,
0.0, 2.5, 0.0,
2.5, 0.0, 0.0,
0.0, -2.5, 0.0];
index = [ 1, 2, 3, 1, 4, 3 ];
```

Two triangles described in **vertex** and **index** arrays – memory saving structure