(I’m Fun) Digital Image Fundamentals

Week 2: Point Operations

Thuong Nguyen Canh

Institute for Datability Science, Osaka University

November 2019

Image Evolution (1)

- Digital Image: A Pixel Array (W × H)
  - RGB color images has three channel (W × H × 3)

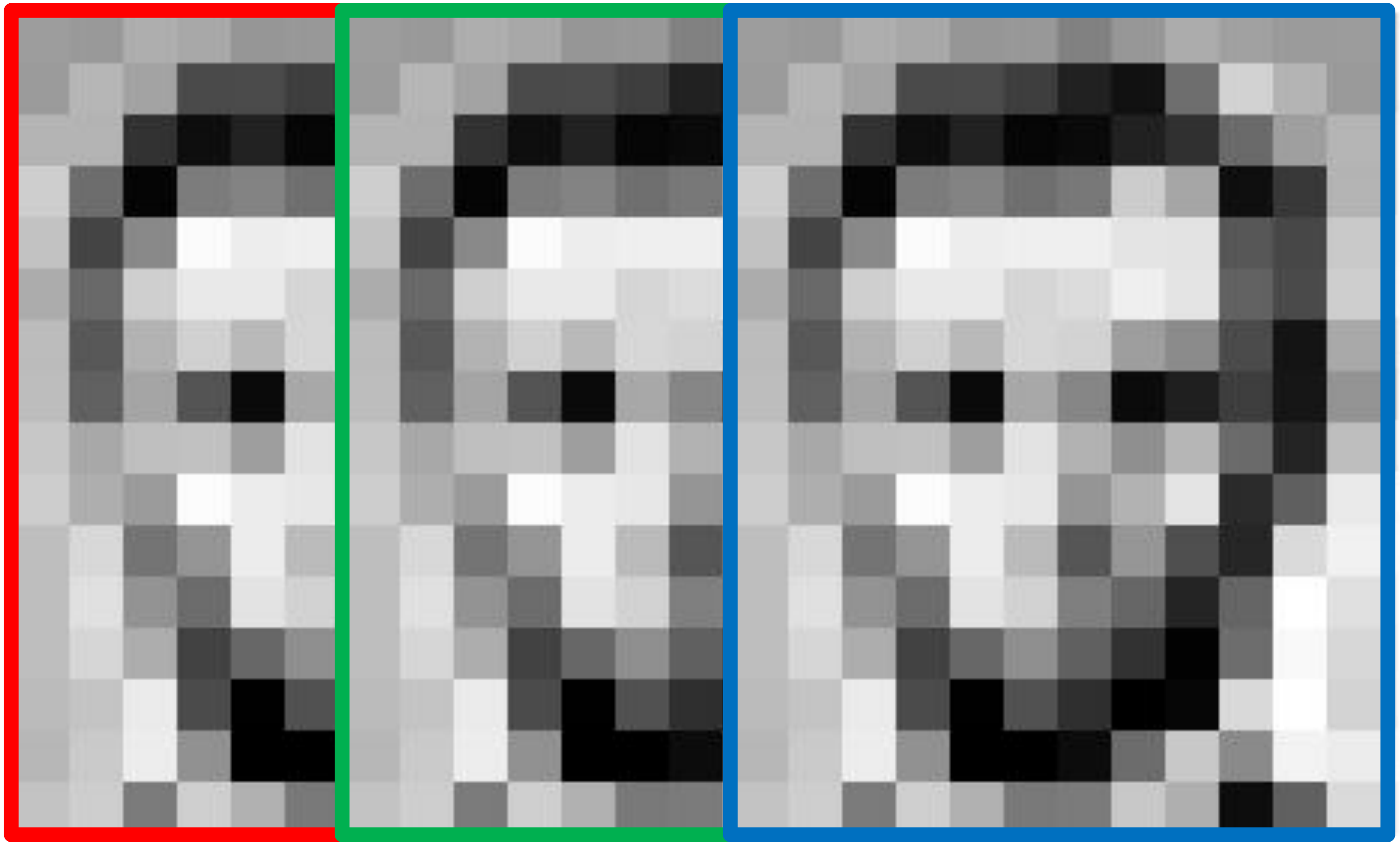
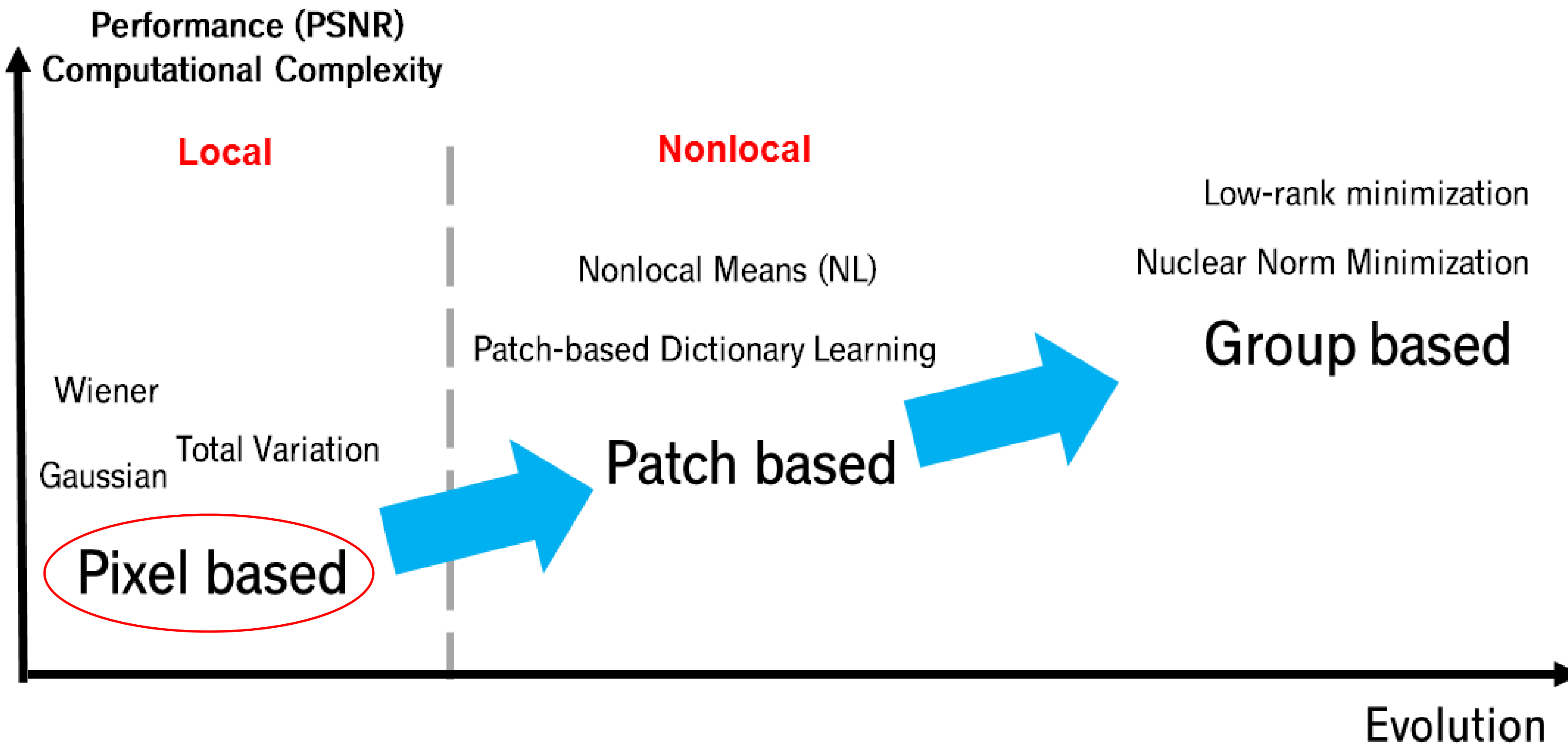


Image Evolution (2)

- Image Priors
  - Local and nonlocal self-similarity and many more

Image Evolution (3)

- From pixel (point) operation to patch-based, and group based



POINT OPERATIONS - INTRO

Point Operations

- How do gray values relate to brightness?
  - Quantization
  - Weber's Law
  - Gamma characteristic
  - Adjusting brightness and contrast

Quantization

- **Dynamic range** is the range between the lightest and darkest part of an image

How many bits per pixels (to represent the range)?

8 bits | 5 bits | 4 bits

3 bits | 2 bits | 1 bits

When do you start to see the “contouring artifact”?

How many gray levels are required?

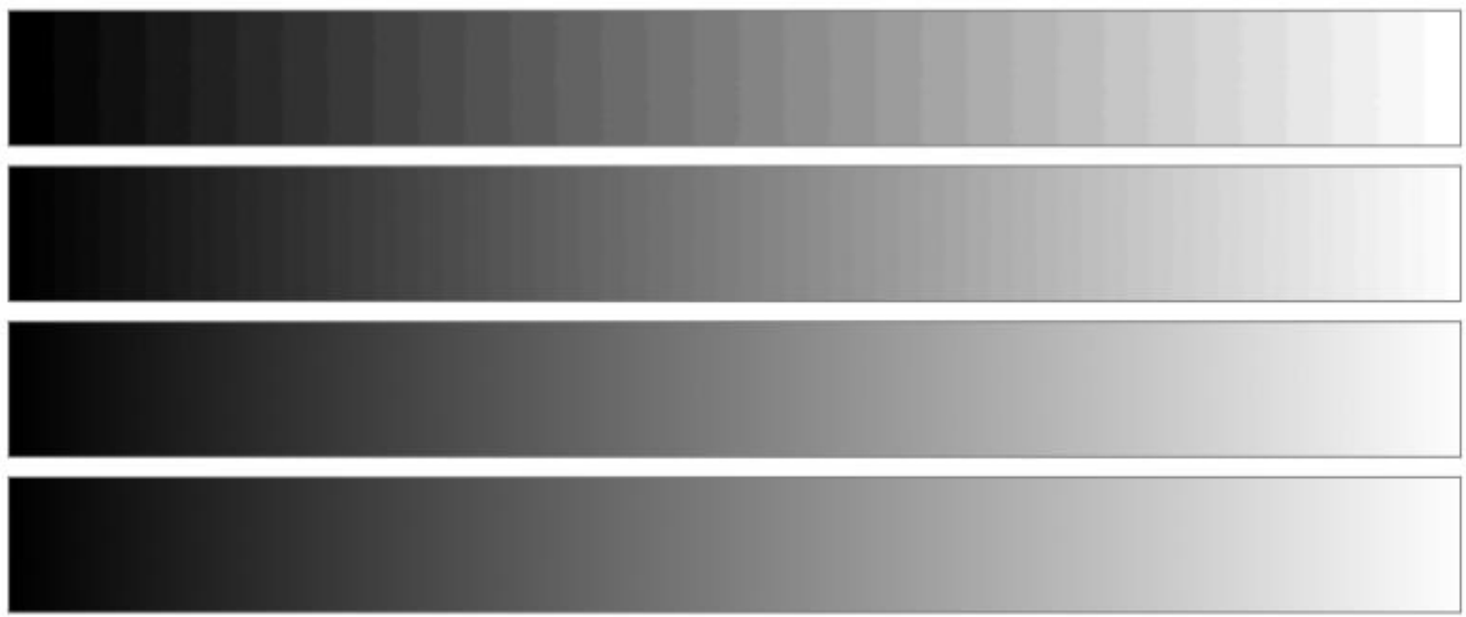
Contouring is most visible for a ramp (slope)

