CS 217: Light and Geometry in Vision **Spring 2009**

Lecture 10 — April 20

Scribe: David Keator *Lecturer: Deva Ramanan*

Note: These lecture notes are still rough, and have only have been mildly proofread.

10.1 Agenda

Numerical methods (calculus of variations, linear systems, conjugate gradients

10.2 Recall

Recall in the last lecture we derived a normal field: $N(x, y) = [abc]^T$ where $p = \frac{\partial f}{\partial x} = \frac{-a}{c}$ and $q = \frac{\partial f}{\partial y} = \frac{-b}{c}$. Now we want to find the minimum function:

$$min_{f(x,y)} = \int\int (\frac{\partial f}{\partial x} - p)^2 + (\frac{\partial f}{\partial y} - q)^2 \partial x \partial y \quad (10.1)$$

10.3 Calculus of Variations (appendix of Robot Vision)

10.3.1 Calculus

Let's us find parameters at the extrema of an expression:

$$\begin{aligned} f(x) &= ax^2 + bx + c \\ \frac{\partial f}{\partial x} &= 0 \\ 2ax + b &= 0 \\ x &= \frac{-b}{2a} \end{aligned} \quad (10.2)$$

10.3.2 Calculus of variations

Let's us find functions at extrema. Say for example we want to find an $f(x)$ that maximizes (or minimizes) the expression I:

$$I = \int_{x_1}^{x_2} F(x, f(x), f'(x))dx \quad (10.3)$$

One idea is to replace $f(x)$ with $f(x) + \epsilon\eta(x)$ where we jitter the function a bit just as we do for normal derivatives with respect to a variable. For $f(x)$ to be the solution to 10.3 we must ensure that I does not change for small ϵ. We can now write I in the form:

$$\tilde{I} = \int_{x_1}^{x_2} F(x, f(x) + \epsilon\eta(x), f^{'}(x) + \epsilon\eta^{'}(x))dx \tag{10.4}$$

To solve 10.4 we expand it into a Taylor series about $F(x, f(x) + \epsilon\eta(x), f^{'}(x) + \epsilon\eta^{'}(x))$ and differentiate with respect to ϵ and use integration by parts. The solution will be the Euler Lagrange Equation:

$$F_f - \frac{\partial}{\partial x} F_{f'} = 0 \tag{10.5}$$

$$F_f = \frac{\partial F(x, f, f^{'})}{\partial f}$$

$$F_{f'} = \frac{\partial F(x, f, f^{'}}{\partial f'}$$

For example:

$$\begin{gathered} min_z \int\int F(z, z_x, z_y)dxdy \\ where\ z(x,y) = height\ field\ we\ want\ to\ reconstruct \\ using\ the\ Euler\ equations: \\ \tfrac{\partial^2 z}{\partial x^2} + \tfrac{\partial^2 z}{\partial y^2} = px + py \\ where\ px = \tfrac{\partial p}{\partial x} \end{gathered} \tag{10.6}$$

10.4 Discretized Linear Systems

Given the function $f(z)$ such that:

$$\begin{gathered} f(z) = \int\int (Zx - p)^2 + (Zy - q)^2 dxdy \\ = \tfrac{1}{2}(Az - r)^T(Az - r) = f(z)\ discretized\ over\ a\ pixel\ grid \end{gathered} \tag{10.7}$$

Such that Az = vector of x and y derivatives. How do we compute this in our computer? Set up two matrixes, one for z and one for A where entries 1 and -1 are pixel selectors:

$$z = \begin{pmatrix} z(1,1) \\ z(1,2) \\ . \\ . \\ z(h,w) \end{pmatrix} A = \begin{pmatrix} 1\ -1\ 0\ 0..... \\ 0\ 1\ -1\ 0..... \\ .... \end{pmatrix} r = \begin{pmatrix} p(1,1) \\ . \\ . \\ p(h,w) \\ q(1,1) \\ . \\ . \\ q(h,w) \end{pmatrix} \tag{10.8}$$

$$[Az]_1 = z(1,1) - z(1,2) \approx \frac{\partial z(1,1)}{\partial x}$$

We then solve the linear system for z:

$$f(z) = \tfrac{1}{2} z^T A^T A z - 2 z^T A^T r + \tfrac{1}{2} r^t r \quad (10.9)$$
$$z = (A^T A)^{-1} A^T r \approx z = A \setminus r \ (in\ matlab,\ gaussian\ elimination)$$

One problem with the above formulation is that A is a very large and very sparse matrix. Inversion might be a problem. For large linear systems we can use relaxation techniques.

10.5 Large Linear Systems (Gilbert Strang - Intro to Linear Algebra)

These are methods for solving $Ax = b$ where A is square. If A is not square we can use $A^T Ax = A^T b$ to make it square.

10.5.1 Relaxation Techniques (Robot Vision)

Start by splitting $A = D + E$. Note that $X^{(k+1)} = -D^{-1}Ex^k + D^{-1}b$. This leads us to an iterative solution $Dx^{k+1} = -Ex^k + b$.

If D is a diagonal matrix, we can use the Jacobi method and invert D and put it on the other side.

If D is a lower triangle matrix, we can use the Gauss Seidel method which uses back substitution instead of inverting D to solve for X^{k+1}.

10.5.2 Analysis of A

Let $B = D^{-1}EandC = D^{-1}b$. Let $X^k = x + \sum_j a_j v_j$ where x is the true solution and $\sum_j a_j v_j$ we can think of as the error of our own solution. $\sum_j a_j v_j$ is the linear combination of eigenvectors of B:

$$\begin{aligned} X^{(1)} &= Bx^{(0)} + C \\ &= B(x + \textstyle\sum_j a_j v_j) + C \\ &= Bx + \textstyle\sum_j a_j \lambda_j v_j + C \ [Bx + C = x] \\ &= x + \textstyle\sum_j a_j \lambda_j v_j \\ x^k &= x + \textstyle\sum_j a_j \lambda_j^k v_j \end{aligned} \quad (10.10)$$

If we want equation (10.10) to converge, the eigenvalues λ_j^k must be strictly less than 1. Further, the maximum eigenvalue is the key quantity in convergence. The relaxation technique will converge if $max|\lambda_j| < 1$. The rate of convergence is faster for smaller values.