CS 675 – Computer Vision – Fall 2007

Instructor: Marc Pomplun

Assignment #3

Sample Solutions

Question 1: Fourier Transform

Since 1D motion analysis results in a 2D image representation ($x \times t$), we can take the Fourier Transform of the image to get an idea of the global motion for the 1D image across t. However, if we are interested in the local motion of the image, we must determine first a localizing method (i.e., a way to partition the image). One very crude way would be to simply partition the image into an equally spaced grid. Then in each grid-space we could perform a Fourier Transform, making sure to first use the Butterworth Filter to hinder any effects from image edges. Each Fouriered grid space, an $m \times n$ array, now represents the local motion for its area of the image.

Let's arrange things so that the horizontal axis is the spatial dimension and the vertical axis is the temporal dimension, just as in the slides. We're assuming the motion has constant velocity. We will then see that there is a set of straight lines created by the motion, as shown in Slide 15. If we then take the Fourier transform of the array, these lines should manifest themselves as a pair of dots or, far more likely, smudged lines in the magnitude plot of the Fourier transform. The effect should be similar to the line at approximately –45° seen in the Fourier transform of the zebra image back in Lecture 5. That line was presumably caused by the zebra's stripes.

We have no idea what the Fourier transform of the rest of the image will look like, but the effect from the motion lines will be superimposed upon whatever it is. The frequency-domain lines will be oriented 90° to the lines in the original array. In the one-dimensional case, there are only two possible directions of motion, left and right (or 0° and 180°, if you prefer). Leftward motion will result in a line with an angle between 0 and 90° in the (magnitude plot of the) Fourier transform (symmetrical about the origin, …); rightward motion will result in a line with an angle between 90° and 180° in the Fourier transform; while no motion will result in a line superimposed on the horizontal axis of the Fourier transform.

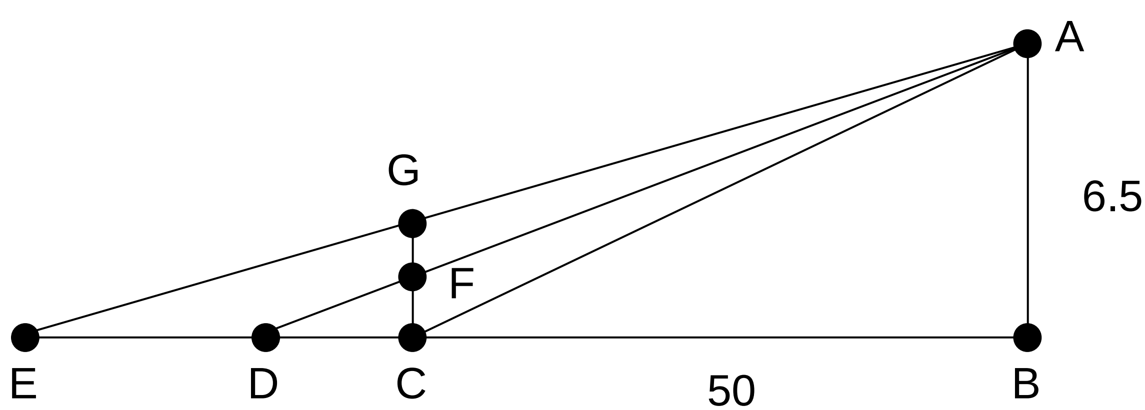
In order to determine local velocity, we must still determine the angle between the extra line in the Fourier transform and the transform's horizontal axis. Once we know this angle, call it θ, we'll know that the direction of motion *in the pixel array* is $\theta - 90°$; let's call that angle ϕ. To determine the speed of the motion, note that we're assuming constant velocity, which means that neither the speed nor the direction of the motion changed in the sampling time. Since each change of one row corresponds to one time unit; that is, moving up one row in the original array corresponds to advancing one more time unit, we're going to simplify things and say that one vertical pixel corresponds to one second. Since we want to know the speed in pixels per second, we need to take the cotangent of ϕ. Therefore, the local velocity of motion has an angle of 0° (rightward) if ϕ is less than 90°, and 180° (leftward) if ϕ is between 90° and 180°. The magnitude of the velocity is cot ϕ, with ϕ being determined as described above.

In the case of two-dimensional motion, we would be starting with a three-dimensional array (two spatial dimensions and one temporal dimension), so we'd end up with a three-dimensional Fourier transform. The motion will result in a plane superimposed on the magnitude plot of the transform. Then you need to find the angles that the plane makes to the *x*-axis and *y*-axis of the transform and use those angles to determine the *x* and *y* components of the velocity, using the cotangent function to find each component with respect to time. Then the magnitude of the velocity would be the hypotenuse function (square root of the sum of the squares) of the *x* and *y* components, and the angle would be the arctangent of *y*/*x*.

