1. Feature Highlights

High Quality Cloth Animation with Less Pain Qualoth is a production-proven cloth simulator adopted by some of the biggest studios worldwide like DreamWorks Studios. Qualoth is based on a sophisticated physical model and numerical algorithms that produce realistic wrinkles and cloth animation, and delivering unmatched computational speed with fully multi-threaded solver. The reliable collision handling and rich set of controls allow users to spend more efforts on creative work rather than tedious trouble-shootings.

Robust Collision Handling Qualoth can handle collisions on fast-moving objects, and the self-collision technology can be resolved robustly even when the cloth is trapped between collided objects. The reliable collision handling allows users to create simulation for complicated multi-layered garments without artifacts.

Seamless Import from 3D Garment Design Tools Qualoth accepts quad-mesh, tri-mesh, or even non-manifold mesh as its input geometry. The shapes of curved 3D surfaces can be simulated as if they are molded to form such shapes in rest state, or they can be simulated as if they were created with panel-based method by utilizing UV coordinates for their initial flat shapes. The latter method enables users to utilize the output of dedicated 3D garment design tools such as DC Suite or Marvelous Designer which are designed for 3D creation of real-life garments.
**Goal Shape Control** Cloth animation is not necessary to be always physically valid. Users can add goal constraints to a cloth object and those goal shapes can be animated easily. The attraction force of goal constraints can be painted over the surface per-vertex wise.

**Partial Re-Simulation** Sometimes only the partial result of a cloth animation is unsatisfactory and in that case you can choose such vertices selectively which need to be re-simulated. This leads to faster computation and thus costs less time to achieve desired results.

**Bidirectional Interaction with Maya Fluid** Qualoth supports bidirectional interaction with Maya fluid. It is easy to make flapping flags, moving cloth in water, waving clothes in wind with maximum reality by combining fluid simulation with cloth simulation.

**Realistic Wrinkles** The material that is simulated on Qualoth is not a spring network. Its physical model is based-on deep understanding of cloth material property that differs from rubbers and other elastic materials. The realistic wrinkles and motions of cloth simulated by Qualoth come from an advanced physical model dedicated to simulating fabric-like materials.

**Supported Platforms** Qualoth supports Autodesk Maya 2011~2015 64 bit versions in Windows, Linux, and Mac OS.
2. Creating Cloth Objects
To create a cloth object, select a polygon mesh and perform “Create Cloth” command. Then following things happen.

1. The original polygon mesh is hidden, and used as the initial state of the cloth object.
2. A cloth node is created named as “qlCloth1”. This node contains cloth related attributes such as material properties
3. A solver node names as “qlSolver1” is created and attached to the cloth node. This node contains simulation parameters such as “frame samples”.
4. A output mesh named “qlCloth1Out” is created. You can perform operations on this mesh like painting attribute maps, applying constraints, or other cloth manipulation operators.

3. Solver and Cloth

A solver calculates the motion of cloth objects, and a cloth has its unique properties like material properties. A solver may have multiple cloth objects, and the cloth objects attached to the same solver can interact to each other through self-collisions or springs. If you create a cloth object, the cloth object is attached to one of the existing solvers automatically. However, you can change the solver assigned to a cloth object or assign a new solver to any cloth objects by “Change Solver” menu after creation of cloth object. When click “Change Solver” menu button, the existing solvers are listed with a new solver creation option at the bottom. If you want a cloth object to be attached to a specific solver at creation time, select the option box at the right of “Create Cloth” menu button.

“Create Cloth” Option Menu
“Change Solver” Submenu
4. Setting Cloth Node Attributes – Material Properties

A cloth object has five attribute categories. The first one is “Material Properties”. Each material property is explained below.

**Active**: Activates or de-activates this cloth object. If de-activated, this cloth node is ignored by the attached solver when it solves new frames, and displayed as wireframe. However, it reads the cache data if they exist.

**Length Scale**: Scale factor for length unit. You can make the cloth look bigger or smaller than the actual size represented by the geometry by setting this value.

**Density**: Mass density per unit area.

**Stretch Resistance**: Resisting force to planar stretching and compression (See Fig. 4-1.)

**Shear Resistance**: Resisting force to shearing. (See Fig. 4-1.) This parameter is interpreted as a scale factor to the stretch resistance. That is, if shear resistance is 1.0, it equals to the stretch resistance so that the resulting material exhibits isotropic behavior.

**Stretch Damp**: Damping factor for stretching motion.

**Bend Resistance**: Resisting force to bending.

**Bend Damp**: Damping factor for bending motion.
**Bend Yield**: Bending yield limit angle (degrees) at which plastic deformation triggers. After this limit angle, the bending angles of the edges (creases) start to deform permanently. Valid range is 0.0~180.0. This value is meaningful only when the Bend Plasticity is greater than 0.0.

**Bend Plasticity**: The amount of preserved creases. If this value is zero (perfectly elastic), all creases disappear completely under no external forces. If this value is 1.0 (perfectly plastic), all creases are preserved 100% under no external forces.

