# EECE 253 Image Processing 

Lecture Notes: Introduction and Overview

Richard Alan Peters II<br>Department of Electrical Engineering<br>and Computer Science<br>Fall Semester 2014

Vanderbilt University School of Engineering

# Introduction and Overview 

This presentation is an overview of some of the ideas and techniques to be covered during the course.

## EECE 253 Image Processing

## Topics

1. Image formation
2. Point processing and equalization
3. Color correction
4. The Fourier transform
5. Convolution
6. Image sampling, warping, and stitching
7. Spatial filtering
8. Noise reduction
9. Mathematical morphology
10. High dynamic range imaging
11. Image compression

EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


## Digital Image Formation: Quantization



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Sampling and Quantization


real image

sampled

quantized

sampled \& quantized

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Digital Image

Color images have 3 values per pixel; monochrome images have 1 value per pixel.

## a grid of squares, each of which contains a single color


each square is called a pixel (for picture element)


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Color Images

। Are constructed from three intensity maps.
। Each intensity map is pro-jected through a color filter (e.g., red, green, or blue, or cyan, magenta, or yellow) to create a monochrome image.

- The intensity maps are overlaid to create a color image.
- Each pixel in a color image is a three element vector.


EECE 253 Image Processing

## Vanderbilt University School of Engineering



## Point Processing



- gamma

histogram mod

- brightness

- contrast

original

original

+ brightness

+ contrast

+ gamma

histogram EQ


## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Color Processing

> requires some knowledge of how we see colors


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

# EECE 253 Image Processing 

Vanderbilt University School of Engineering

## COSS: The human retina

COSS: The human retina
1 https://www.youtube.com/watch?v=-zzRamRKKdc
1 https://www.youtube.com/watch?v=nbwPPcwknPU
COSS: Image formation
1 https://www.youtube.com/watch?v=HGVUVFcyc6o

EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Eye’s Light Sensors





EECE 253 Image Processing
Vanderbilt University School of Engineering

## Color Sensing / Color Perception



## Color Sensing / Color Perception



The simultaneous red + blue response causes us to perceive a continuous range of hues on a circle. No hue is greater than or less than any other hue.

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Color Sensing / Color Perception



The brain transforms RGB into separate brightness and color channels (e.g., LHS).
brain photo receptors

Color Perception $16 \times$ pixelization of:

EE luminance and chrominance (hue+saturation) are perceived Vanc with different resolutions, as are red, green and blue.


Color Perception $16 \times$ pixelization of:

EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Color Balance and Saturation

Uniform changes in color components result in change of tint.
E.g., if all G pixel values are multiplied by $\alpha>1$ then the image takes a green cast.


## Color Transformations



Image aging: a transformation, $\Phi$, that mapped:

$$
\left[\begin{array}{l}
17 \\
122 \\
114
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
17 \\
121 \\
171
\end{array}\right]\right\} \quad\left[\begin{array}{l}
222 \\
222 \\
185
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
222 \\
222 \\
218
\end{array}\right]\right\} \quad\left[\begin{array}{l}
240 \\
171 \\
103
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
240 \\
171 \\
160
\end{array}\right]\right\} \quad\left[\begin{array}{l}
236 \\
227 \\
106
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
240 \\
230 \\
166
\end{array}\right]\right\}
$$

EECE 253 Image Processing

## The 2D Fourier Transform of a Digital Image

Let $I(r, c)$ be a single-band (intensity) digital image with $R$ rows and C columns. Then, $I(r, c)$ has Fourier representation
$I(r, c)=\sum_{u=0}^{R-1} \sum_{v=0}^{C-1} \mathscr{G}(u, v) e^{+i 2 \pi\left(\frac{u r}{R}+\frac{v c}{C}\right)}$,
where

$$
\mathfrak{G}(u, v)=\frac{1}{R C} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} I(r, c) e^{-i 2 \pi\left(\frac{u r}{R}+\frac{v c}{C}\right)}
$$

are the $R$ x $C$ Fourier coefficients.

EECE 253 Image Processing
Vanderbilt University School of Engineering

## 2D Sinusoids: <br> $$
I(r, c)=\frac{A}{2}\left\{\cos \left[\frac{2 \pi}{\lambda}\left(\frac{c}{\mathrm{C}} \cos \theta-\frac{r}{\mathrm{R}} \sin \theta\right)+\varphi\right]+1\right\}
$$

... are plane waves with grayscale amplitudes, periods in terms of lengths, ...


## 2D Sinusoids:

... specific orientations, and phase shifts.


EECE 253 Image Processing
Vanderbilt University School of Engineering

## The Value of a Fourier Coefficient ...



> ... is a complex number with a (real)part and an imaginary part.

If you represent that number as a magnitude, $A$, and a phase, $\phi$, ...
..these represent the amplitude and offset of the sinusoid with frequency $\omega$ and direction $\theta$.

EECE 253 Image Processing
Vanderbilt University School of Engineering

## The Sinusoid from the Fourier Coeff. at (u,v)



EECE 253 Image Processing
Vanderbilt University School of Engineering

## The Fourier Transform of an Image



I

$|\mathscr{F}\{I\}|$

$\angle[\mathscr{F}\{I\}]$

## Continuous Fourier Transform



## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Discrete Fourier Transform

The discrete Fourier transform assumes a digital image exists on a closed surface, a torus.


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Convolution

Sums of shifted and weighted copies of images or Fourier transforms.


## Sum times $1 / 5$

## EECE 253 Image Processing

## Convolution Property of the Fourier Transform

Let functions $f(r, c)$ and $g(r, c)$ have
Fourier Transforms $F(u, v)$ and $G(u, v)$.
Then,

$$
\mathscr{F}\{f * g\}=F \cdot G .
$$

Moreover,

$$
\mathscr{F}\{f \cdot g\}=F * G .
$$

* represents convolution
- represents pointwise multiplication

Then, a spatial convolution can be computed by

$$
f * g=\mathscr{F}^{-1}\{F \cdot G\}
$$

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Sampling, Aliasing, \& Frequency Convolution

$$
\operatorname{samp}_{l / N}(u, v)==_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta\left(u-\frac{j}{N}\right) \delta\left(v-\frac{k}{N}\right)
$$

$$
\operatorname{samp}_{J / N}(u, v)=\sum_{j=\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta\left(u-\frac{j}{N}\right) \delta\left(v-\frac{k}{N}\right)
$$


aliasing (the jaggies)

no aliasing (smooth lines)

## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Sampling, Aliasing, \& Frequency Convolution

(a) aliased
(b) power spectrum
(c) unaliased
(d) power spectrum


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Resampling



## (resizing)



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Rotation



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Image Warping



## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Panorama via Overlay

Originals


Merged*


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Panorama via Stitching

Originals


EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Frequency Domain (FD) Filtering

Image size: $512 \times 512$ SD filter sigma $=8$


Original Image

## Power Spectrum

Gaussian LPF in FD

EECE 253 Image Processing

## Vanderbilt University School of Engineering

FD Filtering: Lowpass

Image size: $512 \times 512$ SD filter sigma $=8$


Filtered Image
Filtered Power Spectrum


Original Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

FD Filtering: Highpass
Image size: $512 \times 512$ FD notch sigma $=8$


Filtered Image
Filtered Power Spectrum
Original Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

## FD Filtering: Highpass

Image size: $512 \times 512$ FD notch sigma $=8$


Filtered Image
Filtered Power Spectrum
Original Image

EECE 253 Image Processing

Vanderbilt University School of Engineering

## Spatial Filtering


blurred

original

sharpened

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Spatial Filtering


bandpass
filter

original

unsharp
masking

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Spatial Filtering

signed image with 0 at middle gray

bandpass
filter

original

unsharp
masking

EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Noise Reduction


blurred image

color noise

color-only blur

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Noise Reduction


blurred image

color noise

$5 \times 5$ Wiener filter

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Noise Reduction



## periodic noise


original

frequency tuned filter

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Shot Noise or Salt \& Pepper Noise



+ shot noise

s\&p noise

- shot noise

EECE 253 Image Processing

Vanderbilt University School of Engineering

## Nonlinear Filters: the Median


original

s\&p noise

median filter

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Nonlinear Filters: Min and Maxmin



+ shot noise

min filter

maxmin filter

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Nonlinear Filters: Max and Minmax



- shot noise

max filter

minmax

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Nonlinear Processing: Binary Morphology


"L" shaped SE
O marks origin


Foreground: white pixels Background: black pixels


Cross-hatched pixels are indeterminate.

## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Nonlinear Processing: Binary Reconstruction

। Used after opening to grow back pieces of the original image that are connected to the opening.
। Permits the removal of small regions that are disjoint from larger objects without distorting the small features of the large objects.

original

opened

reconstructed

## Nonlinear Processing: Grayscale Morphology


"L" shaped SE
O marks origin Foreground: white pixels Background: black pixels


Cross-hatched pixels are indeterminate.

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Nonlinear Processing: Grayscale Reconstruction



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Image Compression



Yoyogi Park, Tokyo, October 1999. Photo by Alan Peters.

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Image Compression: JPEG



EECE 253 Image Processing
Vanderbilt University School of Engineering
Image Compression: JPEG


EECE 253 Image Processing

## Image Compositing

, Combine parts from separate images to form a new image. It's difficult to do well.
Requires relative positions, orientations, and scales to be correct.
Lighting of objects must be consistent within the separate images.
। Brightness, contrast, color balance, and saturation must match.
Noise color, amplitude, and patterns must be seamless.

# EECE/CS 253 Image Processing 

Lecture Notes: Digital Images and Matlab

Richard Alan Peters II<br>Department of Electrical Engineering and<br>Computer Science

Fall Semester 2014

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Digital Image

Color images have 3 values per pixel; monochrome images have 1 value per pixel.

```
a grid of squares, each of which contains a single color
```


each square is called a pixel (for picture element)


## Pixels

, A digital image, $\mathbf{I}$, is a mapping from a 2D grid of uniformly spaced discrete points, $\{\mathbf{p}=(r, c)\}$, into a set of positive integer values, $\{\mathbf{I}(\mathbf{p})\}$, or a set of vector values, e.g., $\left\{[\mathbf{R G B}]^{\top}(\mathbf{p})\right\}$.
, At each column location in each row of $\mathbf{I}$ there is a value.
The pair ( $\mathbf{p}, \mathbf{I}(\mathbf{p})$ ) is called a "pixel" (for picture element).

## Pixels

- $\mathbf{p}=(r, c)$ is the pixel location indexed by row, $r$, and column, $c$.
$\mathbf{I}(\mathbf{p})=\mathbf{I}(r, c)$ is the value of the pixel at location $\mathbf{p}$.
, If $\mathbf{I}(\mathbf{p})$ is a single number then $\mathbf{I}$ is monochrome.
। If $\mathbf{I}(\mathbf{p})$ is a vector (ordered list of numbers) then I has multiple bands (e.g., a color image).

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Pixels



## Pixel Location: $\mathbf{p}=(r, c)$ Pixel Value: $\mathbf{I}(\mathbf{p})=I(r, c)$

## Pixel : [ $\mathbf{p}, \mathbf{I}(\mathbf{p})$ ]

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Pixels

## Pixel : [ $\mathbf{p}, \mathbf{I}(\mathbf{p})]$



$$
\begin{aligned}
\mathbf{p} & =(r, c) \\
& =(\text { row } \#, \text { col } \#) \\
& =(272,277)
\end{aligned}
$$

$$
\mathbf{I}(\mathbf{p})=\left[\begin{array}{c}
\text { red } \\
\text { green } \\
\text { blue }
\end{array}\right]=\left[\begin{array}{l}
12 \\
43 \\
61
\end{array}\right]
$$

EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering

continuous colors, discrete locations.
discrete realvalued image

## Sampling and Quantization


sampled

quantized

sampled \& quantized

## Sampling and Quantization



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Sampling

Take the average within each square.

$I_{C}(\rho, \chi)$
continuous image

$$
I_{S}(r, c)=\frac{1}{\Delta^{2}} \int_{r \Delta}^{(r+1) \Delta(c+1) \Delta} \int_{c \Delta} I_{C}(\rho, \chi) \delta \rho \delta \chi
$$



I

$I_{S}(r, c)$
sampled image

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Sampling

Take the average within each square.

$I_{C}(\rho, \chi)$
continuous image

$$
I_{S}(r, c)=\frac{1}{\Delta^{2}} \int_{r \Delta}^{(r+1) \Delta(c+1) \Delta} \int_{c \Delta} I_{C}(\rho, \chi) \delta \rho \delta \chi
$$



I

$I_{S}(r, c)$
sampled image

## EECE 253 Image Processing

Vanderbilt University School of Engineering

## Sampling

Take the average within each square.

$I_{C}(\rho, \chi)$
continuous image

$$
I_{S}(r, c)=\frac{1}{\Delta^{2}} \int_{r \Delta}^{(r+1) \Delta(c+1) \Delta} \int_{c \Delta} I_{C}(\rho, \chi) \delta \rho \delta \chi
$$

sampled image

# EECE 253 Image Processing 

Vanderbilt University School of Engineering

## Sampling

Take the average within each square.

$I_{C}(\rho, \chi)$
continuous image

$$
I_{S}(r, c)=\frac{1}{\Delta^{2}} \int_{r \Delta}^{(r+1) \Delta} \int_{c \Delta}^{(c+1) \Delta} I_{C}(\rho, \chi) \delta \rho \delta \chi
$$

sampled image

## Digital Image Formation: Quantization



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Read a Truecolor Image into Matlab



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Read a Truecolor Image into Matlab



EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Read - Figure 1 File Edit View Insert Tools Desktop Window Help

4. Command Window

File Edit Debug Desktop Wind
To get start
$\gg \mathrm{Cd}$ ' $\mathrm{E}: \backslash i m a g e$
$\gg I=$ imread ( $\gg c l a s s(I)$
ans =
uint 8
>> size (I)
ans $=$
600
>> figure
$\gg$ image (I)
$\gg$ title('Les
$\gg$ xlabel (' Pho
$\gg$ truesize
$\gg$


## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Read $\tau^{\circ}$ 

A. Command Window

File Edit Debug Desktop Wind


Les Boingeoisie: The Boing-Boing Bloggers


EECE 253 Image Processing

## Vanderbilt University School of Engineering

Crop t



## EECE 253 Image Processing

Vanderbilt University School of Engineering


## EECE 253 Image Processing <br> Vanderbilt University School of Engineering

## 

## d. Command Window

$-\square x$ oing Bloggers


EECE 253 Image Processing

## Vanderbilt University School of Engineering



## d. Command Window

File Edit Debug Desktop wind

| To get |
| :---: |
| >> Cd 'E: \imag |
| >> I = imread( |
| >> class(I) |
| ans = |
| uint8 |
| >> size(I) |
| ans = |
| 600 |
| >> figure |
| >> image (I) |
| > |

$\gg$ xlabel('Phot
>> truesize
$\gg J=I(125: 42$
>> figure
$\gg$ image (J)
>> truesize
$\gg$ figure (1)
$\gg$

$\square$

EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Crop the Image

```
#. Command Window 
>> cd 'E:\images\Animals\People\Famous'
>> I = imread('Les_Boingeoisie.jpg','jpg');
>> class(I)
ans =
uint8
>> size(I)
ans =
    600
    1200
>> figure
>> image(I)
>> title('Les Boingeoisie: The Boing-Boing Bloggers')
>> xlabel('Photo: Bart Nagel, 2006, WWW.bartnagel.com')
>> truesize
>> J = I(125:425,700:1050,:);
>> figure
>> image(J)
>> truesize
>> figure(1)
>> close
>>
then type 'close' at the prompt.
```

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Read a Colormapped Image into Matlab



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Read a Colormapped Image into Matlab



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Colormapped vs. Truecolor in Matlab



## EECE 253 Image Processing

## Vanderbilt University School of Engineering



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Colormapped vs. Truecolor in Matlab



EECE 253 Image Processing
Vanderbilt University School of Engineering

## Colormapped vs. Truecolor in Matlab




EECE 253 Image Processing
Vanderbilt University School of Engineering

## Example truecolor and colormapped images



24-bit truecolor


8-bit colormapped to 24 bits

EECE 253 Image Processing
Vanderbilt University School of Engineering

## Example truecolor and colormapped images



24-bit truecolor


8-bit colormapped to 24 bits

## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Colormapped Image, Indices, \& Color Map

>> [M, CMAP] = imread('button_mapped.bmp','bmp);

Indices contained in $M(254: 258,254: 258)$

actual values in $\operatorname{CMAP}(109: 113$, :)




## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## How to convert a colormapped image to true color

$M$ is a $512 \times 512 \times 1,8$-bit image. It has 262,144 pixels.
Each pixel has a value between 0 \& 255.
cmap is the colormap that is stored in 'button_mapped.bmp' along with image. cmap is a $256 \times 3$ type-double matrix, each row of which lists a color in terms of its R, G, \& B intensities, which are given as fractions between 0 and 1.

>> [M, cmap] = imread(‘button_mapped.bmp',‘bmp');
$\gg T=$ uint8 $(\operatorname{reshape}(\operatorname{cmap}(M+1,:),[\operatorname{size}(M) 3]) * 255)$;
[512 512]

The $262,144 \times 3$ matrix of intensity values is reshaped into a $512 \times 512 \times 3$ image of type double. The values are scaled to lie between 0 \& 255 then converted to type uint8.

By concatenating M's columns, Matlab rearranges $M$ into a $262,144 \times 1$ list. Each number in the list (if it has 1 added to it) refers to a row of the colormap. Then, cmap( $M+1$,: ) produces a $262,144 \times 3$ matrix of intensity values of type double between $0 \& 1$.

## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## How to Make Colormaps



This code, 0:255, generates a 1 row by 256 element vector of class double that contains numbers 0 through 255 inclusive.

This, (0:255)' , has the same contents and class but is a 256 row by 1 column vector. The apostrophe (') is the matrix transpose operator.
$R, G, \& B$ bands of a truecolor image displayed with grayscale colormaps

## EECE 253 Image Processing

Vanderbilt University School of Engineering


$R, G, \& B$ bands of a truecolor image displayed with grayscale colormaps

## EECE 253 Image Processing

Vanderbilt University School of Engineering

$R, G, \& B$ bands of a truecolor image displayed with tinted colormaps

EECE 253 Image Processing
Vanderbilt University School of Engineering

$R, G, \& B$ bands of a truecolor image displayed with tinted colormaps

EECE 253 Image Processing
Vanderbilt University School of Engineering

$R, G, \& B$ bands of a truecolor image displayed with grayscale colormaps

## EECE 253 Image Processing

Vanderbilt University School of Engineering


EECE 253 Image Processing
Vanderbilt University School of Engineering

## Saving Images as Files

```
| Command Window 
```


## Assuming that

'I' contains the image of the correct class,
that
'cmap' is a colormap, and that
'image_name' is the
file-name that you want.

EECE 253 Image Processing
Vanderbilt University School of Engineering


Jim Woodring - Bumperillo


## Double Exposure: Adding Two Images

Mark Rayden - The Ecstasy of Cecelia

## Double Exposure: Adding Two Images

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round(((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```


## Vanderbilt University School of Engineering

## Double Exposure: Adding Two Images

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round(((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```


## Vanderbilt University School of Engineering

## Double Exposure: Adding Two Images

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8(((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```


## Vanderbilt University School of Engineering

## Double Exposure: Adding Two Images

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8(double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:)) double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
conversions.
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```

EECE 253 Image Processing
Vanderbilt University School of Engineering


Jim Woodring - Bumperillo


## Intensity Masking: Multiplying Two Images

Mark Rayden - The Ecstasy of Cecelia

## Intensity Masking: Multiplying Two Images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round(((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Extacsy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
>> JWtimesMR = double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:)).*double(MR);
>> M = max(JWtimesMR(:));
>> m = min(JWtimesMR(:));
>> JWtimesMR = uint8(255*(double(JWtimesMR)-m)/(M-m));
>> figure
>> image(JWtimesMR)
>> truesize
>> title('EcstasyBumperillo')
```


## Intensity Masking: Multiplying Two Images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Extacsy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
>> JWtimesMR = double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:)).*double(MR);
>> M = max(JWtimesMR(:));
>> m = min(JWtimesMR(:));
>> JWtimesMR = uint8(255*(double(JWtimesMR)-m)/(M-m));
>> figure
>> image(JWtimesMR)
>> truesize
>> title('EcstasyBumperillo')
```

Note how the image intensities are scaled back into the range 0-255.

## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## A note on image intensity scaling

In the code on the previous slide we scaled the result of the image multiplication so that it would have the maximum possible dynamic range (0-255). The formula for such linear scaling is

$$
m=\min (\mathbf{I}), \quad M=\max (\mathbf{I}), \text { and } J=255(\mathbf{I}-m) /(M-m) .
$$



In Matlab, if the images are of type uint8, this requires class conversions:
J = uint8(255*double(I-m)/double(M-m));

Without the double casts, the arithmetic is performed with 8 bits of precision, which yields incorrect results.

## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## More on image intensity scaling



Linear scaling is not always appropriate. If the image has some important features with very low intensity values and others that are large, the darker features may not be visible in a linearly scaled result. In such cases nonlinear scaling can perform better. The images below are differently scaled versions of $D=\operatorname{abs}(I-J)$; where $\mathbf{I}$ is the image to the left and $\mathbf{J}$ was a quality-0 jpeg copy of itself.

uint8(255*(D-m)/(M-m)))


## EECE 253 Image Processing

## Vanderbilt University School of Engineering

## Pixel Indexing in Matlab

"For" loops in Matlab are inefficient, whereas Matlab’s native indexing procedures are very fast.

Rather than
use, if possible
J = IP_Function(I);

But, sometimes that is not possible.
For example, if the output, $\mathbf{J}$, is decimated with respect to the input, $\mathbf{I}$,

> "IP_Function" is some arbitrary image processing function that you or someone else has written.

```
for r = 1:R
```

for r = 1:R
for c = 1:C
for c = 1:C
J(r,c,:) = IP_Function(I(r,c,:));
J(r,c,:) = IP_Function(I(r,c,:));
end
end
end

```
end
``` the above will not work (unless, of course, it is done within IP_function).

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixel Indexing in Matlab}


To decimate the above image by a factor of \(n\), create a vector, \(\mathbf{r}\), that contains the index of every \(n\)th row, and a similar vector, \(\mathbf{c}\).


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixel Indexing in Matlab}
 \(n=3\)


Then, vectors \(\mathbf{r}\) and \(\mathbf{c}\) used as index arguments for image I select every \(n\)th column in every \(n\)th row.


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Pixel Indexing in Matlab \\ Here, \(n=3\)}

\[
\text { image, } \mathbf{I}
\]
\[
r=1: n: R ; \quad c=1: n: C
\]


\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from \(I\) as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops:
 vectorize instead.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from \(\mathbf{I}\) as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops:
 vectorize instead.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from I as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops;
 vectorize instead.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from \(\mathbf{I}\) as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops;
 vectorize instead.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from \(I\) as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops:
 vectorize instead.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Indexing in Matlab}

Indexing in Matlab is fully general.
If \(\mathbf{I}\) is \(R \times C \times B\), vectors \(\mathbf{r}\) and \(\mathbf{c}\) can contain any numbers \(1 \leq r_{k} \leq R\) and \(1 \leq c_{k} \leq C\).

The numbers can be in any order and can be repeated within \(\mathbf{r}\) and \(\mathbf{c}\). The result of \(\mathbf{I}(\mathbf{r}, \mathbf{c})\) is an ordinal shuffling of the pixels from \(I\) as indexed by \(\mathbf{r}\) and \(\mathbf{c}\).

Whenever possible, avoid using 'for' loops:
 vectorize instead.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Figure 1 \\ - \(\mid\) - \(\mid x\) \\ Pixel Tndexing \\ File Edit View Insert Tools Desktop Window Help}
>> I = imread('Lawraa - Fl
>> size(I)
ans =
\(576 \quad 768 \quad 3\)
\(\gg=\) randperm(576);
\(\gg c=r a n d p e r m(768) ;\)
\(\gg J=I(r, c,:) ;\)
>> figure
>> image(J)
>> truesize
>> title('Scrambled Image
>> xlabel('What is it?')

Fun (if you're an imaging geek) thing to try with Matlab indexing: Scramble an image!


EECE 253 Image Processing
Vanderbilt University School of Engineering


The image can be unscrambled using the row and column permutation vectors, \(\mathbf{r}\) \& c.
```

>>
>> xlabel('What is it?')
>> K(r,c,:) = J;
>> figure
>> image(K)
>> truesize
>> title('Yay!!!')

```
>> xlabel('Photo: Lawraa on Flickr.com')

\title{
EECE/CS 253 Image Processing
}

\author{
Lecture Notes: Image Histograms
}

\section*{Richard Alan Peters II}

Department of Electrical Engineering and
Computer Science
Fall Semester 2014

EECE 253 Image Processing

\section*{Image Histograms}

The histogram of an image is a tally of the number of pixels at each intensity level or color. The shape of the histogram is related to the ranges and groupings of intensity values in the image.
In the following monochrome examples notice how the peaks of in the histogram correspond to concentrations of intensities in the image globally.
In the color examples the primary that has the largest value at any intensity dominates the image.

EECE 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Monochrome Intensity Distributions}


This image is a small, monochrome version of a huge color mosaic made by the ESO \({ }^{1}\). It contains both celestial hemispheres; it is what you would see in \(360^{\circ}\) from empty space in the plane of the galaxy above or below the earth.


\footnotetext{
http://www.eso.org/public/usa/images/eso0932a/
}

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Monochrome Intensity Distributions}


This picture, taken in the morning fog, displays low contrast - a narrow range of


\footnotetext{
http://hqwallbase.com/21961-trees-fog-wallpaper-[2]/
}

EECE 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Monochrome Intensity Distributions}


Castner glacier in the Delta mountains, Alaska. Monochrome extracted from original color image. Note how the peaks in the histogram correspond to regions in the image.


\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Color Intensity Distributions}


Castle Rock, Sedona, Arizona. There is one histogram for each of red, green, and blue. The red rock's color is in the midrange of intensities while the greenery is darker. Blue peaks correspond to the haze on the mountainside (dark) and the sky (bright).


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Intensity Distributions}


Unidentified place in a photo from the website below. Notice that the intensity of green dominates the others over much of the range. Red dominant corresponds to yellow-green regions. Blue dominates in the shadows.

http://forum.baboo.com.br/index.php?/gallery/image/20033-floresta-80/

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Color Intensity Distributions}


Blue Poison Dart Frog (Dendrobates azureus) in the Frankfurt Zoo, Germany. Dominant colors in increasing intensity: brown, blue, tan brown, blue.


Photo by Wikipedia user, Quartl: http://en.wikipedia.org/wiki/File:Dendrobates azureus qtl1.jpg/ .

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Intensity Distributions}


Photo taken in the gardens at Keukenhof, Holland, The Netherlands. RGB primaries dominant at different intensities: blue shadows, green tulip stems, blue hyacinths, red tulip flowers.


\footnotetext{
Photo by Jim Pyre: http://thedude.com/archives/2005/04/amsterdam.htm|
}

\section*{Image Histograms: Monochrome}

The histogram of an image is a tally of the number of pixels at each intensity level or color. For a monochrome image G,
\[
H_{\mathbf{G}}(g)=\#\{\mathbf{p} \mid \mathbf{G}=g\} .
\]

The value of the histogram at \(g\) is the number of pixels for which image \(\mathbf{G}\) has intensity level \(g\). For an 8-bit image, \(H\) has 256 values
\[
H_{\mathrm{G}}:\{0, \ldots, 255\} \rightarrow\{0, \ldots, R C\}
\]

If \(G\) is an \(R \times C\) image and all its pixels have the same intensity, \(g_{0}\), then \(H\left(g_{0}\right)=R C\) and \(H(g)=0\) for all intensities \(g \neq g_{0}\).

\section*{Image Histograms: Monochrome}

। If I is a 1-band (monochrome) image, then
। the pixel \(\mathbf{I}(r, c)\) is an 8-bit integer between 0 and 255.
। The histogram, \(h_{\mathbf{I}}\), of II is:
- a 256-element array, \(h_{\mathrm{I}}\), where
- \(h_{\mathrm{I}}(g)\) is an integer for \(g=1,2,3, \ldots, 256\), such that
- \(h_{\mathbf{I}}(g)=\) number of pixels in \(\mathbf{I}\) that have value \(g-1\).

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Histogram of a Monochrome Image}


\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Giant Mesquite Bug}
a.k.a. Banjo Beetle


24-bit truecolor image
Photo by Alan Peters,
Tucson, Arizona 1986.


During early summer in the Sonoran Desert of Southern Arizona, clusters of large, strange-looking, red and white bugs can be spotted on the foliage of mesquite trees (Prosopis spp.). These colorful bugs are the immature, wingless nymphs of the Giant Mesquite Bug or Leaf-footed Bug (Thasus neocalifornicus).

Photo and description by T. Beth Kinsey
http://fireflyforest.net, 2008

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Histogram of a Monochrome Image}


16-level (4-bit) image
lower RHC: number of pixels with intensity \(g\)

black marks pixels with intensity \(g\)

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Histogram of a Monochrome Image}


Plot of histogram: number of pixels with intensity \(g\)

Black marks pixels with intensity \(g\)


\author{
EECE 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{The Histogram of a Monochrome Image}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Image Histograms: Color}

The histogram of an image is a tally of the number of pixels at each intensity level or color. For a color image I, there are several possible histograms [...]
\[
H_{\mathbf{G}}(g)=\#\{\mathbf{p} \mid \mathbf{G}=g\}
\]

The value of the histogram at \(g\) is the number of pixels for which image \(\mathbf{G}\) has intensity level \(g\). For an 8-bit image, \(H\) has 256 values
\[
H_{\mathrm{G}}:\{0, \ldots, 255\} \rightarrow\{0, \ldots, R C\}
\]

If \(G\) is an \(R \times C\) image and all its pixels have the same intensity, \(g_{0}\), then \(H\left(g_{0}\right)=R C\) and \(H(g)=0\) for all intensities \(g \neq g_{0}\).

\section*{The Histograms of a Color Image}

1 If \(\mathbf{I}\) is a 3-band image (truecolor, 24-bit)
। then \(\mathbf{I}(r, c, b)\) is an integer between 0 and 255.
- Either I has 3 histograms:
- \(h_{\mathbf{R}}(g+1)=\) \# of pixels in \(\mathbf{I}(:,:, 1)\) with intensity value \(g\)
- \(h_{\mathbf{G}}(g+1)=\#\) of pixels in \(\mathbf{I}(:,:, 2)\) with intensity value \(g\)
- \(h_{\mathbf{B}}(g+1)=\#\) of pixels in \(\mathbf{I}(:,:, 3)\) with intensity value \(g\)

। or 1 vector-valued histogram, \(h(g, 1, b)\) where
- \(h(g+1,1,1)=\) \# of pixels in I with red intensity value \(g\)
- \(h(g+1,1,2)=\) \# of pixels in I with green intensity value \(g\)
- \(\quad h(g+1,1,3)=\#\) of pixels in I with blue intensity value \(g\)

EECE 253 Image Processing
Vanderbilt University School of Engineering

The Histograms
of a Color Image



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Value Image}

\section*{How to extract a monochrome intensity image from a color image.}

If I is an rgb image, then I's value image, \(\mathbf{V}\), has one band that is the pixel-wise average of \(\mathbf{I}\) 's \(\mathbf{R}, \mathbf{G}, \& \mathbf{B}\) bands:
\[
\mathbf{V}(r, c)=\frac{1}{3}[\mathbf{R}(r, c)+\mathbf{G}(r, c)+\mathbf{B}(r, c)] .
\]

This is easily computed in Matlab by

\section*{V=sum(I, 3)/3;}

The 3 in the \(2^{\text {nd }}\) argument of sum tells it to act along dimension 3 of the image - across the color bands.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Luminance Image}

\section*{How to extract a monochrome intensity image from a color image.}

I's luminance image, \(\mathbf{L}\), is a 1-band image that is a specific, weighted, pixelwise average of \(\mathbf{I}\) 's \(\mathbf{R}, \mathbf{G}\), and \(\mathbf{B}\) bands:
\[
\mathbf{L}(r, c)=0.299 \cdot \mathbf{R}(r, c)+0.587 \cdot \mathbf{G}(r, c)+0.114 \cdot \mathbf{B}(r, c)
\]

The numbers were derived by the NTSC \(^{1}\) to weight each color band according to the relative intensity resolution that color by the human eye. The following Matlab code will compute it
\[
\begin{aligned}
\mathrm{L}= & \text { uint8(sum(bsxfun(@times, double(I),... } \\
& \text { reshape([0.299 0.587 0.114],[1 1 3])), 3)); }
\end{aligned}
\]
but it is not obvious how it does. That is explained on the next slide.

\footnotetext{
\({ }^{1}\) National Television System Committee, 1953, http://en.wikipedia.org/wiki/NTSC
}

\section*{Computing the Luminance Image in Matlab}

The first steps are to create a \(1 \times 1 \times 3\) matrix (vector) containing the weights:
\[
\text { w = reshape([0.299 0.587 0.114],[ } \left.\left.1 \begin{array}{lll}
1 & 1 & 3
\end{array}\right]\right) ;
\]
and to convert the uint8, 3-band image, I, to class double,
J = double(I);
bsxfun combines image J and vector w using @times, a multiplication operator. Effectively, it makes an image, W , the same size as J with a copy of w in every pixel location. Then it multiplies the 2 images together pointwise.
T = bsxfun(@times,J,w);

T has bands \(0.299 \mathbf{R}, 0.587 \mathbf{G}\), and \(0.144 \mathbf{B}\), which are then summed and converted (rounded) to class uint8.
L = uint8(sum(T));

All the steps are combined into 1 line of code on the previous page.

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Value Histogram}


Value image, \(\mathbf{V}\).


Histogram of the value image.

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Luminance Histogram}


Luminance image, \(\mathbf{L}\).


Histogram of the luminance image.

The value histogram is not the EECE 253 Image Processing average of the three 1-D color intensity histograms.

\section*{Value Histogram vs. Average of R,G,\&B Histograms}



\section*{Multi-Band Histogram Calculator in Matlab}
```

% Multi-band histogram calculator
function h=histogram(I)
[R C B]=size(I);
% allocate the histogram
h=zeros(256, 1, B);
% range through the intensity values
for g=0:255
h(g+1,1,:) = sum(sum(I==g)); % accumulate
end
return;

```

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Multi-Band Histogram Calculator in Matlab}
```

% Multi-band histogram calc
function h=histogram(I)
[R C B]=size(I);
% allocate the histogram
h=zeros(256,1,B);
% range through the intensi
Loop through all intensity levels (0-255)
Tag the elements that have value g.
The result is an RxC\timesB logical array that
has a 1 wherever I (r,c,b) =g and 0's
everywhere else.
Compute the number of ones in each band of
the image for intensity g.
Store that value in the 256\times1\timesB histogram
at h(g+1,1,b).
for g=0:255
h(g+1,1,:) = sum(sum(I==g)); % accumulate
end
If B==3, then h(g+1, 1,:) contains
3 numbers: the number of pixels in
sum(sum(I==g)) computes one
number for each band in the image.

```

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Vectorized Multi-Band Histogram Calculator}
```

% Vectorized multi-band histogram calculator using
% Matlab's built-in histogram calculator, histc().
% Result, h, is 256\times1 for a one-band image and 256\times1\times3
% for a three-band image.
function h = histogram(I)
h = sum(histc(I,0:255),2);
end;

```

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Density Function of an Image}

Let \(A=\sum_{g=0}^{255} h_{\mathbf{I}_{k}}(g+1)\).
pdf
[lower case]
Note that since \(h_{\mathbf{I}_{k}}(g+1)\) is the number of pixels in \(\mathbf{I}_{k}\) (the \(k\) th color band of image \(\mathbf{I}\) ) with value \(g\), \(A\) is the number of pixels in \(\mathbf{I}\). That is if \(\mathbf{I}\) is \(R\) rows by \(C\) columns then \(A=R \times C\).

Then,
\[
p_{\mathbf{I}_{k}}(g+1)=\frac{1}{A} h_{\mathbf{I}_{k}}(g+1)
\]

This is the probability that an arbitrary pixel from \(\mathrm{I}_{k}\) has value \(g\).
is the graylevel probability density function of \(\mathbf{I}_{k}\).

\section*{EECE 253 Image Processing}

\section*{The Probability Density Function of an Image}
- \(p_{\text {band }}(g+1)\) is the fraction of pixels in (a specific band of) an image that have intensity value \(g\).
- \(p_{\text {band }}(g+1)\) is the probability that a pixel randomly selected from the given band has intensity value \(g\).
- Whereas the sum of the histogram \(h_{\text {band }}(g+1)\) over all \(g\) from 1 to 256 is equal to the number of pixels in the image, the sum of \(p_{\text {band }}(g+1)\) over all \(g\) is 1 .
- \(p_{\text {band }}\) is the normalized histogram of the band.

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Distribution Function of an Image}

Let \(\mathbf{q}=\left[\mathrm{q}_{1} \mathrm{q}_{2} \mathrm{q}_{3}\right]=\mathbf{I}(r, c)\) be the value of a randomly selected pixel from I. Let \(g\) be a is given by
\(P_{\mathbf{I}_{k}}(g+1)=\sum_{\gamma=0}^{g} p_{\mathbf{I}_{k}}(\gamma+1)=\frac{1}{A} \sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)=\frac{\sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{\mathbf{I}_{k}}(\gamma+1)}\),
where \(h_{\text {Ik }}(\gamma+1)\) is the histogram of the \(k\) th band of \(\mathbf{I}\).

This is the probability that any given pixel from \(\mathbf{I}_{k}\) has value less than or equal to \(g\).

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Distribution Function of an Image}

Let \(\mathbf{q}=\left[\mathrm{q}_{1} \mathrm{q}_{2} \mathrm{q}_{3}\right]=\mathbf{I}(r, c)\) be the value of a randomly selected pixel from I. Let \(g\) be a specific graylevel. The probability that \(\mathrm{q}_{\mathrm{k}} \leq \mathrm{g}\) is given by

Also called CDF for "Cumulative Distribution Function".
\(P_{I_{k}}(g+1)=\sum_{\gamma=0}^{g} p_{\mathbf{I}_{k}}(\gamma+1)=\frac{1}{A} \sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)=\frac{\sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{\mathbf{I}_{k}}(\gamma+1)}\),
where \(h_{\mathrm{Ik}}(\gamma+1)\) is the histogram of the \(k\) th band of \(\mathbf{I}\).

This is the probability that any given pixel from \(\mathbf{I}_{k}\) has value less than or equal to \(g\).

\section*{The Cumulative Distribution Function of an Image}
- \(P_{\text {band }}(g+1)\) is the fraction of pixels in (a specific band of) an image that have intensity values less than or equal to \(g\).
- \(P_{\text {band }}(g+1)\) is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to \(g\).
- \(P_{\text {band }}(g+1)\) is the cumulative (or running) sum of \(p_{\text {band }}(g+1)\) from 0 through \(g\) inclusive.
- \(P_{\text {band }}(1)=\mathrm{p}_{\text {band }}(1)\) and \(P_{\text {band }}(256)=1 ; P_{\text {band }}(g+1)\) is nondecreasing.

Note: the Probability Distribution Function (PDF, capital letters) and the Cumulative Distribution Function (CDF) are exactly the same things. Both PDF and CDF will refer to it. However, pdf (small letters) is the density function.

