comp 471 / cart 498c

Operations on Monochrome Images

Monday 18 Sep 06

Monochrome Operations

- Linear Point Operations
- Nonlinear Point Operations
- Histogram Shaping and Matching
- Arithmetic Image Operations
- Geometric Image Operations

Vector Notation

- A vector is a one-dimensional array. Vectors are assumed columns (N x 1). The unity vector is:

$$e = \begin{bmatrix} 1 \\ . \\ . \\ . \\ 1 \\ 1 \end{bmatrix} \qquad \text{(N x 1) matrix}$$

- The transpose is a row vector (1 x N), and is denoted

$$e^T = [1 \;\; 1 \;\; 1 \ldots 1]$$

Image Notation

- A digital image is an array or matrix of values, a mapping

  $F\colon\ \mathbb{R} \times \mathbb{R} \longrightarrow V$ or $F\colon\ \mathbb{Z} \times \mathbb{Z} \longrightarrow V$

  Typically $V = \mathbb{Z}_{256}$ or $\mathbb{Z}_{256} \times \mathbb{Z}_{256} \times \mathbb{Z}_{256} \times \mathbb{Z}_{256}$

- Denote an image matrix

$$\mathit{F} = [\mathit{F}(\mathrm{i}, \mathrm{j});\ 0 \le \mathrm{i} \le \mathrm{M}\text{-}1,\ 0 \le \mathrm{j} \le \mathrm{N}\text{-}1]$$

$$\begin{pmatrix} F[0,0] & F[0,1] & \ldots & F[0, \mathrm{N}\text{-}1] \\ F[1,0] & F[1,1] & \ldots & F[1, \mathrm{N}\text{-}1] \\ . & & & . \\ . & & & . \\ F[\mathrm{M}\text{-}1,0] & F[\mathrm{M}\text{-}1,1] & \ldots & F[\mathrm{M}\text{-}1, \mathrm{N}\text{-}1] \end{pmatrix}$$

Basic Matrix Algebra

- Transpose of a matrix:

  $$F^{\mathsf{T}}[\mathrm{i},\mathrm{j}] = F[\mathrm{j},\mathrm{i}]$$

- A symmetric matrix satisfies $F^{\mathsf{T}} = F$
- Recall: inner product of two vectors a and b of the same length (N x 1) is

  $$\boldsymbol{u}\cdot\boldsymbol{v} = \sum_{i=1}^{n} \boldsymbol{u}[i]\boldsymbol{v}[i]$$

  which is a scalar.

Matrix Product

- The matrix product R of two matrices S (MxN) and T (NxP) is of dimensions M x P
- The elements of the matrix product K are:

$$\boldsymbol{R}[i,j] = \sum_{k=1}^{n} \boldsymbol{S}[i,k]\boldsymbol{T}[k,j]$$

- The inner product of the i-th column vector of S^T and the j-th column vector of T.

Matrix Product

- Matrix product does not commute. Generally

  $S \; T \neq T \; S$

  The (N x N) identity matrix I is

$$\begin{pmatrix} 1 & 0 & & . & 0 \\ 0 & 1 & 0 & . & 0 \\ . & & & . & 0 \\ 0 & & . & . & 1 \end{pmatrix}$$

- Set of square matrices of dimension N is a multiplicative group with identity element I:

$$I\,R = R\,I = R$$

for any N x N matrix R.

Matrix Inverse

- The matrix inverse of an N x N matrix R is another N x N matrix denoted R^{-1}
- It satisfies $R R^{-1} = R^{-1}R = I$.
- Note that R^{-1} commutes with R.
- When does R^{-1} exist? When is it stable?
- Of course, $[R^{-1}]^{-1} = R$
- Computing R^{-1} is a laborious process, especially for large matrices.

Histograms

Simple Histogram Operations

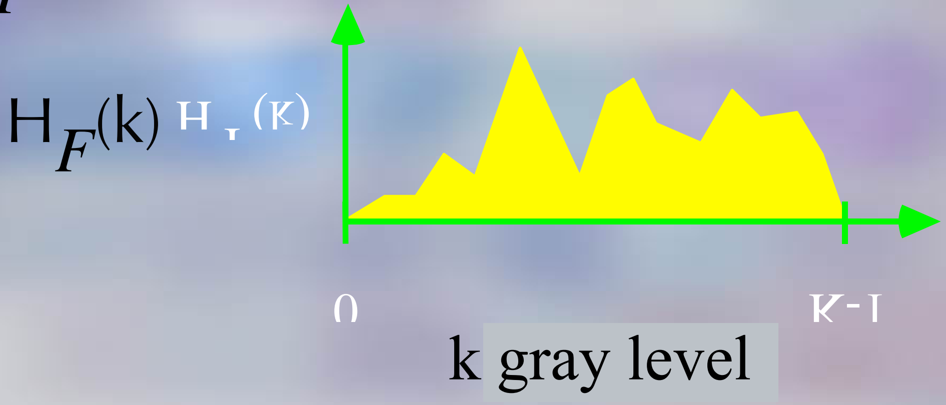
- The gray-level histogram H_F of an image F is a graph of the frequency of occurrence of each gray level in F.
- H_F is a one-dimensional function :

  $H_F : \{0, \ldots , K\text{-}1\} \rightarrow \mathbb{Z}^+$ non-negative integers

  where k_{max} = the number of gray levels (e.g. 255)
- $H_F(k) = n$ if gray-level k occurs (exactly) n times in F, for each k, $0 \leq k \leq k_{max} - 1$

(histogram)

- H_F contains no spatial information - only the relative frequency of intensities.
- Much useful info is obtainable from H_F.
- Image quality is effected (enhanced, modified) by altering H_F.



Average Optical Density (AOD)

- The average intensity of the NxM image I:

$$AOD[\boldsymbol{F}] = \frac{1}{NM} \sum_{i=0}^{M} \sum_{j=0}^{M} \boldsymbol{F}[i,j]$$

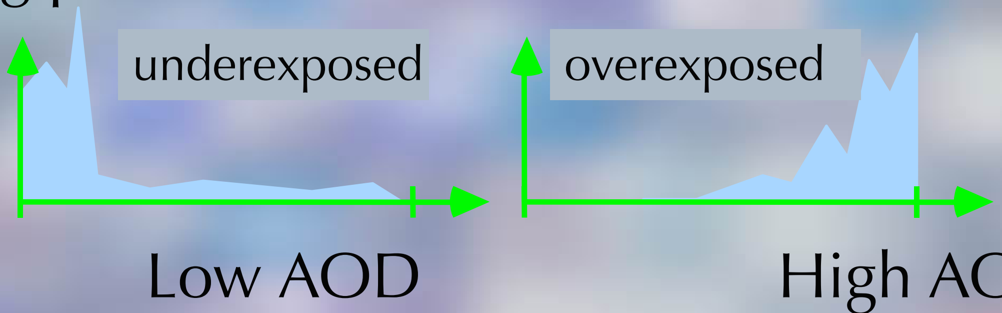
- Can compute it from the histogram as well:

$$AOD[\boldsymbol{F}] = \frac{1}{NM} \sum_{k=0}^{k_{max}} k H_{\boldsymbol{F}}[k]$$

- Exercise: Prove this fact.

