Outline - I

• Algebraic distance
  – Definition
  – Problems
  – Scaling and Normalization

• Different ways of computing the Cost function
  – Errors in both coordinates
  – Transfer Error and Reprojection Error

• “Physics/Geometry” based distances
  – General Examples
  – Examples in Vision

• Constraints
  – Equality constraints
    • Lagrange multipliers and Penalty function methods
  – Inequality Constraints
Outline - II

• Other Metrics
  – Riemann Lebesgue lemma
  – Sobolev norms

• Statistical Cost Functions
  – Mahalanobis distance
  – Maximum Likelihood (ML), Expectation Maximization (EM) and Maximum a Posteriori (MAP)

• Robust Estimation
  – Outliers and Inliers
  – Median Estimators
  – RANSAC
Typical Optimization Problems

- **Model fitting**
  - Fit a straight line or polynomial through data
    \[ y_i = \sum_j a_j x_j^i \]
  - Fit a sum of cosines, exponentials etc.
    \[ y_i = \sum_j a_j \phi_j(x_i) \]
    Model \( \phi_j \) s, parameters \( a_j \) s data \((x_i, y_i)\)

- **Determine a transformation**
  - Determine a homography matrix
    \[ x' = Hx \]
  - Determine the fundamental matrix
    \[ x'^tFx = 0 \]
Least Squares

- Look for a solution to a linear system of equations
  \[ \mathbf{A}x = \mathbf{b} \]
- Number of equations and unknowns need not match
- Look for solution by minimizing \[ \| \mathbf{A}x - \mathbf{b} \| \]
  – minimize the distance between the vectors \( \mathbf{A}x \) and \( \mathbf{b} \)
- Differentiate \( (A_{ij}x_{j} - b_{i}).(A_{ik}x_{k} - b_{i}) \) with respect to \( x_{l} \)
- Recall \( \frac{\partial x_{i}}{\partial x_{l}} = \delta_{il} \)
  \[
  \frac{\partial}{\partial x_{l}} (A_{ij}x_{j} - b_{i}) \cdot (A_{ik}x_{k} - b_{i}) = 0
  \]
  \[
  (A_{ij}\delta_{jl})\cdot (A_{ik}x_{k} - b_{i}) + (A_{ij}x_{j} - b_{i})\cdot (A_{ik}\delta_{kl}) = 0
  \]
  \[
  A_{il} \cdot (A_{ik}x_{k} - b_{i}) + (A_{ij}x_{j} - b_{i}) \cdot A_{il} = 2(A_{il}A_{ik}x_{k} - A_{il}b_{i}) = 0
  \]
  \[
  A_{il}A_{ik}x_{k} = A_{il}b_{i}
  \]
- Same as the solution of \( \mathbf{A}^t\mathbf{A}x = \mathbf{A}^t\mathbf{b} \)
 Optimization – Physical Cost Function

- Adjust parameters of a system or model to maximize or minimize something
  - $, Distance

- Ideally there is a real cost being minimized
  - E.g. Dollars or distance travelled
  - Then each equation makes sense

- Airlines: minimize costs, crew movement and plane takeoffs and landings, subject to regulatory constraints

- Traders: maximize returns for a given level of risk

- Some other physically measurable quantity
Algebraic Distance

- Algebraic system $Ax = b$
- Approximate solution $x'$
- Residual $\|Ax' - b\|$
- Residual is also called algebraic distance
- Algorithms that seek to reduce the residual are called “minimum residual” algorithms
Properties of the Algebraic Distance

- Each row in a linear equation can be multiplied by an arbitrary number
  \[ a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \]
  is the same as
  \[ c(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n) = cb_1 \]

However given an approximate solution \( \tilde{x} \)

\[ a_{11}\tilde{x}_1 + a_{12}\tilde{x}_2 + a_{13}\tilde{x}_3 + \cdots + a_{1n}\tilde{x}_n - b_1 \]

is not the same as

\[ c(a_{11}\tilde{x}_1 + a_{12}\tilde{x}_2 + a_{13}\tilde{x}_3 + \cdots + a_{1n}\tilde{x}_n - b_1) \]
Scaling