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{The pdf vs. the CDF}


\[
0.8 \text { ( } \operatorname{CDFF}(x)=\int_{-\infty}^{x} \operatorname{pdf}(\xi) d \xi
\]

\title{
EECE/CS 253 Image Processing
}

\author{
Lecture Notes: The Point Processing of Images
}

\section*{Richard Alan Peters II}

Department of Electrical Engineering and
Computer Science
Fall Semester 2014

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processing of Images}
- In a digital image, a point = a pixel. । Point processing transforms a pixel's value as function of its value alone; । it does not depend on the values of the pixel's neighbors.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processing of Images}

\section*{Examples include: \\ , Brightness and contrast adjustment , Gamma correction \\ , Histogram equalization \\ , Histogram matching \\ , Color correction.} Point Processing

- gamma

histogram mod

- brightness

- contrast

original

original

+ brightness

+ contrast

+ gamma

histogram EQ

\section*{Point Processing: Pixel Values}

A point process transforms one intensity level (or color) into another as a function of that one alone. So a point process is
\[
\mathbf{p}_{\text {out }}=f\left(\mathbf{p}_{\text {in }}\right) .
\]

That is, the pixel value output is dependent on only the pixel value input. That implies
\[
\mathbf{p}_{\text {out }}(r, c)=f\left(\mathbf{p}_{\text {in }}(r, c)\right) .
\]

In words, the output at one location is dependent only the value of the input image at that same location. Other locations don't matter.

\section*{Point Ops via Functional Mappings}

Image:

\[
\mathbf{J}=\Phi[\mathbf{I}]
\]

Input

Pixel:


The transformation of image \(\mathbf{I}\) into image \(\mathbf{J}\) is accomplished by replacing each input intensity, \(g\), with a specific output intensity, \(k\), at every location ( \(r, c, b\) ) where \(\mathbf{I}(r, c, b)=g\).

The rule that associates \(k\) with \(g\) is usually specified with a function, \(f\), so that \(f(g)=k\).

\section*{Point Ops via Functional Mappings}

\section*{One-band Image}
\(\mathbf{J}(r, c)=f(\mathbf{I}(r, c))\),
for all pixel locations, \((r, c)\).

\section*{Three-band Image}
\[
\begin{aligned}
& \mathbf{J}(r, c, b)=f(\mathbf{I}(r, c, b)), \text { or } \\
& \mathbf{J}(r, c, b)=f_{b}(\mathbf{I}(r, c, b)), \text { or } \\
& \text { for } b=1,2,3 \text {, and all }(r, c) .
\end{aligned}
\]

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Ops via Functional Mappings}

\section*{One-band Image}

Either all 3 bands are mapped through the same function, f, or
\(\mathbf{J}(r, c)=f(\mathbf{I}(r, c))\),
for all pixel locations, \((r, c)\).

Three-band Image
\[
\begin{aligned}
& \mathbf{J}(r, c, b)=f(\mathbf{I}(r, c, b)) \text {, or } \\
& \mathbf{J}(r, c, b)=f_{b}(\mathbf{I}(r, c, b)) \text {, or } \\
& \text { for } b=1,2,3 \text {, and all } \\
& \begin{array}{l}
\text {... each band is } \\
\text { maped through } \\
\text { a separate func- } \\
\text { tion, } f_{b} .
\end{array}
\end{aligned}
\]

\section*{EECE 253 Image Processing}

\section*{Lookup Tables}

A lookup table is an indexed list of numbers - a vector - that can be used to implement a discrete function - a one-to-one mapping from one set of numbers, \(\left\{g_{\mathrm{in}, 1}, g_{\mathrm{in}, 2}, \ldots, g_{\mathrm{in}, n}\right\}\) to another, \(\left\{g_{\text {out }, 1}, g_{\text {out }, 2}, \ldots, g_{\text {out }, n}\right\}\). A lookup table can implement a function such as:
\[
\begin{aligned}
& \text { if } g_{\text {out }}=f\left(g_{\text {in }}\right) \text {, where } g_{\text {in }} \in\{0, \ldots, n-1\} \text { and } g_{\text {out }} \in\left\{g_{\text {out }, k}\right\}_{k=1}^{n} \\
& \text { then define } \operatorname{LUT}\left(g_{\text {in }, k}+1\right)=\operatorname{LUT}(k)=g_{\text {out }, k} \text {, for } k=1, \ldots n \text {. }
\end{aligned}
\]

We've already seen one of these, the colormap. \({ }^{1}\)

\footnotetext{
\({ }^{1}\) Lecture 2, slides 29-40.
}

\section*{Point Operations using Lookup Tables}

A lookup table (LUT) can implement a functional mapping.

If \(k=f(g)\),
for \(g=0, \ldots, 255\),
and if \(k\) takes on
values in \(\{0, \ldots, 255\}, \ldots\)

... then the LUT that implements \(f\) is a \(256 \times 1\) array whose \((g+1)^{\text {th }}\) value is \(k=f(g)\).

LUT is \(256 \times 1\). But I may be \(R \times C\) or \(R \times C \times 3\).

To remap an image, I, to J :

\section*{Point Operations \(=\) Lookup Table Ops}

\begin{tabular}{|c|c|c|}
\hline E.g.: & index & value \\
\cline { 2 - 3 } & \(\ldots\) & \(\ldots\) \\
\cline { 2 - 3 } & 101 & 64 \\
\cline { 2 - 3 } & 102 & 68 \\
\cline { 2 - 3 } & 103 & 69 \\
\cline { 2 - 3 } & 104 & 70 \\
\hline & 105 & 70 \\
\cline { 2 - 3 } & 106 & 71 \\
\hline \multicolumn{3}{c|}{ … } \\
\multicolumn{3}{c|}{ input } \\
output
\end{tabular}

\section*{How to Generate a Lookup Table}

For example, a sigmoid:
\[
\begin{aligned}
& \text { Let } a=2 . \\
& \text { Let } x \in\{0, \ldots, 255\} \\
& \sigma(x ; a)=\frac{255}{1+e^{-a(x-127) / 32}}
\end{aligned}
\]

Or in Matlab:
a = 2;
\(x=0: 255\);
LUT \(=255 . /\left(1+\exp \left(-a^{*}(x-127) / 32\right)\right)\);

\section*{This is just one example.}

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Ops on RGB Images using Lookup Tables}

If I is 3 -band, then
a) each band is mapped separately using the same LUT for each band or
b) each band is mapped using different LUTs one for each band.
a) \(\mathbf{J}=\operatorname{LUT}(\mathbf{I}+1)\),
b) \(\mathbf{J}(:,:, b)=\operatorname{LUT}_{b}(\mathbf{I}(:,:, b)+1)\), for \(b=1,2,3\).

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{1-Band Lookup Table for 3-Band Image}


\section*{Example 3-Band Image ...}


Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{... with 3-Band LUT.}


Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{3-Band Image with 3-Band LUT.}




Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s s/8624344496/in/pool-tbasab/ Point Processing

- gamma

histogram mod

- brightness

- contrast

original

original

+ brightness

+ contrast

+ gamma

histogram EQ

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Point Processes：Original Image}


Kinkaku－ji（金閣寺，Temple of the Golden Pavilion），also known as Rokuon－ji（鹿苑寺，Deer Garden
Temple），is a Zen Buddhist temple in Kyoto，Japan．

Photo by Alan Peters，August 1993.


\section*{Luminance Histogram}

For more information on this unique place read the historical novel by Mishima，Yukio，The Temple of the Golden Pavilion，translated by Ivan Morris， Shinchosha Publishing Co，Ltd．， 1956.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Increase Brightness}

\[
\mathbf{J}(r, c, b)= \begin{cases}\mathbf{I}(r, c, b)+g, & \text { if } \mathbf{I}(r, c, b)+g<256 \\ 255, & \text { if } \mathbf{I}(r, c, b)+g>255\end{cases}
\]
\[
g \geq 0 \text { and } b \in\{1,2,3\} \text { is the band index. }
\]


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Decrease Brightness}

\[
\mathbf{J}(r, c, b)= \begin{cases}0, & \text { if } \mathbf{I}(r, c, b)-g<0 \\ \mathbf{I}(r, c, b)-g, & \text { if } \mathbf{I}(r, c, b)-g>0\end{cases}
\]
\(g \geq 0\) and \(b \in\{1,2,3\}\) is the band index.



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Decrease Contrast}


Here, \(s=127\)
\[
\begin{array}{rlrl}
\mathbf{T}(r, c, b) & =a[\mathbf{I}(r, c, b)-s]+s, & & s \text { is the } \\
\text { where } 0 & \leq a<1.0, & & \text { center of } \\
& \mathrm{s} \in\{0,1,2, \ldots, 255\}, \text { and } & \text { the contrast } \\
& b \in\{1,2,3\} . & &
\end{array}
\]



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Increase Contrast}

\[
\begin{aligned}
& \mathbf{T}(r, c, b)=a[\mathbf{I}(r, c, b)-s]+s \\
& \mathbf{J}(r, c, b)= \begin{cases}0, & \text { if } \mathbf{T}(r, c, b)<0, \\
\mathbf{T}(r, c, b), & \text { if } 0 \leq \mathbf{T}(r, c, b) \leq 255, \\
255, & \text { if } \mathbf{T}(r, c, b)>255 .\end{cases} \\
& a>1, \quad \mathrm{~s} \in\{0, \ldots, 255\}, \quad b \in\{1,2,3\}
\end{aligned}
\]


Here, \(s=127\)

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Contrast Stretch}


Let \(m_{\mathbf{I}}=\min [\mathbf{I}(r, c)], \quad M_{\mathbf{I}}=\max [\mathbf{I}(r, c)]\), \(m_{\mathbf{J}}=\min [\mathbf{J}(r, c)], \quad M_{\mathbf{J}}=\max [\mathbf{J}(r, c)]\).
Then,
\[
\mathbf{J}(r, c)=\left(M_{\mathbf{J}}-m_{\mathbf{J}}\right) \frac{\mathbf{I}(r, c)-m_{\mathbf{I}}}{M_{\mathbf{I}}-m_{\mathbf{I}}}+m_{\mathbf{J}} .
\]
histograms

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Information Loss from Contrast Adjustment}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Information Loss from Contrast Adjustment}

abbreviations: original low-contrast high-contrast restored difference
 original and restored low-contrast

difference between original and restored high-contrast

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Increased Gamma}

\[
\mathbf{J}(r, c)=255 \cdot\left[\frac{\mathbf{I}(r, c)}{255}\right]^{\frac{1}{\gamma}} \text { for } \gamma>1.0
\]


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Point Processes: Decreased Gamma}

\[
\mathbf{J}(r, c)=255 \cdot\left[\frac{\mathbf{I}(r, c)}{255}\right]^{\frac{1}{\gamma}} \text { for } \gamma<1.0
\]



\section*{Gamma Correction: Effect on Histogram}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Density Function of an Image}

Let \(A=\sum_{g=0}^{255} h_{\mathbf{I}_{k}}(g+1)\).
pdf
[lower case]
Note that since \(h_{\mathbf{I}_{k}}(g+1)\) is the number of pixels in \(\mathbf{I}_{k}\) (the \(k\) th color band of image \(\mathbf{I}\) ) with value \(g\), \(A\) is the number of pixels in \(\mathbf{I}\). That is if \(\mathbf{I}\) is \(R\) rows by \(C\) columns then \(A=R \times C\).

Then,
\[
p_{\mathbf{I}_{k}}(g+1)=\frac{1}{A} h_{\mathbf{I}_{k}}(g+1)
\]

This is the probability that an arbitrary pixel from \(I_{k}\) has value \(g\).
is the graylevel probability density function of \(\mathbf{I}_{k}\).

\section*{EECE 253 Image Processing}

\section*{The Probability Density Function of an Image}
- \(p_{\text {band }}(g+1)\) is the fraction of pixels in (a specific band of) an image that have intensity value \(g\).
- \(p_{\text {band }}(g+1)\) is the probability that a pixel randomly selected from the given band has intensity value \(g\).
- Whereas the sum of the histogram \(h_{\text {band }}(g+1)\) over all \(g\) from 1 to 256 is equal to the number of pixels in the image, the sum of \(p_{\text {band }}(g+1)\) over all \(g\) is 1 .
- \(p_{\text {band }}\) is the normalized histogram of the band.

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Distribution Function of an Image}

Let \(\mathbf{q}=\left[\mathrm{q}_{1} \mathrm{q}_{2} \mathrm{q}_{3}\right]=\mathbf{I}(r, c)\) be the value of a randomly selected pixel from I. Let \(g\) be a specific graylevel. The probability that \(\mathrm{q}_{\mathrm{k}} \leq \mathrm{g}\)

PDF [upper case] is given by
\(P_{\mathbf{I}_{k}}(g+1)=\sum_{\gamma=0}^{g} p_{\mathbf{I}_{k}}(\gamma+1)=\frac{1}{A} \sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)=\frac{\sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{\mathbf{I}_{k}}(\gamma+1)}\),
where \(h_{\mathrm{Ik}}(\gamma+1)\) is the histogram of the \(k\) th band of \(\mathbf{I}\).

This is the probability that any given pixel from \(\mathbf{I}_{k}\) has value less than or equal to \(g\).

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Probability Distribution Function of an Image}

Let \(\mathbf{q}=\left[\mathrm{q}_{1} \mathrm{q}_{2} \mathrm{q}_{3}\right]=\mathbf{I}(r, c)\) be the value of a randomly selected pixel from I. Let \(g\) be a specific graylevel. The probability that \(\mathrm{q}_{\mathrm{k}} \leq \mathrm{g}\) is given by

Also called CDF for "Cumulative Distribution Function".
\(P_{\mathbf{I}_{k}}(g+1)=\sum_{\gamma=0}^{g} p_{\mathbf{I}_{k}}(\gamma+1)=\frac{1}{A} \sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)=\frac{\sum_{\gamma=0}^{g} h_{\mathbf{I}_{k}}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{\mathbf{I}_{k}}(\gamma+1)}\),
where \(h_{\mathrm{Ik}}(\gamma+1)\) is the histogram of the \(k\) th band of \(\mathbf{I}\).

This is the probability that any given pixel from \(\mathbf{I}_{k}\) has value less than or equal to \(g\).

\section*{The Cumulative Distribution Function of an Image}
- \(P_{\text {band }}(g+1)\) is the fraction of pixels in (a specific band of) an image that have intensity values less than or equal to \(g\).
- \(P_{\text {band }}(g+1)\) is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to \(g\).
- \(P_{\text {band }}(g+1)\) is the cumulative (or running) sum of \(p_{\text {band }}(g+1)\) from 0 through \(g\) inclusive.
- \(P_{\text {band }}(1)=\mathrm{p}_{\text {band }}(1)\) and \(P_{\text {band }}(256)=1 ; P_{\text {band }}(g+1)\) is nondecreasing.

Note: the Probability Distribution Function (PDF, capital letters) and the Cumulative Distribution Function (CDF) are exactly the same things. Both PDF and CDF will refer to it. However, pdf (small letters) is the density function.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{The pdf vs. the CDF}



EECE 253 Image Processing

\section*{Point Processes: Histogram Equalization}

Task: remap a 1-band image I so that its histogram is as close to constant as possible. This maximizes the contrast evenly across the entire intensity range.

Let \(P_{\mathbf{I}}(\gamma+1)\) be the cumulative (probability) distribution function of \(\mathbf{I}\). Then \(\mathbf{J}\) has, as closely as possible, a flat (constant) histogram if:
\[
\mathbf{J}(r, c, b)=255 \cdot P_{\mathbf{I}}[\mathbf{I}(r, c, b)+1] .
\]

The scaled CDF itself is used as the LUT.
one-band image

That is, to equalize a one-band image, map it through its own CDF multiplied by the maximum desired output value.

\section*{EECE 253 Image Processing}

\section*{Point Processes: Histogram Equalization}

Task: remap image \(\mathbf{I}\) with \(\min =m_{\mathbf{I}}\) and \(\max =M_{\mathbf{I}}\) so that its histogram is as close to constant as possible and has \(\min =m_{\mathbf{J}}\) and \(\max =M_{\mathbf{J}}\).

Let \(P_{\mathbf{I}}(\gamma+1)\) be the cumulative (probability) distribution function of \(\mathbf{I}\).

Then \(\mathbf{J}\) has, as closely as possible, the correct histogram if

\section*{Using intensity extrema}
\[
\mathbf{J}(r, c)=\left(M_{\mathbf{J}}-m_{\mathbf{J}}\right) \frac{P_{\mathbf{I}}[\mathbf{I}(r, c)+1]-P_{\mathbf{I}}\left(m_{\mathbf{I}}+1\right)}{P_{\mathbf{I}}\left(M_{\mathbf{I}}+1\right)-P_{\mathbf{I}}\left(m_{\mathbf{I}}+1\right)}+m_{\mathbf{J}} .
\]

EECE 253 Image Processing Vand The CDF (cumulative distribution) \(\times 255\) is
Histogram EQ the LUT for remapping.


Value Image


Histogram EQ'd Value Image


EECE 253 Image Processing
Vane The CDF (cumulative distribution) \(\times 255\) is
Histogram EQ the LUT for remapping.


Value Image


Histogram EQ'd Value Image


EECE 253 Image Processing Vand The CDF (cumulative distribution) \(\times 255\) is the LUT for remapping.

\section*{Histogram EQ}


Value Image


Histogram EQ'd Value Image


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ}


Value Image


Histogram EQ'd Value Image

(c) 1999-2014 by Richard Alan Peters II

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Histogram EQ of a Grayscale Image}


Luminance Image


Luminance Histogram

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of a Grayscale Image}


Luminance Image


\section*{Histogram EQ of a Grayscale Image}


Luminance Image


\section*{Histogram EQ of a Grayscale Image}


Equalized Luminance Image


Histogram of Eq'd Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of a Grayscale Image}


Luminance Image


\section*{Equalized Luminance Image}

Note the detail loss in saturated areas.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of the Individual Bands}


Original Color Image


Color Histograms

One histogram for each band.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of the Individual Bands}


Original Color Image


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of the Individual Bands}


Original Color Image


Equalization LUTs

Each band is mapped through its own LUT.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of the Individual Bands}


Equalized Color Image


Histogram of Eq'd Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram EQ of the Individual Bands}


Original Color Image


Equalized Color Image
Note the unnatural color shifts.

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Luminance EQ of a Color Image}


Original Color Image


Color Histograms

One histogram for each band.

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Luminance EQ of a Color Image}


Original Color Image


EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Luminance EQ of a Color Image}


Original Color Image


The scaled luminance CDF is used as the LUT for all 3 bands.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Luminance EQ of a Color Image}


Luminance Eq'd Color Image


Histo of Lum Eq'd Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Luminance EQ of a Color Image}


Original Color Image


Luminance Eq'd Color Image
Note the detail loss in saturated areas.

\section*{Point Processes: Histogram Matching}

Task: remap image I so that it has, as closely as possible, the same histogram as image \(\mathbf{J}\).

Q: Why do this?
A: Restore a degraded image based on an original. Match the characteristics of images of the same scene from different cameras.

EECE 253 Image Processing

\section*{Point Processes: Histogram Matching}

Task: remap image I so that it has, as closely as possible, the same histogram as image \(\mathbf{J}\).

Because the images are digital it is not, in general, possible to make \(h_{\mathbf{I}} \equiv h_{\mathbf{J}}\). Therefore, \(p_{\mathbf{I}} \equiv p_{\mathbf{J}}\).

Q:How, then, can the matching be done?
A: By matching percentiles.

\section*{EECE 253 Image Processing}

\section*{Matching Percentiles}
... assuming a 1-band image, one band of a color image or its luminance image.

Recall:
- The CDF of image \(\mathbf{I}\) is such that \(0 \leq P_{\mathrm{I}}\left(g_{\mathrm{I}}\right) \leq 1\).
- \(P_{\mathbf{I}}\left(g_{\mathrm{I}}+1\right)=c\) means that \(c\) is the fraction of pixels in \(\mathbf{I}\) that have a value less than or equal to \(g_{\mathrm{I}}\).
- \(100 c\) is the percentile of pixels in \(\mathbf{I}\) that are less than or equal to \(g_{\mathrm{I}}\).

To match percentiles, replace all occurrences of value \(g_{\mathrm{I}}\) in image \(\mathbf{I}\) with the value, \(g_{\mathbf{J}}\), from image \(\mathbf{J}\) whose percentile in \(\mathbf{J}\) most closely matches the percentile of \(g_{\mathbf{I}}\) in image \(\mathbf{I}\).

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Matching Percentiles} one band of a color image or its luminance image.

So, to create an image, \(\mathbf{K}\), from image \(\mathbf{I}\) such that \(\mathbf{K}\) has nearly the same CDF as image \(\mathbf{J}\) do the following:

Example:
\(\mathbf{I}(r, c)=5\)
\(P_{\mathrm{I}}(5)=0.65\)
\(P_{\mathbf{J}}(9)=0.56\)
\(P_{\mathbf{J}}(10)=0.67\)
\(\mathbf{K}(r, c)=10\)



\section*{Histogram Matching Algorithm} one band of a color image or its luminance image.
```

$[R, C]=\operatorname{size}(\mathbf{I}) ;$
$\mathbf{K}=\operatorname{zeros}(R, C)$;
$g_{\mathrm{J}}=m_{\mathbf{J}}$;
for $g_{\mathrm{I}}=m_{\mathrm{I}}$ to $M_{\mathrm{I}}$
while $g_{\mathrm{J}}<255$ AND $P_{\mathrm{I}}\left(g_{\mathrm{I}}+1\right)<1$ AND
$P_{\mathrm{J}}\left(g_{\mathrm{J}}+1\right)<P_{\mathrm{I}}\left(g_{\mathrm{I}}+1\right)$
$g_{\mathrm{J}}=g_{\mathrm{J}}+1$;
end
$\mathbf{K}=\mathbf{K}+\left[g_{\mathbf{J}} \cdot\left(\mathbf{I}==g_{\mathbf{I}}\right)\right]$
end

```

This directly matches image I to image J.
\[
\begin{gathered}
P_{\mathbf{I}}\left(g_{\mathbf{I}}+1\right): \operatorname{CDF} \text { of } \mathbf{I} \\
P_{\mathbf{J}}\left(g_{\mathbf{J}}+1\right): \operatorname{CDF} \text { of } \mathbf{J} . \\
m_{\mathbf{J}}=\min \mathbf{J}, \\
M_{\mathbf{J}}=\max \mathbf{J}, \\
m_{\mathbf{I}}=\min \mathbf{I}, \\
M_{\mathbf{I}}=\max \mathbf{I} .
\end{gathered}
\]

Better to use a LUT. See slide 66 .

EECE 253 Image Processing
Vanderbilt University School of Engineering

Example: Histogram Matching
Image pdf



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Histogram Matching}

\section*{Image CDF}


*a.k.a Cumulative Distribution Function, \(\mathrm{CDF}_{\mathrm{I}}\).

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Histogram Matching}

Target pdf
Target with 16 intensity values



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Histogram Matching}

\section*{Target CDF}


*a.k.a Cumulative Distribution Function, \(\mathrm{CDF}_{\mathrm{J}}\).

\section*{Histogram Matching with a Lookup Table}

The algorithm on slide 61 matches one image to another directly. Often it is faster or more versatile to use a lookup table (LUT). Rather than remapping each pixel in the image separately, one can create a table that indicates to which target value each input value should be mapped. Then
\[
\mathbf{K}=\operatorname{LUT}[\mathbf{I}+1]
\]

In Matlab if the LUT is a \(256 \times 1\) matrix with values from 0 to 255 and if image \(\mathbf{I}\) is one-band of type uint8, it can be remapped with the following code:
K = uint8(LUT(I+1));

\section*{Histogram Matching with a Lookup Table}

The E-Z teenage New York version* on the previous page only works for one-band images. For truecolor or other multiband images you need to execute the LUT on each band separately. Viz:
```

if E == 1 % single band LUT
for d = 1:D
J(:,:,d) = LUT(1+double(I(:,:,d)));
end
else % multiband LUT
for d = 1:D
LUT1D = squeeze(LUT(:,1,d));
J(:,:,d) = LUT1D(1+double(I(:,:,d)));
end
end

```

\footnotetext{
*http://www.science.uva.nl/~robbert/zappa/albums/Zappa_In_New_York/10.html \& http://youtu.be/GDwRJK8bpb4
}

\section*{LUT Creation}

EECE 253 Image Processing
Vanderbilt University School of Engineering



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Look Up Table for Histogram Matching}

LUT = zeros(256,1);
\(g_{\mathrm{J}}=0\);
for \(g_{\mathrm{I}}=0\) to 255
while \(P_{\mathrm{J}}\left(g_{\mathrm{J}}+1\right)<P_{\mathrm{I}}\left(g_{\mathrm{I}}+1\right)\) AND \(g_{\mathrm{J}}<255\)
\(g_{J}=g_{\mathrm{J}}+1 ;\)
end
\(\operatorname{LUT}\left(g_{\mathrm{I}}+1\right)=g_{\mathrm{J}} ;\)
end

This creates a look-up table which can then be used to remap the image.
\(P_{\mathbf{I}}\left(g_{\mathrm{I}}+1\right): \mathrm{CDF}\) of \(\mathbf{I}\),
\(P_{\mathbf{J}}\left(g_{\mathbf{J}}+1\right):\) CDF of \(\mathbf{J}\),
\(\operatorname{LUT}\left(g_{\mathrm{I}}+1\right)\) : Look- Up Table

EECE 253 Image Processing
Vanderbilt University School of Engineering

Input \& Target CDFs, LUT and Resultant CDF





EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Histogram Matching}

original

target

remapped

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Probability Density Functions of a Color Image}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Cumulative Distribution Functions (CDF)}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Probability Density Functions of a Color Image}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Cumulative Distribution Functions (CDF)}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Remap an Image to have the Lum. CDF of Another}

original

target

luminosity remapped

\author{
Vanderbilt University School of Engineering
}

\section*{CDFs and the LUT}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Luminance Remapping on pdfs}






Before
After

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Luminance Remapping on CDFs}


\section*{Histogram Equalization Revisited}

Our previous attempts to equalize the color version of Kinkaku-ji led to the unsatisfactory results below:

(a) Direct equalization of the color bands by mapping each band though its own CDF. \(\mathrm{LUT}_{\mathrm{b}}=255^{*} \mathrm{CDF}_{\mathrm{b}}\).

(b) Equalization of the color bands by mapping each band though the CDF of the luminance image. \(\mathrm{LUT}_{\mathrm{b}}=255^{*} \mathrm{CDF}_{\mathrm{L}}\).

\section*{Histogram Equalization Revisited}

Histogram matching presents another alternative: Match each color band CDF with the CDF from the luminance image. Then we get:

(a) Original image.

(b) Equalization by mapping band, \(b\), though \(\mathrm{LUT}_{b}=255^{*}\left[\mathrm{CDF}_{L}\right]^{-1} .{ }^{*} \mathrm{CDF}_{b}\).

\section*{EECE 253 Image Processing}

\section*{EQ of a Color Image via Luminance Matching}

In 6 (count 'em) 6 E-Z steps!
1. Convert image \(\mathbf{I}\) into grayscale image, \(\mathbf{L}\), via your favorite weighting scheme.
2. Compute the 3 color histograms, \(h_{\mathrm{C}}, \mathrm{C} \in\{\mathrm{R}, \mathrm{G}, \mathrm{B}\}\) of \(\mathbf{I}\) and the histogram, \(h_{\mathrm{L}}\), of \(\mathbf{L}\).
3. Compute the 4 probability density functions, \(p_{\mathrm{C}}\), \(\mathrm{C} \in\{\mathrm{R}, \mathrm{G}, \mathrm{B}, \mathrm{L}\}\), from the \(h_{\mathrm{C}}\).
4. Compute the 4 cumulative distribution functions, \(H_{C}\), \(\mathrm{C} \in\{\mathrm{R}, \mathrm{G}, \mathrm{B}, \mathrm{L}\}\), from the \(p_{\mathrm{C}}\).
5. Generate 3 lookup tables, \(T_{\mathrm{C}}, \mathrm{C} \in\{\mathrm{R}, \mathrm{G}, \mathrm{B}\}\), by matching \(H_{\mathrm{R}}\) to \(H_{\mathrm{L}}, H_{\mathrm{G}}\) to \(H_{\mathrm{L}}\), \& \(H_{\mathrm{B}}\) to \(H_{\mathrm{L}}\).
6. Map each image band \(\{\mathbf{R}, \mathbf{G}, \mathbf{B}\}\) through its corresponding lookup table \(T_{\mathrm{C}}\).

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{EQ of a Color Image via Luminance Matching}


Original Color Image


Color \& Lum. Histograms

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{EQ of a Color Image via Luminance Matching}


Original Color Image


Color \& Lum. CDFs

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{EQ of a Color Image via Luminance Matching}


Original Color Image


Each color band CDF is matched to the luminance CDF to generate the LUT..

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{EQ of a Color Image via Luminance Matching}


Luminance matched image


Histo of lum matched image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{EQ of a Color Image via Luminance Matching}


Original Color Image


\section*{Luminance matched image}

Note the slight desaturation of the colors.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Remap an Image to have the rgb CDF of Another}

original

target


R, G, \& B remapped

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{CDFs and the LUTs}


\section*{Effects of RGB Remapping on pdfs}






After

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of RGB Remapping on CDFs}


\section*{Histogram Matching for Image Restoration}

Degraded image


Another image of the same scene，not degraded


Lotus Flowers at Turtle Head Park，Lake Tai，Wuxi，Jiangsu Province，China．莲花在晋头渚 太湖 无锡市 江苏省 中国．Photos by R．A．Peters II，July 2013.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram Matching for Image Restoration}







\section*{Histogram Matching for Image Restoration}

Degraded image


Another image of the same scene，not degraded


Lotus Flowers at Turtle Head Park，Lake Tai，Wuxi，Jiangsu Province，China．莲花在晋头渚 太湖 无锡市 江苏省 中国．Photos by R．A．Peters II，July 2013.

\section*{Histogram Matching for Image Restoration}

Remapped degraded image


Another image of the same scene，not degraded


Lotus Flowers at Turtle Head Park，Lake Tai，Wuxi，Jiangsu Province，China．莲花在晋头渚 太湖 无锡市 江苏省 中国．Photos by R．A．Peters II，July 2013.

\section*{Histogram Matching for Image Restoration}

Left Image


Right Image


Images from a stereo pair of inexpensive web cams. Such cameras have different color characteristics of-the-shelf. Once can be corrected to match the other using histo. matching.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram Matching for Image Restoration}







\section*{Histogram Matching for Image Restoration}

Left Image


Right Image


Images from a stereo pair of inexpensive web cams. Such cameras have different color characteristics of-the-shelf. Once can be corrected to match the other using histo. matching.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Histogram Matching for Image Restoration}

\section*{Left Image}


Right Image


Right image histogram matched to left image.

\title{
EECE 253 Image Processing
}

\section*{Lecture Notes on Color Perception}

\author{
Richard Alan Peters II \\ Department of Electrical Engineering and \\ Computer Science \\ Fall Semester 2014
}

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Color Images}

। Are constructed from three intensity maps.
- Each intensity map is projected through a color filter (e.g., red, green, or blue, or cyan, magenta, or yellow) to create a single color image.
The intensity maps are overlaid to create a color image.
- Each pixel in a color image is a three element vector.


\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Color Images on a CRT or LCD Display}


Projected image primary colors: red, green, and blue.

Intensity images are projected through dot-array color filters which are slightly offset from one another.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Images on a CRT or LCD Display}

Photographs of various displays, showing various pixel geometries. Clockwise from top left, a standard definition CRT television, a CRT computer monitor, a laptop LCD, and the OLPC XO-1 LCD display. [Peter Halasz (user:Pengo), Wikipedia, http://en.wikipedia.org/wiki/Pixel geometry]


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Color Images In Print}


Images are separated into four color bands, each of which is printed as a grid regularly spaced dots. A dot's diameter varies in proportion to the intensity of the color.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Images in Print}


The four colors are magenta, cyan, yellow, and black

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Standard Halftone Screen Angles}

The dot grids are created with a screen that overlays the intensity images.


The screens are oriented at different angles.
The resulting patterns are called "rosettes".

\section*{Color Separation / Halftoning}


The original is separated into an intensity image for each of the four color bands.


\section*{Color Separation / Halftoning}


\author{
Vanderbilt University School of Engineering
}

\section*{Color Separation / Halftoning}

Each intensity image is multiplied by a corresponding "screen",


Each screened image is printed in its own color on the same page.


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Separation / Halftoning}

\section*{Results: \\ }


\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{The Eye}


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

Diagram from http://webvision.med.utah.edu/

\section*{The Retina}

When measured in 104 healthy people, the horizontal angle from the center of the fovea to the meridian through the center of the optic nerve head varied from \(13.0^{\circ}\) to \(17.9^{\circ}\); the vertical angle from the foveal center to the parallel through the optic nerve head was in the range -3.65 to \(+0.65^{\circ} .{ }^{1}\)


Diagram from http://webvision.med.utah.edu/

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{The Retina}


Fig. 2. Simple diagram of the organization of the retina.
Diagram from http://webvision.med.utah.edu/

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Retinal Mosaic}


Cepko, Connie, "Giving in to the blues", Nature Genetics, 24, 99-100 (2000) cepko@genetics.med.harvard.edu

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Photoreceptor Densities}


Fig. 20. Graph to show rod and cone densities along the horizontal meridian.

Diagrams from http://webvision.med.utah.edu/


Fig. 21. Cone densities in human retina as revealed in whole mount. The foveal area is enlaged in B. (from Curcio et al., 1987).

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Photoreceptor Densities}

The density of cone photoreceptors decreases from the high-resolution fovea to the periphery of the eye. A human eye's field of view is about \(155^{\circ}\) of that, the fovea comprises the central \(2^{\circ}\). To see the world in detail requires active scanning by the eyes. A person does not see much more than he or she does see in most situations. The slides that follow mimic a multiresolution scan of a painting by a single eye. (The digital image processing in this case was done with a log-polar transform.)

Figure: Anatomical Distribution of Rods and Cones from Neuroscience. 2nd edition.
Purves D, Augustine GJ, Fitzpatrick D, et al., editors.
 Sunderland (MA): Sinauer Associates; 2001. http://www.ncbi.nlm.nih.gov/books/NBK10848/

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retinal Space-Variant Sensing}


\section*{Retinal Space-Variant Sensing}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{The Log Polar Transform}
\(\mathbf{L}(\rho(x, y), \phi(x, y))=\mathbf{I}(x, y)\), where
\[
\rho(x, y)=M \log (r(x, y)+\alpha)-\beta \text {, with }
\]
\[
\mathrm{r}(x, y)=\sqrt{\left(x-x_{0}\right)^{2}+\left(y-y_{0}\right)^{2}}, \text { and }
\]
\[
\phi(x, y)=\tan ^{-1}\left(\frac{y-y_{0}}{x-x_{0}}\right) .
\]
\(M, \alpha, \beta \in \mathbb{R}^{+}\)such that \(\beta \geq M \log \alpha\),

(a)

(e)
(a) Original image with a left-handed coordinate system originating at \(\left(\mathrm{x}_{0}, \mathrm{y}_{0}\right)\), the center of the image. (b) Fovea (actual size for this particular transform).
(c) Fovea (enlarged \(2 \times\) ). (d) Log-polar transform (LPT), unwrapped (actual size). The origin is in the upper left-hand corner. The \(\rho\)-axis is down and the \(\varphi\)-axis is to the right. (e) LPT, unwrapped, enlarged \(2 \times\). (f) LPT on cylinder. The origin is at the bottom. The \(\rho\)-axis is up and the \(\varphi\)-axis is clockwise to the left.
(g) LPT image backward-mapped onto the original image.

(g)

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Space Variant Visual Motion}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Retina: CenterSurround Edge Detector}

The interconnection of the photoreceptors by the other cells in the retina cause its output to be an edge map, similar to the action of a Laplacian of Gaussian filter on a digital image.


\[
\nabla^{2} g(r)=\frac{1}{\pi \sigma^{4}}\left[1-\frac{r^{2}}{2 \sigma^{2}}\right] \exp \left(-\frac{r^{2}}{2 \sigma^{2}}\right)
\]
\[
G(\omega)=\omega^{2} \exp \left(-\frac{\sigma^{2} \omega^{2}}{2}\right)
\]

Laplacian of Gaussian
(LOG) Filter
\[
r^{2}=x^{2}+y^{2}
\]

\[
\omega^{2}=u^{2}+v^{2}
\]

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Retinal Edge Detection}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Space Variant Retinal Edge Detection} Louis Boilly (1761-1845) Thirty-Six Faces of Expression. Photo negative of LoG output.


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Retinal Transform Minimizes Data Bandwidth}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Pixelization of Color Images: All Bands Equal}


\author{
Vanderbilt University School of Engineering
}

\section*{\(16 \times\) Pixelization of Color Images: R, G, \& B Bands}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Visual Areas in the Brain}

Retina: center-surround color feature detectors
LGN: (lateral geniculate nucleus) relay to V1; audio attention
V1: selective spatiotemporal filters
V2: feature aggregation
V4: visual attention
IT: (Inferior temporal gyrus) complex object features


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{In the Brain: from RGB to LHS}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{\(16 \times\) Pixelization of Color Images: Luminance Only}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{\(16 \times\) Pixelization} of Color Images: Chrominance (H+S) Only

(c) 1999-2014 by Richard Alan Peters II

Photo: R. A. Peters II, 1998, The Lake, Central Park, NYC.

\author{
Vanderbilt University School of Engineering
}


\author{
Vanderbilt University School of Engineering
}


cc) 1999-2014 by Richard Alan Peters II

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Sensing / Color Perception}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Color Sensing / Color Perception}



The simultaneous red + blue response causes us to perceive a continuous range of hues on a circle. No hue is greater than or less than any other hue.

Vanderbilt University School of Engineering

\section*{Complementary Colors}


Colors opposite each other on the color disk are called "complementary".

\title{
EECE 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Complementary Colors}

photoreceptor response is represented as proportional to brightness

To complementary colors, the response of the retina's photoreceptors is opposite.

\section*{Color Perception: The Afterimage Effect}


Stare at the dot in the center of the image




\author{
EECE 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Perception: The Afterimage Effect}


The color "negatives" saturate the local receptors so that when the color is removed the agonist (opposite) color receptors remain saturated.

\section*{Color Perception: the Cornsweet Effect}


Dale Purves, R. Beau Lotto, Surajit Nundy, "Why We See What We Do", American Scientist, Volume 90, No. 3, May-June 2002

\section*{Color Perception: the Cornsweet Effect}


Dale Purves, R. Beau Lotto, Surajit Nundy, "Why We See What We Do", American Scientist, Volume 90, No. 3, May-June 2002

\section*{Color Perception: the Cornsweet Effect}


Wrong!

Dale Purves, R. Beau Lotto, Surajit Nundy, "Why We See What We Do", American Scientist, Volume 90, No. 3, May-June 2002

\section*{Color Perception: the Cornsweet Effect}


Dale Purves, R. Beau Lotto, Surajit Nundy, "Why We See What We Do", American Scientist, Volume 90, No. 3, May-June 2002

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Perception: the Munker Illusion}


Blue and green spirals?

\section*{Color Perception: the Munker Illusion}


No. Blue-green spirals.

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Brightness Perception}

image

intensity profile

Linear intensity changes are not seen as such.

\title{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Brightness Perception}

The previous slide demonstrates the WeberFechner relation. The linear slope of the intensity change is perceived as logarithmic.
\[
\Delta g=\frac{\left|g_{1}-g_{2}\right|}{g_{1}+g_{2}}
\]

The green curve is the actual intensity; the blue curve is the perceived intensity.


\section*{Uniform Change in Frequency and Contrast}


\section*{An Excellent and Amusing Website}

\section*{Optical Illusions \& Visual Phenomena}
\begin{tabular}{|l|}
\hline\(\rightarrow\) Tour \\
\hline\(\rightarrow\) New \\
\hline
\end{tabular}

\author{
113 of them - by Michael Bach (G+) \\ [other languages: Bulgarian, German, Russian]
}

Optical illusion are fascinating! They also teach us about our visual perception, and its limitations. Emphasis here is on the beauty of perceptual phenomena, on interactive experiments, and explanation of the visual mechanisms involved - to the degree that they are understood

Befriending mobile devices: >70 interactive demos now without Flash, but requiring up-to-date browser versions.

