Image Restoration

SCC0251/5830 – Image Processing

Moacir A. Ponti

ICMC/USP — São Carlos, SP, Brazil

2020

Agenda

1. Introduction
2. Noise
   - Sources and models of noise
   - Noise generation
   - Noise reduction
   - Bilateral filtering
3. Blur
   - Degradation functions
   - Inverse and pseudo-inverse filtering
   - Least squares filtering

Agenda

Obtaining better images

Problem — to improve the visual quality of the images

- **Enhancement** $\times$ **Restoration**

Obtaining better images

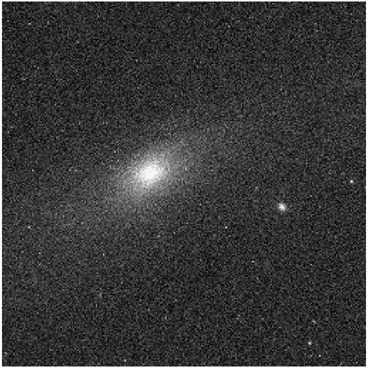
Problem — to improve the visual quality of the images

- **Enhancement** $\times$ **Restoration**
  - Enhancement: subjective method based on operations that supposedly improve image quality
  - Restoration: objective method based on prior knowledge about the image degradation model

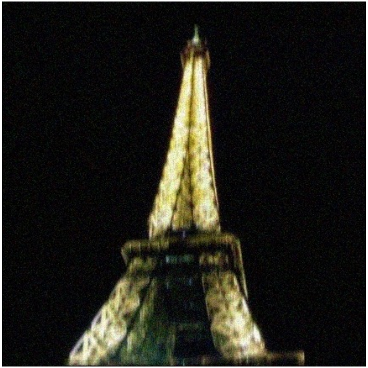
Degradation: blur

Degradation: motion blur

Degradation: noise



Degradation: blur and noise



Problem

$$g(\mathbf{x}) = \mathcal{N}\left\{f(\mathbf{x}) * h(\mathbf{x})\right\}$$

- g — observed (degraded) image
- f — ideal or original image
- $*$ — convolution
- h — degrading function
- $\mathcal{N}()$ — noise generation process

Problem

When the nature of the noise is "additive"

$$g(\mathsf{x}) = f(\mathsf{x}) * h(\mathsf{x}) + n(\mathsf{x})$$

- g — observed (degraded) image
- f — ideal or original image
- $*$ — convolution
- h — degrading function
- n — additive noise function

Problem

This equation tries to capture the idea of an imaging system

1. the image is capture via a system: microscope, telescope, camera lens — $f(\mathbf{x}) * h(\mathbf{x})$.

Problem

This equation tries to capture the idea of an imaging system

1. the image is capture via a system: microscope, telescope, camera lens — $f(\mathbf{x}) * h(\mathbf{x})$.
2. the electronic acquisition of the sensor generates additive noise — $[f(\mathbf{x}) * h(\mathbf{x})] + n(\mathbf{x})$.

Restoration algorithms aim to achieve a restored image $\hat{f}(\mathbf{x})$ that is as similar as possible to the original/ideal image $f(\mathbf{x})$.

Problem

This equation tries to capture the idea of an imaging system

1. the image is capture via a system: microscope, telescope, camera lens — $f(\mathbf{x}) * h(\mathbf{x})$.
2. the electronic acquisition of the sensor generates additive noise — $[f(\mathbf{x}) * h(\mathbf{x})] + n(\mathbf{x})$.

Restoration algorithms aim to achieve a restored image $\hat{f}(\mathbf{x})$ that is as similar as possible to the original/ideal image $f(\mathbf{x})$.

- In order to to that, we use knowledge about the *point spread function* and *noise*.

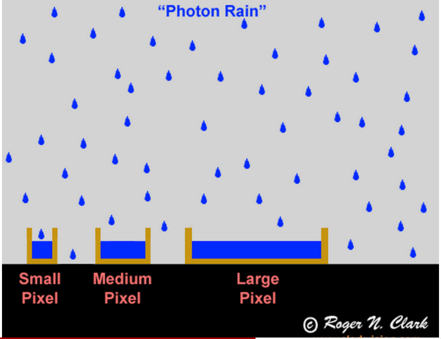
Sources of noise

Generally, the source defines the noise characteristic. Most images has noise that is accumulated through several acquisition steps

- **Photo counting**
- **Thermal**
- **Quantisation**
- **Transmission/display**

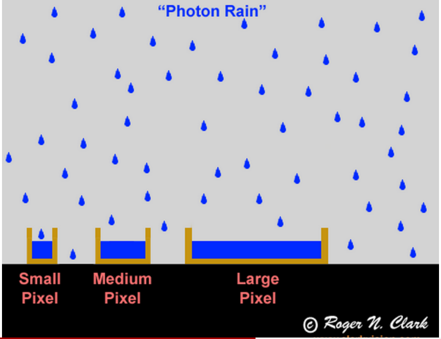
Sources of noise — photon counting

- **Photon counting**: light detection via a sensor is a statistical process, well modeled by a Poisson distribution.
- The precision of the measured signal is proportional to the mean of the signal (the amount of photons).



