Fundamentals of Image Registration and Mosaicking

M. Zuliani[1]

[1]zuliani@ece.ucsb.edu
Vision Research Lab
Department of Electrical and Computer Engineering
University of California, Santa Barbara

October 30, 2007

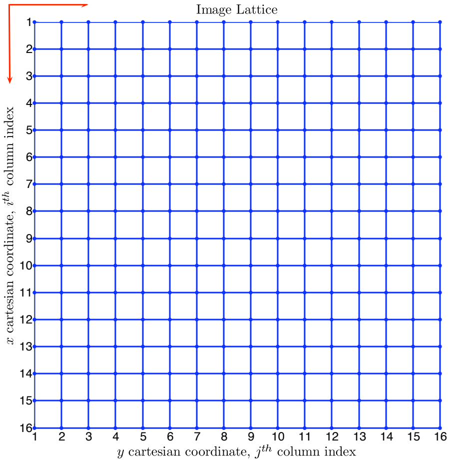
Outline

Outline

A Qualitative Definition

- Image registration:
  - establish a mapping between two or more images possibly taken:
    - at different times,
    - from different viewpoints,
    - under different lighting conditions,
    - and/or by different sensors
  - align the images with respect to a common coordinate system coherently with the three dimensional structure of the scene
- Image mosaicking: images are combined to provide a representation of the scene that is both geometrically and photometrically consistent.

The Image Lattice



Finite Differences Derivatives

- On a continuous domain: $\frac{df}{dx}(x) \stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$
- On a discrete lattice: $I_x(\boldsymbol{x}_{i,j}) \stackrel{\text{def}}{=} \frac{I(\boldsymbol{x}_{i+1,j})-I(\boldsymbol{x}_{i-1,j})}{2h}$

Smoothing Before Deriving

- Prewitt operator:

$$P_{x_1} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}}_{\text{first central difference}} \underbrace{\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}}_{\text{average smoothing}} = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sobel operator: changing the smoothing kernel to $\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$:

$$S_{x_1} = \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Transpose the kernels to derive along x_2

How Much Smoothing? The Issue of Scale.

- As noted by Lindeberg:

  > *[...] objects in the world may appear in different ways depending upon the scale of observation.*

- Thus we need different tools to describe them:
  - quantum mechanics
  - particle physics
  - thermodynamics
  - classical mechanics
  - general relativity
- Similarly with images:
  - construct derivative operators that depend continuously on a smoothing parameter
  - must be capable of capturing signal variations at different scales

Scale Space Signal Representation

- From image I to its scale space representation $\mathcal{L} = \{L_\sigma(\boldsymbol{x})\}_\sigma$
- Recipe: convolve the original image with a Gaussian kernel:

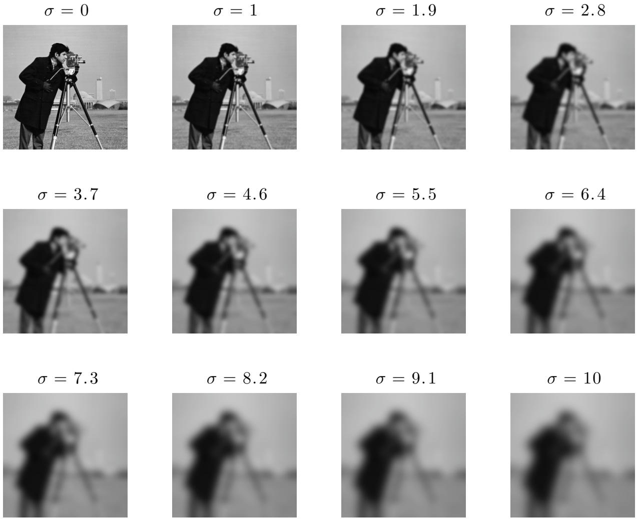
$$L_\sigma(\boldsymbol{x}) \stackrel{\text{def}}{=} (I * G_\sigma)(\boldsymbol{x})$$

- $G_\sigma(\boldsymbol{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{\|\boldsymbol{x}\|^2}{\sigma^2}}$
- Physical intuition: solution to the heat diffusion equation:

$$\begin{aligned} \frac{\partial L_{\sqrt{t}}}{\partial t}(\boldsymbol{x}) &= \frac{1}{2}\nabla^2_{\boldsymbol{x}} L_{\sqrt{t}}(\boldsymbol{x}) \\ L_0(\boldsymbol{x}) &= I(\boldsymbol{x}) \end{aligned}$$

for $t = \sigma^2$.

The Scale Space Representation of a Cameraman



Why Gaussian Kernels?

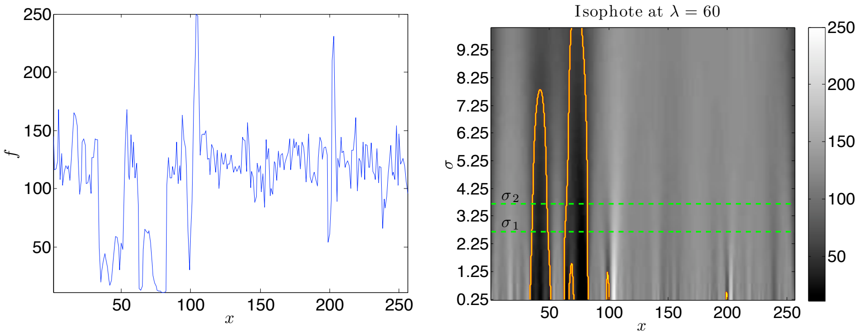
- Let's define:
  - Shift: $T_\Delta I(\boldsymbol{x}) \stackrel{\text{def}}{=} I(\boldsymbol{x} - \Delta)$
  - Rotation: $R_\theta I(\boldsymbol{x}) \stackrel{\text{def}}{=} I(R(\theta)\boldsymbol{x})$, where $R(\theta)$ is the 2×2 matrix that rotates a vector of an angle θ.
  - Scaling: $S_\alpha I(\boldsymbol{x}) \stackrel{\text{def}}{=} I(\alpha \boldsymbol{x})$.
- and the functional:

$$\begin{aligned} \mathcal{T}_{\sigma^2} : \mathcal{I} \times \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ (I, \boldsymbol{x}) &\longmapsto \mathcal{T}_{\sigma^2}[I](\boldsymbol{x}) \end{aligned}$$

Because They Satisfy the Scale Space Axioms!