Question 2: Trouble with the Stereo

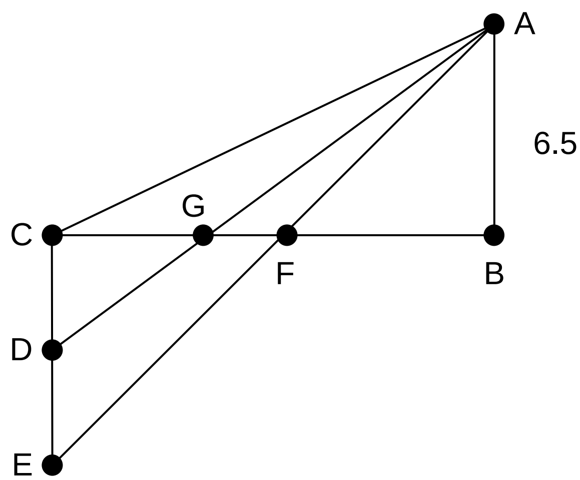
a) If there are several repetitions inof the pattern on the image plane in horizontal direction, and your program (or your own visual system!) does a poor job finding correspondences, then the wrong instance of the pattern can be used to determine the discrepancy, which will result in an incorrect value for the distance.

b) Look at the diagram below, which is grossly out of scale and tipped on its side compared to the drawings we're used to seeing. Let's say that the left camera is centered at B, the right camera is centered at A, the object is at C, and repetitions of the object are 3 cm from C, at F; and 3 more cm away, at G. To find two artificially long distances for the object, let's say that the left camera's image is interpreted correctly, but the correspondence for the right camera's image is off. First, let's say that the right image thinks we're looking at point F. The triangle FCD is going to be similar to the triangle ABD. The ratio of CF to CD will be the same as the ratio of 6.5 to 50+CD. We know CF is 3 cm, so 3/CD = 6.5/(50+CD). Churning through the algebra yields CD=42.86, which means that the perceived distance will be 92.86 cm. Using the same logic, but with the similar triangles being GCE and ABE, we get a perceived distance of 650 cm.



Now to find two incorrect distances smaller than 50 cm, consider the diagram below, also grossly out of scale. Once again, B is the center of the left camera, A is the center of the right camera, C is the correct pattern, and D and E are repetitions spaced 3 cm apart from each other. BC is our original 50 cm distance. ABC is the triangle formed by the correct correspondence. If the right image is misinterpreted, it could think it's looking at Point D. ABG and DCG are similar triangles. We already know that DC is 3 cm, so 3/6.5 = CG/BG. But BG+CG = 50, so we can substitute

50−BG for CG. Solving for BG, we get a perceived distance of 34.21 cm. Using the same logic, but with the right correspondence error perceiving Point E as the correct one, we have similar triangles ABF and ECF. Now we have 6/6.5 = CF/BF. Substituting 50−BF for CF eventually yields a value for BF, the erroneous distance, of 26.



Question 3: Stereo Vision with Paper and Pencil

a) First note that I used Microsoft Paint to get the pixel locations in gary.jpg. Each image was 375 pixels wide, so the values for x_l, taken from the right image, are relative to an x value of 562.5 pixels (= the 375 pixels from the left image plus the 187.5 pixels to the center of the right image). The values for x_r, taken from the left image, are relative to the center of that image, which is x = 187.5 pixels. The corresponding points were assumed to have the same y values; there was actually a discrepancy of a few pixels in Points B and E. I'm going to ignore the y values in the following data.

We start with Point A. We know that its depth is 200 cm. The baseline is 6.5 cm. The images are each 5 cm wide, which means that there are 375/5 = 75 pixels/cm or 0.01333… cm/pixel. Using the formula from Lecture 8, Slide 7 (or Slide 13), we can calculate the focal length of the lenses if we know the discrepancy, $x_l - x_r$, for Point A. For Point A, I got values of x = 17.5 pixels for the left image and x = –11.5 pixels for the right image. This translates to x_l = 0.2333 cm and x_r = –0.15333 cm. Plugging all these numbers into the formula yields f = 11.897 cm.

b) Now that we know f, we can use the formula to calculate the z values for the other four points, once we've found each of their x_l and x_r values. I've put them in the following table. The discrepancy values are in pixels; to convert to cm, just divide by 75. Finally, to calculate the z values, divide the discrepancies by 75 to convert them to centimeters; then multiply the baseline, 6.5, by the focal length, 11.897, then divide that product by the discrepancy (in cm) and you'll get z in cm. The values I got are shown in the last column of the table.

	x_l	x_r	$x_l - x_r$ **(pixel)**	$x_l - x_r$ **(cm)**	z
Point A	17.5	–11.5	29	0.38666…	200.00
Point B	–30.5	–73.5	43	0.57333…	134.88
Point C	–168.5	–181.5	13	0.17333…	446.13
Point D	–71.5	–85.5	14	0.18666…	414.27
Point E	21.5	–34.5	56	0.74666…	103.57

I was a little surprised to find that Point C seems to be farther away than Point D, but I double-checked the image and calculations. The numbers for Points B, D, and E are entirely plausible, given our starting value for Point A.