Sources of noise — photon counting

- **Photon counting**: light detection via a sensor is a statistical process, well modeled by a Poisson distribution.
- The precision of the measured signal is proportional to the mean of the signal (the amount of photons).
- The amount of noise can be approximated by the squared root of the number of photons.

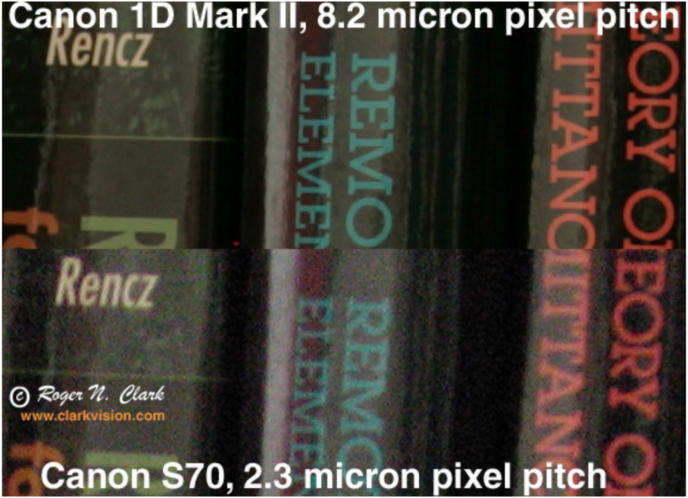


Photons	Noise	NR
9	3	1/3
100	10	1/10
900	30	1/30
10000	100	1/100
90000	300	1/300

Sources of noise — photon counting

- That is why two cameras with the same pixel quantities but different sensor sizes can result in different images.
- Below two images from the same maker, number of pixels, ISO parameter, aperture and shutter speed, but different sensors.

Sources of noise — photon counting



thanks to Roger Clark

http://www.clarkvision.com/articles/telephoto_reach/

Sources of noise — thermal

- When imaging under extreme focal distances (e.g. small objects imaged at close distance / large objects imaged from far away):
  - Smaller pixels allow to capture better fine details,

Sources of noise — thermal

- When imaging under extreme focal distances (e.g. small objects imaged at close distance / large objects imaged from far away):
  - Smaller pixels allow to capture better fine details,
  - Each pixel will have a lower amount of photons.
  - Therefore, a sharper image, but still

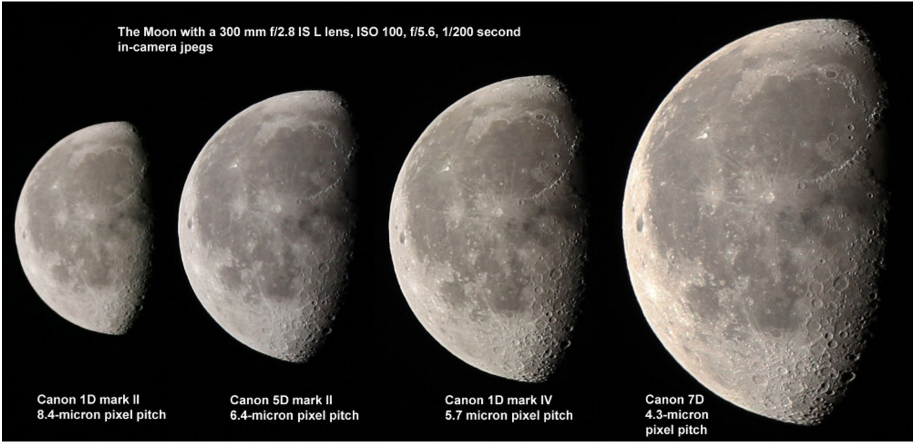
Sources of noise — thermal

- When imaging under extreme focal distances (e.g. small objects imaged at close distance / large objects imaged from far away):
  - Smaller pixels allow to capture better fine details,
  - Each pixel will have a lower amount of photons.
  - Therefore, a sharper image, but still **noisier**.

Sources of noise — thermal

- When imaging under extreme focal distances (e.g. small objects imaged at close distance / large objects imaged from far away):
  - Smaller pixels allow to capture better fine details,
  - Each pixel will have a lower amount of photons.
  - Therefore, a sharper image, but still **noisier**.
- Smaller pixels allow to observe more details, paying the cost of a lower signal-to-noise ratio per pixel.

Sources of noise — photon counting



thanks to Roger Clark `http://www.clarkvision.com/articles/telephoto_reach/`.

Sources of noise — photon counting

- Sparse images, with low exposure time, has noise characterised by Poisson distribution. Examples are:
    - Astronomic images
    - Microscopy images

Sources of noise — photon counting

- Sparse images, with low exposure time, has noise characterised by Poisson distribution. Examples are:
  - Astronomic images
  - Microscopy images
- Noise is **signal dependent** (correlated).

