Course notes, CS664, 9/2/04

- Administrivia: Ashish B-exam, Friday 10-11, 310 Rhodes

STEREO (AGAIN)

- Back to stereo. Depth is not even a well defined term, at least at the level of pixels. Consider transparency, mixed pixels, occlusions.

- Almost all stereo algorithms ignore these issues.

- Even in pretty simple cases, the light rays see things at slightly different angles.

- There are surfaces that look the same from all angles (Lambertian). There are surfaces that change dramatically with a small change in viewpoint (specularities).

- Most stereo algorithms assume the world is Lambertian. It isn't.

No stereo algorithm handles specularities very well.

- However, with typical camera geometry, a lot of surfaces are approximately Lambertian. So good stereo algorithms work surprisingly well, as long as you stay away from the places where they don't work at all!

- Most stereo algorithms work by trying to find a disparity map. Assign each pixel an integer disparity (shift). Can also view this as a mapping from the left image to the right image, or as a depth map.

- In fact, only with the simplest camera geometry is the shift 1-D (along rows in the image, called "scan lines"). This leads to a related problem of determining the relationship between one camera and another. Usually solved by having a calibration object in the scene. In the simplest case, the relationship is a rigid body motion (6 dof, 3 rotation and 3 translation).

- Note that for most applications you're particularly interested in the discontinuities. We will talk about this issue a lot...

- For example, you might want to use stereo depth to aid object segmentation. That way you can compute affinity from something more interesting/meaningful than intensity.

- The biggest distinction between stereo algorithms involves how they compute disparities. This can be done either locally or globally.

- Local algorithm definition: the same method is run independently at each pixel. Usually some variant of correlation. Based on statistics.

- Local item tradeoffs: fast, simple to implement. But results aren't very good, especially in low-texture areas and at discontinuities.

- Global algorithm definition: compute an entire disparity map at once. Usually based on optimization, called "energy minimization" in vision.

- Not as fast as local methods, but gives much better results.

- A natural generalization of stereo involves more than 2 cameras. Sometimes called "multicamera scene reconstruction".

- Hot topic in the last few years. Useful for various graphics tasks, especially model creation.

- Harder than (2-camera) stereo because (1) the viewing angles really change (2) you have to pay attention to visibility, unlike occlusion (3) there is no "simple case" for camera geometry.

MOTION

- Motion problem definition: like stereo, but temporally separated images.

- Can be viewed as very similar to stereo, just with 2D disparities (shifts).

- Almost all motion algorithms work for stereo and vice-versa, just nowhere near as well. Visual correspondence is just code for "motion and stereo".

- A natural problem to think about in a continuous framework, as if we are sampling from $I(x, y, t)$ at closely spaced intervals in t.

- This leads to some very elegant math that doesn't work particularly well in practice. But it's a good thing to know about, so we will cover it.