Calculus of Variations and Computer Vision

Sharat Chandran

Computer Science & Engineering Department,
Indian Institute of Technology,
Bombay.
http://www.cse.iitb.ernet.in/~sharat
January 8, 2002
The Top Ten Algorithms of the Century

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- QR Algorithm for Computing Eigenvalues
- The Fortran Optimizing Compiler
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method
Overview

1. Sample problems from computer vision (fade out computer vision gurus)
   - The problem of shape
   - The problem of obtaining 3D information
   - The problem of motion

2. The beauty of mathematics (fade out math gurus)
   - The method of the calculus of variations
   - Allied mathematical methods: Singular Value Decomposition, and the Method of Lagrange Multipliers

3. Sample solution to problems posed (or how computer scientists teach/cheat)

4. Concluding remarks
Shape

- When placed out of context, images can be quite involved
- What you get is not what you see (WYGINWYS)
- What you want to compute: A contour (useful for obtaining properties such as size)
- Key mathematical concept: parametric curve $x(s), y(s)$
Motion

• Optical flow is an intensity-based approximation to image motion from sequential time-ordered images

• Key mathematical concept: Two functions: $u(x, y)$ and $v(x, y)$

• What does it look like?

• How do we compute optical flow?
Application of Optical Flow

A short video segment
Specifications to help.
Two Problems

• Determine the equation of the curve joining two points \((x_1, y_1)\) and \((x_2, y_2)\) on the plane such that the length of the curve joining them is minimum

• Goal: Prove that your answer is correct

• Similar problem: Determine the equation of the support so that a ball placed at \((x_1, y_1)\) reaches (due to gravity) \((x_2, y_2)\) in the least possible time
Functions and Functionals

- Differential segment length is \((dx^2 + dy^2)^{1/2}\) where \(dx\) and \(dy\) are infinitesimal lengths in the x and y directions along the curve.
- Therefore the length of the curve joining the two end points \((x_1, y_1)\) and \((x_2, y_2)\) is

\[
J = \int_{x_1}^{x_2} \sqrt{1 + y'} \, dx = \int_{x_1}^{x_2} F(x, y, y') \, dx
\]

- For the first problem, what should the function \(y\) be so that \(J\) is minimized?
- Compare: Find \(x_0\) such that \(y = f(x)\) is minimum.
- \(J\) is a functional, a quantity that depends on functions rather than dependent variables.
Solution to the Shortest Length Problem

• The Euler-Lagrange equation is a necessary condition for minimising $J$

$$F_y - \frac{d}{dx} F_{y'} = 0$$  \hspace{1cm} (1)

• In our problem, $F = \sqrt{1 + y'}$. Applying Equation 1 we get

$$\frac{d}{dx} \left( \frac{y'}{\sqrt{1 + y'}} \right) = 0$$

• Solution: $y'$ is a constant.

• The shortest distance curve is a straight line.
Detour: Why the Euler equation

- \( J = \int_{x_1}^{x_2} F(x, f, f') \, dx \) with boundary conditions \( f(x_1) = f_1 \) and \( f(x_2) = f_2 \).
- Let \( \eta(x) \) be a test function. Deform \( f \) by \( \epsilon \eta(x) \). Then, \( \frac{dJ}{d\epsilon} = 0 \) at the “right” place.
- This is true for all test functions \( \eta(x) \). The boundary conditions assert \( \eta(x_1) = \eta(x_2) = 0 \).
- If \( f(x) \) is replaced by \( f(x) + \epsilon \eta(x) \) then \( f'(x) \) will be replaced by \( f'(x) + \epsilon \eta'(x) \).
- The integral then becomes

\[
J = \int_{x_1}^{x_2} F(x, f + \epsilon \eta, f' + \epsilon \eta') \, dx
\]
Detour: Why the Euler equation