- Linearity: $\mathcal{T}_{\sigma^2}[\alpha_1 I_1 + \alpha_1 I_2](\boldsymbol{x}) = \alpha_1 \mathcal{T}_{\sigma^2}[I_1](\boldsymbol{x}) + \alpha_2 \mathcal{T}_{\sigma^2}[I_2](\boldsymbol{x})$
- Shift invariance: $T_\Delta \mathcal{T}_{\sigma^2}[I] = \mathcal{T}_{\sigma^2}[T_\Delta I]$
- Scale invariance: There must exist a a strictly increasing continuous function ψ such that $\psi(0) = 0$ and $\lim_{s \to \infty} \psi(s) = \infty$ so that $S_\alpha \mathcal{T}_{\sigma^2}[I] = \mathcal{T}_{\psi(\sigma^2)}[S_\alpha I]$
- Rotation invariance: $R_\theta \mathcal{T}_{\sigma^2}[I] = \mathcal{T}_{\sigma^2}[R_\theta I]$.
- Semi-group structure: $\mathcal{T}_{\sigma_1^2}[\mathcal{T}_{\sigma_2^2}[I]] = \mathcal{T}_{\sigma_1^2 + \sigma_2^2}[I]$
- Causality constraints: The causality constraints can be divided in:
  - *Weak Causality Constraint*: Any scale space isophote $L_\sigma(\boldsymbol{x}) = \lambda$ is connected to a point $L_0(\boldsymbol{x}) = I(\boldsymbol{x}) = \lambda$.
  - *Strong Causality Constraint*: For every choice of $\sigma_2 > \sigma_1 \geq 0$ the intersection of an isophote within the domain $\left\{(\boldsymbol{x}, \sigma) \in \mathbb{R}^2 \times \mathbb{R}_+ : \boldsymbol{x} \in \mathbb{R}^2, \sigma \in (\sigma_1, \sigma_2]\right\}$ with the plane $\sigma = \sigma_1$ should not be empty.

The Scale Space Representation of a 1D Signal



Scale Space Differentiation Filters

- Fact:

$$\frac{\partial L_\sigma}{\partial x_i}(\boldsymbol{x}) = \frac{\partial}{\partial x_i}(I * G_\sigma)(\boldsymbol{x}) = \left(I * \frac{\partial G_\sigma}{\partial x_i} \right)(\boldsymbol{x})$$

- Image gradient:

$$\nabla L_\sigma = \left[\begin{array}{cc} \frac{\partial L_\sigma}{\partial x_1} & \frac{\partial L_\sigma}{\partial x_2} \end{array} \right]$$

- Magnitude of the gradient:

$$\|\nabla L_\sigma\| = \sqrt{\left(\frac{\partial L_\sigma}{\partial x_1}\right)^2 + \left(\frac{\partial L_\sigma}{\partial x_2}\right)^2}$$

Scale Space Gradient Magnitude of a Cameraman

$\sigma = 0$

$\sigma = 1$

$\sigma = 1.9$

$\sigma = 2.8$

$\sigma = 3.7$

$\sigma = 4.6$

$\sigma = 5.5$

$\sigma = 6.4$

$\sigma = 7.3$

$\sigma = 8.2$

$\sigma = 9.1$

$\sigma = 10$

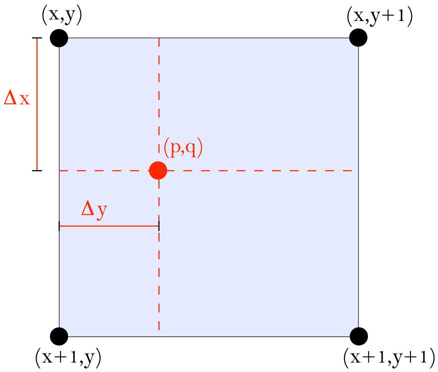
Why do we Need Interpolation?

- Because we may want to recover the intensity value at non integer pixel locations.
- Interpolation methods:
    - Nearest neighbour
    - Bilinear
    - Cubic
    - Lanczos
    - ...

Nearest Neighbor Interpolation

What is the value of f at $\begin{bmatrix} p & q \end{bmatrix}^T$?

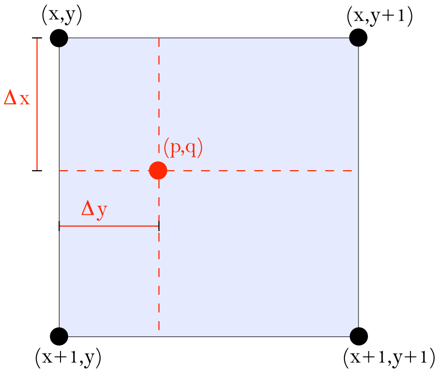
$$\hat{f}(p, q) = f(\text{round}(p), \text{round}(q))$$



Bilinear Interpolation: Notation

What is the value of f at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- $F_{0,0} \stackrel{\text{def}}{=} f(x, y)$
- $F_{1,0} \stackrel{\text{def}}{=} f(x+1, y)$
- $F_{0,1} \stackrel{\text{def}}{=} f(x, y+1)$
- $F_{1,1} \stackrel{\text{def}}{=} f(x+1, y+1)$
- $\Delta x \stackrel{\text{def}}{=} p - x$ and $\Delta y \stackrel{\text{def}}{=} q - y$



Bilinear Interpolation

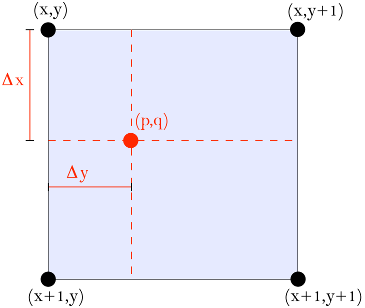
What is the value of f at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- Linear interpolation in the x direction:

$$\begin{aligned} f_y(\Delta x) &= (1-\Delta x)F_{0,0} + \Delta x F_{1,0} \\ f_{y+1}(\Delta x) &= (1-\Delta x)F_{0,1} + \Delta x F_{1,1} \end{aligned}$$

- Linear interpolation in the y direction:

$$\hat{f}(p,q) = (1-\Delta y) f_y + \Delta y f_{y+1}$$



$$\hat{f}(p,q) = (1-\Delta y)(1-\Delta x)F_{0,0} + (1-\Delta y)\Delta x F_{1,0} + \Delta y(1-\Delta x)F_{0,1} + \Delta y \Delta x F_{1,1}$$

Some Definitions

Definition

Two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ correspond between the reference and the sensed image: $\boldsymbol{x} \leftrightarrow \boldsymbol{x}'$ if they are the projection of the same point in the scene onto the camera image plane.

Definition

A mapping $\boldsymbol{T}_{\boldsymbol{\theta}}$ is a function:

$$\begin{aligned} \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x}) : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ (\boldsymbol{x}; \theta) &\mapsto \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x}) \end{aligned}$$

where θ is the vector of parameters of the transformation and $\boldsymbol{x}$ is the point to be mapped.

More Definitions

Definition

The overlapping area $\mathcal{O}$ in the reference image, according to the transformation $\boldsymbol{T}_{\boldsymbol{\theta}}$, is the set of points:

$$\mathcal{O} \stackrel{\text{def}}{=} \{\boldsymbol{x} \in \mathcal{D} : \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathcal{D}'\}$$

Definition

The overlapping area $\mathcal{O}'$ in the sensed image, according to the transformation $\boldsymbol{T}_{\boldsymbol{\theta}}$, is the set of points:

$$\mathcal{O}' \stackrel{\text{def}}{=} \{\boldsymbol{x}' \in \mathcal{D}' : \exists \boldsymbol{x} \in \mathcal{D} \text{ such that } \boldsymbol{x}' = \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x})\}$$

Image Registration: a Formal Definition

Definition (Registered Image Pair)

An image pair $(\boldsymbol{I}, \boldsymbol{I}')$ is registered if there exists a parameter vector $\hat{\theta}$ such that $\forall \boldsymbol{x} \in \mathcal{O}$ the points $\boldsymbol{x}$ and $\boldsymbol{x}' = \boldsymbol{T}_{\hat{\theta}}(\boldsymbol{x})$ correspond, i.e. $\boldsymbol{x} \leftrightarrow \boldsymbol{T}_{\hat{\theta}}(\boldsymbol{x})$.