Don't let it irk you if you don't see all the phenomena described. For many illusions, there is a percentage of people with perfectly normal vision who just don't see it, often for reasons currently unknown.
If you are not a vision scientist, you might find my explanation attempts too highbrow. That is not on purpose, but vision research is not trivial, like any science. So, if the explanation seems gibberish, simply enjoy the phenomenon ;-). More: Bach \& Poloschek (2006) Optical Illusions Primer.
»Optical illusion« sounds pejorative, as if exposing a malfunction of the visual system. Rather, I view these phenomena as highlighting particular good adaptations of our visual system to experience with standard viewing situations. These experiences are based on normal visual experiences, and thus under unusual contexts can lead to inappropriate interpretations of a visual scene ( \(=\) Bayesian interpretation of perception).

Before we delve in, I'd like to express my thanks for your @feedback; any advice is appreciated.


\title{
EECE 253 Image Processing
}

\section*{Lecture Notes: Color Correction}

\author{
Richard Alan Peters II \\ Department of Electrical Engineering and \\ Computer Science \\ Fall Semester 2014
}

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Images}

। Are constructed from three overlaid intensity maps.
। Each map represents the intensity of a different "primary" color.
। The actual hues of the primaries do not matter as long as they are distinct. The primaries are 3 vectors (or axes) that form a "basis" of the color space.


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Vector-Valued Pixels}


Each color corresponds to a point in a 3D vector space

\section*{Color Space}

\section*{for standard digital images}
- primary image colors red, green, and blue
- correspond to R,G, and B axes in color space.
- 8-bits of intensity resolution per color
- correspond to integers 0 through 255 on axes.
- no negative values
- color "space" is a cube in the first octant of 3-space.
- color space is discrete
\(-256^{3}\) possible colors \(=16,777,216\) elements in cube.

\section*{Different Axis Sets in Color Space}


RGB axes


CMY axes

\section*{Color With Respect To Different Axes}


The same color has different RGB and CMY coordinates.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Correction}

Global changes in the coloration of an image to alter its tint, its hues or the saturation of its colors with minimal changes to its luminant features


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of Frozen Impact (PorchLight Entertainment, 2003).

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}
red \(\gamma=2\)



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Gamma Adjustment of Color Bands}
red \(\gamma=0.5\)


reduced red \(=\) increased cyan

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands | green \(\gamma=0.5\)}




EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}
blue \(\gamma=2\)



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gamma Adjustment of Color Bands}
blue \(\gamma=0.5\)


reduced blue \(=\) incr. yellow

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Images}
are represented by three bands (not uniquely) e.g., R, G, \& B or L, a*, \& b*.


Red
Green
Blue
B


Luminance a*-chroma b*-chroma

EECE 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{RGB to LHS: A Perceptual Transformation}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Maxwell's Triangle}

Probably the first attempts to produce color curves describing the trichromatic theory of color were those by Maxwell (1857, 1860).... [The] first chromaticity diagram was a circle devised by Newton. Later, Maxwell used an equilateral triangle.... In his trichromatic theory, each of the three primary colors-red, green, and blue-is located at a corner of the triangle. The white color is in the middle. Other colors are formed by a combination of the \(r, g, b\) components depending on the distances from each of the three sides of the triangle. This triangular representation has been used often with several modifications.

D. Malacara-Hernandez, Color Vision and Colorimetry: Theory and Applications, SPIE Press, (2002).

\section*{Brightness + Chrominance Representation}

There are many different ways to encode color in terms of 1D brightness and 2D chrominance. Chrominance is usually represented in terms of hue and saturation. A given brightness measure (e.g. value or NTSC luminance) defines a planar surface in the color cube on which the brightness is constant. One point on that surface is gray. The saturation of any color with the given brightness is defined as the distance on the plane from the color to the gray point. The hue is defined as the angular deviation from red measured in the same plane.


\section*{Brightness + Chrominance Representation}

The HSV encoding scheme presented in the first part of this lecture is a direct implementation of the vector math. Although it is nonstandard, it demonstrates the ideas that underlie most of these representations.

For a good explanation of more standard HSV and LHS representations please see: HSL and HSV - Wikipedia, the free encyclopedia.

A brief explanation of the hexagonal representation of HSV is given later in this lecture.


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Equivalue Color Triangle}

A plane through the colors
\[
\left[\begin{array}{l}
r \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
c \\
0 \\
0
\end{array}\right],\left[\begin{array}{l}
r \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
0 \\
c \\
0
\end{array}\right] \text {, and }\left[\begin{array}{l}
r \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
0 \\
0 \\
c
\end{array}\right] \text {, }
\]
forms a triangle inside the color cube if \(c \leq 255\) or \(c \geq 510\), or a hexagon if \(255<c<510\). Every color on the planar surface is such that \(r+g+b=c\). Therefore its value is \(c / 3\). It is on this equivalue plane that hue and saturation are computed.


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Equivalue Color Triangle}
```

c=255,v=85

```

On the \(g=0\) face of the cube the triangle traces the line, \(r+b=255\).


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Equivalue Color Hexagon}

A plane through the colors
\[
\left[\begin{array}{l}
r \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
c \\
0 \\
0
\end{array}\right],\left[\begin{array}{l}
r \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
0 \\
c
\end{array}\right], \text { and }\left[\begin{array}{l}
r \\
0 \\
g \\
b
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right], \quad \text { R-25s }
\]
forms a triangle inside the color cube if \(c \leq 255\) or \(c \geq 510\), or a hexagon if \(255<c<510\). Every color on the planar surface is such that \(r+g+b=c\). Therefore its value is \(c / 3\). It is on this equivalue plane that hue and saturation are computed.


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Equivalue Color Hexagon}

Outside looking in.
\(c=383, v=128\)

On the \(g=0\) face of the cube the hexagon traces the line, \(r+b=383\).


\section*{Equivalue Color Hexagon}

Outside looking in.
\[
c=383, v=128
\]

On the \(g=255\) face of the cube the hexagon traces the line, \(r+b+g=383\).
\[
\mathrm{B}=0
\]

LEFT SIDE
\[
G=255
\]


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Equivalue Color Hexagon}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Cube: Equivalue Triangle}

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Cube: Equivalue Triangle}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{HSV Color Representation}


\author{
Vanderbilt University School of Engineering
}

\section*{Color Point on Equivalue Triangle}


Vanderbilt University School of Engineering

\section*{Color Vector Associated with Point}


Vanderbilt University School of Engineering

\section*{Color Coordinates and Component Vectors}


\author{
Vanderbilt University School of Engineering
}

\section*{Color Cube, Equivalue Triangle, \& Gray Line}


\section*{Color Point and Gray Line}


\section*{Saturation Component of Color Vector}


\author{
Vanderbilt University School of Engineering
}

\section*{Saturation and Value Components of Color Vector}


\author{
Vanderbilt University School of Engineering
}

\section*{Hue, Saturation and Value}


\section*{Hue and Saturation on Equivalue Plane}


\section*{Hue, Saturation and Value with Gray Line}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{HSV Color Representation}


\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

Equivalue plane at \(v=0\) : single point, pure black.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

Equivalue plane at \(v=85\) : largest upright triangle, start of hexagonal intersections.

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

Equivalue plane at \(v=128\) : symmetric hexagon, intersecting plane with largest area.

\author{
EECE 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

Vanderbilt University School of Engineering

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

Equivalue plane at \(v=170\) : largest inverted triangle, end of hexagonal intersections.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{Equivalue Plane Intersecting Color Cube}


Projection: the gray line is perpendicular to this page.

\section*{RGB to HSV Conversion}
\(\mathbf{v}_{0}=\frac{1}{3}\left[\begin{array}{l}c \\ c \\ c\end{array}\right]\), where \(c=r_{0}+g_{0}+b_{0}\).
\(v_{0}=\frac{1}{3} c\), whereas \(\left\|\mathbf{v}_{0}\right\|=\frac{\sqrt{3}}{3} c\).
\[
\mathbf{s}_{0}=\mathbf{p}_{0}-\mathbf{v}_{0}=\left[\begin{array}{c}
r_{0}-v_{0} \\
g_{0}-v_{0} \\
b_{0}-v_{0}
\end{array}\right] . \quad \mathbf{p}_{0}=\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{0}
\end{array}\right] .
\]

\[
s_{0}=\left\|\mathbf{s}_{0}\right\|=\sqrt{\left(r_{0}-v_{0}\right)^{2}+\left(g_{0}-v_{0}\right)^{2}+\left(b_{0}-v_{0}\right)^{2}} .
\]

\section*{RGB to HSV Conversion}
\[
\lceil c\rceil
\]
\(\mathrm{c} / 3\) is the usual value- re \(c=r_{0}+g_{0}+b_{0}\). image intensity (the average of \(r, g, \& b\) ) ...
\[
v_{0}=\frac{1}{3} c \text {, whereas }\left\|\mathbf{v}_{0}\right\|=\frac{\sqrt{3}}{3} c \text {. }
\]

\[
s_{0}=\left\|\mathbf{s}_{0}\right\|=\sqrt{\left(r_{0}-v_{0}\right)^{2}+\left(g_{0}-v_{0}\right)^{2}+\left(b_{0}-v_{0}\right)^{2}} .
\]

\section*{RGB to HSV Conversion}
\[
\begin{aligned}
& \mathbf{x}=\mathbf{R}-\mathbf{v}_{0}=\left[\begin{array}{l}
c \\
0 \\
0
\end{array}\right]-\frac{1}{3}\left[\begin{array}{l}
c \\
c \\
c
\end{array}\right]=\frac{c}{3}\left[\begin{array}{r}
2 \\
-1 \\
-1
\end{array}\right] \\
& \mathbf{s}_{0}=\mathbf{p}_{0}-\mathbf{v}_{0}=\left[\begin{array}{l}
r_{0}-v_{0} \\
g_{0}-v_{0} \\
b_{0}-v_{0}
\end{array}\right] . \\
& h_{0}=\angle\left(\mathbf{s}_{0}, \mathbf{x}\right)=\cos ^{-1}\left(\frac{\mathbf{s}_{0} \cdot \mathbf{x}}{\left\|\mathbf{s}_{0}\right\|\|\mathbf{x}\|}\right) .
\end{aligned}
\]


\section*{RGB to HSV Conversion}
\[
\mathbf{x}=\mathbf{R}-\mathbf{v}_{0}=\left[\begin{array}{l}
c \\
0 \\
0
\end{array}\right]-\frac{1}{3}\left[\begin{array}{l}
c \\
c \\
c
\end{array}\right]=\frac{c}{3}\left[\begin{array}{c}
2 \\
-1 \\
-1
\end{array}\right]
\]

Note that:
(1) For \(c>85\), the red vector, \(\mathbf{x}\) extends beyond the color cube.
(2) Vector \(\mathbf{x}=0\) if and only if \(c=0\).


\section*{RGB to HSV Conversion}

In summary,
\[
v_{0}=\frac{1}{3} c
\]
where \(c=r_{0}+g_{0}+b_{0}\), the sum of the components of \(\mathbf{p}_{0}\).
\[
s_{0}=\sqrt{\left(r_{0}-v_{0}\right)^{2}+\left(g_{0}-v_{0}\right)^{2}+\left(b_{0}-v_{0}\right)^{2}}
\]
and
\[
h_{0}=\cos ^{-1}\left(\frac{\mathbf{s}_{0} \cdot \mathbf{x}}{\left\|\mathbf{s}_{0}\right\|\|\mathbf{x}\|}\right)
\]

Usually, \(s_{0}\) is normalized to lie in the interval \((0,1)\) and \(h_{0}\) is shifted to lie in \((0,2 \pi)\).


\section*{EECE 253 Image Processing}

\section*{Normalizing the Saturation}

The scalar saturation,
\[
s_{0}=\sqrt{\left(r_{0}-v_{0}\right)^{2}+\left(g_{0}-v_{0}\right)^{2}+\left(b_{0}-v_{0}\right)^{2}},
\]
usually is normalized to lie between 0 and 1 . There are a number of possible ways to do this. One is to use the largest possible length of a saturation vector in the color cube. That vector lies in the triangle with vertices \(\left[\begin{array}{lll}r & g & b\end{array}\right]^{\top}=\) [255 000 边 \({ }^{\top}\), [ \(\left.\begin{array}{lll}0 & 255 & 0\end{array}\right]^{\top}\), and \(\left[\begin{array}{lll}0 & 0 & 255\end{array}\right]^{\top}\). There are 3 such vectors, from the gray point to pure red, pure green, or pure blue. The red one is
\[
\mathbf{s}_{\max }=\left[\begin{array}{lll}
255 & 0 & 0
\end{array}\right]^{\top}-\frac{1}{3}\left[\begin{array}{lll}
255 & 255 & 255
\end{array}\right]^{\top}=\left[\begin{array}{lll}
170 & -85 & -85
\end{array}\right]^{\top},
\]
which has length \(s_{\max }=\left\|s_{\max }\right\| \approx 208.2066\).


Therefore, \(s_{0}\) is replaced by \(S_{0} / s_{\text {max }}\)

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Other Steps to Include in RGB \(\rightarrow\) HSV}

There are a few places in the algorithm where computation can be problematic. These include division by zero, exceeding limits due to round-off errors, and values returned by library functions that are inconsistent with the RGB \(\rightarrow\) HSV algorithm.

Recall that the hue calculation for color, \(\mathbf{p}\), requires division by the product of the length of the red vector, \(\mathbf{x}(\mathbf{p})\), and the length of the saturation vector, \(\mathbf{s}(\mathbf{p})\). If either of these is 0 , then the argument could be undefined. Matlab returns NaN for acos(0/0).
\[
h=\theta=\cos ^{-1}\left(\frac{\mathbf{s} \cdot \mathbf{x}}{\|\mathbf{s}\|\|\mathbf{x}\|}\right)
\]
\(\mathbf{s}\) is the zero vector whenever \(\mathbf{p}\) is a gray level (all its components are equal). Vector \(\mathbf{x}=\mathbf{0}\) if and only if color \(\mathbf{p}=\mathbf{0}\). The workaround is to add 1 to the denominator at every pixel where either \(\mathbf{p}=0\) or \(s=0\). Then set \(h=0\) at those same pixel locations.

\section*{Other Steps to Include in RGB \(\rightarrow\) HSV}

Inverse cosine routines, like Matlab's acos, return angles in the range \([0, \pi]\), whereas the RGB \(\rightarrow\) HSV algorithm needs them to be in the range \([0,2 \pi)\).

Note that \(\theta \in[0, \pi]\) if and only if \(b \leq g\) in \(\mathbf{p}\) and \(\theta \in(-\pi, 0)\) if and only if \(b>g\). Therefore the workaround is to let:
\[
h=\left\{\begin{array}{c}
\theta \text { if } b \leq g, \\
2 \pi-\theta \text { if } b>g,
\end{array}\right.
\]
where
\[
\theta=\cos ^{-1}\left(\frac{\mathbf{s} \cdot \mathbf{x}}{\|\mathbf{s}\|\|\mathbf{x}\|}\right), \quad \mathbf{p}=\left[\begin{array}{c}
r \\
g \\
b
\end{array}\right] .
\]


\section*{HSV to RGB Conversion}

The equivalue plane is perpendicular

This conversion requires a change of coordinates through a rotation and a translation. to the value vector, \(\mathbf{v}\).
The plane contains vector \(\mathbf{x}\) defined on slide 81.
Therefore, \(\mathbf{v}\) is perpendicular to \(\mathbf{x}\) and \(\mathbf{y}=\mathbf{v} \times \mathbf{x}\) is also in the plane.
If we keep the directions but ignore the magnitudes, the unit vectors
\[
\hat{\mathbf{x}}=\left[\begin{array}{l}
1 \\
0 \\
0
\end{array}\right], \quad \hat{\mathbf{y}}=\left[\begin{array}{l}
0 \\
1 \\
0
\end{array}\right], \quad \text { and } \hat{\mathbf{v}}=\left[\begin{array}{l}
0 \\
0 \\
1
\end{array}\right],
\]
form an orthonormal basis with respect to the equivalue plane.

\section*{HSV to RGB Conversion}

Given values \(h, s\), and \(v\), where

Since the \(x\) - and \(y\)-axes lie in the equivalue plane and the \(c d t\). origin is the gray point, we set \(v=0\) for now. the saturation vector is
\[
[\mathbf{s}]_{\mathrm{xyv}}=\left[\begin{array}{c}
s \cos (h) \\
s \sin (h) \\
0
\end{array}\right]_{\mathrm{xyv}},
\]
with respect to unit vectors \(\hat{\mathbf{x}}, \hat{\mathbf{y}}\), and \(\hat{\mathbf{v}}\), in the equivalue plane.
\[
\mathbf{s}=s \cos (h) \hat{\mathbf{x}}+s \sin (h) \hat{\mathbf{y}}+0 \hat{\mathbf{v}} .
\]

\section*{HSV to RGB Conversion}

Given values \(h, s\), and \(v\), where
If \(s\) is in the range 0 to 1 , then it must be denormalized first by multiplying by smax.
\[
h \in[0,2 \pi), \quad s \in\left[0, s_{\max }\right], \text { and } v \in[0,255] \text {, }
\] the

These are the coordinates of \(s\) with respect to \(\hat{x}, \hat{y}, \& \hat{v}\).
\[
[\mathbf{s}]_{\mathrm{xyv}}=\left[\begin{array}{c}
s \cos (h) \\
s \sin (h) \\
0
\end{array}\right]_{\mathrm{xyv}},
\]
with res This is \(s\) written as a linear and \(\hat{\mathbf{v}}\), it combination of vectors \(\hat{x}, \hat{y}, \& \hat{\text { equivalue piaile. }}\)
\[
\mathbf{s}=s \cos (h) \hat{\mathbf{x}}+s \sin (h) \hat{\mathbf{y}}+0 \hat{\mathbf{v}} .
\]

\section*{HSV to RGB Conversion}
\(\hat{\mathbf{x}}, \hat{\mathbf{y}}\), and \(\hat{\mathbf{v}}\), are not in the same directions as the red, green, and blue unit vectors, \(\hat{\mathbf{r}}, \hat{\mathbf{g}}\), and \(\hat{\mathbf{b}}\). Therefore, \([\mathbf{s}]_{\mathrm{xyv}}\), which we know, is not equal to \([\mathrm{s}]_{\mathrm{rgb}}\), which we do not know, but need in order to find the color, \(\mathbf{p}_{0}\), with respectt fo \(\mathbf{r}, \mathbf{g}\), and \(\mathbf{b}\).
\[
\begin{aligned}
{[\mathbf{s}]_{\mathrm{rgb}} } & =\left[\begin{array}{lll}
r_{0} & g_{0} & b_{0}
\end{array}\right]^{\top} \\
\mathbf{s} & \leftrightarrow r_{0} \hat{\mathbf{r}}+g_{0} \hat{\mathbf{g}}+b_{0} \hat{\mathbf{b}}, \\
\mathbf{s} & \leftrightarrow s \cos (h) \hat{\mathbf{x}}+s \sin (h) \hat{\mathbf{y}}+0 \hat{\mathbf{v}} .
\end{aligned}
\]


\section*{EECE 253 Image Processing}

\section*{HSV to RGB Conversion}
... which means we have to find the \(x, y, \& v\) vectors in terms of the \(r, g, \& b\) vectors.

Vector \(\mathbf{s}\) written as a linear combination of vectors, \(\hat{\mathbf{r}}, \hat{\mathbf{g}}\), and \(\hat{\mathbf{b}}\), and \(\mathbf{s}\) written as a linear combination of vectors, \(\hat{\mathbf{x}}, \hat{\mathbf{y}}\), and \(\hat{\mathbf{v}}\) both refer to the same point on the equivalue plane.
\[
\begin{aligned}
& \mathbf{s} \leftrightarrow r_{0} \hat{\mathbf{r}}+g_{0} \hat{\mathbf{g}}+b_{0} \hat{\mathbf{b}}, \\
& \mathbf{s} \leftrightarrow s \cos (h) \hat{\mathbf{x}}+s \sin (h) \hat{\mathbf{y}}+0 \hat{\mathbf{v}} .
\end{aligned}
\]

The specific numbers in \([\mathbf{s}]_{\text {rgb }}\) and in \([\mathrm{s}]_{\mathrm{xyv}}\) (that represent the point w.r.t. the two coordinate systems) are, however, different.
\[
\begin{aligned}
& {[\mathbf{s}]_{\mathrm{rgb}}=\left[\begin{array}{lll}
r_{0} & g_{0} & b_{0}
\end{array}\right]^{\top} \text { and }} \\
& {[\mathbf{s}]_{\mathrm{xyz}}=\left[\begin{array}{lll}
s \cos (h) & s \sin (h) & 0
\end{array}\right]^{\top} \text { but }[\mathbf{s}]_{\mathrm{rgb}} \neq\left[\begin{array}{l}
\mathbf{s}
\end{array}\right]_{\mathrm{xyz}}}
\end{aligned}
\]

\section*{HSV to RGB Conversion}

We can find \(r_{0}, g_{0}\), and \(b_{0}\), from \(h_{0}, s_{0}\), \(\hat{r}\) and \(v_{0}\), if we know how the unit vectors \(\hat{\mathbf{x}}, \hat{\mathbf{y}}\), and \(\hat{\mathbf{v}}\), are expressed with respect to \(\hat{\mathbf{r}}, \hat{\mathbf{g}}\), and \(\hat{\mathbf{b}}\). That relationship is in the form of a rotation matrix, \(A\), such that,
\[
[\hat{\mathbf{x}}]_{\mathrm{rbb}}=A[\hat{\mathbf{x}}]_{\mathrm{xy}}, \quad[\hat{\mathbf{y}}]_{\mathrm{rgb}}=A[\hat{\mathbf{y}}]_{\mathrm{xy}}, \quad[\hat{\mathbf{v}}]_{\mathrm{rbb}}=A[\hat{\mathbf{v}}]_{\mathrm{xyv}} \cdot \hat{\mathbf{y}}
\]

Then,
\[
\begin{aligned}
{[\mathbf{s}]_{\mathrm{rgb}} } & =A[\mathbf{s}]_{\mathrm{xyv}} \\
& =A\left[s \cos (h)[\hat{\mathbf{x}}]_{\mathrm{xyv}}+s \sin (h)[\hat{\mathbf{y}}]_{\mathrm{xvv}}+0[\hat{\mathbf{v}}]_{\mathrm{xvv}}\right] \\
& =s \cos (h) A[\hat{\mathbf{x}}]_{\mathrm{xyv}}+s \sin (h) A[\hat{\mathbf{y}}]_{\mathrm{xyv}}+0 A[\hat{\mathbf{v}}]_{\mathrm{xyv}} \\
& =s \cos (h)[\hat{\mathbf{x}}]_{\mathrm{rgb}}+s \sin (h)[\hat{\mathbf{y}}]_{\mathrm{rgb}}+0[\hat{\mathbf{v}}]_{\mathrm{rgb}} .
\end{aligned}
\]

\section*{HSV to RGB Conversion}

When written w.r.t the \(\mathbf{x y z}\) coordinate system we have
\[
\hat{\mathbf{x}}=\left[\begin{array}{l}
1 \\
0 \\
0
\end{array}\right], \quad \hat{\mathbf{y}}=\left[\begin{array}{l}
0 \\
1 \\
0
\end{array}\right], \quad \text { and } \hat{\mathbf{v}}=\left[\begin{array}{l}
0 \\
0 \\
1
\end{array}\right],
\]

So that,
\[
[\hat{\mathbf{x}}]_{\mathrm{rgb}}=A\left[\begin{array}{l}
{\left[\begin{array}{l}
1 \\
0 \\
0
\end{array}\right]}
\end{array}, \quad[\hat{\mathbf{y}}]_{\mathrm{rgb}}=A\left[\begin{array}{l}
{\left[\begin{array}{l}
0 \\
0
\end{array}\right]}
\end{array}\right] \quad[\hat{\mathbf{v}}]_{\mathrm{rgb}}=A A_{\left[\begin{array}{l}
0 \\
0
\end{array}\right]}^{[ } .\right.
\]

But that implies,

\[
A=\left[[\hat{\mathbf{x}}]_{\mathrm{rgb}}[\hat{\mathbf{y}}]_{\mathrm{rgb}}[\hat{\mathbf{v}}]_{\mathrm{rgb}}\right] .
\]

The columns of the rotation matrix, \(A\), are the \(x, y, \& v\) unit vectors in \(r, g, \& b\) coordinates.

\section*{HSV to RGB Conversion}

How to find the \(x, y, \& z\) unit vectors in \(r, g, \& b\) coordinates:
\(\hat{\mathbf{v}}\) is the unit vector in the direction [1111边 when written w.r.t rgb coordinates.
\[
[\hat{\mathbf{v}}]_{\mathrm{rgb}}=\frac{\sqrt{3}}{3}\left[\begin{array}{l}
1 \\
1 \\
1
\end{array}\right]
\]

\(\hat{\mathbf{x}}\) is perpendicular to \(\hat{\mathbf{v}}\) and has equal \(\hat{\mathbf{g}}\) and \(\hat{\mathbf{b}}\) components.
\(\hat{\mathbf{y}}\) is the cross product of \(\hat{\mathbf{v}}\) with \(\hat{\mathbf{x}}\).
\[
\begin{gathered}
{[\hat{\mathbf{x}}]_{\mathrm{rgb}}=\frac{\sqrt{6}}{6}\left[\begin{array}{r}
2 \\
-1 \\
-1
\end{array}\right]} \\
{\left[\begin{array}{rl}
{[\hat{\mathbf{y}}]_{\mathrm{rgb}}} & =[\hat{\mathbf{v}}]_{\mathrm{rgb}} \times[\hat{\mathbf{x}}]_{\mathrm{rgb}} \\
& =\frac{\sqrt{2}}{2}\left[\begin{array}{r}
0 \\
1 \\
-1
\end{array}\right]
\end{array}\right.}
\end{gathered}
\]

\section*{HSV to RGB Conversion}

Therefore, the rotation matrix is
\[
A=\frac{\sqrt{6}}{6}\left[\begin{array}{rrr}
2 & 0 & \sqrt{2} \\
-1 & \sqrt{3} & \sqrt{2} \\
-1 & -\sqrt{3} & \sqrt{2}
\end{array}\right] .
\]

Substitute that into the \(2^{\text {nd }}\) equation on slide 94 to get:
\[
\begin{aligned}
{[\mathbf{s}]_{\mathrm{rgb}} } & =s \frac{\sqrt{6}}{6} \cos (h)\left[\begin{array}{c}
2 \\
-1 \\
-1
\end{array}\right]+s \frac{\sqrt{2}}{2} \sin (h)\left[\begin{array}{c}
0 \\
1 \\
-1
\end{array}\right]+0 \frac{\sqrt{3}}{3}\left[\begin{array}{l}
1 \\
1 \\
1
\end{array}\right] \\
& =s \frac{\sqrt{6}}{6} \cos (h)\left[\begin{array}{c}
2 \\
-1 \\
-1
\end{array}\right]+s \frac{\sqrt{2}}{2} \sin (h)\left[\begin{array}{c}
0 \\
1 \\
-1
\end{array}\right] .
\end{aligned}
\]

Finally, \([\mathbf{s}]_{\mathrm{rgb}}\) must be translated to the value vector to obtain
The \(x, y, \& z\) unit vectors in \(r, g\), \& \(b\) coordinates are the columns of the rotation matrix:
 the rgb color of \(\mathbf{p}_{0}\) :
\[
\mathbf{p}_{0}=[\mathbf{p}]_{\mathrm{rgb}}=[\mathbf{s}]_{\mathrm{rgb}}+[\mathbf{v}]_{\mathrm{rgb}} \text {, where } \mathbf{s}_{0}=[\mathbf{s}]_{\mathbf{r g b}} \text { and }[\mathbf{v}]_{\mathbf{r g b}}=\mathbf{v}_{0} \text { as def 'd. on slide } 81 .
\]

\section*{Color Hexagon at Value 128}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Hexagon at Value 128}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Hexagon at Value 128}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Saturation Adjustment}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Saturation Adjustment}

\section*{saturation \(+50 \%\)}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Saturation Adjustment}

\section*{original}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Saturation Adjustment}

\section*{saturation - 50\%}



The \(r, g, \& b\) histograms approach the value histogram as the color fades to grayscale.

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Hue Shifting}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Hue Shifting}



The effects of a hue shift are nonlinear. They difficult to characterize on the \(r, g, \& b\) histograms

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}



\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Hue Shifting}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}


\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}



EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Hue Shifting}


\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Hue Shifting}



\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Hue Shifting}

\begin{tabular}{|l|l|}
\hline \(\mathbf{R} \rightarrow \mathbf{G}\) \\
\(\mathbf{Y} \rightarrow \mathbf{C}\) \\
\(\mathbf{G} \rightarrow \mathbf{B}\) \\
\(\mathbf{C} \rightarrow \mathbf{M}\) \\
\(\mathbf{B} \rightarrow \mathbf{R}\) \\
\(\mathbf{M} \rightarrow \mathbf{Y}\) \\
\hline
\end{tabular}
hue \(+120^{\circ}\)

Hue \(+120^{\circ}\)

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}

Hue + 180
Hue + 180
The part of the histogram that leaves one side appears on the other.

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Hue Shifting}



\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Hue Shifting}



\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Hue Shifting}



\section*{Color Correction via Linear Transformation}
is a point process; the transformation is applied to each pixel as a function of its color alone.
\[
\mathbf{J}(r, c)=\Phi[\mathbf{I}(r, c)], \quad \forall(r, c) \in \operatorname{supp}(\mathbf{I})
\]

Each pixel is vector valued, therefore the transformation is a vector space operator.
\[
\left.\mathbf{I}(r, c)=\left[\begin{array}{l}
\mathbf{R}_{\mathbf{I}}(r, c) \\
\mathbf{G}_{\mathbf{I}}(r, c) \\
\mathbf{B}_{\mathbf{I}}(r, c)
\end{array}\right], \mathbf{J}(r, c)=\left[\begin{array}{l}
\mathbf{R}_{\mathbf{J}}(r, c) \\
\mathbf{G}_{\mathbf{J}}(r, c) \\
\mathbf{B}_{\mathbf{J}}(r, c)
\end{array}\right]=\Phi\{\mathbf{I}(r, c)\}=\Phi\left[\begin{array}{l}
{\left[\begin{array}{l}
\mathbf{R}_{\mathbf{1}}(r, c) \\
\mathbf{G}_{\mathbf{I}}(r, c) \\
\mathbf{B}_{\mathbf{I}}(r, c)
\end{array}\right]}
\end{array}\right]\right\} .
\]

\section*{EECE 253 Image Processing}

Vanderbilt University School of Engineering


\section*{Color Vector Space Operators}

Linear operators are matrix multiplications
\[
\left[\begin{array}{l}
r_{1} \\
g_{1} \\
b_{1}
\end{array}\right]=255 \cdot\left[\begin{array}{l}
\left(r_{0} / 255\right)^{1 / \gamma_{r}} \\
\left(g_{0} / 255\right)^{1 / \gamma_{g}} \\
\left(b_{0} / 255\right)^{1 / \gamma_{b}}
\end{array}\right]
\]
\[
\left[\begin{array}{l}
r_{1} \\
g_{1} \\
b_{1}
\end{array}\right]=\left[\begin{array}{lll}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}\right]\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{0}
\end{array}\right]
\]

Example of a nonlinear operator: gamma correction

\section*{Linear Transformation of Color}

\(\left[\begin{array}{l}r_{1} \\ g_{0} \\ b_{0}\end{array}\right]=\left[\begin{array}{ccc}r_{1} / r_{0} & 0 & 0 \\ 0 & 1 & 0\end{array}\right]\left[\begin{array}{l}r_{0} \\ 0\end{array} 001\right]\left[\begin{array}{l}g_{0} \\ b_{0}\end{array}\right]\)


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Linear Transformation of Color}

\[
\left[\begin{array}{l}
r_{0} \\
g_{1} \\
b_{0}
\end{array}\right]=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & g_{1} / g_{0} & 0 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{0}
\end{array}\right]
\]


\section*{Linear Transformation of Color}

\[
\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{1}
\end{array}\right]=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & b_{1} / b_{0}
\end{array}\right]\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{0}
\end{array}\right]
\]


\section*{Linear Transformation of Color}

\[
\left[\begin{array}{l}
r_{1} \\
g_{1} \\
b_{1}
\end{array}\right]=\left[\begin{array}{ccc}
r_{1} / r_{0} & 0 & 0 \\
0 & g_{1} / g_{0} & 0 \\
0 & 0 & b_{1} / b_{0}
\end{array}\right]\left[\begin{array}{l}
r_{0} \\
g_{0} \\
b_{0}
\end{array}\right]
\]
\[
\left.\left[\begin{array}{l}
175 \\
150 \\
225
\end{array}\right] \stackrel{[ }{125} \begin{array}{r}
75 \\
175
\end{array}\right]
\]


\section*{Color Transformation}

Assume \(\mathbf{J}\) is a discolored version of image \(\mathbf{I}\) such that \(\mathbf{J}=\Phi[\mathbf{I}]\). If \(\Phi\) is
linear then it is represented by a \(3 \times 3\)
matrix, \(\mathbf{A}\) :
\[
\mathbf{A}=\left[\begin{array}{lll}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}\right]
\]

Then \(\mathbf{J}=\) AI or, more accurately, \(\mathbf{J}(r, c)=\mathbf{A I}(r, c)\) for all pixel locations \((r, c)\) in image \(\mathbf{I}\).

\section*{Color Transformation}

If at pixel location \((r, c)\),
image \(\mathbf{I}(r, c)=\left[\begin{array}{l}\rho_{\mathbf{I}} \\ \gamma_{\mathbf{I}} \\ \beta_{\mathbf{I}}\end{array}\right]\) and
image \(\mathbf{J}(r, c)=\left[\begin{array}{l}\rho_{\mathbf{J}} \\ \gamma_{\mathbf{J}} \\ \beta_{\mathbf{J}}\end{array}\right]\),
then \(\mathbf{J}(r, c)=\mathbf{A I}(r, c)\), or
\[
\begin{aligned}
{\left[\begin{array}{l}
\rho_{\mathbf{J}} \\
\gamma_{\mathbf{J}} \\
\beta_{\mathbf{J}}
\end{array}\right] } & =\left[\begin{array}{lll}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}\right]\left[\begin{array}{l}
\rho_{\mathbf{I}} \\
\gamma_{\mathbf{I}} \\
\beta_{\mathbf{I}}
\end{array}\right] \\
& =\left[\begin{array}{ll}
a_{11} \rho_{\mathbf{I}}+a_{12} \gamma_{\mathbf{I}}+a_{13} \beta_{\mathbf{I}} \\
a_{21} \rho_{\mathbf{I}}+a_{22} \gamma_{\mathbf{I}}+a_{23} \beta_{\mathbf{I}} \\
a_{31} \rho_{\mathbf{I}}+a_{32} \gamma_{\mathbf{I}}+a_{33} \beta_{\mathbf{I}}
\end{array}\right] .
\end{aligned}
\]

\section*{EECE 253 Image Processing}

\section*{Color Transformation}

The inverse transform \(\Phi^{-1}\) (if it exists) maps the discolored image, \(\mathbf{J}\), back into the correctly colored version, I, i.e., \(\mathbf{I}=\Phi^{-1}[\mathbf{J}]\). If \(\Phi\) is linear then it is represented by the inverse of matrix \(\mathbf{A}\) :
\[
\begin{aligned}
\mathbf{A}^{-1}= & {\left[a_{11} a_{22} a_{33}-a_{11} a_{23} a_{32}-a_{12} a_{21} a_{33}+\right.} \\
& \left.a_{12} a_{23} a_{31}+a_{13} a_{21} a_{32}-a_{13} a_{22} a_{31}\right]^{-1} \cdot \\
& {\left[\begin{array}{lll}
a_{22} a_{33}-a_{23} a_{32} & a_{13} a_{32}-a_{12} a_{33} & a_{12} a_{23}-a_{13} a_{22} \\
a_{23} a_{31}-a_{21} a_{33} & a_{11} a_{33}-a_{13} a_{31} & a_{13} a_{21}-a_{11} a_{23} \\
a_{21} a_{32}-a_{22} a_{31} & a_{12} a_{31}-a_{11} a_{32} & a_{11} a_{22}-a_{12} a_{21}
\end{array}\right] . }
\end{aligned}
\]

\section*{Color Correction}

Assume we know \(n\) colors in the discolored image, \(\mathbf{J}\), that correspond to another set of \(n\) colors (that we also know) in the original image, \(\mathbf{I}\).
\(\left\{\left[\begin{array}{l}\rho_{\mathbf{J}, k} \\ \gamma_{\mathbf{J}, k} \\ \beta_{\mathbf{J}, k}\end{array}\right]\right\}_{k=1}^{n}\)
known
wrong colors

for \(k=1, \ldots, n\).
known
correspondence
\[
\left\{\left[\begin{array}{l}
\rho_{\mathbf{I}, k} \\
\gamma_{\mathbf{I}, k} \\
\beta_{\mathbf{I}, k}
\end{array}\right]\right\}_{k=1}^{n}
\]
known correct colors

\section*{Color Correction}

To remap the discolored image so that the result matches the original image in a linearly optimal way, we need to find the matrix, \(\mathbf{A}\), that minimizes
\[
\varepsilon^{2}=\sum_{k=1}^{n}\left\|\left[\begin{array}{c}
\rho_{\mathbf{I}, k} \\
\gamma_{\mathbf{I}, k} \\
\beta_{\mathbf{I}, k}
\end{array}\right]-\mathbf{A}^{-1}\left[\begin{array}{c}
\rho_{\mathbf{J}, k} \\
\gamma_{\mathbf{J}, k} \\
\beta_{\mathbf{J}, k}
\end{array}\right]\right\|^{2}
\]

\section*{Color Correction}

To find the solution of this problem, let
\[
\mathbf{Y}=\left[\left[\begin{array}{l}
\rho_{\mathbf{I}, 1} \\
\gamma_{\mathbf{I}, 1} \\
\beta_{\mathbf{I}, 1}
\end{array}\right] \cdots\left[\begin{array}{c}
\rho_{\mathbf{I}, n} \\
\gamma_{\mathbf{1}, n} \\
\beta_{\mathbf{I}, n}
\end{array}\right]\right] \text {, and } \mathbf{X}=\left[\left[\begin{array}{c}
\rho_{\mathbf{J}, 1} \\
\gamma_{\mathbf{J}, 1} \\
\beta_{\mathbf{J}, 1}
\end{array}\right] \ldots\left[\begin{array}{c}
\rho_{\mathbf{J}, n} \\
\gamma_{\mathbf{J}} \\
\beta_{\mathbf{J}, n}
\end{array}\right]\right] \text {. }
\]

Then \(\mathbf{X}\) and \(\mathbf{Y}\) are known \(3 \times n\) matrices such that
\[
\mathbf{Y} \approx \mathbf{A}^{-1} \mathbf{X}
\]
where \(\mathbf{A}\) is the \(3 \times 3\) matrix that we want to find.

\section*{EECE 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Color Correction}

The linearly optimal solution is the least mean squared solution that is given by
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}
\]
where \(\mathbf{X}^{\top}\) represents the transpose of matrix \(\mathbf{X}\).
Notes: 1. n, the number of color pairs, must be \(\geq 3\),
2. \(\mathbf{X} \mathbf{X}^{\top}\) must be invertible, i.e., \(\operatorname{rank}\left(\mathbf{X} \mathbf{X}^{\top}\right)=3\),
3. If \(\mathrm{n}=3\), then \(\mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}=\mathbf{X}^{-1} \xrightarrow{\text { important }}\)

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Color Correction}

\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}
\]
where \(\mathbf{X}^{\top}\) represe output colors (wanted): of matrix \(\mathbf{X}\).
Notes:
or pairs, must be \(\geq 3\),
2. \(\mathbf{X} \mathbf{X}^{\top}\) must be invertible, i.e., \(\operatorname{rank}\left(\mathbf{X X}^{\top}\right)=3\),
3. If \(n=3\), then \(\mathbf{X}^{\top}\left(\mathbf{X X}^{\top}\right)^{-1}=\mathbf{X}^{-1}\).