**Air Drag**: If there are Maya dynamic field such as Air Field or Turbulence Field connected to this cloth, the cloth motion follows the field and this value controls the amount of influence from those air fields. In case there is no attached field to this cloth, “Air Drag” simply drags the cloth motion in the direction of face normals of each triangle.

**Viscous Damp**: This damping force drags the motion of each cloth vertex in all directions uniformly regardless of the directions of normals of each vertex.

**Rubber**: This value scales the area of the cloth in rest state. Bigger values than 1.0 make the cloth expand, smaller values than 1.0 make it shrink.

**Friction**: Controls the friction among cloth vertices. The vertices in self-collision are affected by this friction value. Note that the friction between cloth and colliders are controlled by collider nodes, not by cloth node.

**Pressure**: The amount of pressure force which are applied to the vertex normal directions of each cloth vertex.

**Gravity**: Gravity force.

**Compression Resistance**: Resisting force to planar compression. When not overridden, this value is set to be equal to Stretch Resistance.

**Warp(U) Resistance Scale**: The scale factor to the planar stretching/compression resistance in warp(U) direction.

**Weft(V) Resistance Scale**: The scale factor to the planar stretching/compression resistance in weft(V) direction.

**Hysteresis Name**: When Bend Plasticity is non-zero, the rest angles of each polygon edges evolve as simulation goes on. You can save such evolved rest angles into a file which can be used later.

![Fig. 4-1: Illustration of cloth deformation modes](image-url)
5. Paintable properties

All material properties above except “Bend Yield” and “Bend Plasticity” are paintable. To paint each property, select a cloth and choose a map to be painted in “Qualoth>Paint Attributes>Cloth Attributes” as in the figure below.

Fig. 5-1 : Choosing an attribute to paint

The maps for material properties are scale factors for each property. The maps which do not belong to material properties are explained below.

fieldMap : scales the external field or fluid force. See “12. Applying Fields and Fluids”

wrinkleMap : specifies the bending angle for each edges. 0.5 corresponds to 0.0 degree, 0.0 to -180 degrees, and 1.0 to +180.0 degrees respectively.

selfCollisionMap : enable or disable self-collisions. Vertices painted under 0.5 are disabled, and vertices painted above 0.5 are enabled.

solidCollisionMap : enable or disable solid-collisions. Vertices painted under 0.5 are disabled, and vertices painted above 0.5 are enabled.

softConstraintMap : scales the strength of the soft constraints. See “13. Constraints”

thicknessMap : scales the thickness attribute for the collision handling. See “6. Setting Cloth Node Attributes – Collision Handling”
proximityCriterionMap: scales the proximity criterion. See “6. Setting Cloth Node Attributes – Collision Handling”

strainMap: This map is for visualizing how much each triangle is stretched or compressed. This map is not for editing by user. This map can be shown by clicking “Qualoth> Show Strain” or hidden by “Qualoth> Hide Strain” after selecting the cloth mesh. (See Fig. 5-2)

Fig. 5-2: Strain Map – Highly stressed region gets red.

The painted maps can be exported and imported to a file. To do that, select a cloth mesh whose maps are to be exported/imported and click “Qualoth> Save/Load> Save/Load Attribute Maps”. This export/import commands store and retrieve the weight values as floating point values, so the accuracy loss observed with the image export/import functions does not occur.
6. Setting Cloth Node Attributes – Collision Handling

**Solid Collision**: Activates collision handling between cloth and colliders

**Self Collision**: Activates self-collision

**Intra Cloth Collision**: Activates self-collision of the cloth object to itself. This option becomes active given that **Self Collision** is turned on.

**Inter Cloth Collision**: Activates self-collision of the cloth object to other cloth objects. This option becomes active given that **Self Collision** is turned on.

**Proximity Criterion**: Specifies the threshold for intra-cloth collision between cloth vertices within this cloth object. This value is automatically determined by the solver attached to this cloth, and can be overridden manually per each cloth object. (See Fig. 6-1,2)

**Thickness**: Specifies the threshold for inter-cloth collision between this cloth object and other cloth objects attached to a same solver. This value is assigned as the same value of proximity
criterion if not overridden. (See Fig. 6-1,2). The thickness also applies to the cloth-to-solid collision handling.

**Fig. 6-1 : Illustration of the difference between Thickness and Proximity Criterion**

**Fig. 6-2 : Illustration of thickness handling.** Collision is detected if the “Distance between cloth 1 and 2” is less than “0.5*Thickness of Cloth 1 + 0.5*Thickness of Cloth 2”.
7. Creating Colliders

To make a polygon object collide to cloth, select the cloth object and the polygon object to be used as a collision object, and click “Qualoqh>Create Collider”. Then, following things happen.

1. A collider node named “qlCollider1” is created.
2. A collider output mesh named “qlCollider1Offset” is created and initially hidden. This mesh is used for painting collider’s attributes such as offset, friction, etc.
3. By default, the collider is set to collide to all the cloth objects under the same solver the first selected cloth belongs to.

