The Belt Drive

for CARRARA 4,5 & 6

BETA Version 0.43

The Belt Drive plugin generates and animates the geometry for a variety of transmission belts. The belt is wrapped around a group of pulley wheels with their axles arranged in the X or Y axis. The plugin comes in 3 parts; a Belt Primitive which maintains the geometry, a Pulley Wheel modifier which is used to identify and describe each wheel in the group, and a Belt Drive modifier which is applied to the group of pulley wheels to keep the belt wrapped around them and to properly animate the system.

To install the plugin extract the contents of the compressed zip package into the Carrara\Extensions sub-folder. The plugin must also be installed into the ..\Extensions folder of all render nodes. To remove the Belt Drive Carrara 4 users must delete the Belt.mcx and Belt.dta files from the ..\Extensions\ folder. Carrara 5 and 6 users must delete the Belt.mcx,Belt.dat and Belt.txt files.

Overview

All of the pulley wheels must be grouped together at the same level in the hierarchy. The pulley wheel axles must turn precisely in either the X or Y axis. The belt is wrapped from one wheel to the next according to their order in the group and then back around from the last wheel to the first wheel. Every wheel in the system must add the special Pulley Wheel modifier to identify it. The pulley wheel group can contain other objects and only those marked with the modifier will be used. The belt geometry is wrapped around each wheel with its cross section centred at the pitch diameter in line with the hot point of each pulley wheel. The pitch diameter is typically equal to the pulley wheel diameter plus the diameter or thickness of the belt.

A convention must be followed when arranging and ordering the grouped pulley wheels. Depending on how each pulley wheel is arranged and set to wrap the belt will pass around the outside of the next wheel or underneath it. If the axles are in the X axis then one wheel is infront of the other if it has a greater Y co-ordinate. A line is traced in the plane of the wheels from the centre of one pulley wheel to the centre of the next wheel in the group. If the next wheel is infront of the previous one and the wrapping mode for that pulley is set to *around* then the belt will wrap over the top and around the next wheel. If the next wheel is behind the previous one and the wrapping mode is set to *around* the belt will wrap around under the wheel. If the next wheel is infront of the previous wheel and the wrapping mode is set to *across* then the belt will wrap under the next wheel. Incorrect ordering or labelling and impossible arrangements of pulley wheels will cause unexpected results or no mesh will appear.

The Belt Primitive is inserted into the scene and once the group of pulley wheels is selected it will generate the geometry and wrap itself around them. The Belt Drive modifier must be added to the same pulley wheel group in order to animate and keep the belt wrapped tightly around the wheels. After the belt drive system has been set up the group of pulley wheels can be freely moved, rotated and scaled in the scene. If the pulleys are moved around in the scene or if their parameters are changed the Belt primitive will not update in the preview until the stop button is pressed or the animation is played.

When the belt drive system is animated the first wheel is expected to be the drive wheel. Animation speed is set by the number of times per second the drive wheel will turn. The belt will travel around the system at a rate of the *Travel* speed x Pi x the *Pitch Diameter* of the drive wheel. Wheels of different sizes will rotate at different speeds. Positive travel of the belt around the system follows the right hand rule consistent with usual rotations in Carrara.

Using the Belt Drive

Model or prepare the pulley wheels in the system before adding a Belt primitive into the scene. Any rotation set for each wheel will be controlled by the Belt Drive once the system is set up and the belt and wheels start turning. If the pulley geometry must be orientated to line it up in the scene with the other wheels then select that wheel and choose Edit | Group to preserve the current orientation.

For this example a simple pulley wheel has been made in the Spline Modeler by using some compounded shapes for the wheel and hub. The spokes are all children of the wheel and made with duplicates of a simple cylinder. The Spline object's extrusion path puts the axle parallel with the Y axis.

With the wheel selected switch to the <Modifiers> tab in the properties tray then add a Belt Pulley modifier (found under the Sparrowhawke3D submenu). The outer diameter of this particular pulley wheel is 7 inches. If the belt is going to be ½ inch thick then the *Pitch Diameter* must be entered as 7.5 inches. Every pulley wheel in the system must have a Belt Pulley modifier added with the correct pitch diameter. The other parameters and the tensioners tab are for advanced configurations.

