Cooking Southeast Asia-inspired Soup in Animated film

Cong Wang, Walt Disney Animation Studios, USA

Dale Mayeda, Walt Disney Animation Studios, USA

Jacob Rice, Walt Disney Animation Studios, USA

Thom Wickes, Walt Disney Animation Studios, USA

Benjamin Huang, Walt Disney Animation Studios, USA

Figure 1: Rendered soup under three different camera angle

ABSTRACT

Walt Disney Animation Studios' "Raya and the Last Dragon" is an animated film inspired by the people and culture of Southeast Asia. In "Raya and the Last Dragon", we created a soup inspired by Thailand's Tom Yum soup to help represent the food cultural richness of that region. Our goal was a more believable and authentic representation of food than we have previously achieved, which required a novel approach, especially on how to simulate the motion of the chili oil on top of the soup and multiple representative ingredients. This talk will describe the collaborative process of designing, simulating and creating materials to achieve the final look of the soup. Figure 1 shows three renders under different camera angles.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**.

KEYWORDS

fx, effects, fluid, flip, houdini

ACM Reference Format:

Cong Wang, Dale Mayeda, Jacob Rice, Thom Wickes, and Benjamin Huang. 2021. Cooking Southeast Asia-inspired Soup in Animated film. In *SIGGRAPH '21: Aug 01–05, 2021.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450623.3464651

1 DESIGN AND INSPIRATION

The initial soup design is inspired by Tom Yum soup which is a type of hot and sour Thai soup usually cooked with shrimp, lemon grass, bamboo shoots, red chili peppers. It is a great reference that helped us capture the key features of dishes found throughout the countries of Southeast Asia.

The soup should appear mostly opaque from afar with yellow/red broth and speckles of green and red. In the soup we can see basil, cilantro, sliced bird chilies, sliced young bamboo shoots, a few bits of chopped lemongrass and some shallot pieces. Some Chili oil with saturated color should be floating on top of the surface and having foam surrounding and mixing with the oil.

2 SIMULATION AND POST-PROCESS

The soup simulation has been executed in three steps: 1. A low resolution simulation driven by a boiling force. This sim was used as an advecting force in the following steps. 2. Simulating heavier ingredients such as bamboo shoots and shrimp paste. 3. Simulating high resolution soup using Houdini Flip solver with a popfluid solver to get the motion of the soup base, chili oil and foam.

2.1 Soup base and heavy ingredients simulation

We first simulated a low resolution soup with a customized boiling force to get the basic boiling motion. The "hot spots" where the water has been pushed up more are mostly stationary with a small amount of drifting driven by noise. At this stage there is no oil and foam simulated yet. This low res simulation is used for advecting the other element's motion.

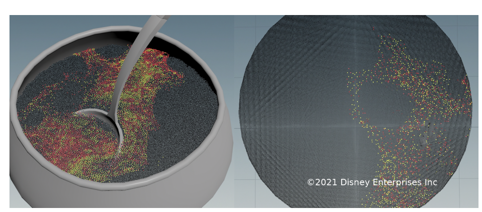

Figure 2: Soup FLIP simulation. Oil in red and foam in yellow

SIGGRAPH, Aug 01–05, 2021, Virtual

ACM ISBN 978-1-4503-8373-8/18/06.

https://doi.org/10.1145/3450623.3464651

Figure 3: Our meshing method(L) compare to conventional method(R)

The second step is to simulate the heavier ingredients like bamboo shoots. We want the bamboo shoots moving up and down while floating, but remaining at a certain depth range without emerging above the surface. We find it is very difficult to get the desired result by just using conventional rigid body simulation. So we added a customized buoyancy force to the bamboo shoots object. We used a simple ray casting technique to loosely constrain the bamboo shoots to the fluid surface. Furthermore at each frame, we would rotate our bamboo shoots transform along the first principal axis, to a align its third principal axis towards the world's up direction over time. This rotational force keeps the bamboo shoots facing up and prevents them from coming to a rest along the edge of the shoot, as opposed to the face of the shoot