The attributes of a collider are,

**Active**: Activates or de-activates the collider. This flag is also paintable by “Collision Map”. See Fig. 7-2 for choosing the attribute to paint.

**Offset**: Collision offset to the original collision mesh. This value is paintable over the offset mesh created per collider (qCollider1Offset in the example).

**Friction**: Friction between the cloth and this collision object.

**Priority**: The collider priority. As this value gets lower, the priority gets higher. This value is paintable, and the final priority of a vertex of collision object is determined as the sum of “Priority” attribute (an integer value) + “Priority Map” value (a floating point value between 0.0–1.0). See Fig. 7-1 to understand how the collider priorities are treated.
Fig. 7-1: Illustration of collision handling with colliders having different priorities

Fig. 7-2: Painting collider attributes – Choose the original collider mesh, and choose a map from “Qualoth>Paint Attributes>Collider Attributes”.

15
8. Controlling Collisions between Objects

When you turn on self-collisions on cloth objects under the same solver, all the cloth objects collide to each other. Similarly, any collider set to collide to a cloth object is also set to collide to other cloth objects under the same solver. Users might want to override this default behavior, and it can be done by “Qualoth>Enable/Disable Collisions Between Two Objects”.

The following Fig. 8-1 shows the default collision settings between three cloth objects and one collider under a solver. All possible collisions between objects are handled.

However, considering the configuration of the cloth objects and collider, one might want to reconfigure the collision handling between objects as following Fig. 8-2.

Changing the collision settings as above can be achieved as follows.
- Select Cloth 1 and Cloth 3, click “Qualoth>Disable Collisions Between Two Objects”.
- Select Cloth 2 and Collider 1, click “Qualoth>Disable Collisions Between Two Objects”.
- Select Cloth 1 and Collider 1, click “Qualoth>Disable Collisions Between Two Objects”.

You can enable/disable the collisions between any two objects at any time, and this flag is keyable. For example, the collision relationship node between Cloth 1 and Collider 1 is created and named as “qlClothShape1__to__qlColliderShape1_collision” after disabiling(or enabling) the collision between those two objects.
The “Collision” check box can be keyed in the above qlCollisionRelation node. To find this node, just select the corresponding cloth locator or collider locator, then the related collision relationship nodes will appear in the attribute editor or in the channel view. (See Fig. 8-3.)

Fig. 8-3: Finding the Collision Relationship Node related to Cloth Objects.
9. Setting Solver Attributes

There are two categories in the solver attributes, which are “Simulator Attributes”, and “Collision Attributes”.

![Attribute Editor](image)

**Simulator Attributes**

**Collision Attributes**
**Simulator Attributes**

**Active**: Activates or de-activates this solver

**Solver Statistics**: Prints the solver statistics as follows.

```
qlSolverShape1 FRAME 1:  #CG:5  SOL:0.009  COL:0.013  DEP:0.003  TOT:0.025  ACCUM:0.025
```

Each field in the above printed information means,

- `qlSolverShape1`: the name of the solver node.
- `FRAME 1`: the current frame
- `#CG`: the number of iterations of the dynamics solver
- `SOL`: the time spent for dynamics computation in seconds
- `COL`: the time spent for collision handling both for self and solid collision
- `DEP`: the time spent for evaluating the input dependency nodes such as colliders, constraints, springs, fields, etc.
- `TOT`: the total time of computation per frame = SOL + COL + DEP
- `ACCUM`: the accumulated computation time for this solver.

**Simple Subsampling**: Determines the sub-sampling method (simple vs. exact). If simple sub-sampling is turned on, the input nodes such as collision objects are interpolated linearly during sub steps, or evaluated exactly at each sub steps otherwise. For fast rotating collision objects, using “exact” sub-sampling methods would produce more accurate results at the cost of increased computation time (which can be verified in DEP time in the printed solver statistics).

![Simple Subsampling](image1)

![Exact Subsampling](image2)

**Fig. 9-1**: Simple sub-sampling vs. Exact sub-sampling of a Collision Object

**Partial Resimulation**: When turned on, only the selected cloth vertices are simulated. This option is only meaningful when all the cloth objects have caches.

**Start Time**: the frame number when this solver starts to compute. This value is connected to the “Start Time” of all the cloth objects under this solver.
Frame Samples: the number of sub steps to compute a frame. Using bigger numbers results in better accuracy in collision handling at the cost of increased computation time. 1~3 is reasonable for slow motions, 8~20 can be used for fast motions.

Time Scale: The time stretching factor. This value scales the time unit used in Maya. Suppose that one frame corresponds to 1/24 seconds in Maya, and if you set “Time Scale” to 10.0, one frame is interpreted as 1/240 seconds by the solver. You can create slow motion effect (bullet time effect) by controlling this value.

Length Scale: The length scale makes the cloth objects under this solver look bigger or smaller than the actual size which is represented by the geometry. Note that each cloth object has “Length Scale” attribute and “Length Scale Map”. The final effective length scale of a cloth vertex is calculated as follows.