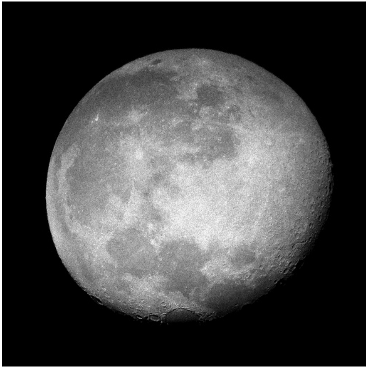
Sources of noise — photon counting

- Sparse images, with low exposure time, has noise characterised by Poisson distribution. Examples are:
  - Astronomic images
  - Microscopy images
- Noise is **signal dependent** (correlated).
- Its image formation is given by $g(\mathbf{x}) = \mathcal{P}\left\{f(\mathbf{x}) * h(\mathbf{x})\right\}$

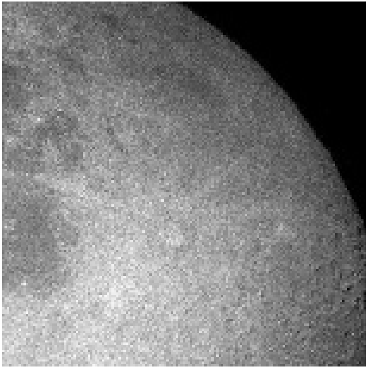
Sources of noise — photon counting

- Sparse images, with low exposure time, has noise characterised by Poisson distribution. Examples are:
  - Astronomic images
  - Microscopy images
- Noise is **signal dependent** (correlated).
- Its image formation is given by $g(\mathbf{x}) = \mathcal{P}\{f(\mathbf{x}) * h(\mathbf{x})\}$
- When imaging with good illumination conditions and adequate exposure, counting noise is often low and can be neglected.
  - This is because the Poisson distribution approaches the Normal distribution, i.e. $\mathcal{P}(\lambda) \sim \mathcal{N}(\lambda, \lambda)$, as $\lambda \to \infty$.

Sources of noise — photon counting



Sources of noise — photon counting

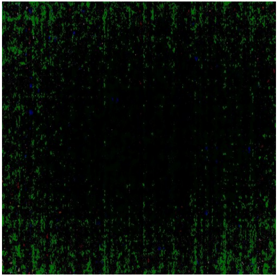


Sources of noise — thermal

- **Thermal**: electrons are generated when the photons are detected. Those will vary given the temperature of the sensor.
- Usually we assume this noise to be Gaussian (Normal) and additive, also called White noise.
  - This noise is independent of the signal.
  - Image formation is given by: $g(\mathbf{x}) = f(\mathbf{x}) * h(\mathbf{x}) + n(\mathbf{x})$

Sources of noise — thermal

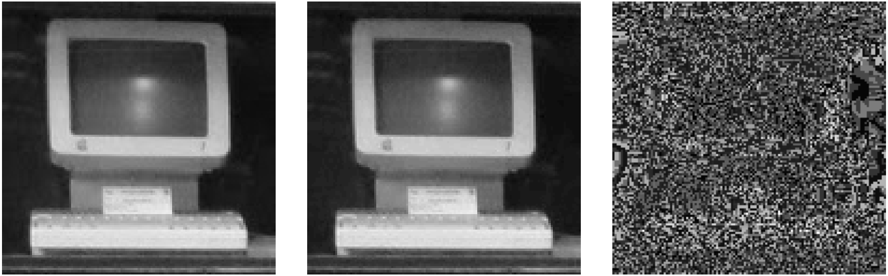
- A possible way to diminish thermal noise is via a Dark Frame capture, an image obtained without light acquisition.
- This image contains a map of the thermal noise. Although it varies with the temperature, it is usually stable after a period.
  - Dark Frame can then be subtracted from acquired images
  - Below: Dark Frames of CCDs from a telescope (left), and a cellphone camera (right), with normalised levels.



Sources of noise — quantisation

- **Quantisation**: noise caused by quantisation of pixels from continuous to unsigned int/char.
  - It often follows uniform distribution.
  - When quantisation level is low, the noise can become signal dependent and correlated to each region of the image (non-uniform).

Sources of noise — quantisation



(a) (b) (c)

(a) 256 level quantisation, (b) 64 level quantisation, (c) quantisation noise with 64 levels

Sources of noise — transmission/display

- Noise often caused by errors in some bits when storing or failure when transmitted.
- Resulting noise is referred to as “impulsive”, but also “salt and pepper”.
    - Can be caused by other processes then transmission/display
    - Affects a smaller number of pixels, but the ones affected are completely destroyed.