5 bits 32 levels

6 bits 64 levels

7 bits 128 levels

8 bits 256 levels



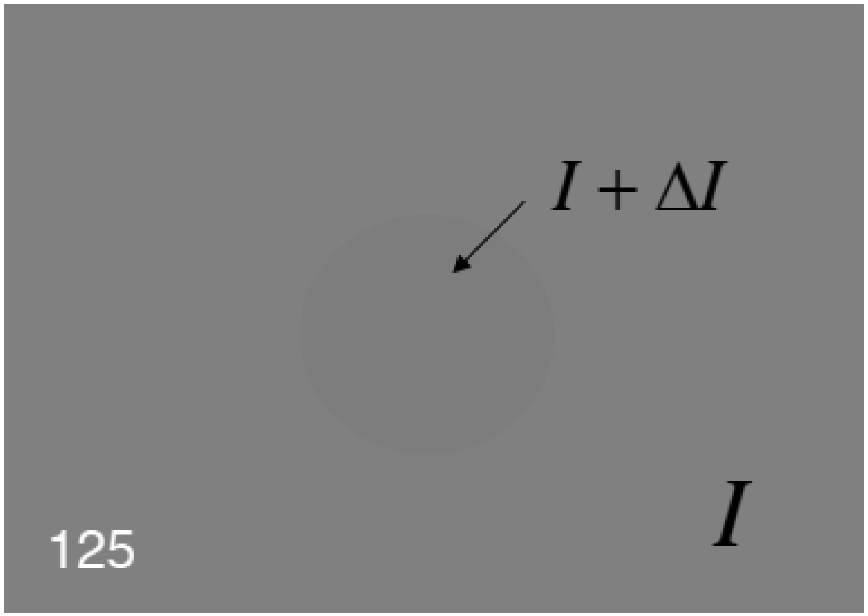
Digital images typically are quantized to 256 gray levels

Standard vs. High Dynamic Range

12 bits/pixel (High)

8 bits/pixel (Standard)

Brightness Discrimination (1)



Can you see the circle?

Visibility Threshold

$$\boxed{\Delta I / I \approx 1 \ldots 2\%}$$

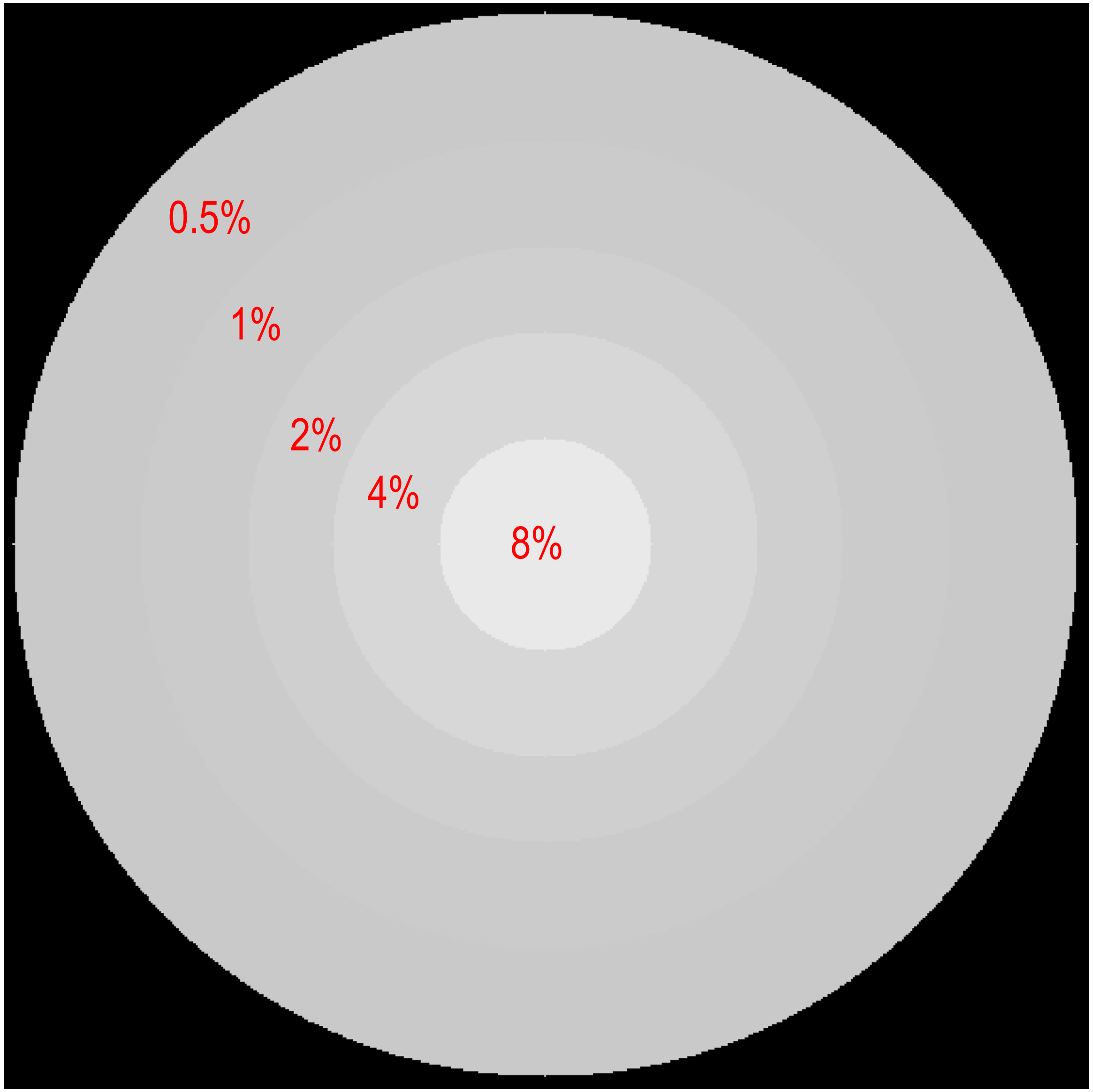
Weber fraction,
Weber's Law

Note: I is luminance, measured in cd/m^2

Human brightness perception is uniform in the log(I) domain

Fenchner's Law

Brightness Discrimination (2)



How many ring can you see?

Contrast ratio without contouring

- Ratio of luminance bet. the brightest and the darkest color of a given system

$$\frac{I_{max}}{I_{min}} = (1 + K_{weber})^{N-1}$$

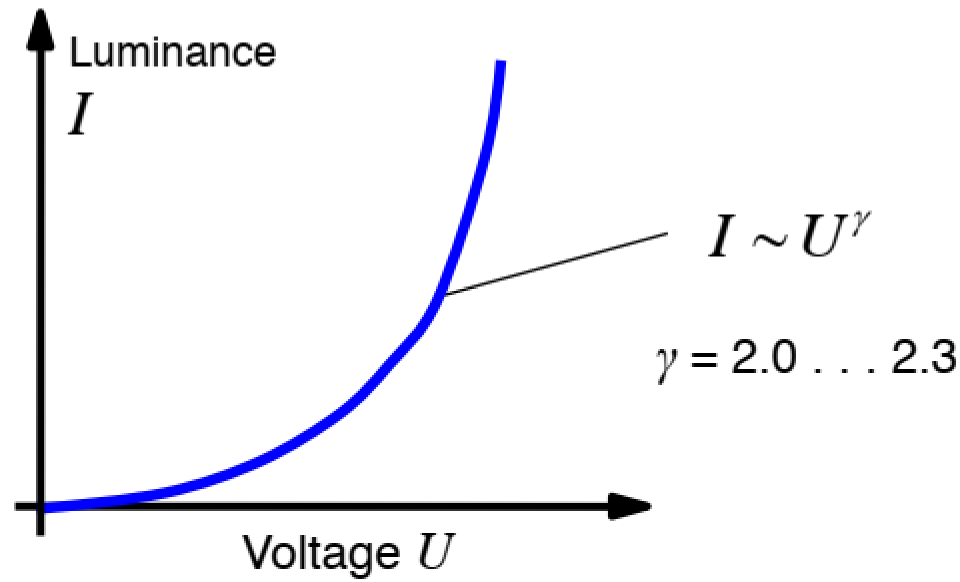
  - $K_{weber} = 0.01, \ldots, 0.02; \quad N = 256; \quad I_{\max}/I_{\min} = 13, \ldots, 156$

- Common display contrast ratio
  - Modern OLED in dark room 1,000,000:1
  - LCD in dark room 1,000:1 ~ 4000:1
  - Cathode ray tube 100:1
  - Print on paper 10:1

- Human eye supports contrast ratio of 1,000,000:1
  - Your eye's contrast ratio is approximately 1,000:1
  - But adjusting ratio for day and night time

