



SpeedDisplay

Displaying geometry is slow. Even with the latest graphic cards, scenes with lots of geometry can slow the editor display to a crawl. That's when 'display sacrifices' start to kick in. i.e. we start cutting back on the display detail to speed things up. The first thing to do is to turn on Backface Culling. If this does not improve the display speed sufficiently, we can start giving up on shading and go to wireframe or even isoparms. Sometimes, when even lowering the Level Of Detail is not enough, we may have to go to Box view.

The problem with all this is that it is not interactive. We must add Display tags to each object that is to be affected and adjust the tag each time we want to change the options.

What would be great would be able to vary the display quality in the editor depending on a dynamic factor, such as the distance from the camera.

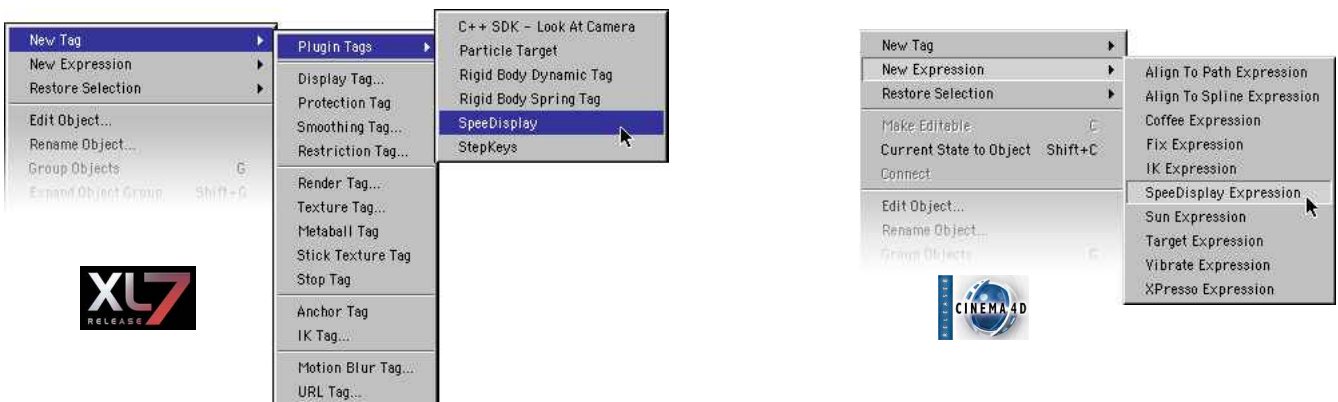
SpeedDisplay provides a solution for that. Think of it as a Display tag on steroids.

SpeedDisplay provides all the options of a standard Display tag but adjusts itself dynamically, based on the distance between objects.

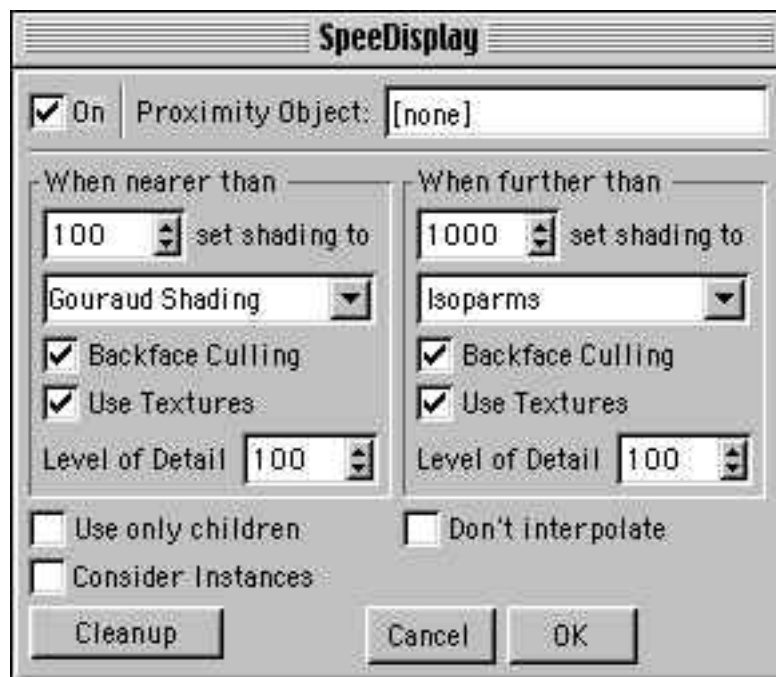
It works with all types of object but is especially efficient with primitive objects (Sphere, Cylinder, Pyramid, etc.) and NURBS objects since these have an adjustable Level Of Detail.