Sources of noise — transmission/display

Sources of noise — transmission/display

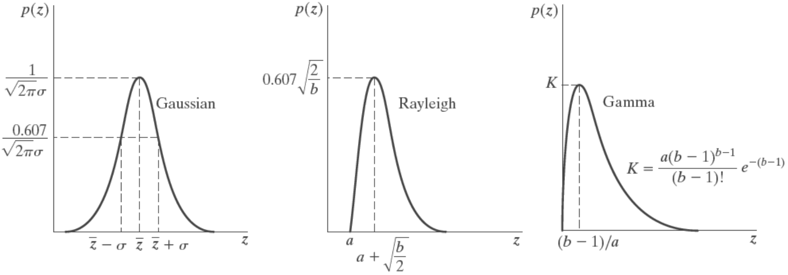
- The mathematical representation of the impulsive noise can seen as two "impulses" (or Dirac functions) in 0 (black) e 255 (white)
  - A random pixel has probability p of been affected by noise, usually $p/2$ for "salt" and $p/2$ for "pepper".

Agenda

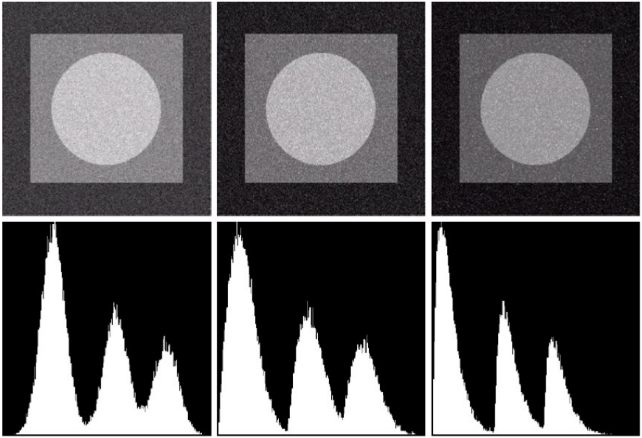
Noise generation

- It is possible to simulate noise in images using known distributions.
- Real noise is difficult to simulate, but by knowing the basic image formation system it is possible to obtain a good approximation.
- Implementation consists in generating random numbers and using probability density functions.

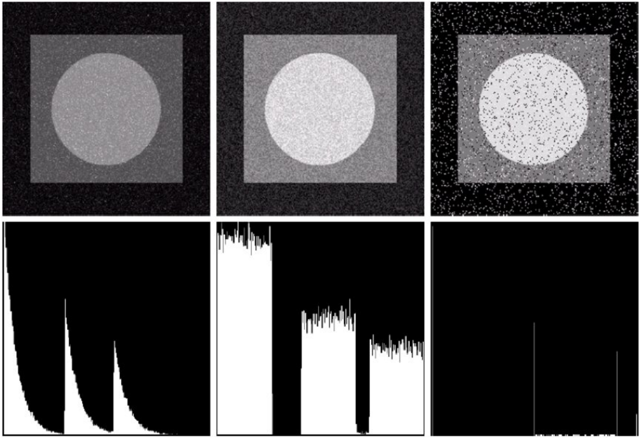
Noise generation



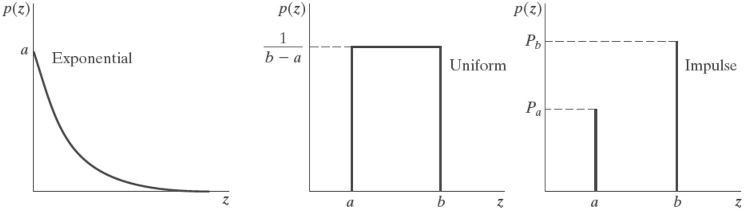
Noise generation



Noise generation



Noise generation



Agenda

Mean filtering

- Smooth out pixels using the contextual information (neighbours),
- Mean operators allow to **reduce the signal variance** and, therefore, noise.
- Variations of mean filtering: arithmetic, geometric, harmonic, weighted.

Mean filtering

- **Arithmetic**: increase the blur by creating a new value based on the average of neighbour pixels $S_{(\mathbf{x})}$, where $(\mathbf{x}) = (x, y)$.
- Neighbourhood is rectangular of size $m \times n$
- when $\lambda_{(s,t)} = 1$ for all s, t, then all pixels have the same weigh

$$\hat{f}(\mathbf{x}) = \frac{1}{nm} \sum_{(s,t) \in S_{\mathbf{x}}} \lambda_{s,t} \cdot g(s, t)$$

Mean filtering

- **Geometric**: can help preserving details when pixel differences are in the order of multiples of a given base (2, 10, etc.), i.e. it is logarithmic.

$$\hat{f}(\mathbf{x}) = \left[\prod_{(s,t) \in S_{\mathbf{x}}} \lambda_{s,t} \cdot g(s,t) \right]^{\frac{1}{nm}}$$

Mean filtering

- **Harmonic**: reduce the influence of outliers.
- This filter is adequate when there is additive noise mixed with salt noise (outlier)

$$\hat{f}(\mathbf{x}) = \frac{mn}{\sum_{(s,t)\in S_{\mathbf{x}}} \frac{1}{g(s,t)}}$$

