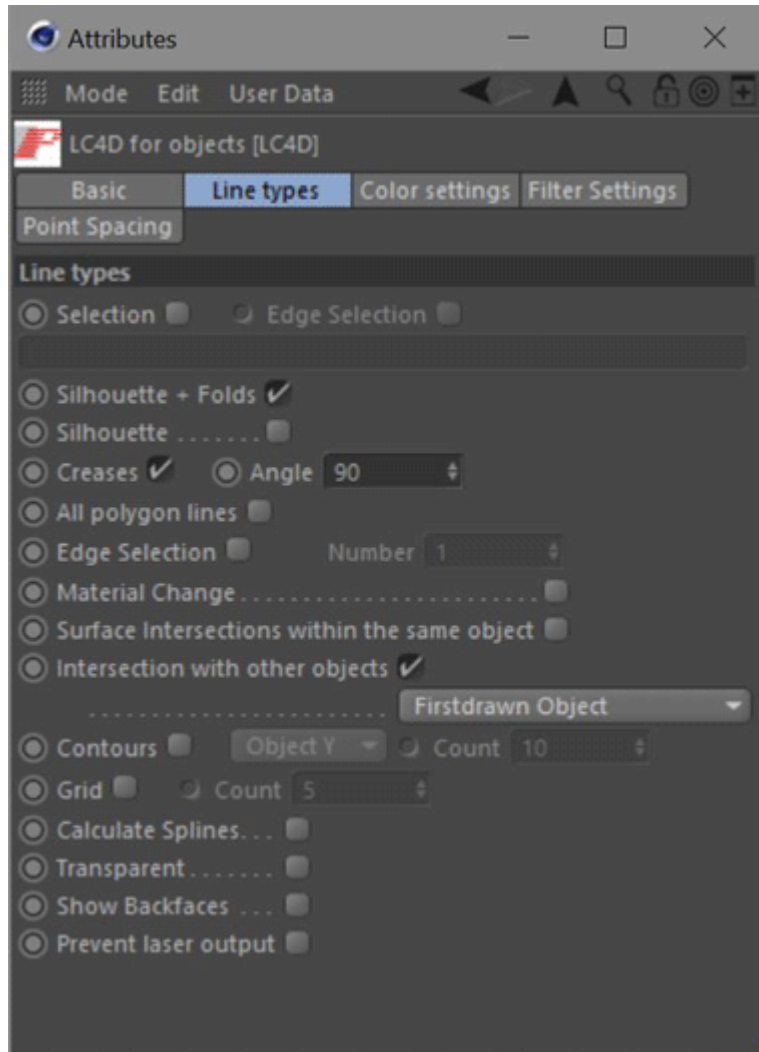# Object Tag: Line Types

This allows you to set the line types on an object-by-object basis.

The Selection checkbox assigns the settings to a specified selection tag (e.g. fold selection). By activating this checkbox, the **Edge selection** checkbox and the selection name field will be activated, and some parameters, not necessary for selection assignment, are deactivated. The selection is specified with its name in the name text field.
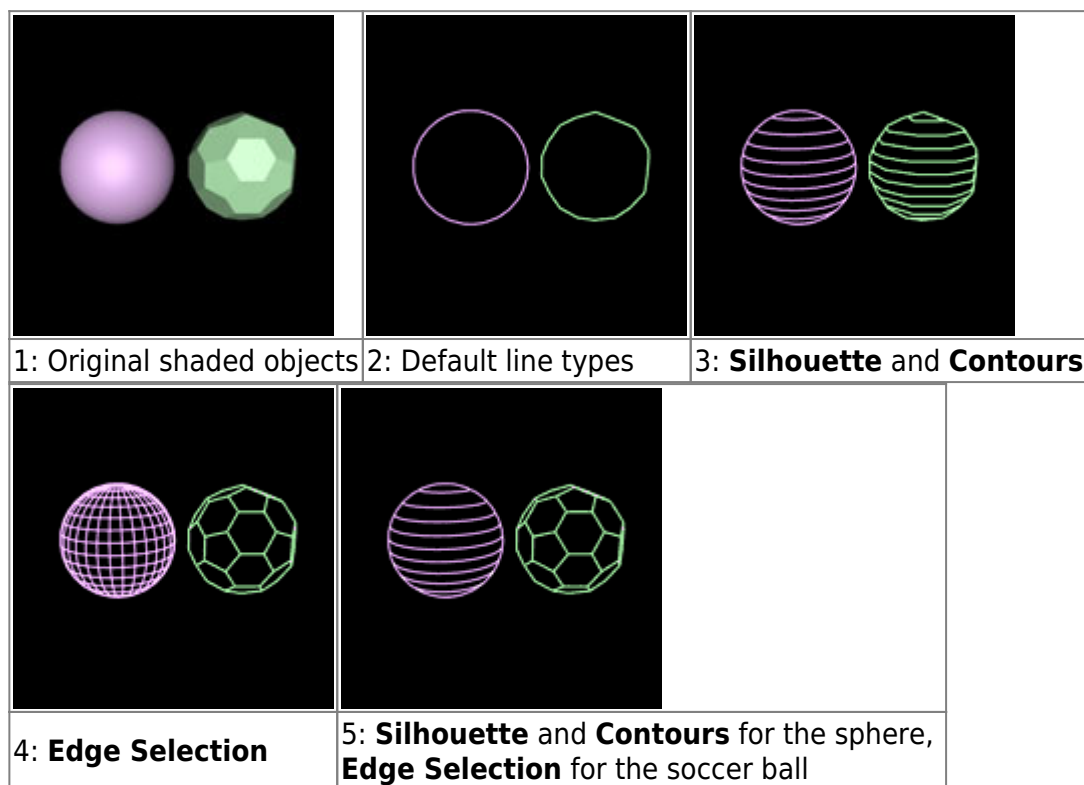
Assigning line types