In this example the first wheel has been duplicated then the second wheel is moved in a positive direction in the X axis. When the wheel axles are in the Y axis they should all be aligned by their hot point in the XZ plane and have the same Y co-ordinate. Wheels can be arranged slightly askew though.

Drag and drop a Belt primitive into the scene list or using the top menu choose Edit | Insert | Belt. There will be no geometry seen until the belt is given a group of pulleys to wrap around. The Belt is put into a list with any other Sparrowhawke3D primitive plugins.

All of the pulley wheels must be in the same group and at the same level in that group. Select the first wheel then all of the other wheels in the system by holding the [Shift] key while clicking on them in the scene list. Using the top menu choose Edit | Group then rename the group to something appropriate like "Pulleys". Switch to the <Modifiers> tab in the properties tray and then add a Belt Drive modifier to the pulley group. The Belt Drive modifier is required to get the system to properly wrap and animate. Hit the button and use the selection dialog to find the Belt primitive in the scene list, Carrara 4 users must type in the exact unique name of the belt primitive. Check the *Wheel Axles In* setting, in this example the pulley wheels axles are in are the Y axis.

Select the Belt primitive in the scene list then enter the model room. The Belt primitive needs to know what the group of pulley wheels is then it will wrap the belt around all of the objects inside that group which have a Belt Pulley modifier. Use the button to bring up the dialog and select the pulley wheel group, Carrara 4 users must type in the exact unique name of the pulley group. Check the *Wheel Axles In* setting, in this example it must change to use the Y axis.

The belt will now appear in the preview window wrapped around the wheels. Adjust the belt width and thickness, in this example the Thickness is ½ inch. 3 different belt types are available, flat, v-belt and round. The *Thickness* parameter changes to become the *Diameter* for a round belt. The level of detail adjusts the number of segments used to make up the belt length and the number of points on the cross section for a round belt can also be adjusted.

Return to the assemble room and hit the play button on the sequencer. The wheels will spin and the belt will pass around them according to the right hand rule when a positive *Travel* value is set for a standard arrangement. Point your thumb in the positive direction of the wheel axis then curl your fingers toward your wrist to tell which rotation directioin is positive. The actual geometry of the belt is animated, not simply the texture co-ordinates, for a high detail setting it may appear to be still.

Apply a suitable shader to the belt. A good way to test the belt is to apply a checkers pattern then used a textured scene preview. If a texture map is used, such as leather, the vertical space or top to bottom is wrapped around the length of the belt and the horizontal space or left to right is wrapped around the cross section of the belt. The belt has two shading domains but only the flat and v-belts use the second domain for the sides. The number of repeats of a textured pattern can be adjusted on the Belt primitive. Procedural and fractal shaders should not be used directly with the Belt Drive because the geometry is constantly changing its position but 3D textures will remain in the same place. By selecting and exporting the finished belt in OBJ or 3DS format the procedural textures can be converted into texture maps ('baking' them) that can loaded in and used in their place.
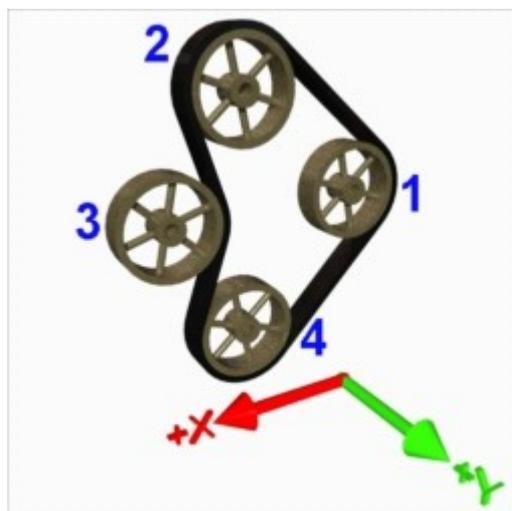
More complex configurations of pulley wheels must be arranged carefully. The belt will wrap around over the top of a wheel that is infront of the previous wheel and around underneath a wheel that is behind it. One wheel is infront of another

when it has a greater value in the axis opposite to the wheel axles. In this example the 2nd pulley is infront of the first wheel so the belt is wrapped around over the top of that wheel. The 3rd wheel has been set so that the belt wraps across it. The 4th wheel is behind the 3rd wheel so the belt passes underneath it. Finally the 1st wheel is behind the 4th wheel so the belt wraps back around underneath it to where it started. Which wheel is actually first in the system doesnt matter so long as they are all in the right order.

