

Splines PLUS

Manual Version 2.0

Content

Important Information	1
Copyright	1
Disclaimer	1
License	1
1.Overview	1
2.SplineCombine	2
Input	2
Parameters	2
Collect and Collect Splines	3
Merge	3
Divide	3
Intersect	3
3.SplineDivide	3
Input	4
Parameters	4
Distribution of Intermediate Points	4
4.SplineClone	5
Input	5
Distribution Parameters	5
Clone Parameters	5
Align Parameters	6
5.SplineMesh	6
Input	6
Parameters	6
6.PointToSpline	7
Input	7
Parameters	7

Ralf Sessler
Dimension 3D
E-Mail: d3d@sesseler.de
Internet: www.ralf-sessler.de/d3d



Important Information

Copyright

Splines PLUS is © Copyright 2005-07 by Ralf Sessler. All rights reserved.

Disclaimer

There is no warranty beyond the legal minimal warranty. In no case, the author shall be liable for any damage on hardware or software caused by using Splines PLUS.

License

The license is non-exclusive and non-transferable. It is restricted to the usage on a single computer at one time. The usage is bound to a single registered instance of Cinema 4D (identified by the serial number), but may be transferred to upgrades of this registration.

1. Overview

Splines PLUS is a Plugin for Cinema 4D R8 and up (Windows and Mac). It contains five generators to facilitate and enhance the work with splines. *SplineCombine* combines several splines similar to the Boole object. *SplineDivide* provides alternate subdivision modes for spline interpolation. *SplineClone* duplicates objects along a spline. *SplineMesh* creates a mesh from splines similar to Loft NURBS. *PointToSpline* converts any point object to a spline.

Generated Splines

SplineCombine and SplineDivide are generator object that create a spline from input objects. The resulting spline may be used as any other spline e.g. as input for spline-based NURBS. However, some generators only accept real splines as input, not generated splines or instance objects of splines. In that case, you have to use the *make editable* command to convert the generated spline into a real spline.

Segments

In Cinema 4D, the term segment means an unconnected part of a spline, but sometimes also the part of a spline between two points. For clarification, I will use spline segment to refer to the first, and line segment for the latter.

2. SplineCombine

With *SplineCombine*, you can combine several splines into one spline. There are several modes how to put the splines together based on their intersections.

Input

SplineCombine takes any number of splines as input objects. If an input object has children, the whole hierarchy is collected into one spline according to the selected append mode. Hence, a single input might be sufficient. The Collect Spline mode does not accept deformed splines as input.

Parameters

The *Combination Type* defines how to combine the input splines (see below for details):

- *Collect Spline*: The splines are collected into one spline without applying any interpolation. You can define one interpolation and subdivision mode for this new spline. (All other modes use the interpolation of the input splines.)
- *Collect*: The splines are interpolated and then collected into one spline.
- *Merge*: The splines are merged at their intersections.
- *Divide*: The first spline is divided at the intersections with other splines.
- *Intersect*: The intersections of all splines are used to create a new spline.

The *Append Mode* defines how to connect splines when collecting them:

- *Don't Append*: The splines are not connected.
- *Open Splines*: If two successive splines are both open, the last segment of the first spline and the first segment of the following spline are connected.
- *All Splines*: For all splines, the last segment of a spline and the first segment of the following spline are connected.
- *Open Segments*: If two successive segments are both open, they are connected, even if they belong to different splines.
- *All Segments*: All segments of the resulting spline are connected with the following segment.

The *Divide Mode* defines which parts of the divided spline are included. *Positive* and *Negative* include each second part starting from the first point or the first intersection respectively. *All* includes all parts.

The *Tolerance* is the maximum distance between two lines to intersect. For splines in the same plane, the default value of 0 should work. Otherwise, you can increase this value if lines passing nearby should intersect. Don't use too high values or the result may become strange.

If you check *Apply to Segments*, the segments of all splines are treated as distinct splines. This is mainly useful if an input object is a generator that creates several splines because SplineCombine collects these splines into one spline with several segments.

The other parameters are well known from other splines. Since Cinema 4D either closes all segments of a spline or none, the *Close Spline* option is always applied to the whole generated spline. The other options are for interpolation and only available for the *Collect Spline* mode.

Collect and Collect Splines

Collect simply puts all input splines together in one spline. You can use an append mode to connect the splines or their segments. The difference between *Collect* and *Collect Spline* is that *Collect* uses the interpolated splines (i.e. lines) as input and *Collect Spline* ignores the interpolation and uses the control points of the splines only. For *Collect Spline*, you set one interpolation and subdivision for the generated spline in the object properties of SplineCombine.

Merge

The very idea of *Merge* is to follow a spline until it is intersected by another spline and then to follow the other spline. At each intersection, the spline is changed and the intersecting spline is followed starting from the intersection. Merging is finished when no new intersection is found, either because the current spline ends or reaches a former intersection.

The merging always starts from the first point of the first spline and follows the direction of the splines. Changing the starting point and reversing the point order will change the result.

Divide

Divide breaks the first spline at each intersection with other splines. Depending on the *Divide Mode*, all parts or each second part is included in the generated spline.

Dividing is done for all spline segments of the first spline. It starts at the first point of each segment. Only the segment starting points and point order of the first spline are relevant.

Intersect

Intersect calculates all intersections between all splines. All intersections of one spline segment with one other spline are put together in one segment of the generated spline.

3. SplineDivide

SplineDivide provides an alternate subdivision method when interpolating splines that is more balanced and gives the user more control on how to subdivide the spline.

Input

Only the first input object is used and it must be a spline. Splines with deformers don't work, apply the deformer to SplineDivide instead. To apply SplineDivide to several splines at once, use SplineCombine to create a single spline. Since SplineDivide creates the intermediate points, you can set the intermediate points of the input spline to None.

