

# 3D Computer Vision - WT 2005

## Nonlinear Least Squares Estimation

Martin Groher

Chair for Computer Aided Medical Procedures

November 30, 2005



- 1 Least-Squares Minimization
- 2 Nonlinear Estimation Methods
- 3 Parameter Estimation

## Definition

### *Def.: Least-Squares Minimization*

- Given a vector-valued function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and a vector of results (measurements)  $X = f(P)$
- We want to find the parameter vector  $P$  for this function
- Define a scalar-mapping on  $f$ :  $\|f(P) - X\|^2$  (squared norm)
- Minimizing the squared norm gives us  $P$ .

# Linear Least-Squares Minimization

*Def.: Linear Least-Squares Minimization*

- The function  $f$  is linear, i.e  $f(P) = AP$ ,  $A \in \mathbb{R}^{n \times m}$ ,  $m \geq n$ .
- Given a set of equations  $AP = X$
- We want to minimize  $\|AP - X\|^2$
- 

$$\sum_{i=1}^n \|b_i^{2D} - \Phi(b_i^{3D}, R, t)\|^2 + \sum_{j=1}^m \|s_j^{2D} - C(s_j^{2D}, \Phi(v_k^{3D}, R, t))\|^2$$

- $s_j^{2D} \in v_k^{2D}$ ,  $\Phi$ ,  $C$

## Solutions to LSQ

- SVD: Not only possible for  $\|AP\|$ , but also for  $\|AP - X\|$
- Normal equations:  $AP - X$  perpendicular to each of the columns of  $A \Rightarrow A^T(AP - X) = 0$ , or

$$(A^T A)P = A^T X$$

$A^T A$  is square and we can solve for  $P$

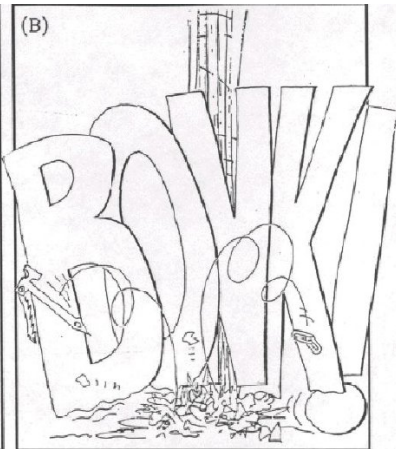
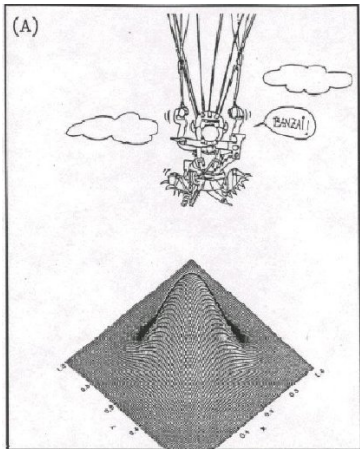
# LSQ Problems

- Estimate a 2D homography  $H$ :  
 $x' = Hx \Rightarrow x' \times Hx = 0 \Rightarrow \text{Minimize } \|x' \times Hx\|$
- Estimate a 2D/3D camera projection matrix  $\mathbf{P}$ :  
 $x_{2D} = \mathbf{P}x_{3D} \Rightarrow x_{2D} \times \mathbf{P}x_{3D} = 0 \Rightarrow \text{Minimize } \|x_{2D} \times \mathbf{P}x_{3D}\|$
- Estimate the fundamental matrix  $F$ :  
 $x'Fx = 0 \Rightarrow \text{Minimize } \|x'Fx\|$
- Estimate the trifocal tensor  $T$ :  
Minimize  $x^k(x'^i x''^l T_k^{33} - x''^l T_k^{i3} - x'^i T_k^{3l} + T_k^{il}) = 0$

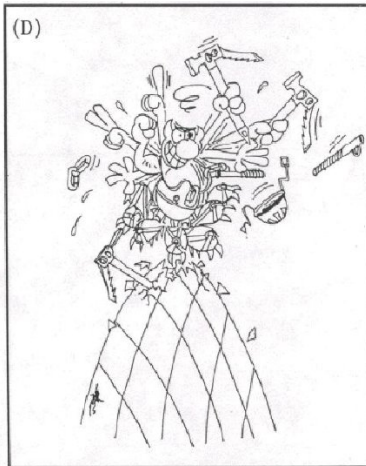
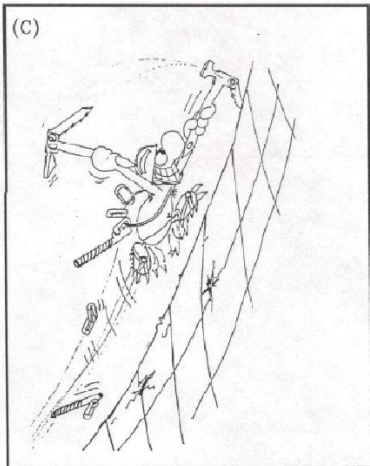
## Algebraic vs Geometric Error