Effective length scale = Cloth Length Scale X Cloth Length Scale Map X Solver Length Scale

MAX CG Iteration: The maximum number of iterations for dynamics computation. To obtain the best results, let the solver not to hit this limit. You can verify the number of iterations used at each frame in the printed solver statistics (#CG)

CG Accuracy: The accuracy index for dynamics iterations. As this value gets bigger, more accurate results are obtained at the cost of increased number of iterations. 4~6 are reasonable for not-stiff cloth, and use 7~9 for stiff cloth. Values greater than 9 are not recommended.

Gravity: Gravity force. Note that each cloth object can have its own gravity. If it is not overridden by each cloth, this value is used by default.

Show Vertex Status: Draw colored dots to show the status of each cloth vertex. You need to unhide the cloth locators to see the colored dots. Green dots represent the vertices in self-contact, blue vertices represent the vertices in contact with collision objects. The red dots show self-intersected vertices which are being pushed backwards.

Consistent Damping: When you increase the frame samples after adjusting material properties, you can see the damping effect is lessened so that the resulting motion of cloth looks different. This is an artifact caused by numerical algorithm used for solver. To remove such artifacts, you can turn on this flag to make the cloth exhibit consistent damping behavior regardless of the frame samples being used.

Reference Frame Samples: When you turn on “Consistent Damping”, this value is used as the reference frame samples. That means, the damping effect appears as if the frame samples were this value other than the actual frame samples being used by the solver.

Acceleration Scale / Velocity Scale: Reserved for future use with “Local Space Simulation”

Post Sim Script: A Mel script name which is invoked at every frame (not at sub steps) after solver computation.

Collision Attributes

Self Collision: Activates self-collision. This flag can be overridden by each cloth.


Self Continuous: When resolving self-collision, detect and resolve continuous collisions.

Self Proximity Force: The repelling force for proximate cloth vertices.

Solid Collision: Activates cloth-to-solid collision. This flag can be overridden by each cloth.
**Sharp Feature**: Perform extra computation for handling sharp features of solid object.

**Solid Proximity Force**: The repelling force between cloth and solid objects in contact. The following Fig. 9-3 shows how continuous collision detection differs from proximity detection. Solid collision handling uses continuous collision detection by default, while users can choose to use it or not for self-collision handling.

![Diagram of collision detection](image)

**Fig. 9-3**: Proximity Check vs. Continuous Check: Continuous collision detection does not miss the thin wall (right) in contrast to proximity only detection (left) since all the trajectories of cloth vertices are checked against colliders.
10. Caching the Simulation Data

As seen in the above attribute editor, a cloth node has its own cache attributes. If no cache file name is specified, a temporary file is created and used by default.

**Start Time**: This value is controlled by the attached solver. Specifies the initial frame offset for caching.

**Per Frame Cache Folder**: A folder name where the cache files are stored

**Cache Name**: Cache file name

**Cache Subframes**: The number of subframes to be stored for a frame.

If “Cache Subframes” is 1, the cache file names are generated as,

```
Cachename.0001
Cachename.0002
Cachename.0003
```

In case “Cache Subframes” is 3, the cache file names would be

```
Cachename.0001
Cachename.0001.333
Cachename.0001.667
Cachename.0002
Cachename.0002.333
```

The “Cache Subframes” could be useful for 3D motion blur for fast rotating movements since the cloth states during the sub-steps are stored to the cache files. The best results will be obtained if this
value coincides with the “Frame Samples” of the solver, since the actual cloth states of sub-steps are being computed according to the “Frame Samples”.

**Connecting existing cache data to a polygon object**
Once you created the cache file, you can apply it to a normal polygon object which has the same topology with the cloth which was used for generating the cache data.

Select a polygon mesh, and click “Qualoth>Connect Objects>Cache”. Then a file dialog will pop up to choose a cache file. Select one of the per-frame cache files regardless of the frame number. Once the cache node is attached to the polygon object, it will deform according to the cache as if it is a cloth object. A cache node simply plays the stored cache, and much lighter than a cloth node since it has no capability of simulating new data.

![Image of a cache node attached to a polygon mesh](image)

A cache node has two attributes: “Cache Name” and “Start Time”.

---
11. Blending the Caches

Frequently, users try multiple times to get satisfactory simulation results and sometimes only some parts of the cloth mesh are satisfactory and other parts are better at different simulation results. In this case, users can combine each part into a single cloth mesh using “Mesh Mixer” of Qualoth.

Suppose that there are three cached meshes (cloth1, cloth2, and cloth 3), then select those three meshes and click “Qualoth>Create Mesh Mixer”.

