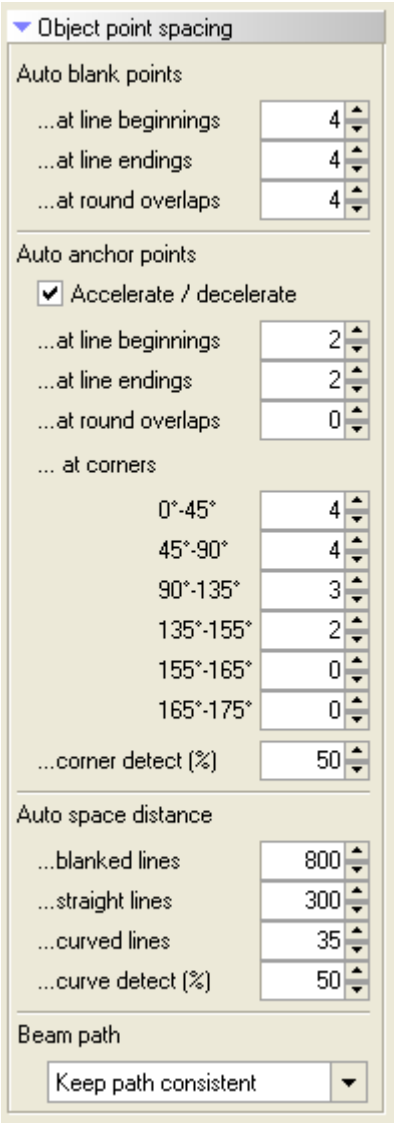
Editing controls: Object Point Spacing

The Object point spacing panel allows you to control the point placement used for the selected objects. Although the default values (shown in the picture below) provide good results for most scenes, it may still be very useful to adjust these settings based on the actual scene contents.



Use default settings

If this checkbox is checked, the selected object will use the default point placement as specified in the Point Placement tab of the Render Settings dialog box. If this checkbox is unchecked, then the settings in this panel will be used for the selected object.

About point spacing

The way points are placed in lines depends on whether the object is rounded or not.

- If the object is square, lines will always start with **AutoBlank Points at line endings**, followed by **AutoAnchor Points**.
- If the object is round, a special point placement technique called “closed contour optimization” is used to hide where the beam starts and ends. In this case, **AutoBlank Points at round overlaps** is used to get the beam going in the direction of the circular object while it is blanked. Once the beam is up to speed, it is turned on. At the end of the circle **AutoBlank Points at round overlaps** is used to keep the beam going, but to blank the beam. The result is a “closed contour” in which you cannot tell where the beam started and ended drawing.

AutoBlank Points

AutoBlank Points at line endings is the number of blanked points that will be automatically placed at the beginning and ending of lines on square objects. These blanked points are all placed on top of each other, at the starting and ending location of the line.

AutoBlank Points at round overlaps is the number of blanked points that will be automatically placed at the beginning and ending of rounded objects. These blanked points are not placed on top of each other, but instead are drawn the same way the object is in order to hide where the drawing started and ended.

If you set this value to 0, no closed contour optimization is used. Instead, the object is always drawn using **AutoBlank Points at line endings** and **AutoAnchor Points**.

AutoAnchor Points

AutoAnchorPoints at line beginning is the number of points that are placed at the beginning of a line.

AutoAnchorPoints at line ending is the number of points that are placed at the ending of a line.

AutoAnchorPoints at round overlaps is the number of extra points that are placed at the point of overlap of a circle or other rounded object. Normally this is set to zero so that you can't see where the beam starts or ends. But if you are using scanners that are not tuned perfectly, you may actually see a visible gap on circles. Increasing this number will hide the gap on poorly tuned scanners but this may actually cause a visible overlap on well-tuned scanners.

AutoAnchorPoints at various angles, is the number of additional points placed at various angle corners. There will always be at least one point at every corner. For example, a setting of “4” means there will be a total of 5 points placed at that corner.