- With linear least-squares we can just minimize the algebraic error  $\|AP - X\|$ , or  $\|AP\|$
- Geometric error cannot be modeled since  $f(P)$  is not linear any more!
- $\Rightarrow$  Move from linear to non-linear minimization
- Separate error (cost) function and method

# Nonlinear Estimation Methods



# Nonlinear Estimation Methods



## Definition

- Problem: Given a scalar-valued function  $g$ , find the *parameter vector*  $P$  such that  $g(P)$  is **minimal**

Nonlinear Estimation Methods are Minimization Methods!

- $P$  is not one single argument but a vector of variables (multi-variate case)!
- **Cost Function**  $g(P)$  is scalar-valued, so  $g(p_1, \dots, p_2) = X$ , with  $X \in \mathbb{R}$

## Different Estimation Methods

- Newton
- Gauss-Newton
- Gradient Descent
- Merging all: Levenberg-Marquardt

# Newton Iteration

- Assume we have an initial guess of the parameter vector  $P_0$  (close to minimum)
- Approximate  $g$  with the Taylor-Expansion

$$\begin{aligned}
 g(P_0 + \Delta) &= g(P_0) + \frac{\partial g}{\partial P}(P_0)\Delta + \Delta^T \frac{\partial^2 g}{\partial^2 P}(P_0)\Delta/2 + \dots \\
 &= g + g_P\Delta + \Delta^T g_{PP}\Delta/2 + \dots \\
 &\quad \text{all evaluated at } P_0
 \end{aligned}$$

- Forget about third and following derivatives, such that

$$g(P_0 + \Delta) = g + g_P\Delta + \Delta^T g_{PP}\Delta/2$$

- Find  $\Delta$  such that  $g(P_0 + \Delta)$  becomes minimal



# Newton Iteration

- Minimize  $g(P_0 + \Delta)$  with respect to  $\Delta$
- Differentiate  $g(P_0 + \Delta)$  **with respect to**  $\Delta$  and set it 0:

$$\begin{aligned}\frac{\partial g}{\partial \Delta} = g_P + g_{PP}\Delta &= 0 \Rightarrow \\ g_{PP}\Delta &= -g_P\end{aligned}$$

- The Newton iteration uses the gradient information ( $g_P$ ) of the cost function to move towards the minimum!

## Newton Iteration: Summary

- We want to minimize a function  $g$  with respect to a parameter vector  $P$
- We build gradient and hessian of the function  $g$  w.r.t.  $P \Rightarrow g_P, g_{PP}$
- We start with an initial estimate of  $P$ ,  $P_0$ , near the minimum of  $g$
- In each iteration we increment  $P_i$  with a  $\Delta$  to get closer to the minimum of  $g$
- $\Delta$  is computed by solving  $g_{PP}\Delta = -g_P$

## Moving to Least-Squares: Gauss-Newton Iteration

- We know now how to minimize a scalar-valued function  $g$
- How do we “minimize” a **vector-valued** function  
 $\epsilon(P) = (\epsilon_1(P), \dots, \epsilon_m(P))^T$ ?
- Think of a mapping that makes vector-valued to scalar-valued function - squared norm:

$$g(P) = \frac{1}{2} \|\epsilon(P)\|^2 = \epsilon(P)^T \epsilon(P) / 2$$

Nonlinear Estimation Methods are Minimization Methods!

Nonlinear Estimation Methods solve Least Squares Problems



## From Newton to Gauss-Newton

- If  $g(P) = \epsilon(P)^T \epsilon(P)/2$ , the gradient and hessian of  $g$  become

$$\begin{aligned} g_P &= \epsilon_P^T \epsilon \\ g_{PP} &= \epsilon_P^T \epsilon_P + \epsilon_{PP}^T \epsilon \end{aligned}$$

- Assumption: Either  $\epsilon(P)$  is linear in  $P$ , or  $\epsilon_{PP}$  negligibly small  
 $\Rightarrow$

$$g_{PP} \approx \epsilon_P^T \epsilon_P$$

- Using Gauss-Newton iteration, we just have to determine the first derivative, the second is approximated by the first
- New update equation:

$$g_{PP} \Delta = -g_P \quad \rightarrow \quad \epsilon_P^T \epsilon_P \Delta = -\epsilon_P^T \epsilon$$

