Math aside, #1

Linear Least Squares (§3.1)

- Very common problem in vision: solve an over-constrained system of linear equations
  - e.g., Ux=y, where U has more rows than needed
  - e.g., Ux=0, |x|=1, where U has more rows than needed
- More equations allows multiple measurements to be used
- Least squares means that you minimize squared error (the difference between your model and your data)
- Least squares minimization is (relatively) easy
- Not very robust to outliers (assumes error is Gaussian)

Linear Least Squares (§3.1)

We will look at two problems

First, $U\mathbf{x} = \mathbf{y}$ where U has more rows than needed

Second, $U\mathbf{x} = 0$ subject to $|\mathbf{x}| = 1$ where U has more rows than needed

We can use the **first** for naïve spectral camera calibration.

We will use the **second** problem for geometric camera calibration.

Non-homogeneous Least Squares*

Problem one $U\mathbf{x} = \mathbf{y}$ where U has more rows than needed

U is not square, so inverting it does not work

In fact, usually **there is no solution**. We need to redefine what it means to "solve the equation".

We seek the "best" answer but what is that?

* This is regression by a different name.

Non-homogeneous Least Squares

Define $\mathbf{e} = U\mathbf{x} - \mathbf{y}$ and $E = |\mathbf{e}|^2 = \mathbf{e}^T\mathbf{e}$

The least squares solution which is the one that has minimum E.

We can derive the answer by differentiating with respect to each x_i, and setting all resulting equations to zero (see supplementary slides and/or homework).

The answer is given by

$\mathbf{x} = U^{\dagger}\mathbf{y}$ where $U^{\dagger} = (U^T U)^{-1} U^T$ is the pseudoinverse of U

Important

Non-homogeneous linear least squares summary
(the part you need to know)

You should be able to set up

$$U\mathbf{x} = \mathbf{y}$$

You should know that it is solved by

$\mathbf{x} = U^{\dagger}\mathbf{y}$ where $U^{\dagger}$ is the pseudoinverse of U

You can assume that you can look up

$$U^{\dagger} = (U^T U)^{-1} U^T$$

*You should also keep in mind that for numerical stability, one may want to use a different approach to solve

$$U^T U\mathbf{x} = U^T\mathbf{y}$$

without matrix inversion.

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

Remember the fact that the camera has a spectral sensitivity R(λ). So how do we find it out?

Recall that $\rho = \int L(\lambda)R(\lambda)d\lambda$

has the discrete version

$$\rho = \mathbf{L} \bullet \mathbf{R}$$

(previously we accounted for multiple channels with the superscript (k), but here we just consider each channel separately)

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

Strategy: measure some spectra entering the camera, $\mathbf{L}_i$, and note the response, ρ_i.

So we have, for a bunch of measurements, i:

$$\rho_i = \mathbf{L}_i \bullet \mathrm{R}$$

If we don't have enough measurements, then the problem is under constrained. To account for noise, we want to use multiple measurements.

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

From previous slide:

$$\rho_i = \mathbf{L}_i \bullet \mathbf{R}$$

(for a number of measurements indexed by *i*.)

We form a matrix L with rows $\mathbf{L}_i$, a vector **P** with elements ρ_i, and solve the least squares equation

$$\mathrm{L}\mathbf{R} = \mathbf{P}$$

(R is the unknown).

Spectral camera calibration improvements

A) Constrain the sensitivities to be positive

B) Promote the sensitivity functions to be smooth
(This is often referred to as regularization).

How to do this? See grad student part of next assignment.