To assign line types, first allocate the LC4D object tag to the specified object that you want to modify. Then click the check boxes to specify which line type you want (silhouette, creases, contours, etc.) for that object. These settings will be assigned automatically.

Example

For example, let's say that you have two shapes: a sphere and a soccer ball (picture 1 below).

- The default line types only shows the outline of both objects (picture 2).
- Selecting Silhouette and Contours shows something that is good for the sphere, but not for the soccer ball (picture 3).
- Selecting Edge Selection shows something that is good for the soccer ball, but probably draws too many lines for the sphere (picture 4).
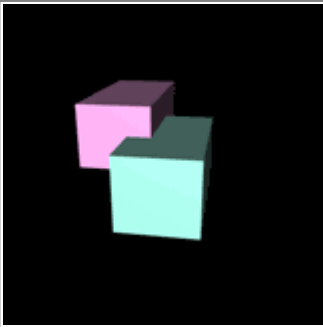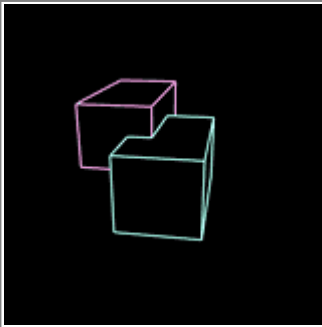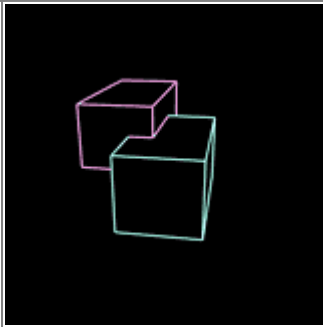
Therefore, for these particular objects, it is best to select the Silhouette and Contours line types for the sphere, and the Edge Selection line type for the soccer ball (picture 5).

1: Original shaded objects	2: Default line types	3: **Silhouette** and **Contours**
4: **Edge Selection**	5: **Silhouette** and **Contours** for the sphere, **Edge Selection** for the soccer ball	

For the Contour line type, you can specify the contour axis. This is discussed in detail on the Contour axis topic page.

The color of intersecting lines

The Lasershow Converter 4D has the capability to create lines where multiple objects intersect. But what object is used to determine the color of the intersecting lines?

Original shaded objects	Cyan cube determines color of intersecting line	Magenta cube determines color of intersecting line

By default, Lasershow Converter 4D will favor the color of the object drawn first. However, you can control this on an object-by-object basis using the **Color of intersecting lines** drop-down list. The options available in this list are:
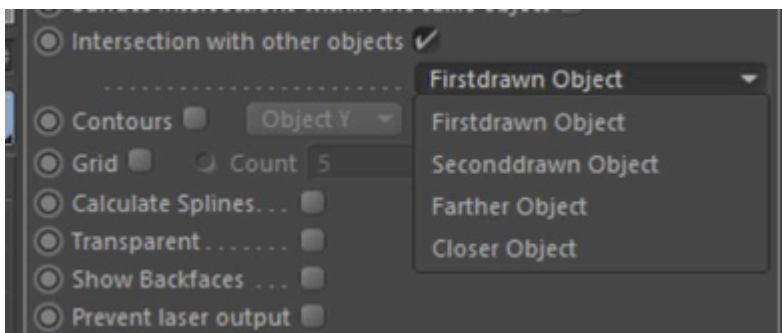
- **First drawn Object**: Uses the color of this object (if it was created before the other object)
- **Second drawn Object**: Uses the color the other object (if it was created after this object)
- **Farther object**: Uses the color of the object that is farther away from the viewer.
- **Closer object**: Uses the color of the object that is closer to the viewer.

What is meant by "created before the other object"?

When you create objects in Cinema 4D, these objects are usually built up from standard types of objects like spheres, cylinders, etc. If the first object that you create is a sphere and the second object you create is a cylinder, the sphere is the "object created before" the cylinder.

The actual object number is shown in the Lasershow Converter 4D Laser Preview panel. As you select different objects, you will see this object number updated.

The main point to understand is that the object drawn first is the object that gets to choose what color the intersecting lines will be.

===== Why not always use the "Color of the closer object" as the color of intersecting lines by default?