Average Optical Density

- Examining the histogram can reveal possible errors in the imaging process:



- By operating on the histogram, such errors can be ameliorated.

Point Operations

- A point operation is a function g: $\mathbb{R} \rightarrow \mathbb{R}$ that operates on single pixels in the image:
- $G(i, j) = g[F(i, j)], 0 \leq i \leq M-1, 0 \leq j \leq N-1$
- The same function g applied at every (i, j).
- Unlike local operations (e.g. OPEN), ***point*** *operations* ***do not use neighboring values*** of F(i, j).
- Point operations don’t modify spatial relationships. They do change the histogram, and therefore the appearance of the image.

Linear Point Operations

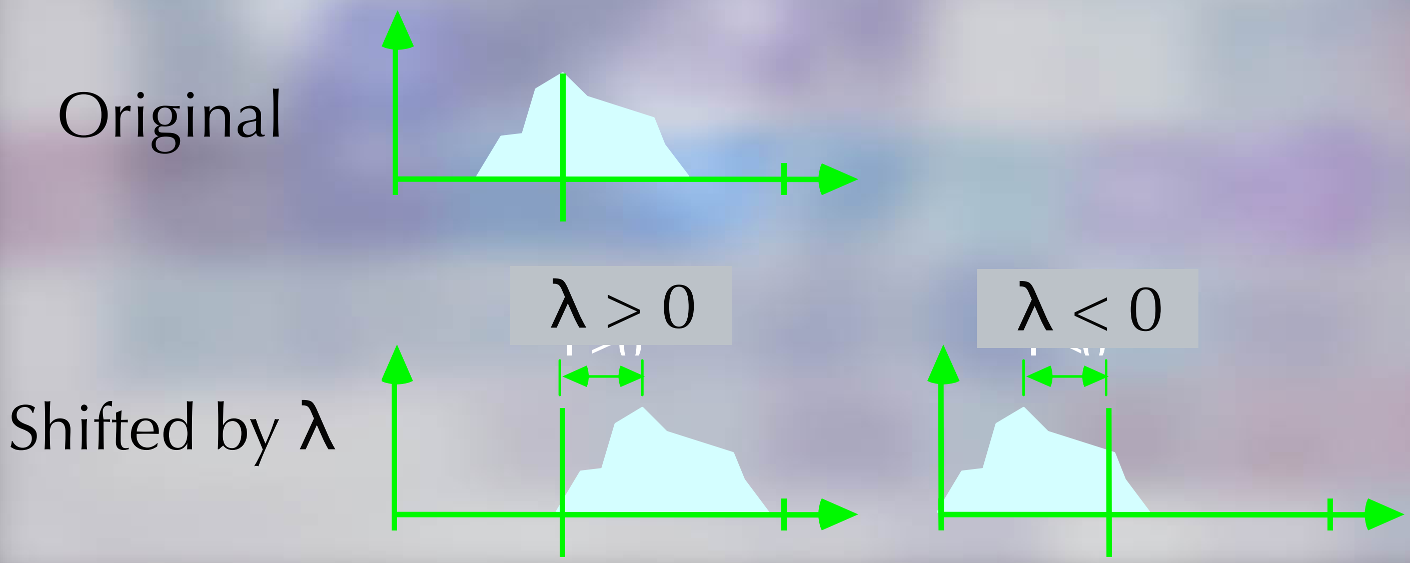
Linear Point Operations

- The simplest class of point operations. They offset and scale the image intensities.
- Suppose $-(k_{max}-1) \leq L \leq k_{max}-1$. An additive image offset is defined by
  $$G(i, j) = F(i, j) + L$$
- Suppose $\lambda > 0$. Image scaling is defined by
  $$G(i, j) = \lambda F(i, j)$$

Image Offset

- If λ > 0, then G is a brightened version of F. If λ < 0, a dimmed version of F.
- The histogram is shifted by amount L:

$$H_G(k) = H_F(k - \lambda)$$



Original

Shifted by λ

λ > 0

λ < 0

- Q. What does the following do to an image?

$$G(i, j) = - F(i, j) + (k_{max}-1)$$

(jitops)

Image Offset Example

- Compare multiple images F_1, F_2 ,..., F_{m_max} of the same scene taken with different exposures or lighting conditions.
- A solution: equalize the image AODs, e.g., set them all to $k_{max}/2$ (for gray scale range 0 ,..., k_{max}-1).
- Let L_m = AOD(F_m), m = 1 ,..., m_{max}. Then equalize via

$$G_m(i, j) = F_m(i, j) - L_m + k_{max}/2$$

AOD Equalization

- Basically, bringing these weighted “beams” to all balnce at the same point $k_{max}/2$

