Course notes, CS664, 9/21/04

- Static clues redux: consider intensities $[99, 101]$ at pixels (x, y) and $(x+1, y)$ in the left image. Let us calculate the energy these contribute at disparity d_a, d_b. Data energy is
$$E_{data} = (99 - I_R(x - d_a, y))^2 + (101 - I_R(x + 1 - d_b, y))^2.$$
Smoothness energy is $E_{smooth} = T[d_a \neq d_b]$. Total energy is
$$E_{data} + \lambda \cdot E_{smooth}$$

- Now consider the intensities $[9, 201]$ at pixels (x, y) and $(x+1, y)$ in the left image. Let us calculate the energy these contribute at disparity d_a, d_b. Data energy is
$$E_{data} = (9 - I_R(x - d_a, y))^2 + (201 - I_R(x + 1 - d_b, y))^2.$$
Smoothness energy is $E_{smooth} = T[d_a \neq d_b]$, the same as above.

- However, it should be cheaper to have a change in discontinuity at a change in intensity, and more expensive to have a change in discontinuity at no change in discontinuity. This means the smoothness term depends to some extent on the data!

- Easy solution (we called it "static clues", it has other names): change
$$E(f) = \sum_p D(f(p)) + \lambda \sum_{p,q \in \mathcal{N}} T[f(p) \neq f(q)]$$

to

$$E(f) = \sum_p D(f(p)) + \lambda \sum_{p,q \in \mathcal{N}} w_{p,q} T[f(p) \neq f(q)]$$

where $w_{p,q}$ is small if $I(p)$ is very different from $I(q)$ and large if $I(p) \approx I(q)$

- Notice the importance of $E_{data} + \lambda \cdot E_{smooth}$. We see this a **lot** in vision!

FEATURE SPACE ANALYSIS

- One problem with this is that the labels are fixed in advance (intensity for image restoration, displacements for motion and stereo).

- Think about color image segmentation. There are 256^3 colors, you don't want these to be your labels. But what a human perceives as a color is typically some kind of "grouping" in color space (RGB). [In fact, a very complicated grouping...]

- You'd like your labels to be these groupings, but you don't know the groupings in advance.

- Natural approach is to dynamically construct your label set from the image. Each pixel contributes a point in RGB space; map the entire image into RGB space, then cluster this space. The clusters will be the labels for your pixel labeling problem.

- A similar problem occurs with motion: think about a rotation of some of the pixels. There are **many** possible rotations (different centers, speeds). Each pixel's local motion is consistent with some rotations, so map to rotation space and cluster there.

- Generalization: each pixel has a vector of features, and we label pixels with clusters in feature space. This is called *feature space analysis* and is very popular, especially for motion and texture.

- Major issue: nice-looking clusters in feature space do not necessarily form nice-looking regions in the image. Standard solution: positional features.

FEATURES

- So much for algorithms that work on the "whole" image (sometimes called dense methods). We'll now look at a few methods that talk do "sparse" processing of images in some manner.

- Feature finding: corners, "interest points" (SIFT)

- Applications of features: Calibration, registration, SFM. Most of these set up some kind of minimization problem somewhat like registration, typically with just a data term (but not always).

LOW/MEDIUM/HIGH TAXONOMY

- Also called "early/late" vision

- Somewhat based on neuroanatomy (very loosely, actually)

- Low-level is pixel based, i.e. edge detection, stereo, motion

- High level is recognition.

- Everything else is up for grabs.

TRACKING

- Tracking: whole person tracking versus motion segmentation; contour-based tracking; tracking over slices in medical imaging; point tracking

- Applications of tracking: surveillance, recognizing activities, recognizing people