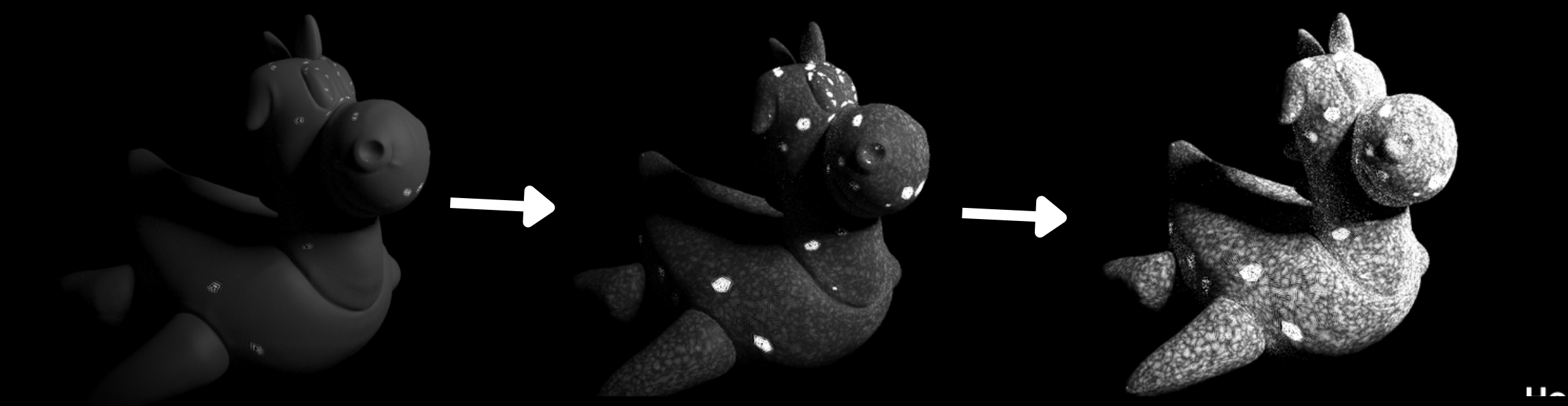
Oriol Manzano
@oriolmanzano.com

How to create patterns with the Heat equation in Vex language



Oriol Manzano
@oriolmanzano.com

1-Initialization

Assign an initial temperature value to each point in your geometry. This could be uniform or vary across the geometry.

oriolmanzano.com vex

```vex
// Parameters
 float alpha = 1; // Thermal diffusivity, adjust based on
your needs
 float dt = 7.0; // Time step, adjust based on your
simulation requirements
```

Oriol Manzano
@oriolmanzano.com

2-Time Stepping

Implement a loop to simulate time progression. Each iteration of the loop represents a small step in time. Alternativelly, you can use a SOP Solver.

oriolmanzano.com vex

```vex
// Iterate over all frames
 for (int frame = 0;  frame < max_frames; ++frame) {
// Heat Equation Code
}
```

Oriol Manzano
@oriolmanzano.com

3-Calculate average temperature

Average the difference between the temperature of a point and its immediate neighbors to be able to later compute the Laplacian

oriolmanzano.com vex

```vex
int neighbors[] = neighbours(0, pt);
int numNeighbors = len(neighbors);
float tempSum = 0.0;
// Iterate over the neighbors of the current point
for (int i = 0; i < numNeighbors; ++i) {
    tempSum += point(0, "temp",  neighbors[i]);
}
```


Oriol Manzano
@oriolmanzano.com

4-Compute Laplacian

For each point, compute the Laplacian, which in this context approximates the second spatial derivative. This usually involves calculating the difference between the temperature of a point and its immediate neighbors average. This average temperature has been calculated in the previous step 3.

oriolmanzano.com vex

```
float laplacian = 0.0;
float currentTemp = 0;

if (numNeighbors > 0) {
    float avgTemp = tempSum / numNeighbors;
    currentTemp = point(0, "temp", pt);
    laplacian = avgTemp - currentTemp;
}
```


Oriol Manzano
@oriolmanzano.com

5-Update Temperatures

Use the computed Laplacian to update the temperature of each point based on the Heat Equation. The new temperature can be calculated using a discrete approximation like:

$u_{new} = u_{old} + \Delta t \cdot \alpha \cdot \text{Laplacian}$

where Δt is the timestep

oriolmanzano.com vex

```
  // Update the temperature of the current point
  float newTemp = currentTemp + dt * alpha *
laplacian;
  setpointattrib(0, "temp", pt, newTemp, "set");
```