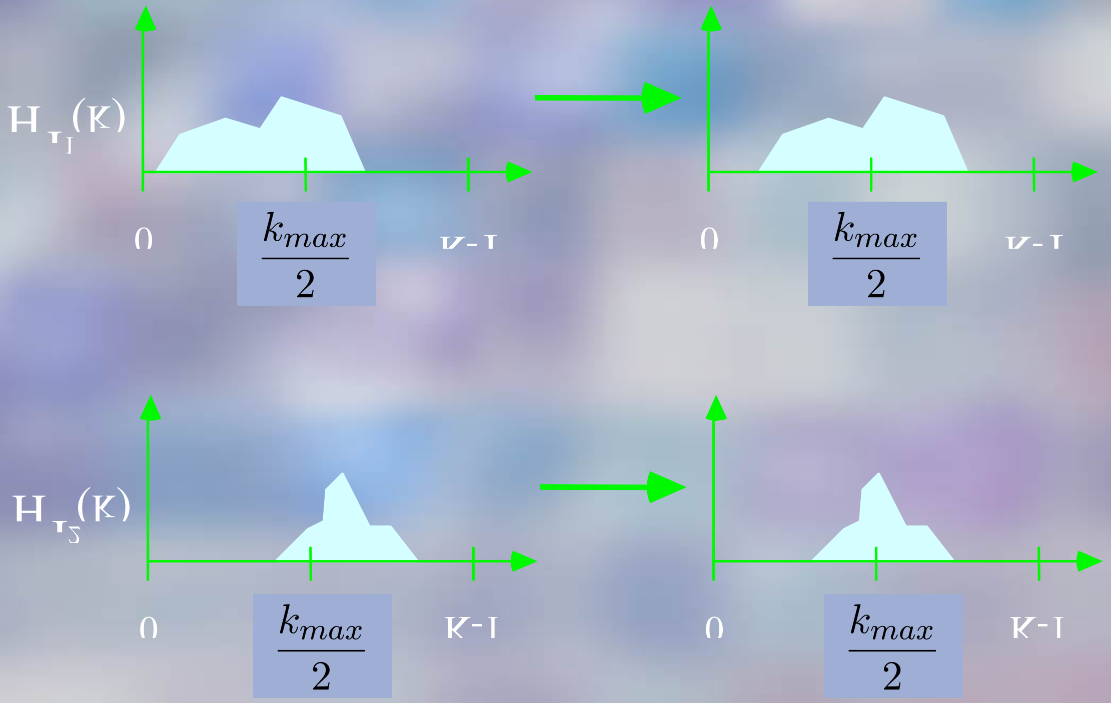


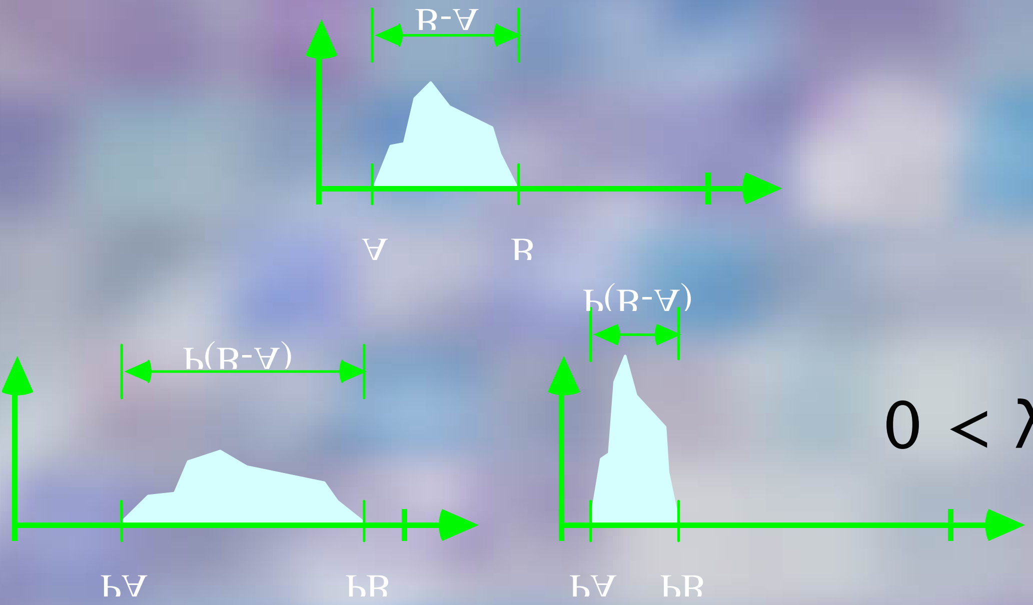
Image Scaling

- $G(i, j) = \lambda \cdot I(i, j)$
- If $\lambda > 1$, the intensity range is widened.
- If $\lambda < 1$, the intensity range is narrowed.
- Multiplying by λ stretches or compresses the image histogram by a factor λ:

$$H_G(k) = H_F(k / \lambda) \qquad \text{(continuous)}$$

$$H_G(k) = H_F[\mathrm{INT}(k/P)] \qquad \text{(discrete)}$$

Image Scaling



$\lambda > 1$

$0 < \lambda < 1$

- An image with a compressed gray level range generally has reduced visibility – a washed out appearance (and vice-versa).

Affine Point Operations: Offset & Scaling

- Given reals λ and μ, an affine ("linear") point operation on F is a function

$$G(i, j) = \mu \cdot F(i, j) + \lambda$$

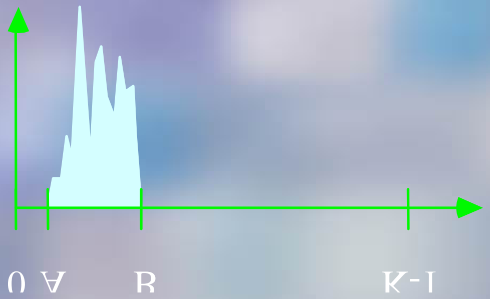
comprising both offset and scaling.

- If μ < 0, the histogram is reversed, creating a negative image. For example, $\mu = -1$, $\lambda = k_{max}-1$:

$$G(i, j) = (k_{max}-1) - F(i, j)$$

Full-Scale Contrast Stretch

- The most common linear point operation. Suppose I has a compressed histogram:



- Let A and B be the min and max gray levels in F. Define

$$G(i, j) = \mu \cdot F(i, j) + \lambda$$

such that $\mu A + \lambda = 0$ and $\mu B + \lambda = (k_{max} - 1)$.

- Q. Can you always find such μ, λ ?

Full-Scale Contrast Stretch

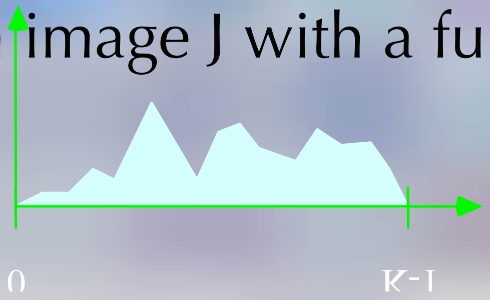
- Solving these 2 equations in 2 unknowns yields:

$$\mu = \frac{k_{max} - 1}{B - A} \qquad \lambda = -A(\frac{k_{max} - 1}{B - A})$$

- so the stretched image becomes

$$G(i, j) = (k_{max}-1)\ [F(i, j) - A] / (B - A)$$

- The result is an image J with a full-range histogram:



Nonlinear Point Operations

Nonlinear Point Operations

- Now consider nonlinear point functions f

$$J(i, j) = f[I(i, j)].$$

- A very broad class of functions!
- Commonly used:

$G(i, j) = |F(i, j)|$ (magnitude)

$G(i, j) = [F(i, j)]^2$ (square-law)

$G(i, j) = \sqrt{}[F(i, j)]$ (square root)

$G(i, j) = \log[1+F(i, j)]$ (logarithm)

$G(i, j) = \exp[F(i, j)] = e^{I(i,j)}$ (exponential)

- Most of these are special-purpose, for example…

Logarithmic Range Compression

- Small groupings of very bright pixels may dominate the perception of an image at the expense of other rich information that is less bright and less visible.

- Astronomical images of faint nebulae and galaxies with dominating stars are an excellent example.

- Rosette Nebula

Logarithmic Range Compression

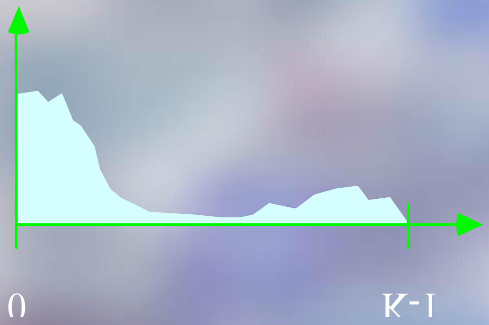
- Logarithmic transformation

$$G(i, j) = \log[1+F(i, j)]$$

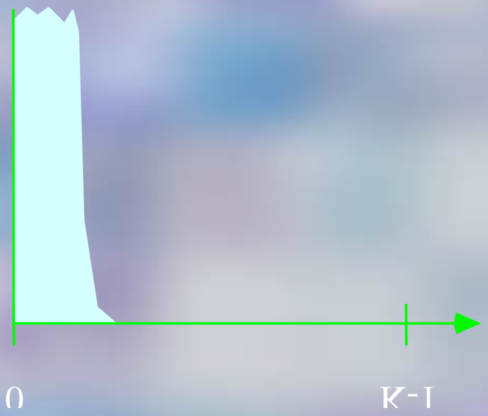
- nonlinearly compresses and equalizes the gray-scales.
- Bright intensities are compressed much more heavily - thus faint details emerge.