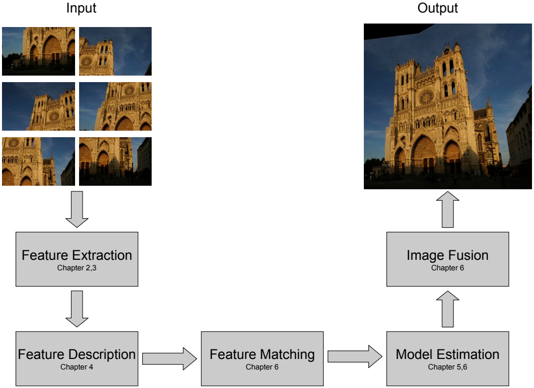
Image Registration: Overlapping

Overlapping area is displayed in green (image courtesy: prof. Chuck Stewart, RPI registration dataset).

Image Registration: Alignment

Outline

A Feature Based Registration System



Overview of the registration system modules (image courtesy of J. Nieuwenhuijse, copyright by New House Internet Services BV).

Translation

- Every point in the image is translated of the same amount

$$\boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{y}) = \boldsymbol{y} + \boldsymbol{\theta}$$

- $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T \in \mathbb{R}^2$
- The parameter vector contains the displacements in the y_1 and y_2 directions.

Rotation, Scale and Translation (RST)

- Every point in the image is is subject to a rotation, to a scaling and to a translation
- The anchor point $\boldsymbol{x}$ specifies the point about which the coordinate system rotates and translates

$$\boldsymbol{T}_{\boldsymbol{\theta},\boldsymbol{x}}(\boldsymbol{y}) = \boldsymbol{x} + \underbrace{\begin{bmatrix} \theta_3 & -\theta_4 \\ \theta_4 & \theta_3 \end{bmatrix}}_{sR}(\boldsymbol{y} - \boldsymbol{x}) + \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{\boldsymbol{t}}$$

- $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T \in \mathbb{R}^4$
- The components θ_3, θ_4 describe the rotation and the scaling and θ_1 and θ_2 encode the translation

Affine

- Every point in the image undergoes an affine transformation
- $\boldsymbol{x}$ is the anchor point

$$\boldsymbol{T}_{\boldsymbol{\theta},\boldsymbol{x}}(\boldsymbol{y}) = \boldsymbol{x} + \underbrace{\begin{bmatrix} \theta_3 & \theta_5 \\ \theta_4 & \theta_6 \end{bmatrix}}_{A}(\boldsymbol{y} - \boldsymbol{x}) + \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{\boldsymbol{t}}$$

- $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \dots & \theta_6 \end{bmatrix}^T \in \mathbb{R}^6$

Homography - I

- Describes how a planar surface transforms when imaged through pin-hole cameras that have a different position and orientation in space.
- An homography is a linear transformation in the projective space $\mathbb{P}^2$.
- From Euclidean space to projective space:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2 \mapsto \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \lambda \end{bmatrix} \in \mathbb{P}^2$$

- From projective space to Euclidean space

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \in \mathbb{P}^2 \mapsto \begin{bmatrix} \frac{p_1}{p_3} \\ \frac{p_2}{p_3} \end{bmatrix} \in \mathbb{R}^2$$

Homography - II

- Two points $\boldsymbol{p}$ and $\boldsymbol{p}'$ in the projective space are related according to a (planar) homography if:

$$\boldsymbol{p}' \sim \underbrace{\begin{bmatrix} \theta_1 & \theta_4 & \theta_7 \\ \theta_2 & \theta_5 & \theta_8 \\ \theta_3 & \theta_6 & \theta_9 \end{bmatrix}}_{H} \boldsymbol{p}$$

- In the Euclidean space an homography is represented via the non linear relation:

$$\boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{y}) = \begin{bmatrix} \frac{\theta_1 y_1 + \theta_4 y_2 + \theta_7}{\theta_3 y_1 + \theta_6 y_2 + \theta_9} \\ \frac{\theta_2 y_1 + \theta_5 y_2 + \theta_8}{\theta_3 y_1 + \theta_6 y_2 + \theta_9} \end{bmatrix}$$

- To fix the 9^{th} degree of freedom of the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^9$ set its norm to 1: $\|\boldsymbol{\theta}\| = 1$.

Preliminaries - I

- An example of an area based method
- Intuition: register in order to maximize the statistical knowledge regarding image *I* given image *I*'

Definition (Mutual Information)

The mutual information $\mathcal{I}(x; y)$ for the random variables x and y is :

$$\mathcal{I}(x; y) \stackrel{\text{def}}{=} \mathcal{H}(x) - \mathcal{H}(x|y) = \mathcal{H}(y) - \mathcal{H}(y|x)$$

Preliminaries - II

Definition

The entropy $\mathcal{H}$ of a (discrete) random variable x that takes values over the alphabet $\mathcal{X}$ is:

$$\mathcal{H}(x) \stackrel{\text{def}}{=} -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

Definition

The conditional entropy $\mathcal{H}(x|y)$ is:

$$\mathcal{H}(x|y) \stackrel{\text{def}}{=} -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 p(x|y)$$

Formalization

- $\boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the transformation that establishes the mapping between the two images
- Goal: to determine the parameter $\widehat{\boldsymbol{\theta}}$ such that $I(\boldsymbol{x}) = I'(\boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x}))$ for every $\boldsymbol{x}$
- Solution: maximize the mutual information:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\text{argmax}}\, \mathcal{I}(I; I')$$

- Simpler to say than to realize...