\section*{EECE 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Color Correction}

where \(\mathbf{X}^{\top}\) repres \({ }^{\text {output colore (wanted): }}\) of matrix \(\mathbf{X}\).
Notes: 1. \(\left.n, 4 \begin{array}{lll}\gamma_{1,1} & \cdots & \gamma_{1, n} \\ \beta_{1,1}\end{array}\right]\)
2. \(\mathbf{X} \mathbf{X}^{\top}\) must be invertible, i.e., \(\operatorname{rank}\left(\mathbf{X X}^{\top}\right)=3\),
3. If \(n=3\), then \(\mathbf{X}^{\top}\left(\mathbf{X X}^{\top}\right)^{-1}=\mathbf{X}^{-1}\).

\section*{Color Correction}

Then the image is color corrected by performing
\[
\mathbf{I}(r, c)=\mathbf{B} \mathbf{J}(r, c), \text { for all }(r, c) \in \operatorname{supp}(\mathbf{J}) .
\]

In Matlab this is easily performed by
I = reshape(((B*(reshape(J, R*C,3))')'),R,C,3);
where \(\mathbf{B}=\mathbf{A}^{-1}\) is computed directly through the LMS formula on the previous page, and \(R \& C\) are the number of rows and columns in the image.

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Linear Color Correction}

NASA Summer Faculty Fellows at Ellington Air Force Base, Houston, TX, July 2002. Airplane is a T-38.


Original Image

"Aged" Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Mapping 1}


Original Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Mapping 2}


Original Image
"Aged" Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Mapping 3}


Original Image
"Aged" Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Mapping 4}


Original Image

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Transformations}


The aging process was a transformation, \(\Phi\), that mapped:
\[
\left[\begin{array}{l}
17 \\
122 \\
114
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
17 \\
121 \\
171
\end{array}\right]\right\} \quad\left[\begin{array}{l}
222 \\
222 \\
185
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
222 \\
222 \\
218
\end{array}\right]\right\} \quad\left[\begin{array}{l}
240 \\
171 \\
103
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
240 \\
171 \\
160
\end{array}\right]\right\} \quad\left[\begin{array}{l}
236 \\
227 \\
106
\end{array}\right]=\Phi\left\{\left[\begin{array}{l}
240 \\
230 \\
166
\end{array}\right]\right\}
\]

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Transformations}


To undo the process we need to find, \(\Phi^{-1}\), that maps:
\[
\left[\begin{array}{c}
17 \\
121 \\
171
\end{array}\right]=\Phi^{-1}\left\{\left[\begin{array}{c}
17 \\
122 \\
114
\end{array}\right]\right\} \quad\left[\begin{array}{l}
222 \\
222 \\
218
\end{array}\right]=\Phi^{-1}\left\{\left[\begin{array}{l}
222 \\
222 \\
185
\end{array}\right]\right\} \quad\left[\begin{array}{l}
240 \\
171 \\
160
\end{array}\right]=\Phi^{-1}\left\{\left[\begin{array}{l}
240 \\
171 \\
103
\end{array}\right]\right\} \quad\left[\begin{array}{l}
240 \\
230 \\
166
\end{array}\right]=\Phi^{-1}\left\{\left[\begin{array}{l}
236 \\
227 \\
106
\end{array}\right]\right\}
\]

\section*{Correction Using 3 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{-1}
\]


\section*{Correction Using 3 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{-1}
\]

\[
\mathbf{X}=\left[\begin{array}{lll}
222 & 17 & 240 \\
222 & 122 & 171 \\
185 & 114 & 103
\end{array}\right]
\]
\[
\mathbf{Y}=\left[\begin{array}{lll}
222 & 17 & 240 \\
222 & 121 & 171 \\
218 & 171 & 160
\end{array}\right]
\]

EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Another Correction Using 3 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{-1}
\]


EECE 253 Image Processing
Vanderbilt University School of Engineering

\section*{Another Correction Using 3 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{-1}
\]

\[
\mathbf{X}=\left[\begin{array}{lll}
222 & 17 & 236 \\
222 & 122 & 227 \\
185 & 114 & 106
\end{array}\right]
\]


\section*{Correction Using All 4 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}
\]

\[
\mathbf{X}=\left[\begin{array}{llll}
222 & 17 & 236 & 240 \\
222 & 122 & 227 & 171 \\
185 & 114 & 106 & 103
\end{array}\right]
\]

\[
\mathbf{Y}=\left[\begin{array}{llll}
222 & 17 & 240 & 240 \\
222 & 121 & 230 & 171 \\
218 & 171 & 166 & 160
\end{array}\right]
\]

\section*{Correction Using All 4 Mappings}
\[
\mathbf{B}=\mathbf{A}^{-1}=\mathbf{Y} \mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}
\]

\[
\mathbf{X}=\left[\begin{array}{cccc}
222 & 17 & 236 & 240 \\
222 & 122 & 227 & 171 \\
185 & 114 & 106 & 103
\end{array}\right]
\]

\[
\mathbf{Y}=\left[\begin{array}{cccc}
222 & 17 & 240 & 240 \\
222 & 121 & 230 & 171 \\
218 & 171 & 166 & 160
\end{array}\right]
\]

\section*{Linear Color Transformation Program}
```

function J = LinTrans(I,A)
[R C B] = size(I);
I = double(I);
J = reshape(((A*(reshape(I, R*C,3))')'),R,C,3);
J = uint8(J);
return;

```

\title{
EECE\CS 253 Image Processing
}

Lecture Notes: The 1\&2-Dimensional Fourier Transforms

Richard Alan Peters II
Department of Electrical Engineering and Computer Science

Fall Semester 2012

\section*{Fact: Any Real Signal has Odd-order harmonics a Frequency-Domain Representation \\ \[
\mathrm{sq}(t)=\sum_{n=-\infty}^{\infty} \frac{1}{2 n+1} \sin \left[\frac{2 \pi}{\lambda}(2 n+1) t\right]
\]}

The modes shown (blue) sum to the rippling square wave (black).

As the number of modes in the sum becomes large, it approaches a square wave (red).


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequency-Domain Representation}

Any periodic signal can be described by a sum of sinusoids.
\[
\operatorname{sq}(t)=\sum_{n=-\infty}^{\infty} \frac{1}{2 n+1} \sin \left[\frac{2 \pi}{\lambda}(2 n+1) t\right]
\]

The sinusoids are called "basis functions".

The multipliers are called "Fourier coefficients".


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequency-Domain Representation}

Any periodic signal can be described by a sum of sinusoids.
\[
\operatorname{sq}(t)=\sum_{n=-\infty}^{\infty} \frac{1}{2 n+1} \sin \left[\frac{2 \pi}{\lambda}(2 n+1) t\right]
\]

The sinusoids are called "basis functions".

The multipliers are called "Fourier coefficients".

\section*{Basis \\ functions}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequency-Domain Representation}

Any periodic signal can be described by a sum of sinusoids.
\[
\operatorname{sq}(t)=\sum_{n=-\infty}^{\infty} \frac{1}{2 n+1} \sin \left[\frac{2 \pi}{\lambda}(2 n+1) t\right]
\]

The sinusoids are called "basis functions".

The multipliers are called "Fourier coefficients".

The Fourier coefficients (of a square
 wave).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Partial Sums of a Square Wave}

The limit of the given sequence of partial sums \({ }^{1}\) is exactly a square wave






\({ }^{1}\) the limit as \(n\) approaches infinity of the sum of \(n\) sines.

\section*{Anatomy of a Sinusoid}

\[
f(t)=A \sin \left(\frac{2 \pi}{\lambda} t-\phi\right)
\]
\(1 / \lambda\) is the frequency of the sinusoid \((\mathrm{Hz})\).
\(2 \pi / \lambda\) is the angular frequency (radians/s).

\section*{The Inner Product: a Measure of Similarity}

The similarity between functions \(f\) and \(g\) on the interval \((-\lambda / 2, \lambda / 2)\) can be defined by
\(\langle f, g\rangle=\int_{-\lambda / 2}^{\lambda / 2} f(t) g^{*}(t) d t\)
where \(g^{*}(t)\) is the complex conjugate of \(g(t)\).
This number, called the inner product of \(f\) and \(g\), can also be thought of as the amount of \(g\) in \(f\) or as the projection of \(f\) onto \(g\).

If \(f\) and \(g\) have the same energy, then their inner product is maximal if \(f=g\). On the other hand if \(\langle f, g\rangle=0\), then \(f\) and \(g\) have nothing in common.

\section*{Inner Products}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering


\section*{Inner Products}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering
(

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inner Product of a Periodic Function and a Sinusoid}
\[
\begin{aligned}
\langle f, g\rangle & =\int_{-\lambda / 2}^{\lambda / 2} f(t) \sin \left(\frac{2 \pi}{\lambda} t\right) d t\langle f, g\rangle=\int_{-\lambda / 2}^{\lambda / 2} f(t) \cos \left(\frac{2 \pi}{\lambda} t\right) d t \\
\langle f, g\rangle & =\int_{-\lambda / 2}^{\lambda / 2} f(t)\left[\cos \left(\frac{2 \pi}{\lambda} t\right)-i \sin \left(\frac{2 \pi}{\lambda} t\right)\right] d t \quad \begin{array}{l}
3 \text { different } \\
\text { representations }
\end{array} \\
& =\int_{-\lambda / 2}^{\lambda / 2} f(t) e^{-i \frac{2 \pi}{\lambda} t} d t \\
& =e_{-\lambda / 2}^{\lambda / 2} f(t) e^{-i \omega t} d t \quad \omega=\frac{2 \pi}{\lambda}
\end{aligned}
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inner Product of a Periodic Function and a Sinusoid}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inner Product of a Periodic Function and a Sinusoid}

\[
\langle f, g\rangle=\int_{\substack{-\lambda / 2 \\ \lambda / 2}}^{\lambda / 2} f(t)\left[\cos \left(\frac{2 \pi}{\lambda} t\right)-i \sin \left(\frac{2 \pi}{\lambda} t\right)\right] d t
\]
\[
=\int^{\lambda / 2} f(t) e^{-i \frac{2 \pi}{\lambda} t} d t \quad \begin{aligned}
& \text { Complex number result } \\
& \text { vields the amblitude ano }
\end{aligned}
\] yields the amplitude and phase of that sinusoid in the function.

\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{The Fourier Series}
is the decomposition of a \(\lambda\)-periodic signal into a sum of sinusoids.
\[
f(t)=A_{0}+\sum_{n=1}^{\infty} A_{n} \cos \left(\frac{2 \pi n}{\lambda} t\right)+B_{n} \sin \left(\frac{2 \pi n}{\lambda} t\right)
\]

\section*{periodic: \(\exists \lambda \in \mathbb{R}\) such that \(f(t \pm n \lambda)=f(t)\)}
\[
\begin{aligned}
& A_{n}=\frac{2}{\lambda} \int_{-\lambda / 2}^{\lambda / 2} f(t)\left[\cos \left(\frac{2 \pi n}{\lambda} t-\varphi_{n}\right)\right] d t \text { for } n \geq 0 \\
& B_{n}=\frac{2}{\lambda} \int_{-\lambda / 2}^{\lambda / 2} f(t)\left[\sin \left(\frac{2 \pi n}{\lambda} t-\varphi_{n}\right)\right] d t \text { for } n \geq 0
\end{aligned}
\]

The representation of a function by its Fourier Series is the sum of sinusoidal "basis functions" multiplied by coefficients.

Fourier coefficients are generated by taking the inner product of the function with the basis.

The basis functions correspond to modes of vibration.

\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{The Fourier Series}
can also be written in terms of complex exponentials
\[
\begin{aligned}
f(t) & =\sum_{n=-\infty}^{\infty} C_{n} e^{+i \frac{2 \pi n}{\lambda} t}=\sum_{n=-\infty}^{\infty}\left|C_{n}\right| e^{+i\left(\frac{2 \pi n}{\lambda} t+\phi_{n}\right)} \\
& =\sum_{n=-\infty}^{\infty}\left|C_{n}\right| \cos \left(\frac{2 \pi n}{\lambda} t+\phi_{n}\right)+\left|C_{n}\right| \sin \left(\frac{2 \pi n}{\lambda} t+\phi_{n}\right)
\end{aligned}
\]
\[
i=\sqrt{-1}
\]
\[
C_{n}=\left|C_{n}\right| e^{+i \phi_{n}}
\]
\[
C_{n}=\left|C_{n}\right| e^{+i \phi_{n}}=\frac{1}{\lambda} \int_{-\lambda / 2}^{\lambda / 2} f(t) e^{-i \frac{2 \pi n}{\lambda} t} d t
\]
\[
=\frac{1}{\lambda} \int_{-\lambda / 2}^{\lambda / 2} f(t)\left[\cos \left(\frac{2 \pi n}{\lambda} t+\phi_{n}\right)+\left|C_{n}\right| \sin \left(\frac{2 \pi n}{\lambda} t+\phi_{n}\right)\right] d t
\]
\[
f(t+n \lambda)=f(t)
\]
\[
\text { for all integers } n
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Why are Fourier Coefficients Complex Numbers?}
\[
f(t)=\sum_{n=-\infty}^{\infty} C_{n} e^{+i \frac{2 \pi n}{\lambda} t} \text { where } C_{n}=\left|C_{n}\right| e^{+i \phi_{n}} .
\]
\(C_{n}\) represents the amplitude, \(A=\left|C_{n}\right|\), and relative phase, \(\phi\), of that part of the original signal, \(f(t)\), that is a sinusoid of frequency \(\omega_{n}=2 \pi n / \lambda\).


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{The Fourier Transform}
is the decomposition of a nonperiodic signal into a continuous sum \({ }^{*}\) of sinusoids.
\[
\begin{aligned}
F(\omega) & =|F(\omega)| e^{i \Phi(\omega)}=\int_{-\infty}^{\infty} f(t) e^{i 2 \pi \omega t} d t \\
& =\int_{-\infty}^{\infty} f(t)[\cos (2 \pi \omega t)+i \sin (2 \pi \omega t)] d t \\
f(t) & =\int_{-\infty}^{\infty} F(\omega) e^{-i 2 \pi \omega t} d \omega=\int_{-\infty}^{\infty}|F(\omega)| e^{-i(2 \pi \omega t+\Phi(\omega))} d \omega \\
& =\int_{-\infty}^{\infty} F(\omega)[\cos (2 \pi \omega t)-i \sin (2 \pi \omega t)] d \omega \\
& =\int_{-\infty}^{\infty}|F(\omega)|[\cos (2 \pi \omega t+\Phi(\omega))-i \sin (2 \pi \omega t+\Phi(\omega))] d \omega
\end{aligned}
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Mammals Use the FT in Hearing}


\title{
EECE/CS 253 Image Processing
}

\section*{The Discrete Fourier Transform}

A discrete signal, \(\left\{h_{k} \mid k=0,1,2, \ldots, N-1\right\}\), of finite length \(N\) can be repre sented as a weighted sum of \(N\) sinusoids, \(\left\{e^{-i 2 \pi k n / N} \mid n=0,1,2, \ldots, N-1\right\}\) through
\[
h_{k}=\sum_{n=0}^{N-1} H_{n} e^{-i 2 \pi k n / N}
\]
where the set, \(\left\{H_{n} \mid n=0,1,2, \ldots, N-1\right\}\), are the Fourier coefficients defined as the projection of the original signal onto sinusoid, \(n\), given by :
\[
H_{n}=\frac{1}{N} \sum_{k=0}^{N-1} h_{k} e^{+i 2 \pi k n / N}
\]

\section*{The Two-Dimensional Fourier Transform}

\section*{Primary Uses of the FT in Image Processing:}
- Explains why down-sampling can add distortion to an image and shows how to avoid it.
। Useful for certain types of noise reduction, deblurring, and other types of image restoration.
- For feature detection and enhancement, especially edge detection.

\section*{The Fourier Transform: Discussion}

The expressions
\[
F(\omega)=\int_{-\infty}^{\infty} f(t) e^{-i 2 \pi \omega t} d t=\left\langle f(t), e^{+i 2 \pi \omega t}\right\rangle
\]
continuous signals defined over all real numbers
and
\[
H_{n}=\frac{1}{N} \sum_{n=0}^{N-1} h_{k} e^{-i 2 \pi k n / N}=\left\langle h_{k}, e^{+i 2 \pi k n / N}\right\rangle
\]
discrete signals with \(N\) terms or samples.
for the Fourier coefficients are "inner products" which can be thought of as measures of the similarity between the functions
\(f(t)\) and \(e^{+i 2 \pi \omega t}\) for \(t \in(-\infty, \infty)\) or between the sequences
\(\left\{h_{k}\right\}_{k=0}^{N-1}\) and \(\left\{e^{+i 2 \pi k n / N}\right\}_{k=0}^{N-1}\).

The Fourier Transform:
EECE/CS 253 Image Processing
Discussion (cont'd.)
In the context of inner products, the complex exponentials
\[
\left\{e^{-i 2 \pi \omega t} \mid \omega \in \Re \text { and } \omega \in(-\infty, \infty)\right\} \text { and }\left\{e^{-i 2 \pi k n / N} \mid \ldots,-2,-1,0,1,2, \ldots\right\}
\]
are called "orthogonal sets" since they have the property:
\[
\begin{aligned}
& \left\langle e^{-i 2 \pi \omega_{1} t}, e^{-i 2 \pi \omega_{2} t}\right\rangle=\int_{-\infty}^{\infty} e^{-i 2 \pi \omega_{1} t} \cdot e^{+i 2 \pi \omega_{2} t} d t=\left\{\begin{array}{l}
\infty, \text { if } \omega_{1}=\omega_{2} \\
0, \text { if } \omega_{1} \neq \omega_{2}
\end{array}\right. \\
& \left\langle e^{-i 2 \pi j n / N}, e^{-i 2 \pi k n / N}\right\rangle=\sum_{n=0}^{N-1} e^{-i 2 \pi j n / N} \cdot e^{+i 2 \pi k n / N}=\left\{\begin{array}{l}
c, \text { if } j=k \\
0, \text { if } j \neq k
\end{array}\right.
\end{aligned}
\]

The function sets are called "orthogonal basis sets"

They are called "basis sets" since for any function \({ }^{1}, f(t)\), of a real variable there exists a complex-valued function \(F(\mathrm{w})\), and for any sequence \({ }^{1}\), \(h_{k}\), there exist complex numbers, \(H_{n}\), such that
\[
f(t)=\int_{-\infty}^{\infty} F(\omega) e^{-i 2 \pi \omega t} d \omega \text { and } h_{k}=\sum_{n=0}^{N-1} H_{n} e^{-i 2 \pi k n / N}
\]
\({ }^{1}\) with finite energy.

\section*{The Fourier Transform: Discussion (cont’d.)}

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

Consider the 2-dimensional functions
\[
\left\{e^{-i 2 \pi(u x+v y)} \mid u, v, x, y \in \Re\right\} \text { and }\left\{\left.e^{-i 2 \pi\left(\frac{j m}{M}+\frac{l m}{N}\right)} \right\rvert\, j, m \in 0, \ldots, M-1, k, n \in 0, \ldots, N-1\right\}
\]

These are, likewise, orthogonal:
\[
\begin{aligned}
&\left\langle e^{-i 2 \pi\left(u_{1} x+v_{1} y\right)}, e^{-i 2 \pi\left(u_{2} x+v_{2} y\right)}\right\rangle=\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i 2 \pi\left(u_{1} x+v_{1} y\right)} \cdot e^{+i 2 \pi\left(u_{2} x+v_{2} y\right)} d x d y \\
&=\left\{\begin{array}{l}
\infty, \text { if } u_{1}=u_{2} \text { and } v_{1}=v_{2} \\
0, \text { otherwise }^{\infty}
\end{array}\right. \\
&\left\langle e^{-i 2 \pi\left(\frac{j_{1} m}{M}+\frac{k_{1} n}{N}\right)}, e^{-i 2 \pi\left(\frac{j_{2} m}{M}+\frac{k_{2} n}{N}\right)}\right\rangle=\sum_{m=0}^{M-1 N-1} \sum_{n=0}^{-i 2 \pi\left(\frac{j_{1} m}{M}+\frac{k_{1} n}{N}\right)} \cdot e^{+i 2 \pi\left(\frac{j_{2} m}{M}+\frac{k_{2} n}{N}\right)} \\
&=\left\{\begin{array}{l}
c, \text { if } j_{1}=j_{2} \text { and } k_{1}=k_{2} \\
0,
\end{array}\right. \\
& \text { otherwise }
\end{aligned}
\]

The Fourier Transform:
EECE/CS 253 Image Processing
Discussion (cont’d.)

\author{
Vanderbilt University School of Engineering
}

Therefore
\(\left\{e^{-i 2 \pi(u x+v y)} \mid u, v, x, y \in \mathbb{R}\right\}\) and \(\left\{\left.e^{-i 2 \pi\left(\frac{j m}{m}+\frac{l n}{N}\right)} \right\rvert\, j, k, m, n, M \in \mathbb{Z}\right\}\)
are orthogonal basis sets. This suggests that function \(f(x, y)\) defined on the real plane, and sequence \(\left\{\left\{h_{m n}\right\}\right\}\) for integers \(m\) and \(n\) have analogous Fourier representations,
\[
f(x, y)=\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{+i 2 \pi(u x+v y)} d u d v \text { and } h_{m n}=\sum_{j=0}^{M-1 N-1} \sum_{k=0} H_{j k} e^{+i 2 \pi\left(\frac{j m}{M}+\frac{k n}{N}\right)} .
\]
where the Fourier coefficients are given by
\[
F(u, v)=\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i 2 \pi(u x+v y)} d x d y \text { and } H_{j k}=\sum_{m=0}^{M-1 N-1} \sum_{n=0} h_{m n} e^{-i 2 \pi\left(\frac{j m}{M}+\frac{k n}{N}\right)} .
\]
(True for finite energy functions \(f(x, y)\) and \(\left\{\left\{h_{m n}\right\}\right\}\).)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Continuous Fourier Transform}


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Discrete Fourier Transform}

The discrete Fourier transform assumes a digital image exists on a closed surface, a torus.


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Discrete Fourier Transform}

The discrete Fourier transform assumes a digital image exists on a closed surface, a torus.

EECE/CS 253 Image Processing

\section*{The 2D Fourier Transform of a Digital Image}

Let \(\mathbf{I}(r, c)\) be a single-band (intensity) digital image with \(R\) rows and C columns. Then, \(\mathbf{I}(r, c)\) has Fourier representation
\[
\mathbf{I}(r, c)=\sum_{u=0}^{R-1} \sum_{v=0}^{C-1} \mathfrak{G}(v, u) e^{+i 2 \pi\left(\frac{v r}{R}+\frac{u c}{C}\right)},
\]
where
these complex exponentials are 2D sinusoids.
\(\mathscr{S}(v, u)=\frac{1}{R C} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \mathbf{I}(r, c) e^{-i 2 \pi\left(\frac{v r}{R}+\frac{u c}{C}\right)}\)
are the \(R \times C\) Fourier coefficients.

\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{What are 2D sinusoids?}

To simplify the situation assume \(R=C=N\). Then
\[
e^{ \pm i 2 \pi\left(\frac{v r}{R}+\frac{u c}{C}\right)}=e^{ \pm i \frac{2 \pi}{N}(v r+u c)}=e^{ \pm i \frac{2 \pi \omega}{N}(r \sin \theta+c \cos \theta)},
\]
where
\[
v=\omega \sin \theta, \quad u=\omega \cos \theta, \quad \omega=\sqrt{v^{2}+u^{2}}, \quad \text { and } \quad \theta=\tan ^{-1}\left(\frac{v}{u}\right) .
\]

Write
\[
\lambda=\frac{N}{\omega},
\]

Note: since images are indexed by row \& col with \(r\) down and \(c\) to the right, \(\theta\) is positive in the clockwise direction.
Then by Euler's relation,
\[
e^{ \pm i 2 \pi \frac{1}{\lambda}(r \sin \theta+c \cos \theta)}=\cos \left[\frac{2 \pi}{\lambda}(r \sin \theta+c \cos \theta)\right] \pm i \sin \left[\frac{2 \pi}{\lambda}(r \sin \theta+c \cos \theta)\right]
\]

Cont'd. on next page.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{What are 2D sinusoids? (cont'd.)}

Both the real part of this,
\[
\operatorname{Re}\left\{e^{ \pm i 2 \pi \frac{1}{\lambda}(r \sin \theta+c \cos \theta)}\right\}=+\cos \left[\frac{2 \pi}{\lambda}(r \sin \theta+c \cos \theta)\right]
\]
and the imaginary part,
\[
\operatorname{Im}\left\{e^{ \pm i 2 \pi \frac{1}{\lambda}(r \sin \theta+c \cos \theta)}\right\}= \pm \sin \left[\frac{2 \pi}{\lambda}(r \sin \theta+c \cos \theta)\right]
\]
are sinusoidal "gratings" of unit amplitude, period \(\lambda\) and direction \(\theta\).
Then \(\frac{2 \pi \omega}{N}\) is the radian frequency, and \(\frac{\omega}{N}\) the frequency, of the wavefront and \(\lambda=\frac{N}{\omega}\) is the wavelength in pixels in the wavefront direction.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{2D Sinusoids:}
\[
\mathbf{I}(r, c)=\frac{A}{2}\left\{\cos \left[\frac{2 \pi}{\lambda}(r \cdot \sin \theta+c \cdot \cos \theta)+\varphi\right]+1\right\}
\]
... are plane waves with grayscale amplitudes, periods in terms of lengths, ...


\section*{2D Sinusoids:}

\section*{... specific orientations, and phase shifts.}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Fourier Transform of an Image}


I

\(\operatorname{Re}[\mathscr{J}\{\mathbf{I}\}]\)

\(\operatorname{Im}[\mathscr{J}\{\mathbf{I}\}]\)

\section*{Points on the Fourier Plane}

If \(R=C=N\) the point at column freq. \(u\) and row freq. v represents a sinusoid with freq. \(w\) and orientation \(\theta\).

If \(R \neq C\) then \(\omega=1 / \wedge\) where \(\wedge\) is the length of vector ( \(C / u, R / v\) ) and the wavefront orientation is \(\theta=\tan ^{-1}[(v / R) /(u / C)]\).


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Points on the Fourier Plane (of a Digital Image)}

In the Fourier transform of an \(R \times C\) digital image, positions \(u\) and \(v\) indicate the number of repetitions
\[
-v \text { direction }
\] of the sinusoid in those directions. Therefore the wavelengths along the column and row axes are
\[
\lambda_{u}=\frac{C}{u} \quad \text { and } \lambda_{v}=\frac{R}{v} \text { pixels, }
\]
and the wavelength in the wavefront direction is
\[
\lambda_{\mathrm{wf}}=R C\left[(u R)^{2}+(v C)^{2}\right]^{-\frac{1}{2}} .
\]

The frequency is the fraction of the sinusoid traversed over one pixel,
\[
\begin{aligned}
& \omega_{u}=\frac{u}{C}, \omega_{v}=\frac{v}{R}, \text { and } \\
& \omega_{\mathrm{wf}}=\frac{1}{R C} \sqrt{(u R)^{2}+(v C)^{2}} \text { cycles. }
\end{aligned}
\]

The wavefront direction is given by
\[
\theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{\omega_{v}}{\omega_{u}}\right)=\tan ^{-1}\left(\frac{v C}{u R}\right) . \quad \frac{\text { row freq. }}{\text { column freq. }}
\]

More about this later (pp. 66-87).


Note that the wave front direction \(=\theta\) only if \(R=C\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Points on the Fourier Plane \\ Note that \(\theta\) is the wavefront direction only if \(R=C\).}


This point represents this particular sinusoidal grating

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Value of a Fourier Coefficient ...}

... is a complex number with a real part and an imaginary part.

If you represent that number as a magnitude, \(A\), and a phase, \(\phi, \ldots\)
..these represent the amplitude and offset of the sinusoid with frequency \(\omega\) and direction \(\theta\).*

\section*{The Value of a Fourier Coefficient}


The magnitude and phase representation makes more sense physically...
...since the Fourier magnitude, \(\mathbf{A}(\omega, \theta)\), at point ( \(\omega, \theta\) ) represents the amplitude of the sinusoid...
and the phase, \(\phi(\omega, \theta)\), represents the offset of the sinusoid relative to origin.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Fourier Coefficient at \((u, v)\)}


So, the point ( \(u, v\) ) on the Fourier plane...
...represents a sinusoidal grating of frequency \(\omega\) and orientation \(\theta\).*

The complex value, \(F(u, v)\), of the FT at point \((u, v)\)...
...represents the amplitude, \(A\), and the phase offset, \(\phi\), of the sinusoid.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Sinusoid from the Fourier Coeff. at (u,v)}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{FT of an Image (Magnitude + Phase)}


I

\(\log \left\{|\mathscr{F}\{\mathbf{I}\}|^{2}+1\right\}\)
\(\angle[\mathscr{f}\{\mathbf{I}\}]\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{FT of an Image (Real + Imaginary)}


I

\(\operatorname{Re}[\mathscr{J}\{\mathbf{I}\}]\)

\(\operatorname{Im}[\mathscr{J}\{\mathbf{I}\}]\)

\section*{The Power Spectrum}

The power spectrum of a signal is the square of the magnitude of its Fourier Transform.

For display, the log of the power spectrum is often used.
\[
\begin{aligned}
|\mathscr{G}(u, v)|^{2} & =\mathscr{G}(u, v) \mathscr{G}^{*}(u, v) \\
& =[\operatorname{Re} \mathscr{G}(u, v)+i \operatorname{Im} \mathfrak{G}(u, v)][\operatorname{Re} \mathfrak{G}(u, v)-i \operatorname{Im} \mathfrak{G}(u, v)] \\
& =[\operatorname{Re} \mathfrak{G}(u, v)]^{2}+[\operatorname{Im} \mathscr{G}(u, v)]^{2} .
\end{aligned}
\]

At each location \((u, v)\) it indicates the squared intensity of the frequency component with period \(\lambda=1 / \sqrt{u^{2}+v^{2}}\) and orientation \(\theta=\tan ^{-1}(v / u)\).
```

For display in Matlab:
PS = fftshift(2*log(abs(fft2(I))+1));

```

\section*{On the Computation of the Power Spectrum}

The power spectrum (PS) is defined by \(\operatorname{PS}(\mathrm{I})=|\mathscr{F}\{\mathrm{I}(u, v)\}|^{2}\).
We take the base-e logarithm of the PS in order to view it. Otherwise its dynamic range could be too large to see everything at once. We add 1 to it first so that the minimum value of the result is 0 rather than -infinity, which it would be if there were any zeros in the PS. Recall that \(\log \left(f^{2}\right)=2 \log (f)\).
Multiplying by 2 is not necessary if you are generating a PS for viewing, since you'll probably have to scale it into the range 0-255 anyway. It is much easier to see the structures in a Fourier plane if the origin is in the center. Therefore we usually perform an fftshift on the PS before it is displayed.
```

>> PS = fftshift(log(abs(fft2(I))+1));
>> M = max(PS(:));
>> image(uint8(255*(PS/M)));

```

If the PS is being calculated for later computational use -- for example the autocorrelation of a function is the inverse FT of the PS of the function -- it should be calculated by
```

>> PS = abs(fft2(I)).^2;

```

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Uncertainty Relation}


If \(\Delta x \Delta y\) is the extent of the object in space and if \(\Delta u \Delta v\) is its extent in frequency then,
\[
\Delta x \Delta y \cdot \Delta u \Delta v \geq \frac{1}{16 \pi^{2}}
\]

A small object in space has a large frequency extent and vice-versa.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Uncertainty Relation}

\(\rightarrow\) small extent \(\leftarrow\)

\(\leftarrow\) large extent \(\rightarrow\)


Recall: a symmetric pair of impulses in the frequency domain becomes a sinusoid in the spatial domain.

A symmetric pair of lines in the frequency domain becomes a sinusoidal line in the spatial domain.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Fourier Transform of an Edge}



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Fourier Transform of a Bar}



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Coordinate Origin of the FFT}
```

Center =
(floor(R/2)+1, floor(C/2)+1)

```


Even


Odd



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Matlab’s fftshift and ifftshift}

I = ifftshift(J):
J = fftshift(I):
from FFT2 or ifftshift

\[
\mathbf{I}(1,1) \rightarrow \mathbf{J}(\lfloor R / 2\rfloor+1,\lfloor C / 2\rfloor+1)
\]
\(\mathbf{J}(\lfloor R / 2\rfloor+1,\lfloor C / 2\rfloor+1) \rightarrow \mathbf{I}(1,1)\)
where \(\lfloor x\rfloor=\mathrm{floor}(\mathrm{x})=\) the largest integer smaller than \(x\).

\section*{Matlab’s fftshift and ifftshift}
\[
\begin{aligned}
\mathbf{J} & =\text { fftshift(I) : } \\
& \mathbf{I}(1,1) \rightarrow \mathbf{J}(\lfloor R / 2\rfloor+1,\lfloor C / 2\rfloor+1)
\end{aligned}
\]

\[
\begin{aligned}
& \mathbf{I}=\text { ifftshift }(\mathrm{J}): \\
& \\
& \mathbf{J}(\lfloor R / 2\rfloor+1,\lfloor C / 2\rfloor+1) \rightarrow \mathbf{I}(1,1)
\end{aligned}
\]

where \(\lfloor x\rfloor=\) floor \((x)=\) the largest integer smaller than \(x\).

\section*{Points on the Fourier Plane (of a Digital Image)}

In the Fourier transform of an \(R \times C\) digital image, positions \(u\) and \(v\) indicate the number of repetitions of the sinusoid in those directions. Therefore the wavelengths along the column and row axes are
\[
\lambda_{u}=\frac{C}{u} \text { and } \lambda_{v}=\frac{R}{v} \text { pixels, }
\]
and the wavelength in the wavefront direction is
\[
\lambda_{\mathrm{wf}}=R C\left[(u R)^{2}+(v C)^{2}\right]^{-\frac{1}{2}} .
\]

The frequency is the fraction of the sinusoid traversed over one pixel,
\[
\begin{aligned}
& \omega_{u}=\frac{u}{C}, \omega_{v}=\frac{v}{R}, \text { and } \\
& \omega_{\mathrm{wf}}=\frac{1}{R C} \sqrt{(u R)^{2}+(v C)^{2}} \text { cycles. }
\end{aligned}
\]

The wavefront direction is given by
\[
\theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{\omega_{v}}{\omega_{u}}\right)=\tan ^{-1}\left(\frac{v C}{u R}\right) . \quad \frac{\text { row freq. }}{\text { column freq. }}
\]


Note that the wave front direction \(=\theta\) only if \(R=C\).

\section*{Geometrical Derivation of Wavelength}

Since the wavelength of a horizontal \({ }^{*}\) wave is \(R / v\) and that of a vertical is \(C / u\), the line segment, \(h\), that connects the two distances is parallel to the wavefront. The wavelength is the "altitude" of the triangle w.r.t. \(h\) (the perpendicular to \(h\) that intersects the origin). The area of the triangle, one half of base times height, is independent of the leg that is taken to be the base. Equate the expression with base \(C / u\) to that with base \(h\), to find \(\lambda\) w.r.t \(R, C, v, u, \& h\). Then replace \(h\) with its expression as a function of \(R, C, v, \& u\) to get the final expression.


\footnotetext{
*The equivalue lines are horizontal in a wave with a vertical wave front and vice versa.
}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Coordinates and Directions in the Fourier Plane}


Since rows increase down and columns to the right, slopes and angles are opposite those of a right-handed coordinate system.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inverse FFTs of Impulses}
"horizontal" is the wavefront direction.

highest-possible-frequency horizontal sinusoid ( \(C\) is even)

\section*{Inverse FFTs of Impulses}
"vertical" is the wavefront direction.

highest-possible-frequency vertical sinusoid ( \(R\) is even)

\section*{Inverse FFTs of Impulses}
a checker-board pattern.

highest-possible-freq horizontal+vertical sinusoid ( \(R\) \& \(C\) even)

\section*{Inverse FFTs of Impulses}
"horizontal" is the wavefront direction.

lowest-possible-frequency horizontal sinusoid

\section*{Inverse FFTs of Impulses}

> "vertical" is the wavefront direction.

lowest-possible-frequency vertical sinusoid

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inverse FFTs of Impulses}
"negative diagonal" is the wavefront direction.


lowest-possible-frequency negative diagonal sinusoid

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Inverse FFTs of Impulses}
"positive diagonal" is the wavefront direction.


lowest-possible-frequency positive diagonal sinusoid

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(4,3)\); wavelengths: \(\left(\lambda_{u}, \lambda_{v}\right)=(128,128)\)
How can that be?
(c) 1999-2013 by Richard Alan Peters II

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(1,0)\); wavelength: \(\lambda_{\mathrm{u}}=512\)

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(0,1)\); wavelength: \(\lambda_{\mathrm{v}}=384\)

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(2,0)\); wavelength: \(\lambda_{u}=256\)

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(0,2) ;\) wavelength: \(\lambda_{\mathrm{v}}=192\)

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(3,0)\); wavelength: \(\lambda_{u}=170^{2 / 3}\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(0,3) ;\) wavelength: \(\lambda_{\mathrm{v}}=128\)

In the Fourier plane of a square image, the orientation of the line through the point pair = the orientation of the \(F\) wave front in the image. Not so for a non-square image.

In the \(F\) plane the angle is \(-45^{\circ}\) essing in this image it's about \(-53^{\circ}\) ineering (yellow line). That's because the fraction of \(R\) covered by eng 1 one pixel is \(4 / 3\) the fraction of \(C\) covered by one pixel.


512 columns


Also as a result, the wavelength is 102.4.
frequencies: \((u, v)=(3,3)\); wavelengths: \(\left(\lambda_{u}, \lambda_{v}\right)=(1702 / 3,128)\)

In general the slope of the wavefront direction in the image is given by \((v / R) /(u / C)\). Therefore its angle is
Fr
\[
\theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{v C}{u R}\right)
\]

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{ngths in the Fourier Plane}

512 columns

and the wavelength is:
frequencies: \((u, v)=(3,3)\); wavelengt \(] \lambda_{\mathrm{wf}}=R C\left[(u R)^{2}+(v C)^{2}\right]^{-\frac{1}{2}}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(3,3)\); wavelengths: \(\left(\lambda_{u}, \lambda_{v}\right)=(1702 / 3,128)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Frequencies and Wavelengths in the Fourier Plane}

frequencies: \((u, v)=(4,3)\); wavelengths: \(\left(\lambda_{u}, \lambda_{v}\right)=(128,128)\)

EECE/CS 253 Image Processing
r.........in サーiversity School of Engineering

The ratio \(R / C=\frac{3}{4}\) in this image. Therefore at
frequency \((4,3)\) the wave front angle is
Fr \(\theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{3 \cdot 512}{4 \cdot 384}\right)=\tan ^{-1}\left(\frac{3 \cdot 4}{4 \cdot 3}\right)=\tan ^{-1}(1)=45^{\circ}\), ourier Plane


512 columns


\section*{and the wavelength is}
frequencies: \((u, v)=(4,:\)
\[
\lambda_{\mathrm{wf}}=384 \cdot 512\left[(3 \cdot 384)^{2}+(4 \cdot 512)^{2}\right]^{-\frac{1}{2}} \approx 83.67
\]

\author{
Vanderbilt University School of Engineering
}

\section*{Power Spectrum of an Image}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Relationship between Image and FT}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Fourier Magnitude and Phase}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Fourier Magnitude}
\[
\log |\mathscr{F}\{\mathbf{I}\}|
\]

\section*{Fourier Phase}
\(\angle \mathscr{F}\{\mathbf{I}\}\)


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering
Q: Which contains more visually relevant information; magnitude or phase?

original image


Fourier log magnitude


Fourier phase

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Magnitude Only Reconstruction}


\section*{Phase Only Reconstruction}


\title{
EECE\CS 253 Image Processing
}

\author{
Lecture Notes: Spatial Convolution
}

\section*{Richard Alan Peters II}

Department of Electrical Engineering and Computer Science

Fall Semester 2011

\section*{Spatial Filtering}

Let \(\mathbf{I}\) and \(\mathbf{J}\) be images such that \(\mathbf{J}=\mathrm{T}[\mathbf{I}]\).
\(\mathrm{T}[\cdot]\) represents a transformation, such that,
\(\mathbf{J}(r, c)=\mathrm{T}[\mathbf{I}](r, c)=\)
\(f(\{\mathbf{I}(\rho, \chi) \mid \rho \in\{r-s, \ldots, r, \ldots r+s\}, \chi \in\{c-d, \ldots, c, \ldots c+d\}\})\).
That is, the value of the transformed image, \(\mathbf{J}\), at pixel location \((r, c)\) is a function of the values of the original image, \(\mathbf{I}\), in a \(2 s+1\) \(\times 2 d+1\) rectangular neighborhood centered on pixel location \((r, c)\).