Logarithmic Range Compression

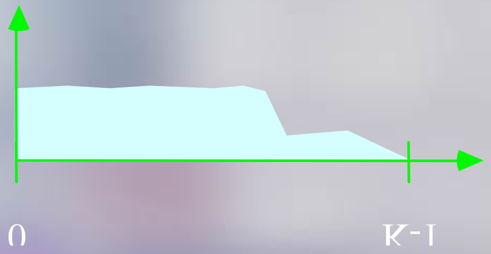
- A full-scale contrast stretch then utilizes the full gray-scale range:



typical histogram

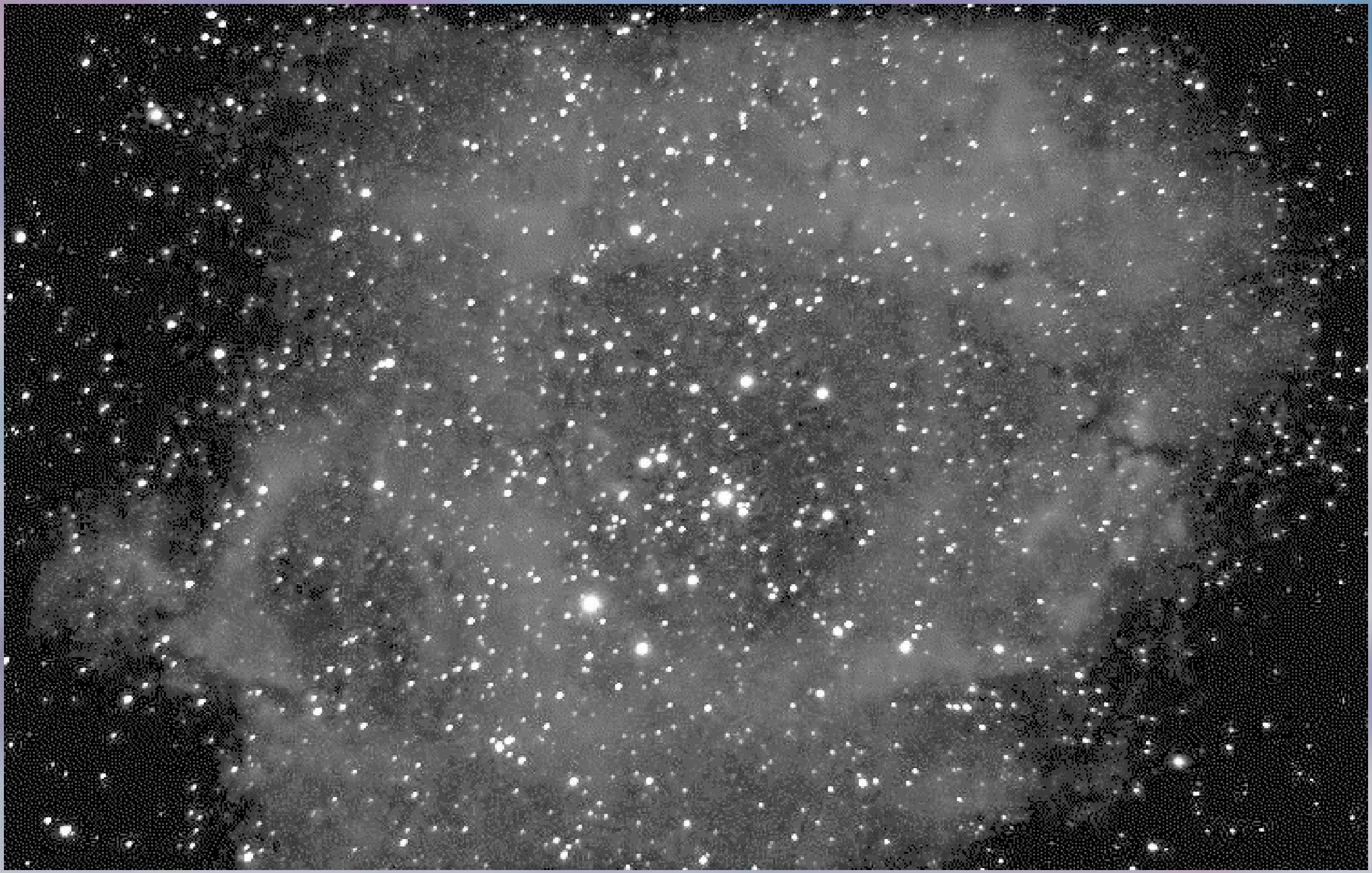


logarithmic transformation

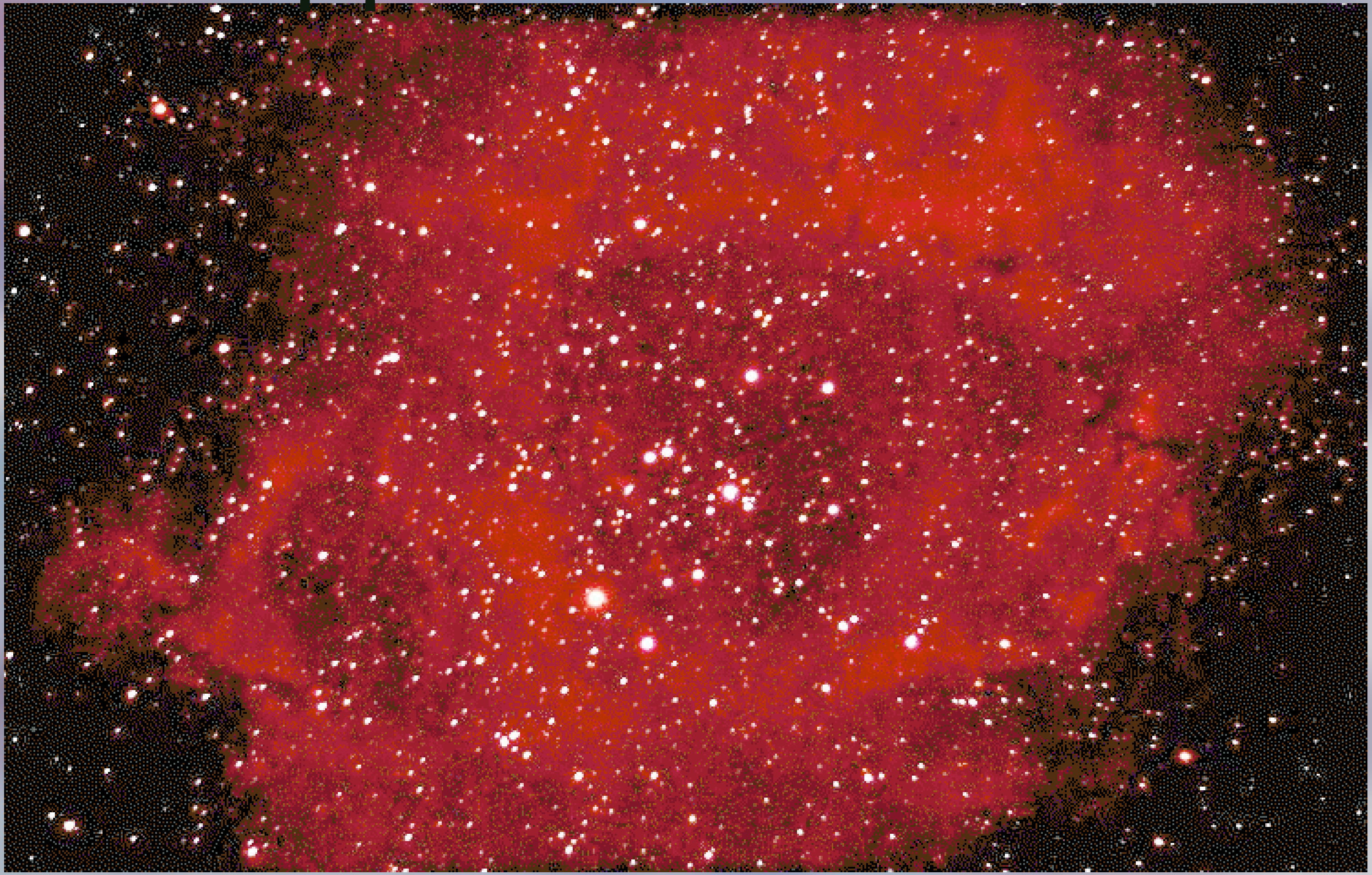


stretched contrast

contrast stretched Rosetta Nebula



mapped to “false” red



- Q. How would you define the “color” of the Rosetta Nebula?

Histogram Shaping

Histogram Distribution

- An image with a flat histogram makes rich use of the available gray-scale range. This might be an image with

  Smooth changes in intensity across many gray levels

  Lots of texture covering many gray levels

- We can obtain an image with an approximately flat histogram using nonlinear point operations.

Histogram Shaping

- Define the normalized histogram:

$$p_F(k) = \frac{1}{MN} H_F(k)$$

- These values $0 \leq p_F(k) \leq 1$ sum to one:

$$\sum_{k=0}^{k_{max}-1} p_F(k) = 1$$

- See: $p_F(k)$ is the probability that gray-level k will occur (at a pixel).

Histogram Shaping

- The cumulative histogram is

$$P_F(r) = \sum_{k=0}^{r} p_F(k)$$

- which is non-decreasing (why?); also, $P_F(k_{max}-1) = 1$.
- Probabilistic interpretation: at any pixel(i,j),

$$P_F(k)(r) = \Pr\{F(i,\ j) \leq r\}$$

$$p_F(r) = P_F(r) - P_F(r-1) \geq 0\ ;\ r = 0, \ldots, k_{max}-1$$

Continuous Histograms

- Suppose p(x) and P(x) are continuous: can regard as probability density (pdf) and cumulative distribution (cdf).
- Then p(x) = dP(x)/dx
- $P^{-1}(x)$ exists or defined by convention.

  Theorem: monotone functions have inverse (a.e.?)
- We'll describe histogram flattening/shaping for the continuous case, then extend to discrete case.

Continuous Flattening and Shaping

- Transform (continuous) image F, with $p_F(x)$, and $P = P_F(x)$ into image K with flat histogram.