Lastly we did a high resolution flip simulation for the final soup. At this step we grouped some points on the top portion of the soup as oil and foam based on a noise pattern, added a popfluid solver to control the force and other physical attributes within the grouped points. We gave the oil and foam points much smaller density values so they can stay on top of the soup. Combining these two steps we got a group of flip particles constrained together to achieve a realistic oil/foam tearing and re-merging motion. Figure 2 shows the screen shot of the simulated flip particles.

2.2 Bubbles and particulates simulation

We used the high res soup simulation to advect the bubbles' simulation. After several early tests, we found the attempt of using advection alone to get the bubbles motion always ended up with having some bubbles "punching" through the floating oil, which is not a realistic behavior. So we added a controller inside the pop simulation to force the bubbles to attach to the oil/foam points if this a distance threshold between the bubble and nearest oil/foam point has been reached, then the bubbles started to move alone with oil/foam afterwards until they burst.

To enhance the realism further and adding a tad bit of Disney Animation art direction to this pure realism, we give each bubble a two frame burst animation to caricature the popping bubbles.

2.3 Meshing the oil and foam

Properly meshing the oil/foam points is challenging but important to achieve the realistic look.

The first issue we met was the oil points were not forming a clear and thin layer even though they were generally floating on top of the soup surface. We had to compress them down, post simulation, to get a thin layer perfectly sitting and matching the upper soup mesh surface. Secondly, using the regular meshing method with the particle fluid surface node from Houdini resulted distracting popping as well as other visual artifacts. To address this problem, we instead instanced thin and round plates to each points, scaled them based on the density of the point cloud surrounding, combined with the speed of the points itself. The instanced plates were then converted into VDB followed by several post processing steps. This method generated a round and flat bubbly shape, which matches the round shape of the real oil due to surface tension when floating above water. Figure 3 shows a comparison of the result of using our meshing method and regular flip meshing method.

2.4 Motion of light weight ingredients

We used a subset of the simulated oil points to drive the motion of the light weight ingredients such as basil leaves, lemon grass and bird chilies. The trace of these points within the shot frame range has been converted into spline curves, later the position of the points on each frames can be restored from the curves. This method offers artists more flexibility to tweak the motion as needed even after simulation by simply modifying the shape of the curve. We also used this method to avoid any interpenetration between elements.

3 MATERIALS

We broke down the oil material into three main aspects: shape, color and specular responses. There are three different triplanar maps to define the shape and pattern of the the oil on top of the shape generated by meshing, control the color of the oil as well as displacement and specular. Figure 4 are the three noise patterns be used.

Both the soup base and foam are rendered with a combination of geometry and volume using Disney's Hyperion Renderer[Burley et al. 2018]. We also added stain on the inner surface of the pot and a wet map to the ingredients to further enhance the integration.

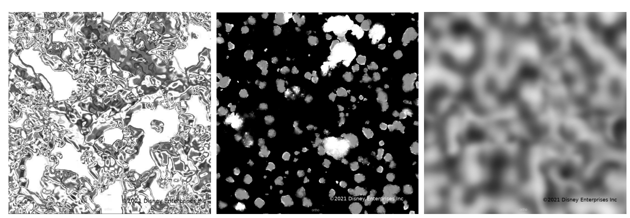

Figure 4: Three noise patterns used for oil material

4 CONCLUSIONS AND FUTURE WORK

In the current state, some repetitive post processing on the simulated oil and foam are required in order to get the thin and clear oil layer. It could be improved in future work so the oil points could form a thin layer inside the simulation without the need of any post process. This will help to automate the workflow even further.

REFERENCES

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Trans. Graph.* 37, 3, Article 33 (July 2018), 22 pages. https://doi.org/10.1145/3182159