- Try to avoid anyone equation being overly represented.
- Scale each equation
  - Scale by largest coefficient so that it becomes 1
    \[ a_{i1}/a_{11} \]
  - Scale so that sum of coefficients is 1
    \[ a_{11}^2 + a_{12}^2 + \ldots + a_{1n}^2 = 1 \]
- Scaling also has the benefit of avoiding round-off.
Weighted Least Squares

- Multiplying an equation by a number will increase its **weight** or **influence** in the cost function.
- Not always a bad thing
  - May want to weight different equations differently
- How to select weights?
  - Number of observations
  - Reliability of measurement
    - Measured variances
- How good is the least squares solution? How “probable” are the parameter estimates?
- *Bring in notions of statistics*
Maximum Likelihood Parameter Estimation

- **likelihood** of the parameters given the data
- Least squares fit is a “maximum likelihood estimator”
- Assume
  - $y_i$ has a measurement error that is normally distributed around true $y \sim x$.
  - Assume errors are independent, and standard deviations $\sigma$ of all these normal distributions are the same.
  - Then probability that the data set and the model predictions are within $\Delta y$ the product of that of each other is

$$P \propto \prod_{i=1}^{N} \left\{ \exp \left[ -\frac{1}{2} \left( \frac{y_i - y(x_i)}{\sigma} \right)^2 \right] \right\} \Delta y$$

- Maximize likelihood that parameters are correct by maximizing $P$ with respect to model parameters.
Least Squares = MLE

• Since logarithm is a monotonic increasing function maximum of $\log P$ is the maximum of $P$
  $$\log P = \left[ \sum_{i=1}^{N} \frac{[y_i - y(x_i)]^2}{2\sigma^2} \right] - N \log \Delta y$$

• Maximizing $\log P$ is equivalent to maximizing the least squares criterion $(y_i - y(x_i))^2$ since the other terms are constants

• What to do when variances are not all the same?
  • Maximize the Mahalanobis distance
  $$\chi^2 \equiv \sum_{i=1}^{N} \left( \frac{y_i - y(x_i; a_1 \ldots a_M)}{\sigma_i} \right)^2$$

• Here the errors were just assumed to be in the measured $y$s
Errors in both coordinates

- Often in computer vision measurements are made in both images and a relationship between them must be deduced.
- Consider the line fit example again
  - Intuitively distances should be perpendicular to the line
- Perpendicular distance between point \((x_i, y_i)\) and a line \(y - a - bx = 0\) is
  \[
  d(x_i, y_i; a, b) = \left[ \frac{(y_i - a - bx_i)^2}{1 + b^2} \right]^{1/2}
  \]
- If \(x_i\) and \(y_i\) are distributed normally with standard deviations \(\sigma_{x_i}\) and \(\sigma_{y_i}\) we can show that \(c\) is
  \[
  \chi^2(a, b) = \sum_{i=1}^{N} \frac{(y_i - a - bx_i)^2}{\sigma_{y_i}^2 + b^2\sigma_{x_i}^2}
  \]
- Makes the cost function nonlinear in parameters
  - Nonlinear in \(b\). In general physical error functions are nonlinear.
Cost functions for *image* based data

- **Notation**
  - Measured value of a point $\mathbf{x}^\sim$
  - True value of a point $\mathbf{x}$
  - Estimated value of a point $\mathbf{x}^\wedge$
  - Transformation or model is denoted $H$
    - Model $\mathbf{y} = H(\mathbf{x})$ and $\mathbf{x} = H^{-1}(\mathbf{y})$

- **Symmetric error functions**
  - Case 1: Error only in one image
    - Could arise if we are imaging a calibration pattern with known coordinates and trying to determine camera calibration
  - Appropriate error function is
    
