jc.clothes

Dec 2008

Overview Walkthrough Create Patterns Stitching Posing Simulation Conclusion Samples

Posing

Posing should be considered as a different process after stitching. The purpose of stitching is to build the default pose of a garment from its flat shape. Garment posing is analogous to character posing which starts from the default pose. In order to avoid going through stitching every time you pose a garment, it is necessary to separate the two processes and make posing start from the default pose. To do this, we have to duplicate garment at the end of stitching. The rest shape of this garment would be the shape at the beginning of stitching.

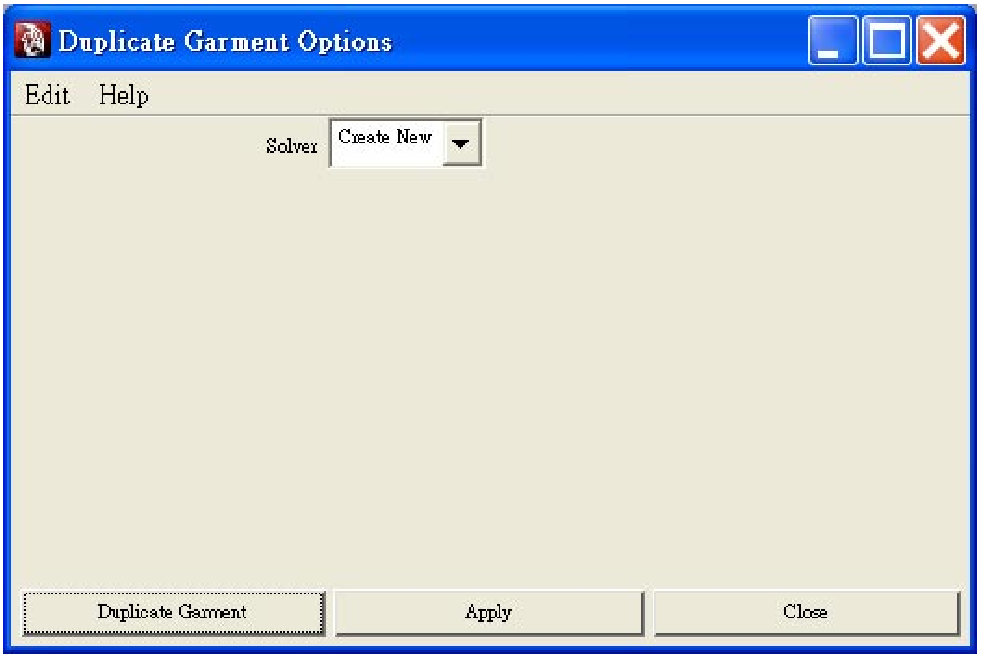
Duplicate Garment

To duplicate garment manually, select garment at the end of stitching. Invoke Edit -> Duplicate Special . From option box, select Edit -> Reset Settings, check "Assign unique name to child nodes". Press "Duplicate Special". The default (start) shape is now duplicated. While the shape is still selected, invoke nCloth -> Create nCloth.

To find the rest shape, open Attribute Editor while the new garment is selected. You'll see there're three shapes. Two of them are intermediate objects. To see them, uncheck their "Intermediate Object" attributes. The rest shape is the flat one before being deformed by stitching. While its tab is being selected in the Attribute Editor, press the "Select" button. Then ctrl-select the new nCloth node from Outliner and invoke Edit nCloth -> Rest Shape -> Connect Selected Mesh to Rest Shape. You can check its "Intermediate Object" attribute after this to hide it again. To make sure the connection is done, you can open Hypergraph to find out.

The next step is to duplicate the weld constraints. Because we do not want to make new stitches which would ruin the seams and affect subsequent deformation. So we want to duplicate stitches to the new garment to make the patterns weld together in exactly the same way as in the old garment. This is possible but it would involve manual duplication and connection of the dynamicConstraint and nComponent nodes using Hypergraph and Connection Editor. Because we need to duplicate the component lists which are present inside the nComponent nodes. The details will not be described here. You may rely on the Duplicate Garment command described below.