Outline

Preliminaries

- $I(\boldsymbol{x})$ is the intensity of a single channel image at point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$
- Ω is a neighborhood about the point of interest $\boldsymbol{x}$
- The gradient matrix is defined as:

$$A(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \begin{bmatrix} I_{x_1}(\boldsymbol{y}_1) & I_{x_2}(\boldsymbol{y}_1) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N) & I_{x_2}(\boldsymbol{y}_N) \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_1) \\ \vdots \\ \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_N) \end{bmatrix}$$

- The gradient normal matrix is defined as:

$$A^T A \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i)^2 & \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) \\ \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) & \sum_{i=1}^N I_{x_2}(\boldsymbol{y}_i)^2 \end{bmatrix}$$

Preliminaries

- $I(\boldsymbol{x})$ is the intensity of a single channel image at point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$
- Ω is a neighborhood about the point of interest $\boldsymbol{x}$
- The gradient matrix is defined as:

$$A(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \begin{bmatrix} I_{x_1}(\boldsymbol{y}_1) & I_{x_2}(\boldsymbol{y}_1) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N) & I_{x_2}(\boldsymbol{y}_N) \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_1) \\ \vdots \\ \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_N) \end{bmatrix}$$

- The gradient normal matrix is defined as:

$$A^T A \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i)^2 & \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) \\ \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) & \sum_{i=1}^N I_{x_2}(\boldsymbol{y}_i)^2 \end{bmatrix}$$

Preliminaries

- $I(\boldsymbol{x})$ is the intensity of a single channel image at point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$
- Ω is a neighborhood about the point of interest $\boldsymbol{x}$
- The gradient matrix is defined as:

$$A(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \begin{bmatrix} I_{x_1}(\boldsymbol{y}_1) & I_{x_2}(\boldsymbol{y}_1) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N) & I_{x_2}(\boldsymbol{y}_N) \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_1) \\ \vdots \\ \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_N) \end{bmatrix}$$

- The gradient normal matrix is defined as:

$$A^T A \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i)^2 & \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) \\ \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) & \sum_{i=1}^N I_{x_2}(\boldsymbol{y}_i)^2 \end{bmatrix}$$

Preliminaries

- $I(\boldsymbol{x})$ is the intensity of a single channel image at point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$
- Ω is a neighborhood about the point of interest $\boldsymbol{x}$
- The gradient matrix is defined as:

$$A(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \begin{bmatrix} I_{x_1}(\boldsymbol{y}_1) & I_{x_2}(\boldsymbol{y}_1) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N) & I_{x_2}(\boldsymbol{y}_N) \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_1) \\ \vdots \\ \nabla_{\boldsymbol{x}} I(\boldsymbol{y}_N) \end{bmatrix}$$

- The gradient normal matrix is defined as:

$$A^T A \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i)^2 & \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) \\ \sum_{i=1}^N I_{x_1}(\boldsymbol{y}_i) I_{x_2}(\boldsymbol{y}_i) & \sum_{i=1}^N I_{x_2}(\boldsymbol{y}_i)^2 \end{bmatrix}$$

Shaking Things

- Consider: $A = \begin{bmatrix} 1.0000 & 2.0000 \\ 2.0000 & 4.0001 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 10 + \varepsilon \\ 20 \end{bmatrix}$.
- Solve $A\boldsymbol{x} = \boldsymbol{b}$. Easy? Not really:

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} = 10000 \begin{bmatrix} 4.0001 & -2.0000 \\ -2.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} 10 + \varepsilon \\ 20 \end{bmatrix} = 10000 \begin{bmatrix} 0.0010 + 4.0001\varepsilon \\ -2.0000\varepsilon \end{bmatrix}$$

- If $\varepsilon = 0$ then $\boldsymbol{x} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$
- If $\varepsilon = 0.01$ then $\boldsymbol{x} = \begin{bmatrix} 410.0100 \\ -200.0000 \end{bmatrix}$

Shaking Things

- Consider: $A = \begin{bmatrix} 1.0000 & 2.0000 \\ 2.0000 & 4.0001 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix}$.
- Solve $A\boldsymbol{x} = \boldsymbol{b}$. Easy? Not really:

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} = 10000 \begin{bmatrix} 4.0001 & -2.0000 \\ -2.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix} = 10000 \begin{bmatrix} 0.0010 + 4.0001\varepsilon \\ -2.0000\varepsilon \end{bmatrix}$$

- If $\varepsilon = 0$ then $\boldsymbol{x} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$
- If $\varepsilon = 0.01$ then $\boldsymbol{x} = \begin{bmatrix} 410.0100 \\ -200.0000 \end{bmatrix}$

Shaking Things

- Consider: $A = \begin{bmatrix} 1.0000 & 2.0000 \\ 2.0000 & 4.0001 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix}$.
- Solve $A\boldsymbol{x} = \boldsymbol{b}$. Easy? Not really:

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} = 10000 \begin{bmatrix} 4.0001 & -2.0000 \\ -2.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix} =$$

$$10000 \begin{bmatrix} 0.0010 + 4.0001\varepsilon \\ -2.0000\varepsilon \end{bmatrix}$$

- If $\varepsilon = 0$ then $\boldsymbol{x} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$
- If $\varepsilon = 0.01$ then $\boldsymbol{x} = \begin{bmatrix} 410.0100 \\ -200.0000 \end{bmatrix}$

Shaking Things

- Consider: $A = \begin{bmatrix} 1.0000 & 2.0000 \\ 2.0000 & 4.0001 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix}$.
- Solve $A\boldsymbol{x} = \boldsymbol{b}$. Easy? Not really:

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} = 10000 \begin{bmatrix} 4.0001 & -2.0000 \\ -2.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} 10+\varepsilon \\ 20 \end{bmatrix} =$$

$$10000 \begin{bmatrix} 0.0010 + 4.0001\varepsilon \\ -2.0000\varepsilon \end{bmatrix}$$

- If $\varepsilon = 0$ then $\boldsymbol{x} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$
- If $\varepsilon = 0.01$ then $\boldsymbol{x} = \begin{bmatrix} 410.0100 \\ -200.0000 \end{bmatrix}$

Differential Condition Number

- The solution of a system of equations is a mapping from the input data $\boldsymbol{b} \in \mathbb{R}^n$ to the solution or output $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{b}) \in \mathbb{R}^m$
- If a small change in $\boldsymbol{b}$ produces a large change in $\boldsymbol{x}(\boldsymbol{b})$ then $\boldsymbol{x}$ is ill-conditioned at $\boldsymbol{b}$

Definition

The local or differential condition number is:

$$K = K(\boldsymbol{x}, \boldsymbol{b}) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \sup_{\|\Delta \boldsymbol{b}\| \leq \delta} \frac{\|\boldsymbol{x}(\boldsymbol{b} + \Delta \boldsymbol{b}) - \boldsymbol{x}(\boldsymbol{b})\|}{\|\Delta \boldsymbol{b}\|}$$

Theorem

For a linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$ *we have* $K = K(\boldsymbol{x}, \boldsymbol{b}) = \|A^\dagger\|$

Differential Condition Number Measuring Shaking

- In the previous example $A = \begin{bmatrix} 1.0000 & 2.0000 \\ 2.0000 & 4.0001 \end{bmatrix}$
- The Frobenius norm of A^{-1} is:

$$\sqrt{\sum_{i,j} |A_{ij}^{-1}|^2} = \sqrt{\sum \sigma(A^{-1})^2} \approx 5 \cdot 10^4$$

- Big if compared to the entries and to the size of A

Short Baseline Correspondences, a.k.a. Optical Flow

- $I = I(\cdot, t)$ is a single channel image sequence parameterized in the time variable t
- A point of interest has time dependent coordinates $\boldsymbol{x} = \boldsymbol{x}(t)$
- The optical flow problem is to discover the time evolution of $\boldsymbol{x}$
- Assumption: constant intensity: $I(\boldsymbol{x}(t), t) = I(\boldsymbol{x}(t) + d\boldsymbol{x}, t + dt) = c$
- Taylor expansions (neglecting higher order terms) yields:

$$I_{x_1}(\boldsymbol{x}, t)\, dx_1 + I_{x_2}(\boldsymbol{x}, t)\, dx_2 + I_t(\boldsymbol{x}, t)\, dt = 0$$