Then, a new mesh is created with a Mesh Mixer node attached. This node has weights for each input mesh, and in this case there will appear three weights named after each input in the attribute editor. (See Fig. 11-1)

Since those weights are equal to each other initially, the output will be the exact average of those three inputs. By changing first two weights to zeros, the mixed mesh will look exactly like the third input (cloth 3). See Fig. 11-2
Fig. 1-2. Changing weights of mesh mixer inputs

The weights are automatically normalized, so users don’t need to adjust the sum of the weights to 1.0 by themselves. To assign different weights to each vertex per each input, you can paint the weight maps assigned to each input. (See Fig. 1-3)

Fig. 1-3: Painting weight maps of a mesh mixer

Select the mixed mesh and right click to choose a weight name for each input meshes. There will appear five weight maps paintable in the list, and you can match the weight map to the corresponding input mesh by the number. (See Fig. 11-4)
In the above example, since there are only three inputs, only the three weight maps are used, and the remaining two maps are discarded. The number of weight maps are fixed and limited to five, so the maximum number of inputs for a mesh mixer node becomes five, too. Note that each weight map finally multiplied by the corresponding weight value in the attribute editor before computing the actual weight for a vertex per input mesh.
12. Applying Fields and Fluids

Applying dynamics field

To attach a Maya dynamic field to a cloth node, do followings.

1. Create a Maya field while selecting nothing. Do not select the cloth mesh at this time. See Fig. 12-1.
2. Select cloth vertices (not the mesh) to be affected by the field, and then select the field node. (The selection order is important) See Fig. 12-2.
3. Click "Qualoth>Connect Objects>Field>As Velocity".
4. A field filter node is created and now you can edit the properties of the node. (Fig. 12-3)

Fig. 12-1 : Step 1 - Creating a Maya dynamics field. Nothing is selected when creating the field.
Fig. 12-2 : Step 2 - Creating a Field Filter node by selecting cloth vertices and a field node.

Fig. 12-3 : Step 3 – Editing the resulting field filter node
To see the air field effects, do the followings

1. Set the *Length Scale* of the cloth solver to 100.0. (1x1 unit plane with 30x30 subdivision)
2. Set the *Magnitude* of the air field to 50.0.
3. Set the *Attenuation* of the air field to 0.0.
4. Turn off *Use Max Distance* of the air field.
5. Hit play.

The resulting simulation is shown at Fig.12-4. The actual force delivered to each cloth vertex is affected by the following factors.

1. The *Magnitude* of the input field.
2. The *Multiplier* of the field filter node. This factor is multiplied to the input field magnitude, and the default value is 1.0.
3. Field filter’s weight map. This map can be painted using the newly created filed filter weight mesh. This mesh is initially hidden as soon as created, so users must unhide it before painting the maps on it. (Or, user can paint it by “Qualoth>Paint Attributes>Field Filter Attributes>Weight Map” after selecting the field filter node). The name of the mesh is the combination of the field filter node name and the Maya field name, and it looks like “qiFieldFilter_airFieldShape_weightMesh”.
4. The *Air Drag* value in the material property of the cloth node. This value is multiplied to the input field magnitude.

*Fig. 12-4:* A cloth object affected by a Maya dynamic field.
5. *Field Map* of the cloth node. This map can be painted on the cloth mesh, and is multiplied to the sum of all field inputs.

![Field Map of the cloth node](image)

**Fig. 12-5 : Connecting a field “As Velocity” or “As Force”**

Connecting a Maya field to a cloth object can be done in two different ways, “As Velocity” or “As Force”. “As Velocity” interprets the output of the field node as velocity field, and “As Force” interprets the output as force field. When it is interpreted as velocity field, the field drags the cloth and the surface normals of each cloth triangle play an important role for resulting motion. When it is applied as force, the field forces are directly applied to each vertex regardless of the surface normals. The difference is illustrated in Fig. 12-6.

![Comparison between “As Velocity” and “As Force”](image)

**Fig. 12-6 : Comparison between “As Velocity” and “As Force”**
Applying fluid effects

To attach a fluid effect to a cloth object, select a 3D fluid container and cloth mesh, and click “Qualoth>Connect Objects>Fluid>As Velocity”. Fig. 12-7 shows the resulting effects. You can adjust Air Drag value to reflect the fluid field at different scale. In the figure, the fluid container is created as 1x1x1 unit cube, and an emitter is created at the bottom of the container with default values. In the example, the Air Drag is 0.1, and Length Scale is 100.0.

Fig. 12-7: A cloth object affected by the field from 3D fluid container

If you make the fluid collide to cloth, you can obtain bi-directional coupling effects as shown in Fig. 12-8. To do that, select the container and cloth mesh, and click “Fluid Effects>Make Collide”.

Fig. 12-8: Bi-directional coupling with 3d fluid and cloth object.
13. Constraints

Point Constraint

To create a point constraint node, select cloth vertices to constrain, and click “Qualoth>Constraint>Point Constraint”. A point constraint fixes the selected vertices to the current positions, and those vertices can be transformed according to the transform of the constraint locator.