The **Accelerate/Decelerate** check box controls how the points are placed from the starting and ending

of lines, and at corners. If this is checked, a special algorithm is used to accelerate and decelerate the beam from corner to corner, and from beginning to ending. When this is checked, it makes the object easier for scanners to scan, and it also reduces the number of corner and beginning and ending points needed. This can also keep objects looking good as they zoom out since it automatically controls the number of points along straight lines.

Corner detect ranges from 0 to 100 percent and controls the sensitivity to corners. At 0 percent, no corners are detected; this means the **AutoAnchorPoints** at various angles will be completely ignored. At 100 percent, any angular change within an object will be considered in light of the angle and **AutoAnchorPoints** will be placed for these particular angles.

AutoSpace Distance

AutoSpace Distance for blanked lines controls the point density of blanked lines. Lasershow Converter FLASH employs a sophisticated point placement scheme for blanked lines that generally allows a lower point density for blanked lines, but also allows the scanner to easily get from one line to the next.

AutoSpace Distance for straight lines controls the point density for visible lines that are completely straight with no curvature.

AutoSpace Distance for curved lines controls the point density for very tiny circles and curved lines.

AutoSpace Curve detect

Lasershow Converter FLASH senses the curvature of lines, and automatically adjusts the point density. If a line is completely straight, it uses the straight line density. If a line is highly curved, it uses the curved line density. For lines that are in-between, it will use an in-between point density; this maximizes efficiency of point placement.

AutoSpace Curve detect controls the sensitivity to curves. The minimum setting of 0 means no curves are detected – the visible density for straight lines is always used. The maximum setting of 100, means Lasershow Converter FLASH will place a high density on lines even if they are only slightly curved.

Note that with all of this flexibility, you have the power to create any kind of point placement that you desire. If you desire strictly even intensity over the entire image and “linear” point placement, simply turn curve detect off by setting this to 0. However, if you don’t mind some brightness variation in return for lower point counts in imagery, set this to a value of 20 or higher. For more information about variable vs. constant point density, see the Tips: Point spacing for lines topic.

Beam path for this object

This controls the path that the laser beam follows, when drawing the selected object.

It is important for animations to have a beam path that is consistent from frame to frame. If the beam path changes substantially from one frame to the next, this will produce noticeable disturbances that could look like jitter, or like missing pieces of the frame, or like flicker. In Lasershow Converter FLASH, there are different rules that can be applied to keep the laser path optimized:

- **Keep path consistent** Uses generic rules to try to keep the beam path consistent (same direction and line traversal order), for each laser frame. In order to create this consistent path, the actual beam path taken in any given frame may not be very optimal. Depending on the object, this could produce a beam path that seems to jump randomly from one line to the next. (The viewer may not be aware of this, since the beam is blanked between lines.)
- **Circular optimization** Uses special rules that are particularly good for circular or curved objects. Where **Keep path consistent** may mistakenly reverse the path occasionally on these objects, **Circular optimization** uses different rules to maintain beam path consistency.
- **Linear optimization** Rearranges lines somewhat to achieve a shorter path, for each laser frame. This tries to minimize the “jumping around” effect, yet still keep the path direction consistent. This tends to result in shorter overall paths and thus fewer points expended in blanked lines.
- **Aggressive optimization** Rearranges lines aggressively to achieve the shortest path and the least number of points, for each laser frame. This will have even shorter overall paths. This results in the shortest possible beam path to draw the object. To ensure beam path consistency, some of the rules of **Circular optimization** are also applied to **Aggressive optimization**.

No matter which optimization is chosen, every attempt has been made to ensure that the beam path stays consistent from frame to frame.

See Also

- Objects Library
- Render Settings: Point Spacing
- Tips: Point spacing for lines

Go Back

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

Permanent link:
https://wiki.pangolin.com/doku.php?id=tools:lcflash:object_point_spacing

Last update: **2021/04/23 13:31**