- In matrix form:

$$\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\, dt$$

Short Baseline Correspondences, a.k.a. Optical Flow

- $I = I(\cdot, t)$ is a single channel image sequence parameterized in the time variable t
- A point of interest has time dependent coordinates $\boldsymbol{x} = \boldsymbol{x}(t)$
- The optical flow problem is to discover the time evolution of $\boldsymbol{x}$
- Assumption: constant intensity: $I(\boldsymbol{x}(t), t) = I(\boldsymbol{x}(t) + d\boldsymbol{x}, t + dt) = c$
- Taylor expansions (neglecting higher order terms) yields:

$$I_{x_1}(\boldsymbol{x}, t)\ dx_1 + I_{x_2}(\boldsymbol{x}, t)\ dx_2 + I_t(\boldsymbol{x}, t)\ dt = 0$$

- In matrix form:

$$\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\ dt$$

Short Baseline Correspondences, a.k.a. Optical Flow

- $I = I(\cdot, t)$ is a single channel image sequence parameterized in the time variable t
- A point of interest has time dependent coordinates $\boldsymbol{x} = \boldsymbol{x}(t)$
- The optical flow problem is to discover the time evolution of $\boldsymbol{x}$
- Assumption: constant intensity: $I(\boldsymbol{x}(t), t) = I(\boldsymbol{x}(t) + d\boldsymbol{x}, t + dt) = c$
- Taylor expansions (neglecting higher order terms) yields:

$$I_{x_1}(\boldsymbol{x}, t)\; dx_1 + I_{x_2}(\boldsymbol{x}, t)\; dx_2 + I_t(\boldsymbol{x}, t)\; dt = 0$$

- In matrix form:

$$\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\; dt$$

Short Baseline Correspondences, a.k.a. Optical Flow

- $I = I(\cdot, t)$ is a single channel image sequence parameterized in the time variable t
- A point of interest has time dependent coordinates $\boldsymbol{x} = \boldsymbol{x}(t)$
- The optical flow problem is to discover the time evolution of $\boldsymbol{x}$
- Assumption: constant intensity: $I(\boldsymbol{x}(t), t) = I(\boldsymbol{x}(t) + d\boldsymbol{x}, t + dt) = c$
- Taylor expansions (neglecting higher order terms) yields:

$$I_{x_1}(\boldsymbol{x}, t)\ dx_1 + I_{x_2}(\boldsymbol{x}, t)\ dx_2 + I_t(\boldsymbol{x}, t)\ dt = 0$$

- In matrix form:

$$\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\ dt$$

Optical Flow: Solving for the Displacement

- Goal: estimate $d\boldsymbol{x} = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix}^T$, i.e. the optical flow vector
- Problem: $\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\, dt$ is one equation in two unknowns
- Solution: assume that dx_1 and dx_2 are constant in a region Ω about $\boldsymbol{x}$.
- Hence (letting $dt = 1$):

$$\begin{bmatrix} I_{x_1}(\boldsymbol{y}_1, t) & I_{x_2}(\boldsymbol{y}_1, t) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N, t) & I_{x_2}(\boldsymbol{y}_N, t) \end{bmatrix} d\boldsymbol{x} = - \begin{bmatrix} I_t(\boldsymbol{y}_1, t) \\ \vdots \\ I_t(\boldsymbol{y}_N, t) \end{bmatrix}$$

- Guess what? An overdetermined linear system of equations!

Optical Flow: Solving for the Displacement

- Goal: estimate $d\boldsymbol{x} = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix}^T$, i.e. the optical flow vector
- Problem: $\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\, dt$ is one equation in two unknowns
- Solution: assume that dx_1 and dx_2 are constant in a region Ω about $\boldsymbol{x}$.
- Hence (letting $dt = 1$):

$$\begin{bmatrix} I_{x_1}(\boldsymbol{y}_1, t) & I_{x_2}(\boldsymbol{y}_1, t) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N, t) & I_{x_2}(\boldsymbol{y}_N, t) \end{bmatrix} d\boldsymbol{x} = - \begin{bmatrix} I_t(\boldsymbol{y}_1, t) \\ \vdots \\ I_t(\boldsymbol{y}_N, t) \end{bmatrix}$$

- Guess what? An overdetermined linear system of equations!

Optical Flow: Solving for the Displacement

- Goal: estimate $d\boldsymbol{x} = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix}^T$, i.e. the optical flow vector
- Problem: $\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\, dt$ is one equation in two unknowns
- Solution: assume that dx_1 and dx_2 are constant in a region Ω about $\boldsymbol{x}$.
- Hence (letting $dt = 1$):

$$\begin{bmatrix} I_{x_1}(\boldsymbol{y}_1, t) & I_{x_2}(\boldsymbol{y}_1, t) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N, t) & I_{x_2}(\boldsymbol{y}_N, t) \end{bmatrix} d\boldsymbol{x} = - \begin{bmatrix} I_t(\boldsymbol{y}_1, t) \\ \vdots \\ I_t(\boldsymbol{y}_N, t) \end{bmatrix}$$

- Guess what? An overdetermined linear system of equations!

Optical Flow: Solving for the Displacement

- Goal: estimate $d\boldsymbol{x} = \begin{bmatrix} dx_1 & dx_2 \end{bmatrix}^T$, i.e. the optical flow vector
- Problem: $\begin{bmatrix} I_{x_1}(\boldsymbol{x}, t) & I_{x_2}(\boldsymbol{x}, t) \end{bmatrix} d\boldsymbol{x} = -I_t(\boldsymbol{x}, t)\, dt$ is one equation in two unknowns
- Solution: assume that dx_1 and dx_2 are constant in a region Ω about $\boldsymbol{x}$.
- Hence (letting $dt = 1$):

$$\begin{bmatrix} I_{x_1}(\boldsymbol{y}_1, t) & I_{x_2}(\boldsymbol{y}_1, t) \\ \vdots & \vdots \\ I_{x_1}(\boldsymbol{y}_N, t) & I_{x_2}(\boldsymbol{y}_N, t) \end{bmatrix} d\boldsymbol{x} = - \begin{bmatrix} I_t(\boldsymbol{y}_1, t) \\ \vdots \\ I_t(\boldsymbol{y}_N, t) \end{bmatrix}$$

- Guess what? An overdetermined linear system of equations!

Optical Flow: Least Square Solution

- More compactly:

$$A(\Omega(\boldsymbol{x}))d\boldsymbol{x} = \boldsymbol{\eta}$$

where $\boldsymbol{\eta} = -\left[\begin{array}{ccc} I_t(\boldsymbol{y}_1, t) & \ldots & I_t(\boldsymbol{y}_N, t) \end{array}\right]^T$.

- Least squares solution recipe:
  - multiply both sides by A^T to obtain a square system
  - multiply both members by $(A^T A)^{-1}$ to get:

$$d\boldsymbol{x}_{computed} = (A^T A)^{-1} A^T \eta = A^{\dagger} \eta$$

- A major problem: some points give better estimates of the true optical flow than others.