- The following image G will have a flattened histogram with range [0, 1]:

$$G = P(F)$$

meaning $G(i, j) = P[F(i, j)]$

Continuous Flattening

- Reason: the cumulative histogram Q of G:

$$Q(x) = \Pr\{G \leq x\} \qquad \text{(at any pixel (i, j)*)}$$

$$= \Pr\{P(F) \leq x\} \qquad \text{(def.)}$$

$$= \Pr\{F \leq P^{-1}(x)\}$$

$$= P[P^{-1}(x)] = x$$

hence $q(x) = dQ(x)/dx = 1$ for $0 < x < 1$

- Finally define image K as full scale contrast stretch of G:

$$K = \mathrm{FSCS}[G]$$

* abuse of notation, interpret at a particular pixel (i,j)

Continuous Shaping

- Suppose we'd like arbitrary q(x), Q(x). Then define image H

$$H = Q^{-1}[P(F)]$$

$$H(i, j) = Q^{-1}\{P[I(i, j)]\} \text{ for all } (i, j)$$

- Since the cumulative histogram of G is:

$$\begin{aligned} \Pr\{H \le x\} &= \Pr\{Q^{-1}[P(F)] \le x\} = \Pr\{P(I) \le Q(x)\} \\ &= \Pr\{F \le P^{-1}[Q(x)]\} = P\{P^{-1}[Q(x)]\} \\ &= Q(x) \end{aligned}$$

- All this can only be approximated with discrete histograms

Discrete Histogram Flattening

- To approximately flatten the histogram of the digital image F:
- Define the cumulative histogram image

$$G = P_F(F)$$

so that

$$G(i, j) = P_F[F(i, j)].$$

- This is the cumulative histogram evaluated at the gray level of the pixel (i, j).

Discrete Histogram Flattening

- Note that

$$0 \leq G(i, j) \leq 1$$

- The elements of J are approximately linearly distributed between 0 and 1.
- Finally, let K = FSCS(J) (full scale contrast stretch of image J) yielding the histogram-flattened image.

Histogram Flattening Example

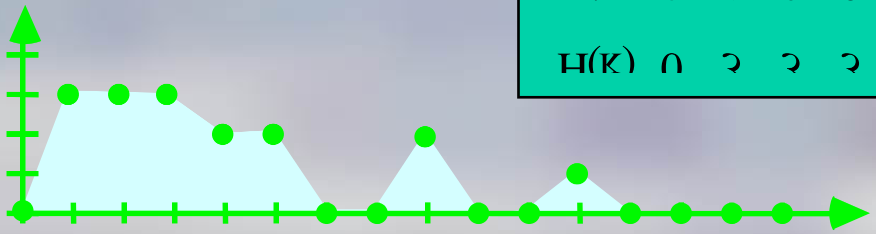
- Given a 4x4 image I with gray-level range {0, ..., 15} (K-1 = 15):

I =

1	1	3	4
2	5	3	2
8	1	8	2
4	5	3	11

- The histogram:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	0	3	3	3	2	2	0	0	2	0	0	1	0	0	0	0