To install *SpeedDisplay*, simply copy the *SpeedDisplay* folder and its contents into your CINEMA 4D Plugins folder.

Once installed and with CINEMA 4D running, select the object for which you want to control the display mode and select **SpeedDisplay** from the **New Tag>Plugin Tags** menu (CINEMA 4D Release 7.x) or from the **New Expression** menu (CINEMA 4D Release 8.x).



The following dialog will appear:



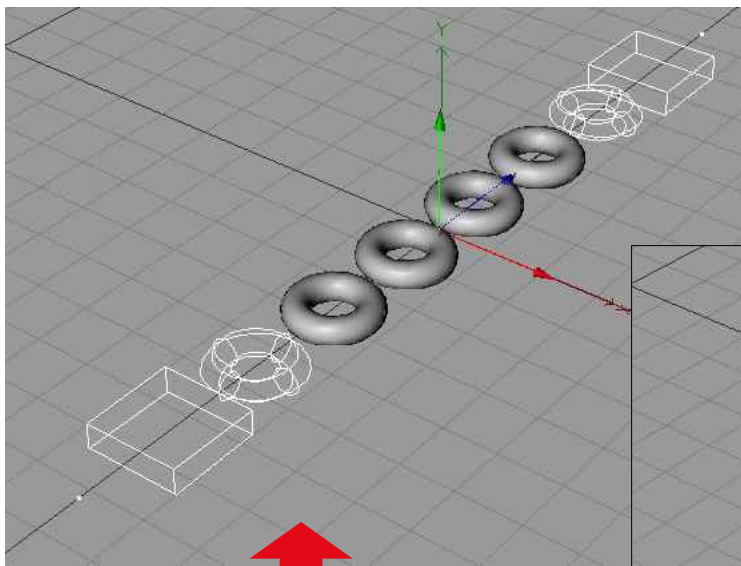
Enabling or disabling **On** turns *SpeedDisplay* on or off respectively; it is on, by default.

In the **Proximity Object** field enter the name of the object that will control the display mode; this will be based on its distance from the object to which we are attaching the tag. This proximity object may be any object, including a camera. The distance is measured between both axes of both objects. If *SpeedDisplay* doesn't find any object in the scene with the name we have entered, it simply does nothing.