Setting the Belt Pulley modifier *Wrapping Mode* to *Across* for a pulley will not only wrap the belt over it differently but that wheel will spin in the opposite direction to the other wheels.

To control the animation adjust the Travel value in the Belt Drive modifier added to the pulley group. The first wheel in the group is always the drive wheel. Arrange or rearrange the order of the wheels in the group as necessary if the drive wheel is not the same size as the other wheels. When the Travel is set to 1.0 cycles per second the drive wheel will complete one whole revolution in one second and animate the whole system accurately.

Tensioners

Tensioners are an advanced feature and recommended for users who are familiar with the basic function of the Belt Drive plugin. In systems where the pulley wheels are able to move or be adjusted one or more tensioner wheels can be used to maintain a constant belt length. Without tensioners the pulley wheels can be moved around and the belt will simply be stretched. Two types of tensioners can be implemented, those which push the wheel in or out and those which swing it through an arc. A tensioner requires two target points positioned correctly to work, use target helper objects.
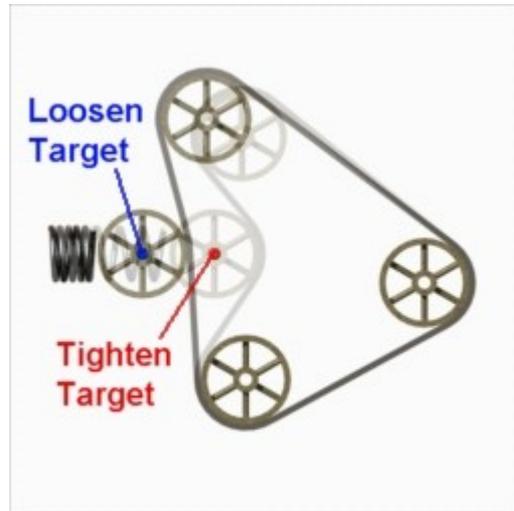
The normal straight tensioner pulley wheel will move toward the *Tighten target* point when the belt needs to be tighter and toward the *Loosen target* point when the belt is too tight.
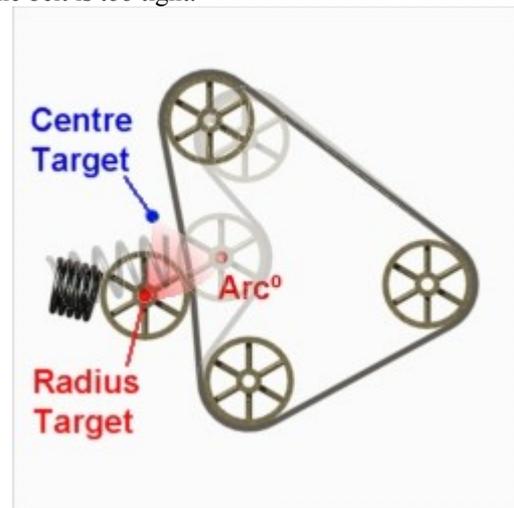
When the *Curved* tensioner option is used the tensioner will swing around the *Centre target* point at a radius measured from the distance to the *Radius target* point. The tensioner wheel will swing through an *Arc* away from the *Radius target* point to tighten the belt.

If the exactly and accurately modelled belt length is unknown then set the system up without the tensioners enabled or disable them in the Belt Drive modifier. Select the Belt primitive and enter the model room then take note of the *Measured Length*. Enter that value as the *Length of Belt Required* in the Belt Drive modifier.

Multiple tensioners can be given unequal weighting by changing the tensioner *Weight* values. A tensioner with a lower weight will not respond as much as a tensioner with a higher weight.

