Image Processing and Imaging

Image Restauration

Christof Kauba
Fachbereich Computerwissenschaften
Universität Salzburg

Wintersemester 2020/21

UNIVERSITÄT SALZBURG

PARIS LODRON UNIVERSITY SALZBURG

Outline

Image Restauration Introduction

- Improves image quality in case of an existing image distortion which should be removed
- Contrasting to image enhancement the aim is to restore a (virtual) original image
- Either the type of distortion is known or it has to be estimated
- Reasons for existing distortions:
  - Defocus, motion blur, noise (transmission errors, sensor noise, ...), defects in the optical system (Hubble), etc.

Deterministic methods: for images with low amount of noise and known distortion function

Stochastic methods: try to identify the best restauration with respect to some statistical error-criterion (e.g. least-squares criterion)

The better the distortion is known, the better it can be removed and the better is the resulting restauration. In many cases, the distortion needs to be estimated:

A priori estimation: distortion is known or is obtained before the restauration starts

A posteriori estimation: image analysis based on *interesting* pixels (e.g. edges, straight lines, homogeneous areas) and it is attempted to estimate / reconstruct their original properties.

Outline

Image Distortion

We assume a position invariant linear distortion and independent additive noise:

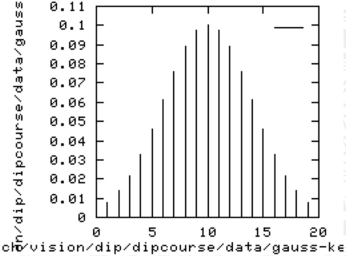
$$g(x, y) = h(x, y) * f(x, y) + v(x, y) \tag{1}$$

$v(x, y)$ ... noise
h ... distortion (position invariant)

Distorted signal can be represented in the DFT domain as follows (linearity and convolution theorem):

$$\hat{g}(u, v) = \hat{h}(u, v) \cdot \hat{f}(u, v) + \hat{v}(u, v)$$

smoothed image



Gauss kernel (1-D)

Figure: Example: Smoothing as image distortion

Outline

Outline

- Several alternatives how to determine (estimate, approximate) the image distortion present
- All the following techniques are approximative by nature → ideas are called “blind deconvolution”

Image Analysis

- In the distorted image we choose image regions with “obvious” image content, e.g. a sharp edge
- Corresponding image part is denoted $g_s(x, y)$
- Create an approximation $f_s^a(x, y)$ for this image part of the original image
- Selection of the region → noise should not affect the result too much
- We are able to compute the distortion function in the DFT domain for the region selected:

$$\hat{h}_s(u, v) = \frac{\hat{g}_s(u, v)}{\hat{f}_s^a(u, v)}$$

- Due to the assumed position invariance, the distortion function can be generalised to the entire image

Distortion Determination - Experimental

- Assume that the equipment used to take the distorted image is available (or at least the same or similar model)
- Capture an image similar to the one subject to restauration
- Try to generate a distortion highly similar to the one to model by systematic testing of the system configurations (e.g. different camera settings)
- Once identified, we take an image of an intensive point light source, to obtain the impulse response of the distortion
  - A distortion of the considered type is uniquely characterised by its impulse response

- DFT of an impulse is a constant A, thus we get:

$$\hat{h}(u, v) = \frac{\hat{g}(u, v)}{A}$$

Outline

Here we exploit knowledge about models of physical processes

Examples for simple distortions are:

Relative (uniform) motion between camera and object:

- $f(x, y)$ moves in a way such that $x_0(t)$ and $y_0(t)$ are the time-dependent motion components in x and y direction
- Entire image exposition is obtained by integrating the image function over the time-frame of the shutter being opened T:

$$g(x, y) = \int_0^T f(x - x_0(t), y - y_0(t))dt$$

$$\hat{g}(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y)e^{-2\pi i(ux+vy)}dxdy$$

$$\hat{g}(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f(x - x_0(t), y - y_0(t))dt\right] e^{-2\pi i(ux+vy)}dxdy$$