- The normalized histogram…

p(k)	0	16 3	16 3	16 3	16 2	16 2	0	0	16 2	0	0	16 1	0	0	0	0
k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- The intermediate image J is computed followed by the "flattened" image K (after rounding):

- J =

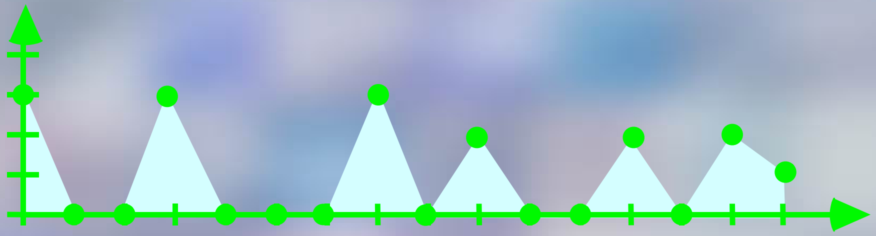
11/16	13/16	9/16	16/16
15/16	3/16	15/16	6/16
6/16	13/16	9/16	6/16
3/16	3/16	9/16	11/16

K =

9	12	7	15
14	0	14	3
3	12	7	3
0	0	7	9

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	3	0	0	3	0	0	0	3	0	2	0	0	2	0	2	1

- The new, flattened histogram looks like this:



- The heights H(k) cannot be reduced - only stacked
- Digital histogram flattening doesn't really "flatten" - just spreads out the histogram – more flat.
- The spaces that appear are characteristic of "flattened" histograms - especially when the original histogram is highly compressed.

Histogram Shaping

- Can create a new image K with an approximate specified histogram shape, such as a triangle or bell-shaped curve.

- Let $\mathrm{H}_K(\mathrm{k})$ be the desired histogram shape, with corresponding normalized values (probabilities) $\mathrm{p}_K(\mathrm{k})$, and cumulative probability distribution function

$$Q_K(y) = \int_{k=0}^{y} p_K(x)dx$$

Discrete cumulative histogram function:

$$Q_K[r] = \sum_{k=0}^{r} p_K(k)$$

Histogram Shaping

- Define the "cumulative histogram" image as before:

$$G = Q^{-1}[P_F(F)]$$

- Then the cumulative histogram of this image G is:

$$P_G(x) = \Pr\{G \leq x\} \qquad \text{(at any pixel (i, j))}$$

$$= \Pr\{Q^{-1}[P_F(F)] \leq x\} \qquad \text{(def.)}$$

$$= \Pr\{P_F(F) \leq Q(x)\}$$

$$= \Pr\{F \leq P^{-1}[Q(x)]\}$$

$$= P[P^{-1}[Q(x)]] = Q(x)$$

as desired.

Histogram Shaping Algorithm

- Need to define an inverse to the cumulative distribution in the discrete case.
- Let n(i, j) denote the minimum value of n such that

$$P_K(n) \geq J(i, j)$$

- Then take K(i, j) = n(i, j).
- This is a convention for

$$n(i, j) = (P_K)^{-1} [J(i,j)]$$

Histogram Shaping Example

- Consider the same image as in the last example. We had

- I =

4	5	3	11
8	1	8	2
2	5	3	2
1	1	3	4

J =

11/16	13/16	9/16	16/16
15/16	3/16	15/16	6/16
6/16	13/16	9/16	6/16
3/16	3/16	9/16	11/16

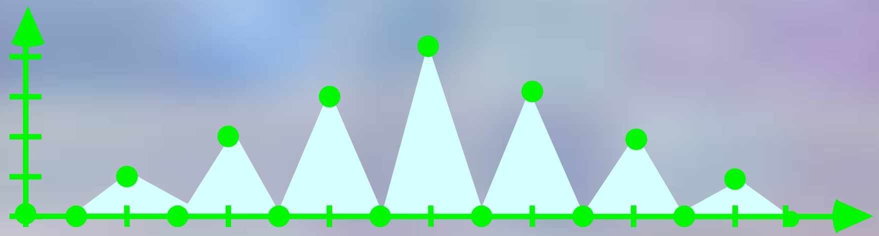
- Fit this to a (triangular) histogram:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$H_K(k)$	0	0	1	0	2	0	3	0	4	0	3	0	2	0	1	0
$P_K(k)$	0	0	$\frac{1}{16}$	0	$\frac{2}{16}$	0	$\frac{3}{16}$	0	$\frac{4}{16}$	0	$\frac{3}{16}$	0	$\frac{2}{16}$	0	$\frac{1}{16}$	0

- The cumulative probabilities:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P_K(n)$	0	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{3}{16}$	$\frac{6}{16}$	$\frac{6}{16}$	$\frac{10}{16}$	$\frac{10}{16}$	$\frac{13}{16}$	$\frac{13}{16}$	$\frac{15}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$

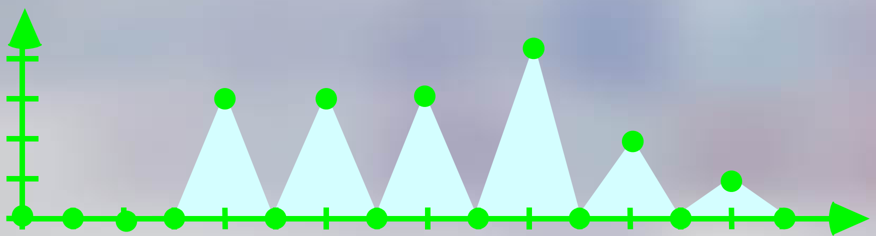
- Visual inspection of J lets us form the new image.



- $$K = \begin{array}{|c|c|c|c|} \hline 10 & 10 & 6 & 14 \\ \hline 12 & 4 & 12 & 8 \\ \hline 8 & 10 & 6 & 8 \\ \hline 4 & 4 & 6 & 10 \\ \hline \end{array}$$

- Here's the new histogram:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$H_K(k)$	0	0	0	0	3	0	3	0	3	0	4	0	2	0	1	0



Histogram Matching

- A special case of histogram shaping.
- The histogram of image I is matched to the histogram of another image I´.
- The procedure is identical, once the cumulative probabilities are computed for the model image I´.
- ***Useful application: Comparing similar images of the same scene obtained under different conditions (e.g., lighting, time of day).*** *Extends the concept of equalizing AOD described earlier.*

Arithmetic Image Operators

Arithmetic Image Operations

- Suppose we have two N x M images I1 and I2. The basic arithmetic operations are:
- Pointwise Addition

  $$J = I1 + I2 \; ; \; J(i, j) = I1(i, j) + I2(i, j)$$

- Pointwise Subtraction

  $$J = I1 - I2 \; ; \; J(i, j) = I1(i, j)\overset{\Delta}{} - I2(i, j)$$

- Pointwise Multiplication

  $$J = I1 \otimes I2 \; ; \; J(i, j) = I1(i, j) \times I2(i, j)$$

- Pointwise Division

  $$J = I1 \, \Delta \, I2 \; ; \; J(i, j) = I1(i, j) / I2(i, j)$$

- The operations ⊗ and Δ are useful when manipulating Fourier Transform matrices.