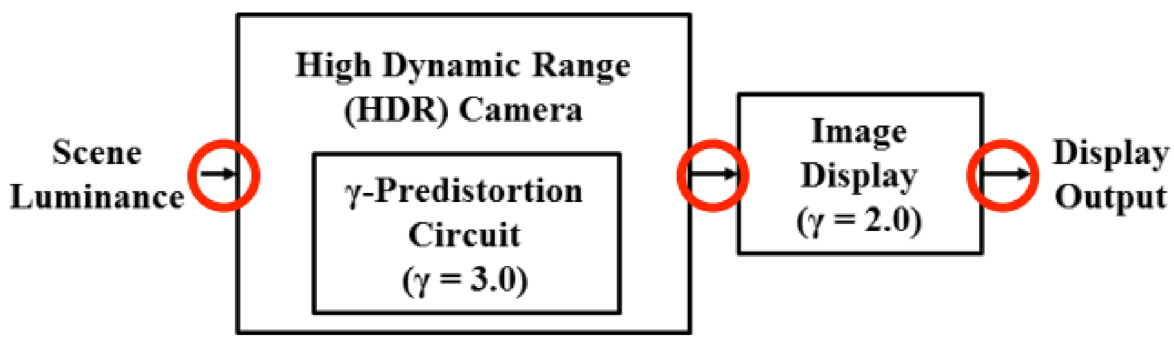
Gamma Characteristic

- Cathode ray tubes (CRTs) are nonlinear

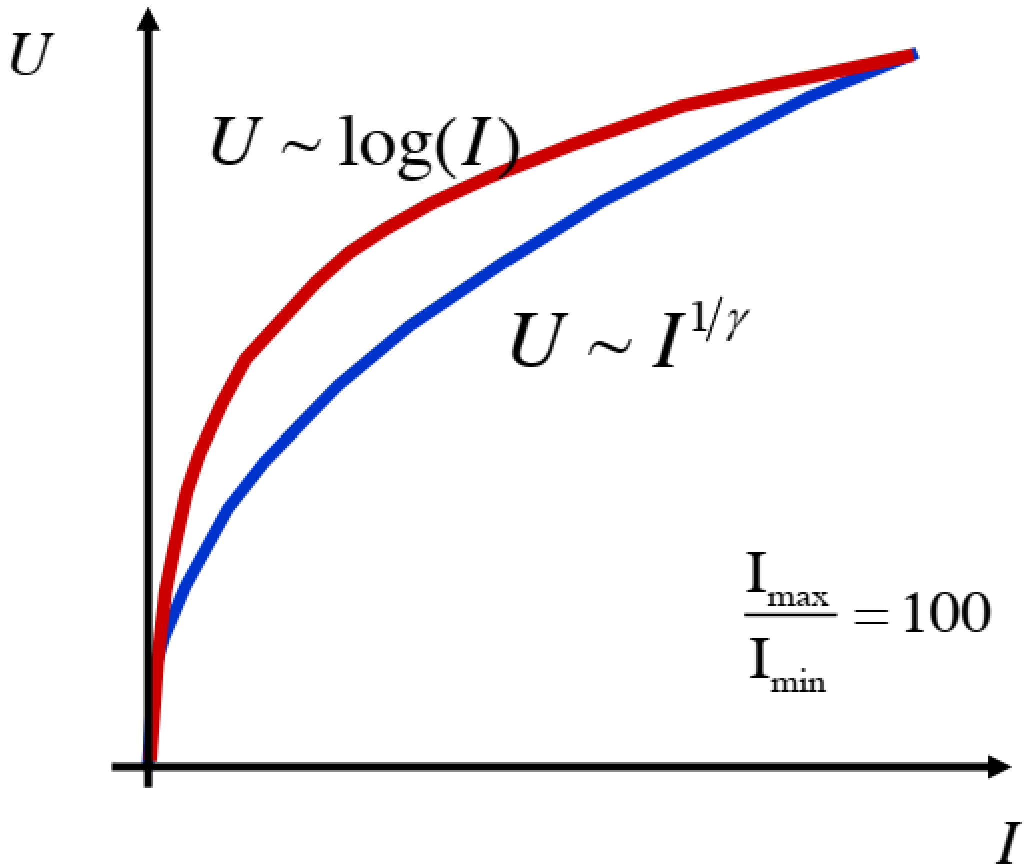


- Cameras contain γ-predistortion circuit

$$\boxed{U \sim I^{1/\gamma}}$$

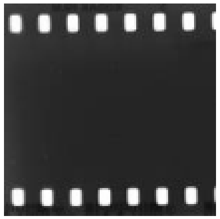


Log vs. γ-predistortion



- Weber's Law suggests uniform perception in the log(*I*) domain
- Similar enough for most practical applications

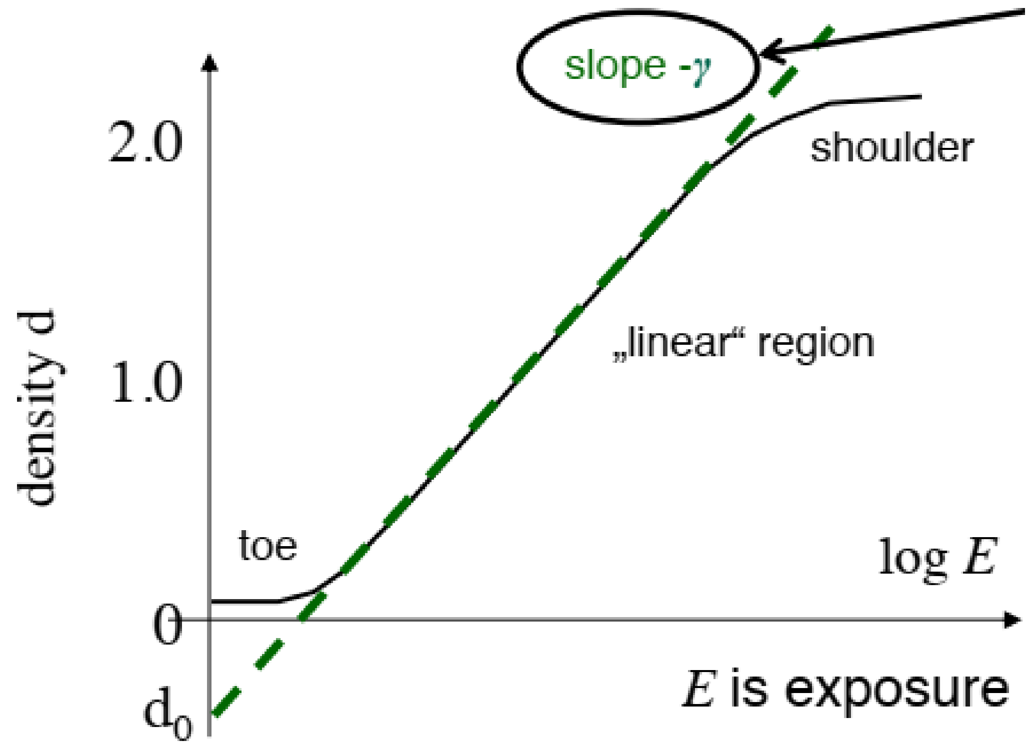
Photographic Film



Luminance

$$I = I_0 \cdot 10^{-d}$$
$$= I_0 \cdot 10^{-(-\gamma \log E + d_0)}$$
$$= I_0 \cdot 10^{-d_0} \cdot E^{\gamma}$$

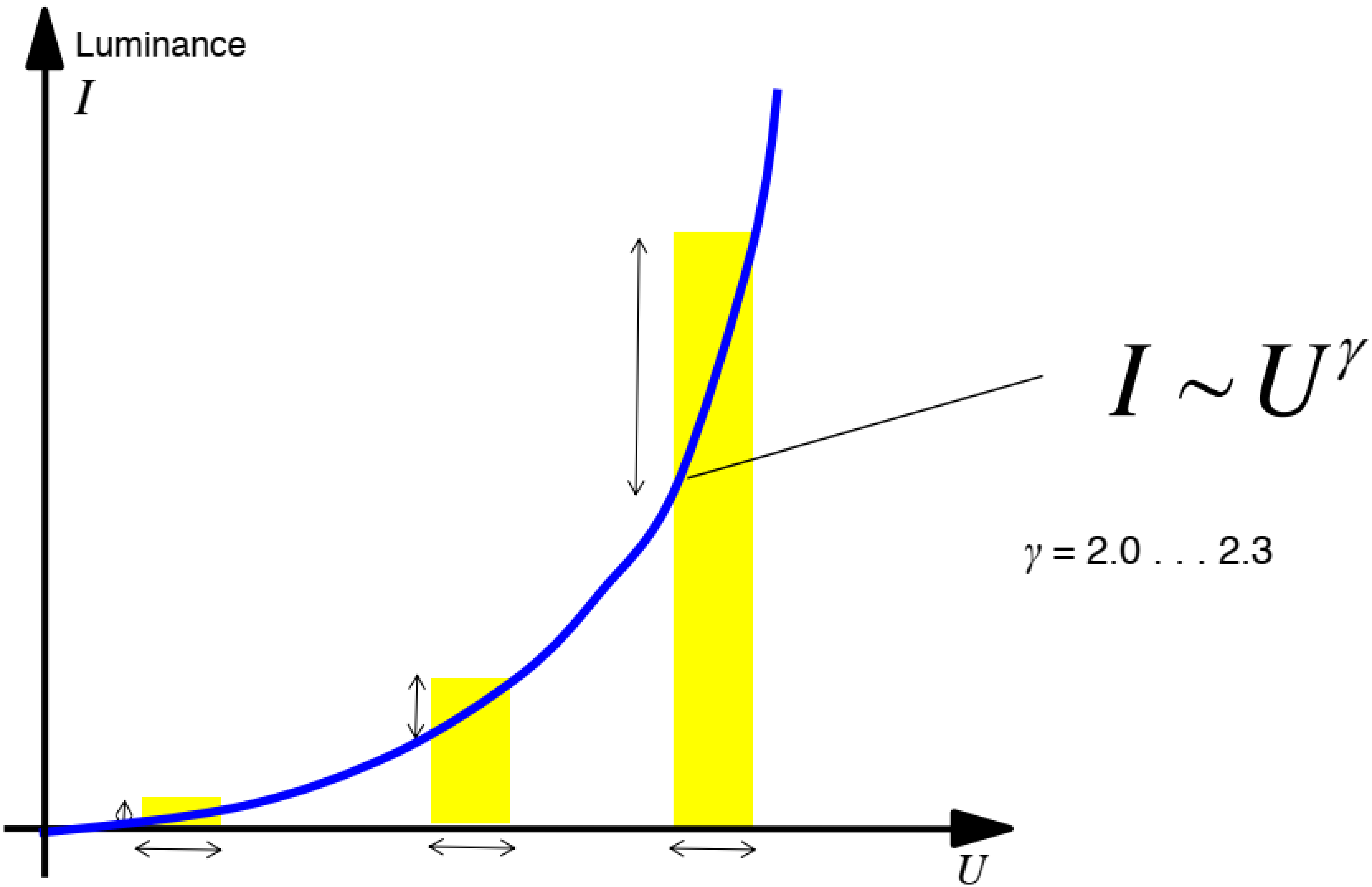
Hurter & Driffield curve (H&D curve)
for photographic negative



γ measures film contras
t

- General purpose films
  γ = -0.7 . . . -1.0
- High-contrast films
  γ = -1.5 . . . -10
- Lower speed films tend to have higher absolute γ

Photographic Film



Brightness Adjustment by Intensity Scaling

$f[x,y]$

$a \cdot f[x,y]$

- Scaling in the γ-domain is equivalent to scaling in the linear luminance domain

$$I \sim \left(a \cdot f[x,y]\right)^{\gamma} = a^{\gamma} \cdot \left(f[x,y]\right)^{\gamma}$$