If the tensioners cannot be adjusted to maintain the required length then the belt will stretch or shrink to fit around the system.

The Belt Primitive UI
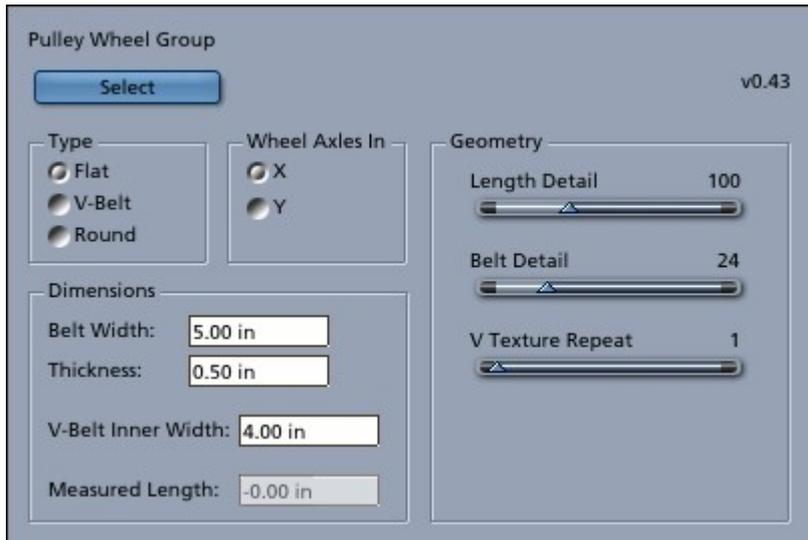
Select the Belt primitive in the scene list then enter the Model room to adjust its parameters.

The belt must have a *Pulley Wheel Group* to wrap itself around. That group of objects must have a Belt Drive modifier added to it and two or more wheels with a Belt Pulley modifier added to them inside the group. Users with Carrara 5 or higher can hit the button to bring up the object selection dialog box. Carrara 4 users must type in the exact unique name of the group.

Three belt *Type* options are available; a *Flat* belt with a square cross section, a *V-Belt* with a trapezoidal cross section and a *Round* belt with a circular cross section. The belts will always be wrapped around each pulley wheel with their cross section centred at the pitch diameter.

Two of the main axes are available to align the *Wheel Axles In*. This is the axis in which the pulley wheel axles are aligned when they have no rotation set. The belt itself will turn around in this axis.

The belt dimensions can be adjusted using scene units. Set the *Belt Width* for flat and v-belts and the *Thickness* for the height (or distance from the surface of the wheel). When the round belt type is selected the *Thickness* parameter becomes the *Diameter*. The *V-Belt Inner Width* is the inner surface of the belt that contacts the wheels and should be less than the belt width. The *Measured Length* cannot be changed by the user and is used to display the current total circumference of the belt around the wheels, this is helpful when using tensioners.

The level of detail of the belt geometry can be set using the sliders. Set higher detail levels if the belt appears to be too chunky. Using very high values will slow the belt down in the scene preview. The *Length Detail* gives the number of points in the mesh of the belt as it wraps around the wheels. The *Belt Detail* gives the number of points used in the cross section of round belts.

The *V Texture Repeat* value can be used instead of setting the tile count with texture maps or when using uv based patterned shaders. The v texture space runs along the length of the belt and the u texture space wraps around the cross section of the belt. For flat and v-belt types the u space is kept in proportion with the thickness and width settings to avoid distorting any texture map.

The Belt Pulley Modifier UI
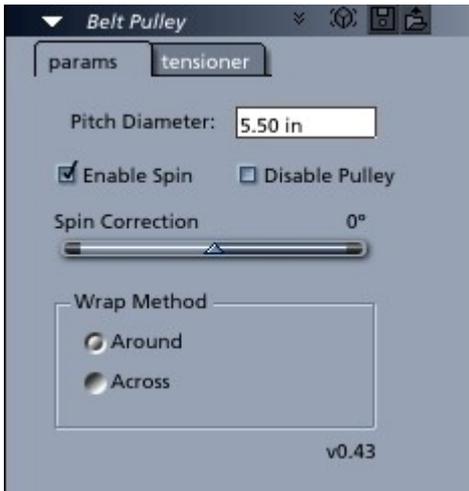
Every pulley wheel in the belt drive system must have a Belt Pulley modifier added to it. It can be found under the Sparrowhawke3D modifiers submenu.