- If $F$ is differentiable, expand the integrand in a Taylor series.
- Differentiate with respect to $\epsilon$ and set to zero.
- Apply integration by parts to get
  \[ \int_{x_1}^{x_2} \eta'(x) F_f dx = [\eta(x) F'_f]_{x_1}^{x_2} - \int_{x_1}^{x_2} \eta(x) \frac{d(F_f)}{dx} dx, \]
- The first term is zero due to the boundary conditions.
- Hence
  \[ \int_{x_1}^{x_2} \eta(x)(F_f - \frac{dF'_f}{dx}) dx = 0 \]
Bottom Line Euler Equations

• Various generalizations are possible
  – Higher order derivatives: \( J = \int_{x_1}^{x_2} F(x, f, f', f'', ...) \, dx \)
  – Integrand may depend on several functions instead of only one
  – Finding functions that have two independent variables

\[
J = \int \int F(x, y, f, f_x, f_y) \, dx \, dy
\]

• The bottom line

<table>
<thead>
<tr>
<th>Function to optimize</th>
<th>The Euler-Lagrange equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \int F(x, u_x) , dx )</td>
<td>( F_u - \frac{d}{dx} F_{u_x} = 0 )</td>
</tr>
<tr>
<td>( \int F(x, u_x, u_{xx}) , dx )</td>
<td>( F_u - \frac{d}{dx} F_{u_x} - \frac{d^2}{dx^2} F_{u_{xx}} = 0 )</td>
</tr>
</tbody>
</table>
| \( \int F(x, u_x, u_x) \, dx \) | \( F_u - \frac{d}{dx} F_{u_x} = 0 \)
| \( \int F(x, y, u_x, u_y) \, dx \, dy \) | \( F_u - \frac{d}{dx} F_{u_x} - \frac{d}{dy} F_{u_y} = 0 \) |
The Snake Formulation

- Start with any old curve (a “snake”). Mathematically deform it to be the desired shape (based on image data).
- Curve is \( \mathbf{v}(s) = (x(s), y(s)) \). Define energy of the curve to be

\[
E(\mathbf{v}(s)) = \int_{0}^{1} E_{\text{snake}}(\mathbf{v}(s)) \, ds \quad (2)
\]

where \( E_{\text{snake}} = E_{\text{int}} + E_{\text{ext}} \)

- Find the curve that minimizes Equation 2

Remember, actual image is not so clean!
More on Formulation

- The energy functional is given by 
  \[ E = \int_0^1 (E_{int}(v(s)) + E_{img}(v(s)) + E_{con}(v(s))) \, ds \]
- \[ E_{int} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2 \]
- \[ E_{con} = -k(v(s) - \bar{x})^2 \]
- \[ E_{img} = w_{line}E_{line}(v(s)) + w_{edge}E_{edge}(v(s)) + w_{term}E_{term}(v(s)) \]
  - Line energy is \( E_{line}(v(s)) = \mathcal{I}(x, y) \)
  - Edge energy is \( E_{edge}(v(s)) = -|\nabla \mathcal{I}(x, y)|^2 \)
Applying Euler-Lagrange to Snakes

We minimize

\[ J_1 = \int \left( \alpha(s)x_s^2 + \beta(s)x_{ss}^2 \right)/2 + E_x(s) \, ds = \int X(s, x, x', x'') \, ds \]

and

\[ J_2 = \int \left( \alpha(s)y_s^2 + \beta(s)y_{ss}^2 \right)/2 + E_y(s) \, ds = \int Y(s, y, y', y'') \, ds \]

For \( X(s, x, x', x'') \) the Euler equation is

\[ X_x - \frac{d}{ds}X_{x'} + \frac{d^2}{ds^2}X_{x''} = 0 \]
More on Application

- $X_x = \frac{\partial E}{\partial s}, \quad X_x' = \alpha(s)x'(s), \quad$ and $X_x'' = \beta(s)x''(s)$$
- \frac{d}{ds}X_x' = \alpha'x' + \alpha x''$ and
- $\frac{d}{ds}(X''(s)) = \beta'x'' + \beta x'''$ and
- $\frac{d^2}{ds^2}(X''(s)) = \beta''x'' + 2\beta'x''' + \beta x'''$