- Integration order can be swapped:

$$\hat{g}(u,v) = \int_0^T \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - x_0(t), y - y_0(t)) e^{-2\pi i (ux+vy)} dxdy \right] dt$$

- Expression between [] is the DFT of the shifted function $f(x - x_0(t), y - y_0(t))$
- Using the translation properties of the DFT and the independence between $\hat{f}(u,v)$ and t we obtain:

$$\hat{g}(u,v) = \int_0^T \hat{f}(u,v) e^{-2\pi i (ux_0(t)+vy_0(t))} dt = \hat{f}(u,v) \int_0^T e^{-2\pi i (ux_0(t)+vy_0(t))} dt$$

- Thus, we set $\hat{h}(u,v) = \int_0^T e^{-2\pi i (ux_0(t)+vy_0(t))} dt$

- Setting for example $x_0(t) = at/T$ and $y_0(t) = 0$ we get motion only in x-direction (e.g. taking pictures from a moving car)

- At time $t = T$ the image moved by distance a. We get

$$\hat{h}(u,v) = \int_0^T e^{-2\pi iuat/T} = \frac{T}{\pi ua}\sin(\pi ua)e^{-\pi iua}$$

- For two-dimensional motion (also $y_0(t) = bt/T$ as well) we get:

$$\hat{h}(u,v) = \frac{T}{\pi(ua+vb)}\sin(\pi(ua+vb))e^{-\pi i(ua+vb)}$$

Distortion Determination - Modeling (4)

Defocus:

$$\hat{h}(u,v) = \frac{J_1(ar)}{ar} \text{ mit } r^2 = u^2 + v^2$$
$$J_1(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{2k+1}}{k!(k+1)!}$$

J_1 ... Bessel function Order 1
a ... Extent of Defocus

Atmospheric Turbulence:

$$\hat{h}(u,v) = e^{-c(u^2+v^2)^{5/6}}$$

c is determined experimentally

Outline

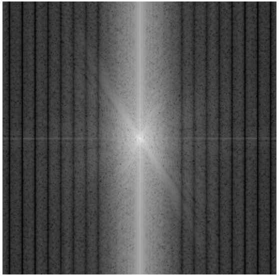
Inverse Filtering:

- Need to construct a restauration filter, exhibiting a transfer function invers to the distortion: $\hat{h}^{-1}(u, v)$
- The resulting procedure is denoted “Inverse Filtering”:

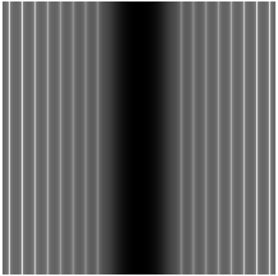
$$\hat{f}(u, v) = \hat{g}(u, v) \cdot \hat{h}^{-1}(u, v) - \hat{v}(u, v) \cdot \hat{h}^{-1}(u, v)$$

- If the noise contribution is not too high, restauration is identical to inverse convolution
- If the noise contribution is too high or $\hat{h}(u, v)$ is too small we result in a large value for $\hat{v}(u, v) \cdot \hat{h}^{-1}(u, v)$
  - Dominates the inverse filtering
- Result of inverse filtering with high noise contribution (see next slide)
- Smoothed image (originally given as `float` data type) has been casted to `char` data type, resulting in significant (quantisation) noise
- Result of inverse filtering is quite poor!