The attributes of a Point Constraint node are,

- **Active**: Activates or de-activates this constraint
- **Enable Collision**: Enable or disable self-collisions of the constrained vertices.
- **Soft**: When turned on, the constraint is treated as springs.
- **Stiffness**: This value controls the stiffness of the constraint. 1.0 makes it behave almost like a hard-constraint, and 0.0 makes it unconstrained.
- **Damping**: This value controls the damping effect of the soft constraint. 1.0 corresponds to the critically damped response, i.e., produces no oscillation.
Line Constraint

A line constraint fixes the moving directions of the selected vertices. In addition to the attributes of Point Constraint node, Line Direction vector attribute is added. The line direction vector can be changed by the manipulator. (Select the constraint and click “Show Manipulator Tool” button.

Plane Constraint

A plane constraint makes the selected vertices move perpendicular to the plane normal direction. That is, the movement along the plane normal directions is eliminated. The Plane Normal can be changed by the manipulator.
Attach Constraint

An attach constraint is used to fix vertices relative to a moving object. At the creation time, each vertex position relative to the closest polygon is computed, and those vertices move along the polygon object afterwards. You can attach vertices to a deforming surface such as a character’s skin.

A typical example use of attach constraint is fixing the waist line of a skirt or pants to the body. Without the constraint in the above figure, the skirt will fall down due to gravity.

Using Soft Constraint Map

All the constraints explained so far can be further controlled by “Soft Constraint Map”, which is a paintable attribute of a cloth node. If the constraint is set to be a “Soft” constraint, this map scales the stiffness of the soft constraints. Suppose that you attached all the cloth vertices to the body surface. By painting the soft constraint map, you can choose which parts will be simulated and which parts will be fixed to the body with continuous weights ranging from 0.0~1.0.
Goal Constraint

A goal constraint is used to directly specify the desired shape of the cloth mesh. Any polygon mesh topologically same to the cloth mesh can be used as a goal constraint. To create a goal constraint, select a cloth mesh and a goal mesh, then click "Qualo>Constraints>Goal Constraint".

Then a goal constraint node and an auxiliary goal mesh are created. The auxiliary goal mesh is initially hidden, and it is used for painting goal-related maps which will be explained later.

Note that you can animate the goal mesh itself using deformers, and you can also use multiple goal constraints for a single cloth node. The methods to attract the cloth mesh to the goal shape can be chosen at the "Attraction Method" attribute of the goal constraint. It has three options.

**Vertex Spring**: Create springs per each vertex pair between the cloth mesh and the goal mesh. In this case, the "Strength" attribute becomes the spring stiffness, and the range is unlimited.

**Wind Field**: Create an air field to attract each triangle to the goal shape. Since the air field acts on the face, not on the vertex, the attracting movement might be more natural.

**Constraint**: Create soft constraint per each vertex pair. In this case, the range of *Strength* value is limited to 0.0 to 1.0. 1.0 corresponds to almost hard constraint, and 0.0 corresponds to unconstrained condition.

The **Vertex Spring** method and **Constraint** method are essentially same, but only differ in mapping the *Strength* value.
In any case, the attraction force can be arbitrarily large unless "Enable Force Limit" is turned on. For example, the vertices farther to the goal vertices are moved much faster than the vertices closer to the goal vertices. This creates some unnatural movements. To reduce this effect, "Force Limit" is used to prevent abrupt changes of the cloth shape to the goal. See Fig. 13-1 to see the effect of using the goal force limit.

**Fig. 13-1 : The effect of the goal force limit (right) compared to the unlimited goal force (left)**

A goal constraint has three paintable attributes: "strengthMap", "dampingMap", and "forceLimitMap". Those attributes are only paintable on the auxiliary goal mesh named as "<Original_Goal_Mesh_Name>_Goal". To paint the maps, you can choose a map from "Qualoth>Paint Attributes>Goal Constraint Attributes" after selecting the original goal mesh, or paint a map directly on the auxiliary goal mesh as in the figure below. All the maps are scale factors to the corresponding attribute values.
14. Creating Springs

To create a spring, select one or two vertices of the cloth meshes under the same solver. If you select single vertex, the spring connects the vertex and an anchor shown in Fig. 26. The anchor is a transform which can be animated. If you select two vertices, the spring couples those two vertices given that they belong to a same solver.

The attributes of a spring are,

- **Active**: Activates or de-activates this spring
- **Vertex Index 1**: The vertex index for the first vertex.
- **Vertex Index 2**: The vertex index for the second vertex. This index is set to -1 in case of a single vertex spring.
- **Stretch Stiffness**: Spring stiffness coefficient for stretching.
- **Compression Stiffness**: Spring stiffness coefficient for compression.
- **Damping**: Spring damping coefficient.
- **Rest Length**: The length the spring tries to maintain.
- **Break Length**: The spring is de-activated beyond this break length. Once broken, it is not automatically restored to the bonded state no matter what the current length is.
- **Original Length**: The length of the spring at the creation time. This is not editable by user.
- **Current Length**: The current length of the spring. This is not editable by user.
15. Create Attach Springs

Attach springs are used to attach a group of vertices of a cloth mesh to another cloth mesh. It acts like the attach constraint which attaches cloth vertices to solid mesh. To create an attach spring, select a group of vertices of a cloth object and another cloth object mesh, and click “Qualoth> Create Attach Spring”. Then, the selected vertices will be attached to the closest points of the other cloth object.