Optical Flow: Least Square Solution

- More compactly:

$$A(\Omega(\boldsymbol{x}))d\boldsymbol{x} = \boldsymbol{\eta}$$

where $\boldsymbol{\eta} = -\left[\begin{array}{ccc} I_t(\boldsymbol{y}_1, t) & \dots & I_t(\boldsymbol{y}_N, t) \end{array}\right]^T$.

- Least squares solution recipe:
  - multiply both sides by A^T to obtain a square system
  - multiply both members by $(A^T A)^{-1}$ to get:

$$d\boldsymbol{x}_{computed} = (A^T A)^{-1} A^T \eta = A^{\dagger} \eta$$

- A major problem: some points give better estimates of the true optical flow than others.

Optical Flow: Least Square Solution

- More compactly:

$$A(\Omega(\boldsymbol{x}))d\boldsymbol{x} = \boldsymbol{\eta}$$

where $\boldsymbol{\eta} = -\begin{bmatrix} I_t(\boldsymbol{y}_1, t) & \dots & I_t(\boldsymbol{y}_N, t) \end{bmatrix}^T$.

- Least squares solution recipe:
  - multiply both sides by A^T to obtain a square system
  - multiply both members by $(A^T A)^{-1}$ to get:

$$d\boldsymbol{x}_{computed} = (A^T A)^{-1} A^T \boldsymbol{\eta} = A^{\dagger} \boldsymbol{\eta}$$

- A major problem: some points give better estimates of the true optical flow than others.

Optical Flow: Least Square Solution

- More compactly:

$$A(\Omega(\boldsymbol{x}))d\boldsymbol{x} = \boldsymbol{\eta}$$

where $\boldsymbol{\eta} = -\begin{bmatrix} I_t(\boldsymbol{y}_1, t) & \dots & I_t(\boldsymbol{y}_N, t) \end{bmatrix}^T$.

- Least squares solution recipe:
  - multiply both sides by A^T to obtain a square system
  - multiply both members by $(A^T A)^{-1}$ to get:

$$d\boldsymbol{x}_{computed} = (A^T A)^{-1} A^T \boldsymbol{\eta} = A^{\dagger} \boldsymbol{\eta}$$

- A major problem: some points give better estimates of the true optical flow than others.

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\mathbf{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then η represents the noise itself
- Error in the optical flow estimate: $\mathbf{e} \stackrel{\text{def}}{=} d\mathbf{x}_{exact} - d\mathbf{x}_{computed}$
- Then:

$$\|\mathbf{e}\| = \|d\mathbf{x}_{exact} - d\mathbf{x}_{computed}\| = \|0 - A^{\dagger}\eta\| = \|A^{\dagger}\eta\| \leq \|A^{\dagger}\| \, \|\eta\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\mathbf{x}, \eta) = \|A^{\dagger}\|$

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\boldsymbol{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then η represents the noise itself
- Error in the optical flow estimate: $\boldsymbol{e} \stackrel{\text{def}}{=} d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}$
- Then:

$$\|\boldsymbol{e}\| = \|d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}\| = \|0 - A^{\dagger}\eta\| = \|A^{\dagger}\eta\| \leq \|A^{\dagger}\| \, \|\eta\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\boldsymbol{x}, \eta) = \|A^{\dagger}\|$

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\boldsymbol{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then η represents the noise itself
- Error in the optical flow estimate: $\boldsymbol{e} \stackrel{\text{def}}{=} d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}$
- Then:

$$\|\boldsymbol{e}\| = \|d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}\| = \|0 - A^{\dagger}\eta\| = \|A^{\dagger}\eta\| \leq \|A^{\dagger}\| \, \|\eta\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\boldsymbol{x}, \eta) = \|A^{\dagger}\|$

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\boldsymbol{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then $\boldsymbol{\eta}$ represents the noise itself
- Error in the optical flow estimate: $\boldsymbol{e} \stackrel{\text{def}}{=} d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}$
- Then:

$$\|\boldsymbol{e}\| = \|d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}\| = \|0 - A^{\dagger}\boldsymbol{\eta}\| = \|A^{\dagger}\boldsymbol{\eta}\| \leq \|A^{\dagger}\| \, \|\boldsymbol{\eta}\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\boldsymbol{x}, \eta) = \|A^{\dagger}\|$

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\mathbf{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then $\boldsymbol{\eta}$ represents the noise itself
- Error in the optical flow estimate: $\mathbf{e} \stackrel{\text{def}}{=} d\mathbf{x}_{exact} - d\mathbf{x}_{computed}$
- Then:

$$\|\mathbf{e}\| = \|d\mathbf{x}_{exact} - d\mathbf{x}_{computed}\| = \|0 - A^{\dagger}\boldsymbol{\eta}\| = \|A^{\dagger}\boldsymbol{\eta}\| \leq \|A^{\dagger}\| \, \|\boldsymbol{\eta}\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\mathbf{x}, \eta) = \|A^{\dagger}\|$

Optical Flow: A Thought Experiment

- Ansatz: the scene is static therefore the true optical flow is zero: $d\boldsymbol{x}_{exact} = 0$
- Suppose the images vary only by additive noise. Then $\boldsymbol{\eta}$ represents the noise itself
- Error in the optical flow estimate: $\boldsymbol{e} \stackrel{\text{def}}{=} d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}$
- Then:

$$\|\boldsymbol{e}\| = \|d\boldsymbol{x}_{exact} - d\boldsymbol{x}_{computed}\| = \|0 - A^{\dagger}\boldsymbol{\eta}\| = \|A^{\dagger}\boldsymbol{\eta}\| \leq \|A^{\dagger}\| \, \|\boldsymbol{\eta}\|$$

- $\|A^{\dagger}\|$ controls the error multiplication factor
- But we also saw that: $K = K(\boldsymbol{x}, \boldsymbol{\eta}) = \|A^{\dagger}\|$

Wide Baseline Correspondences: Estimating Local Transformations

- Consider two corresponding neighborhoods: $\Omega(\boldsymbol{x})$ and $\Omega'(\boldsymbol{x}')$
- Define the cost function:

$$C_{\boldsymbol{T}}(\theta) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{\boldsymbol{y} \in \Omega(\boldsymbol{x})} w(\boldsymbol{y} - \boldsymbol{x}) \| \boldsymbol{I}(\boldsymbol{y}) - \boldsymbol{I}'(\boldsymbol{T}_{\theta,\boldsymbol{x}}(\boldsymbol{y})) \|^2$$

- Goal: estimate the parameter vector that minimizes $C_{\boldsymbol{T}}(\theta)$, i.e. :

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^p} C_{\boldsymbol{T}}(\theta)$$

