Video: Computing the Velocity Field

[00:01] [slide 1] In this segment of the course, we're exploring how we can compute the motion of features in the changing image. In particular, how can we compute a velocity field that assigns a direction and speed of movement to regions in the image? In the last video, you learned about the aperture problem - if we initially detect motion by analyzing changes taking place over time in small regions of the image, we can only directly sense the component of motion in the direction perpendicular to moving edges. To resolve the two-dimensional velocity at each location, we need to integrate information from multiple motion components over an extended region. And in the process, we need to incorporate additional constraints that will enable us to compute a unique pattern of motion that's consistent with the changing image.

[00:59] The last video elaborated on how we can compute the motion components, by combining measurements of how intensity is changing in the image and over time. We showed how we can compute a unit vector in the direction of the gradient of image intensity, which indicates the direction of the motion component. This vector is denoted here with ux and uy. We also showed how we can compute the velocity of the edge along the direction of the gradient - this is the quantity v-perpendicular here. The velocity that we ultimately want to compute is denoted here as Vx and Vy. At each location where we can measure the gradient and perpendicular motion, we can write an equation like the ones in the blue box here, that captures the possible velocities Vx and Vy that are consistent with these measurements. Given two motion components in two different directions, we can solve for Vx and Vy - we can find one velocity that's consistent with both measurements.

[02:01] [slide 2] In practice, there's likely to be error in the perpendicular motions that are computed from the image, so the constraint lines from any pair of measurements could intersect at a different velocities. So rather than just consider only two measurements and see where the lines intersect, we combine lots of measurements within an extended area of the image, and find a single velocity that best fits all the measurements simultaneously. Previously we said that the calculation on the left of each of these equations at the top here, needs to be exactly equal to the perpendicular motion that we measured on the right of this equation. Now we're going to say that we want the difference between the two sides of this equation to be as small as possible. This is expressed at the bottom here, where we take the difference between the two sides of these equations at the top, square it, and sum up these differences over all the motion components. We want to compute a single value for Vx and Vy that minimizes this sum - a velocity like the one shown with the black arrow in the diagram, this is as close as possible to all the constraint lines simultaneously. This is a general computational method known as least squares fitting. We compute parameters that best fit a set of data that captures constraints on those parameters. This is the same process that's used for performing linear regression - fitting a line to a set of data points. We solve problems like this by taking the derivative of this expression with respect to each of the parameters Vx and Vy, and setting the derivatives to zero. This gives the velocity that best fits the data by minimizing this sum. In the next assignment, you'll see what the solution looks like in this case.

[04:10] This approach assumes that regions of the image just undergo pure translation across the image, so there's only one velocity to compute for the region over which these measurements are being combined. I'd like to briefly introduce a more general constraint that will allow us to preserve the variations of motion that may exist from one location to the next across the image. It's a common approach used in many computer vision applications, and it's a form of the continuity constraint that we described for stereo matching.

[04:48] [slide 3] The basic idea is that we still want to compute a velocity field that's consistent with the local motion components, but rather than say the velocities in a region are all the same, we'll say we want them to vary as little as possible. Neighboring velocities could be different, but we want to minimize the change in velocity from one location to the next. This strategy encompasses the case of pure translation. Suppose we have a figure like this ellipse, that's translating down and to the right, like the little segments around the contour on the left indicate. On the right are the perpendicular components of motion at each point around the contour, that you could measure from the image. If these measurements are consistent with a pure translation of the object, then this strategy will find that solution - there's no variation in the computed velocity field in this case, so it certainly must be the solution with the least amount of variation possible, given the image measurements.

[05:57] [slide 4] How do we formally express the computation of a velocity field that varies the least? First, each location of the image can have a different velocity. If we imagine computing velocity vectors along a contour, there's a different velocity Vx and Vy at each location on the contour, so an index i is added to Vx and Vy to denote the velocity at a particular location i on the contour. And at each location, there's a motion component that constrains the velocity that you could assign at that location, and we again try to find velocities that best fit the motion components. What's different here is that we also try to minimize the change in velocity from one location to the next. Here, that change is expressed as the vector difference between the velocity at some location i and the velocity at its neighbor i+1. We can add up all the changes in velocity over an extended contour. In the end, we compute a set of velocity vectors along the contour that minimize the combination of fit to the motion components and amount of variation in velocity over the contour. There's an extra parameter lambda in the middle that just weighs the relative importance of fitting the data versus smoothness of the velocity field. And there are standard methods for solving problems like this - here I'd really just like you to give you a sense of the basic concept. It's an approach that comes up a lot in visual processing, and sometimes goes under the name of "regularization". Does this have any relevance to human motion perception?

[08:00] [slide 5] To address this question and also get an intuition for situations where you really need a more general assumption like this, let's look at some examples of velocity fields computed using this method. Suppose you have an object like this polygon in the upper left corner, that's rotating in the image around its center. The left picture shows the true velocities at each point around the object, and you can see, they differ from one location to the next, because of the rotation. To the right are the perpendicular components of motion that can be measured from the changing image. It's not possible to compute a single velocity that's consistent with all these measurements, but you can compute a pattern of velocity that has the least amount of variation around the object, and that in this case, is the correct pattern of motion using this strategy. On the bottom is a similar example, but this time, it's the two-dimensional projection of a three-dimensional wire-frame object that's rotating around a central vertical axis. All the true image motions in this case are horizontal in this case, but the speed of motion varies along the edges of the figure. There would again be no single velocity that's consistent with these motion components, which are shown in the middle here, but the smoothest velocity field you could compute from these components is the correct one.