## Gradient Descent Iteration

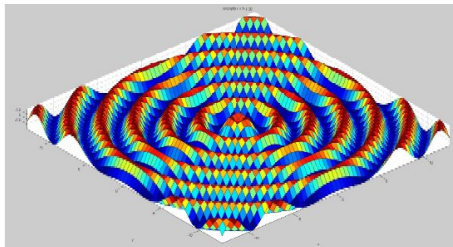
- Gradient Descent also uses the gradient information  $g_P$  to reach the minimum
- Where Newton uses the Hessian, gradient descent uses an arbitrary value  $\lambda$
- Update equation becomes

$$g_P^T \Delta = -g_P \quad \rightarrow \quad \lambda I \Delta = -g_P,$$

or, for vector-valued functions:

$$\epsilon_P^T \epsilon_P \Delta = -\epsilon_P^T \epsilon \quad \rightarrow \quad \lambda I \Delta = -\epsilon_P^T \epsilon$$

## Comparison: Gauss-Newton and Gradient Descent



- + **GN**:  $P_0$  near minimum: very fast (quadratic) convergence
- **GD**: sometimes slow convergence (zig-zagging)
- – **GN**:  $P_0$  far from minimum: no convergence for functions with local minima
- + **GD**:  $P_0$  far from minimum: still convergence

## Levenberg-Marquardt Iteration

- Take the benefits of Gradient Descent and Gauss-Newton and merge them
- Switch between **GD** and **GN** adaptively: If  $P_i$  is near minimum use **GN** else use **GD**
- How? Define  $\lambda$  adaptively in equation

$$(\epsilon_P^T \epsilon_P + \lambda I) \Delta = -\epsilon_P^T \epsilon$$

- If  $\lambda$  is big:  $\epsilon_P^T \epsilon_P$  is neglected compared to  $\lambda I$  and algorithm acts like **GD**
- If  $\lambda$  is small:  $\epsilon_P^T \epsilon_P$  dominates and algorithm behaves like **GN**

# LM Algorithm

- 1 Compute  $\|\epsilon(P_0)\|^2$ , set  $i \leftarrow 0$
- 2 Pick a modest value for  $\lambda$ , say  $\lambda = 0.001$
- 3 Solve  $(\epsilon_P^T \epsilon_P + \lambda I) \Delta = -\epsilon_P^T \epsilon$  for  $\Delta$  evaluated at  $P_i$
- 4 Evaluate new parameter vector  $P_{i+1} = P_i + \Delta$ 
  - If  $\|\epsilon(P_i + \Delta)\|^2 \geq \|\epsilon(P_i)\|^2$ , increase  $\Delta$  by a factor of 10, go back to 3.
  - If  $\|\epsilon(P_i + \Delta)\|^2 < \|\epsilon(P_i)\|^2$ , decrease  $\Delta$  by a factor of 10, update the trial solution  $P_{i+1} \leftarrow P_i + \Delta$ ,  $i \leftarrow i + 1$ , go back to 3.

## Comparing Gauss-Newton to Levenberg-Marquardt - Examples

①  $g(P) = x^2 + y^2, P = \begin{pmatrix} x \\ y \end{pmatrix}$

②  $g(P) = x^2 + y^2 - 10(\cos \sqrt{x^2 + y^2}), P = \begin{pmatrix} x \\ y \end{pmatrix}$

# From Least-Squares to Parameter Estimation

- Given a parameter vector  $P$  and Measurement Vector  $X$
- Functional Relationship  $f(P) = X$
- We want to minimize  $f(P) - X := \epsilon$
- Task: Solve for  $P$  in  $f$ , i.e. determine the parameter vector  $P$  that fits best the model  $f$  to the data  $X$

Nonlinear Estimation Methods are Minimization Methods!

Nonlinear Estimation Methods solve Least Squares Problems

Nonlinear Estimation Methods solve Parameter Estimation Problems

# From Least-Squares to Parameter Estimation

- *Be careful:* We do not want to perform a least-squares minimization on  $f(P)$ !
- We want to do a least-squares minimization on  $f(P) - X$ !
- Our cost/error function  $g$  is now  $\|f(P) - X\|^2$ !

## How to differentiate?

- For parameter estimation  $\epsilon = f(P) - X$
- $\epsilon_P = \frac{\partial(f(P)-X)}{\partial P} = \frac{\partial f(P)}{\partial P} = J$  with  $J$  the Jacobian matrix of  $f(P)$

# Different Cost Functions

- We can now use all the cost functions of the homography estimation:
  - Geometric Error  $\sum_i d(x'_i, Hx_i)^2$
  - Symmetric Transfer Error  $\sum_i (d(x'_i, Hx_i)^2 + d(H^{-1}x'_i, x_i)^2)$
  - Reprojection Error  $\sum_i (d(x'_i, \hat{x}'_i)^2 + d(\hat{x}_i, x_i)^2)$
- All other estimation problems in 3D Computer Vision can also be handled with the non-linear parameter estimation

# Literature

- Hartley / Zisserman: Multiple View Geometry, Appendices 5 and 6
- Press et al: Numerical Recipes in C, chapters 10 and 15
- Vachenauer: Höhere Mathematik 1