The attributes are,

- **Stiffness**: Spring stiffness coefficient.

- **Damping**: Spring damping coefficient.

- **Rest Length Scale**: Scales the rest length of each individual spring. Each rest length is measured from the initial distance between the corresponding vertex and the closest point of the other cloth surface.

- **Break Length**: Each individual spring is de-activated beyond this break length.
16. Weld Proximate Vertices

This command can be used to attach nearby cloth vertices to each other automatically within a threshold. Select one or two cloth meshes, and click “Qualoth>Weld Proximate Vertices”.

The command option window looks like,

![Command Option Window](image1)

All cloth vertex pairs within “Proximity Threshold” are bonded together by each qlSpring object. To exclude internal cloth edges for welding candidates, turn on “Border Vertices Only” which confines the pairs to be searched within mesh border vertices, and turn on “Avoid Self Welding” by which each vertex of a pair shall belong to different cloth objects.
Note that this function is similar to the attach spring, but the vertices to be welded are not specified directly, but found automatically with the given criteria. Also, this command allows attaching vertices of a cloth object to those of the same cloth object, which enables some interesting effects such as volumetric soft body simulation. For example, create a unit cube and convert it to a cloth, and weld vertices using the following options.

The resulting cube will behave like a soft volumetric object, and with high stiffness, it will look like a rigid cube.
17. Local Simulation

By local simulation, users can simulate the cloth objects without increasing the time. When you start local simulation, the current frame is fixed but the selected cloth objects are simulated and the cache of the current frame is replaced with the new result until the local simulation is stopped.

To initiate local simulation, select cloth objects and click "Qualoth>Local Simulation>Start". To stop the simulation, click ESC key or "Qualoth>Local Simulation>Stop". Note that when you try to stop the local simulation, the cloth being simulated must be in the selected state.

While the selected cloth objects are simulated locally, they will collide to the other cloth objects attached to the same solver unless they are de-activated.

If you performed local simulation of a cloth object at the start frame of the solver, the initial state of the cloth mesh will be replaced with the locally simulated result. If you want to restore the original initial state, select the cloth mesh and click "Qualoth>Reinitialize Solver".

What “Reinitialize Solver” does

When you click "Qualoth>Reinitialize Solver", following things happen.

1. The cache of the selected cloth object is cleared.
2. The initial state of the cloth is restored to the original mesh.
3. The hysteresis information is initialized if it was used.
4. The velocities are initialized to zero.

Basically, “Reinitialize Solver” turns the cloth object to the original state as if it is just created, except the painted maps. To reset the maps to the initial state too, click "Qualoth>Reinitialize Maps".
18. Manipulating the Rest Shape

The rest shape of a cloth is the shape the cloth returns to when there is no external force. Typically, the original mesh used for creating cloth object is used as the rest shape, as well as the initial shape. However, the rest shape can be changed with no need to modify the original mesh. One of the methods is connecting an explicit polygon mesh as the rest shape to the cloth object. To do so, select a cloth object and the rest shape then click “Qualoth>Connect Objects>Rest Shape”. Note that those two meshes must be topologically identical.

The polygon mesh connected in this way can be deformed and animated during simulation. The cloth solver retrieves two kinds of information from the rest shape. One is the stretch information and the other is bending information. The stretch information comes from the shapes of each polygon, and the bending information comes from the edge angles of adjacent polygons.

As shown in the above image, users can determine each information to be updated or not. If you turn off “Update Bending”, only the stretch information will be updated to the solver from the rest shape, or vice versa. The turned off information comes from the original mesh instead.

A typical case of using this separated stretch/bending updating is when handling stretching limbs of a character. An arm of a character may be stretched far more than the original size, and if the cloth the character is wearing does not comply with it, the bare arm will be exposed which is probably not desirable. In this case, users can use a rest shape and animate it to comply with the stretching arm. In this case, the wrinkles near the elbow are desired to be simulated naturally, not from the rest shape, so users can turn off updating bending information from the rest shape for that purpose.
Note that the rest shape is different from goal shape used by “Goal Constraint”. The solver attract each cloth vertex to the goal shape in world coordinate, so any transform applied to the goal shape will be tracked by the goal constraint, while the rigid transform (rotation and translation) of the rest shape does not affect the solver at all. Another difference is that the attracting strength for goal constraint is not relevant to material properties such as the stretch or bending stiffness, while the stretch and bending stiffness count for attracting the cloth to the rest shape. As the stretch stiffness gets bigger, the cloth polygons will be attracted to the rest shape more strongly, and the wrinkles of the rest shape will bend the cloth edges more strongly with bigger bending stiffness.

Fig. 18-1: Difference between goal shape and rest shape. The goal shape attracts the cloth in world space, and the rest shape deforms the cloth in local space.