\section*{Moving Windows}

। The value, \(\mathbf{J}(r, c)=T[\mathbf{I}](r, c)\), is a function of a rectangular neighborhood centered on pixel location ( \(r, c\) ) in \(\mathbf{I}\).
There is a different neighborhood for each pixel location, but if the dimensions of the neighborhood are the same for each location, then transform T is sometimes called a moving window transform.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}

photo: R.A.Peters II, 1999

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}


Pixelize the section to better see the effects.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}
 better see the effects.

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}

\section*{lets get some perspective on this}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}


\section*{Moving-Window Transformations}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

Linear Moving-Window Transformations
( i.e. convolution)

The output of the transform at each pixel is the (weighted) average of the pixels in the neighborhood.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving-Window Transformations}


EECE/CS 253 Image Processing

\section*{Convolution: Mathematical Representation}

If a MW transformation is linear then it is a convolution:
\[
\mathbf{J}(r, c)=[\mathbf{I} * \mathbf{h}](r, c)=\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{I}(r-\rho, c-\chi) \mathbf{h}(\rho, \chi) d \rho d \chi,
\]
for a real image \((\mathbf{I}: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R})\), or for a digital image \((\mathbf{I}: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z})\) :
\[
\mathbf{J}(r, c)=[\mathbf{I} * \mathbf{h}](r, c)=\sum_{\rho=-s}^{s} \sum_{\chi=-d}^{d} B(r-\rho, c-\chi) \mathbf{h}(\rho, \chi)
\]

\section*{Convolution Mask (Weight Matrix)}
- The object, \(\mathbf{h}(\rho, \chi)\), in the equation is a weighting function, or in the discrete case, a rectangular matrix of numbers.
- The matrix is the moving window.
- Pixel \((r, c)\) in the output image is the weighted sum of pixels from the original image in the neighborhood of \((r, c)\) traced by the matrix.
- Each pixel in the neighborhood of \((r, c)\) is multiplied by the corresponding matrix value - after the matrix is rotated by \(180^{\circ}\). (See slide 22).
- The sum of those products is the value of pixel \((r, c)\) in the output image

\section*{Convolution Masks: Moving Window}



\section*{Convolution Masks: Moving Window}


\section*{Convolution by Moving Window}


\section*{Moving Window Transform: Example}

original


\section*{\(3 \times 3\) average}

\section*{Moving Window Transform: Example}


> original

\section*{\(3 \times 3\) average}

\section*{Moving Window Transform: Example}


> original

\section*{\(3 \times 3\) average}

\section*{Moving Window Transform: Example}

original

\section*{\(3 \times 3\) average}

\section*{Moving Window Transform: Example}

original

\section*{\(3 \times 3\) average}

\section*{Moving Window Transform: Example}


> original

\section*{\(3 \times 3\) average}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Moving Window Transform: Example}

original

\section*{\(3 \times 3\) average}

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Convolution by Rotating and Shifting the Weight Matrix}


Result of sum of products


The weight matrix has a gray 'L' at its left and zeros (white) elsewhere.


At the locations not shown, the results were zeros.


The resulting image has a copy of the weight matrix pegged to the impulse location.

\section*{Symmetric Weight Matrix}
\begin{tabular}{|l|l|l|l|l|}
\hline\(f\) & \(e\) & \(\boldsymbol{d}\) & \(e\) & \(f\) \\
\hline\(e\) & \(\boldsymbol{c}\) & \(\boldsymbol{b}\) & \(\boldsymbol{c}\) & \(e\) \\
\hline \(\boldsymbol{d}\) & \(\boldsymbol{b}\) & \(\boldsymbol{a}\) & \(\boldsymbol{b}\) & \(\boldsymbol{d}\) \\
\hline\(e\) & \(\boldsymbol{c}\) & \(\boldsymbol{b}\) & \(\boldsymbol{c}\) & \(e\) \\
\hline\(f\) & \(e\) & \(\boldsymbol{d}\) & \(e\) & \(f\) \\
\hline
\end{tabular}

A symmetric weight matrix is unchanged by rotation through \(180^{\circ}\).

\section*{Three ways to compute a convolution}
1. Moving window transform as just shown.
2. Shift multiply add.
3. Fourier transform.


EECE/CS 253 Image Processing

\section*{Shift-Multiply-Add Approach}
- The image is copied 1 time for each element in the convolution mask.
- Each copy is shifted relative to the original by the displacement of its associated mask element.
- Each copy is multiplied by the value of its associated mask element.
- The set of shifted and multiplied images is summed pixel wise.

\section*{Convolution by an Impulse}

An impulse is a digital image, that has a single pixel with value 1 ; all others have value zero. An impulse at location ( \(\rho, \chi\) ) is represented by:
\[
\delta(r-\rho, c-\chi)=\left\{\begin{array}{l}
1, \text { if } r=\rho \text { and } c=\chi \\
0, \text { otherwise }
\end{array}\right.
\]

If an image is convolved with an impulse of weight \(w\) at location ( \(\rho, \chi\) ), then the image is multiplied by \(w\) and shifted in location down by \(\rho\) pixels and to the right by \(\chi\) pixels.
\[
[\mathbf{I} * w \delta(r-\rho, c-\chi)](r, c)=w \mathbf{I}(r-\rho, c-\chi)
\]

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution by an Impulse}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution by Two Impulses}


Two copies, one moved, one not moved, averaged.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution by Three Impulses}


Three copies, two moved, one not moved, averaged.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution by Five Impulses}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution by Five Impulses}


Moved adjacent to each other, the convolution becomes a blurring filter.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution by Five Impulses}


The impulses become values in a \(3 \times 3\) neighborhood.


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution by Five Impulses}



The convolution mask has five elements at \(1 / 5\) and four at 0 .


EECE/CS 253 Image Processing

\section*{Convolution by Copying, Multiplying, and Shifting the Image}

For each element \(\mathbf{h}\left(r_{\mathrm{h}}, c_{\mathrm{h}}\right)\) in weight matrix, \(\mathbf{h}\), image \(\mathbf{I}\) is copied into a zero-padded image, \(\mathbf{P}\), starting at \(\left(r_{\mathrm{h}}, c_{\mathrm{h}}\right)\).
Each \(\mathbf{P}\) is multiplied by the corresponding weight, \(\mathbf{h}\left(r_{\mathrm{h}}, c_{\mathrm{h}}\right)\).
All the \(\mathbf{P}\) images are summed pixel-wise then divided by the sum of the elements of \(\mathbf{h}\). The result is cropped out of the center of the accumulated Ps.

original image, I
padded image, \(\mathbf{P}\)
effective neighborhood
\begin{tabular}{|l|l|l|}
\hline\(h(-1,-1)\) & \(h(-1,0)\) & \(h(-1,1)\) \\
\hline\(h(0,-1)\) & \(h(0,0)\) & \(h(0,1)\) \\
\hline\(h(1,-1)\) & \(h(1,0)\) & \(h(1,1)\) \\
\hline
\end{tabular}

х!иреш ұиб!әм
aligned pixels to be summed
weight for image \(\bigcirc\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution by Copying, Multiplying, and Shifting the Image}

original image, I effective neighborhood padded image, \(\mathbf{P}\)


In the result, the origin of the weight matrix coincides with the original location of the impulse.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Convolution by}

Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white \(=0\) ) before being accumulated (pixelwise) in the padded image

The position of the black square relative the center of the weight matrix indicates the shift of the original image relative to the middle of the padded image.


In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Convolution by}

Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white \(=0\) ) before being accumulated (pixelwise) in the padded image




In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

Zero Padding an Image for Convolution: Variable Names.

CE/CS 253 Image Processing nderbilt University School of Engineering


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

Convolution by Copying and Shifting the Image

To use the image shift-multiplyaccumulate algorithm, create an accumulator image, \(A\), that is \(R+m-1\) rows by \(C+n-1\) columns

Image \(I\) is \(R \times C\)
\[
C+n-1
\]


Convolution by Copying, Multiplying, and Shifting the Image
\(13 \times 13\) image convolved by \(6 \times 6\) mask.

Image is constant; mask has only 6 nonzero values all on the diagonal.

Image is shifted to mask location, multiplied by value,
 and accumulated.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

Convolution by Copying and Shifting the Image

When done, copy the output image from the accumulator starting at (hrorig, hcorig) and ending at (hrorig+R-1, hcorig+C-1)


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Convolution Examples: Original Images}


\author{
Vanderbilt University School of Engineering
}

\section*{Convolution Examples: \(3 \times 3\) Blur}

\(\frac{1}{25}\left[\begin{array}{lllll}1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1\end{array}\right]\)

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution Examples: 5×5 Blur}

\(\frac{1}{81}\left[\begin{array}{llllllllll}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{array}\right]\)

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution Examples: 9×9 Blur}

\(\left[\begin{array}{lllllllllllll}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{array} 1111\right.\) 11111111111111111 11111111111111111 11111111111111111 11111111111111111 11111111111111111 11111111111111111
1111111111111111
111111111111

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution Examples: \(17 \times 17\) Blur}


\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}


\section*{Symmetric Edge Detection}

\author{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\[
\begin{array}{r}
2 I(r, c)-I(r, c-1) \\
-I(r, c+1)
\end{array}
\]
\[
\begin{array}{r}
2 I(r, c)-I(r-1, c) \\
-I(r+1, c)
\end{array}
\]
\[
\begin{aligned}
& 4 I(r, c)- \\
& I(r-1, c)-I(r+1, c)- \\
& I(r, c-1)-I(r, c+1)
\end{aligned}
\]

\begin{tabular}{|r|r|r|}
\hline & -1 & \\
\hline & 2 & \\
\hline & -1 & \\
\hline
\end{tabular}

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Convolution Examples: Original Images}


\section*{Convolution Examples: Vertical Difference}


\author{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Convolution Examples: Horizontal Difference}


\section*{Convolution Examples: H + V Diff.}


\author{
Vanderbilt University School of Engineering
}

\section*{Convolution Examples: Diagonal Difference}


\section*{Convolution Examples: Diagonal Difference}


\section*{Convolution Examples: D + D Difference}


\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Convolution Examples: H + V + D Diff.}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Convolution Examples: Original Images}


\title{
EECE\CS 253 Image Processing
}

\author{
Lecture Notes: Frequency Filtering
}

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2011

EECE/CS 253 Image Processing

\section*{Convolution Property of the Fourier Transform}

Let functions \(\mathbf{f}(r, c)\) and \(\mathbf{g}(r, c)\) have
Fourier Transforms \(\mathrm{F}(u, v)\) and \(\mathrm{G}(u, v)\).
\[
\begin{aligned}
& *=\text { convolution } \\
& \cdot=\text { multiplication }
\end{aligned}
\]

Then,
\[
\mathscr{J}\{\mathbf{f} * \mathbf{g}\}=\mathbf{F} \cdot \mathbf{G} .
\]

Moreover,
\[
\mathscr{J}\{\mathbf{f} \cdot \mathbf{g}\}=\mathbf{F} * \mathbf{G} .
\]

The Fourier Transform of a convolution equals the product of the Fourier Transforms. Similarly, the Fourier Transform of a convolution is the product of the Fourier Transforms

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering


\section*{Convolution via Fourier Transform}

Image \& Mask
Transforms



Pixel-wise Product

Inverse
Transform

EECE/CS 253 Image Processing

\section*{How to Convolve via FT in Matlab}
1. Read the image from a file into a variable, say \(\mathbf{I}\).
2. Read in or create the convolution mask, \(\mathbf{h}\). The mask is usually 1 -band
3. Compute the sum of the mask: \(\mathbf{s}=\operatorname{sum}(\mathrm{h}(:))\);
4. If \(\mathrm{s}=\mathbf{0}\), set \(\mathrm{s}=1\);
5. Replace \(\mathbf{h}\) with \(\mathbf{h}=\mathbf{h} / \mathbf{s}\);
6. Create: H = zeros(size(I));

For color images you may need to do each step for each band separately.
7. Copy \(\mathbf{h}\) into the middle of \(\mathbf{H}\).
8. Shift H into position: H = ifftshift(H);
9. Take the 2D FT of I and H: FI=fft2(I); FH=fft2(H);
10. Pointwise multiply the FTs: FJ=FI . *FH;
11. Compute the inverse FT: J = real(ifft2(FJ));

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Coordinate Origin of the FFT}
Center \(=\)
(floor(R/2)+1, floor(C/2)+1)


Even


Odd



\section*{Matlab’s fftshift and ifftshift}
\[
\begin{aligned}
\mathbf{J} & =\text { fftshift(I) : } \\
& \mathbf{I}(1,1) \rightarrow \mathbf{J}(\lfloor\mathrm{R} / 2\rfloor+1,\lfloor\mathrm{C} / 2\rfloor+1)
\end{aligned}
\]

\[
\begin{aligned}
& \text { I }=\text { ifftshift }(J): \\
& \quad \mathbf{J}(\lfloor\mathrm{R} / 2\rfloor+1,\lfloor\mathrm{C} / 2\rfloor+1) \rightarrow \mathbf{I}(1,1)
\end{aligned}
\]

where \(\lfloor x\rfloor=\) floor \((x)=\) the largest integer smaller than \(x\).

\section*{Blurring: Averaging / Lowpass Filtering}

\section*{Blurring results from:}

। Pixel averaging in the spatial domain:
- Each pixel in the output is a weighted average of its neighbors.
- Is a convolution whose weight matrix sums to 1 .

। Lowpass filtering in the frequency domain:
- High frequencies are diminished or eliminated
- Individual frequency components are multiplied by a nonincreasing function of \(\omega\) such as \(1 / \omega=1 / \sqrt{\left(u^{2}+v^{2}\right)}\).

The values of the output image are all non-negative.

\section*{Sharpening: Differencing / Highpass Filtering}

Sharpening results from adding to the image, a copy of itself that has been:

। Pixel-differenced in the spatial domain:
- Each pixel in the output is a difference between itself and a weighted average of its neighbors.
- Is a convolution whose weight matrix sums to 0 .

। Highpass filtered in the frequency domain:
- High frequencies are enhanced or amplified.
- Individual frequency components are multiplied by an increasing function of \(\omega\) such as \(\alpha \omega=\alpha \sqrt{\left(u^{2}+v^{2}\right)}\), where \(\alpha\) is a constant.

The values of the output image positive \& negative.

\section*{Recall:}

\section*{Convolution Property of the Fourier Transform}

Let functions \(\mathbf{f}(r, c)\) and \(\mathbf{g}(r, c)\) have
Fourier Transforms \(\mathbf{F}(u, v)\) and \(\mathbf{G}(u, v)\).
\[
\begin{aligned}
& *=\text { convolution } \\
& \cdot=\text { multiplication }
\end{aligned}
\]

Then,
\[
\mathscr{F}\{\mathbf{f} * \mathbf{g}\}=\mathbf{F} \cdot \mathbf{G} .
\]

Moreover,
\[
\mathscr{F}\{\mathbf{f} \cdot \mathbf{g}\}=\mathbf{F} * \mathbf{G} .
\]

Thus we can compute \(\mathbf{f} * \mathbf{g}\) by
\[
\mathbf{f} * \mathbf{g}=\mathscr{F}^{-1}\{\mathbf{F} \cdot \mathbf{G}\} .
\]

The Fourier Transform of a convolution equals the product of the Fourier Transforms. Similarly, the Fourier Transform of a convolution is the product of the Fourier Transforms

\section*{Ideal Lowpass Filter}

\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Ideal Lowpass Filter}

Image size: \(512 \times 512\) FD filter radius: 16


Fourier Domain Rep. Spatial Representation

Central Profile

\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Ideal Lowpass Filter}

Image size: \(512 \times 512\) FD filter radius: 8


Fourier Domain Rep. Spatial Representation


Central Profile

Consider the

\section*{EECE/CS 253 Image Processing} image below:

\section*{Power Spectrum and Phase of an Image}


Original Image


\section*{Power Spectrum}


Phase

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Ideal Lowpass Filter}

Image size: \(512 \times 512\) FD filter radius: 16


Original Image

Power Spectrum


Ideal LPF in FD

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Ideal Lowpass Filter}

Image size: \(512 \times 512\) FD filter radius: 16


Filtered Image
Filtered Power Spectrum


Original Image

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Ideal Lowpass Filter}

Image size: \(512 \times 512\) FD filter radius: 16


Original Image

Filtered Power Spectrum
Filtered Image

\section*{Ideal Highpass Filter}

\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Ideal Highpass Filter}

Image size: \(512 \times 512\) FD notch radius: 16


Fourier Domain Rep.
Spatial Representation
Central Profile

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Ideal Highpass Filter}

Image size: \(512 \times 512\) FD notch radius: 16


Original Image


Power Spectrum
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Ideal Highpass Filter}

Image size: \(512 \times 512\) FD notch radius: 16


Original Image

Filtered Power Spectrum


Filtered Image*

\section*{Ideal Bandpass Filter}

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Ideal Bandpass Filter}

A bandpass filter is created by
(1) subtracting a FD radius \(\rho_{2}\) lowpass filtered image from a FD radius \(\rho_{1}\) lowpass filtered image, where \(\rho_{2}<\rho_{1}\), or
(2) convolving the image with a mask that is the difference of the two lowpass masks.


FD LP mask with radius \(\rho_{1}\)


FD LP mask with radius \(\rho_{2}\)


FD BP mask \(\rho_{1}-\rho_{2}\)
*signed image: 0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal Bandpass Filter}


\section*{image LPF radius \(\rho_{1}\)}
image BPF radii \(\rho_{1}, \rho_{2}{ }^{*}\)
*signed image: 0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal Bandpass Filter}

original image*

filter power spectrum

filtered image
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{A Different Ideal Bandpass Filter}

original image
filter power spectrum

filtered image*

\section*{The Optimal Filter}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Uncertainty Relation}


If \(\Delta x \Delta y\) is the extent of the object in space and if \(\Delta u \Delta v\) is its extent in frequency then,
\[
\Delta x \Delta y \cdot \Delta u \Delta v \geq \frac{1}{16 \pi^{2}}
\]

A small object in space has a large frequency extent and vice-versa.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{The Uncertainty Relation}

\(\rightarrow\) small extent \(\leftarrow\)


Recall: a symmetric pair of impulses in the frequency domain becomes a sinusoid in the spatial domain.

A symmetric pair of lines in the frequency domain becomes a sinusoidal line in the spatial domain.

\section*{Ideal Filters Do Not Produce Ideal Results}


A sharp cutoff in the frequency domain...
...causes ringing in the spatial domain.

\section*{Ideal Filters Do Not Produce Ideal Results}


Blurring the image above w/ an ideal lowpass filter...
...distorts the results with ringing or ghosting.

\section*{Optimal Filter: The Gaussian}


The Gaussian filter optimizes the uncertainty relation. It provides the sharpest cutoff with the least ringing.

\section*{One-Dimensional Gaussian}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Two-Dimensional Gaussian}

If \(\mu\) and \(\sigma\) are
\[
R=512, C=512
\]


\section*{Optimal Filter: The Gaussian}


With a gaussian lowpass filter, the image above ...
... is blurred without ringing or ghosting.

\section*{Compare with an "Ideal" LPF}


Blurring the image above w/ an ideal lowpass filter...
...distorts the results with ringing or ghosting.

\section*{Gaussian Lowpass Filter}

\section*{Gaussian Lowpass Filter}

Image size: \(512 \times 512\) SD filter sigma \(=8\)


Fourier Domain Rep. Spatial Representation


Central Profile

\section*{Gaussian Lowpass Filter}

Image size: \(512 \times 512\) SD filter sigma \(=2\)



Fourier Domain Rep.
Spatial Representation
Central Profile

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gaussian Lowpass Filter}

Image size: \(512 \times 512\) SD filter sigma \(=8\)


Original Image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gaussian Lowpass Filter}

Image size: \(512 \times 512\) SD filter sigma \(=8\)


Filtered Image
Filtered Power Spectrum

Original Image

\section*{Gaussian Highpass Filter}

\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Gaussian Highpass Filter}

Image size: \(512 \times 512\) FD notch sigma \(=8\)


Fourier Domain Rep.
Spatial Representation
Central Profile

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gaussian Highpass Filter}


Original Image


Power Spectrum


Gaussian HPF in FD

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Gaussian Highpass Filter}

Image size: \(512 \times 512\) FD notch sigma \(=8\)


Original Image
Filtered Power Spectrum


Filtered Image*
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Another Gaussian Highpass Filter}

original image

filter power spectrum

filtered image*

\section*{Gaussian Bandpass Filter}

\section*{Vanderbilt University School of Engineering}

\section*{Gaussian Bandpass Filter}

A bandpass filter is created by
(1) subtracting a FD radius \(\rho_{2}\) lowpass filtered image from a FD radius \(\rho_{1}\) lowpass filtered image, where \(\rho_{2}<\rho_{1}\), or
(2) convolving the image with a mask that is the difference of the two lowpass masks.


FD LP mask with radius \(\sigma_{1}\)


FD LP mask with radius \(\sigma_{2}\)


FD BP mask \(\sigma_{1}-\sigma_{2}\)
*signed image: 0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Gaussian Bandpass Filter}

image LPF radius \(\rho_{1}\)

image LPF radius \(\rho_{2}\)

image BPF radii \(\rho_{1}, \rho_{2}{ }^{\star}\)
*signed image: 0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal Bandpass Filter}

original image
filter power spectrum
filtered image*

\section*{Gaussian Bandpass Filter}

Image size: \(512 \times 512\) sigma \(=2-\) sigma \(=8\)


Fourier Domain Rep.
Spatial Representation
Central Profile

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Gaussian Bandpass Filter}

Image size: \(512 \times 512\) sigma \(=2-\) sigma \(=8\)


Original Image

Power Spectrum
Gaussian BPF in FD
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Gaussian Bandpass Filter}

Image size: \(512 \times 512\) sigma \(=2-\) sigma \(=8\)


Filtered Image*

Filtered Power Spectrum


Original Image

\title{
Ideal vs. Gaussian Filters
}
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal Lowpass and Highpass Filters}


Ideal LPF


Original Image


Ideal \(\mathrm{HPF}^{*}\)
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Gaussian Lowpass and Highpass Filters}


Gaussian LPF



Gaussian HPF*

Original Image
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal and Gaussian Bandpass Filters}


Ideal BPF \(^{*}\)
Original Image
Gaussian BPF*
*signed image:
0 mapped to 128

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Gaussian and Ideal Bandpass Filters}


Gaussian BPF*


Original Image


Ideal BPF*

\section*{Effects on Power Spectrum}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum and Phase of an Image}

original image

power spectrum

phase

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum and Phase of a Blurred Image}

blurred image

power spectrum

phase

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum and Phase of an Image}

original image

power spectrum

phase

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum and Phase of a Sharpened Image}

sharpened image

power spectrum

phase

\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Sharpening and Edge Enhancement

\section*{Richard Alan Peters II}

Department of Electrical Engineering and Computer Science

Fall Semester 2011

\section*{Sharpening}

। Results from high frequency enhancement since small features correspond to short wavelength sinusoids.

। Relative amplification of high frequencies in the Fourier domain corresponds to differentiation in the spatial domain.

। On a discrete image, differentiation corresponds to pixel differencing.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Derivative Property of the Fourier Transform}

The FT of the partial derivative w.r.t. \(r\) (in the row direction) of an image, I ...
\[
\begin{aligned}
\mathscr{F}\left\{\frac{\partial \mathbf{I}}{\partial r}\right\} & =\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\partial}{\partial r} \mathbf{I}(r, c) e^{-i 2 \pi(u c+v r)} d c d r \\
& =\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{I}(r, c) \cdot \frac{\partial}{\partial r} e^{-i 2 \pi(u c+v r)} d c d r \quad \text { by parts } \\
& =\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{I}(r, c) \cdot(-i 2 \pi v) e^{-i 2 \pi(u c+v r)} d c d r
\end{aligned}
\]
... is equal to the product of
the FT of the image and the corresponding frequency variable, v.

This results in
\(=-i 2 \pi v \mathscr{C}\{\mathbf{I}\}=-i 2 \pi v \mathbf{F}(u, v)\).
horizontal HF enhancement

\section*{Differentiation is Highpass Filtering}

Directional derivative in \(c\).

Vertical HF Enhancement
\[
\begin{aligned}
& \mathscr{F}\left\{\frac{\partial \mathbf{I}}{\partial c}\right\}(u, v) \propto u \mathscr{F}\{\mathbf{I}\}(u, v) \\
& \mathscr{F}\left\{\frac{\partial \mathbf{I}}{\partial r}\right\}(u, v) \propto v \mathscr{F}\{\mathbf{I}\}(u, v)
\end{aligned}
\]

Directional derivative in \(r\).

Horizontal HF Enhancement

\section*{Fourier Transforms of Sums of Derivatives}
\[
\mathfrak{F}\left\{\left[\frac{\partial}{\partial r}+\frac{\partial}{\partial c}\right] \mathbf{I}\right\}=-i 2 \pi(u+v) \mathscr{F}\{\mathbf{I}\}=-i 2 \pi(u+v) \mathbf{F}(u, v)
\]

Sum of first-order ...linear amplification partial derviatives... of high frequencies
\[
\mathscr{F}\left\{\left[\frac{\partial^{2}}{\partial r^{2}}+\frac{\partial^{2}}{\partial c^{2}}\right] \mathbf{I}\right\}=-4 \pi^{2}\left(u^{2}+v^{2}\right) \mathscr{F}\{\mathbf{I}\}=-4 \pi^{2}\left(u^{2}+v^{2}\right) \mathbf{F}(u, v)
\]

Sum of second-order partial derviatives...
...quadratic amplification of high frequencies

\section*{Sharpening: Differencing / Highpass Filtering}

Sharpening results from adding to the image a copy of itself that has been:

। Pixel-differenced in the spatial domain:
- Each pixel in the output is a difference between itself and a weighted average of its neighbors.
- Is a convolution whose weight matrix sums to 0 .

1 Highpass filtered in the frequency domain:
- High frequencies are enhanced or amplified.
- Individual frequency components are multiplied by an increasing function of \(\omega\) such as \(\alpha \omega=\alpha \sqrt{\left(u^{2}+v^{2}\right)}\), where \(\alpha\) is a constant.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Differencing / Highpass Filtering}

original image, I

power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Differencing / Highpass Filtering}

power spectrum of \(\mathbf{h}=\left[\begin{array}{ll}-1 & 1\end{array}\right]\)

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I}(r, c)-\mathbf{I}(r, c-1)\)

\section*{Sharpening: Differencing / Highpass Filtering}

negative pixels in differenced image

positive pixels in differenced image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Differencing / Highpass Filtering}

original image, I

power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Differencing / Highpass Filtering}

sharpened image, \(2 \mathbf{I}(r, c)-\mathbf{I}(r, c-1)\)

power spectrum

HP Filter: Direct Linear Frequency Enhancement

original image

power spectrum


HF enhanced: \(\sqrt{u^{2}+v^{2}} \cdot \mathscr{F}\{\mathbf{I}\}(u, v)\)

\section*{Sharpening: Direct Linear Frequency Enhancement}

original image

power spectrum

original + linearly enhanced

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{HF Enhancement and Edge Detection}


thresholded linear HF image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{HF Filtering: Direct Quadratic Frequency Enhancement}

original image

power spectrum


HF enhanced: \(\left(u^{2}+v^{2}\right) \cdot \mathscr{F}\{\mathbf{I}\}(u, v)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Direct Quadratic Frequency Enhancement}

original image

power spectrum

original + quadratically enhanced

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{HF Enhancement and Edge Detection}


thresholded quad. HF image

\section*{Differentiation Through Integration}
1.
\[
\frac{\partial}{\partial w}[\mathbf{I} * \mathbf{h}](r, c)=\frac{\partial}{\partial w} \iint_{\text {supp(I) }} \mathbf{I}(\rho-r, \chi-c) \mathbf{h}(\rho, \chi) d \rho d \chi
\]
\[
w=\alpha x+\beta y, \quad \alpha+\beta=1
\]

Assume that \(h(\rho, x)=\) \(\delta(\rho, \mathrm{X})\). Then \(\mathrm{I} * \mathrm{~h}=\mathrm{I}\). and \(\partial I / \partial w=\partial(I * h) / \partial w\).
2. \(\mathcal{E}\left\{\frac{\partial}{\partial w} \mathbf{J}(r, c)\right\}=J Z \mathcal{F}\{\mathbf{J}(r, c)\}\)
\[
z=\alpha u+\beta v, \quad \alpha+\beta=1
\]

Differentiation property of the Fourier Transform.
3.
\(\mathfrak{F}\{\mathbf{I} * \mathbf{h}\}=\mathscr{F}\{\mathbf{I}\} \cdot \mathscr{F}\{\mathbf{h}\}\)
\(\mathbf{I} * \mathbf{h}=\mathscr{F}^{-1}\{\mathscr{F}\{\mathbf{I}\} \cdot \mathscr{F}\{\mathbf{h}\}\}\)

Convolution property of the Fourier Transform.

\section*{Differentiation Through Integration}
4.
\[
\begin{aligned}
\mathscr{F}\left\{\frac{\partial}{\partial w}[\mathbf{I} * \mathbf{h}]\right\} & =j z \cdot \mathscr{F}\{\mathbf{I}\} \cdot \mathscr{F}\{\mathbf{h}\} \\
& =[j z \cdot \mathscr{F}\{\mathbf{I}\}] \cdot \mathscr{F}\{\mathbf{h}\} \\
& =\mathscr{F}\{\mathbf{I}\} \cdot[j z \cdot \mathscr{F}\{\mathbf{h}\}]
\end{aligned} \begin{aligned}
& w=\alpha x+\beta y \quad \alpha+\beta=1 \\
& z=\alpha u+\beta v, \quad \alpha+\beta=1
\end{aligned}
\]
5. \(\frac{\partial}{\partial w}[\mathbf{I} * \mathbf{h}](r, c)=\left[\mathbf{I} * \frac{\partial}{\partial w} \mathbf{h}\right](r, c)\)

Apply 2 and 3 to 1 to get this.
\(w\) and \(z\) are directions in the plane.

The derivative of a convolution of I by \(h\) is the convolution of I by the derivative of \(h\).

\section*{Vertical Edge Detection}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering


\section*{Symmetric Edge Detection}

\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\[
\begin{array}{r}
2 \mathbf{I}(r, c)-\mathbf{I}(r, c-1) \\
-\mathbf{I}(r, c+1)
\end{array}
\]
\[
\begin{array}{r}
2 \mathbf{I}(r, c)-\mathbf{I}(r-1, c) \\
-\mathbf{I}(r+1, c)
\end{array}
\]
\[
\begin{aligned}
& 4 \mathbf{I}(r, c)- \\
& \mathbf{I}(r-1, c)-\mathbf{I}(r+1, c)- \\
& \mathbf{I}(r, c-1)-\mathbf{I}(r, c+1)
\end{aligned}
\]

\begin{tabular}{|r|r|r|}
\hline & -1 & \\
\hline & 2 & \\
\hline & -1 & \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|}
\hline & -1 & \\
\hline-1 & 4 & -1 \\
\hline & -1 & \\
\hline
\end{tabular}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Original Image}

power spectrum of I

image I

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Left Difference}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{ll}-1 & 1\end{array}\right]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{ll}-1 & 1\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Left Difference}

image I

\(\mathbf{I}+(\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{ll}-1 & 1\end{array}\right]\right)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Right Difference}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *[1-1]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{ll}1 & -1\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Right Difference}

image I

\(\mathbf{I}+(\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{ll}-1 & 1\end{array}\right]\right)\)

\section*{Vertical Edges (L+R Diffs)}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *[-12-1]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{lll}-1 & 2 & -1\end{array}\right]\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Original Image + Vertical Edges}

image I

\(\mathbf{I}+(\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{lll}-1 & 2 & -1\end{array}\right]\right)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Down Difference}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{r}1 \\ -1\end{array}\right]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{r}1 \\ -1\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Down Difference}

image I

\(\mathbf{I}+\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{r}1 \\ -1\end{array}\right]\right)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Up Difference}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{r}1 \\ -1\end{array}\right]\)

\(\boldsymbol{T} \boldsymbol{n}=\boldsymbol{T}\left[\begin{array}{r}1 \\ -1\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Up Difference}

image I

\(\mathbf{I}+(\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{r}1 \\ -1\end{array}\right]\right)\)

\section*{Horizontal Edges (D+U Diffs)}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{c}-1 \\ -1 \\ -1\end{array}\right]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{c}-1 \\ -1 \\ -1\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Horizontal Edges}

image I

\(\mathbf{I}+(\mathbf{I} * \mathbf{h})=\mathbf{I}+\left(\mathbf{I} *\left[\begin{array}{c}{\left[\begin{array}{c}-1 \\ -1 \\ -1\end{array}\right)}\end{array}\right)\right.\)

\section*{Horiz. + Vert. Edges (L+R+D+U Diffs)}

power spectrum of \(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{cc}-0.1 \\ 0 & -1 \\ 0 & -1 \\ 0\end{array}\right]\)

\(\mathbf{I} * \mathbf{h}=\mathbf{I} *\left[\begin{array}{ccc}0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0\end{array}\right]\)

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Horiz. + Vert. Edges}

original

sharpened

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Original Image + Horiz. + Vert. Edges}

sharpened

original

EECE/CS 253 Image Processing

\section*{Sharpening Through Blurring: Unsharp Masking}

Let I be an image.
Let \(\mathbf{G}_{\sigma}\) be a Gaussian convolution mask.
Then \(\mathbf{J}=\mathbf{I} * \mathbf{G}_{\sigma}\) is a blurred image and \(\mathbf{K}=\mathbf{I}-\mathbf{J}\) contains all the high spatial frequencies from I.

Define:
\[
\mathbf{U}=(1+\alpha) \mathbf{K}+\mathbf{J}=\alpha \mathbf{K}+\mathbf{I},
\]
where, typically \(0<\alpha<2\).

Often, the control, a, is given as a percent value.
Then the formula is \((a / 100) \star K+I\).
\(\mathbf{U}\) is called the unsharp masking of image \(\mathbf{I}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening Through Blurring: Unsharp Masking}

original image

log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening Through Blurring: Unsharp Masking}


Gaussian blur \(\sigma=4\)

log power spectrum

\section*{Sharpening Through Blurring: Unsharp Masking}

original minus Gaussian blur
log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening Through Blurring: Unsharp Masking}

unsharp masked image

log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening Through Blurring: Unsharp Masking}

original image

unsharp masked image

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Image Sharpening}

original image

linearly enhanced

quadratic enh.

unsharp masked

\section*{Noise Enhancement: Problem with Sharpening}

। Noise occurs in every natural imaging device
- Quantum effects in CCD arrays
- Random distribution of silver halide grains in film
- Neuronal noise in the retina

। Spatially independent noise
- The noise in one sensor has no effect on that in its neighbors
- \(\Rightarrow\) the autocorrelation of the signal is an impulse at the origin
- The chances of getting repeated patterns of any frequency are virtually nil
\(-\quad \Rightarrow\) the frequency spectrum of the noise is flat
Recall: Autocorrelation = inverse Fourier transform of power spectrum; Fourier transform of an impulse at \((0,0)\) is a constant.

\section*{Gaussian IID Noise Field}

\section*{No spatial correlation}



IID: Independent, Identically Distributed

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum and Autocorrelation of IID}

\[
\operatorname{PS}(\mathbf{I})=\log [|\mathscr{F}(\mathbf{I})|+1]
\]

\[
R(\mathbf{I})=\operatorname{Re}\left\{\mathscr{F}^{-1}[|\mathscr{f}(\mathbf{I})|]\right\}
\]

\section*{Noise Enhancement: Problem with Sharpening}
- The spectra of most natural images fall-off toward the high frequencies.

। IID noise has a flat spectrum.
। Therefore, at some relatively high frequency (HF) the energy in the noise is greater than that in the uncorrupted image.

। Sharpening multiplies the FT of the image by \(u\) and \(v\) (or by linear combinations of them) which, at HF, increases the noise more than the uncorrupted image.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Images}

image

noise field

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Images}

image center row log power spectrum

noise field center row log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Images}

image + noise field

image + noise field center row log PS

\section*{Effects of Noise on Images (Power Spectra)}

original image

noise image

\section*{Effects of Noise on Images (Power Spectra)}

original image

noisy image

\section*{Effects of Noise on Images (Power Spectra)}

original image

blue indicates noise > image

\section*{Effects of Noise on Images (Power Spectra)}

noise image

red indicates image > noise

\section*{Effects of Noise on Images (Power Spectra)}

noisy image

image \& noise

\section*{Effects of Noise on Linear Enhancement of HF}

original image

noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Linear Enhancement of HF}


HF enhanced original


HF enhanced noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Linear Enhancement of HF}

original image

noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Linear Enhancement of HF}


HF enhanced original


HF enhanced noisy image

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Effects of Noise on Linear Enhancement of HF}

original image


HF enhanced original

\section*{Effects of Noise on Linear Enhancement of HF}

noisy image


HF enhanced noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Noise on Linear Enhancement of HF}

noisy image

power spectrum


HF enhanced: \(\sqrt{u^{2}+v^{2}} \cdot \mathscr{F}\{\mathbf{I}\}(u, v)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

Sharpening: Effects of Noise on Linear Enhancement

noisy image

power spectrum

noisy + linearly enhanced

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

Effects of Noise on Quadratic Enhancement of HF

noisy image

power spectrum


HF enhanced: \(\left(u^{2}+v^{2}\right) \cdot \mathscr{F}\{\mathbf{I}\}(u, v)\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sharpening: Effects of Noise on Quadratic Enhancement}

noisy image

power spectrum

original + quadratically enhanced

\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Pixelization and Quantization

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2011

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixelization ...}

... is a special effect often used to hide identities ...

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixelization and Quantization}

high-res image

pixelated

quantized

pixelated \& quantized

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Pixelization and Quantization}


\section*{Pixelization}

Take the average within each square.

\(\mathbf{I}(\rho, \chi)\)

\(\uparrow\)

\(\mathbf{I}_{p}(r, c)\)
high-res image
\[
\mathbf{I}_{\mathrm{DS}}(r, c)=\frac{1}{\Delta^{2}} \sum_{\rho=r \Delta}^{(r+1) 1 /-1} \sum_{\chi=c \Delta}^{(c+1) \Delta-1} \mathbf{I}(\rho, \chi) ; \quad \mathbf{I}_{\mathbf{P}}=\mathbf{I}_{\mathrm{D}} \uparrow \uparrow \Delta
\]
pixelated image

\section*{Pixelization}

Take the average within each square.

\(\mathbf{I}(\rho, \chi)\)

\(\uparrow\)

\(\mathbf{I}_{p}(r, c)\)
high-res image
\[
\mathbf{I}_{\mathrm{DS}}(r, c)=\frac{1}{\Delta^{2}} \sum_{\rho=r \Delta}^{(r+1) 1 /-1} \sum_{\chi=c \Delta}^{(c+1) \Delta-1} \mathbf{I}(\rho, \chi) ; \quad \mathbf{I}_{\mathbf{P}}=\mathbf{I}_{\mathrm{D}} \uparrow \uparrow \Delta
\]
pixelated image

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Pixelization}

Take the average within each square.