The most important parameter is the *Pitch Diameter*. This adjusts the distance from the surface of the pulley wheel at which the centre of the cross section of the belt geometry will be wrapped. For old fashioned flat belts the pitch diameter is normally equal to the diameter of the wheel plus the thickness of the belt. V-belts and round belts require grooved pulley wheels, the pitch diameter must be set so the the belt sits in the groove and touches the surface of the pulley wheel in the middle of the groove.

The next most important setting is the *Wrap Method* which tells the Belt Primitive whether to wrap the belt *around* the outside of the wheel or to pass *across* the wheel on the opposite side. When the next wheel is infront of the previous wheel (in the axis opposite to the axle) the belt will wrap over the top of the wheel and *around* the outside or *across* the gap and then pass under the wheel.

Unchecking the *Enable Spin* box will prevent the Belt Drive modifier from turning the wheel when the system is animated.
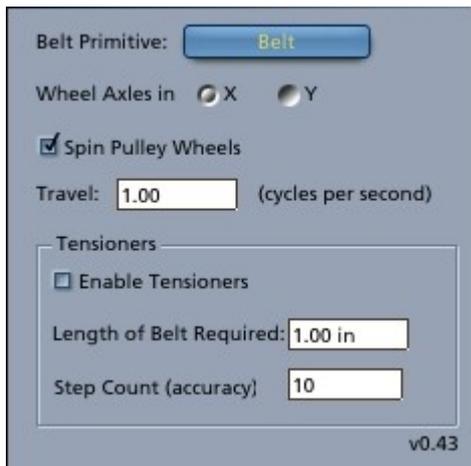
When the *Disable Pulley* option is checked the Belt primitive and Belt Drive will ignore the pulley wheel and the belt will pass by or through it.

Adjust the *Spin Correction* to rotate the pulley around its axle. This value can be key framed and used to orientate the spokes a particular way or to make the wheel slip. The correction is only applied after the Belt Drive has spun the wheel unless disabled.
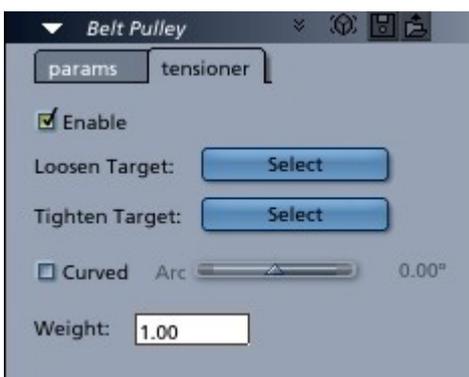
Tensioners are an advanced feature that allows the Belt Drive to automatically adjust and move a pulley wheel to maintain a constant belt length. Check the *Enable* box to activate any pulley wheel as a tensioner (note that this property can be key framed). Use the object selection dialog box to choose the target points. Carrara 4 users must enter exact unique names into the field. When the *Curved* option box is checked the target labels change and the *Arc* range of the curve can be set. The positive or negative rotation through the *Arc* from the *Radius* target follows the right hand rule.

The Belt Drive Modifier UI

Belt Primitive:	Belt
Wheel Axles in	◉ X ○ Y
☑ Spin Pulley Wheels	
Travel: 1.00	(cycles per second)

Tensioners

☐ Enable Tensioners

Length of Belt Required: 1.00 in

Step Count (accuracy)   10

v0.43

The Belt Drive modifier must be added to the parent group of the pulley wheels. This makes sure that the Belt is properly wrapped around those wheels and its used to animate the belt drive.

Set the *Wheel Axles in* axis to match the setting in the Belt primitive. This is the major axis in which the pulley wheels will spin. Leave the *Spin Pulley Wheels* box checked to keep them turning during the animation.