Wide Baseline Correspondences: Estimating Local Transformations

- Consider two corresponding neighborhoods: $\Omega(\boldsymbol{x})$ and $\Omega'(\boldsymbol{x}')$
- Define the cost function:

$$C_{\boldsymbol{T}}(\theta) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{\boldsymbol{y} \in \Omega(\boldsymbol{x})} w(\boldsymbol{y} - \boldsymbol{x}) \| \boldsymbol{I}(\boldsymbol{y}) - \boldsymbol{I}'(\boldsymbol{T}_{\theta,\boldsymbol{x}}(\boldsymbol{y})) \|^2$$

- Goal: estimate the parameter vector that minimizes $C_{\boldsymbol{T}}(\theta)$, i.e. :

$$\hat{\theta} = \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\operatorname{argmin}} \, C_{\boldsymbol{T}}(\theta)$$

The Right Question (and Hopefully the Right Answer)

- Which points allow to estimate θ reliably?
- Those points such that small amounts of noise will not cause the estimate $\hat{\theta}$ to be inaccurate
- Modeling the effect of noise:

$$\boldsymbol{I}'(\boldsymbol{T}_{\theta+\Delta\theta,\boldsymbol{x}}(\boldsymbol{y})) = \boldsymbol{I}(\boldsymbol{y}) + \eta$$

- Small amounts of η should not produce large perturbations $\Delta\theta$

Definition (Differential Condition Number for Point Neighborhoods)

The condition number associated with the point neighborhood $\Omega(\boldsymbol{x})$ with respect to $\boldsymbol{T}_{\theta,\boldsymbol{x}}$ is:

$$K_{\boldsymbol{T}_{\theta,\boldsymbol{x}}}(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \sup_{\|\boldsymbol{\eta}\| \le \delta} \frac{\|\Delta\theta\|}{\|\boldsymbol{\eta}\|}$$

The Right Question (and Hopefully the Right Answer)

- Which points allow to estimate θ reliably?
- Those points such that small amounts of noise will not cause the estimate $\hat{\theta}$ to be inaccurate
- Modeling the effect of noise:

$$I'(\boldsymbol{T}_{\theta+\Delta\theta,\boldsymbol{x}}(\boldsymbol{y})) = I(\boldsymbol{y}) + \eta$$

- Small amounts of η should not produce large perturbations $\Delta\theta$

Definition (Differential Condition Number for Point Neighborhoods)

The condition number associated with the point neighborhood $\Omega(\boldsymbol{x})$ with respect to $\boldsymbol{T}_{\theta,\boldsymbol{x}}$ is:

$$K_{\boldsymbol{T}_{\theta,\boldsymbol{x}}}(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \sup_{\|\eta\| \leq \delta} \frac{\|\Delta\theta\|}{\|\eta\|}$$

The Right Question (and Hopefully the Right Answer)

- Which points allow to estimate θ reliably?
- Those points such that small amounts of noise will not cause the estimate $\hat{\theta}$ to be inaccurate
- Modeling the effect of noise:

$$\boldsymbol{I}'(\boldsymbol{T}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta},\boldsymbol{x}}(\boldsymbol{y})) = \boldsymbol{I}(\boldsymbol{y}) + \boldsymbol{\eta}$$

- Small amounts of η should not produce large perturbations $\Delta\theta$

Definition (Differential Condition Number for Point Neighborhoods)

The condition number associated with the point neighborhood $\Omega(\boldsymbol{x})$ with respect to $\boldsymbol{T}_{\theta,\boldsymbol{x}}$ is:

$$K_{\boldsymbol{T}_{\theta,\boldsymbol{x}}}(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \sup_{\|\boldsymbol{\eta}\| \leq \delta} \frac{\|\Delta\theta\|}{\|\boldsymbol{\eta}\|}$$

The Right Question (and Hopefully the Right Answer)

- Which points allow to estimate θ reliably?
- Those points such that small amounts of noise will not cause the estimate $\hat{\theta}$ to be inaccurate
- Modeling the effect of noise:

$$\boldsymbol{I}'(\boldsymbol{T}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta},\boldsymbol{x}}(\boldsymbol{y})) = \boldsymbol{I}(\boldsymbol{y}) + \boldsymbol{\eta}$$

- Small amounts of $\boldsymbol{\eta}$ should not produce large perturbations $\Delta\boldsymbol{\theta}$

Definition (Differential Condition Number for Point Neighborhoods)

The condition number associated with the point neighborhood $\Omega(\boldsymbol{x})$ with respect to $\boldsymbol{T}_{\boldsymbol{\theta},\boldsymbol{x}}$ is:

$$K_{\boldsymbol{T}_{\boldsymbol{\theta},\boldsymbol{x}}}(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \lim_{\delta \to 0} \sup_{\|\boldsymbol{\eta}\| \leq \delta} \frac{\|\Delta\boldsymbol{\theta}\|}{\|\boldsymbol{\eta}\|}$$

The Quantitative Answer

Theorem (Estimate of the Differential Condition Number for Point Neighborhoods)

The expression for the estimate of the condition number for the point neighborhood $\Omega(\boldsymbol{x})$ is:

$$\hat{K}_{\boldsymbol{T}_{\theta,\boldsymbol{x}}}(\Omega(\boldsymbol{x})) = \|A^{\dagger}\left(\Omega(\boldsymbol{x})\right)\|$$

where the matrix $A(\Omega(\boldsymbol{x}))$:

$$A\left(\Omega(\boldsymbol{x})\right) \stackrel{\text{def}}{=} \begin{bmatrix} A(\boldsymbol{y}_1) \\ \vdots \\ A(\boldsymbol{y}_N) \end{bmatrix} \in \mathbb{R}^{mN\times p}$$

is formed by the N sub-matrices:

$$A(\boldsymbol{y}_i) \stackrel{\text{def}}{=} w(\boldsymbol{y}_i - \boldsymbol{x}) J\boldsymbol{I}'(\boldsymbol{y}_i)\, J_\theta \boldsymbol{T}_{\theta,\boldsymbol{x}}(\boldsymbol{y}_i)$$

obtained from a set of N points that sample the neighborhood $\Omega(\boldsymbol{x})$

Standpoint Summary

- ”Good points”, a.k.a. corners, are related to the (spectral) properties of the generalized gradient matrix:

$$A(\Omega(\boldsymbol{x})) \stackrel{\text{def}}{=} \begin{bmatrix} A(\boldsymbol{y}_1) \\ \vdots \\ A(\boldsymbol{y}_N) \end{bmatrix} \in \mathbb{R}^{mN \times p}$$

where:

$$A(\boldsymbol{y}_i) = w(\boldsymbol{y}_i - \boldsymbol{x}) J_\theta \boldsymbol{I}(\boldsymbol{T}_{\overline{\theta},\boldsymbol{x}}(\boldsymbol{y}_i)) = w(\boldsymbol{y}_i - \boldsymbol{x}) J\boldsymbol{I}(\boldsymbol{y}_i)\, J_\theta \boldsymbol{T}_{\overline{\theta},\boldsymbol{x}}(\boldsymbol{y}_i)$$

Spectral Corner Detectors

Definition (Spectral Corner Detector)

A spectral corner detector is a functional that depends solely on the singular values of the generalized gradient matrix:

$$\begin{aligned} f : \mathcal{I} \times \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (\boldsymbol{I}, \boldsymbol{x}) &\mapsto f\left(\sigma(A(\Omega(\boldsymbol{x})))\right) \end{aligned}$$