Similar effect as changing the camera exposure time

Contrast adjustment by changing γ

Original image

$f[x,y]$

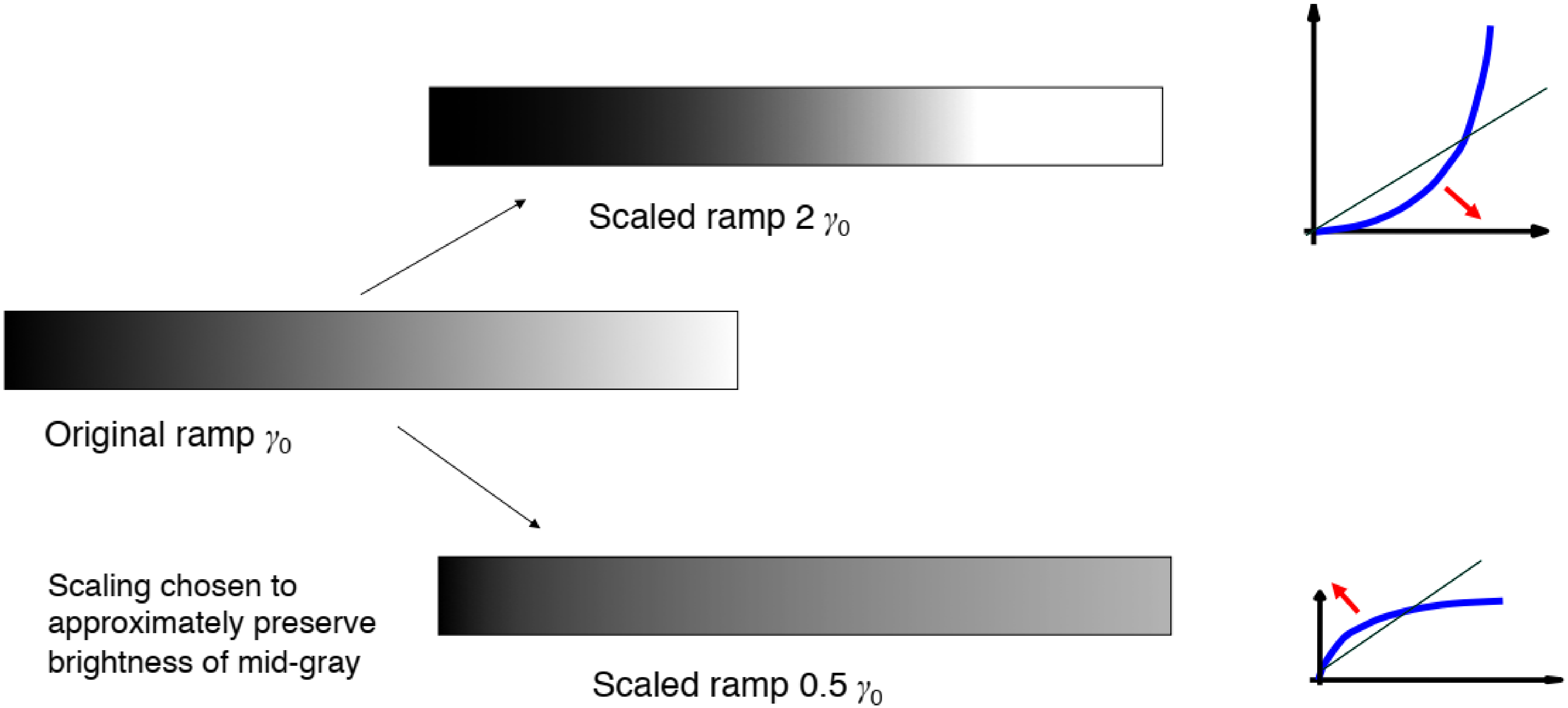
γ increased by 50%

$a \cdot (f[x,y])^{\gamma}$

with $\gamma = 1.5$

. . . same effect as using a different photographic film . . .

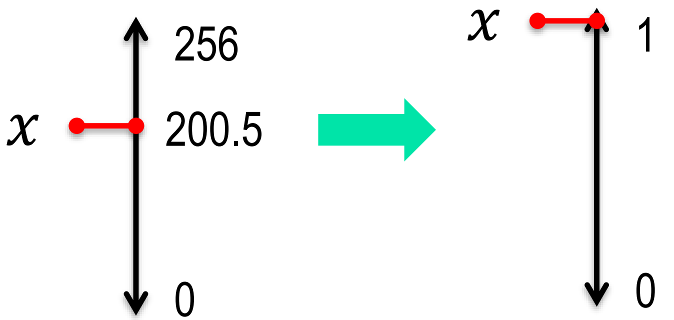
Contrast adjustment



Code Review: Quantization

- Quantization in Analog to Digital Converter (ADC)

Uniform Quantization
Scalar Quantization



Quantization

$$x_q = \left\lfloor \frac{x}{\Delta} + \theta \right\rfloor = \left\lfloor \frac{200.5}{256} + \frac{1}{2} \right\rfloor = 1$$

De-Quantization

$$Q(x) = \Delta \left\lfloor \frac{x}{\Delta} + \theta \right\rfloor = 256 \times 1 = 256$$

Mid-tread
involves rounding

$$Q(x) = \Delta \cdot \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor = \Delta \cdot \mathrm{floor}\left(\frac{x}{\Delta} + \frac{1}{2} \right),$$

Δ: quantization step
θ: quantization offset

Mid-riser
involves truncate

$$Q(x) = \Delta \cdot \left(\left\lfloor \frac{x}{\Delta} \right\rfloor + \frac{1}{2} \right)$$

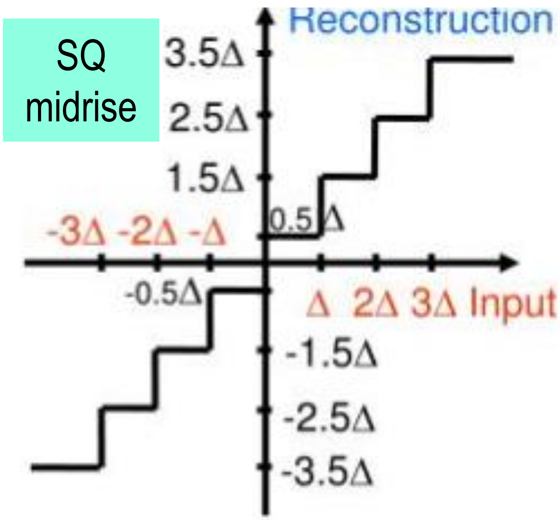
Dead-zone quantizers

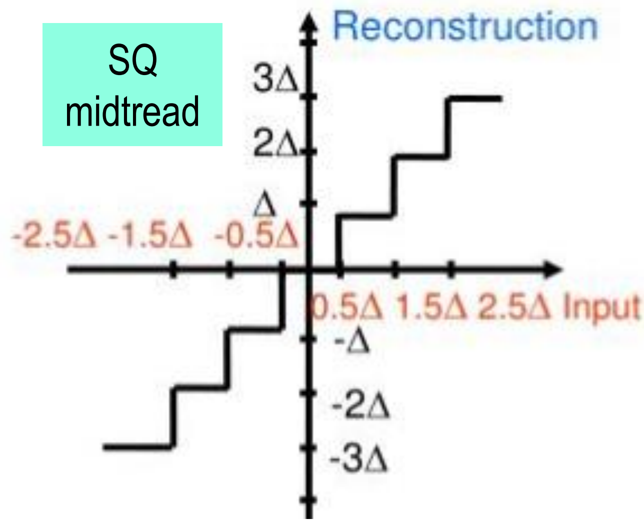
symmetric behavior around 0

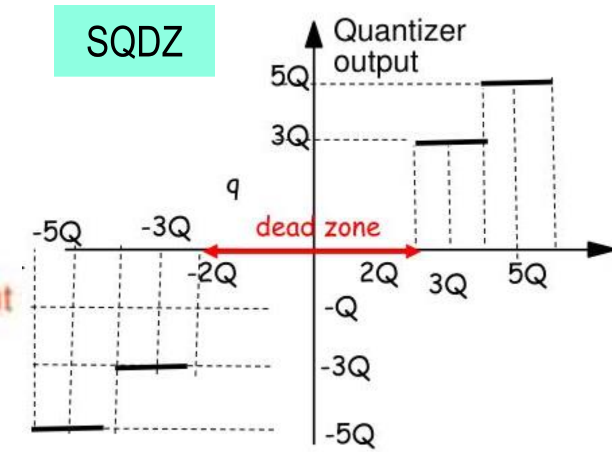
$$k = \mathrm{sgn}(x) \cdot \max\left(0, \left\lfloor \frac{|x| - w/2}{\Delta} + 1 \right\rfloor \right),$$

$$y_k = \mathrm{sgn}(k) \cdot \left(\frac{w}{2} + \Delta \cdot (|k| - 1 + r_k) \right),$$

Code Review: Quantization (3)







x	SQ-midrise	x	SQ-midtread	x	SQDZ
[-2, 1)	-1.5	(-2.5, -1.5]	-2	(-2.5, -1.5]	-2
[-1, 0)	-0.5	(-1.5, -0.5]	-1	(-1.5, -0.5]	0
[0, 1)	0.5	(-0.5, 0.5]	0	(-0.5, 0.5]	0
[1, 2)	1.5	(0.5, 1.5]	1	(0.5, 1.5]	0
[2, 3)	2.5	(1.5, 2.5]	2	(1.5, 2.5]	2

Heejun Ahn, Image and video compression, presentation 2013.

Code Review: Quantization

- What type of quantization in this code?

$$x_q = \left\lceil \frac{x}{\Delta} \right\rceil \times \Delta - 1$$

$$= \left(\left\lfloor \frac{x}{\Delta} \right\rfloor + 1 \right) \times \Delta - 1$$

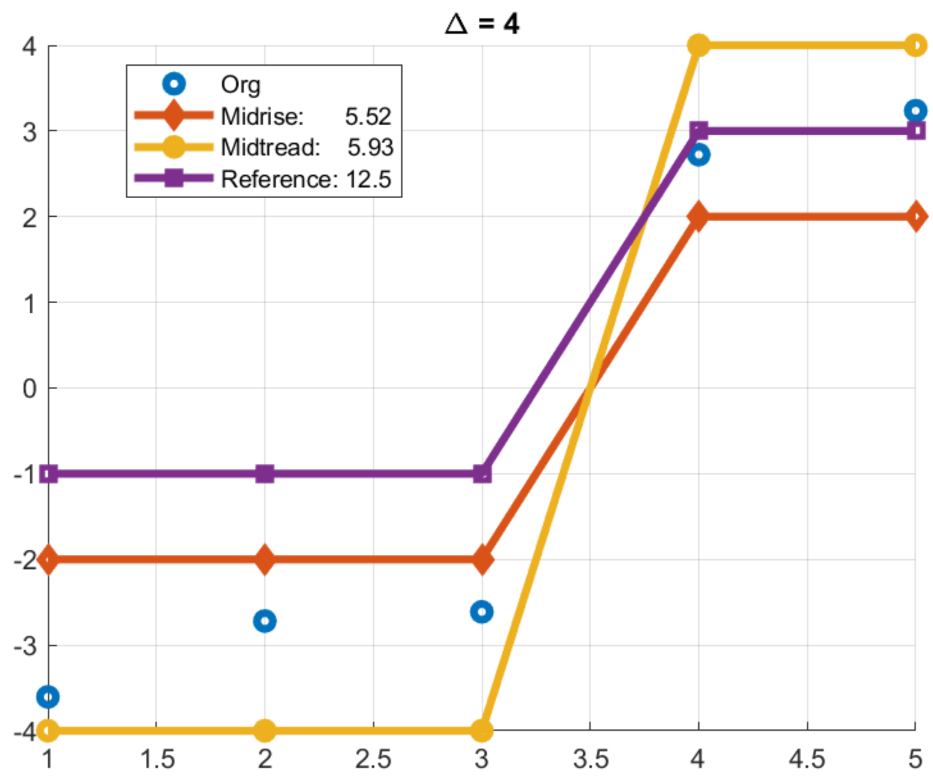
$$= \left(\left\lfloor \frac{x}{\Delta} \right\rfloor + \frac{1}{2} \right) \times \Delta + \left(\frac{\Delta}{2} - 1 \right)$$

x

Δ

```matlab
clear, clc, close all;

% Load test image
img = double(imread('face.jpg'));

% Loop over number of bits
for numOfBit = 1 : 8
    % Quantize to given number of bits
    numOfLevel = 2.^ numOfBit;
    levelGap = 256 / numOfLevel;
    quantizedImg = uint8(ceil(img / levelGap) * levelGap - 1); % quantization

    % Plot image
    subplot(2, 4, 9 - numOfBit), imshow(quantizedImg);
    if numOfBit == 1
        name = [num2str(numOfBit) '-bit'];
    else
        name = [num2str(numOfBit) '-bits'];
    end
    title(name);

    % Save image
    imwrite(quantizedImg, ['Quantization_' name '.png'] );
end %end numOfBit
```

Code Review: Quantization



Code Review: Contrast

```
% Load test image
img = im2double(imread('parrot.jpg'));

%% Brightness adjustment by intensity scaling
scale = 1.2;
scaledImg = scale .* img;

subplot(1, 2, 1), imshow(img, [0, 1]);
title('Original image');
subplot(1, 2, 2), imshow(scaledImg, [0, 1]);
title('Scaled image');
imwrite(scaledImg, 'chap2_Brightness_scaled.png');

%% Contrast adjustment by changing "gamma"
gamma = 1.5;
gammaImg = img.^gamma;
```

$$I \sim \left(a \cdot f[x,y]\right)^{\gamma} = a^{\gamma} \cdot \left(f[x,y]\right)^{\gamma}$$

scaleImg

Img

gammaImg

POINT OPERATIONS – IMAGE COMBINATION

Image Combination

- Image averaging for noise reduction
- Combination of different exposures for high-dynamic range imaging
- Image subtraction for change detection
- Need for accurate alignment
- Displacement estimation

Image Averaging (1)

- Image averaging for noise reduction

1 image 2 images

8 images 32 images

Image Averaging (2)

- Why averaging?

