

Course notes, CS664, 9/28/04

- Administrivia: first exam today.

RECOGNITION (RECENT)

- Face detection is actually pretty good these days, mostly by using learning methods. We will talk about these a bit when we discuss statistical methods.
- Deformable models/collections of parts; attempts to handle objects such as people.
- You can view this as combining the old-school work on articulated objects with appearance models.
- Example: pictorial structures, which consist of a set of features linked by springs. Somewhat similar to physics-based approaches.

- Again, we have

$$E_{data} + \lambda \cdot E_{smooth}$$

where the data term comes from the appearance models, and the smoothness term imposes geometric constraints.

- For many applications, the structure is actually a tree (think of a torso with head and limbs, for example).
- Under these circumstances there are good optimization algorithms

OPTIMIZATION: GENERAL PRINCIPLES

- General comment: many vision problems can be looked at via optimization. Sometimes a small tweak to the problem definition (energy function) will result in something you can actually optimize, hence solve fast.
- Problem definition: candidate solutions, search space, energy function, global versus local minimum

- Search space can be continuous or discrete. Continuous optimization methods (clever versions of gradient descent) mostly concentrate on how fast you can get to a local minimum, not how good it is. We will talk about these briefly in a few lectures (their classic use in vision is for motion).
- If the problem has a unique global minimum it is said to be convex. (Sometimes the term is used to describe a slightly stronger property related to geometry, which implies a unique global minimum but is not necessary.)
- A non-convex problem effectively requires you to go “uphill” for a while before going downhill to the solution.
- One classical technique is to begin by making your function convex, for instance by smoothing it a lot. You can then incrementally make it more non-convex, using the output from last time as your new starting point. This is called a *continuation method*.
- Approximation algorithms are very powerful but rare in vision. Note that some problems are known to be hard even to approx-

imate! Example: graph coloring.

- Problem reduction cuts both ways. If you can turn your problem into something easy, that's great. If you can turn something known to be hard into your problem, you know you can't solve it fast. Also note that reductions must be efficient (usually authors keep this implicit).
- Advantages of optimization: (1) clean separation between what you compute and how you compute it (2) easy to add new constraints (terms) (3) simple to explain (cf Physics-Based Vision)
- Disadvantages of optimization: doing it is difficult, and the clean separation can easily hurt you (perhaps a nearby problem formulation is tractable?) This is especially the case in early vision, where you tend to end up with non-convex energy functions and tens of thousands of dimensions.

GENERAL PURPOSE OPTIMIZATION METHODS

- General purpose optimization is intractable even if $P = NP$. Consider a well in Kansas.

- Consequences: such methods are asymptotically equivalent to exhaustive search (though they may work for a particular problem). Moreover, strong methods are restricted to a limited set of energy functions.
- Several examples with low status in vision, such as genetic algorithms. Idea: population of candidate solutions, represented somehow. Create new populations by crossovers or mutations, replacing the worse (highest energy) candidates.
- Choice of representation and energy function are key. With good ones this can work well (same is true of exhaustive search).
- Most popular general-purpose optimization method in vision: simulated annealing

SIMULATED ANNEALING

- Annealing is an attempt to deal with local minima (non-convexity).
- If no local minima, you can just roll down-hill. What if we could

sometime move uphill?

- The dimensionality of the search problem also matters, a lot. In early vision, dimension is number of pixels (about 100K).
- We will go into annealing in some detail, and ask you to implement it (first problem set).
- Annealing is closely related to the Metropolis algorithm, a very old (1953) paper with a famous co-author.
- We'll begin with the Metropolis algorithm, a simple randomized algorithm with a critical property.

METROPOLIS ALGORITHM

- Metropolis Algorithm, parameterized by energy E , temperature T .

- Start somewhere (i.e., with a labeling f of the image, that has some energy).
 1. Generate a new random state change f' (i.e., change the label of a pixel from f). [This step is called sampling]
 2. Let $\Delta E = E(f') - E(f)$. If $\Delta E < 0$ (downhill move), set $f = f'$. Otherwise (uphill move), with probability proportional to $\exp(-\frac{\Delta E}{T})$, set $f = f'$.