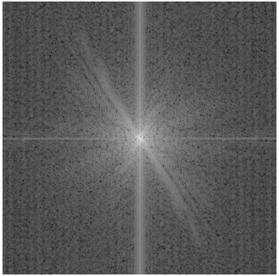
Distortion Removal - Inverse Filtering Example



DFT of the smoothed image



DFT of the Gauss kernels

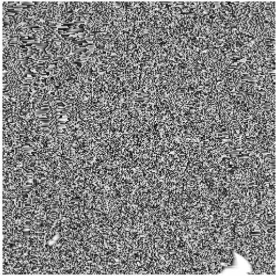


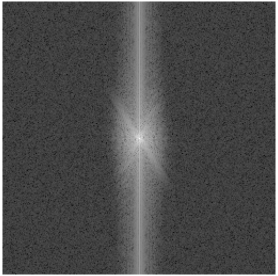
DFT after inverse filtering

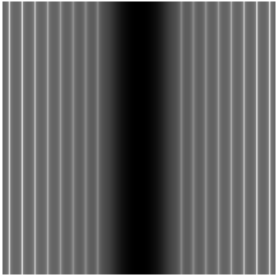
image after restauration

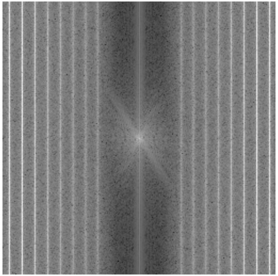
Distortion Removal - Inverse Filtering High Noise Example


smoothed image (char)


difference (with respect to float)


DFT of smoothed img (Byte)


DFT of Gauss kernel


DFT after inverse filtering

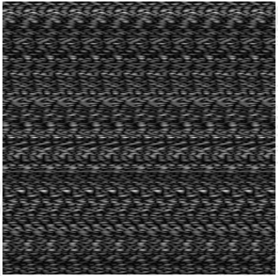


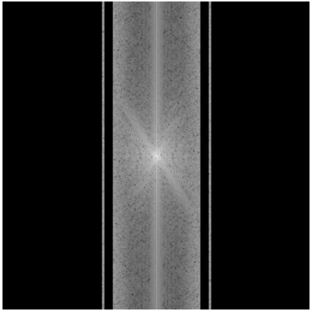
image after restauration

Pseudo-Inverse Filtering:

$$\hat{h}^{-1}(u,v) = \begin{cases} h^{-1}(u,v) \ \textit{if} \ |\hat{h}(u,v)| > T \\ 0 \ \textit{if} \ |\hat{h}(u,v)| \leq T \end{cases}$$

Case of $\hat{h}(u,v)$ being too small is "corrected" to prevent a large value for $\hat{v}(u,v) \cdot \hat{h}^{-1}(u,v)$

Still, large values for $\hat{v}(u,v)$ remain unsolved (noise contribution)



DFT after pseudo-inverse Filtering

Image after pseudo-inverse Filtering

Outline

Wiener Filtering:

- Additionally exploits a priori knowledge about the noise contribution
- Delivers an estimation of the non-distorted image $\bar{f}$ with minimal error $f(i,j) - \bar{f}(i,j)$ with respect to some determined metric

$$\hat{\bar{f}}(u,v) = \hat{h}_w(u,v) \cdot \hat{g}(u,v) \tag{2}$$

$$\hat{h}_w(u,v) = \frac{\hat{h}^*(u,v)}{|\hat{h}(u,v)|^2 + \frac{s_{xx}(u,v)}{s_{\eta\eta}(u,v)}} \tag{3}$$

- s_{xx} and $s_{\eta\eta}$ are the spectral densities of the noise and the original (non-distorted) image
  - Of course, these values are difficult to obtain
- To be able to conduct this type of filtering, information about the distortion and statistical knowledge about the noise are required

Distortion Removal - Wiener Filtering (2)

- Actual knowledge required for this process (e.g. computation of the spectral density of the noise) might be hard or impossible to obtain
- A "parameterised" Wiener filter can be employed:

$$\hat{h}_w^K(u,v) = \frac{\hat{h}^*(u,v)}{|\hat{h}(u,v)|^2 + K}$$

- In the process, K is optimised until the best result is reached
- Can be improved by using so-called "constrained least square" filters
- Intuitively this means that the optimisation of K is done with respect to some least squares criterion
  - E.g. to maximise image smoothness (or minimisation of image variations), expressed in terms of a gradient operator

Distortion Removal - Wiener Filtering Example

Figure: Wiener Filtering