Additive noise model $y = x + n$

  - We assume noise is random and zero mean

$$y_1 = x + n_1$$
$$y_2 = x + n_2$$

$$\bar{y} = \frac{1}{2}(y_1 + y_2) = x + \frac{(n_1 + n_2)}{2}$$

$$y_n = x + n_2$$

$$\bar{y} = \frac{1}{n}(y_1 + \cdots + y_2) = x + \frac{(n_1 + \cdots + n_n)}{n}$$

$$\cong 0$$

More images is better for random noise removal

High-dynamic range imaging

-8 f-stops

-2 f-stops

+2 f-stops

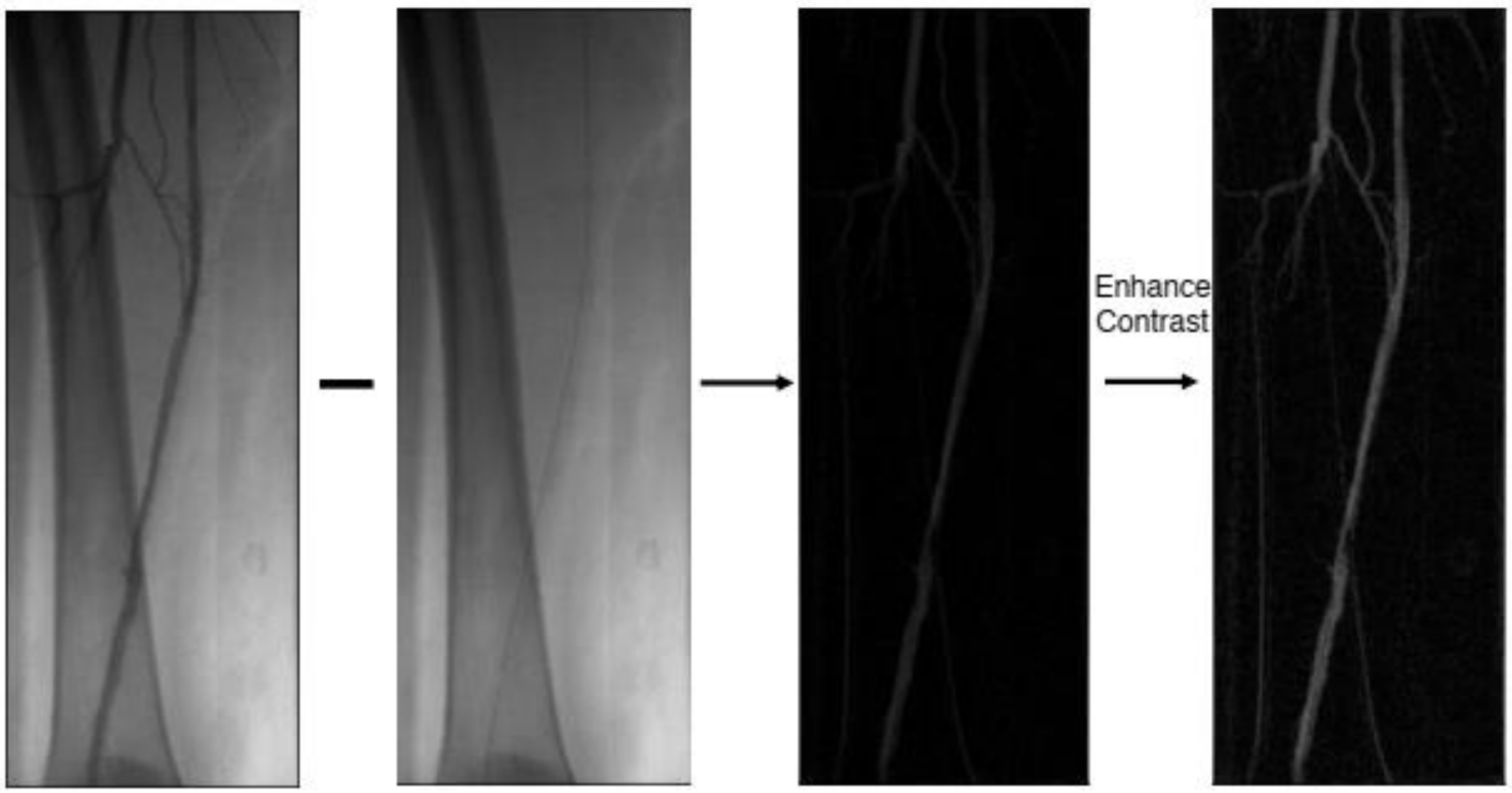
+4 f-stops

Blended image from Exposure Fusion

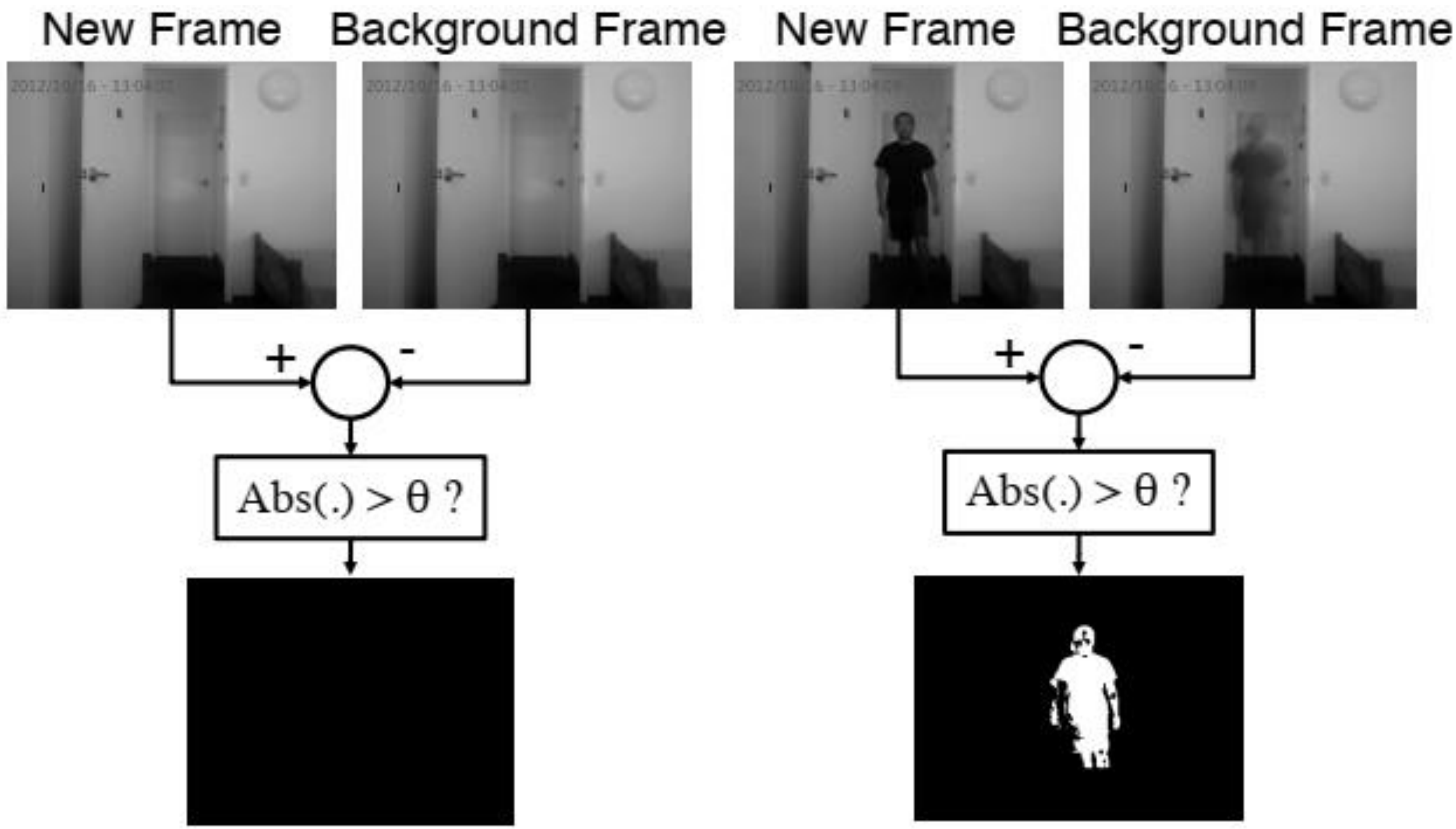
[Tom Mertens et al. 2007]

Image subtraction

- Find differences/changes between 2 mostly identical images
- Example: digital subtraction angiography