Common Corner Detectors:

- Harris-Stephens: $f_{HS} = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 = \det(A^T A) - \alpha\,\mathrm{trace}(A^T A)^2$
- Rohr: $f_R = \sqrt{\lambda_1\lambda_2}$
- Noble-Förstner: $f_{NF} = \frac{\lambda_1\ \lambda_2}{\lambda_1+\lambda_2} = \frac{\det(A^T A)}{\mathrm{trace}(A^T A)}$
- Shi-Tomasi: $f_{ST} = \lambda_{min}$

Spectral Corner Detectors

Definition (Spectral Corner Detector)

A spectral corner detector is a functional that depends solely on the singular values of the generalized gradient matrix:

$$\begin{aligned} f : \mathcal{I} \times \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (\boldsymbol{I}, \boldsymbol{x}) &\mapsto f\left(\sigma(A(\Omega(\boldsymbol{x})))\right) \end{aligned}$$

Common Corner Detectors:

- Harris-Stephens:
  $f_{HS} = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 = \det(A^T A) - \alpha\, \mathrm{trace}(A^T A)^2$
- Rohr: $f_R = \sqrt{\lambda_1 \lambda_2}$
- Noble-Förstner: $f_{NF} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(A^T A)}{\mathrm{trace}(A^T A)}$
- Shi-Tomasi: $f_{ST} = \lambda_{min}$

Condition Number Corner Detectors

Definition (Condition Number Corner Detector)

A condition number corner detector is a spectral corner detector such that:

$$\begin{aligned} f : \mathcal{I} \times \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (\boldsymbol{I}, \boldsymbol{x}) &\mapsto \frac{1}{\|A^\dagger(\Omega(\boldsymbol{x}))\|^2_{S,2q}} \end{aligned}$$

Definition (Schatten Matrix q-norm)

The Schatten matrix q-norm is defined as:

$$\|A\|_{S,q} \stackrel{\text{def}}{=} \left(\sum_i \sigma_i(A)^q \right)^{\frac{1}{q}}$$

where $\sigma_i(A)$ is the i^{th} singular value of the matrix A.

Condition Number Corner Detectors

Definition (Condition Number Corner Detector)

A condition number corner detector is a spectral corner detector such that:

$$\begin{aligned} f : \mathcal{I} \times \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (\boldsymbol{I}, \boldsymbol{x}) &\mapsto \frac{1}{\|A^\dagger(\Omega(\boldsymbol{x}))\|_{S,2q}^2} \end{aligned}$$

Definition (Schatten Matrix q-norm)

The Schatten matrix q-norm is defined as:

$$\|A\|_{S,q} \stackrel{\text{def}}{=} \left(\sum_i \sigma_i(A)^q \right)^{\frac{1}{q}}$$

where $\sigma_i(A)$ is the i^{th} singular value of the matrix A.

Putting Everything Together

Theorem (Corner Detectors Equivalence Relations)

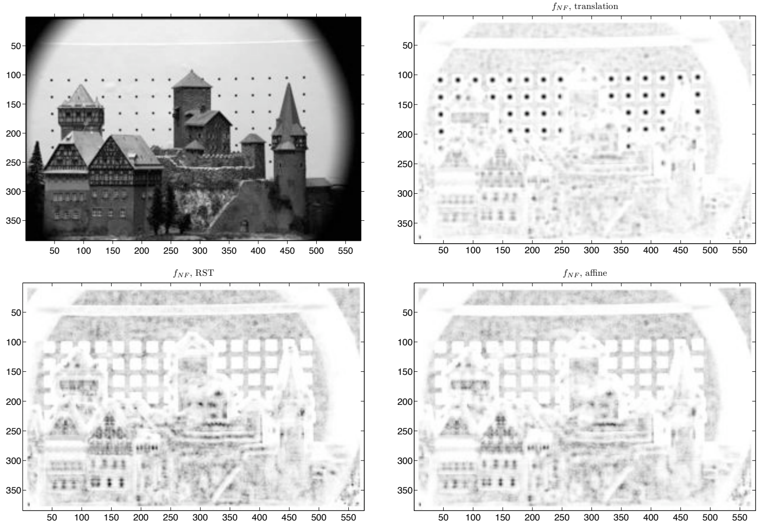
The following interesting relations hold among the spectral corner detectors when the transformation $\boldsymbol{T}_{\boldsymbol{\theta},\boldsymbol{x}}$ models a simple translation:

- *Generalized Rohr equivalence:* $\lim_{q\to 0} \sqrt[q]{p} f_{K,q} = f_R$
- *Generalized Noble-Förstner equivalence:* $f_{K,1} = f_{NF}$
- *Generalized Shi-Tomasi equivalence:* $f_{K,\infty} = f_{ST}$

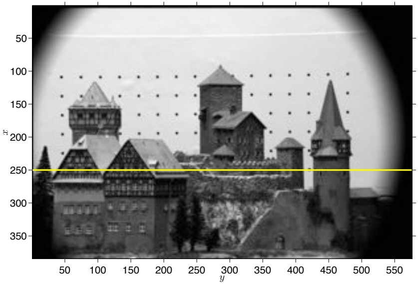
Theorem (Analytical Bounds)

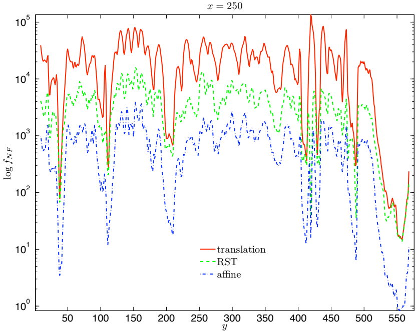
$$f_{K,q}^{Translation} \geq f_{K,q}^{RST} \geq f_{K,q}^{Affine}$$

Noble-Förstner Reponse for Different $\boldsymbol{T}_{\theta,\boldsymbol{x}}$



Noble-Förstner Reponse for Different $T_{\theta,x}$





Homework

Write a Matlab function to detect the corners in an arbitrary gray level image using the Noble-Förstner detector. The syntax of the function should be `[x y f] = compute_corners(I, sigma, r)`, where:

- `I` is the single channel input image.
- `sigma` is the standard deviation of the Gaussian differentiation filter in pixels
- `r` is the radius of the circular neighborhood $\Omega(\boldsymbol{x})$
- `x`, `y` is the position of the interest points
- `f` is the detector map, i.e. the reponse of the detector at each location of the image

Protecting Luca's Mental Health

A necessary (but not sufficient) condition to complete the assignment is that your function will satisfy the following testing protocol:

- The workspace will contain the image `I` and the variables of `sigma`, `r`
- The command `[x y f] = compute_corners(I, sigma, r);` will be issued
- The results will be evaluated superimposing the detected point on the original image and displaying the detector map:
- `figure`
- `imshow(I);`
- `hold on;`
- `plot(y, x, 'r+');`
- `figure`
- `imagesc(f); axis equal tight; colormap gray;`