[09:44] There are situations where this strategy would give an incorrect solution, but one that's consistent with human motion perception. One example is the barber pole illusion, where the stripes of the barber pole are really moving horizontally in the image, but the stripes look like they're moving downward. In this case, the movement happens to be consistent with a downward translation of the stripes, so actually any method for computing the velocity field that assumes pure translation would predict this result. In the case of a rotating spiral, as you see here, the two spirals in this pattern look like they're either expanding outward or contracting inward, but that's just an illusion - the pattern is really rotating around its center, so all the points on the contours are following a circular path and not really moving inward or outward. This percept is consistent with the smoothest pattern of motion that could be computed for this pattern. Finally, if you spin an egg around its center and observe its motion as it rotates, it'll look like the long and short axes of the egg are wobbling in and out, and the egg looks nonrigid rather than looking like a rigid object rotating around its center. This model also predicts this percept of nonrigid motion. And there are many other examples like this, of smooth contours in rotation, where you get an incorrect pattern of motion if you just find the smoothest velocity that's consistent with the changing image, but we also perceive an incorrect pattern of motion. The take-home message here is that there are other constraints that we can use to compute a velocity field, including something very general like this, that enable us to capture the variations in velocity that can be important for tasks like recovering three-dimensional shape and analyzing nonrigid motions like we see in facial expressions or wiggling jello.

[12:05] [slide 6] The last thing I'd like to touch on is an aspect of the neural processing of motion information, and in particular, ask if there's evidence of neurons analyzing image motion in two stages where they first measure only the components of motion perpendicular to moving edges, because of the aperture problem, and later integrate these components to determine the two-dimensional motion of features in the image. Movshon and colleagues explored this question in monkeys, and in their study they recorded the activity of neurons in two areas, V1 that you learned about earlier, and an area known as MT, which stands for Middle Temporal area. Neurons in area V1 project to neurons in MT, and in V1, complex cells are often selective for the direction of motion - if you move a bar at their preferred orientation back and forth across their receptive field, they might respond to one direction of motion but not the other. In MT, most neurons are selective for the direction and speed of motion, and if you damage MT, there's a loss of visual abilities that depend on motion information, like the ability to track moving objects with the eyes.

[13:29] [slide 7] These experiments used visual stimuli that consist of superimposed sine-wave gratings of the sort you saw in the lecture notes by Michael Landy on spatial frequency channels in human vision. I'll go to an online demonstration to see what these plaids look like. On the left are two individual sine-wave gratings, each with a single orientation moving in a particular direction, and we only sense the motion perpendicular to their orientation here. If you superimpose those two patterns at each moment, you create something like what's shown in the upper right corner, which looks like a single rigid plaid pattern moving down and to the left. Movshon and his colleagues recorded the responses of neurons to moving plaids with different component gratings.

[14:40] [slide 8] This slide captures the basic logic. Imagine you're recording from a neuron engaged in the first step of motion measurement - measuring just the perpendicular components of motion, and suppose in particular, that you're observing a neuron that responds to vertically oriented image features moving to the right. Along the top here are three sample stimuli, just shown in a diagrammatic way. The first two have a vertical component embedded in the image that's moving to the right, but because of the addition of a horizontal pattern moving up or down, the overall motion of the combined pattern is in an oblique direction, upward or downward, as shown with the red arrows. The third example has components that are oriented obliquely, but when combined, the overall pattern is moving to the right.

[15:42] So now let's look at the signature response pattern that we'd expect for neurons engaged in the two stages of motion measurement. If a particular neuron only measures motion components, and let's say it prefers a vertically oriented component moving to the right. It should respond to the first two patterns here that have this component embedded. On the other hand, such a neuron would not respond to the last example, because the last example has an overall motion to the right, but doesn't have a vertical component embedded inside. Now what if you were recording from a neuron that integrates motion components together to resolve the real direction of motion in two dimensions. And suppose this neuron prefers an overall pattern motion to the right. This type of neuron would not respond in the first two cases, because their overall motion is in an oblique direction. But it would respond to the last pattern whose overall motion is to the right.

[16:58] [slide 9] The researchers discovered that in the earlier stage of processing in area V1, all the neurons behave like component cells. Visual cortex is a layered structure with 6 primary layers. Inputs from other areas of the brain enter a particular area in certain layers like layers 4 and 6, and information is later processed by neurons in other layers, like 2, 3, and 5, and then passed on to other areas of the brain. They discovered that in area MT, neurons closer to the input layer behave like component cells and in the later stages of processing you find neurons that behave like pattern cells. So there’s direct evidence for a two-stage motion measurement process in the brain of monkeys.

And finally, just a quick note that not all components are alike - perceptually we don’t combine them when their spatial frequency is very different, or they move with very different speeds, or the components have very different stereo disparity and are perceived to be located at different depths. These observations provide some hints about how the motion integration might take place at different scales or at different depths.