Although this might work for some scenes, it might allow the colors to change in ways that are not obvious at first. Complex camera motions can make it so that an object that is closer on one frame, is not
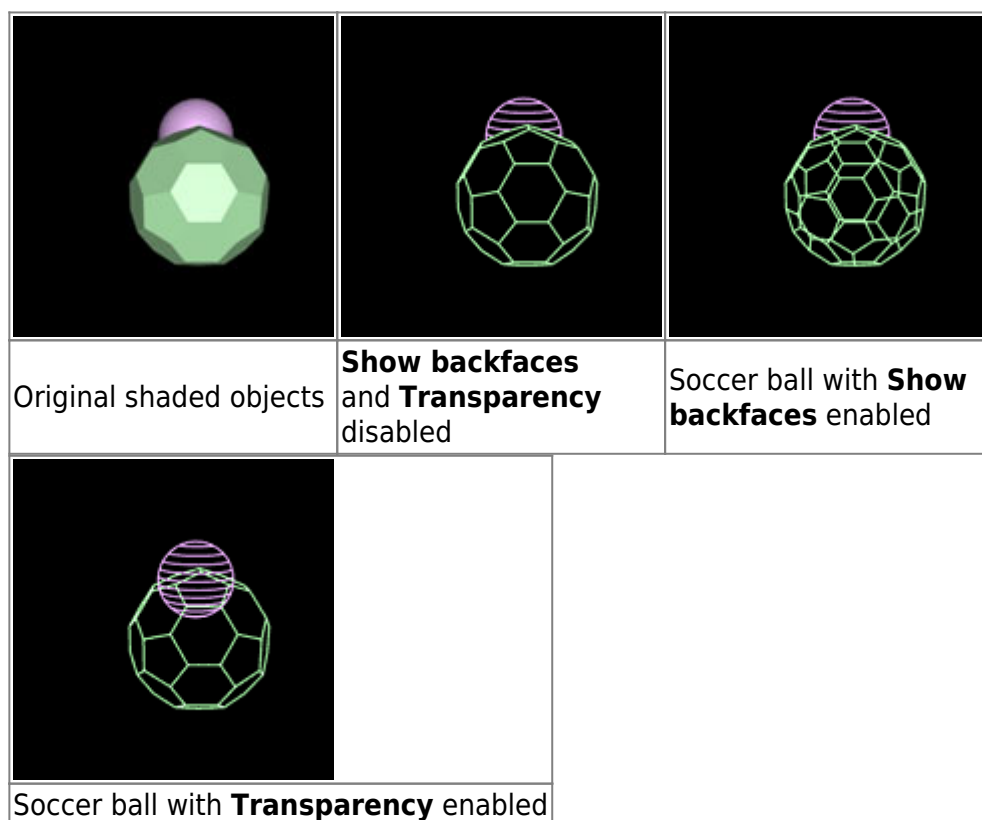
the same object that is closer on another frame. This can make the colors of intersecting lines change from frame to frame, which can be disturbing.

Most of the time, you should select either **Color of this object** or Color of object drawn later. Selecting one of these will ensure that no matter what happens, the color of intersecting lines will always be consistent.

Special effects

In addition to the standard line types, there are three other line types available in this panel. These give you more control over the objects being rendered and allow special effects.

- **Show Backfaces** prevents hidden line removal from happening within the selected object. This makes the object look like a typical 3D laser frame with no hidden line removal within this object. Note that even though there is no hidden line removal within this object, this object will still hide other objects because it is not defined as **Transparent**.

- **Transparent** prevents this object from hiding other objects behind it. This is useful for creating transparent domes over other objects.

Original shaded objects	**Show backfaces** and **Transparency** disabled	Soccer ball with **Show backfaces** enabled
Soccer ball with **Transparency** enabled		

Note that transparency and backfaces are two separate and different options. Making an object transparent does not allow you to see that object's backfaces. Likewise, showing an object's backfaces does not allow you to see through that object to other objects.

- **Prevent Laser Output** calculates the object, but does not actually write any data (points) to the

laser frame. This lets the object mask or intersect other objects, without being seen. In a sense, the object becomes "black". This is useful in several cases:

1. To produce a hole in the laser output that exactly matches video or a slide. For example, to do laser on top of video, render the Cinema 4D scene twice: once normally (AVI output) to produce the video background, and once using Lasershow Converter 4D to produce the laser overlay. For the video's AVI file, make the laser objects invisible. For the laser file, use Prevent Laser Output so the video objects are not visible.
2. To do multi-scanner shows. For example, for a two-scanner-head show, render the Cinema 4D scene twice: once for scanner 1 and once for scanner 2. For scanner 1, use Prevent Laser Output on all objects to be projected by scanner 2. Similarly, for scanner 2, use Prevent Laser Output for all objects to be projected by scanner 1.
3. To hide certain objects (such as walls used to block the camera from seeing other objects) and keep them from producing points.

See Also

- Contour axis
- Post Render Effect: Line Types
- Tips: Using the Selection Tool
- Tips: Using Edge Selection

Go Back