At the end of stitching, select garment and invoke jc.clothes -> Duplicate Garment. Its option box is shown below:



Solver Create New

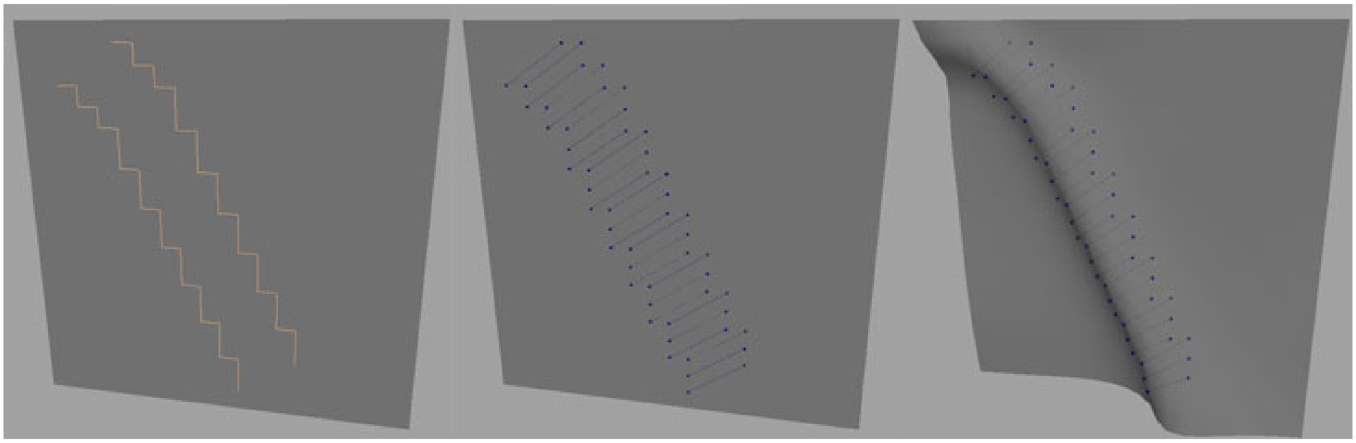
It has the option to let you create new solver for the new garment so that it won't collide with the old one which can be delete from the scene. It'll duplicate the rest shape to a new transform node and you'll see it in the Outliner. Only weld constraints will be duplicated. You'll have to create other contraints if there's any. It'll also disable and hide the old garment after execution. In fact, you can not only duplicate garment from the default pose (shape after stitching), you can also duplicate other poses so that you can perform different stages of posing. But these duplicated garments will share the same rest shape. That's why it is parented under a separate tranform so that it would remain intact in case you want to delete poses.

After duplication, the start frame of the new garment will be the beginning of the stitching process (start frame of the nucleus node which is 1 by default). This is when you start your posing simulation or animation of your character.

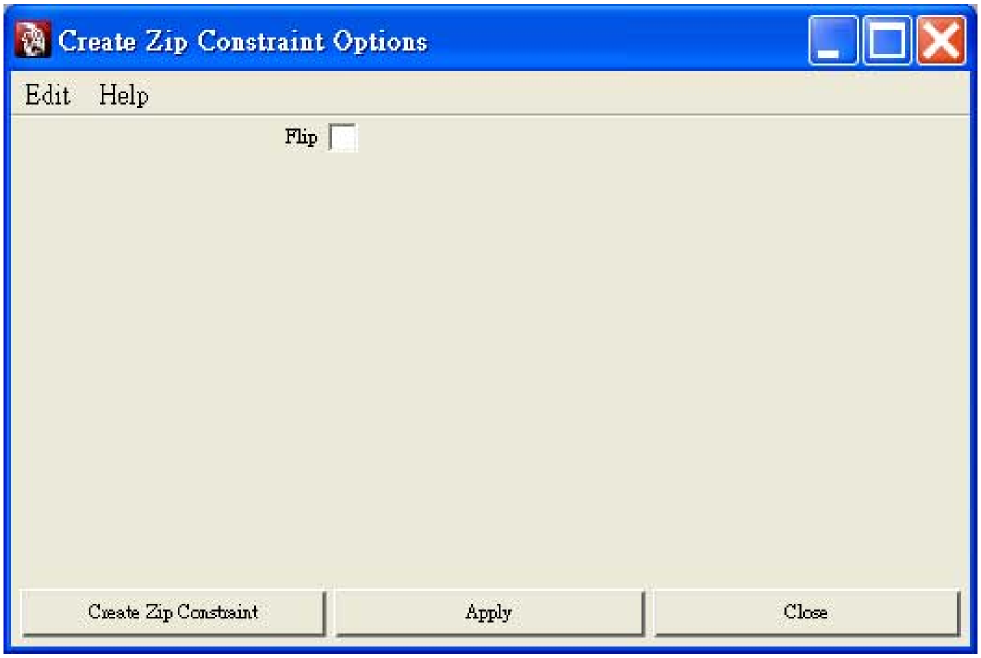
Group Constraints

To further control the shape of the garment, you need to build constraints to manipulate specific areas. This can be complicated because of the limited usefulness that the generic constraints can provide. That's why I think of group constraints which are groups of large amount of generic constraints. There're two of them: Zip Constraint and Hinge Constraint.