\(\mathbf{I}(\rho, \chi)\)
high-res image
\[
\boldsymbol{I}_{\mathrm{DS}}(r, C)=\frac{1}{\Delta^{2}} \sum_{\rho=r \Delta}^{(r+1) \Delta-1} \sum_{\chi=c \Delta}^{(c+1) \Delta-1} \boldsymbol{T}(\rho, \chi), \quad \mathbf{T}_{\mathbf{p}}=\boldsymbol{T}_{\mathrm{DS}} \uparrow \underbrace{}_{(\text {upsampled })}
\]

\(\mathbf{I}_{p}(r, c)\)
pixelated image

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Pixelization}

Take the average within each square.

\(\mathbf{I}(\rho, \chi)\)
high-res image
\[
\mathbf{I}_{\mathrm{DS}}(r, c)=\frac{1}{\Delta^{2}} \sum_{\rho=r \Delta}^{(r+1) \Delta-1} \sum_{\chi=c \Delta}^{(c+1) \Delta-1} \mathbf{I}(\rho, \chi) ; \quad \mathbf{I}_{\mathbf{P}}=\underset{\text { (upsampled) }}{ } \mathbf{I}_{\mathrm{DS}} \uparrow \Delta
\]

\(\mathbf{I}_{p}(r, c)\)

pixelated image

EECE/CS 253 Image Processing

\section*{Pixelization Procedure (Part 1)}

To pixelize an \(R \times C \times B\) image I by a factor, \(s\) :
Let \(\mathbf{I}_{\mathrm{DS}}\) be an \(\lfloor R / s\rfloor \times\lfloor C / s\rfloor \times B\) image.
Let the value of \(\mathbf{I}_{\mathrm{DS}}\) at location \((r, c, b)\) be the average value of \(\mathbf{I}\) in the \(s\)-square neighborhood starting at ( \(r s, c s, b\) ).
\[
\mathbf{I}_{\mathrm{DS}}(r, c, b)=\operatorname{mean}_{(\rho, \chi) \in \mathfrak{\vartheta}_{\mathbf{I}}(r, c)}\{\mathbf{I}(\rho, \chi, b)\},
\]
where
\[
\mathscr{\mathscr { T }}_{1}(r, c)=\left\{(\rho, \chi) \left\lvert\, \begin{array}{l}
\rho \in\{r s, r s+1, \ldots,(r+1) s-1\}, \\
\chi \in\{c s, c s+1, \ldots,(c+1) s-1
\end{array}\right.\right\} .
\]

Notation:
\[
\mathbf{I}_{\mathrm{DS}}=\mathbf{I} \downarrow s
\]
\(\mathbf{I}_{\mathrm{DS}}\) is \(\mathbf{I}\) downsampled by a factor of \(s\). \(r\) \& \(c\) are indices in the downsampled image. \(r \cdot s\) \& \(c \cdot s\) are indices in the original image.

\section*{Pixelization Procedure (Part 2)}

Let \(\mathbf{I}_{\mathrm{P}}\) be an \(R \times C \times B\) image.

\[
\mathbf{I}_{\mathrm{P}}(\rho, \chi, b)=\mathbf{I}_{\mathrm{DS}}(r, c, b) \text { for }(\rho, \chi) \in \mathscr{T}_{\mathbf{I}_{\mathrm{P}}}(r, c) \text {. }
\]
where
\[
\mathscr{\mathscr { G }}_{\mathbf{I}_{\mathrm{p}}}(r, c)=\left\{(\rho, \chi) \left\lvert\, \begin{array}{l}
\rho \in\{r s, r s+1, \ldots,(r+1) s-1\}, \\
\chi \in\{c s, c s+1, \ldots,(c+1) s-1\}
\end{array}\right.\right\} .
\]

Notation:
\[
\mathbf{I}_{\mathrm{P}}=\mathbf{I}_{\mathrm{DS}} \uparrow s
\]
\(\mathbf{I}_{\mathrm{P}}\) is \(\mathbf{I}_{\mathrm{DS}}\) upsampled by a factor of \(s\).
\(r \& c\) are indices in the downsampled image. \(r \cdot s\) \& \(c \cdot s\) are indices in the upsampled image.

\section*{Pixelization}

\section*{8 of 8: \(256 \times 256\)}

\section*{Pixelization}

\section*{Pixelization}

\section*{6 of \(8: 64 \times 64\)}


\section*{Pixelization}


\section*{Pixelization}

\section*{4 of \(8: 16 \times 16\)}


\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Pixelization 3 of 8: \(8 \times 8\)}


\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Pixelization}

\section*{2 of \(8: 4 \times 4\)}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixelization}

1 of \(8: 2 \times 2\)


\section*{Pixelization}

\section*{original image}


\section*{Pixelization by Factor 32 \\ original image}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Quantization}

sıoןov uo!!!!um 9 I



\section*{EECE/CS 253 Image Processing}

\section*{Vanderbilt University School of Engineering}

\section*{Quantization}


8 bits 256 levels


4 bits 16 levels


7 bits 128 levels


3 bits 8 levels


6 bits 64 levels


2 bits 4 levels


5 bits 32 levels


\section*{Intensity Quantization}


\section*{Intensity Quantization}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enhancement of an 8-bit image}

a. original

b. contrast enh.

b. dark enhanced

b. bright enh.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enhancement of a 16-bit image}

a. original

b. contrast enh.

b. dark enhanced

b. bright enh.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effect of Quantization on Equalization}



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effect of Quantization on Equalization}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effect of Quantization on Equalization}

enhanced 8-bit

enhanced 16-bit

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effect of Quantization on Equalization}

enhanced 16-bit

enhanced 8-bit

\section*{Effect of Quantization on Equalization}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{What's in the lower eight bits?}




\section*{Application of Quantization: Steganography}

\begin{tabular}{|c|}
\hline \begin{tabular}{l}
 \\
 \\

\end{tabular} \\
\hline
\end{tabular}

If an image is quantized, say from 8 bits to 6 bits and redisplayed it can be all but impossible to tell the difference visually between the two.

\section*{Application of Quantization: Steganography}

\begin{tabular}{|c|}
\hline \begin{tabular}{l}
 \\
 \\

\end{tabular} \\
\hline
\end{tabular}

If an image is quantized, say from 8 bits to 6 bits and redisplayed it can be all but impossible to tell the difference visually between the two.

\section*{Application of Quantization: Steganography}
green-band histogram of 8-bit image

green-band histogram of 6-bit image


With simple image analysis, it is easy to tell the difference: The histograms of the two versions indicate which is which. If the 6 -bit version is displayed as an 8 -bit image it has only pixels with values \(0,4,8, \ldots, 252\).

\section*{Application of Quantization: Steganography}

If the 6 -bit version is displayed as an 8 -bit image then the 8 -bit pixels all have zeros in the lower 2 bits:


This introduces the possibility of encoding other information in the low-order bits.

That other information could be a message, perhaps encrypted, or even another image.

\(X\)-Shift \(n=\) logical left or right shift by \(n\) bits.

\section*{Application of Quantization: Steganography}


The second image is invisible because the value of each pixel is between 0 and 3 .
For any given pixel, its value is added to the to the collocated pixel in the first image that has a value from the set \(\{0,4,8, \ldots, 252\}\). The \(2^{\text {nd }}\) image is noise on the \(1^{\text {st }}\).

Image 1 in upper 6-bits. Image 2 in lower 2-bits.

\section*{EECE/CS 253 Image Processing}

\section*{Application of Quantization: Steganography}


To recover the second image (which is 2 bits per pixel per band) simply left shift the combined image by 6 bits.

\section*{Application of Quantization: Steganography}


To recover the second image (which is 2 bits per pixel per band) simply left shift the combined image by 6 bits.

\section*{Application of Quantization: Steganography}

This is so effective that two 4-bit-per-pixel images can be superimposed with only the image in the high-order bits visible. Both images contain the same amount of information but because one takes on values between 0 and 15 , the other takes on values from \{16, 32, \(48, \ldots, 240\}\), and the smaller values are added to the larger, the image in the low-order bits is effectively invisible

Images 1 and 2 each have 4-bits per pixel when combined.


\section*{Application of Quantization: Steganography}


Image quantized to 4-bits per pixel.

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Application of Quantization: Steganography}


Image 1 in upper 4-bits. Image 2 in lower 4-bits.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Application of Quantization: Steganography}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Application of Quantization: Steganography}


\section*{}

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Separate EQ of Dark and Light Regions in 16-bit Images}

original 16-bit image

contr. stretched dark reg.

contr. stretched light reg.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Separate EQ of Dark and Light Regions in 16-bit Images}

original 16-bit image

contr. stretched light + dark reg.

\section*{Dithering: Noise Improves Quantization}
- Quantizing an image into 1,2 , or 3 bits can introduce false contours.
- The addition of signed noise to the image before quantization can improve the appearance of the result. This is called dithering.
- The noise usually should have \(\mu=0\).
- The \(\sigma\) of the noise must be determined through experimentation since it depends on the image being quantized. A reasonable first choice for uniformly distributed noise is \(\sigma=\frac{1}{4} M / q\), where \(M\) is the maximum intensity value in the image (e.g. 255) and \(q\) is the number of bits in the quantized image.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dithering: use noise to reduce quant. error}


8 bits


4 bits


4 bits + noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dithering: use noise to reduce quant. error}


8 bits


3 bits


3 bits + noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dithering: use noise to reduce quant. error}


8 bits


2 bits


2 bits + noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dithering: use noise to reduce quant. error}


8 bits


1 bit


1 bit + noise

\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Sampling and Aliasing

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2011

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Image Resampling Can Lead to the Jaggies!}


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Downsampling (Decimation)}


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Downsampling (Decimation)}


\section*{Power Spectrum from Discrete Fourier Transform}


\section*{Power Spectrum from Discrete Fourier Transform}


The DFT of an image is the same size as the image.

\section*{The Scaling Property of the FT}

If \(\mathfrak{F}\{\mathbf{I}\}(v, u)=\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{I}(r, c) e^{-i 2 \pi(u c+v r)} d c d r\),
then \(\quad \mathbf{I}\left(\frac{r}{a}, \frac{c}{b}\right) \Leftrightarrow|a b| \mathfrak{F}\{\mathbf{I}\}(a v, b u)\).

This implies that if an image is reduced in size, its features in the spatial domain become smaller and its features in the frequency domain become larger.

\section*{The Uncertainty Relation}


If \(\Delta x \Delta y\) is the extent of the object in space and if \(\Delta x \Delta y\) is its extent in frequency then
\[
\Delta x \Delta y \cdot \Delta u \Delta v \geq \frac{1}{16 \pi^{2}}
\]

A small object in space has a large frequency extent and vice-versa.

\section*{First Alternative Explanation of Aliasing}

The aliasing phenomenon can be described in terms of the so-called "wagon wheel" effect. The name comes from the appearance in a 24 frame per second movie of a wagon wheel rotating slightly faster than 12 frames per second. It appears to be rotating slowly backward.
https://www.youtube.com/watch?v=6XwgbHjRo30


\section*{Second Alternative Explanation of Aliasing}

The aliasing phenomenon can also be described in terms of the convolution property of the Fourier Transform. In fact, this is the way it is usually explained.


\section*{The Sampling Function}

The sampling
\[
\operatorname{samp}_{N}(r, c)=\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(r-j N) \delta(c-k N)
\] function is a set of impulses evenly spaced on a grid.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Sampling of an Image}

An image is sampled by multiplying it by the sampling function
\[
\mathbf{I}(r, c) \cdot \operatorname{samp}_{N}(r, c)
\]


\section*{Image Sampling in the Spatial Domain}


EECE/CS 253 Image Processing

\section*{Convolution Property of the Fourier Transform}

Let functions \(f(r, c)\) and \(g(r, c)\) have
Fourier Transforms \(F(u, v)\) and \(G(u, v)\).
Then,
\[
\mathfrak{F}\{f * g\}=F \cdot G
\]

Moreover,
\[
\mathfrak{F}\{f \cdot g\}=F * G
\]
* represents convolution
- represents pointwise multiplication

The Fourier Transform of a product equals the convolution of the Fourier Transforms. Similarly, the Fourier Transform of a convolution is the product of the Fourier Transforms

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{The Fourier Transform of the Sampling Function}

The Fourier Transform of the
\[
\operatorname{samp}_{l / N}(u, v)=\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta\left(u-\frac{j}{N}\right) \delta\left(v-\frac{k}{N}\right)
\] sampling function is another sampling function but with impulses \(1 / N\) apart.

\section*{Cf. slide 23}


\section*{Convolution by an Impulse}

An impulse is a digital image, that has a single pixel with value 1; all others have value zero. An impulse at location \((\rho, \chi)\) is represented by:
\[
\delta(r-\rho, c-\chi)=\left\{\begin{array}{l}
1, \text { if } r=\rho \text { and } c=\chi \\
0, \text { otherwise }
\end{array}\right.
\]

If an image is convolved with an impulse at location ( \(\rho, \chi\) ), the image is shifted in location down by \(\rho\) pixels and to the right by \(\chi\) pixels.
\[
[\mathbf{I} * \delta(r-\rho, c-\chi)](r, c)=\mathbf{I}(r-\rho, c-\chi) .
\]

\section*{Convolution by an Impulse}


The convolution of any function with a delta function translates the function to the location of the impulse.

EECE/CS 253 Image Processing

\section*{The Fourier Transform of a Sampled Image}


\section*{The FT of a Sampled Image with No Aliasing}

For aliasing no \(\dagger\) to occur the frequency support of I must have a radius of < \(1 /(2 N)\)


\section*{Part of the Transform Computed by the FFT}

as computed (aliasing)

origin shifted (aliasing)

as computed (no aliasing)

origin shifted (no aliasing)

Although the FT of the sampled image continues on indefinitely in theory, only one copy of the complete pattern is required for processing. The Fast Fourier Transform (FFT) algorithm computes the transform with the origin in the upper left. Usually the transform is displayed with the origin in the center.

EECE/CS 253 Image Processing

\section*{Filtering Before Downsampling to Prevent Aliasing}

To sample \(\mathbf{I}(r, c)\) every \(N\) pixels multiply \(\mathscr{S}(u, v)=\mathscr{F}\{\mathbf{I}\}\) by a window \(\mathbf{H}(u, v)\) such that
\[
\mathbf{H}(u, v)=0 \text { for } u>\frac{1}{2 N} \text { or } v>\frac{1}{2 N}
\]

That is equivalent to convolving \(\mathbf{I}(r, c)\) with a spatial filter of width \(2 N\).


Thus, to downsample \(\mathbf{I}(r, c)\) by a factor of 2 , use a spatial filter of width 4. Since this is midway between 3 and 5, use a 5x5 convolution mask.

If the image is a bit blurry then a \(3 x 3\) will usually suffice

Filtering to Prevent Aliasing

\author{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

original image

power spectrum

Filtering to Prevent Aliasing

\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

to decimate original by 2 ...

... zero outside red square

Filtering to Prevent Aliasing

notice ringing in the result

ideal filter

Filtering to Prevent Aliasing


To reduce ringing try multiplying the FT by a Gaussian w/ \(\left(\sigma_{\mathrm{v}}, \sigma_{\mathrm{u}}\right)=(1 / 4 \mathrm{R}, 1 / 4 \mathrm{C})\)

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering


PS of FT \(\times\) Gaussian

ringing is reduced but result is blurry

Filtering to Prevent Aliasing

try multiplying the FT

by a Gaussian w/ \(\left(\sigma_{v}, \sigma_{u}\right)=(1 / 2 R, 1 / 2 C)\)

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}


PS of FT \(\times\) Gaussian

some aliasing but result is less blurry

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

ideal filtered

original

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

original


Gaussian w/ \(\left(\sigma_{v}, \sigma_{u}\right)=(1 / 4 R, 1 / 4 C)\)

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering


Gaussian w/ \((\sigma \mathrm{V}, \sigma \mathrm{u})=(1 / 2 \mathrm{R}, 1 / 2 \mathrm{C})\)

original

Filtering to Prevent Aliasing

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

original

decimated by factor of 2

\section*{Effect of Decimation on the DFT of an Image}
1. Decimation of an \(R \times C\) image, \(\mathbf{I}\), by a factor of \(n\) results in an \(\lfloor R / n\rfloor \times\lfloor C / n\rfloor\) image, \(\mathbf{J}\).
2. The DFT of image \(\mathbf{J}\) is the same size as \(\mathbf{J}\).
3. The uncertainty relation implies that the FT of \(\mathbf{J}\) should be \((R-1) \times(C-1)<n\lfloor R / n\rfloor \times n\lfloor C / n\rfloor \leq R \times C\).

Q:How can these 3 facts be true simultaneously?
A: The FT of \(\mathbf{J}\) folds over or aliases itself on the DFT of \(\mathbf{J}\) because the DFT is defined on a torus.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Discrete FT is on a Torus}


Recall: the DFT of a digital image assumes the image has a toroidal topology...

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Discrete Fourier Transform is on a Torus}


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Discrete FT is on a Torus}


\section*{Discrete FT is on a Torus}


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Ideal PS of \(2 \times\) Decimated Image}


All of the larger one is

\section*{To Make Actual PS of \(2 \times\) Decimated Image:}



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Actual PS of \(2 \times\) Dec:}


Each of the 4 PS regions forms a torus. The 4 torii are superpositioned onto 1.

\section*{Actual PS of \(2 \times\) Decimated Image}


\section*{DFT Aliasing on the Torus}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{PS of Original and PS of \(2 \times\) Decimated Image}

... lead to jaggies in the decimated image.

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Original Image and \(2 \times\) Decimated Image}


\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Resizing Images

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2011

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Three Methods for Resizing Images}
original image


EECE/CS 253 Image Processing

\section*{Enlarging Images Through Pixel Replication}


Fill in every 4th pixel in every 4th row with the original pixel values.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enlarging Images Through Pixel Replication}


Detail showing every 4th pixel in every 4th row with the original pixel values.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enlarging Images Through Pixel Replication}


For each original value: replicate it 15 times to create a new, larger "pixel".

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Reducing Images Through Pixel Decimation}


\section*{Reducing Images Through Pixel Decimation}


\section*{Reducing Images Through Pixel Decimation}


Example: decimate this image
\(4 x\) to get
this image.


EECE/CS 253 Image Processing

\section*{Reducing Images Through Pixel Decimation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Reducing Images Through Pixel Decimation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Reducing Images Through Pixel Decimation}


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest \\ Neighbor Resampling}

The "Nearest Neighbor" algorithm is a generalization of pixel replication and decimation.

It also includes fractional resizing, i.e. resizing an image so that it has \(p / q\) of the pixels per row and \(p / q\) of the rows in the original. ( \(p\) and \(q\) are both integers.)


Nearest
Neighbor Resampling

Zoom in on a section for a closer look at the process



\author{
Vanderbilt University School of Engineering
}

\section*{Nearest \\ 3/7 resize \\ Neighbor Resampling}


Outlined in blue: \(7 \times 7\) pixel squares

Nearest
Neighbor Resampling


In yellow: 3 pixels for every 7 rows, 3 pixels for every 7 cols.

© 1999-2011 by Richard Alan Peters II

\section*{Nearest 3/7 resize Resampling}
 pixels...

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest \\ Neighbor Resampling}


\section*{3/7 resize}

don't keep the others.

\section*{Nearest \\ Neighbor Resampling}













\section*{3/7 resize}


Copy them into a new image.

\author{
Vanderbilt University School of Engineering
}


Copy them into a new image.

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest 3/7 resize \\ Neighbor Resampling}


Copy them into a new image.

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest 3/7 resize \\ Neighbor Resampling}


Copy them into a new image.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest 3/7 resize \\ Neighbor Resampling}


Copy them into a new image.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest 3/7 resize \\ Neighbor Resampling}



3/7 times the linear dimensions of the original

\author{
Vanderbilt University School of Engineering
}

Nearest
Neighbor Resampling


\author{
Vanderbilt University School of Engineering
}

Nearest Neighbor Resampling

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest Neighbor Resampling}


Resize to 3/7 of
 the original dims.

\section*{Nearest Neighbor Resampling}


\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Nearest} Neighbor Resampling

\section*{7/3 resize}


\section*{Nearest Neighbor Resampling}


\section*{\(3 \times 3\) blocks distributed over 7x7 blocks}


\section*{Nearest Neighbor Resampling}


> Empty pixels filled with color from ULH non-empty pixel


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest \\ Neighbor Resampling}


Empty pixels filled with color from ULH non-empty pixel

7/3 resize


\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest \\ Neighbor Resampling}


Original image


\section*{Nearest Neighbor Resampling}

Size of original image, \(\mathbf{I}: \quad R \times C\) Size of scaled image, \(\mathbf{J}: \quad R^{\prime} \times C^{\prime}\) Row scale factor (input to output):
\[
S_{r}= \begin{cases}R / R^{\prime}, & \text { if } R>R^{\prime}, \\ (R-1) / R^{\prime}, & \text { if } R<R^{\prime},\end{cases}
\]

Column scale factor (input to output):
\[
S_{c}= \begin{cases}C / C^{\prime}, & \text { if } C>C^{\prime}, \\ (C-1) / C^{\prime}, & \text { if } C<C^{\prime}\end{cases}
\]

For each ( \(r^{\prime}, c^{\prime}\) ) in \(\mathbf{J}\), the corresponding fractional pixel location, \(\left(r_{f}, c_{f}\right)\), in I is:
\[
\left(r_{f}, c_{f}\right)=\left(S_{R} \cdot r^{\prime}, S_{C} \cdot c^{\prime}\right)
\]
for \(r^{\prime}=\alpha, \ldots, R^{\prime}\),
\[
c^{\prime}=\beta, \ldots, C^{\prime} .
\]

If \(S_{r} \geq 0.5\) then \(\alpha=1\).
If \(S_{c} \geq 0.5\) then \(\beta=1\).
If \(S_{r}<0.5\) then \(\alpha=\left\lfloor\frac{1}{S_{r}}\right\rfloor\).
If \(S_{c}<0.5\) then \(\beta=\left\lfloor\frac{1}{S_{c}}\right\rfloor\).
The closest integer pixel location ( \(r, c\) ), in I is
\[
(\bar{r}, \bar{c})=\operatorname{round}\left(r_{f}, c_{f}\right)
\]

Then
\[
\mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\mathbf{I}(\bar{r}, \bar{c}) .
\]

\section*{Nearest Neighbor Resampling}

If the output image is
Size of larger than the input Size of image then the scale Row sc factor is less than 1.
\[
S_{r}= \begin{cases}R / R^{\prime}, & \text { if } R>R^{\prime}, \\ (R-1) / R^{\prime}, & \text { if } R<R^{\prime},\end{cases}
\]

Colum If the output image is smaller than the input \(S_{c}\) image then the scale factor is greater than 1.
For each ( \(r^{\prime}, c^{\prime}\) ) in \(\mathbf{J}\), the corresponding fractional pixel location, ( \(r_{f} c_{f}\) ), in I is:
\[
\left(r_{f}, c_{f}\right)=\left(S_{R} \cdot r^{\prime}, S_{C} \cdot c^{\prime}\right)
\]

Which pixel to start with?
\[
\text { for } \begin{aligned}
r^{\prime} & =\alpha, \ldots, R^{\prime}, \\
c^{\prime} & =\beta, \ldots, C^{\prime} . \\
& \text { If } S_{r} \geq 0.5 \text { then } \alpha=1 . \\
& \text { If } S_{c} \geq 0.5 \text { then } \beta=1 . \\
& \text { If } S_{r}<0.5 \text { then } \alpha=\left\lfloor\frac{1}{S_{r}}\right\rfloor . \\
& \text { If } S_{c}<0.5 \text { then } \beta=\left\lfloor\frac{1}{S_{c}}\right\rfloor .
\end{aligned}
\]

The closest integer pixel location ( \(r, c\) ), in I is
\[
(\bar{r}, \bar{c})=\operatorname{round}\left(r_{f}, c_{f}\right) .
\]

Then
\[
\mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\mathbf{I}(\bar{r}, \bar{c}) .
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest Neighbor Resampling}

Size The idea here is that the \(4 \times 4\) Size neighborhood in I of the point \(\left(r_{f}, c_{f}\right)\) has \((\bar{r}, \bar{c})=\left[\left(r_{f}, c_{f},\right)\right]\) as its upper left
Ron corner and has ( \(\bar{r}+1, \bar{c}+1\) ) as its lower right corner.
Thus for each ( \(r_{f}, c_{f}\) ): (1) neither \(\bar{r}\) nor \(\bar{c}\) can be less than one and (2) \(\bar{r}+1\)
Coll cannot be greater than \(R\) and \(\bar{c}+1\) cannot be greater than \(C^{\prime}\).

If the set of all indices \(\{(\bar{r}, \bar{c})\}\) do not satisfy (1) or (2), you must adjust the indices so that they do.
For eacin \((r, c)\), 11 J , me conrespunumg fractional pixel location, \(\left(r_{f}, c_{f}\right)\), in I is:
\[
\left(r_{f}, c_{f}\right)=\left(S_{R} \cdot r^{\prime}, S_{C} \cdot c^{\prime}\right)
\]
\[
\text { for } \begin{aligned}
r^{\prime}= & \alpha, \ldots, R^{\prime}, \\
c^{\prime}= & \beta, \ldots, C^{\prime} . \\
& \text { If } S_{r} \geq 0.5 \text { then } \alpha=1 . \\
& \text { If } S_{c} \geq 0.5 \text { then } \beta=1 . \\
& \text { If } S_{r}<0.5 \text { then } \alpha=\left\lfloor\left.\frac{1}{S_{r}} \right\rvert\, .\right. \\
& \text { If } S_{c}<0.5 \text { then } \beta=\left\lfloor\frac{1}{S_{c}}\right\rfloor .
\end{aligned}
\]

The closest integer pixel location ( \(r, c\) ), in \(I\) is
\[
(\bar{r}, \bar{c})=\operatorname{round}\left(r_{f}, c_{f}\right)
\]

Then
\[
\mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\mathbf{I}(\bar{r}, \bar{c})
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest Neighbor Resampling}

These 4 blocks represent the corners of an image - the extrema of \(r\) and \(c\) in the input image.
\[
r+1=R
\]

\(c=1\)


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Nearest Neighbor Resampling}

Here the image is supersampled with the original pixels in the center. This is not actually none by the algorithm but it helps one visualize the procedure
\[
r+1=R
\]


\(c+1=C\)

\section*{Nearest Neighbor Resampling}

If the output image is smaller than the input image, then \(R^{\prime}<R, C^{\prime}<C\).

\(c=1\)


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering
Nearest Neighbor Resampling


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering
Nearest Neighbor Resampling


\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering
Nearest Neighbor Resampling


\section*{Nearest Neighbor Resampling}


\section*{Nearest Neighbor Resampling}

If the output is smaller than the input, \(R^{\prime}<R\) and \(C^{\prime}<C\).
\[
\left.\begin{array}{l}
(1,1)_{\text {OUT }} \leftrightarrow(1,1)_{\text {IN }} \\
(2,2)_{\text {out }} \leftrightarrow(1,1)_{\text {IN }}+\left(\frac{R}{R^{\prime}}, \frac{C}{C^{\prime}}\right) \\
(n, n)_{\text {out }} \leftrightarrow(1,1)_{\mathbb{N}} \\
(2,2)_{\mathbb{N}}- \\
\text { IN }
\end{array}+(n-1) \frac{R}{R^{\prime}},(n-1) \frac{C}{C^{\prime}}\right), ~ l
\]
lies between \(\left(\rho_{n}+1, \chi_{n}+1\right)\) and \(\left(\rho_{n}+2, \chi_{n}+2\right)\)
\[
\rho_{n}=\left\langle(n-1) \frac{R}{R^{\prime}}\right\rfloor \text { and } \chi_{n}=\left|(n-1) \frac{C}{C^{\prime}}\right|
\]

This and the next 7 slides explain the scale factor selection algebraically

\section*{Nearest Neighbor Resampling}

If the output is smaller than the input, \(R^{\prime}<R\) and \(C^{\prime}<C\).
\((1,1)_{\text {out }} \leftrightarrow(1,1)_{\text {IN }}\)
\((2,2)_{\text {out }} \leftrightarrow(1,1)_{\text {IN }}+\left(\frac{R}{R^{\prime}}, \frac{C}{C^{\prime}}\right)\)
\((n, n)_{\text {out }} \leftrightarrow(1,1)_{\text {IN }}+\left((n-1) \frac{R}{R^{\prime}},(n-1) \frac{C}{C^{\prime}}\right)\)

lies between \(\left(\rho_{n}+1, \chi_{n}+1\right)\) and \(\left(\rho_{n}+2, \chi_{n}+2\right)\)
\((2,2)_{\text {out }}\) lies between \((2,2)_{\text {IN }}\) and \((3,3)_{\text {IN }}\) since \(R / R^{\prime}>1 \& C / C^{\prime}>1\).
\[
\rho_{n}=\left\langle(n-1) \frac{R}{R^{\prime}}\right\rfloor \text { and } \chi_{n}=\left|(n-1) \frac{C}{C^{\prime}}\right|
\]

\section*{Nearest Neighbor Resampling}

If the output is smaller than the input, \(R^{\prime}<R\) and \(C^{\prime}<C\).
\(\left(R^{\prime}, C^{\prime}\right)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\left(R^{\prime}-1\right) \frac{R}{R^{\prime}},\left(C^{\prime}-1\right) \frac{C}{C^{\prime}}\right)\)
but \(R^{\prime}=\alpha R\) and \(C^{\prime}=\beta C\) where \(\alpha<1\) and \(\beta<1\).
\(\rho_{R}=\left\lfloor\left(R^{\prime}-1\right) \frac{R}{R^{\prime}}\right\rfloor=\left\lfloor(\alpha R-1) \frac{R}{\alpha R}\right\rfloor=\left\lfloor\frac{1}{\alpha}(\alpha R-1)\right\rfloor=\left\lfloor R-\frac{1}{\alpha}\right\rfloor=R-2\).
Similarly, \(\quad \chi_{C}=C-2\).
Thus \(\left(R^{\prime}, C^{\prime}\right)_{\text {out }}\) lies between
\((R-1, C-1)_{\mathrm{IN}}\) and \((R, C)_{\mathrm{IN}}\).
\((R-1, C-1)_{\mathbb{I N}^{N}}\)
\begin{tabular}{ccc|c}
\(\lrcorner\) & \(\lrcorner\) & \(\lrcorner\) \\
\(\lrcorner\) & \(\lrcorner\) & \(\lrcorner\) & \\
\hline & \(\lrcorner\) & \(\lrcorner\) & \((R, C)_{\mathbb{N}}\) \\
\hline
\end{tabular}

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest Neighbor Resampling}

If the output is smaller than the input, \(R^{\prime}<R\) and \(C^{\prime}<C\).
\(\left(R^{\prime}, C^{\prime}\right)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\left(R^{\prime}-1\right) \frac{R}{R^{\prime}},\left(C^{\prime}-1\right) \frac{C}{C^{\prime}}\right)\)
but \(R^{\prime}=\alpha R\) and \(C^{\prime}=\beta C\) where \(\alpha<1\) and \(\beta<1\).
\(\rho_{R}=\left\lfloor\left(R^{\prime}-1\right) \frac{R}{R^{\prime}}\right\rfloor=\left\lfloor(\alpha R-1) \frac{R}{\alpha R}\right\rfloor=\left\lfloor\frac{1}{\alpha}(\alpha R-1)\right\rfloor=\left\lfloor R-\frac{1}{\alpha}\right\rfloor=R-2\).
Similarly, \(\quad \chi_{C}=C-2\).
Thus \(\left(R^{\prime}, C^{\prime}\right)_{\text {OUt }}\) lies between
\((R-1, C-1)_{\mathrm{IN}}\) and \((R, C)_{\mathrm{IN}}\).
\((R-1, C-1)_{\mathbb{I N}^{N}}\)


\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest Neighbor Resampling}

If the output is larger than the input, \(R<R^{\prime}\) and \(C<C^{\prime}\).
\((1,1)_{\text {OUT }} \leftrightarrow(1,1)_{\text {IN }}\)
\((2,2)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\frac{R-1}{R^{\prime}}, \frac{C-1}{C^{\prime}}\right)\)
\((n, n)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left((n-1) \frac{R-1}{R^{\prime}},(n-1) \frac{C-1}{C^{\prime}}\right)\)

lies between \(\left(\rho_{n}, \chi_{n}\right)\) and \(\left(\rho_{n}+1, \chi_{n}+1\right)\)
\(\rho_{n}=\left\lfloor(n-1) \frac{R-1}{R^{\prime}}\right\rfloor\) and \(\chi_{n}=\left\lfloor(n-1) \frac{C-1}{C^{\prime}}\right\rfloor\)

\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Nearest Neighbor Resampling}

If the output is larger than the input, \(R<R^{\prime}\) and \(C<C^{\prime}\).
\((1,1)_{\text {OUT }} \leftrightarrow(1,1)_{\text {IN }}\)
\((2,2)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\frac{R-1}{R^{\prime}}, \frac{C-1}{C^{\prime}}\right)\)
\((n, n)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left((n-1) \frac{R-1}{R^{\prime}},(n-1) \frac{C-1}{C^{\prime}}\right)\)
lies between \(\left(\rho_{n}, \chi_{n}\right)\) and \(\left(\rho_{n}+1, \chi_{n}+1\right)\)
\((2,2)_{\text {out }}\) lies between
\((1,1)_{\text {IN }}\) and \((2,2)_{\text {IN }}\) since \((R-1) / R^{\prime}<1\).
\(\rho_{n}=\left\lfloor(n-1) \frac{R-1}{R^{\prime}}\right\rfloor\) and \(\chi_{n}=\left\lfloor(n-1) \frac{C-1}{C^{\prime}}\right\rfloor\)

\section*{Nearest Neighbor Resampling}

If the output is larger than the input, \(R<R^{\prime}\) and \(C<C^{\prime}\).
\(\left(R^{\prime}, C^{\prime}\right)_{\mathrm{OUT}} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\left(R^{\prime}-1\right) \frac{R-1}{R^{\prime}},(C-1) \frac{R-1}{C^{\prime}}\right)\)
but \(R^{\prime}=\alpha R\) and \(C^{\prime}=\beta C\) where \(\alpha>1\) and \(\beta>1\).
\(\rho_{R}=\left\lfloor\left(R^{\prime}-1\right) \frac{R-1}{R^{\prime}}\right\rfloor=\left\lfloor(\alpha R-1) \frac{R-1}{\alpha R}\right\rfloor=\left\lfloor R-1-\frac{1}{\alpha}+\frac{1}{\alpha R}\right\rfloor=R-2\).
Similarly, \(\quad \chi_{C}=C-2\).
Thus \(\left(R^{\prime}, C^{\prime}\right)_{\text {Out }}\) lies between
\((R-1, C-1)_{\mathrm{IN}}\) and \((R, C)_{\mathrm{IN}}\).
\begin{tabular}{|c|c|c|}
\hline - & ل & - \\
\hline - & ل & - \\
\hline - & - & - \\
\hline
\end{tabular}

\section*{Nearest Neighbor Resampling}

If the output is larger than the input, \(R<R^{\prime}\) and \(C<C^{\prime}\).
\(\left(R^{\prime}, C^{\prime}\right)_{\text {OUT }} \leftrightarrow(1,1)_{\mathrm{IN}}+\left(\left(R^{\prime}-1\right) \frac{R-1}{R^{\prime}},(C-1) \frac{R-1}{C^{\prime}}\right)\)
but \(R^{\prime}=\alpha R\) and \(C^{\prime}=\beta C\) where \(\alpha>1\) and \(\beta>1\).
\(\rho_{R}=\left\lfloor\left(R^{\prime}-1\right) \frac{R-1}{R^{\prime}}\right\rfloor=\left\lfloor(\alpha R-1) \frac{R-1}{\alpha R}\right\rfloor=\left\lfloor R-1-\frac{1}{\alpha}+\frac{1}{\alpha R}\right\rfloor=R-2\).
Similarly, \(\quad \chi_{C}=C-2\).
Thus \(\left(R^{\prime}, C^{\prime}\right)_{\text {out }}\) lies between
\((R-1, C-1)_{\mathrm{IN}}\) and \((R, C)_{\mathrm{IN}}\).


\section*{Enlarging Images: Replication vs. Interpolation}


> Pixel replication produces a "jagged" result since each individual square pixel is made larger.

Bilinear interpolation creates new pixels that have values intermediate between the originals. The result is smoother but blurry.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Pixel Replication}

Red dots mark original pixel values.


Small square pixels become large square pixels.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Bilinear Interpolation}

Red dots mark original pixel values.


Intermediate locations are filled with intermediate values.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

\author{
want to \\ upsample this \\ image by a factor of two
}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation} new samples to be added at light gray locations.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}
treat gray levels as heights above the image plane

center = weighted average of four corners

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}
treat gray levels as heights above the image plane
center = weighted average of four corners

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}
the results


\section*{Bilinear Interpolation Example}


We'll enlarge this image by a factor of 4 ..
... via bilinear interpolation and compare it to a neares \(\dagger\) neighbor enlargement.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Bilinear Interpolation - 4× Zoom}


To better see what happens, we'll look at the parrot's eye.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Bilinear Interpolation \(-4 \times\) Zoom}


To better see what happens, we'll look at the parrot's eye.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Bilinear Interpolation}

For a \(4 \times\) zoom, create a blank image, four times the size of the original.

\section*{Bilinear Interpolation}

Then fill in every 4th pixel in every 4th row with the original values.



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Bilinear} Interpolation

Then fill in every 4th pixel in every 4th row with the original values.


\section*{Bilinear Interpolation}

Then fill in every 4th pixel in every 4th row with the original values.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Bilinear Interpolation}

Next you want to fill in the other 15 pixels in each block with the intermediate values.


\author{
Vanderbilt University School of Engineering
}

\section*{Bilinear Interpolation}

Next you want to fill in the other 15 pixels in each block with the intermediate values.


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Bilinear} Interpolation

Next you want to fill in the other 15 pixels in each block with the intermediate values.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Bilinear Interpolation}

The result:


Compare to the next slide which contains a \(4 x\) pixel zoom via pixel replication.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Pixel Replication}

The result:


Compare to the prev. slide which contains a \(4 \times\) pixel zoom via bilinear interp.

\section*{Bilinear Interpolation}

Result with original pixels marked:


\section*{Pixel Replication}

Result with original pixels marked:


\section*{Pixel Replication vs. Bilinear Interpolation}


Pixel replication


Bilinear interpolation

\section*{Pixel Replication vs. Bilinear Interpolation}


Pixel replication


Bilinear interpolation

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

Example: reduce the cactus image to \(3 / 7\) its original size using bilinear interpolation


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

For each \(7 \times 7\) block of pixels select \(3 \times 3=9\) pixel locations.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

For neares \(\dagger\) neighbor sampling the 9 pixel locations correspond to pixel locations in the original image.


Nearest neighbor selected pixels outlined in yellow.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

In bilinear interpolation the 9 pixel locations are distributed evenly.


Nearest neighbor selected pixels outlined in yellow.

Bilinear interp. pixels locations outlined in red.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

In bilinear interpolation the 9 pixel locations are distributed evenly.


Nearest neighbor selected pixels outlined in yellow.

Bilinear interp. pixels locations outlined in red.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

In bilinear interpolation the 9 pixel locations are distributed evenly.


Notice that the locations overlap pixels in the original image.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

For each \(7 \times 7\) block of pixels select \(3 \times 3=9\) pixel locations.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

Examine one section in detail.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


\section*{Resampling Through Bilinear Interpolation}

The blue square is the output location.


Shrink the pixels for visualization of this example.


\section*{Resampling Through Bilinear Interpolation}


The blue square is the output location.