Applying Arithmetic Operations

- We will look at two simple but important applications of algebraic operations on images:

  - Frame averaging for noise reduction

  - Image differencing for motion detection

Frame Averaging for Noise Reduction

- An image J is often corrupted by additive noise:

  Surface radiation scatter

  Noise in the camera

  Thermal noise in a computer circuit

  Channel transmission noise

- Model: The sum of an original image I and a noise image N:

$$J = I + N,$$

- where the elements N(i, j) of N are random variables.

Zero-Mean Ergodic Noise

- We won't explore the math of random variables (other courses).
- Just assume the noise is zero mean (ergodic): the sample mean of *n* noise matrices q_{m} tends towards a zero matrix as *n* grows large:

$$\frac{1}{n}\sum_{m=1}^{n} q_m \rightarrow 0$$

as

$$n \rightarrow \infty$$

Frame Averaging for Noise Reduction

- Suppose we obtain M images K_1, K_2,... , K_M of the same scene
- in rapid succession, so that there is little motion between frames
- or suppose there is no motion in the scene.
- However, the frames are noisy:

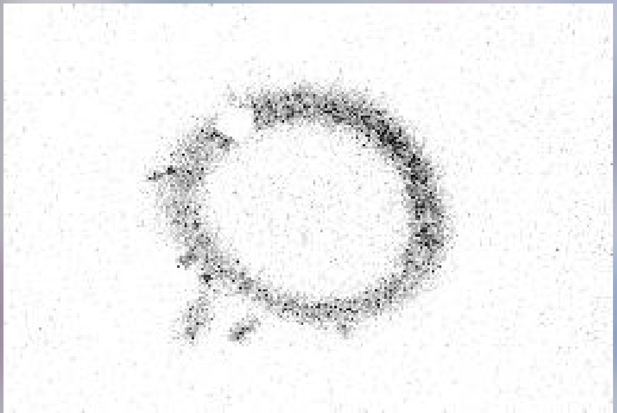
$$K_m = F + q_m \text{ for } i = 1 ,..., M.$$

Frame Averaging for Noise Reduction

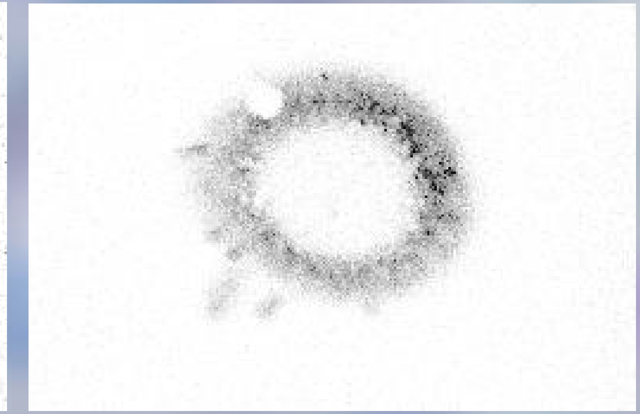
$$
\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m} K_i &= \frac{1}{m}\sum_{i=1}^{m}(F + q_i) \\
&= \frac{1}{m}\sum_{i=1}^{m} F + \frac{1}{m}\sum_{i=1}^{m} q_i \\
&= F + \frac{1}{n}\sum_{i=1}^{m} F_i \\
&\approx F
\end{aligned}
$$

Frame Averaging Example

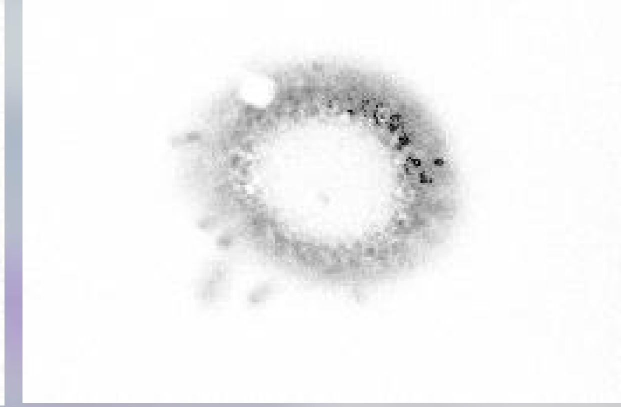
- Macroalga Valonia microphysa, imaged with laser scanning confocal microscope (LSCM). The ring is chlorophyll fluorescing under Ar laser excitation.



Single frame
(no averaging)



Average of
four frames



Average of
sixteen frames.

Motion Detection by Frame Differencing

- Often it is of interest to detect object motion between frames.
- Many applications: video compression, target recognition, tracking, security, surveillance, automated inspection, etc.

Motion Detection by Frame Differencing

- A simple approach:
- Let I1, I2 be consecutive frames in close time proximity, e.g., from a video camera.
- Form the absolute difference image.

$$J = |I1 - I2|$$

- A full-scale contrast stretch will give a more visually dramatic result.

(test_framedifference5)

Geometric Operations

Geometric Image Operations

- Certain geometric image operations widely used in image processing.
- Many concepts also overlap computer graphics.
- Image processing is primarily concerned with correcting or improving images of the *textural* world.
- Computer graphics is primarily concerned with creating images of synthetic world, more based on taxonomy of objects.

Geometric Image Operations

- Geometric image operations complement pointwise operations: they modify spatial positions but not gray levels.
- A geometric operation generally requires two steps:

  - A spatial mapping of image coordinates

  - Interpolation

Spatial Mapping

- The image coordinates are mapped to create the new image:

$$J(i, j) = I[a(i, j), b(i, j)]$$

- The coordinates a(i, j) and b(i, j) are not generally integers!
- For example if

$$a(i, j) = i/\pi \text{ and } b(i, j) = j/\pi$$

then $J(i, j) = I(i/\pi, j/\pi)$ has non-integer coordinates!

- This implies a need for interpolation.
- Example: jit.repos

Interpolation

- It is necessary to interpolate non-integer coordinates a(i, j) and b(i, j) to integer values.
- We will look at two types:

  - Nearest neighbor interpolation

  - Bilinear interpolation

Nearest Neighbor Interpolation

- Simple-minded.
- The geometrically transformed coordinates are modified by mapping to the nearest integer coordinates:

$$a'(i,j) = \mathrm{INT}[a(i, j)+0.5] \quad \text{and} \quad b'(i,j) = \mathrm{INT}[b(i, j)+0.5]$$

$$J(i, j) = I[\ a'[i,j], b'[i,j]\ ]$$

- Serious drawback: Pixel replication can occur, creating a “jagged” edge effect in non-smooth regions.

Nearest Neighbor Interpolation

- Caveat

If for some coordinate (i, j), the index map

$$a'[i,j] \notin [0,M] \quad \text{or} \quad b'[i,j] \notin [0,N]$$

Then use some convention, e.g.

- Usually assign J[i, j] = 0 for these values, or wraparound (What arithmetic function might you use to wrap-around?)