Zip Constraint is a group of component-to-component constraints. Each of them constraints only two vertices. Its intended use is to create wrinkle between two continuous edges. Vertices along one continuous edge are connected to those along the other as shown below:



To create Zip Constraint, select two contiuous edges which must contain equal numbers of vertices. Then invoke jc.clothes-> Create Zip Constraint. Its option box is as follows:

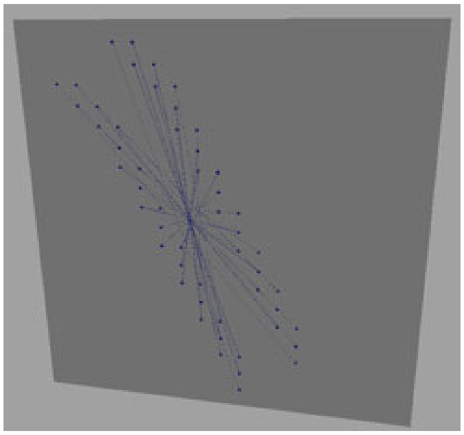


Create Zip Constraint Options

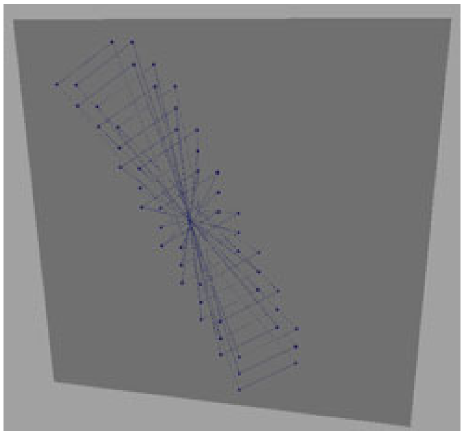
Edit Help

Flip □

The Flip option is to correct this undesirable case:



If the above happens, select the edges again (ie. those between the constrainted vertices), then invoke jc.clothes -> Create Zip Constraint with Flip toggled (ie. on or off depending on the previous state). Now you've got two Zip Constraints:

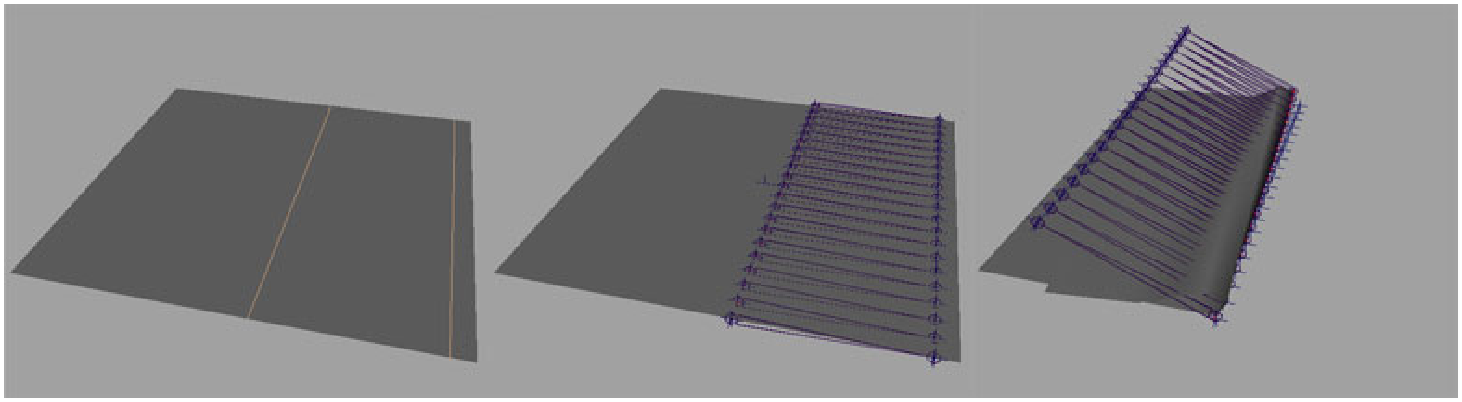


Select the previous Zip Constraint from the outliner. It is just a usual dynamicConstraint node whose attributes are connected to the corresponding ones in the individual component-to-component constraints parented under it. Then invoke jc.clothes -> Delete Group Constraint.