\section*{Resampling Through Bilinear Interpolation}


The blue square is the output location.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

locs \& colors of new pixels

locations of new pixels

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}

locs \& colors of new pixels

locs \& colors of new pixels

\section*{Resampling Through Bilinear Interpolation}


New image from new pixels

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Resampling Through Bilinear Interpolation}


\section*{Resampling Through Bilinear Interpolation}

Let \(\mathbf{I}\) be an \(R \times C\) image.
We want to resize I to \(R^{\prime} \times C^{\prime}\).
Call the new image \(\mathbf{J}\).
Let \(s_{R}=R / R^{\prime}\) and \(s_{C}=C / C^{\prime}\).
Let \(r_{f}=r^{\prime} \cdot s_{R}\) for \(r^{\prime}=1, \ldots, R^{\prime}\) and \(c_{f}=c^{\prime} \cdot s_{C}\) for \(c^{\prime}=1, \ldots, C^{\prime}\).
Let \(r=\left\lfloor r_{f}\right\rfloor\) and \(c=\left\lfloor c_{f}\right\rfloor\).
Let \(\Delta r=r_{f}-r\) and \(\Delta c=c_{f}-c\).
Then \(\mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\mathbf{I}(r, c) \cdot(1-\Delta r) \cdot(1-\Delta c)\)
\[
\begin{aligned}
& +\mathbf{I}(r+1, c) \cdot \Delta r \cdot(1-\Delta c) \\
& +\mathbf{I}(r, c+1) \cdot(1-\Delta r) \cdot \Delta c \\
& +\mathbf{I}(r+1, c+1) \cdot \Delta r \cdot \Delta c
\end{aligned}
\]


For each pixel ( \(r^{\prime}, c^{\prime}\) ) in output image, \(J\), compute the fractional

\section*{Resampling Through Bilinear In}
\[
\text { location }\left(r_{f}, c_{f}\right) \text { in } I \text {. Use }(r, c) \text {, the }
\]
\(\qquad\) integer part of \(\left(r_{f}, c_{f}\right)\), to find the 4 neighboring locations in I. Compute \(J\left(r^{\prime}, c^{\prime}\right)\) from a weighted

Let It scale factors
We wi ulu lestize \(10 \mathrm{ut} \times C^{\prime}\).
Call tt !new image \(\mathbf{J}\).
Let \(s_{R}=R / R^{\prime}\) and \(s_{C}=C / C^{\prime}\).
Let \(r_{f}=r^{\prime} \cdot s_{R}\) for \(r^{\prime}=1, \ldots, R^{\prime}\) and \(c_{f}=c^{\prime} \cdot s_{C}\) for \(c^{\prime}=1, \ldots, C^{\prime}\).
Let \(r=\left\lfloor r_{f}\right\rfloor\) and \(c=\left\lfloor c_{f}\right\rfloor\).
Let \(\Delta r=r_{f}-r\) and \(\Delta c=c_{f}-c\).
Then \(\mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\mathbf{I}(r, c) \cdot(1-\Delta r) \cdot(1-\Delta c)\)
\[
\begin{aligned}
& +\mathbf{I}(r+1, c) \cdot \Delta r \cdot(1-\Delta c) \\
& +\mathbf{I}(r, c+1) \cdot(1-\Delta r) \cdot \Delta c \\
& +\mathbf{I}(r+1, c+1) \cdot \Delta r \cdot \Delta c
\end{aligned}
\]

Fractional location ( \(r_{f}, c_{f}\) ) in I of pixel ( \(r^{\prime}, c^{\prime}\) ) in \(J\).

\[
\text { Integer part of }\left(r_{f}, c_{f}\right) \text {. }
\]


The location of \(\left(r_{f}, c_{f}\right)\) as fractions of the distances between ( \(r, c\) ) and its 3 neighbors.

\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Resampling Through Bilinear Interpolation}

Size of original image: \(R \times C\)
Size of scaled image: \(\quad R^{\prime} \times C^{\prime}\)
Row scale factor:
\[
S_{r}=\left\{\begin{array}{lll}
R / R^{\prime}, & \text { if } \quad R>R^{\prime} \\
(R-1) / R^{\prime}, & \text { if } \quad R<R^{\prime}
\end{array}\right.
\]

Column scale factor:
\[
S_{c}= \begin{cases}C / C^{\prime}, & \text { if } \quad C>C^{\prime} \\ (C-1) / C^{\prime}, & \text { if } R<C^{\prime}\end{cases}
\]
\(\left(r_{f}, c_{f}\right)\) is the fractional location in the input image from which to sample the output pixel ( \(r, c\) ).
\[
r_{f}=\left[\left(1, \ldots, R^{\prime}\right) \cdot S_{r}\right], \quad c_{f}=\left[\left(1, \ldots, C^{\prime}\right) \cdot S_{c}\right]
\]
( \(r, c\) ) are the row and column indices of the pixels in the input image to use in the algorithm.
\[
\left.(r, c)=\left(\left\lfloor r_{f}\right\rfloor, \mid c_{f}\right\rfloor\right)
\]
( \(\Delta r, \Delta c\) ) are the fractional parts of the row and column locations, \(\left(r_{f}, c_{f}\right)\).
\[
(\Delta r, \Delta c)=\left(r_{f}-r, c_{f}-c\right)
\]

Then the value of each output pixel is given by
\[
\begin{aligned}
\mathbf{J}\left(r^{\prime}, c^{\prime}\right)= & \mathbf{I}(r, c) \cdot(1-\Delta r) \cdot(1-\Delta c) \\
& +\mathbf{I}(r+1, c) \cdot \Delta r \cdot(1-\Delta c) \\
& +\mathbf{I}(r, c+1) \cdot(1-\Delta r) \cdot \Delta c \\
& +\mathbf{I}(r+1, c+1) \cdot \Delta r \cdot \Delta c .
\end{aligned}
\]

\section*{Resampling Through Bilinear Interpolation}

Size of original image: \(R \times C\)
Like nearest neighbor resampling the \(4 \times 4\) neighborhood in \(I\) of the point \(\left(r_{f}, c_{f}\right)\) has ( \(r, c\) ) \(=\left\lfloor\left(r_{f}, c_{f}\right)\right\rfloor\) as its upper left corner and has ( \(r+1, c+1\) ) as its lower right corner.
Thus for each ( \(r_{f}, c_{f}\) ): (1) neither \(r\) nor \(c\) can be less than one and (2) \(r+1\) cannot be greater than \(R\) and \(c+1\) cannot be greater than \(C\).

If the set of all indices \(\{(r, c)\}\) do not satisfy (1) or (2), you must adjust the indices so that they do.
\[
\begin{aligned}
& \text { pixel }(r, c) \\
& r_{f}=\left[\left(1, \ldots, R^{\prime}\right) \cdot S_{r}\right], \quad c_{f}=\left[\left(1, \ldots, C^{\prime}\right) \cdot S_{c}\right]
\end{aligned}
\]
( \(r, c\) ) are the row and column indices of the pixels in the input image to use in the algorithm.
\[
(r, c)=\left(\left\lfloor r_{f}\right\rfloor,\left\lfloor c_{f}\right\rfloor\right)
\]
( \(\Delta r, \Delta c\) ) are the fractional parts of the row and column locations, \(\left(r_{f}, c_{f}\right)\).
\[
(\Delta r, \Delta c)=\left(r_{f}-r, c_{f}-c\right)
\]

Then the value of each output pixel is given by
\[
\begin{aligned}
\mathbf{J}\left(r^{\prime}, c^{\prime}\right)= & \mathbf{I}(r, c) \cdot(1-\Delta r) \cdot(1-\Delta c) \\
& +\mathbf{I}(r+1, c) \cdot \Delta r \cdot(1-\Delta c) \\
& +\mathbf{I}(r, c+1) \cdot(1-\Delta r) \cdot \Delta c \\
& +\mathbf{I}(r+1, c+1) \cdot \Delta r \cdot \Delta c .
\end{aligned}
\]

\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Bilinear Interpolation}


5:7


1:1


11:7

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Bi-Interp Example: resize to 5/7 of original dimensions.}


\section*{Resampling Through Bicubic Interpolation}

Bilinear interpolation computes a value for \(\mathbf{J}\left(r^{\prime}, c^{\prime}\right)\) as the weighted combination of \(\mathbf{I}(r, c)\), \(\mathbf{I}(r+1, c), \mathbf{I}(r+1, c+1)\), and \(\mathbf{I}(r, c+1)\).

Bicubic interpolation uses not only those 4 input image pixels, but also their partial derivatives:
\[
\begin{aligned}
& \left\{\left\{\frac{\partial}{\partial c^{\prime}} \mathbf{I}(r+i, c+j),\right.\right. \\
& \quad \frac{\partial}{\partial r^{\prime}} \mathbf{I}(r+i, c+j), \\
& \left.\left.\frac{\partial}{\partial c^{\prime} r^{\prime}} \mathbf{I}(r+i, c+j)\right\}_{j=0}^{1}\right\}_{i=0}^{1}
\end{aligned}
\]
\begin{tabular}{|l|l|l|l|}
\hline\((r-1, c-1)\) & \((r-1, c)\) & \((r-1, c+1)\) & \((r-1, c+2)\) \\
\hline\((r, c-1)\) & \((r, c)\) & \((r, c+1)\) & \((r, c+2)\) \\
\hline\((r+1, c-1)\) & \((r+1, c)\) & \((r+1, c+1)\) & \((r+1, c+2)\) \\
\hline\((r+2, c-1)\) & \((r+2, c)\) & \((r+2, c+1)\) & \((r+2, c+2)\) \\
\hline & & & \\
\hline
\end{tabular}

Since the derivatives are are computed digitally on the 8 neighborhoods of the 4 pixel locations, a \(4 \times 4\) neighborhood of the input image is needed for each output value.

\section*{Resampling Through Bicubic Interpolation}
\(\mathcal{N}(r, c)\) is the 8 -pixel neighborhood of \((r, c)\).
\[
\begin{aligned}
\frac{\partial}{\partial r} \mathbf{I}(r, c)= & \text { avg. of forward diff and } \\
& \text { backward diff @ }(r, c) \\
= & \frac{1}{2}[(\mathbf{I}(r+1, c)-\mathbf{I}(r, c)) \\
& +(\mathbf{I}(r, c)-\mathbf{I}(r-1, c))] \\
= & \frac{1}{2}[\mathbf{I}(r+1, c)-\mathbf{I}(r-1, c)]
\end{aligned}
\]
\begin{tabular}{|c|c|c|c|}
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\boxplus\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline
\end{tabular}
\(\mathcal{N}(r, c)\)

\(\mathcal{N}(r+1, c)\)

\(\mathcal{N}(r, c+1)\)
\begin{tabular}{|c|c|c|c|}
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\boxplus\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\boxed{ }\) \\
\hline\(\square\) & \(\square\) & \(\boxed{ }\) & \(\square\) \\
\hline \multicolumn{5}{|c|}{\(\mathcal{N}(r+1, c+1)\)}
\end{tabular}

\section*{Resampling Through Bicubic Interpolation}
\[
\begin{aligned}
& \hline \mathcal{N}(r, c) \\
& \hline \begin{aligned}
& \frac{\partial}{\partial r} \mathbf{I}(r, c)= \frac{1}{2}[\mathbf{I}(r+1, c)-\mathbf{I}(r-1, c)], \\
& \\
& \hline \frac{\partial}{\partial c} \mathbf{I}(r, c)=\frac{1}{2}[\mathbf{I}(r, c+1)-\mathbf{I}(r, c-1)], \\
& \hline
\end{aligned} \\
& \hline \begin{array}{r}
\frac{\partial}{\partial c r} \mathbf{I}(r, c)=\frac{1}{4}[\mathbf{I}(r+1, c+1)-\mathbf{I}(r+1, c-1) \\
\\
\\
\quad \mathbf{I}(r-1, c-1)-\mathbf{I}(r-1, c+1)]
\end{array}
\end{aligned}
\]


\section*{Resampling Through Bicubic Interpolation}
\[
\left.\begin{array}{l}
\mathcal{N}(r, c+1) \\
\hline \frac{\partial}{\partial r} \mathbf{I}(r, c+1)=\frac{1}{2}[\mathbf{I}(r+1, c+1)-\mathbf{I}(r-1, c+1)], \\
\hline
\end{array} \begin{array}{r}
\frac{\partial}{\partial c} \mathbf{I}(r, c+1)=\frac{1}{2}[\mathbf{I}(r, c+2)-\mathbf{I}(r, c)],
\end{array}\right] \begin{array}{r}
\begin{array}{r}
\frac{\partial}{\partial c r} \mathbf{I}(r, c+1)=\frac{1}{4}[\mathbf{I}(r+1, c+2)-\mathbf{I}(r+1, c) \\
+\mathbf{I}(r-1, c)-\mathbf{I}(r-1, c+2)]
\end{array}
\end{array}
\]


\section*{Resampling Through Bicubic Interpolation}
\[
\begin{gathered}
\mathcal{N}(r+1, c+1) \\
\hline \frac{\partial}{\partial r} \mathbf{I}(r+1, c+1)=\frac{1}{2}[\mathbf{I}(r+2, c+1)-\mathbf{I}(r, c+1)], \\
\hline \frac{\partial}{\partial c} \mathbf{I}(r+1, c+1)=\frac{1}{2}[\mathbf{I}(r+1, c+2)-\mathbf{I}(r+1, c)], \\
\hline
\end{gathered}
\]
\begin{tabular}{|l|l|l|l|}
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline\(\square\) & \(\square\) & \(\square\) & \(\square\) \\
\hline
\end{tabular}

\section*{Resampling Through Bicubic Interpolation}
\[
\begin{aligned}
& \mathcal{N}(r+1, c) \\
& \begin{array}{l}
\frac{\partial}{\partial r} \mathbf{I}(r+1, c)=\frac{1}{2}[\mathbf{I}(r+2, c)-\mathbf{I}(r, c)],
\end{array} \\
& \hline \begin{array}{r}
\frac{\partial}{\partial c} \mathbf{I}(r+1, c)=\frac{1}{2}[\mathbf{I}(r+1, c+1)-\mathbf{I}(r+1, c-1)],
\end{array} \\
& \hline \begin{array}{r}
\frac{\partial}{\partial c r} \mathbf{I}(r+1, c)=\frac{1}{4}[\mathbf{I}(r+2, c+1)-\mathbf{I}(r+2, c-1) \\
+\mathbf{I}(r, c-1)-\mathbf{I}(r, c+1)]
\end{array}
\end{aligned}
\]


\section*{Resampling Through Bicubic Interpolation}
\[
\left.\left.\begin{array}{l}
S_{r}= \begin{cases}R / R^{\prime}, & \text { if } R>R^{\prime}, \\
(R-1) / R^{\prime}, & \text { if } R<R^{\prime} .\end{cases} \\
S_{c}= \begin{cases}C / C^{\prime}, & \text { if } C>C^{\prime}, \\
(C-1) / C^{\prime}, & \text { if } \\
C<C^{\prime} .\end{cases} \\
r_{f}=\left[\left(1, \ldots, R^{\prime}\right) \cdot S_{r}\right], c_{f}=\left[\left(1, \ldots, C^{\prime}\right) \cdot S_{c}\right]
\end{array} \right\rvert\, \begin{array}{ll}
\text { This part is the } \\
\text { same as NN }
\end{array}\right\}
\]


\section*{Resampling Through Bicubic Interpolation}
\[
\begin{aligned}
& \mathbf{J}\left(r^{\prime}, c^{\prime}\right)=\sum_{m=-1}^{2} \sum_{n=-1}^{2} \mathbf{I}(r+m, c+n) P(d r-m) P(n-d c) \\
& P(x)=\frac{1}{6}\left[Q(x+2)^{3}-4 Q(x+1)^{3}-\right. \\
& \left.6 Q(x)^{3}-4 Q(x-1)^{3}\right] \\
& Q(x)= \begin{cases}x \text { for } x>0 & \text { The differentials } \\
0 \text { for } x \leq 0 & \begin{array}{l}
\text { on pp 103-107 are } \\
\text { combined in this } \\
\text { sum of products } \\
\text { of polynomials. }
\end{array}\end{cases}
\end{aligned}
\]


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7}

original



ग!qno!̣

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7}

original

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7 (zoomed)}

nearest neighbor

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7 (zoomed)}

bilinear

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7 (zoomed)}

bicubic

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Size Reduction 3/7}

original

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enlargement}

nearest neighbor 7/3


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Enlargement}

bilinear 7/3


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Enlargement}

bicubic 7/3


\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Warping and Rotating Images

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2011

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Geometric Remapping}


\section*{Geometric Remapping}
1. Assume the input image, \(\mathbf{I}\), has infinite spatial resolution.
2. Calculate the size, \(R_{\text {out }} \times C_{\text {out }} \times B\), of the output image, \(\mathbf{J}\), and allocate it.
3. Create an image map (a warping function, \(\Phi\) ) as follows:
a) Allocate an \(R_{\text {out }} \times C_{\text {out }} \times 2\) array, \(\Phi\).
b) For every pixel location ( \(r, c\) ) in \(\mathbf{J}\) find the corresponding real-valued pixel location \(\left(r_{f}, c_{f}\right)\) in \(\mathbf{I}\).
c) Set \(\Phi(r, c, 1)=r_{f}\) and set \(\Phi(r, c, 2)=c_{f}\).
4. Create an interpolation function, \(\Theta\), that generates a pixel

5. Then set \(\mathbf{J}(r, c)=\Theta\left\{\mathbf{I} ; \mathscr{T}\left(r_{f}, c_{f}\right)\right\}\).

\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Linear Warping of Images}


Original image with perspective distortion.

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}

\section*{Linear Warping of Images}


Image warped to correct perspective distortion.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Linear Warping of Images}

Select at least 4 key
points that are easy to correct.

E.g. in this case, the lines should be perfectly vertical.


Selected correction points.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Linear Warping of Images}
Here, align the top
points with the
bottom points.


Target correction points.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Linear Warping of Images}

The top part of the image has been stretched more than the bottom


Result of linear LMS point remapping.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Linear Warping of Images}

The top part of the image has been stretched more than the bottom


Image warped to correct perspective distortion.

\title{
EECE/CS 253 Image Processing
}

\section*{Linear Warping of Images - How Not To Do It}

Given a set, \(\mathbf{X}\), of points from image \(\mathbf{I}\) and a set, \(\mathbf{Y}\), of target points find \(\mathbf{H}\) such that:
\[
\mathbf{Y}=\mathbf{H} \mathbf{X} \quad \text { where }
\]
\[
\begin{aligned}
& \mathbf{X}=\left[\begin{array}{llll}
\mathbf{x}_{1} & \mathbf{x}_{2} & \cdots & \mathbf{x}_{p}
\end{array}\right]=\left[\begin{array}{llll}
x_{1} & x_{2} & \cdots & x_{p} \\
y_{1} & y_{2} & \cdots & y_{p}
\end{array}\right] \\
& \mathbf{Y}=\left[\begin{array}{llll}
\mathbf{y}_{1} & \mathbf{y}_{2} & \cdots & \mathbf{y}_{p}
\end{array}\right]=\left[\begin{array}{llll}
u_{1} & u_{2} & \cdots & u_{p} \\
v_{1} & v_{2} & \cdots & v_{p}
\end{array}\right]
\end{aligned} \text { and } \mathbf{H}=\left[\begin{array}{ll}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}\right],
\]

\section*{Linear Warping of Images - How Not To Do It}

Given a set, \(\mathbf{X}\), of points from image I and a set, \(\mathbf{Y}\), of target points find \(\mathbf{H}\) such that:
\[
\mathbf{Y}=\mathbf{H X}
\]

The \(\mathbf{H}\) that minimizes the square of the difference between \(\mathbf{Y}\) and \(\mathbf{H X}\) is:
\[
\mathbf{H}=\mathbf{Y} \mathbf{X}^{\top}\left(\mathbf{X} \mathbf{X}^{\top}\right)^{-1}
\]

Then for each pixel in the output image, select the corresponding input pixel:
\[
\mathbf{J}(r, c)=\mathbf{I}\left(r_{h}, c_{h}\right),
\]
where
\[
\left[\begin{array}{l}
r_{h} \\
c_{h}
\end{array}\right]=\left\lfloor\left[\begin{array}{l}
0.5 \\
0.5
\end{array}\right]+\mathbf{H}^{-1}\left[\begin{array}{l}
r \\
c
\end{array}\right]\right\rfloor .
\]

\section*{EECE/CS 253 Image Processing}

\section*{Not \\ Linear Warping of Images - How To Do It}

Example:
\[
\begin{array}{ll}
\mathbf{X}=\left[\begin{array}{rrrrrr}
52 & 80 & 403 & 412 & 913 & 872 \\
632 & 326 & 652 & 34 & 624 & 239
\end{array}\right] & \mathbf{H}=\left[\begin{array}{rrr}
1.0285 & -0.0228 \\
0 & 1
\end{array}\right] \\
\mathbf{Y}=\left[\begin{array}{rrrrrrr}
52 & 52 & 403 & 403 & 913 & 913 \\
632 & 326 & 652 & 34 & 624 & 239
\end{array}\right] & \mathbf{H}^{-1}=\left[\begin{array}{ccc}
0.9723 & 0.0221 \\
0 & 1
\end{array}\right] \\
\mathbf{H X}=\left[\begin{array}{rrrrrr}
39 & 75 & 400 & 423 & 925 & 891 \\
632 & 326 & 652 & 34 & 624 & 239
\end{array}\right] & \begin{array}{l}
\text { Next 2 images: } \\
\text { p. 13 original; } \\
\text { p. 14 result of warping } \\
\text { the image with the H } \\
\text { above. }
\end{array}
\end{array}
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Linear Warping of Images - How \({ }_{\text {No }}\) To Do It}


Original image with distorted (red) and undistorted (blue) lines.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Not \\ Linear Warping of Images - How/To Do It}

This is pretty bad! Why?
Because the stretch that we want to perform is not linear in 2D.


Warped image with distorted (red) and undistorted (blue) lines.

\section*{Linear Warping of Images - How Not To Do It}

Multiplication of a vector by an arbitrary matrix rotates the vector, stretches the result along its coordinates and rotates the result of that.
\[
\mathbf{H}=\mathbf{U S V}^{\top}
\]
where \(\mathbf{U}\) and \(\mathbf{V}\) are rotations and \(\mathbf{S}\) is diagonal. For the example:
\[
\begin{aligned}
& \mathbf{V}^{\top}=\left[\begin{array}{rr}
0.9419 & -0.3359 \\
0.3359 & 0.9419
\end{array}\right] \quad \text { CW rotation } 19.6293^{\circ} \quad
\end{aligned} \begin{aligned}
& \text { Recall: } \\
& \mathbf{H}=\left[\begin{array}{cc}
1.0285 & -0.0228 \\
0 & 1
\end{array}\right] \\
& \mathbf{U}=\left[\begin{array}{rr}
0.9456 & 0.3253 \\
-0.3253 & 0.9456
\end{array}\right] \quad \text { CCW rotation }-18.9863^{\circ}
\end{aligned}
\]

\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Not \\ Linear Warping of Images - How/To Do It}


Original image rotated by \(\mathbf{V}^{\top}\).

\author{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Not \\ Linear Warping of Images - How \({ }_{\text {T }}\) To Do It}


Original image rotated by \(\mathbf{V}^{\boldsymbol{\top}}\) and stretched by \(\mathbf{S}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Not \\ Linear Warping of Images - How/To Do It}


Original image rotated by \(\mathbf{V}^{\top}\), stretched by \(\mathbf{S}\), and rotated by \(\mathbf{U}\).

\section*{Linear Warping of Images - How Not To Do It}

Given a set, \(X\), of points from image \(I\) and a set, \(Y\), of target p There is a problem, however. This does not work very well. \(\mathbf{X}=\left[\begin{array}{llll}\mathbf{x}_{1} & \mathbf{x}_{2} & \ldots\end{array}\right.\) This is a 2D linear transform. What we need is a 3D affine transform. A \(3^{\text {rd }}\) dimension
\(\mathbf{Y}=\left[\begin{array}{lll}\mathbf{y}_{1} & \mathbf{y}_{2} & \text {. }\end{array}\right.\) enables us to model more ge-
\[
\mathbf{I}=\left[\begin{array}{ll}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}\right]
\] neral projections. Thus we use 3D homogeneous coordinates.
\(\mathbf{H}^{-1}\left[\begin{array}{l}c_{\mathbf{J}} \\ r_{\mathbf{J}}\end{array}\right]=\left[\begin{array}{l}c_{\mathbf{I}} \\ r_{\mathbf{I}}\end{array}\right]\) then \(\mathbf{J}\left(r_{\mathbf{J}}, c_{\mathbf{J}}\right)=\mathbf{I}\left(\mathbf{H}^{-1}\left[\begin{array}{l}c_{\mathbf{J}} \\ r_{\mathbf{J}}\end{array}\right]\right) \begin{aligned} & \text { is the warped } \\ & \text { image. }\end{aligned}\)

\section*{Linear Warping of Images - Homogeneous Cdts.}
\[
\begin{gathered}
\mathbf{x}_{i}=\left[\begin{array}{l}
x_{i} \\
y_{i} \\
1
\end{array}\right] \quad \begin{array}{l}
\text { Place a } 1 \text { in the 3rd } \\
\text { dimension of each } \\
\text { input pixel location. }
\end{array} \\
\mathbf{y}_{i}=\mathbf{H x}_{i} \quad \begin{array}{l}
\mathrm{H} \text { is } \\
3 \times 3 .
\end{array} \\
\mathbf{y}_{i}=\left[\begin{array}{c}
k_{i} u_{i} \\
k_{i} v_{i} \\
k_{i}
\end{array}\right] \begin{array}{l}
\text { The remapped pixel locs are } \\
\begin{array}{l}
\text { also } 30 . \text { Call each 3rd element } \\
k_{i} \text { and write the other two as } \\
\text { proportional to } \mathrm{k}_{\mathrm{i}} .
\end{array}
\end{array}
\end{gathered}
\]

\section*{Linear Warping of Images - Homogeneous Cdts.}
\[
\mathbf{X}=\left[\begin{array}{cccc}
x_{1} & x_{2} & \cdots & x_{p} \\
y_{1} & y_{2} & \cdots & y_{p} \\
1 & 1 & \cdots & 1
\end{array}\right]
\]

Input pixel locations in homogeneous form written as a \(3 \times 3\) matrix.
\[
\mathbf{Y}=\mathbf{H X} \quad \left\lvert\, \begin{gathered}
\mathrm{H} \text { is } \\
3 \times 3
\end{gathered}\right.
\]

Remapped pixel locations in homogeneous form written as a \(3 \times 3\) matrix. Note that usually each \(k_{i}\) is different.
\[
\left[\begin{array}{c}
k_{i} u_{i} \\
k_{i} v_{i} \\
k_{i}
\end{array}\right]=\left[\begin{array}{lll}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
h_{31} & h_{32} & h_{33}
\end{array}\right]\left[\begin{array}{c}
x_{i} \\
y_{i} \\
1
\end{array}\right]
\]
\[
\mathbf{Y}=\left[\begin{array}{cccc}
k_{1} u_{1} & k_{2} u_{2} & \cdots & k_{p} u_{p} \\
k_{1} v_{1} & k_{2} v_{2} & \cdots & k_{p} v_{p} \\
k_{1} & k_{2} & \cdots & k_{p}
\end{array}\right]
\]

Here, for vectors \(x_{i}\) and \(y_{i}\) are the elements of the transform.

\section*{Linear Warping of Images - Derivation}
\[
\begin{aligned}
k_{i} u_{i} & =h_{11} x_{i}+h_{12} y_{i}+h_{13} \\
k_{i} v_{i} & =h_{21} x_{i}+h_{22} y_{i}+h_{23} \\
k_{i} & =h_{31} x_{i}+h_{32} y_{i}+h_{33}
\end{aligned}
\]

For each pixel location there are 3 equations in 3 unknowns.

Divide the \(1^{\text {st }}\) and \(2^{\text {nd }}\) equations by the \(3^{\text {rd }}\).
\[
\begin{aligned}
& u_{i}=\frac{h_{11} x_{i}+h_{12} y_{i}+h_{13}}{h_{31} x_{i}+h_{32} y_{i}+h_{33}} \\
& v_{i}=\frac{h_{21} x_{i}+h_{22} y_{i}+h_{23}}{h_{31} x_{i}+h_{32} y_{i}+h_{33}}
\end{aligned}
\]
\[
\begin{aligned}
& u_{i}=\frac{h_{11} x_{i}+h_{12} y_{i}+h_{13}}{h_{31} x_{i}+h_{32} y_{i}+1} \\
& v_{i}=\frac{h_{21} x_{i}+h_{22} y_{i}+h_{23}}{h_{31} x_{i}+h_{32} y_{i}+1}
\end{aligned}
\]

Divide both numerator and denominator by
\(h_{33}\). Then relabel the coefficients.

\section*{Linear Warping of Images - Derivation}
\[
\begin{aligned}
& \left(h_{31} x_{i}+h_{32} y_{i}+1\right) u_{i}=h_{11} x_{i}+h_{12} y_{i}+h_{13} \\
& \left(h_{31} x_{i}+h_{32} y_{i}+1\right) v_{i}=h_{21} x_{i}+h_{22} y_{i}+h_{23}
\end{aligned}
\]

Multiply both sides by the right's denominator.
\[
-h_{11} x_{i}-h_{12} y_{i}-h_{13}+h_{31} x_{i} u_{i}+h_{32} y_{i} u_{i}+u_{i}=0
\]
\[
-h_{21} x_{i}-h_{22} y_{i}-h_{23}+h_{31} x_{i} v_{i}+h_{32} y_{i} v_{i}+v_{i}=0
\] side from both sides.
\[
\left[\begin{array}{rrrrrrrr}
-x_{i} & -y_{i} & -1 & 0 & 0 & 0 & x_{i} u_{i} & y_{i} u_{i} \\
0 \\
0 & 0 & 0 & -x_{i} & -y_{i} & -1 & x_{i} v_{i} & y_{i} v_{i} \\
v_{i}
\end{array}\right] \mathbf{h}=0
\]

Write as a matrix equation where ...
\[
\mathbf{h}=\left[\begin{array}{lllllll}
h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31}
\end{array} h_{32} 1\right]^{\mathrm{T}}
\]
... the elements of matrix H are written as vector \(h\).

\section*{Linear Warping of Images - Derivation}
\[
\mathbf{A}=\left[\right]
\]

Collect p 4 pixel locations, remap them as desired, and form the matrix A.

Solve for \(h\) \(\mathbf{h}=\left[\begin{array}{lllllllll}h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & 1\end{array}\right]^{\mathrm{T}}\) such that,
\[
\mathbf{A h}=0
\]

If the previous def of \(A\)

\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering led to H that gave wrong results, try this one:

\section*{Linear Warping of Images - Derivation}
\[
\mathbf{A}=\left[\begin{array}{rrrrrrrrr}
0 & 0 & 0 & -x_{1} & -y_{1} & -1 & x_{1} v_{1} & y_{1} v_{1} & v_{1} \\
x_{1} & y_{1} & 1 & 0 & 0 & 0 & -x_{1} u_{1} & -y_{1} u_{1} & -u_{1} \\
0 & 0 & 0 & -x_{2} & -y_{2} & -1 & x_{2} v_{2} & y_{2} v_{2} & v_{2} \\
x_{2} & y_{2} & 1 & 0 & 0 & 0 & -x_{2} u_{2} & -y_{2} u_{2} & -u_{2} \\
0 & 0 & 0 & -x_{p} & -y_{p} & -1 & x_{p} v_{p} & y_{p} v_{p} & v_{p} \\
x_{p} & y_{p} & 1 & 0 & 0 & 0 & -x_{p} u_{p} & -y_{p} u_{p} & -u_{p}
\end{array}\right]
\]

Collect p 24 pixel locations, remap them as desired, and form the matrix A.

Solve for \(h\) such that,
\(\mathbf{h}=\left[\begin{array}{lllllllll}h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & 1\end{array}\right]^{\mathrm{T}}\) \(\mathbf{A h}=0\).

EECE/CS 253 Image Processing

\section*{Linear Warping of Images - Derivation}
\[
\operatorname{svd}(\mathbf{A})=\mathbf{U S V}^{\mathrm{T}}
\]

To find \(h\) compute the singular value decomposition (svd) of \(A\).
\(S\) is a diagonal matrix of singular values.
\[
\mathbf{V}=\left[\begin{array}{llll}
\mathbf{v}_{1} & \mathbf{v}_{2} & \cdots & \mathbf{v}_{p}
\end{array}\right]
\]

Write matrix \(V\) in
\[
\mathbf{S}=\left[\begin{array}{cccc}
\sigma_{1} & 0 & \cdots & 0 \\
0 & \sigma_{2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_{p}
\end{array}\right]
\] terms of its columns.

Find \(\sigma_{k}\), the smallest sv.
\[
k=\arg \min \left\{\sigma_{1}, \sigma_{2}, \ldots, \sigma_{p}\right\}
\]

Then vector \(h\) is given by the \(\mathrm{k}^{\text {th }}\) column vector, \(\mathbf{v}_{\mathrm{k}}\).

\section*{Linear Warping of Images - Derivation}

If \(\mathbf{v}_{\mathrm{k}}\) is the column vector of \(\mathbf{V}\) that corresponds to the smallest singular value, then H is given by

Now ( \(r_{I}, c_{I}\) ) maps to ( \(r_{J}, c_{J}\) ) through H so that.

But we want to scan the output image, J , and at each location \(\left(r_{J}, c_{J}\right)\) take a value from I at location \(\left(r_{I}, c_{I}\right)\), so we do this:
\[
\mathbf{H}=\left[\begin{array}{lll}
v_{1 k} & v_{2 k} & v_{3 k} \\
v_{4 k} & v_{5 k} & v_{6 k} \\
v_{7 k} & v_{8 k} & v_{9 k}
\end{array}\right]
\]
\[
\mathbf{J}\left(r_{\mathbf{J}}, c_{\mathbf{J}}\right)=\mathbf{I}\left(r_{\mathbf{I}}, c_{\mathbf{I}}\right)
\]
\[
\mathbf{J}\left(r_{\mathbf{J}}, c_{\mathbf{J}}\right)=\mathbf{I}\left(\mathrm{N}\left\{\mathbf{H}^{-1}\left[\begin{array}{l}
c_{\mathbf{J}} \\
r_{\mathbf{J}} \\
1
\end{array}\right]\right\}\right)
\]

\section*{Linear Warping of Images - Remapping}

The inverse mapping
of ( \(r_{J}, c_{J}, 1\) ) through
\(H^{-1}\) is \(\left(k r_{I}, k c_{J}, k\right) \ldots\)
... and that must be normalized as follows:

Thus for each pixel location \(\left(r_{J}, c_{J}\right)\) in the warped image,
\[
\mathbf{H}^{-1}\left[\begin{array}{c}
c_{J} \\
r_{J} \\
1
\end{array}\right]=\left[\begin{array}{c}
k_{(r, c)} c_{\mathbf{I}} \\
k_{(r, c)} r_{\mathbf{I}} \\
k_{(r, c)}
\end{array}\right]
\]
\[
\mathrm{N}\left\{\left[\begin{array}{c}
k_{(r, c)} c_{I} \\
k_{(r, c)} r_{I} \\
k_{(r, c)}
\end{array}\right]\right\}=\frac{1}{k_{(r, c)}}\left[\begin{array}{l}
k_{(r, c)} c_{I} \\
k_{(r, c)} r_{I}
\end{array}\right]=\left[\begin{array}{l}
c_{\mathbf{I}} \\
r_{\mathbf{I}}
\end{array}\right]
\]
\[
\mathbf{J}\left(r_{J}, c_{J}\right)=\mathbf{I}\left(\mathbf{N}\left\{\mathbf{H}^{-1}\left[\begin{array}{c}
c_{J} \\
r_{J} \\
1
\end{array}\right]\right\}\right)
\]

\section*{Linear Warping of Images - Steps}
1. Select at least four pixel locations from I, the image to be warped.
2. Create target locations by altering the values of the selected locs.
3. Construct from the location pairs, matrix A as described on slide 16.
4. Compute the singular value decomposition of \(\mathbf{A}=\mathbf{U S V}^{\top}\).
5. Select the vector \(\mathbf{v}_{\mathbf{k}}\) that corresponds to the smallest singular value.
6. Construct \(\mathbf{H}\) from \(\mathbf{v}_{\mathbf{k}}\).
7. Compute \(\mathbf{H}^{-1}\).
8. Create an output image \(\mathbf{J}\).
9. For each \(\left(r_{\mathbf{J}}, c_{\mathbf{J}}\right)\) in \(\mathbf{J}\), select \(\left(r_{\mathbf{I}}, c_{\mathbf{I}}\right)\) from \(\mathbf{I}\) using the eqns. on slide 19.
10. Since \(\left(r_{\mathrm{I}}, c_{\mathrm{I}}\right)\) is fractional, interpolate on the neighborhood of \(\left(r_{\mathrm{I}}, c_{\mathrm{I}}\right)\) in \(\mathbf{I}\) to compute \(\mathbf{J}\left(r_{\mathbf{J}}, c_{\mathbf{J}}\right)\).

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation}
image size:
\[
\left[R_{\mathrm{in}}, C_{\mathrm{in}}, B\right]=\operatorname{size}(\mathbf{I})
\]
aspect angle:
\[
\theta_{A}=\tan ^{-1}\left[\frac{R_{\mathrm{in}}}{C_{\mathrm{in}}}\right]
\]
length of diagonal:
\[
D=\sqrt{R_{\mathrm{in}}^{2}+C_{\mathrm{in}}^{2}}
\]

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Image Rotation}
angle of rotation: \(\theta\)
rotation matrix:
\[
\mathbf{P}(\theta)=\left[\begin{array}{rr}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{array}\right]
\]
transforms input image cdts. \({ }^{1}\) to output image cdts.
\[
\left[\begin{array}{l}
r \\
c
\end{array}\right]=\mathbf{P}(\theta)\left[\begin{array}{l}
r_{\mathrm{in}}-R_{\text {in } 0} \\
c_{\mathrm{in}}-C_{\mathrm{in} 0}
\end{array}\right]+\left[\begin{array}{l}
R_{\text {out0 }} \\
R_{\text {out0 }}
\end{array}\right]
\]


\footnotetext{
\({ }^{1}\) cdts. measured w.r.t. the center of the image: \(\left(R_{\text {in } 0}, C_{\text {in0 }}\right)=\left(\left\lfloor\frac{1}{2} R_{\text {in }}\right\rfloor+1,\left\lfloor\frac{1}{2} C_{\text {in }}\right\rfloor+1\right)\)
}

\section*{Image Rotation}

Compute the dimensions of the output image: number of rows.
\[
R_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dsin}\left(\theta+\theta_{A}\right)\right|\right),
\]

if \(0^{\circ}<\theta \leq 90^{\circ}\).

\author{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation}

Compute the dimensions of the output image: number of columns.
\(C_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta-\theta_{A}\right)\right|\right)\)

if \(0^{\circ}<\theta \leq 90^{\circ}\).

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Image Rotation}

The output dimension calculation depends on the value of \(\theta\) as follows:

If \(0^{\circ} \leq \theta<90^{\circ}\) :
\[
\begin{aligned}
& R_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Din}\left(\theta+\theta_{A}\right)\right|\right), \\
& C_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta-\theta_{A}\right)\right|\right) .
\end{aligned}
\]
\[
\text { If }-90^{\circ} \leq \theta<0^{\circ} \text { : }
\]
\[
R_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Din}\left(\theta-\theta_{A}\right)\right|\right),
\]
\[
C_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta+\theta_{A}\right)\right|\right) .
\]
\[
\text { If } 90^{\circ} \leq \theta<180^{\circ}
\]
\[
\begin{aligned}
& R_{\mathrm{out}}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta-90-\theta_{A}\right)\right|\right), \\
& C_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Din}\left(\theta-90+\theta_{A}\right)\right|\right)
\end{aligned}
\]
\[
\text { If }-180^{\circ} \leq \theta<-90^{\circ} \text { : }
\]
\[
R_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta+90+\theta_{A}\right)\right|\right),
\]
\[
C_{\text {out }}=\operatorname{round}\left(\left|D \sin \left(\theta+90-\theta_{A}\right)\right|\right) .
\]

\author{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Image Rotation}
\(\mathrm{C}_{\text {out }}\)

Allocate an output image with dimensions ( \(\mathrm{R}_{\text {out }}, \mathrm{C}_{\text {out }}\) ), where \({ }^{1}\) \(R_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Din}\left(\theta+\theta_{A}\right)\right|\right)\) \(C_{\text {out }}=\operatorname{round}\left(\left|\operatorname{Dcos}\left(\theta-\theta_{A}\right)\right|\right)^{\mathrm{R}_{\text {out }}}\) and center point
\(\left(R_{\text {out0 }}, C_{\text {out0 } 0}\right)=\)
\(\left(\left\lfloor\frac{1}{2} R_{\text {out }}\right\rfloor+1,\left\lfloor\frac{1}{2} C_{\text {out }}\right\rfloor+1\right)\).