Bilinear Interpolation

- Produces a much smoother interpolation than nearest neighbor approach.
- Given four neighboring image coordinates I(i0, j0), I(i1, j1), I(i2, j2), and I(i3, j3), the new image J(i, j) is computed as

$$J(i, j) = A0 + A1 \cdot i + A2 \cdot j + A3 \cdot i \cdot j$$

- where bilinear weights A0, A1, A2, and A3 are found by solving

$$\begin{bmatrix} A^3 \\ A^2 \\ A^1 \\ A^0 \end{bmatrix} = \begin{bmatrix} 1 & i^3 & j^3 & i^3j^3 \\ 1 & i^2 & j^2 & i^2j^2 \\ 1 & i^1 & j^1 & i^1j^1 \\ 1 & i^0 & j^0 & i^0j^0 \end{bmatrix}^{-1} \begin{bmatrix} I(i^3, j^3) \\ I(i^2, j^2) \\ I(i^1, j^1) \\ I(i^0, j^0) \end{bmatrix}$$

A linear combination of the four closest values. The best planar fit to the four nearest values.

Basic Geometric Transformations

- The basic geometric transformations are
  - Translation
  - Rotation
  - Zoom

Translation

- The simplest geometric operation - requires no interpolation. Let

  $$a(i, j) = i - i0, \qquad b(i, j) = j - j0$$

- where (i0, j0) are constants. In this case

  $$G(i, j) = F(i - i0, j - j0)$$

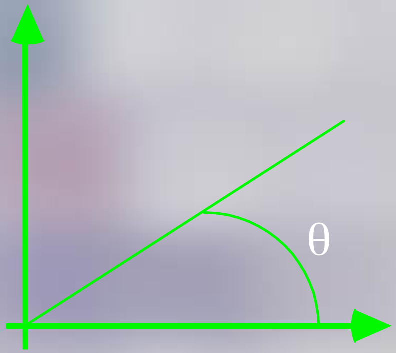
- a shift of the image by amounts (i0, j0) in the (row, column) directions.

Rotation

- Rotation of an image by an angle q relative to the x-axis is accomplished by the following transformation:

$$a(i, j) = i \cos(\theta) - j \sin(\theta)$$

$$b(i, j) = i \sin(\theta) + j \cos(\theta)$$



- Simplest cases:

$$\theta = \pi/2 : [a(i, j), b(i, j)] = (-j, i)$$

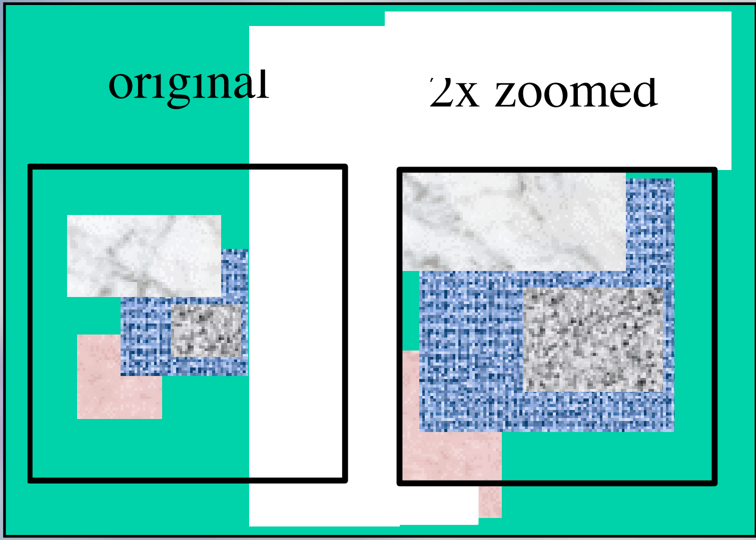
$$\theta = \pi : [a(i, j), b(i, j)] = (-i, -j)$$

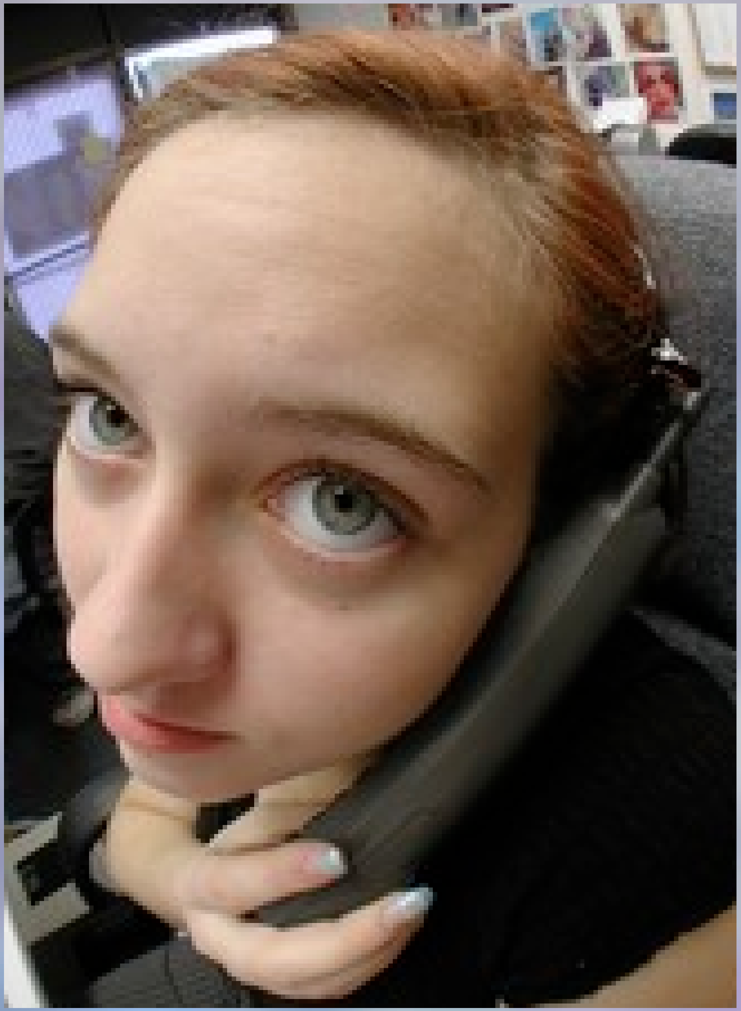
$$\theta = -\pi/2 : [a(i, j), b(i, j)] = (j, -i)$$

- A translation is usually require afterwards to obtain coordinate values in the nominal range.

Zoom

- Zooming magnifies an image by the mapping functions
- $a(i, j) = i / c$ and $b(i, j) = j / d$
- where $c \geq 1$ and $d \geq 1$.
- For large magnifications,
- A zoom image will look "blotchy" if nearest neighbor interpolation is used.
- A bilinear interpolation works quite well.





- “Fish-Eye” geometric distortion (more complex).
- Challenge: Can you define such a deformation?

- *(jit.repos-distortquad.pat)*

Wednesday

- Morphological Operators on Binary Images
    - Erode
    - Dilate
- Examples