

Procedural Animation: Elevator Conveyor Belt

Houdini v. 19.5.569

Renderer: Redshift

Average Render Time: 1 min/per frame

Resolution: 1280 x 720

Number of Lights: 2 (Environment Light & Sun Light)

Bucket Quality: Medium

Threshold: 0.01

Progressive Passes: 1024

Global illumination: 4

Reflection: 4

Refraction: 6

Volume: 1

Combined: 6

Transparency: 16

Project Description

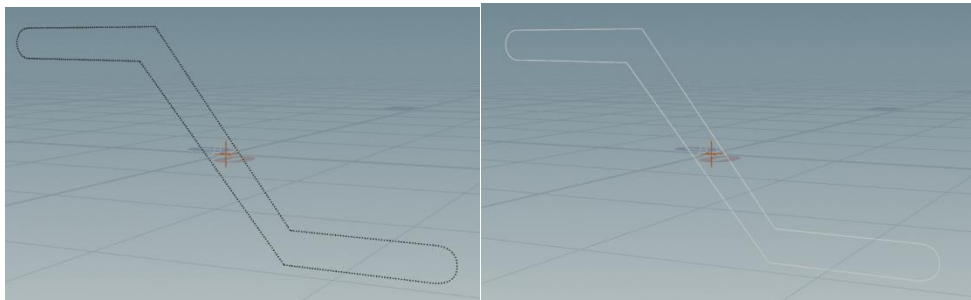
My project is about recreating an automatic elevator conveyor belt and a carton that can never be transported up. This project was inspired by an installation art work named "Futile Ascent" by Bao Rong.



Breakdown

1. Xyzdist

For the animation of conveyor belt, I mainly used *xyzdist* and *primuv* function. Firstly, I prepared two objects, one is consisted of only points, one is a line with normal.



```
VExpression
1 vector a = set(0,0,0);
2 @N = normalize(a+@P);
```

(Assign each points' normal)

Then I used *primuv* and *xyzdist* function to control the points movement. *Primuv* function can return to the position on the geometry according to the *prim* and *uv* attribute. *U* and *V* value always from 0-1. *Xyzdist* function can change the *prim* and *uv* attribute, and back to the distance which from current position to the third geometry's position.

```
VExpression
1 vector primuv;
2 float distance = xyzdist(3,@P,0,primuv);
3 f@u = primuv.x;
```

u
1.0
0.00225223
0.0045134
0.00684192
0.00860839
0.0111242
0.0135038
0.0155032
0.017504
0.0198534
0.0222109
0.0243524

(Primuv and Xyzdist function)

Then I assigned each points' *u* attribute a new value, using the offset channel to control the points' movement.

```

VExpression
1 float updated_u = (f@u + chf("offset"))% 1.0;
2 @P = primuv(3,"P",0,updated_u);

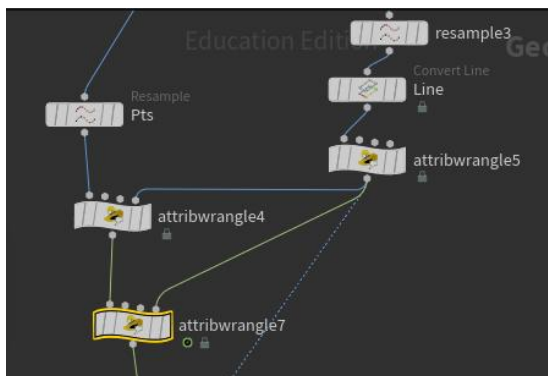
```

Attributes to Create *

Enforce Prototypes

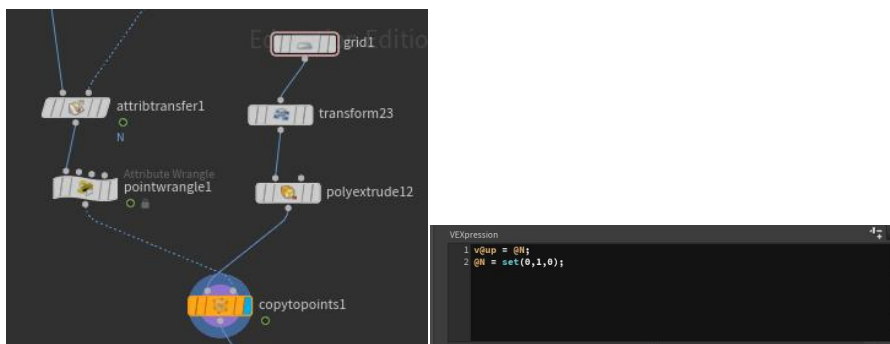
Offset 0

(Animation Controller)



(Node network)

For the shape of conveyor, I used copy to points, transfer and change the N attribute to make sure each point has a right normal direction.

