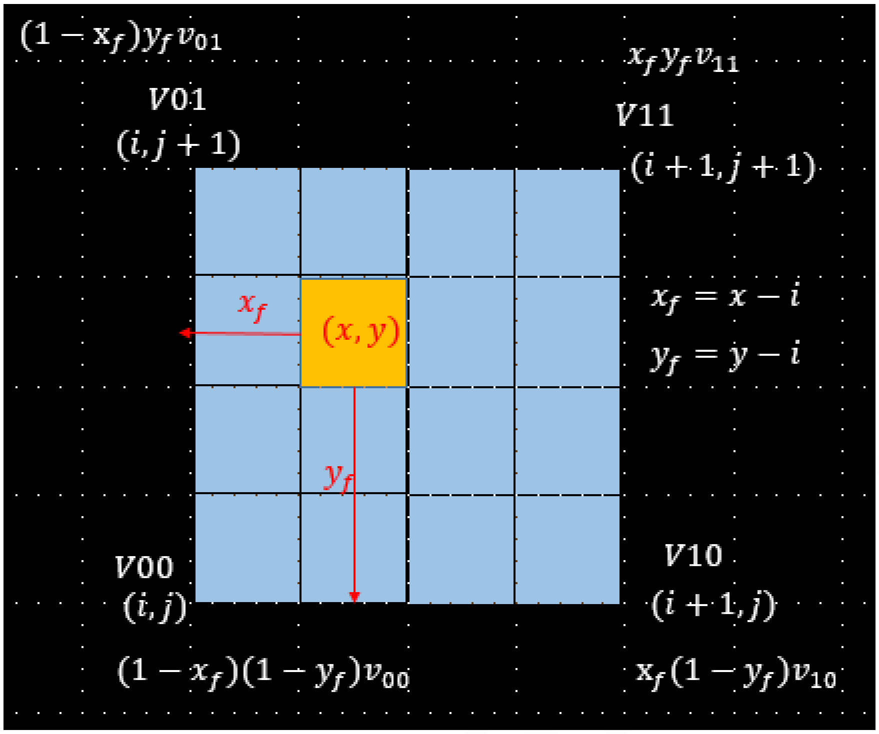
2025-10-25

Reading code directly is confusing, because the notations are horrible. After a week, I would like to go through step1-3 again. This time I asked for explanation of the logics using algebra and graph. So the linear interpolation is pretty straightforward:



And the bilinear formula:

$$n(x, y) = (1 - x_f)(1 - y_f)\, v_{00} + x_f(1 - y_f)\, v_{10} + (1 - x_f)y_f\, v_{01} + x_f y_f\, v_{11}.$$

Although the model is intuitive: the contribution from each corner pixel is related to the distance: the further away the target is from a corner, the smaller that corner's contribution should be. But we should ask:

1. Why this particular model for valuing the contribution from a particular corner?
2. When it comes to math modeling rather than solving a differential equation, it's more empirical than theoretical, more flexibilities. As long as it 'feels right', you can go with whatever model you want. And of course, then the results must meet your expectation. So throughout many math experiences, I want to know what's the general rationale in picking one model? Save this question for later.

Step3: gradient noise

The notations are terrible, but once you view them graphically, it's not that hard. The notations for offset vector is terrible. But the idea is to measure how far the target pixel is from each individual corner, it's very similar to those $x_f, x_f - 1, y_f, y_f - 1$, except the sign is flipped. I am not very bothered now. So the distance from the pixel to each corner is kept as the offset vector. You should ask why, because if you want to know the distance, sure you use $\sqrt{(x - x_i)^2 + (y - y_i)^2}$, why you store it as a vector? Of course it is related to what how we want to use its distance. We want to know how its location is compared to the direction of the 'wind-blow'. This is the genius part, also the abstract part: because you can easily ask, where is the wind-blow?

This is the typical explanation (as GPT gives):

1. You can explain that the gradient vector (1,0) is an example of wind blowing to the right. (0,1) is an example of wind blowing up.
2. Then you can explain the doing the dot product is calculating directionally we are in common

But the biggest problem with the above explanation is:

1. It's hard to connect two numbers (1,0) to a direction. Yes I know you place a dot at location (1,0), then draw an arrow from the origin, that's the direction. **But the problem is: this is what you say so! It is your definition! How would I know it makes sense or not???** All I am seeing is just two numbers, I don't see an arrow! **So we need to understand, how to define math concepts, and when it makes sense, when it might not.** *I would say this is one conceptual leap for vectors.*

Say we understand the vector notation, the meaning of dot product. This influence number tells us how much in line those two directions are. The more they are in line, the more influence they have. Does it make sense? Yes, if you believe that the gradient vector is the direction of wind-blowing, then it will blow wind along that direction, hitting a certain pixel. But does not mean nobody else get the blow? Perhaps not? Everyone gets some blow, proportional to how inline you are, unless you are totally 'tangential'. The dot product is capturing that in-line portion. So the results make sense: 0 if parallel, positive in line, negative if opposite. So far I don't know the quantitative scale of the influence, but we can zoom in later.

So after this step, we know each corner's contribution depending how the pixel is in-line with the wind-blowing direction. Next is interpolation.

What is d00, d10??? Oh, the influence number! That pixel number determines the contribution from each corner, but it only factors in the 'directional alignment', not distance. Now the weights are dependent on the location. Of course, the further away you are, even though directional it's in line with the wind, yet it's too far to be relevant. So the next interpolation step does that, but with the distance 'processed'.