Order statistic filters

- Given a series of observations of some random variable, the order statistics are obtained by sorting those observations in ascending order.
- In context of images, the observations are pixels in a neighbourhood.
- Result in non-linear filters such as
  - Median
  - Maximum, mininum
  - Mean point

Order statistic filters

- **Median**: widely used in image pre-processing
- Remove texture, preserve edges.
- Very effective to remove impulsive noise.
- The resulting pixel is the percentile 50 of a ordered sequence of numbers

$$\hat{f}(\mathbf{x}) = \text{median}_{(s,t)\in S_{\mathbf{x}}} \{g(s,t)\}$$

Order statistic filters

- **Max**: 100º percentile (maximum value)
- Can be used to locate bright points in the image

$$\hat{f}(\mathbf{x}) = \max_{(s,t)\in S_{\mathbf{x}}} \{g(s,t)\}$$

- **Min**: 0º percentile (minimum value)
- Can be used to locate dark points in the image

$$\hat{f}(\mathbf{x}) = \min_{(s,t)\in S_{\mathbf{x}}} \{g(s,t)\}$$

Order statistic filters

- **Mean point**: combines order statistics with mean
- Usually produces an effect similar to median, but often thickens the borders/edges.

$$\hat{f}(\mathrm{x}) = \frac{1}{2}\left[\max_{(s,t)\in S_{\mathrm{x}}}\{g(s,t)\} + \min_{(s,t)\in S_{\mathrm{x}}}\{g(s,t)\}\right]$$

Adaptive filtering

- Take into account local statistics.
- The objective is to allow smoother results mostly in flat regions (with less detail);
- Any filter can be developed in an adaptive fashion. For example:
  - Adaptive noise reduction using mean and local variance,
  - Adaptive noise reduction using median and local inter-quartile range (IQR).

Adaptive noise reduction using mean and variance

Considering a local region $S_{\mathbf{x}}$, the response of the adaptive filter needs:

1. $g(\mathbf{x})$: the value of noisy image at $\mathbf{x}$
2. σ_η^2 : the variance of noise in the image (global)
3. m_L : local mean of pixels in $S_{\mathbf{x}}$
4. σ_L^2 : local variance of pixels in $S_{\mathbf{x}}$

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(\mathbf{x}) - m_L\right]$$

Adaptive noise reduction using mean and variance

Considering a local region $S_{\mathbf{x}}$, the response of the adaptive filter needs:

1. $g(\mathbf{x})$: the value of noisy image at $\mathbf{x}$
2. σ_η^2 : the variance of noise in the image (global)
3. m_L : local mean of pixels in $S_{\mathbf{x}}$
4. σ_L^2 : local variance of pixels in $S_{\mathbf{x}}$

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(\mathbf{x}) - m_L\right]$$

- We need to estimate (or know — strong assumption) the noise variance
    - It is possible to estimate σ_η^2 measuring variance in a flat region of the image.

Adaptive noise reduction using mean and variance

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(\mathbf{x}) - m_L\right]$$

The filter behaves in each point as follows:

- if $\sigma_L^2 = 0$, then the response is $g(\mathbf{x})$,
- if $\sigma_L^2 \gg \sigma_\eta^2$, then it approaches $g(\mathbf{x})$,
- if $\sigma_L^2 \approx \sigma_\eta^2$, then the response is the local mean at region $S_\mathbf{x}$.

Adaptive noise reduction using mean and variance

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(\mathbf{x}) - m_L\right]$$

The filter behaves in each point as follows:

- if $\sigma_L^2 = 0$, then the response is $g(\mathbf{x})$,
- if $\sigma_L^2 \gg \sigma_\eta^2$, then it approaches $g(\mathbf{x})$,
- if $\sigma_L^2 \approx \sigma_\eta^2$, then the response is the local mean at region $S_\mathbf{x}$.

We need that $\sigma_\eta^2 \leq \sigma_L^2$

- if we observe $\sigma_\eta^2 > \sigma_L^2$, then the ratio between the variances must be defined as 1 to avoid spurious values.

Adaptive noise reduction using mean and variance

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(\mathbf{x}) - m_L\right]$$

The filter behaves in each point as follows:

- if $\sigma_L^2 = 0$, then the response is $g(\mathbf{x})$,
- if $\sigma_L^2 \gg \sigma_\eta^2$, then it approaches $g(\mathbf{x})$,
- if $\sigma_L^2 \approx \sigma_\eta^2$, then the response is the local mean at region $S_\mathbf{x}$.

We need that $\sigma_\eta^2 \leq \sigma_L^2$

- if we observe $\sigma_\eta^2 > \sigma_L^2$, then the ratio between the variances must be defined as 1 to avoid spurious values.
- this condition makes the filter non-linear.