As mentioned before, in case there is no explicit rest shape connected, the original mesh is used for rest shape as well as the initial shape. Even in that case, users can further control how the rest shape is computed from the original mesh. If you click the “input connections” button in the attribute editor after selecting the cloth node, you can see a “Converter” node whose name is typically “qlConvert1”. This node is attached to every cloth node and what it does is converting the polygon mesh data to simulation-ready geometric information data. This node actually computes the rest shape of the cloth from the original input mesh. This node has two options: “Preserve Wrinkle” and “Metric From UV.”
With “Preserve Wrinkle” turned on, the bending angles of the original mesh are used for computing the rest angles of each edge. Otherwise, only the area and shape of each polygon are used for rest shape and the rest angles of the edges are set to be zeros.

“Metric From UV” is used to compute the rest shape of each polygon from the UV map, not from the 3D geometry. This option is especially useful to create panel-based cloth effects for 3D modeled cloth geometry. If you turn off “Preserve Wrinkle” option and use the UV map for the rest shape, the cloth will deform as if it was created from flat panels.

Flat rest shapes are crucial to create natural draping and cloth-like wrinkle patterns, but most cloth meshes are sculpted in 3D space rather than stitching flat panels. These days, several dedicated 3D draping solutions are commercially available, and most of them export the UV maps with the same shapes of flat panels originally used for 3D draping as well as the draped 3D geometry. In this case, using “Metric From UV” would maximize the utilization of the output of the 3D draping solutions.

Fig. 18-2 : An illustration of the use of “Metric From UV”. The object at the right used a sphere as the original input geometry, and turned “Metric From UV” on.
19. Misc. Functions

Update Initial Pose

Select a cloth mesh and a reference mesh which has the same number of vertices with the cloth, and click “Qualoth>Update Initial Pose”. Then, the cloth mesh will match the reference mesh and the simulation will start from the given reference mesh.

When a garment is set up for character animation, all the cloth solver settings such as constraints, material properties, or colliders are used unchanged across multiple sequences. At each shot, only the starting pose of the character is different from others. In this case, users do not need to setup the whole things again from scratch. You can import the animation data to the character first, create a reference cloth mesh to fit the character pose at the start frame of animation, and use “Update Initial Pose” command. Then the simulation starts with the given initial pose of cloth.

Fig. 19-1: The reference mesh is created to fit the character body at start frame, and the cloth mesh is located at other place. After “Update Initial Pose”, the cloth mesh matches the reference mesh.

To create the reference mesh to be used as the initial pose, users can just transform a copied cloth mesh and deform it, or perform draping simulation to fit the character pose exactly in a separate process. In the latter case, a smooth transition from draped pose (T or A pose) to the starting pose of the character being animated might be needed for fitting simulation.
Bake Mesh

Any polygon mesh being animated can be baked into cache file which can be used for a cloth object. For example, the output mesh of the Mesh Mixer (See 11. Blending the Caches) can be baked into a cache file using this command. Select a polygon mesh and click “Qualoth>Bake Mesh Per Frame”. Then after selecting a file name, the geometry animation in the range of time slider is baked into per frame cache files which are acceptable as a cloth cache.

Update Tweaks

The simulated results might be not satisfactory at small portion of the cloth at certain frames. In that case, users can modify the cloth vertices directly using move tools or smoothing tools, and reflect such modification into the cache file. After modifying the cloth vertices at the given frame, select the cloth mesh and click “Qualoth>Update Tweaks”. Then the modified results are written into the cache file at the current frame. If the simulation restarts from this modified frame, the currently updated modification is used as the restarting pose.

Button Objects

A small polygon object such as buttons can be attached to the cloth mesh using “Qualoth>Connect Objects>Button Object”. Then the button object will follow the cloth surface deformation afterwards. At the creation time, the button object computes the transform relative to the closest triangle of the cloth mesh, and the relative transform is kept during animation. In fact, the button objects can be attached to any kind of polygon objects as well as cloth objects.
**Mesh Extrusion**

Select a cloth mesh and click “Qualoth>Cloth Mesh Extrusion”. Then a new mesh with non-zero thickness is created. The thickness can be controlled by “Offset” attributes of the “Rounded Extrude” node, and the subdivision for the side wall can be controlled by “Subdivision” attribute. The side walls of the resulting mesh are rounded, which distinguishes itself from the normal extrusion of Maya.

**Batch Simulation**

To create cache data in batch mode, qlBatchRun script can be used.

*Usage*: `maya -file <filename> -batch -command "qlBatchRun <start_frame> <end_frame>"`

qlBatchRun will compute all the cloth objects in the scene between the start/end frames given as parameters.

**Multi-threading Control**

Click “Qualoth>Global Settings”, then the following dialog appears.

At the performance option tab, you can choose the number of threads to be used for simulation. By default, this number is set to the number of physical cores of the CPUs in the system. Setting this value above the physical cores exhibits performance loss than gain. (In hyper-threaded systems, halve the number of CPUs in the system monitor to get the number of physical cores.)