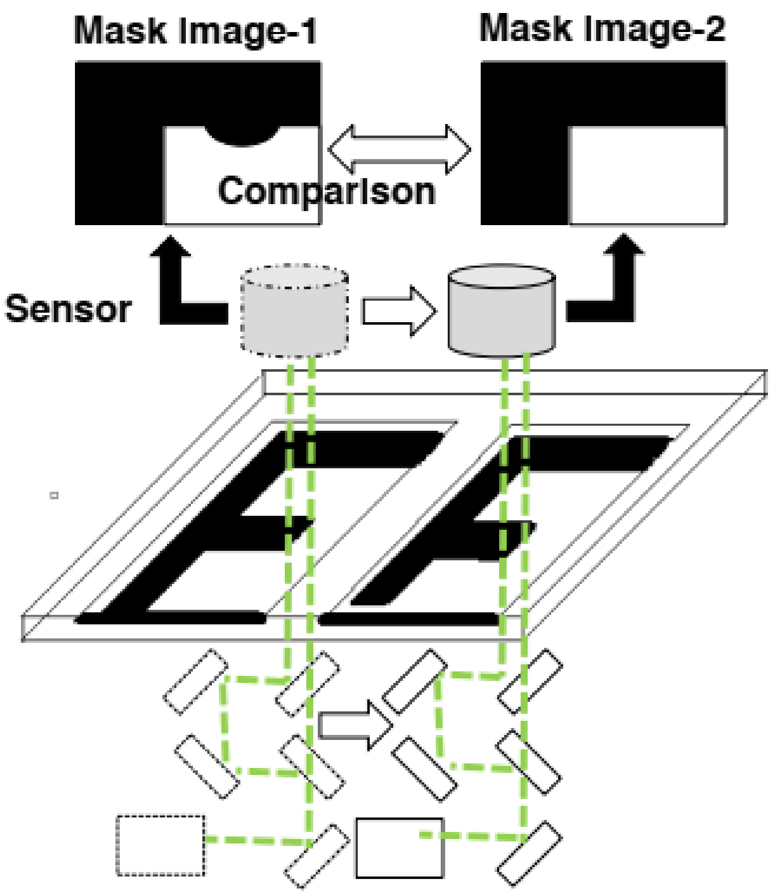
Video background subtraction



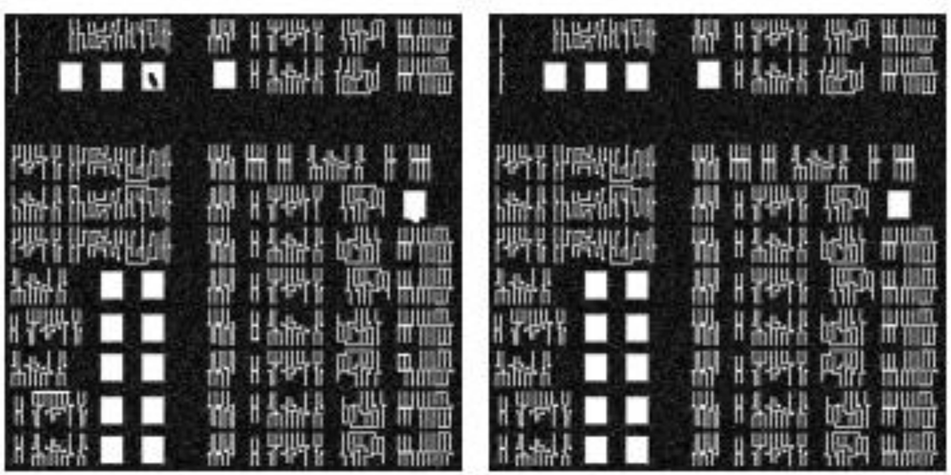
Update:

Background[t] := α Background[t-1] + (1- α) New[t]

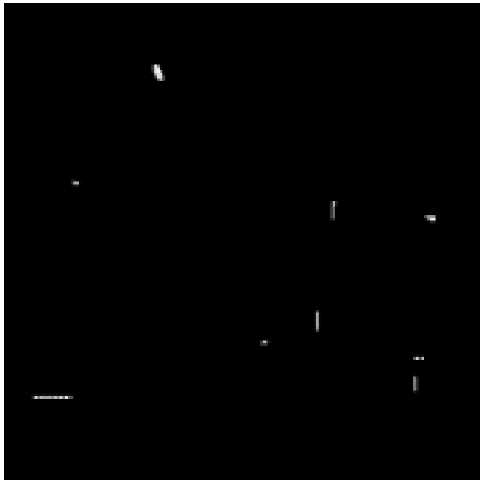
Image Subtraction in IC Manufacturing



Laser system



Mask image-1 Mask image-2



Difference image

Where is the defect?

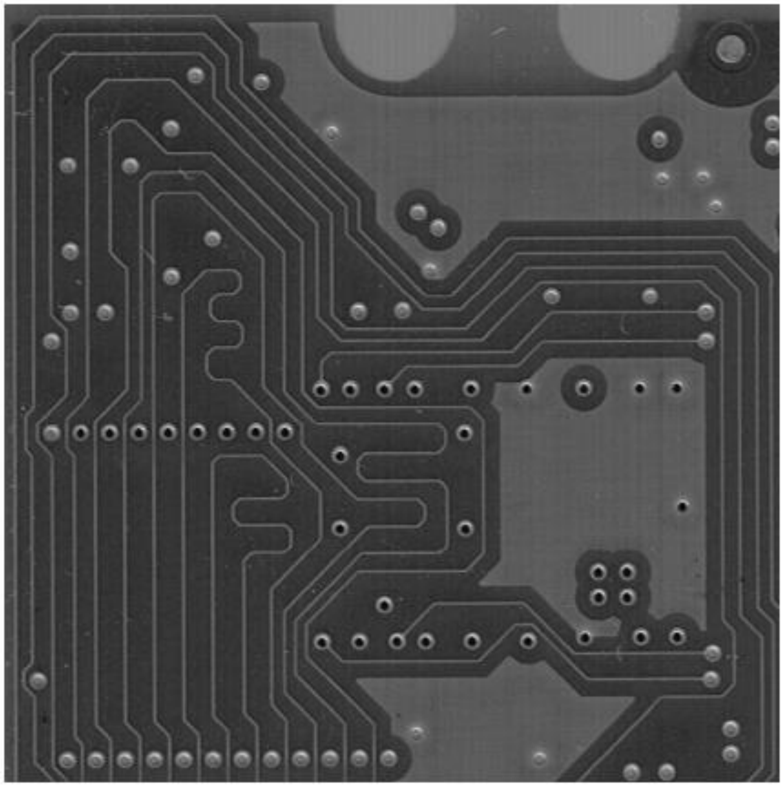


Image $g[x,y]$ (no defect)

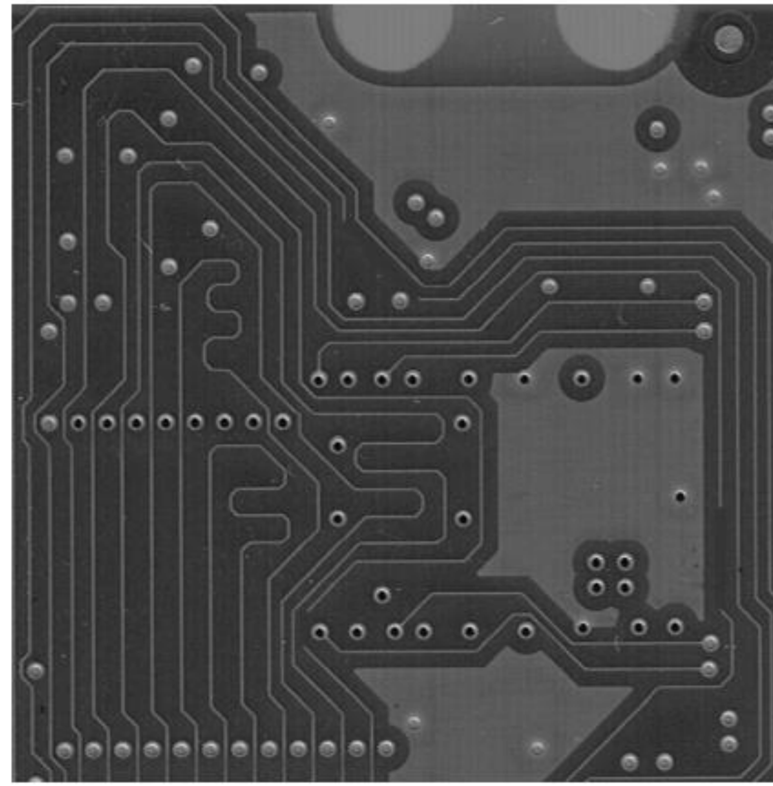
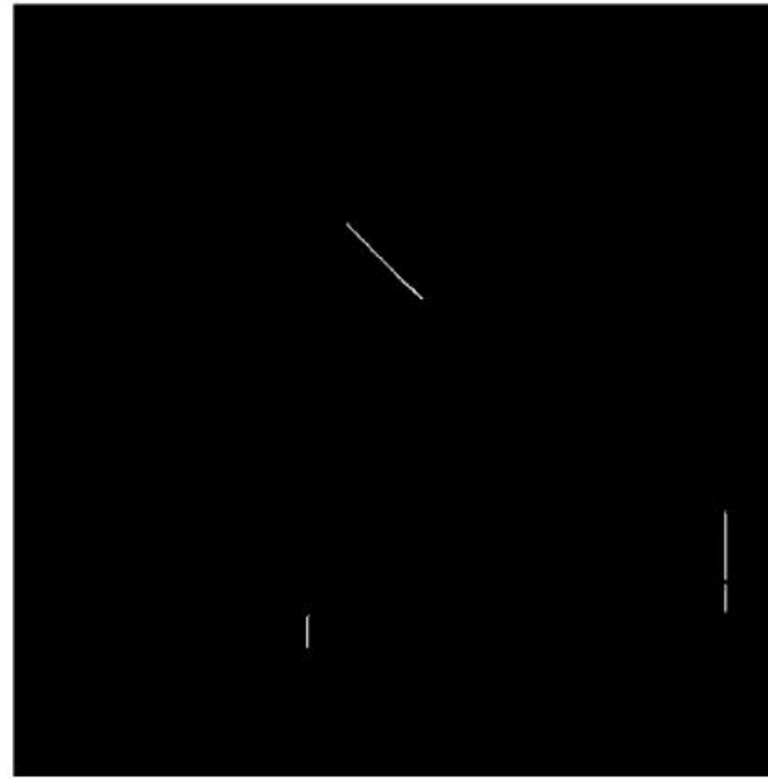


Image $f[x,y]$ (w/ defect)

Where is the defect?