Bilateral filtering

Noise reduction filter with edge preservation that uses the image content in order to avoid averaging across edges. Centered at a pixel **p**, it is given by:

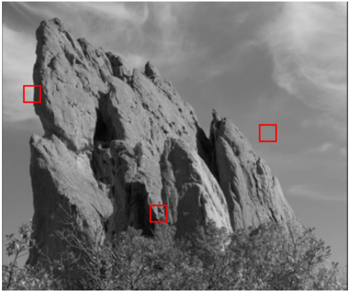
normalisation — A: not new — B: new!

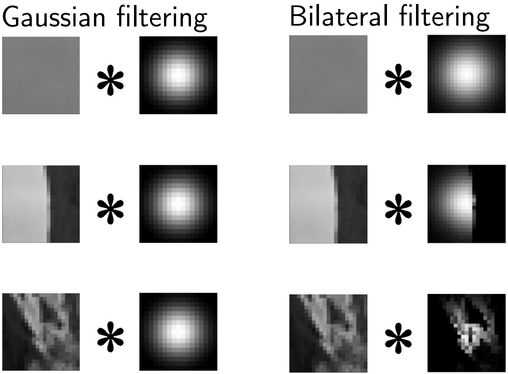
$$BF(g(\mathbf{p})) = \underbrace{\frac{1}{F_{\mathbf{p}}}}_{\text{normalisation}} \sum_{\mathbf{q}} \underbrace{G_{\sigma_s}(||\mathbf{p}-\mathbf{q}||)}_{\text{A: not new}} \underbrace{G_{\sigma_r}(||g_{\mathbf{p}} - g_{\mathbf{q}}||)}_{\text{B: new!}} \, g_{\mathbf{q}}$$

- term A defines the weight in space (difference in coordinates),
- term B controls the range weight (differences in intensities), avoiding filtering over edges.

OBS: removing the normalisation and the term B, we have a Gaussian filter.

Bilateral filtering





Bilateral filtering

$$BF(g(\mathbf{p})) = \frac{1}{F_{\mathbf{p}}} \sum_{\mathbf{q}} G_{\sigma_s}(||\mathbf{p} - \mathbf{q}||)\, G_{\sigma_r}(||g_{\mathbf{p}} - g_{\mathbf{q}}||)\, g_{\mathbf{q}}$$

- σ_s parameter for the size of neighbourhood, e.g. 2% of the image diagonal
- σ_r minimum amplitude to consider presence of an edge, e.g. mean of the image gradient

OBS: because each neighbourhood has a different filter, cannot be precomputed to use with FFT. Naive implementation is slow, but there are approximations with good quality/speed ratio.

Agenda

Problem

Assuming a noise-free scenario, the image formation model is given by:

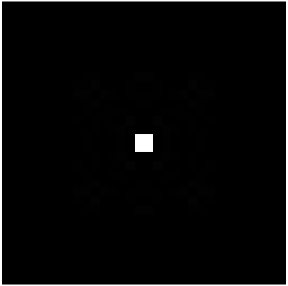
$$g(\mathbf{x}) = f(\mathbf{x}) * h(\mathbf{x})$$

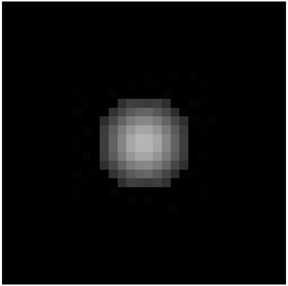
- g — degraded/observed image
- f — ideal or original image
- $*$ — convolution
- h — degradation function

Problem

Function $h(\mathbf{x})$ represents the **impulse response** of the imaging system

- in an image it models how the system responds when the input is a single point (or impulse)





Problem

Function $h(\mathbf{x})$ represents the **impulse response** of the imaging system

- in an image it models how the system responds when the input is a single point (or impulse)
- often called point spread function (PSF)

Degradation functions

- h are non-negative due to the physics of image formation,
- if the image is real (yes, there are complex images), PSF is also real,
- imperfections of the imaging system are modelled so that the energy of the signal is preserved:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)dxdy = 1$$

$$\sum_{\mathbf{x}=(0,0)}^{(N-1,M-1)} h(\mathbf{x}) = 1$$

Degradation functions

No blur

$$h(x,y) = \delta(x,y) = \begin{cases} 1, & \text{if } x,y = (0,0) \\ 0, & \text{other positions} \end{cases}$$

Degradation functions

No blur

$$h(x,y) = \delta(x,y) = \begin{cases} 1, & \text{if } x,y = (0,0) \\ 0, & \text{other positions} \end{cases}$$

Uniform blur

$$h(x,y;R) = \begin{cases} \frac{1}{\pi R^2}, & \text{if } \sqrt{x^2+y^2} \leq R^2, \\ 0, & \text{otherwise} \end{cases}$$

Degradation functions

No blur

$$h(x,y) = \delta(x,y) = \begin{cases} 1, & \text{if } x,y = (0,0) \\ 0, & \text{other positions} \end{cases}$$