- For purpose of illustration use only two terms: $\frac{\partial E}{\partial s} - \alpha'x' - \alpha x'' = 0$

- Numerically approximating we have
  - $-\frac{\partial E}{\partial x} \simeq E_{i+1} - E_i$ at location $i$
  - $\alpha'(s) \simeq \alpha_{i+1} - \alpha_i$
  - $x'(s) \simeq x_{i+1} - x_i$
  - $\alpha'x' = (\alpha_{i+1} - \alpha_i)(x_{i+1} - x_i) = (\alpha_{i+1} - \alpha_i)x_{i+1} - (\alpha_{i+1} - \alpha_i)x_i$
  - $x''(s) \simeq x_{i+1} - 2x_i + x_{i-1}$
Reducing Snakes To A Matrix Equation

\[
\begin{pmatrix}
-1 & 1 & \ldots & 0 \\
0 & -1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & -1
\end{pmatrix}
\begin{pmatrix}
E_1 \\
E_2 \\
\vdots \\
E_n
\end{pmatrix}
- \\
\begin{pmatrix}
-(\alpha_2 - \alpha_1) & (\alpha_2 - \alpha_1) & 0 & \ldots & 0 \\
0 & -(\alpha_3 - \alpha_2) & (\alpha_3 - \alpha_2) & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & -(\alpha_{n+1} - \alpha_n)
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
= 0
\]

This can be written as \( Ax = B \), and solved using SVD.
The Method of Lagrange Multipliers

Minimize a function \( f(x, y) \) subject to \( g(x, y) = 0 \).

• For example, find a point on the line \( x \sin \theta - y \cos \theta + p = 0 \) closest to origin

• That is, minimize \( x^2 + y^2 \) subject to the given constraint.

In MOLM, we generate a new function \( E = f + \lambda g \) and set the partial derivatives with respect to \( x, y \) and \( \lambda \) to zero.

In the example

• \( 2x + \lambda \sin \theta = 0 \)

• \( 2y + \lambda \cos \theta = 0 \)

The solution is the obvious one

• \( x = -p \sin \theta \)

• \( y = +p \cos \theta \)

The method can be generalized to larger number of constraints.
Formulation for Motion

- Let $E(x, y, t)$ be the intensity value at point $(x, y)$ in the image.
- Due to motion, let this point correspond to a point $(x + \delta x, y + \delta y)$ at time instance $t + \delta t$.
- Key assumption: $E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t)$.
- Two parts to determine $u(x, y)$ and $v(x, y)$:
  - Use Taylor’s theorem to get $E_x u + E_y v + E_t = 0$.
  - Introduce a smoothness constraint: $f(u, v, u_x, v_x, u_y, v_y) = u_x^2 + u_y^2 + v_x^2 + v_y^2$.
- Using MOLM we have a functional that needs to be minimized:
  $\int \int f(u, v, u_x, v_x, u_y, v_y) + \lambda g(u, v, t)^2 \, dx \, dy$
  $\int \int u_x^2 + u_y^2 + v_x^2 + v_y^2 + \lambda (E_x u + E_y v + E_t)^2 \, dx \, dy$.
Applying Euler-Lagrange for Optical Flow

• We have
  \[ F_u - \frac{\partial}{\partial x} F_{ux} - \frac{\partial}{\partial y} F_{uy} = 0 \]
  \[ F_v - \frac{\partial}{\partial x} F_{vx} - \frac{\partial}{\partial y} F_{vy} = 0 \]

• Applying these we get
  \[ F_u = 2\lambda E_x (E_x u + E_y v + E_t) \]
  \[ F_{ux} = 2u_x, \frac{\partial}{\partial x} F_{ux} = 2u_{xx} \]
  \[ F_u = 2\lambda E_x (E_x v + E_y v + E_t) \]
  \[ F_{vx} = 2v_x, \frac{\partial}{\partial x} F_{vx} = 2v_{xx} \]