Absolute difference between two images

$|f\text{-}g|$ w/o alignment



$|f\text{-}g|$ w/ alignment

Displacement Est. by block matching

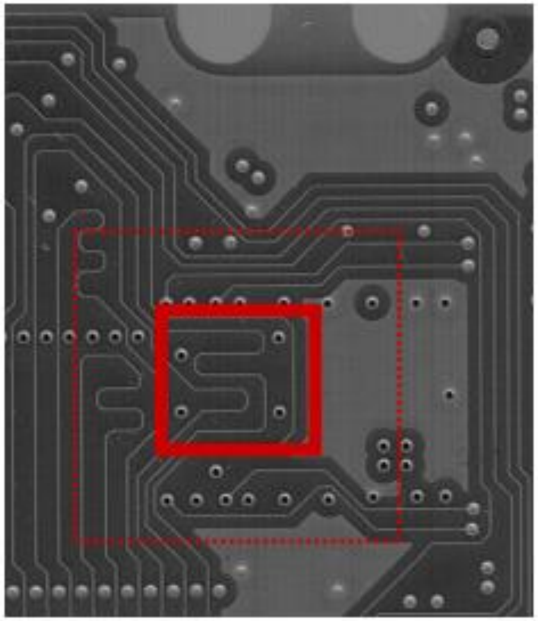


Image $g[x,y]$

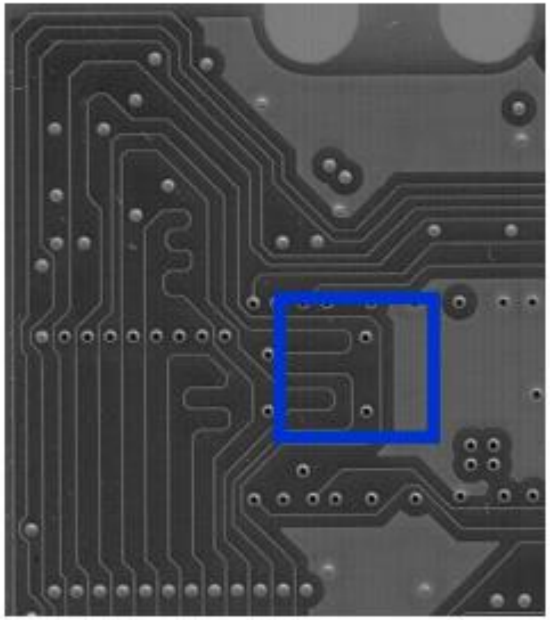
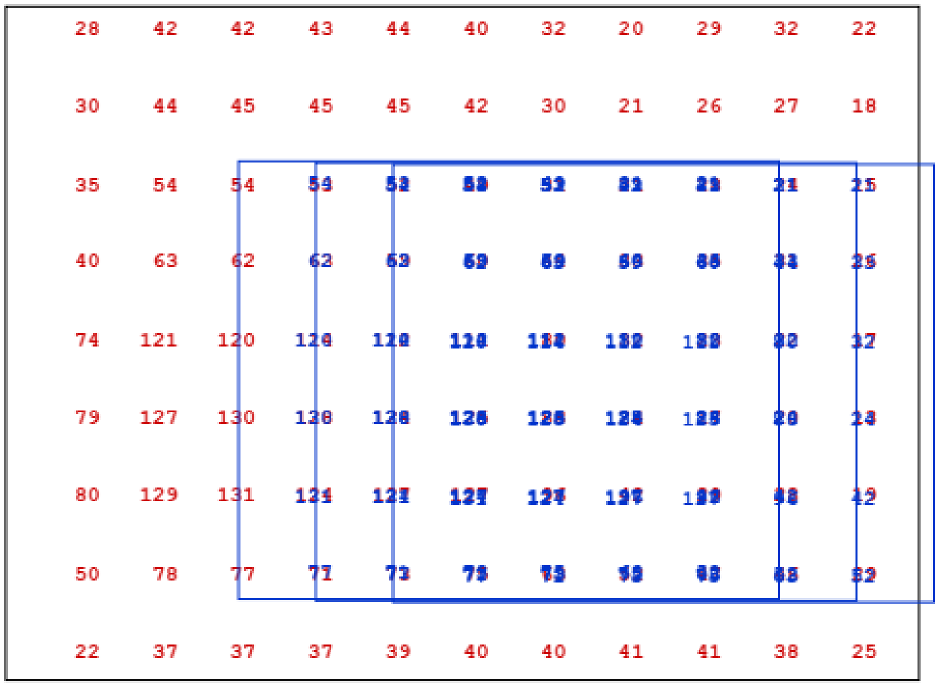


Image $f[x,y]$

Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

Rectangular array of pixels is selected as a measurement window

Integer pixel shifts



54	53	52	49	31	21
62	63	59	60	44	33
120	114	112	111	80	32
130	128	124	125	88	24
131	124	127	127	96	42
77	71	73	75	63	52

Rectangular array of pixels is selected as a measurement window

Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

Error metric

- Sum of Squared Differences

Sum all values in measurement window

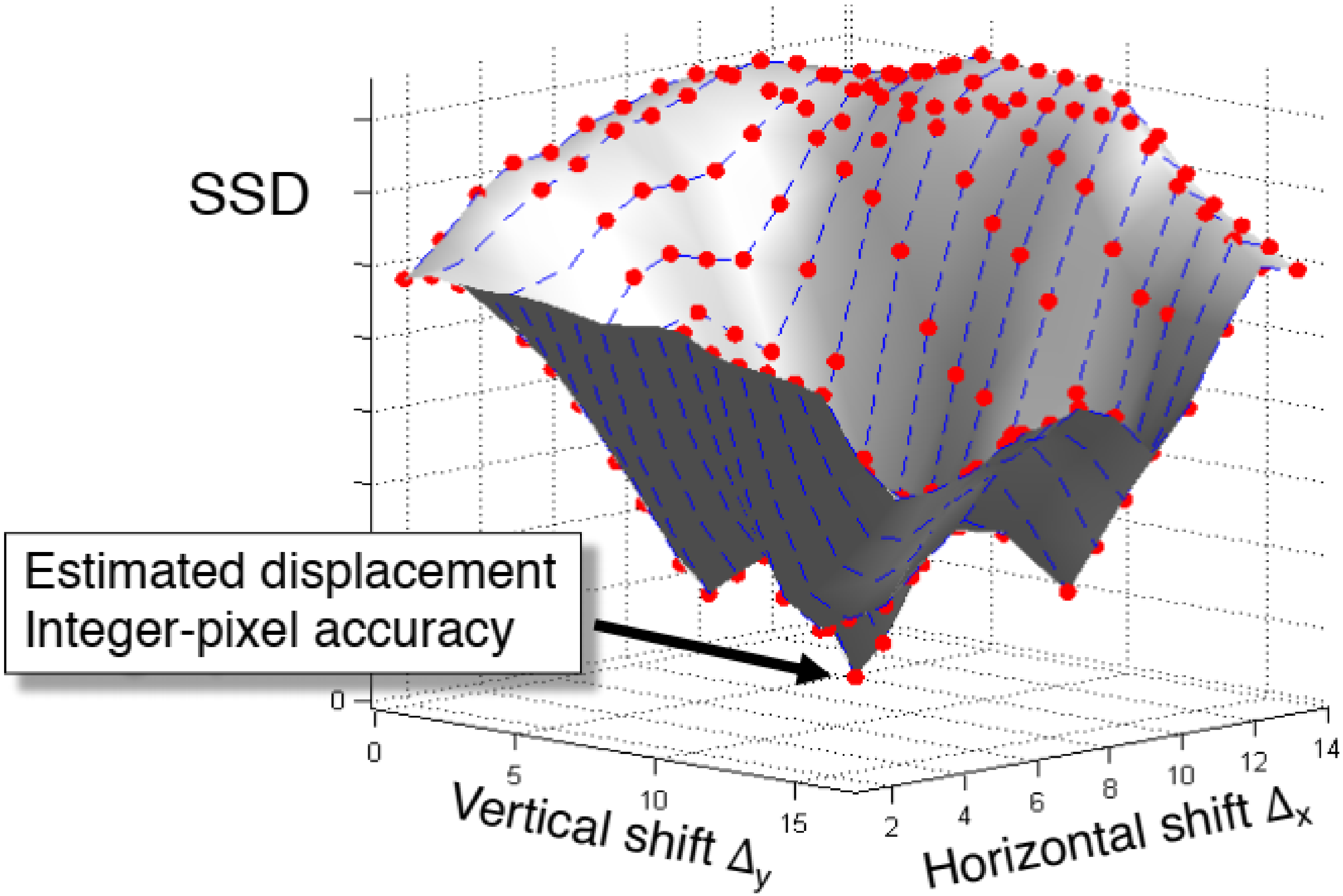
$$SSD\left[\Delta_x,\Delta_y\right]=\sum_{[x,y]\in\text{msmnt window}}\left(f\left[x,y\right]-g\left[x+\Delta_x,y+\Delta_y\right]\right)^2$$

Horizontal displacement

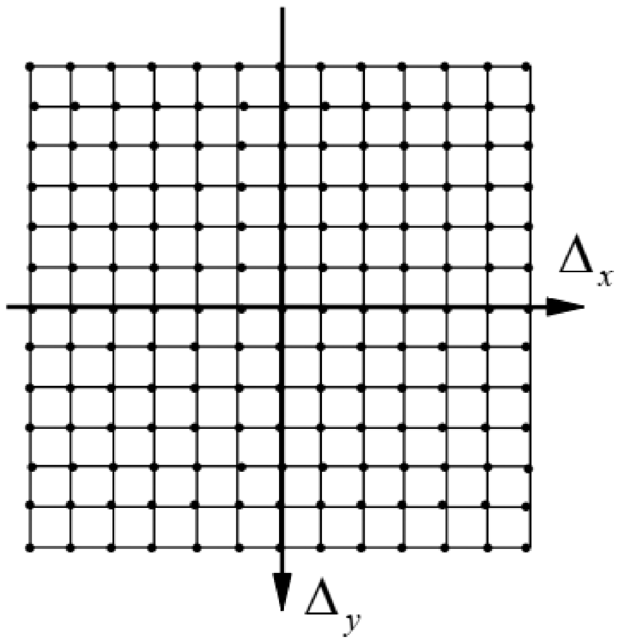
Vertical displacement

- Alternatives: SAD (Sum of Absolute Differences), cross correlation, mutual info rmation . . .
- Robustness against outliers: sum of saturated squared differences, median of squared differences . . .

SSD Value from block matching



Block Matching: Search Strategies

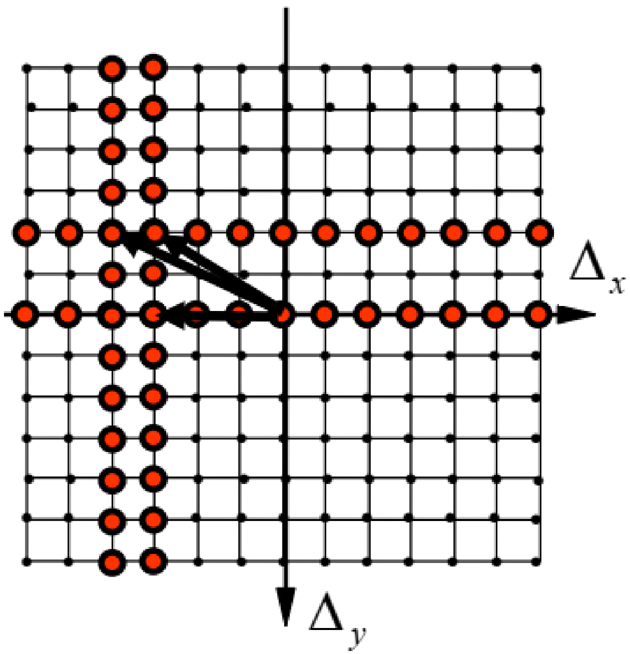


Full search

- All possible displacements within the search range are compared.
- Computationally expensive
- Highly regular, parallelizable

Block Matching: Search Strategies

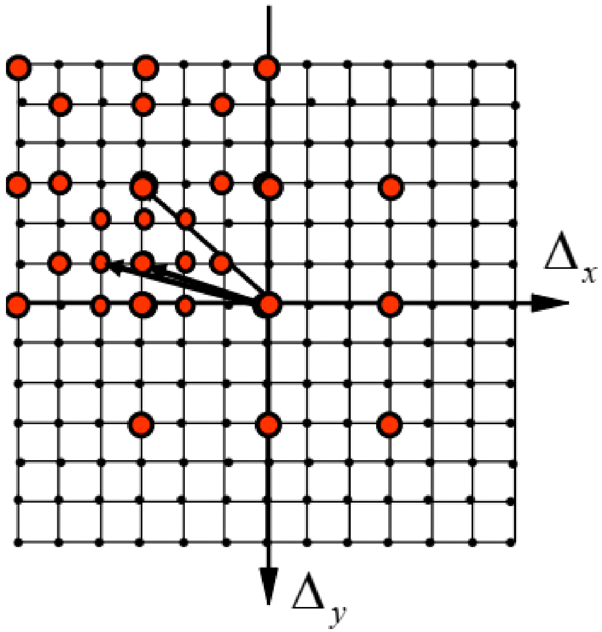
Conjugate direction search



- Alternate search in x and y directions
- Stop when there is no further improvement