Uniform blur

$$h(x,y;R) = \begin{cases} \frac{1}{\pi R^2}, & \text{if } \sqrt{x^2+y^2} \leq R^2, \\ 0, & \text{otherwise} \end{cases}$$

Motion blur

$$h(x,y;L,\phi) = \begin{cases} \frac{1}{L}, & \text{if} \sqrt{x^2+y^2} \leq \frac{L}{2} \text{ and } \frac{x}{y} = -\tan\phi, \\ 0, & \text{otherwise} \end{cases}$$

Problem

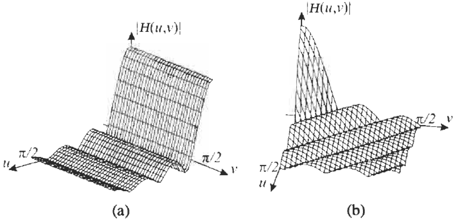


FIGURE 2 PSF of motion blur in the Fourier domain, showing $|H(u, v)|$, for (a) $L = 7.5$ and $\phi = 0$; (b) $L = 7.5$ and $\phi = \pi/4$

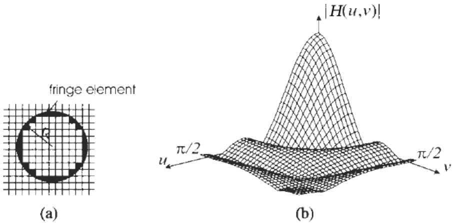


FIGURE 3 (a) Fringe elements of discrete out-of-focus blur that are calculated by integration; (b) PSF in the Fourier domain, showing $|H(u, v)|$, for $R = 2.5$.

Discrete degrading functions

Uniform blur

$$h(\mathbf{x}; R) = \begin{cases} \frac{1}{C} & \text{if } \sqrt{x_1^2 + x_2^2} \leq R^2, \\ 0 & \text{otherwise} \end{cases}$$

where C is a constant so that the sum of the coefficients is 1.

Discrete degrading functions

Uniform blur

$$h(\mathbf{x}; R) = \begin{cases} \frac{1}{C} & \text{if } \sqrt{x_1^2 + x_2^2} \leq R^2, \\ 0 & \text{otherwise} \end{cases}$$

where C is a constant so that the sum of the coefficients is 1.

Discrete degrading functions

Uniform blur

$$h(\mathbf{x}; R) = \begin{cases} \frac{1}{C} & \text{if } \sqrt{x_1^2 + x_2^2} \leq R^2, \\ 0 & \text{otherwise} \end{cases}$$

where C is a constant so that the sum of the coefficients is 1.

Motion blur

$$h(\mathbf{x}; L) = \begin{cases} \frac{1}{L} & \text{if } x_1 = 0, |x_2| \leq \lfloor \frac{L-1}{2} \rfloor \\ \frac{1}{2L} \left\{ (L-1) - 2\lfloor \frac{L-1}{2} \rfloor \right\} & \text{if } x_1 = 0, |x_2| = \lfloor \frac{L-1}{2} \rfloor \\ 0, & \text{otherwise} \end{cases}$$

Inverse filtering

We want to invert h, so that:

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) * h^{-1}(\mathbf{x})$$

Example: Gaussian degradation function 5×5:

0.003	0.014	0.025	0.014	0.003
0.014	0.058	0.095	0.058	0.014
0.025	0.095	0.150	0.095	0.025
0.014	0.058	0.095	0.058	0.014
0.003	0.014	0.025	0.014	0.003

Inverse filtering

We want to invert h, so that:

$$\hat{f}(\mathbf{x}) = g(\mathbf{x}) * h^{-1}(\mathbf{x})$$

Example: Gaussian degradation function 5×5:

0.003	0.014	0.025	0.014	0.003
0.014	0.058	0.095	0.058	0.014
0.025	0.095	0.150	0.095	0.025
0.014	0.058	0.095	0.058	0.014
0.003	0.014	0.025	0.014	0.003

Matrix is **singular**, there is no inverse!

Inverse filtering

If we know the PSF of the imaging system, the image formation can also be considered in frequency domain:

$$G(\mathbf{u}) = F(\mathbf{u})H(\mathbf{u})$$

Inverse and pseudo-inverse filtering

Now we divide the Fourier transform of the observed image by the PSF Fourier transform H, also called OTF (Optical Transfer Function).

$$\hat{F}(\mathbf{u}) = \frac{G(\mathbf{u})}{H(\mathbf{u})}$$

Inverse and pseudo-inverse filtering

Now we divide the Fourier transform of the observed image by the PSF Fourier transform H, also called OTF (Optical Transfer Function).

$$\hat{F}(\mathbf{u}) = \frac{G(\mathbf{u})}{H(\mathbf{u})}$$

When we know the OTF and we have a well-behaved transform (such as the Gaussian function), this operation is possible and approaches a perfect restoration.