    Find $H^\wedge$ that minimizes $\Sigma_j d(\mathbf{x}^\sim_j, H^\wedge \mathbf{x}_j)^2$
Cost functions for image data

- Errors in both images
  Find $\mathbf{H}^\wedge$ that minimizes $\sum_j d(x_j^\sim, \mathbf{H}^\wedge x_j^\sim)^2 + d(\mathbf{H}^\wedge^{-1} x_j^\sim, x_j^\sim)^2$

- Reprojection error
  - Instead of determining parameters of a transformation that minimizes distances on erroneous data, find corrections to the wrong data and find the transformation that maps corrected data.
  - Get estimates $\mathbf{x}^\wedge$ and $\mathbf{x'}^\wedge$ such that
    $\sum_j d(x_j^\sim, x_j^\wedge)^2 + d(x_j'^\sim, x_j^\wedge)^2$ subject to $\mathbf{H}^\wedge x_j^\wedge = x_j'$
Cost Functions

Ideally there is a real cost being minimized—e.g., Dollars or distance travelled—then each equation makes sense.

Statistical measures

Review concepts of Metrics

Fig. 3.2. A comparison between symmetric transfer error (upper) and reprojection error (lower) when estimating a homography. The points $\mathbf{x}$ and $\mathbf{x}'$ are the measured (noisy) points. Under the estimated homography the points $\mathbf{x}'$ and $H\mathbf{x}$ do not correspond perfectly (and neither do the points $\mathbf{x}$ and $H^{-1}\mathbf{x}'$). However, the estimated points, $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, do correspond perfectly by the homography $\hat{\mathbf{x}}' = H\hat{\mathbf{x}}$. Using the notation $d(\mathbf{x}, \mathbf{y})$ for the Euclidean image distance between $\mathbf{x}$ and $\mathbf{y}$, the symmetric transfer error is $d(\mathbf{x}, H^{-1}\mathbf{x}')^2 + d(\mathbf{x}', H\mathbf{x})^2$; the reprojection error is $d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$. 
Optimization Techniques

• Different problem here
  – Given a set of locations $x_i$ where one has measured a fitness function $\chi^2/f(x)$ find a vector of parameters $-x-$ that minimizes it

• For the case where the function was linear we already have methods such as SVD to solve the linear system

• Here we are concerned with systems where the equation is not so simple
  – In particular $f$ may be a nonlinear function of parameters $x$

• Differential calculus provides us with ways of estimating extrema
  – The minimum $-\text{max}-$ of $f$ occurs at $\nabla f / \partial x$ or
  – $\nabla f$ is in the direction of increasing $f$ or
  – Given an interval $\nabla f$ has opposite signs at the boundary there must be a point inside where $\nabla f$ must be zero

• However calculus is local
  – So these methods can only guarantee a local extremum
Bisection methods

- Given a function $f$ at three points $a, b, c$ with $[a < b < c]$, and a way to evaluate $f$ at a new point
  - Given 2 initial guesses $f(a)$ and $f(b)$, if $f(a) > f(b)$ move in the direction $a$ to $b$ and choose a new parameter $c$.
  - Find a triplet $[a, b, c]$ so that $f(c) > f(b)$ and $f(a) > f(b)$
  - Choose a new point between $a$ and $b$ or $b$ and $c$
  - Repeat until the points $a, b$ and $c$ are sufficiently close

Figure 10.1.1. Successive bracketing of a minimum. The minimum is originally bracketed by points 1, 3, 2. The function is evaluated at 4, which replaces 2; then at 5, which replaces 1; then at 6, which replaces 4. The rule at each stage is to keep a center point that is lower than the two outside points. After the steps shown, the minimum is bracketed by points 5, 3, 6.
Figure 10.2.1. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1,2,3 on the given function (solid line). The function is evaluated at the parabola’s minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1,4,2. The minimum of this parabola is at 5, which is close to the minimum of the function.