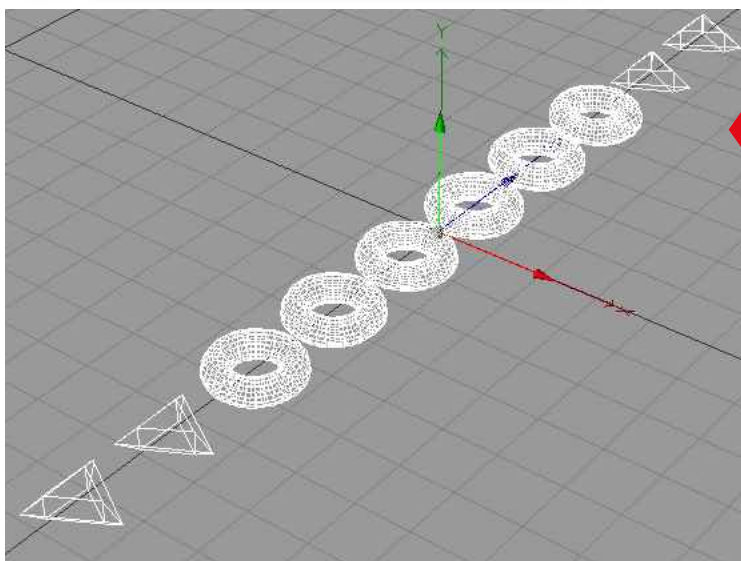
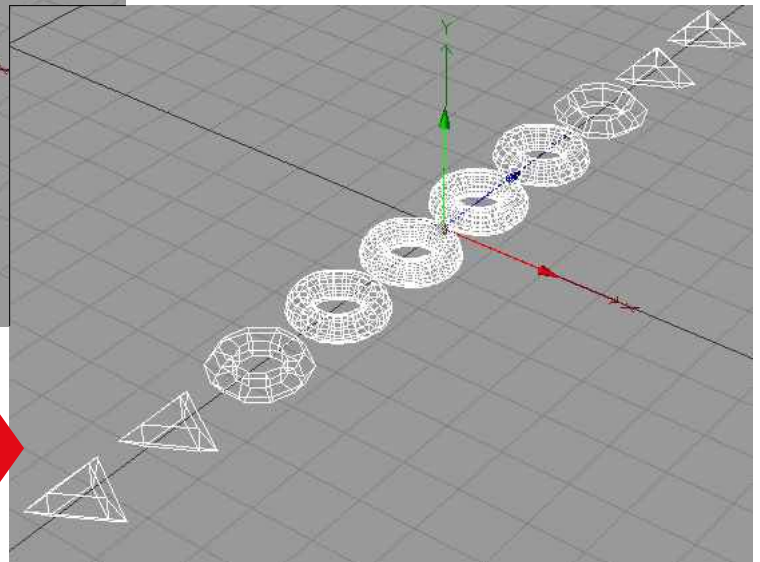
The **When Nearer than** and **When Further than** groups have the same options that a Display tag has and they mean the very same things. The numbers you input here are in CINEMA 4D units and they define the minimum and maximum limits of the distance between your object and the object defined in **Proximity Object**. If the distance between the two objects is between these limits, all options are interpolated.

For instance, if the **When Nearer than** shading mode is set to Gouraud Shading and the **When Further than** shading mode is set to isoparms, as the distance between the objects shortens, the display mode changes first to Quick Shading and, after that, to Wireframe. The same happens with all other options. Both **Backface Culling** and **Use Textures** change when the distance between the objects is close to the average of both distances.

If the **Don't Interpolate** option is on, *SpeedDisplay* doesn't interpolate any values. It just applies the values of **When Nearer than** and **When Further than** groups directly, depending on the distance being smaller or larger than the defined limits.



Sometimes, interpolating the shading modes can be confusing or distracting. A good compromise is to interpolate only the Level of Detail.



Turning off the interpolation shows only two possible extreme states.

Now, it would be a pain to set a *SpeedDisplay* tag for every object in the scene. Of course, we can place all the objects inside a Null (or any other object) and assign a *SpeedDisplay* tag to the parent. But the problem then is that the controlling distance is calculated between the axis of that parent and the axis of the **Proximity Object**. This means that all children will have the same settings as their parent.

What would be really good would be if each child axis was evaluated individually and its display mode adjusted accordingly. This way, the parent would only serve as a *placeholder* for both the objects and the *SpeedDisplay* tag. This is exactly what the **Only use children** option does. With this option enabled, *SpeedDisplay* doesn't evaluate the parent axis. Instead, it runs through all the children and adjusts each one according to its distance from the **Proximity Object**.