• And \( F_v = 2\lambda E_x (E_x u + E_y v + E_t) \)
  \[ F_{vy} = 2u_y, \frac{\partial}{\partial y} F_{vy} = 2u_{yy} \]
  \[ F_v = 2\lambda E_x (E_x v + E_y v + E_t) \]
  \[ F_{vy} = 2v_y, \frac{\partial}{\partial y} F_{vy} = 2v_{yy} \]

• This implies a coupled system of equations
  \[ \Delta^2 u = \lambda (E_x u + E_y v + E_t) E_x \]
  \[ \Delta^2 v = \lambda (E_x u + E_y v + E_t) E_y \]
Solution using Euler Lagrange

Discretizing we have

- $u_{k,l}$, $u_{k+1,l}$ $u_{k,l+1}$ $u_{k-1,l}$ $u_{k,l-1}$: $u$ component of the velocity at points $(k, l)$, $(k + 1, l)$, $(k, l + 1)$, $(k - 1, l)$ and $(k, l - 1)$ respectively.

- $v_{k,l}$, $v_{k+1,l}$ $v_{k,l+1}$ $v_{k-1,l}$ $v_{k,l-1}$: $v$ component of the velocity at points $(k, l)$, $(k + 1, l)$, $(k, l + 1)$, $(k - 1, l)$ and $(k, l - 1)$ respectively.

\[
\begin{align*}
  u_{k,l} &= \frac{\bar{u}_{k,l} - \lambda E_x E_t}{1 + \tilde{\lambda} E_x^2} - \frac{v_{k,l} E_x E_t}{1 + \tilde{\lambda} E_x^2} \\
  v_{k,l} &= \frac{\bar{v}_{k,l} - \lambda E_x E_t}{1 + \tilde{\lambda} E_x^2} - \frac{u_{k,l} E_x E_t}{1 + \tilde{\lambda} E_x^2}
\end{align*}
\]
# Uses of Euler Lagrange

<table>
<thead>
<tr>
<th>Problem</th>
<th>Regularization principle</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contours</strong></td>
<td>$\int E_{\text{snake}}(\mathbf{v}(s)) ds$</td>
</tr>
<tr>
<td>Area based Optical flow</td>
<td>$\int [(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \lambda(E_x u + E_y v + i_v)^2] dxdy$</td>
</tr>
<tr>
<td>Edge detection</td>
<td>$\int [(Sf - i)^2 + \lambda(f_{xx})^2] dx$</td>
</tr>
<tr>
<td>Contour based Optical flow</td>
<td>$\int [(V \cdot N - V^N)^2 + \lambda(\frac{\partial V}{\partial x})^2] dxdy$</td>
</tr>
<tr>
<td>Surface reconstruction</td>
<td>$\int [(S \cdot f - d^2 + \lambda(f_{xx} + 2f_{xy} + f_{yy})^2)] dxdy$</td>
</tr>
<tr>
<td>Spatiotemporal approximation</td>
<td>$\int [(S \cdot f - i)^2 + \lambda(\nabla f \cdot V + ft)^2] dxdydt$</td>
</tr>
<tr>
<td><strong>Colour</strong></td>
<td>$|I_y - Ax|^2 + \lambda|Pz|^2$</td>
</tr>
<tr>
<td>Shape from shading</td>
<td>$\int [(E - R(f_g))^2 + \lambda(f_x^2 + f_y^2 + g_x^2 + g_y^2)] dxdy$</td>
</tr>
</tbody>
</table>
| Stereo                           | $\int \{[\nabla^2 G \ast (L(x,y) - R(x + d(x,y),y))]^2 + \lambda(\nabla d)^2\} dxdy$
Concluding Remarks

- Two interesting problems have been described
- Defined and derived the Euler Lagrange equations
- Used mathematics to solve the computer vision problem