Inverse and pseudo-inverse filtering

In a noisy image, we have:

$$\hat{F}(\mathbf{u}) = \frac{H(\mathbf{u})F(\mathbf{u}) + N(\mathbf{u})}{H(\mathbf{u})}$$

Inverse and pseudo-inverse filtering

In a noisy image, we have:

$$\hat{F}(\mathbf{u}) = \frac{H(\mathbf{u})F(\mathbf{u}) + N(\mathbf{u})}{H(\mathbf{u})}$$

$$\hat{F}(\mathbf{u}) = F(\mathbf{u}) + \frac{N(\mathbf{u})}{H(\mathbf{u})}$$

In this scenario and in those in which H shows values near zero, the ratio $\frac{N(\mathbf{u})}{H(\mathbf{u})}$ dominates the sum, and the resulting image is just noise.

Inverse and pseudo-inverse filtering

In some cases, it is possible to use the pseudo-inverse filtering, changing H below the threshold γ:

$$W(\mathbf{u}) = \begin{cases} H(\mathbf{u}), & H(\mathbf{u}) > \gamma \\ \gamma, & \text{otherwise} \end{cases}$$

The threshold is often between 0.0001 and 0.1. The filter W is then used to achieve the inverse:

$$\hat{F}(\mathbf{u}) = \frac{G(\mathbf{u})}{W(\mathbf{u})}$$

Least squares filtering

The pseudo-inverse filter allows to deal with null or small values, but its formulation does not include explicitly the noise model.

Least squares filters were developed in this context: the constrained least squares filter (CLS) and the Wiener filter are important examples.

Considering image and noise as random variables, this method tries to find an image estimate $\hat{f}$ so that the mean squared error is minimized:

$$e^2 = E\left\{(f - \hat{f})^2\right\}$$

Least squares filtering: Wiener

Assuming:

1. noise is not correlated;
2. noise has zero mean (centered at each pixel);
3. the intensities of the restored image can be written as a linear function of the degraded image.

$$\hat{F}(\mathbf{u}) = \left[\frac{H^*(\mathbf{u}) S_f(\mathbf{u})}{|H(\mathbf{u})|^2 S_f(\mathbf{u}) + S_\eta(\mathbf{u})} \right] \times G(\mathbf{u}),$$

Least squares filtering: Wiener

Assuming:

1. noise is not correlated;
2. noise has zero mean (centered at each pixel);
3. the intensities of the restored image can be written as a linear function of the degraded image.

$$\hat{F}(\mathbf{u}) = \left[\frac{H^*(\mathbf{u})S_f(\mathbf{u})}{|H(\mathbf{u})|^2 S_f(\mathbf{u}) + S_\eta(\mathbf{u})}\right] \times G(\mathbf{u}),$$

- $S_f(\mathbf{u}) = |F(\mathbf{u})|^2$ — power spectrum of the ideal image
- $S_\eta(\mathbf{u}) = |N(\mathbf{u})|^2$ — power spectrum of the noise
- $H^*(\mathbf{u})$ is the complex conjugate of $H(\mathbf{u})$

Least squares filtering: Wiener

How can we know the power spectrum of the ideal/original image and of the additive noise.

- Using the noise variance as parameter, and the direct method of periodogram:
  - $\hat{S}_\eta(\mathbf{u}) = \sigma_\eta^2$ for all $(\mathbf{u})$
  - $\hat{S}_f(\mathbf{u}) = 1/N^2\,[G(\mathbf{u})G^*(\mathbf{u})] - \sigma_\eta^2$

Least squares filtering: Wiener

How can we know the power spectrum of the ideal/original image and of the additive noise.

- Using the noise variance as parameter, and the direct method of periodogram:
  - $\hat{S}_\eta(\mathbf{u}) = \sigma_\eta^2$ for all $(\mathbf{u})$
  - $\hat{S}_f(\mathbf{u}) = 1/N^2\,[G(\mathbf{u})G^*(\mathbf{u})] - \sigma_\eta^2$

There are other methods to obtain S_η and S_f, but it required additional knowledge about image and noise.

Constrained Least squares filtering

From a similar formulation, considering a constraint in the least squares, a method was proposed by regularizing the solution via a Laplacian operator:

$$\hat{F}(\mathbf{u}) = \left[\frac{H^*(\mathbf{u})}{|H(\mathbf{u})|^2 + \gamma |P(\mathbf{u})|^2} \right] \times G(\mathbf{u}),$$

where $P(\mathbf{u})$ is the Fourier transform of a Laplacian operator:

$$p(\mathbf{x}) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

γ controls the influence of the regularization

References

- Gonzalez, R.C.; Woods, R.E. Processamento Digital de Imagens. 3.ed. Capítulo 5. 2010.
- Lagendijk, R.L.; Biemond, J. Basic Methods for Image Restoration and Identification (Capítulo 3.5). In: Bovik, A. Handbook of Image and Video Processing, 2000.
- Image quality in digital cameras, Roger Clark: www.clarkvision.com