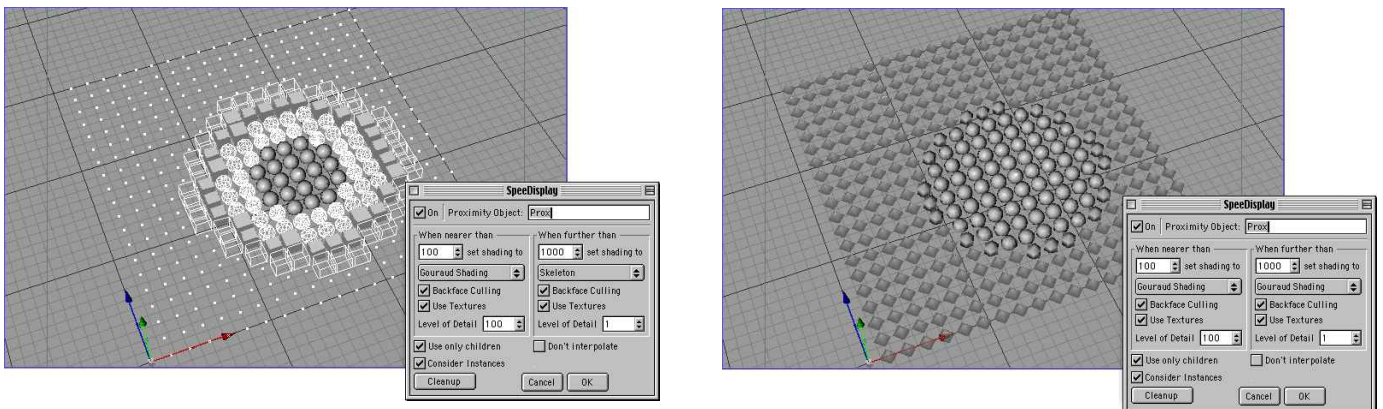
SpeedDisplay clears any Display tag assigned to the parent and assigns one Display tag to each child. These Display tags are completely controlled by *SpeedDisplay* and changing their values manually is of no use.

If we want to pass *SpeedDisplay* control only to the parent, **Only use children** should be disabled and **Cleanup** should be enabled (in that order). This procedure, conveniently, deletes all the tags of the children and reassigns a Display tag to the parent.

Another situation needs some explanation. If inside a hierarchy there are Instances, all the Instances are displayed using the same display mode set in the Display tag of the corresponding instanced object. This means that, if the object defined in **Proximity Object** is close to the instanced object, this object will be set to the **When Nearer than display** mode. So, all Instances will receive the display mode of the instanced object, no matter how far or near from the object defined in **Proximity Object**.

Actually, this is the logical behavior for Instances. But Instances are usually used to recreate a large amount of geometry and they can slow down the editor display substantially. To solve this limitation there is a special mode called **Consider Instances**. Enabling this mode makes *SpeedDisplay* apply a Display tag only to Instances and prevents it from applying Display tags to all other types of object. It's a small compromise since there are usually more Instances than instanced objects. Optimally, the instanced object should be placed outside of the hierarchy of Instances.

After clicking **OK**, *SpeedDisplay* creates one or more Display tags, depending on the options set. This Display tag, or tags, is completely controlled by *SpeedDisplay* and changing its values will have no effect. If the Display tag is accidentally deleted, just double-click *SpeedDisplay* and hit **OK**. A new Display tag will be created. This method (double clicking *SpeedDisplay* and hitting **OK**) is also useful when drastic changes are made to the scene and *SpeedDisplay* needs to be refreshed to do its stuff.



IMPORTANT!

In order for *SpeedDisplay* to work, **Use Shading Property** and **Backface Culling** must be turned on in the Display menus of each viewport.

Known issues

Due to an internal limitation with Cinema4D, applying *SpeedDisplay* to an object inside an Array will *not* update the Array object in realtime.

Solution: Apply *SpeedDisplay* to the Array object itself.