Parameters

If you check *Natural Subdivision*, the natural distribution of the spline is used. Otherwise, a uniform distribution relative to the length of the spline is used.

If you check *Equal Subdivisions*, each line segment is divided with the same number of intermediate points. Otherwise, the points are distributed relative to the length and the bend of each segment.

The *Number* is the number of intermediate points to be used. For equal subdivisions, this is the number of points for each line segment that is subdivided. Otherwise, it is the number of points for the whole spline. If you check *Per Segment*, the number of points is multiplied by the number of line segments to subdivide, but unlike Equal Subdivisions the points are distributed over all segments.

With *Length / Bend*, you change the balance of how to distribute the intermediate points. A value of 0 will distribute the intermediate points in regard to the length of each line segment, a value of 1 in regard to the bend. 0.5 mixes both with the same weight.

If you check *No Straight Lines*, lines without or with a small bend are not subdivided. The *Angle* is the bend limit, line segments with less bend are considered as straight lines.

Distribution of Intermediate Points

SplineDivide is an improved version of the natural and uniform distributions included in Cinema 4D. If you select Natural Subdivision and Equal Subdivisions, you get the same distribution as the build-in natural interpolation, but in addition you can exclude straight lines from subdivision. Unlike the build-in uniform distribution, the uniform distribution of SplineDivide always keeps the control points of the spline and just subdivides each line segment uniformly according to the length of the segment.

For Equal Subdivisions, each line segment is divided by the same number of intermediate points. Otherwise, the total number of intermediate points is distributed homogeneously according to the length of each segment and how much it is bent. Short and nearly straight line segments get only few intermediate points, while longer and more curved segments get more points. You can change the weight between the influence of length and bend to the distribution for the best result.

4. SplineClone

SplineClone duplicates one or several objects along a spline.

Input

The first input object must be a spline (without deformers). You can use *SplineCombine* to collect or otherwise combine several splines. All other input objects are cloned.

Distribution Parameters

There are five modes for distributing the clones along the spline:

- *Spline Points*: Clones are placed on the control points of the spline.
- *Interpolated Points*: Clones are placed on the points of the interpolated spline.
- *Natural*: A number of clones is distributed naturally along the spline.
- *Uniform*: A number of clones is distributed uniformly along the spline.
- *Spline Distance*: Clones are placed with the same distance along the spline.

For uniform and natural distribution, the *Point Number* is the total number of points distributed along the spline.

For spline distance distribution, the *Distance* is the distance on the spline between two points.

Clone Parameters

The *First Clone* is the point to start with when creating clones.

The *Clone Number* is the total number of clones. The value -1 means the total number of points along the whole spline. If you check *Loop*, the creation of clones starts again at the beginning of the spline if they exceed the spline.

If you clone more than one object, the *Clone Order* determines the order of the clones:

- *Relative*: The first input object is the first clone that is created.
- *Absolute*: The first input object is the first clone at the first point of the spline.
- *Random Relative*: The order is random, but stays the same relative to the clones to create.
- *Random Absolute*: The order is random, but stays the same for each point on the spline.
- *Random Loop*: The order is random, but stays the same for each point on the spline, even if the creation starts again at the beginning of the spline.

For random order, the *Seed* selects a fixed random order.

Align Parameters

If you check *Align to Tangent*, the clones are aligned to the tangent at the point where they are placed on the spline. If you check *Banking*, the banking around the tangent is also calculated.

Note: To calculate the banking, at least three clones must be on a spline segment and the clones must not be on a straight line.

Offset Delta, *Scale Delta*, and *Rotate Delta* are values that are added linearly to the coordinates of the clones, starting from zero and ending at the entered values.

If you check *Include Last Delta*, the delta values themselves are included, otherwise they are target values that are not reached. The latter is useful for loops where the target value should join the first value.

If you check *Delta on Range*, the delta is applied to the clones that are created. Otherwise, it is applied to the spline.

Note: For the spline distance mode, the delta always ends at the last created clone, not at the spline end.

5. SplineMesh

The *SplineMesh* creates a mesh from splines similar to Loft-NURBS, but with additional parameters to control and optimize the created mesh.

Input

Input nodes are any number of splines. The splines may have several segments. For each segment, a mesh is created if there are at least two splines for this segment.

Parameters

The following parameters are used for the U direction along the splines:

- *Subdivision U*: The number of subdivisions on each spline.
- *Uniform*: If checked, a uniform distribution is used instead of a natural.
- *Divide on points*: If checked, subdivision is forced to include the points of each spline.
- *Loop*: The mesh is closed in U direction.

The following parameters are used for the V direction interconnecting the splines:

- *Subdivision V*: The number of subdivisions between the splines. If *variable* is checked, the number of subdivisions depends on the length of the interconnection.
- *Interpolation*: The interpolation used for the interconnections.
- *Subdivide*: The subdivision method for the interconnections. *Natural* uses a natural distribution for each interconnection, *uniform* a uniform distribution. *Constant* uses a constant distance for all interconnections.
- *Divide on splines*: If checked, subdivision is forced to include the splines.
- *Loop*: The mesh is closed in V direction.

With *flip normals*, the orientation of the polygons is reverted.

6. PointToSpline

PointToSpline creates a spline from any object with points. This can be useful e.g. to use a polygon object with *SplineClone*.

Input

The first input must be any object with points, including generated objects. Any other inputs are ignored.

Parameters

The first parameters are those of any spline. The *type* defines the interpolation method, with *closed* the spline is closed. *Intermediate points*, *number*, and *angle* define the interpolated points.

Segments is the number of segments for the spline. The points are distributed uniformly to the segments, i.e. each segments gets the same number of points.

With *reverse*, the point of the spline is reversed.