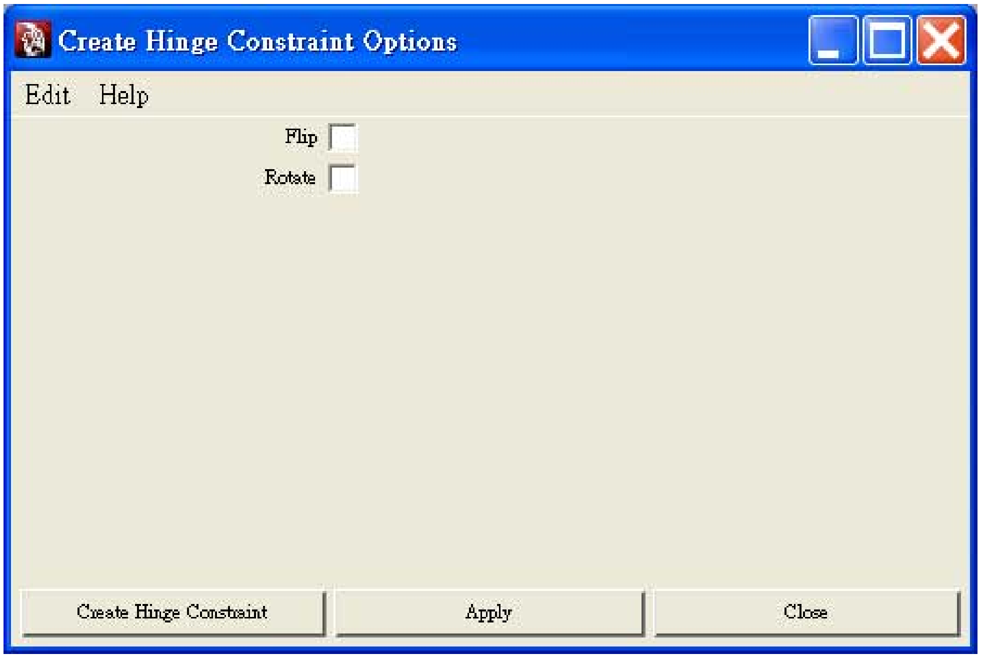
To create wrinkle, set key for the Zip Constraint's Rest Length Scale from 1 to something lower. The wrinkle will appear after simulation.

Another use of Zip Constraint is to connect two separate patterns like zipper connecting front patterns of a jacket. It can also be used to pull two continuous edges further away, ie. to stretch the area between them.

Hinge Constraint contains a lot of transform constraints which are grouped in pairs. Each of them works with a joint chain so that one transform constraint can rotate around another. If there're large amount of them along two continuous edges, the garment can be folded as shown below:



To create Hinge Constraint, select two contiuous edges which must contain equal number of vertices. Then invoke jc.clothes-> Create Hinge Constraint. Its option box is as follows:



Create Hinge Constraint Options

Flip □
Rotate □

The Flip option works in exactly the same way as that in Zip Constraint. The Rotate option is to toggle direction of joint chains.

To obtain the fold shown above, turn on Hinge Constraint and set z-rotation of joints to zero at the beginning frame. Turn off Hinge Constraint and set z-rotation of joints to something close to 180 at the ending frame. Play simulation and you'll have the garment folded.

This Hinge Constraint actually causes calculation cycles to exist. Inside Hinge Constraint, a follicle is created on the nCloth object and parents the root joint. The end joint parents a transform constraint. This transform constraint when being rotated would deform the nCloth object which would cause the follicle to move. So there's a cycle. But under this construction, because rotation of follicle is not connected to its transform node and the transform constraint at the root joint is outside the follicle (ie. no matter how the nCloth object is deformed, the root joint won't get translated nor rotated), these cycles wouldn't cause unpredictable result. So this constraint is safe to use.

Creative use of constraints is very important to pose garment for a desirable shape. There's no need to create constraints before stitching. They can be created at any stage of posing so that undesirable deformations can be fixed or desirable ones can be made.

Overview Walkthrough Create Patterns Stitching Posing Simulation Conclusion Samples