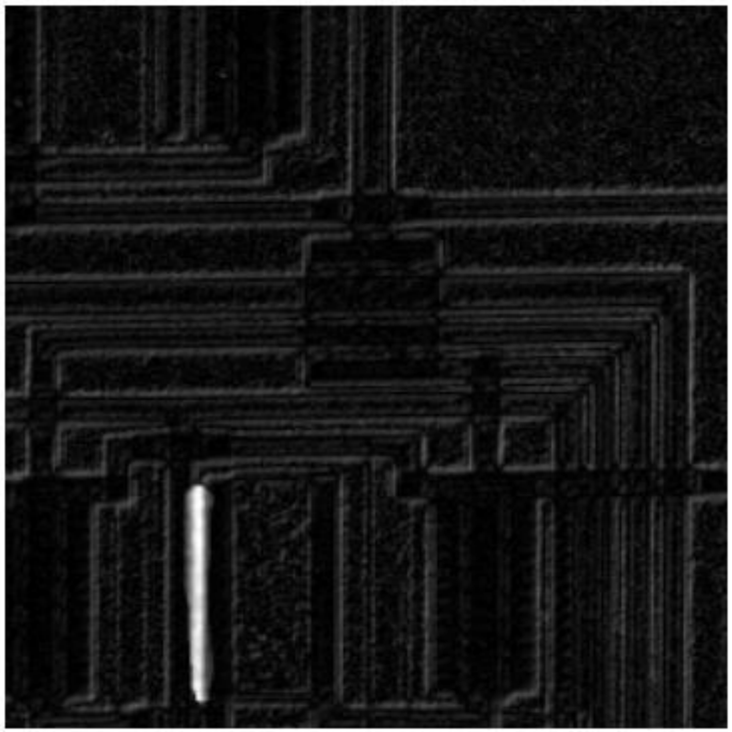
Block Matching: Search Strategies

Coarse-to-fine

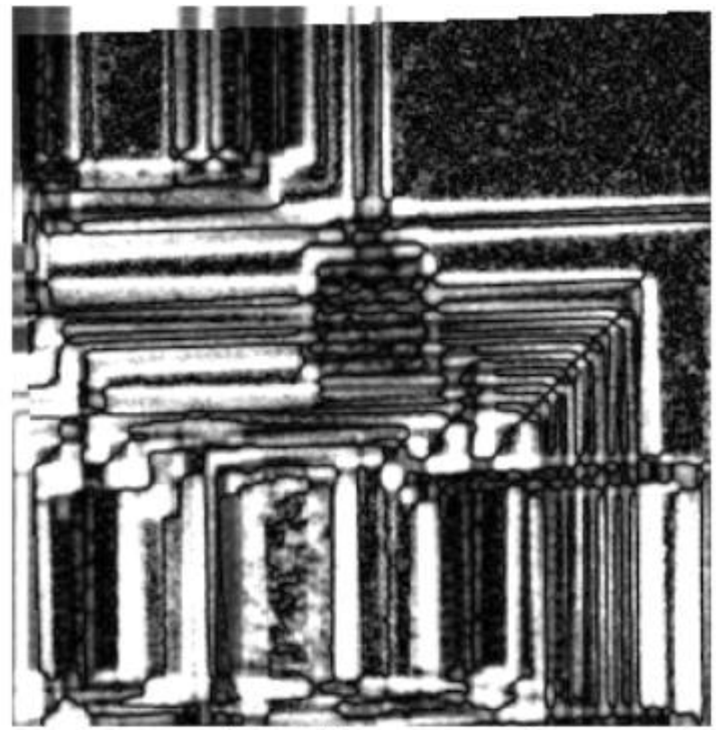


- Start with coarsely spaced candidate displacements
- Smaller pattern when best match is in the middle
- Stop when desired displacement accuracy is reached

Absolute Difference bet. images

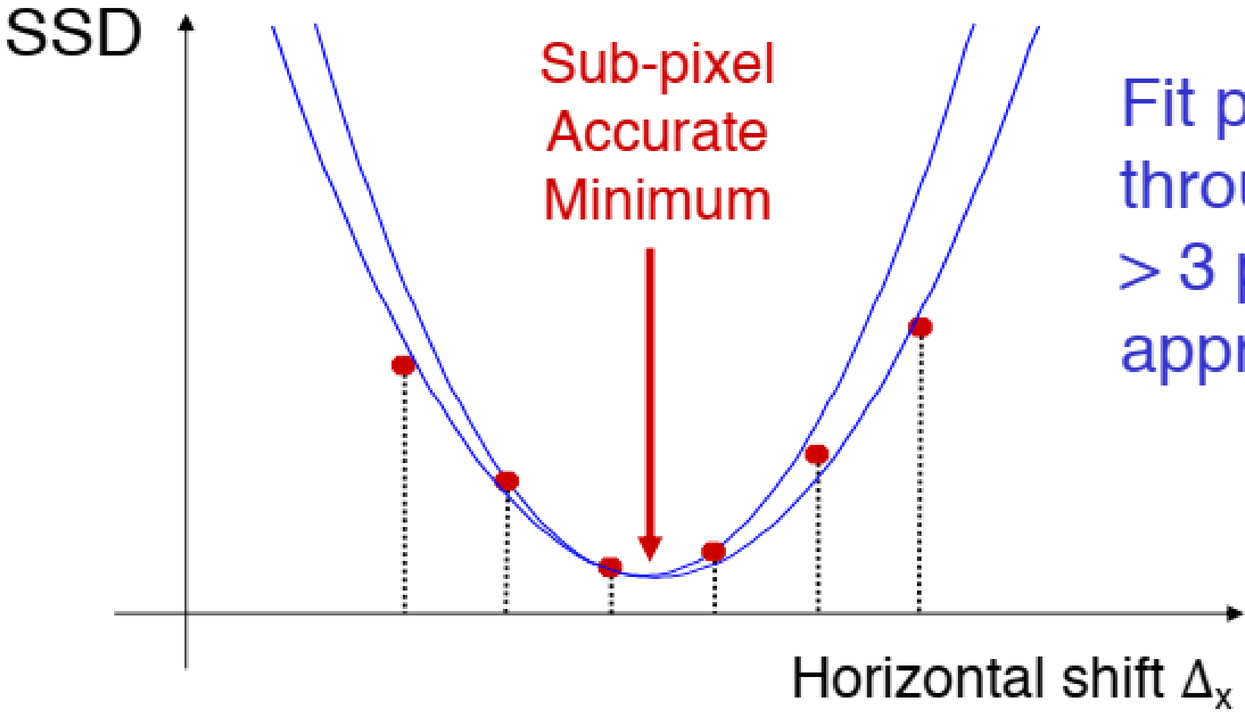


w/ integer-pixel alignment



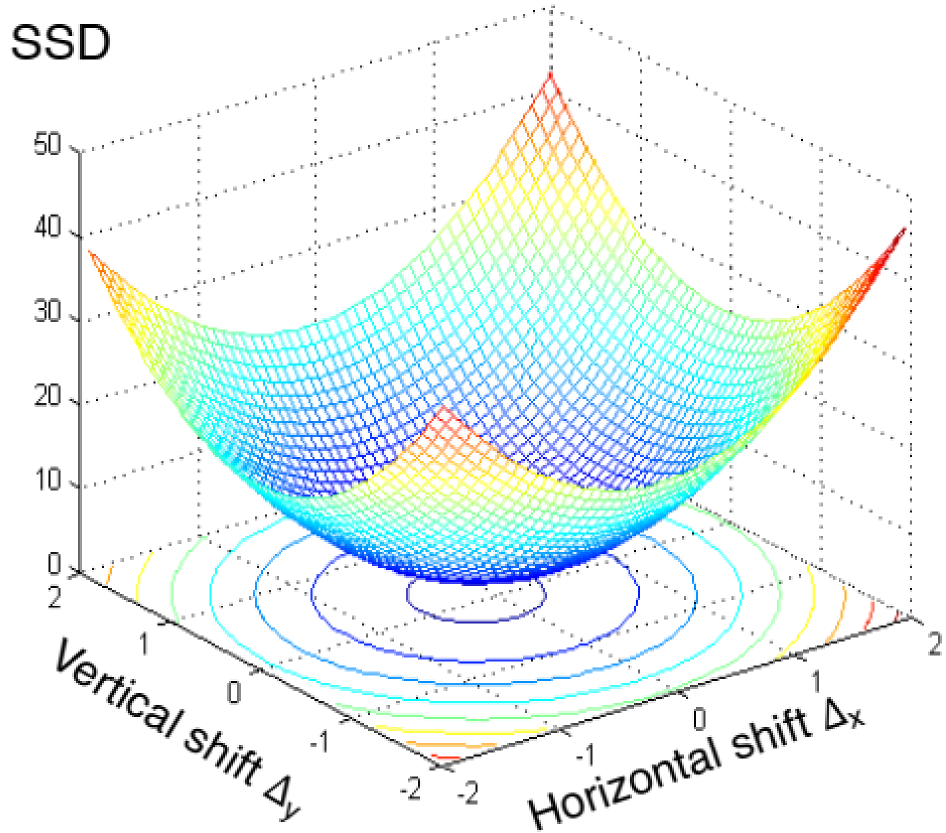
w/o alignment

Interpolation of the SSD Minimum



Fit parabola through > 3 points approximately

2D Interpolation of SSD Minimum

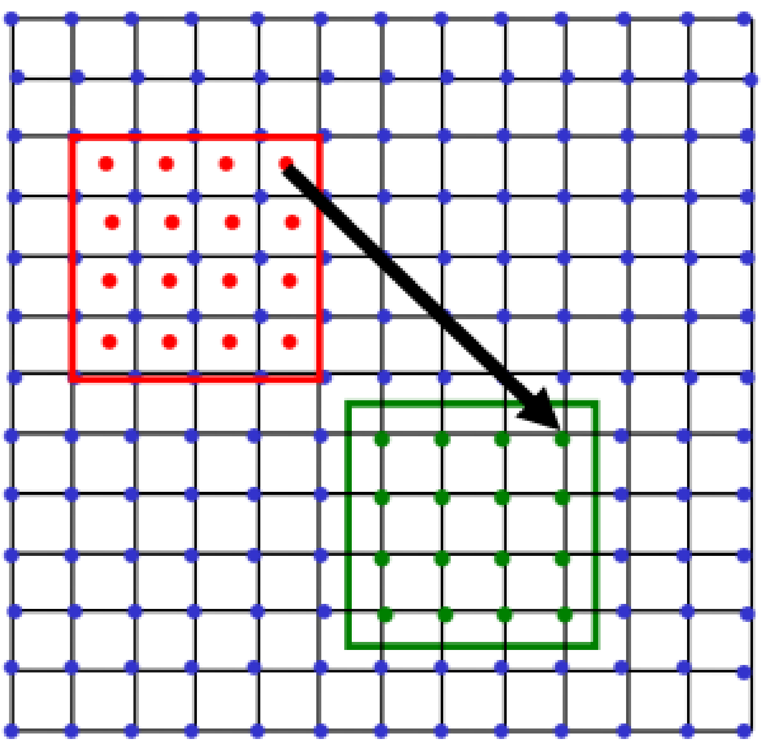


<u>Paraboloid</u>

- Perfect fit through 6 points
- Approximate fit through > 6 points

Sub-pixel Accuracy

- Interpolate pixel raster of the reference image to desired sub-pixel accuracy (e. g., by bi-linear or bi-cubic interpolation)
- Straightforward extension of displacement vector search to fractional accuracy
- Example: half-pixel accurate displacements



$$\begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix} = \begin{pmatrix} 4.5 \\ 4.5 \end{pmatrix}$$