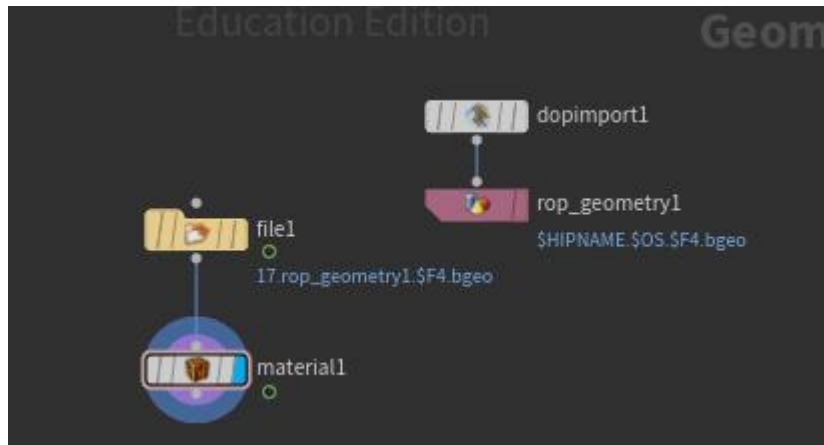


2. Simulation RBD

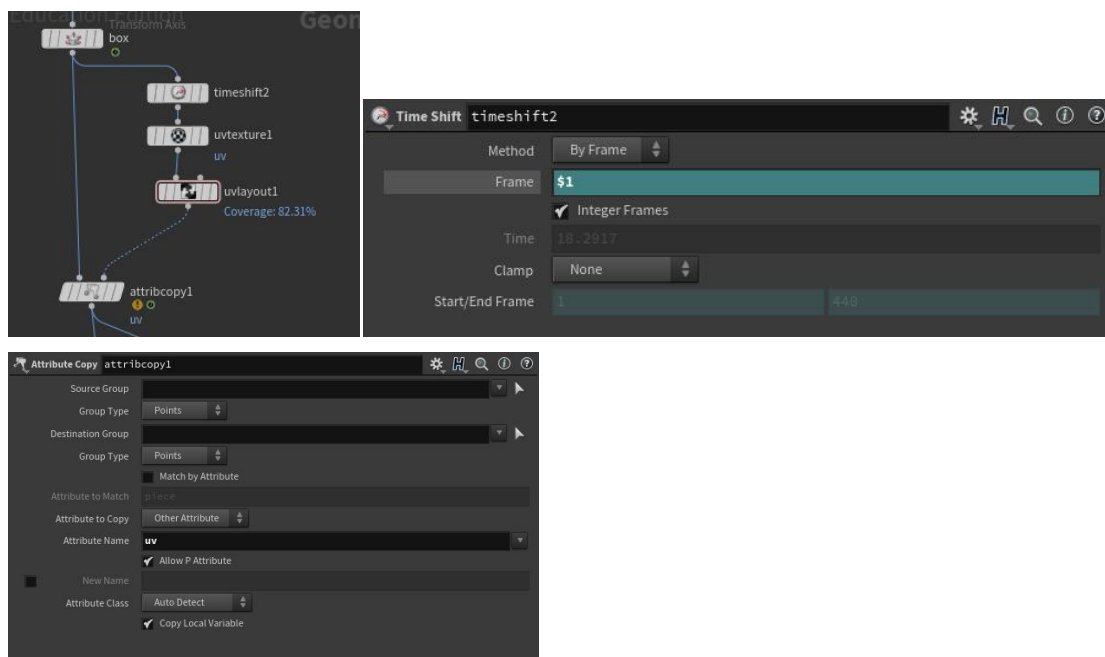
To create the animation of the rotating carton influence by the conveyor, I used RBD, which can help the carton's animation more natural and physically-based.

I set the box as my RBD objects and then belt geometry as my collision

objects, and mainly modified the velocity, gravity, bounce and friction parameters. I also imported the dop network into geometry network and did the cache work.



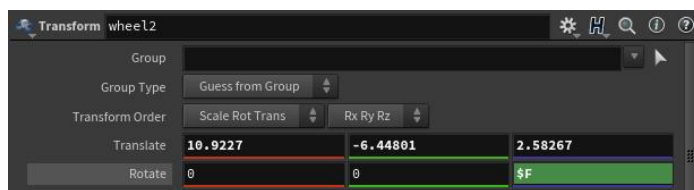
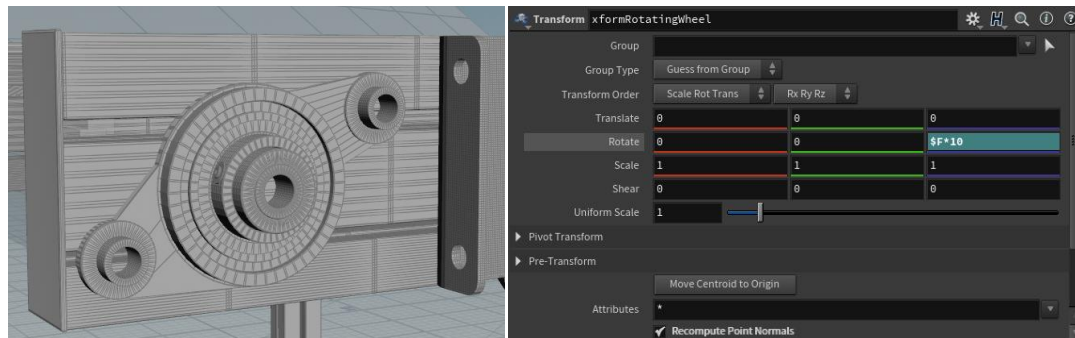
The biggest issue during using RBD I met was the uv problem. The uv will change every frame and took longer time to simulate. To solve the problem, I used timeshift and attribcopy nodes.



3. Modeling & Simple animation

There are many hard surface modeling part in this scene, which mainly

created by *bevel*, *resample*, and *remesh* to grid node. For the tiny wheel's rotating animation, I mainly used $\$F$ and $\$F * 10$.



4. Box texture

I used substance painter to create simple carton's texture maps. (For optimization I did not use height map in the final output version.)

