

StrandTree v1.0 - Quick Guide

developed by Fabricio Chamon

fabricio.chamon@gmail.com



Not all the compounds will be covered in this guide. Only the most complex to understand or not self-explanatory ones, also I'll try to compile some tips on how to construct and optimize a tree.

Emit Branches Compound

Grow Vector - vector per point parameter that defines the direction of the strand. This will create a straight line from the particle position defining a branch, where you can later displace with the turbulize strands compound or any other array modification nodes.

Branch Width/ Branch Width Falloff Percentage - changes to these parameters will only affect rendering of the strands.

Important: Every modification to strands, like turbulize strands or bend strands should be plugged in the **execute on emit** ports, this will ensure the branches are always connected to each other.

Grow Control

Even Distribution Grow - creates equally sized and rotated vectors at each branch iteration. This causes the tree to have a more uniform branch flow.

Random Distribution Grow - this is almost a randomize around value node only, with base vector pointing at positive Y.

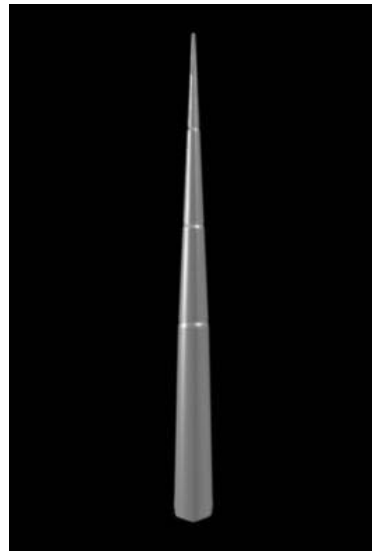
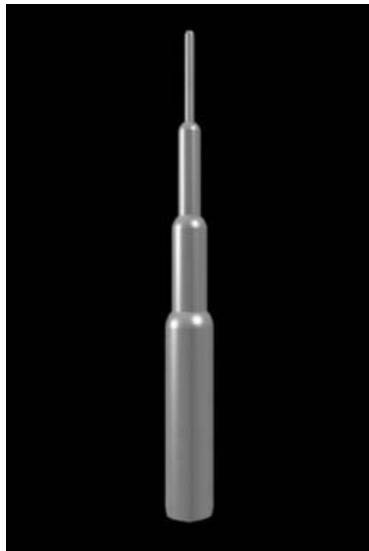
The cool thing is you can blend these two distribution types using the blend node, leading to a pleasing uniform tree but slightly randomized. Just experiment with these vectors to see a range of different results you can get from it.

note: The difference between Even and Random distributions are more easily understandable if you set the "Branch Grow Start" parameter = 1 (on the "Emit Branches" compound).

Modifiers

While "Modify Branch Color by Iteration", "Modify Branch Width by Iteration" and "Modify Particle Size by Iteration" are self explanatory compounds, the "Modify Value by Iteration" is a multi-use compound where you can control any value with a curve falloff going from first to last iteration. This is useful, for example, for modulating turbulence scale/strength on a turbulize strands node, or bend amount on bend strands towards vector node. This way you can get less effect on the trunk and more near the leaves which looks more natural on a tree. Note: these four compounds work on the *value per point* context.

Modify Value by Normalized Strand Segment - this is also a multi-use modifier just like "Modify Value by Iteration", but instead of working on a *value per point* context, it works on *array of value per point*. This means that if you want to modulate point attributes or a whole strand at a time (same value for the entire strand array) you can use "Modify Value by Iteration", but if you want to modulate strand array elements individually you can use "Modify Value by Normalized Strand Segment", which will give you a value of 0 for the first strand point on the trunk, going all way to 1 on the last points of the last strands of the last iteration. Here are two example images:



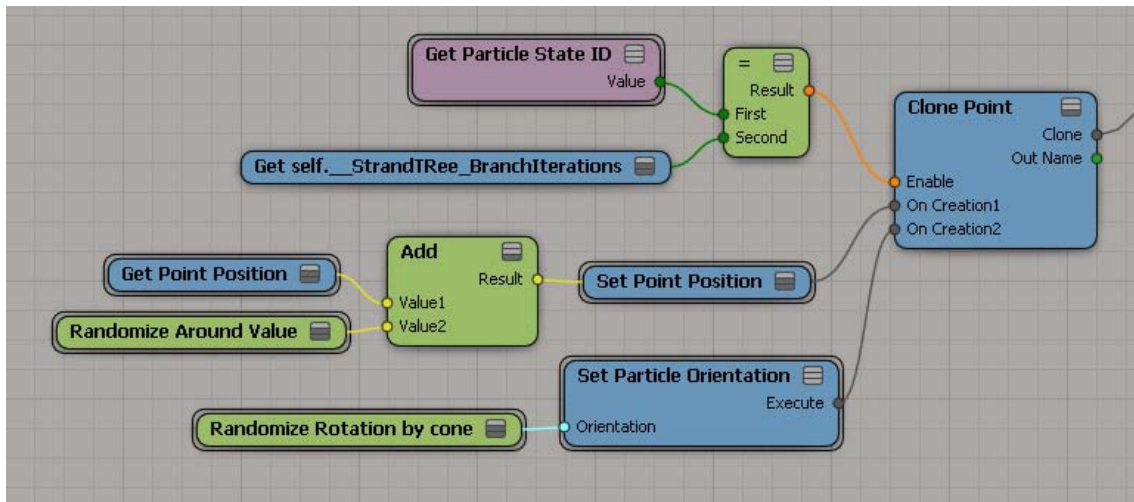
The left image shows a 4 iterations tree with strandsize modulated by "Modify Value by Iteration", and the right image shows the same tree but with strandsize modulated by "Modify Value by Normalized Strand Segment".

Emitting Leaves

Align to camera - this parameter allow you to align the particles to the camera specified in Alignment camera name. Sometimes just a little bit of particle alignment will give you more leaf volume in the tree.

If you want your tree to have more leaves than what you got from the Emit Leaves compound, but you don't want to increase the iterations, you can clone the leaves and do a randomize rotation by clone, and maybe a slightly change in the original leaf position (not too much or it will disconnect from the branch).

To achieve this we should use a clone node, with enable parameter set to a condition where `state_id = hidden strandtree` variable that returns the total number of iterations - explained on the next topic - Like in the image below:



Things to know

- Each branch particle has the STATE_ID attribute set to its iteration number, starting at 0 (the main trunk), and ending at [num of iterations - 1] for the last branches. You can use this attribute in the ICE tree to filter effects or in the render tree for color correcting individual branch groups or switching textures.
- The Pine Tree preset is a good example of optimization, where we deal with the problem of having to emit lots of branches growing only from the trunk, but not getting a super dense final particle count. Be sure to dig inside the compound and read the comments.
- You can access some hidden variables to use whenever you need, that are: "self.__strandTree_BranchIterations", " self.__strandTree_NumBranchesPerIteration" (both integer per object) and " self.__StrandTree_TreeNormalizedSegment" (array of integer per point - defining a normalized strand ratio that is used by the Modify Value by Normalized Strand Segment compound).
- The "Deform Tree by Curve" strand effect is a prototype compound, and it's not 100% working yet. For now it just displaces the first iteration (tree trunk).
- A nice and easy way to create tree roots is to build a branch only tree then rotate the point cloud so that it grows from top to bottom.