Video: Solving the Observer Motion Problem

[00:01] [slide 1] The last video introduced the observer motion problem. When we move relative to the environment, we may be translating through space in the horizontal, vertical, or forward directions, and we may be rotating our eyes around the horizontal, vertical, or forward axes. These motions of the observer are captured by six parameters - Tx, Ty, Tz are the parameters of translation, and Rx, Ry, Rz are the parameters of rotation. We also typically move toward a scene containing surfaces at different depths, and we'll refer to the distance to the nearest surface at each image location as Z(x,y). The observer motion problem refers to the task of recovering this information about the motion of the observer and 3D layout of the scene from the motions that we compute from the changing two-dimensional image.

[01:00] [slide 2] If we were just translating relative to the scene, we'd generate an expanding pattern of motion that emanates outward from a focus of expansion at our heading point, and we could easily locate our heading point by seeing where the directions of image motion intersect. That situation is shown in the picture in the upper left corner. The speeds of motion in this picture depend on the depth of the surfaces being viewed, and in this example, the observer is moving toward a wall that has a square object floating out in front. The dashed red lines show the outline of this square object, and you can see that in the vicinity of the border of the square, where we have a jump in depth, there's a change in the speed of motion across the border. If we stay very still and just rotate our eyes, then everything in the image just shifts in a direction opposite the direction we're rotating the eyes, as shown in the picture on the right, where the eyes are rotating to the left. When we translate and rotate at the same time, which is what we do most of the time, we generate a more complex pattern of image motions, as shown at the bottom, and it's more challenging to recover our direction of motion in this case. But we know from perceptual studies that we can determine our heading very accurately in this general situation.

[02:32] [slide 3] In the last video, we also showed the equations that capture how the image velocity at each location, denoted by Vx and Vy on the left here, depends on the observer's motion parameters and depth. The strategy that we'll describe for solving the observer motion problem builds on a key property of these equations, which is that only the translational component of image velocity depends on depth. The image motions due to the observer's rotation don't depend on depth, they only depend on image position, and they're fairly uniform across the image, as you saw in the previous slide.

[03:14] [slide 4] So how will we take advantage of the way image velocities depend on depth? We're going to use a suggestion that was first made by Longuet-Higgins and Prazdny. On the left here I show the same velocity field that you saw earlier that combines a rotation of the observer with translation toward a wall with the square object in front. If you look closely at the image velocities in the vicinity of the borders of the square object, you'll notice that there's a change in velocity across the border. For example, consider the two velocity vectors in the blue oval here. The velocity vector just inside the border is different from the velocity vector that's just outside the border. I drew the two vectors enlarged on the right here, at a common origin, so you can see the difference between them more clearly. Suppose we subtract these two vectors - let's take the blue vector and subtract the green vector. Geometrically we start with the blue vector and add a negative version of the green vector, and the difference is this red vector. It's the same as the vector between the endpoints of the blue and green vectors. The red vector is vertical, and if we translate it back to the border and shrink it back in size, we see that it lies along a line from the border point to the focus of expansion or heading point in the middle. This is true in general. Along a depth discontinuity like we have at the border here, the velocity differences that we get by subtracting velocities on the two sides of the border - these differences only depend on observer translation, and in particular, they point to the focus of expansion of the translational component of motion, which is our heading point. These differences only depend on the observer's translation because the motion due to the observer's rotation is essentially the same on both sides of the border, so when you subtract two velocities that are close to the border, the rotational motions just cancel each other out.

[05:36] On the right are all the velocity differences around the border of the square object. The differences between neighboring velocities were actually computed everywhere in the image, but these differences are large only at places where there's a change in depth. Here I took three sample difference vectors and drew these blue lines through them to see where their directions intersect, and they intersect at the location of the focus of expansion, or heading point. So one approach to computing the observer's heading direction is to measure the change in velocity everywhere in the image, and where there are large changes, record the directions of the lines containing these differences, and see where the lines intersect. When we compute image motions in real images, there will be errors, so here we'll compute an intersection point that best fits the lines, just like we did in the motion measurement case. And note that finding places where there are changes in velocity is also important for another reason - it helps us to find object boundaries in the scene, so it's something we want to do in any case.

[07:01] [slide 5] This analysis provides a way to get the direction of motion of the observer, but what about their rotation, and what about the depths of object surfaces in the scene? I'll just touch briefly on a couple ideas here. Consider four sample locations in the image, and let's say the true FOE is located at this black dot in the middle, in the left diagram. At each of the four locations I drew three vectors that separately show the velocity due to the observer's translation and rotation, in red and green, and the resulting velocity combining those two motions, in blue. The red vectors from the observer's translation all point away from the true FOE. The observer is rotating their eyes to the right, so the green rotation vectors all point to the left. The blue vectors showing the final velocity at each of these points, were generated by adding the red and green vectors.

[08:10] Suppose that an estimate of the location of the FOE was computed by analyzing all the image motions, as we described previously, and the black dot on the right here shows the estimated location of the FOE, which may be off. Let's say we now want to try to compute the parameters for the observer's rotation, and we'll see in a moment why this is useful information to know. Once we have an estimate of the FOE location, we know that at each location in the image, the translational component of motion should lie along a line from the FOE to that location. We refer to these lines as the translational field lines. Any movement that we observe in the direction perpendicular to these lines, like the purple components of motion here - these perpendicular motions must be due to the observer's rotation, because the translations would be along the translational field lines. And it's possible to compute a set of three rotation parameters that best explain the motions that we observe perpendicular to these field lines.

[09:42] [slide 6] Why is it helpful to know the observer's rotation? If we know the rotation parameters, we can reconstruct the motions at each location that would arise from that rotation, as shown here with the pure rotation diagram. And we can subtract those velocity components from the full velocity field, which will leave us with the velocities that result from the observer's translation alone. And from those motions, the speed of image motion at each location can be used to recover the depth of the surface in space that's viewed at that location. A caveat here is that we can only figure out the ratio between our speed of motion and depth, we can't pin down depth absolutely.

[10:37] There's an assumption that's lurking around here, in the formulation of the problem and ideas for its solution. We're assuming that the observer is the only thing moving, and that the object surfaces in the scene are stationary. The situation is different when objects in the scene are moving themselves. Finding places where velocity is changing is still very important. Around the borders of a moving object, there will be a difference in motion between the moving object and the background surface. But now, the velocity differences will not necessarily point to the focus of expansion, so the strategy needs to be modified to handle moving objects in the scene. The human visual system can sometimes be thrown off by the presence of moving objects - you've probably had experiences yourself, where you thought you were moving when you really weren't - it was the other objects around you that were moving.

[11:45] [cheetah clip] I'm going to end here with a video clip of a slow-motion film of a cheetah running, and I want you to observe how incredibly still the cheetah maintains its head - the real key to accurately recovering your heading in a high-speed scenario is to not rotate, which the cheetah has learned through its evolution … It's beautiful to watch cheetahs run, and you can see the link here: https://www.youtube.com/watch?v=B4nd9GF1dRg