

Tips: Color and path consistency

As was discussed in the [Object Point Spacing](#) topic, it is highly desirable for the beam path to stay consistent from one frame to the next. Lasershow Converter MAX ensures that this beam path consistency is maintained by using several techniques.

Drawing objects in order

The first way that Lasershow Converter MAX ensures path consistency is by controlling the order that the objects are drawn – in other words, the beam path from object to object.

If on the first frame, object A was drawn first and object B was drawn last, but on the second frame object B was drawn first and object A was drawn last, this would not be consistent. It would lead to undesired visual effects including flicker, jitter and the like.

Lasershow Converter MAX resolves this by always drawing the objects in the same order that they were created in 3ds Max. In other words, if you draw a cube first, then you draw a sphere, object number 1 is the cube and object number 2 is the sphere. The actual object number is indicated on the main **Laser Preview** panel and also on the [Object line color](#) panel when an object is selected.

Path-finding within an object

The second way that Lasershow Converter MAX ensures path consistency is by using sophisticated path finding algorithms to ensure that the path within an object does not change from frame to frame. This path finding is very difficult and can never be made 100% perfect due to the fact that the number of lines and the placement of these lines can change from one frame to the next.

Lasershow Converter MAX lets you control the path within an object, by using the [Object point spacing](#) panel **Beam path for this object** drop-down list.

Note that whatever option you select,, this only affects the beam path taken within that specific object. This does not control the overall beam path taken from one object to the next. The beam path from one object to the next is always and exclusively controlled by the order in which these objects are created. The combination of these two techniques ensure that the path is as consistent as possible from one frame to the next.

How color affects path consistency

There is one way in which color settings can affect path consistency . The **Color of intersecting lines** setting controls more than just the color – it controls which object the intersecting lines “belong to”.

For example, if you have a cube that is object number 1 and a sphere that is object number 2, and the two intersect, the intersecting line will be drawn when the cube is drawn if Color of this object is selected. If **Color of intersecting lines** is set to color of the object drawn later, the intersecting line will be drawn when the sphere is drawn.

If the two objects are of different colors, it will be obvious which object the intersecting line “belongs to”, because the intersecting line will be the same color as that object. But where it gets tricky is if two objects are the same color and they intersect. In this case, you will get an intersecting line, but it could be that a better beam path with fewer points could be derived if you had set a different selection for **Color of intersecting lines**.

In many cases this will not matter. Lasershow Converter MAX is capable of producing beautiful imagery, no matter which object the intersecting lines belong to. It is just that if you analyze the resulting rendered frame with LD and follow the beam path from point to point, you may notice that it seems to jump around unnecessarily, or that extra blanking points were included during the drawing of the intersecting lines. Lasershow Converter MAX allows you to control when these intersecting lines are drawn, and the color of these intersecting lines – all based on the setting for **Color of intersecting lines**.

See Also

- [Tips: Beam path optimization](#)

[Go Back](#)

From:
<https://wiki.pangolin.com/> - **Complete Help Docs**

Permanent link:
https://wiki.pangolin.com/doku.php?id=tools:lcmax:tips_color_and_path_consistency

Last update: **2021/05/02 23:41**