Adjust the *Travel* value to set the rate at which the drive wheel revolves in *cycles per second*. The first wheel found in the pulley wheel group is always taken to be the drive wheel.

Check the *Enable Tensioners* box to use the advanced tensioners feature. Enter the *Length of Belt Required* in scene units and the Belt Drive will move any pulley wheels that are enabled as tensioners to maintain a constant belt length. The *Step Count* value sets a limit on the number of 'guesses' that are made to find the right position for all the tensioners, using a very high value may slow down the Belt Drive.

Shading the Belt

The Belt primitive has two shading domains. Flat and v-belts can use both domains with one for the outer/contact surface and the second domain for the sides. Round belts only use the first shading domain. Applying a simple shader to the belt without using the domains will always cover the entire belt surface.
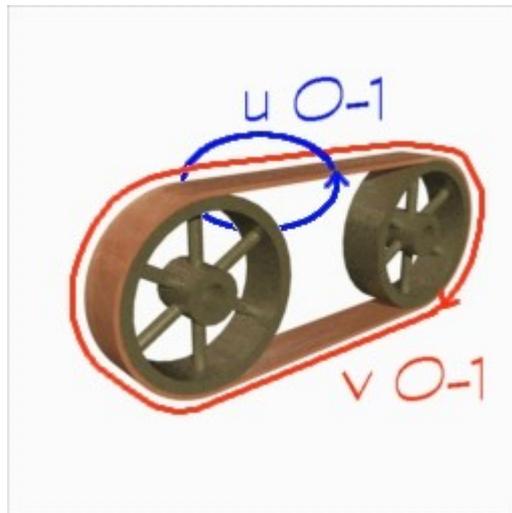
The v-texture space runs along the length of the belt with one or more repeats. Adjust the *V Texture Repeat* setting of the Belt primitive to increase the number of repeats. The u-texture space wraps once around the thickness of the belt. When flat and v-belts are used the u-texture co-ordinates are adjusted to keep the texture in proportion and thus prevent distortions.

The Belt primitive works best with seamless texture maps and 2D shaders. 3D procedural and fractal shaders will not animate properly because their patterns do not move with the belt but remain in the same place. Some fractal shaders have a UV space mode and if so they can be used with the Belt primitive but will be likely to show the seam. However any procedural shader can be used to generate texture maps by exporting the primitive from Carrara. Select the belt primitive then from the top menu choose File | Export. Set the *Save as type*: to use "Wavefront(*.obj)" format and make sure that the Options box is checked. In the OBJ Export dialog under the *Materials export* section select the "Convert Procedural Shaders to Textures" and set a resolution that will give enough detail. For best results use the same proportion of the belt thickness to belt length for the dimensions with the second number being the larger v-texture space. For flat and v-belts two texture maps will be generated one named "top" and the other "sides". If the two domains with layers have not been used the texture maps should be identical. If a large texture repeat count or tiling is used the generate maps will contain those tiles. Best results may be had by deliberately distorting and stretching the texture out in the v-space and then setting the repeat count for the texture map or by applying the complex shader to a simple plane primitive and exporting that to generate the maps. Carrara does not export bump maps, get around this by temporarily dragging the bump shader up into the colour channel, export it then undo the operation on the shader.

Tips and Tricks

- To create a belt drive system that is not orientated in X or Y set up the pulley wheel group then rotate it
- Pulley wheels do not have to line up exactly, the belt will angle when spanning the gap from one wheel to the next.
- Unlock and move the hot point in the axle axis then use a noise tweener to get a flat belt to slip around

Planned Improvements

- Add in Z axis option for pulley wheel axles
- Add tool tips to the user interface
- Allow for movement in the scene to turn the belt drive system with a multiplied factor so the belt can be placed in a vehicle etc
- Possibly add geometry for toothed timing belts

Support

Sparrowhawke3D is an amateur project so I can't make any promises regarding support, however I'm keen to improve my simple practical plug-ins and share them with the Carrara community.

Please do not distribute the Belt Drive beta.

For comments, feedback and suggestions email me, Jeremy@Sparrowhawke3D.com.