\({ }^{1} R_{\text {out }}\) and \(C_{\text {out }}\) below are valid for \(0^{\circ}<\theta \leq 90^{\circ}\). Otherwise see slide 7 .

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image Rotation}

\(\mathbf{P}^{-1}(\theta)=\left[\begin{array}{cc}\cos \theta & -\sin \theta \\ \sin \theta & \cos \theta\end{array}\right]\)
\(=\mathbf{P}(-\theta)\)
Work backward. For every output loc. \((r, c)\) select an input loc. \(\left(r_{f}, c_{f}\right)\) by rotating \((r, c)\) around the image center by \(-\theta\).
\(\Phi(r, c,:)=\left[\begin{array}{l}r_{f} \\ c_{f}\end{array}\right]\)
\[
=\mathbf{P}^{-1}(\theta)\left[\begin{array}{l}
r-R_{\text {out } 0} \\
c-C_{\text {out } 0}
\end{array}\right]
\]
\[
+\left[\begin{array}{l}
R_{\mathrm{in0}} \\
C_{\mathrm{in0}}
\end{array}\right]
\]

\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Image Rotation}

Rotating the input image by \(\theta\) is equivalent to rotating the ouput image by \(-\theta\).
\(\Phi(r, c,:)=\mathbf{P}^{-1}(\theta)\left[\begin{array}{l}r-R_{\text {out0 }} \\ c-C_{\text {out } 0}\end{array}\right]+\left[\begin{array}{l}R_{\text {in0 }} \\ C_{\text {in0 } 0}\end{array}\right]\)


\author{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Image Rotation}

After rotation by \(-\theta\), \(\mathbf{J}(r, c)\) is in nearly the same position as \(\mathbf{I}\left(r_{p} c_{f}\right)\).
\(\Phi(r, c,:)=\left[\begin{array}{l}r_{f} \\ c_{f}\end{array}\right]\)
\(=\mathbf{P}^{-1}(\theta)\left[\begin{array}{l}r-R_{\text {out } 0} \\ c-C_{\text {out } 0}\end{array}\right]\)
\(+\left[\begin{array}{l}R_{\text {in0 }} \\ C_{\mathrm{in0} 0}\end{array}\right]\)

\author{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation}

After rotation by \(-\theta\), \(\mathbf{J}(r, c)\) is in nearly the same position as \(\mathbf{I}\left(r_{f}, c_{f}\right)\).
\[
\begin{aligned}
\Phi(r, c,:) & =\left[\begin{array}{l}
r_{f} \\
c_{f}
\end{array}\right] \\
& =\mathbf{P}^{-1}(\theta)\left[\begin{array}{l}
r-R_{\mathrm{out} 0} \\
c-C_{\mathrm{out} 0}
\end{array}\right] \\
& +\left[\begin{array}{l}
R_{\mathrm{in} 0} \\
C_{\mathrm{in} 0}
\end{array}\right]
\end{aligned}
\]

\section*{Image Rotation}

After rotation by \(-\theta\), \(\mathbf{J}(r, c)\) is in nearly the same position as \(\mathbf{I}\left(r_{p} c_{f}\right)\).
\[
\begin{aligned}
\Phi(r, c,:) & =\left[\begin{array}{l}
r_{f} \\
c_{f}
\end{array}\right] \\
& =\mathbf{P}^{-1}(\theta)\left[\begin{array}{l}
r-R_{\text {out } 0} \\
c-C_{\text {out } 0}
\end{array}\right]
\end{aligned}
\]

\[
+\left[\begin{array}{l}
R_{\mathrm{in} 0} \\
C_{\mathrm{in} 0}
\end{array}\right]
\]

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Image Rotation}
\[
\mathbf{J}(r, c)=\Theta\left\{\mathbf{I} ; \mathfrak{N}\left(r_{f}, c_{f}\right)\right\} .
\]

Interpolation:
The output pixel value is (usually) a function of the values on a neighborhood - a set of pixels that surrounds - \(\left(r_{f}, c_{f}\right)\).

Bilinear Interp. uses a \(2 \times 2\) neighborhood, bicubic uses a \(4 \times 4\).

Nearest neighbor is simply,
 \(\mathbf{J}(r, c)=\mathbf{I}\left(r_{i}, c_{i}\right)\) where \(\left(r_{i}, c_{i}\right)=\operatorname{round}\left(r_{f}, c_{f}\right)\).

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation with Interpolation}


Original image: San Francisco financial district

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation with Nearest Neighbor Interpolation}


Bicubic - Nearest Neighbor
Nearest Neighbor - Bilinear

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image Rotation with Bilinear Interpolation}


Nearest Neighbor - Bilinear
Bilinear - Bicubic

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Image Rotation with Bicubic Interpolation}


Bilinear - Bicubic
Bicubic - Nearest Neighbor

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example of Warping: Map Image to Sphere}

For the output image,
\[
\rho_{0}=\frac{1}{2} \min \left(R_{\mathrm{out}}, C_{\mathrm{out}}\right) .
\]

For each output pixel,
\[
\begin{aligned}
\rho & =\sqrt{r_{\text {out }}^{2}+c_{\text {out }}^{2}}, \\
\theta & =\tan ^{-1}\left(\frac{r_{\text {out }}}{c_{\text {out }}}\right), \\
\phi & =\sin ^{-1}\left(\frac{\rho}{\rho_{0}}\right) .
\end{aligned}
\]


For the input image,
\[
d_{0}=\frac{1}{2} \max \left(R_{\mathrm{in}}, C_{\mathrm{in}}\right) .
\]

For each output pixel, the input pixel loc is,
\[
\begin{aligned}
d & =\frac{2}{\pi} d_{0} \phi \\
r_{\mathrm{in}} & =d \sin (\theta) \\
c_{\mathrm{in}} & =d \cos (\theta)
\end{aligned}
\]

\section*{Example of Warping: Map Image to Sphere}

d: radial distance from center of input image. \(\rho\) : same for output image.

\title{
EECE\CS 253 Image Processing
}

\author{
Lecture Notes: Reduction of Uncorrelated Noise
}

\section*{Richard Alan Peters II}

\section*{Department of Electrical Engineering and Computer Science}

Fall Semester 2011

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Noise in Images}

\section*{All images created through optical projection onto a sensor array are noisy.}

\section*{Correlated noise}
- Due to electrical interference
- Due to source / sensor interference
- Halftone distortion / moiré patterns


\section*{Uncorrelated noise}
- Quantum noise in CCD arrays
- Silver halide grains in film photography
- Neuronal noise in a retina
- Quantization noise in digital photographs

\section*{Image with Additive Noise}

\[
\mathbf{J}(r, c)=\mathbf{I}(r, c)+\mathbf{N}(r, c) .
\]

spatial domain

frequency domain

Intensity distributions -

\section*{Uncorrelated Noise}


Each pixel's value has probability of occurrence given by the associated distribution.

Intensity distributions EECE/CS 253 Image Processing normalized histograms

\author{
Vanderbilt University School of Engineering
}

The most likely value is 1288 Jncorrelated Noise
with an average with an average difference of 25 from 128 (std. dev.).





Each pixel's value has probability of occurrence given by the associated distribution.

\section*{Uncorrelated Color Noise: Gaussian}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Uncorrelated Color Noise: Uniform}


\section*{Gaussian IID Noise Field}



IID: Independent, Identically Distributed

\section*{Gaussian IID Noise Field}
IID \(\Rightarrow\) no spatial correlation


IID: Independent, Identically Distributed

\section*{Autocorrelation of an Image}

Let the support of \(\mathbf{I}\) be a torus. ( \(\mathbf{I}\) is defined on a torus a la the Fourier transform.) Let \(\tilde{\mathbf{I}}\) be \(\mathbf{I}\) minus the mean value of \(\mathbf{I}\). Make a copy of \(\tilde{\mathbf{I}}\). Shift the copy by ( \(\rho, \chi\) ) on the torus. Pixel-wise multiply the shifted version by the original and sum the products.
\[
\mathbf{A}_{\mathbf{I}}(\rho, \chi)=\frac{1}{R C} \sum_{r=1}^{R} \sum_{c=1}^{C} \tilde{\mathbf{I}}(r, c) \tilde{\mathbf{I}}(\psi(r+\rho ; R), \psi(c+\chi ; C))
\]
where
\[
\tilde{\mathbf{I}}(\rho, \chi)=\mathbf{I}(\rho, \chi)-\frac{1}{R C} \sum_{r=1}^{R} \sum_{c=1}^{C} \mathbf{I}(r, c)
\]
and
\(\psi(x ; N)=\left\{\begin{array}{lll}\bmod (x, N) & \text { if } & x \geq 0 \\ \bmod (x+N, N) & \text { if } & x<0\end{array}\right.\)
\(A_{I}(\rho, X)\), the autocorrelation of I at offset ( \(\rho, x\) ), is a measure of the similarity of I to itself when shifted by ( \(\rho, \mathrm{x}\) ).

\section*{Power Spectrum \& Autocorrelation of IID Noise}

\[
\operatorname{PS}(\mathbf{I})=|\mathscr{F}(\mathbf{I})|^{2}
\]

\[
\mathbf{A}_{\mathbf{I}}(\rho, \chi)=\operatorname{Re}\left[\mathscr{F}^{-1}\left\{|\mathscr{F}(\mathbf{I})|^{2}\right\}\right]
\]

\section*{Power Spectrum \& Autocorrelation of IID Noise}

\[
\operatorname{PS}(\mathbf{I})=|\mathscr{F}(\mathbf{I})|^{2}
\]

\[
\mathbf{A}_{\mathbf{I}}(\rho, \chi)=\operatorname{Re}\left[\mathscr{F}^{-1}\left\{|\mathscr{F}(\mathbf{I})|^{2}\right\}\right]
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Noise-Free Image and Uncorrelated Noise Field}

image


Gaussian noise field

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Spectra of Noise-Free Image and Uncorr. Noise Field}

image center row log power spectrum

noise field center row log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Sum of Noise-Free Image and Uncorrelated Noise Field}

image + noise field

image + noise field center row log PS

\section*{Power Spectra of Noise-Free Image and Noise Field}

original image

noise image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectra of Sum of Image and Noise Field}

original image

noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectra of Sum of Image and Noise Field}

original image

blue indicates noise > image

\section*{Power Spectra of Sum of Image and Noise Field}

noise image

red indicates image > noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectra of Sum of Image and Noise Field}

noisy image

image \& noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise: Another Example}

original image

noise image

image+noise

\section*{Additive Noise: Another Example \\ displayed: \\ \(\log \left\{|\mathscr{F}(\mathbf{I})|^{2}+1\right\}\)}

image PS

noise PS

image+noise PS

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise: Another Example \\ displayed: \(\log \left\{|\mathscr{F}(\mathbf{I})|^{2}+1\right\}\)}

image PS

image+noise PS

image PS > noise PS

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Additive Noise: Reduce Through Blurring?}

red indicates image \(>\) noise

> At some frequency, \(f_{0}\), there are more components where the noise power is greater than the image power. image PS > noise PS

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Additive Noise: Reduce Through Blurring?}

red indicates image > noise

image PS > noise PS

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise: Reduction Through Blurring.}


PS of Gaussian blurred image

> The result is actually no better. There's less noise but the blurring looks worse.

Gaussian Blurred Image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise: Reduction Through Blurring.}


PS of Gaussian blurred image

The result is actually no better. There's less noise but the blurring looks worse.

\section*{Noise Masking}

power spec. of noisy image
red: image > noise blue: image < noise

\section*{Noise Masking}

power spec. of noisy image

image < noise masked out

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Noise Masking}

noisy image

noise-masked mage

\section*{Noise Masking}

Although the noise-masked image looks better than the blurred one, it is still noisy. Moreover, this example is unrealistic because we know the exact noise power spectrum. In any real case we will at most know its statistics.

blurred noisy image

noise-masked mage

\section*{Image Degradation Model}

So far, we have considered only additive noise. Before going further it will be useful to consider a more general model of image degradation, one that includes convolution with a pointspread \({ }^{1}\) function, H , as well as additive noise.

\({ }^{1} \mathrm{H}\) is also referred to as the optical transfer function.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Pointspread Operators}

A pointspread operator is a linear model of the distortion acquired during the imaging process. Since it is a linear model, it is a convolution operator. One example of this is aperture distortion, an unavoidable consequence of making an image with a camera that has an opening larger than a point.


\section*{Pointspread Operators}
pinhole camera


A pinhole camera maps one object point to one image point; it is one-to-one.
aperture camera


An aperture camera maps one object point to many image points; it spreads the points.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Pointspread Operators and Convolution}
\[
\mathbf{I}(r, c)
\]
\[
\mathbf{J}(r, c)=\mathbf{I}(r, c) * \mathbf{H}(r, c)
\]


Recall how a convolution works through multiply, shift, and add (See Lect. 7 p. 25ff). That is precisely the effect of imaging through an aperture. It results in a blurry image.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Lenses}

A properly designed lens will focus the light emanating from a point and thereby reduce the blurring. But no lens can do this perfectly. In fact, the lens adds its own distortion. The result is an optical transfer function, \(\mathbf{H}(r, c)\), that is convolved with the image.


\section*{Image Degradation Model}


Note: The term pointspread operator refers to convolution by the pointspread function.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image Degradation Model}

\(\mathrm{I}(r, c) * \mathrm{H}(r, c)\)


\(\mathrm{N}(r, c)\)
\[
\mathrm{J}(r, c)=\mathrm{I}(r, c) * \mathrm{H}(r, c)+\mathrm{N}(r, c)
\]

\section*{Image Degradation Model (Frequency Domain)}


\section*{Image Degradation Model (Frequency Domain)}


\section*{Image Restoration}

Let \(\mathbf{I}\) be a perfect image and let \(\mathbf{K}\) be the image convolved with a pointspread function, \(\mathbf{H}\). Then in the frequency domain:
\[
\mathcal{K}(u, v)=\mathcal{I}(u, v) \mathcal{H}(u, v)
\]

If the process of imaging adds noise then we get \(\mathbf{J}=\mathbf{K}+\mathbf{N}\), or in freq.:
\[
\mathcal{J}(u, v)=\mathcal{K}(u, v)+\boldsymbol{\mathcal { N }}(u, v)
\]

We want a filter, \(\mathbf{W}\), to remove as much of the noise from \(\mathbf{J}\) as possible:
\[
\tilde{\mathcal{K}}(v, u)=\mathcal{W}(v, u) \mathcal{J}(v, u)
\]

Then an estimate of \(\mathbf{I}\) would be the inverse Fourier transform of
\[
\tilde{\mathcal{I}}(u, v)=\frac{\tilde{\mathcal{K}}(u, v)}{\mathcal{H}(u, v)}=\frac{\mathcal{W}(u, v) \mathcal{J}(u, v)}{\mathcal{H}(u, v)}
\]

We want to find the filter, \(\mathbf{W}\), that results in the closest possible estimate of \(\mathbf{I}\) i.e. the \(\mathbf{W}\) that minimizes the energy of the difference between the estimate and \(\mathbf{I}\). That is we want to find \(\mathbf{W}\) such that
\[
\varepsilon^{2}=\iint|\mathcal{I}-\tilde{\mathcal{I}}|^{2} d u d v
\]
is as small as possible. This is called least mean squared (LMS) minimization.

\section*{Image Restoration}

There are a number of ways to solve for the minimum squared error. All make use of the assumption that the image and the noise are uncorrelated. Depending on how that fact is used, slightly different solutions are found. The most common one used in image processing is the Wiener filter:
\[
\mathcal{W}=\frac{\mathcal{H}^{*}|\mathcal{I}|^{2}}{|\mathcal{H}|^{2}|\mathcal{I}|^{2}+|\mathcal{N}|^{2}}
\]

Then, with a little bit of algebra, we get
\[
\mathcal{W} \mathcal{J}=\frac{|\mathcal{H}|^{2} \boldsymbol{I}+\mathcal{H}^{*} \mathcal{N}}{|\mathcal{H}|^{2}+\frac{|\mathcal{N}|^{2}}{|\boldsymbol{I}|^{2}}}
\]

For frequencies ( \(u, v\) ) where noise power is smaller than the image power \(\mathcal{W}\) acts like an inverse filter since
\(\boldsymbol{\mathcal { N }}(u, v) / \mathcal{I}(u, v)<1\) and
\[
\mathcal{W} \mathcal{J}(u, v) \approx \frac{|\mathcal{H}|^{2}}{|\mathcal{H}|^{2}} \mathcal{I}(u, v)=\mathcal{I}(u, v)
\]
and at frequencies where the noise power dominates, \(\mathcal{N}(u, v) / \mathcal{I}(u, v)>1\) and
\(\mathcal{W} \mathcal{J}(u, v)=\frac{|\mathcal{I}|^{2} \mathcal{H}^{*}}{|\mathcal{I}|^{2}|\mathcal{H}|^{2}+|\mathcal{N}|^{2}} \boldsymbol{\mathcal { N }}(u, v)\),
the fraction is small so the noise power is diminished.

\section*{Image Restoration}
\[
\begin{array}{rlrl}
\varepsilon^{2} & =\iint|\mathcal{I}-\tilde{\mathcal{I}}|^{2} d u d v & \begin{array}{l}
\text { This is one of the }
\end{array} \\
& =\iint\left|\frac{\mathcal{K}}{\mathcal{H}}-\frac{\mathcal{W} \mathcal{J}}{\mathcal{H}}\right|^{2} d u d v & \begin{array}{l}
\text { possible derivations } \\
\text { of the Wiener filter }
\end{array} \\
& =\iint|\mathcal{H}|^{-2}|\mathcal{K}-\mathcal{W}(\mathcal{K}+\mathcal{N})|^{2} d u d v & \\
& =\iint|\mathcal{H}|^{-2}|\mathcal{K}(1-\mathcal{W})+\mathcal{W N}|^{2} d u d v & \\
& =\iint|\mathcal{H}|^{-2}[\mathcal{K}(1-\mathcal{W})+\mathcal{W N}][\overline{\mathcal{K}(1-\mathcal{W})+\mathcal{W N}]} d u d v \\
& =\iint|\mathcal{H}|^{-2}\left[|\mathcal{K}(1-\mathcal{W})|^{2}+\mathcal{K}(1-\mathcal{W}) \overline{\mathcal{W N}}+\mathcal{W N} \overline{\mathcal{K}(1-\mathcal{W})}+|\mathcal{W N}|^{2}\right] d u d v \\
& =\iint|\mathcal{H}|^{-2}\left\{|\mathcal{K}(1-\mathcal{W})|^{2}+2 \operatorname{Re}[\mathcal{K}(1-\mathcal{W}) \overline{\mathcal{W N}}]+|\mathcal{W N}|^{2}\right\} d u d v \\
& =\iint|\mathcal{H}|^{-2}\left[|\mathcal{K}(1-\mathcal{W})|^{2}+|\mathcal{W N}|^{2}\right] d u d v+2 \operatorname{Re} \iint|\mathcal{H}|^{-2}(1-\mathcal{W}) \overline{\mathcal{W}} \mathcal{K} \overline{\mathcal{N}} d u d v
\end{array}
\]

\section*{Image Restoration}

From the previous page, the squared error is
\[
\varepsilon^{2}=\iint|\mathcal{H}|^{-2}\left[|\mathcal{K}(1-\mathcal{W})|^{2}+|\mathcal{W} \mathcal{N}|^{2}\right] d u d v+2 \operatorname{Re} \iint|\mathcal{H}|^{-2}(1-\mathcal{W}) \overline{\mathcal{W}} \mathcal{K} \overline{\mathcal{N}} d u d v .
\]

The second term should be small compared to the first since it can be written
\[
2 \operatorname{Re} \iint|\mathcal{H}|^{-1}(1-\mathcal{W}) \overline{\mathcal{W}} \mathcal{I} \overline{\mathcal{N}} d u d v,
\]
and the image and the noise are assumed to be uncorrelated \({ }^{1}\). Thus the error can be approximated by
\[
\varepsilon^{2}=\iint|\mathcal{H}|^{-2}\left[|\mathcal{K}(1-\mathcal{W})|^{2}+|\mathcal{W N}|^{2}\right]^{2} d u d v .
\]

The mean squared error, \(\varepsilon^{2}\), is minimized when \(\mathbf{W}\) is given by,
\[
\mathcal{W}=\frac{\mathcal{H}{ }^{*}|\mathcal{I}|^{2}}{|\mathcal{H}|^{2}|\mathcal{I}|^{2}+|\mathcal{N}|^{2}} .
\]
\[
\iint \mathcal{I} \overline{\mathcal{N}} d u d v=0
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Noise Reduction Through LMS Filtering \({ }^{1}\)}

image


Gaussian noise field

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Noise Reduction Through LMS Filtering \({ }^{1}\)}

image

noisy image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise (Power Spectra)}

original image

noisy image

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Additive Noise (Power Spectra)}

noisy image


Wiener filter

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise (Power Spectra)}

noisy image


Wiener filter

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise (Power Spectra)}


Wiener filtered image

original image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise}

noisy image


Wiener filtered image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise}


Wiener filtered image

original image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Additive Noise}


Gaussian blurred image

original image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Noise Reduction Through LMS Filtering \({ }^{1}\)}

image

noisy image \(\mathbf{J}=\mathbf{I} * \mathbf{h}+\mathbf{N}\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image*PSF + Noise (Power Spectra)}

original image

noisy image \(\mathbf{J}=\mathbf{I} * \mathbf{h}+\mathbf{N}\)

\section*{Image*PSF + Noise (Power Spectra)}


Wiener filtered image

In this example we knew the exact image and noise power spectra and the PSF was Gaussian w/ \(\mu=0\), \(\sigma=2\). In a real example, none of that is true.


Wiener filter

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image*PSF + Noise (Power Spectra)}


Wiener filtered image


Wiener filter

\section*{Image*PSF + Noise (Power Spectra)}


Wiener filtered image

original image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image*PSF + Noise}

noisy image \(\mathbf{J}=\mathbf{I} * \mathbf{h}+\mathbf{N}\)


Wiener filtered image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image*PSF + Noise}


Wiener filtered image

original image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{LMS Image Restoration (Real Example)}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{LMS Image Restoration (Real Example)}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Noise Estimation}
deviations of

\[
\begin{aligned}
& \sigma_{R}=5.0981 \\
& \sigma_{G}=4.0672 \\
& \sigma_{B}=6.9212
\end{aligned}
\] each band:

\section*{Pointspread Function Estimation}


To estimate the PSF, find the image of a point and construct a convolution mask from it.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{LMS Image Restoration (original)}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{LMS Image Restoration (filtered)}


\title{
EECE/CS 253 Image Processing
}

\author{
Vanderbilt University School of Engineering
}

\section*{Detail of Results}

The contrast of these has been increased to make the differences more visible.

original image

filtered image

matlab's wiener2

\title{
EECE\CS 253 Image Processing
}

Lecture Notes: Reduction of Correlated Noise

\section*{Richard Alan Peters II}

\section*{Department of Electrical Engineering and \\ Computer Science}

Fall Semester 2011

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Periodic Noise}

original image

image + noise

\section*{Noise Reduction through Directional Blurring}

image + noise

blurred image

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Power Spectrum of Image with Periodic Noise}

original image

image + noise

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Low Frequency Region}

original image

image + noise

\section*{Noise Reduction through Notch Filtering}

noise mask

masked power spectrum

\section*{Inverse of Masked Fourier Transform}

original image

noise reduced image

\section*{Notch filter reduction of periodic noise}
1. Take the FFT of the input image. \(\mathbf{F}=\mathrm{fft} 2(\mathbf{I})\).
2. Display the log power spectrum of the fftshift-ed \(\mathbf{F}\).
3. Find the locations, \(\mathbf{x}_{i}\), of the spikes that correspond to the periodic distortion.
4. Create a 1-band image, \(\mathbf{M}\), of class double the same size as \(\mathbf{I}\).
5. Set all M's pixels to 1.0.
6. Let \(\mathscr{\mathscr { H }}\left(\mathbf{x}_{i}\right)\) be a neighborhood of \(\mathbf{x}_{i}\) with area sufficient to cover the spike at \(\mathbf{x}_{i}\).
7. For each \(\mathbf{x}_{i}\) do: for each \(\mathbf{y}_{j} \in \mathscr{T}\left(\mathbf{x}_{i}\right)\), set \(\mathbf{M}\left(\mathbf{y}_{j}\right)=0\).
8. Blur \(\mathbf{M}\) with a Gaussian whose \(\sigma\) is smaller than \(1 / 2\) the radius of \(\mathcal{O}\left(\mathbf{x}_{i}\right)\).
9. Take the ifftshift of \(\mathbf{M}\).
10. For each band, \(k\), of the image let \(\mathbf{G}_{k}=\mathbf{F}_{k} .{ }^{*} \mathbf{M}\).
11. Then the noise-reduced image is: \(\mathbf{J}=\operatorname{real}(\operatorname{ifft2(G))}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{How to determine the frequency and period of a point in the log power spectrum of an image (ex.):}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{How to determine the frequency and period of a point in the log power spectrum of an image (ex.):}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering
How to determine the frequency and period of a point in the log power spectrum of an image:


How to determine the frequency and period of a point in the log power spectrum of an image:

\[
\begin{aligned}
v & =r_{2}-\lfloor R / 2\rfloor-1 \\
-v & =\lfloor R / 2\rfloor+1-r_{1} \\
u & =c_{2}-\lfloor C / 2\rfloor-1 \\
-u & =\lfloor C / 2\rfloor+1-c_{1}
\end{aligned}
\]
\[
\begin{aligned}
& \lambda_{\mathrm{wf}}=\sqrt{\left(\frac{c}{u}\right)^{2}+\left(\frac{R}{v}\right)^{2}} \\
& \omega_{\mathrm{wff}}=\frac{1}{\lambda_{\mathrm{wf}}} \\
& \theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{v C}{u R}\right)
\end{aligned}
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Points on the Fourier Plane (of a Digital Image)}

In the Fourier transform of an \(R \times C\) digital image the wavelengths, \(\lambda_{u}\) and \(\lambda_{v}\) represent a fraction of the \(R\) and \(C\) values. That is,
\[
\lambda_{u}=\frac{C}{u} \text { and } \lambda_{v}=\frac{R}{v} \text { pixels. }
\]

The wavefront direction is given by
\[
\theta_{\mathrm{wf}}=\tan ^{-1}\left(\frac{v C}{u R}\right),
\]
and the wavelength is
\[
\lambda_{\mathrm{wf}}=\sqrt{\left(\frac{C}{u}\right)^{2}+\left(\frac{R}{v}\right)^{2}} .
\]

The frequencies represent fractions of \(R \& C\),
\[
\begin{aligned}
& \omega_{u}=\frac{u}{C}, \omega_{v}=\frac{v}{R}, \text { and } \\
& \omega_{\mathrm{wf}}=1 / \sqrt{\left(\frac{C}{u}\right)^{2}+\left(\frac{R}{v}\right)^{2}} \text { cycles. }
\end{aligned}
\]


EECE/CS 253 Image Processing
How to determine the frequency plane location of a sinusoid:
 plane location of a sinusoid:

cosine grating \(\omega=8 \sqrt{2}\), orientation \(=3 \pi / 4 \Rightarrow \omega_{u}= \pm 8, \omega_{v}= \pm 8\)

Sinusoidal grating image with an even number or rows and


\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{FT of Sinusoidal Grating Image}
\[
\begin{aligned}
& \omega_{u}= \pm R / \lambda_{r} \\
& \omega_{v}= \pm C / \lambda_{c}
\end{aligned}
\]



\section*{Processing Scanned Press-Printed Images}

\section*{4-color printing:}
1. A photograph or other color image is separated into four intensity band images: cyan, magenta, yellow, and black.
2. Each of these is multiplied by a halftone screen \({ }^{1}\) - a dot mask with a unique orientation.
3. Each of the resulting four images shows a pattern of dots whose individual sizes indicate the amount of ink to be applied at each point.
4. The four images are printed, one atop the other, in the corresponding color.
\({ }^{1}\) Merriam-Webster Dictionary: half•tone 2 (1): a photoengraving made from an image photographed through a screen and then etched so that the details of the image are reproduced in dots.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering is the pointwise product of

\section*{Halftone Screen ( \(45^{\circ}\) )} 2 sinusoidal gratings with perpendicular orientations.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Halftone Screens ( \(90^{\circ}\) )}

The orientation of the screen is the average of the grating orientations.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Standard Halftone Screen Angles}


Image from Adobe Photoshop CS2 documentation.

Each band has 2 perpendicular sinusoids + a DC component...

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{CMYK Standard Halftone Screens}



\section*{Power Spectra}
... which creates rectangular grids at 4 different angles.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{CMYK Standard Halftone Screens}


When the 4 are summed, the result is a "rosette" image.

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{CMYK Standard Halftone Screens}

space domain images

To print an image, it is separated into 4 color bands.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Color Separation / Halftoning}


That is, an intensity image is created for each of the four color bands.

... each of which is multiplied by EECE/CS 253 Image Processing a corresponding screen.

\author{
Vanderbilt University School of Engineering
}

\section*{Color Separation / Halftoning}

Each intensity image is multiplied by a corresponding screen, then

each screened image is printed in its own color on the same page.


To print an image, it is separated into 4 color bands

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: Color Separation / Halftoning}


Here the bands tinted in their corresponding colors.


\section*{Example: Color Separation / Halftoning}


Here the screens tinted in their corresponding colors.

...to get dot patterns for printing. The 4 are printed over each other to get the final result.

\section*{Example: Color Separation / Halftoning}


\section*{Halftone Dots}


Image scanned (600 dpi) from a magazine


Detail: Circular patterns, the rosettes, are the result of the halftone dots.

\section*{Filtering Out Halftone Dot Distortion}

original

log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering Out Halftone Dot D}

Each pair of peaks corresponds to a sinusoidal sub-pattern

original

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\int=1\) )}

blurred image \(\lceil=1\)

log power spectrum \(\lceil=1\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\int=2\) )}

blurred image \(\lceil=2\)

\(\log\) power spectrum \(\lceil=2\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\int=4\) )}

blurred image \(\sigma=4\)

log power spectrum \(\sigma=4\)

\section*{Blurring with a Gaussian \((\sigma=8)\)}

blurred image \(\sigma=8\)

\(\log\) power spectrum \(\sigma=8\)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\sigma=1\) )}

original

difference

blurred \(\sigma=1\)
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\sigma=2\) )}

blurred \(\sigma=2\)

difference

original
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian ( \(\sigma=4\) )}

original

difference

blurred \(\sigma=4\)
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurring with a Gaussian \((\sigma=8)\)}

blurred \(\sigma=8\)

difference

original
middle gray \(=0\), normalized

EECE/CS 253 Image Processing

\section*{Problem with Blurring to Reduce HTD Distortion}
" It blurs everything.
Better to remove the HTD frequency components selectively:
1. Read in the image.
2. Compute the log power spectrum of the image.
3. Find the locations of the HTD spectrum peaks.
4. Mark these on a mask.
5. Enlarge the points to regions that cover most of the energy.
6. Blur the mask for used as a notch filter.
7. Multiply the Fourier transform of the image by the mask.
8. Take the inverse Fourier transform of the result.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Remove HTD Distortion Selectively}

2.

6.
3.

HTDlocs \(=499297\) 545320
\(\begin{array}{ll}571 & 358 \\ 569 & 400\end{array}\)
\(\begin{array}{ll}569 & 400 \\ 542 & 438\end{array}\)
\(\begin{array}{ll}542 & 438 \\ 493 & 458\end{array}\)
\(\begin{array}{ll}493 & 458 \\ 439 & 457\end{array}\)
\(\begin{array}{ll}439 & 457 \\ 393 & 434\end{array}\)
367396
\(\begin{array}{ll}369 & 354 \\ 396 & 316\end{array}\)
396316
45296

7.
4.

1. Read in image; 2. Compute power spectrum; 3. Locate HTD frequency components; 4. Mark locs on a mask;
5. Enlarge points to regions; 6. Blur the mask; 7. Multiply FT of image by mask; 8. Take inverse FT of result;

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Notch Filtering of Halftone Dot Distortion}

original

log power spectrum

\section*{EECE/CS 253 Image Processing}

Vanderbilt University School of Engineering

\section*{Notch Filtering}

Since not much distortion was removed, these must be subharmonics of the true dot frequencies

frequency masked 1

log power spectrum

\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Notch Filtering}

The outer ring are the actual HTD frequencies. Can we do any better?

frequency masked 2

log power spectrum

\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{Notch Filtering}

Not much. The harmonics contribute little energy to the image.

frequency masked 3

log power spectrum

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Notch Filter Difference Images}

original

difference

frequency masked 1
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Notch Filter Difference Images}

frequency masked 2

difference

original
middle gray \(=0\), normalized

EECE/CS 253 Image Processing Vanderbilt University School of Engineering

\section*{Notch Filter Difference Images}

original

difference

frequency masked 3
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Notch Filter Difference Images}

frequency masked 1

difference
middle gray \(=0\), normalized

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Notch Filter Difference Images}

frequency masked 2

difference

frequency masked 3
middle gray \(=0\), normalized

\section*{EECEICS 253 Image Processing}

\section*{Lecture Notes on Mathematical Morphology: The Median Filter}

\section*{Richard Alan Peters II}

Department of Electrical Engineering and
Computer Science

Fall Semester 2011

\section*{The Median Filter}

। Returns the median value of the pixels in a neighborhood
, Is non-linear
। Is a morphological filter
, Is similar to a uniform blurring filter which returns the mean value of the pixels in a neighborhood of a pixel
। Unlike a mean value filter the median tends to preserve step edges


\section*{Median Filter: General Definition}
\[
\operatorname{med}\{\mathbf{I}, \mathbf{Z}\}(\mathbf{p})=\operatorname{median}_{\mathbf{q} \in \operatorname{supp}(\mathbf{Z}+\mathbf{p})}\{\mathbf{I}(\mathbf{q})\}
\]

This can be computed as follows:
1. Let \(\mathbf{I}\) be a monochrome (1-band) image.
2. Let \(\mathbf{Z}\) define a neighborhood of arbitrary shape.
3. At each pixel location, \(\mathbf{p}=(r, c)\), in \(\mathbf{I} \ldots\)
4. ... select the \(n\) pixels in the \(\mathbf{Z}\)-neighborhood of \(\mathbf{p}\),
5. ... sort the \(n\) pixels in the neighborhood of \(\mathbf{p}\), by value, into a list \(L(j)\) for \(j=1, \ldots, n\).
6. The output value at \(\mathbf{p}\) is \(L(m)\), where \(m=\lfloor n / 2\rfloor+1\).

\section*{Median Filter: General Definition}


\section*{A Noisy Step Edge}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurred Noisy 1D Step Edge}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Blurred Noisy 1D Step Edge}


\section*{Median Filtered Noisy 1D Step Edge}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Median Filtered Noisy 1D Step Edge}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Median vs. Blurred}


\section*{Median vs. Blurred}


\section*{Median vs. Blurred}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Median Filtering of Binary Images}


Noisy
Original

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Median Filtering of Binary Images}


Median Filtered Noisy


Original

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}


Noisy
Original

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}


Noisy


Noisy

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}

\(3 \times 3\)-blur x 1

\(3 \times 3\)-median x 1

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}

\(3 \times 3\)-blur x 2


3x3-median x 2

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}


3x3-blur x 3

\(3 \times 3\)-median x 3

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}

\(3 \times 3\)-blur x 4

\(3 \times 3\)-median x 4

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}

\(3 \times 3\)-blur x 5


3x3-median x 5

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}

\(3 \times 3\)-blur x 10

\(3 \times 3\)-median x 10

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Filtering of Grayscale Images}


Noisy


Noisy

\section*{Limit and Root Images}

Fact: if you repeatedly filter an image with the same blurring filter or median filter, eventually the output does not change. That is, let
\[
\begin{aligned}
& \mathbf{I}[* \mathbf{h}]^{k} \equiv(((\mathbf{I} * \mathbf{h}) * \mathbf{h}) \cdots * \mathbf{h}), \quad k \text { times, and } \\
& \mathbf{I}[\operatorname{med} \mathbf{Z}]^{k} \equiv(((\mathbf{I} \operatorname{med} \mathbf{Z}) \operatorname{med} \mathbf{Z}) \cdots \operatorname{med} \mathbf{Z}), \quad k \text { times. }
\end{aligned}
\]

Then
\[
\begin{aligned}
& \lim _{k \rightarrow \infty} \mathbf{I}[* \mathbf{h}]^{k}=\mathbf{I}[* \mathbf{h}]^{n}=\mathbf{I}_{0}, \quad \text { and } \\
& \lim _{k \rightarrow \infty} \mathbf{I}[\operatorname{med} \mathbf{Z}]^{k}=\mathbf{I}[\operatorname{med} \mathbf{Z}]^{m}=\mathbf{I}_{\mathrm{r}},
\end{aligned}
\]
where \(n\) and \(m\) are integers \((<\infty), \mathbf{I}_{0}\) is a single-valued image and \(\mathbf{I}_{\mathrm{r}}\) is called the median root of \(\mathbf{I}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Limit and Root Images}

\(3 \times 3\)-blur x 10

\(3 \times 3\)-median x 10

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Limit and Root Images}

\(3 \times 3\)-blur x \(\mathrm{n} \rightarrow \infty\)

\(3 \times 3\)-median root

\section*{Median Filter Algorithm in Matlab}
```

function D = median_filt(I,SE,origy,origx)
[R,C] = size(I); % assumes 1-band image
[SER,SEC] = size(SE); % SE < 0 m not in nbhd
N = sum(sum(SE>=0)); % no. of pixels in nbhd
A = -ones(R+SER-1,C+SEC-1,N); % accumulator
n=1; % copy I into band n of A for nbhd pix n
for j = 1 : SER % neighborhood is def'd in SE
for i = 1 : SEC
if SE(j,i) >= 0 % then is a nbhd pixel
A(j:(R+j-1),i:(C+i-1),n) = I;
n=n+1; % next accumulator band
end
end
end
% pixel-wise median across the bands of A
A = shiftdim(median(shiftdim(A,2)),1);
D = A( origy:(R+origy-1) , origx:(C+origx-1) );
return;

```

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Vector Median Filter}

A vector median filter selects from among a set of vectors, the one vector that is closest to all the others.

That is, if \(S\) is a set of vectors, in \(\mathbb{F}^{n}\) the median, \(\overline{\mathbf{v}}\), is
\[
\overline{\mathbf{v}}=\underset{k \neq j}{\arg \min }\left\{\left\|\mathbf{v}_{k}-\mathbf{v}_{j}\right\| \mid \mathbf{v}_{k}, \mathbf{v}_{j} \in S\right\} .
\]

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Color Median Filter}

If we let \(\mathbb{F}^{n}=\mathbb{R}^{3}\) then the vector median can be used as a color median filter.
(a) original image
(b) image (a) with sparse noise
(c) image (b) color median filtered
(d) image (c) color median filtered

Median filter performed on \(3 \times 3\) nbhd.


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering The output color at ( \(r, c\) ) is

\section*{Color Median Filter}


EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

The output color at ( \(r, c\) ) is

\section*{Color Median Filter}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering The output color at ( \(r, c\) ) is

\section*{Color Median Filter}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering The output color at ( \(r, c\) ) is

\section*{Color Median Filter}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Median Filter}


Jim Woodring - A Warm Shoulder www.jimwoodring.com


Sparse noise, 32\% coverage in each band

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Median Filter}

\(3 \times 3\) color median filter applied once

\(3 \times 3\) color median filter applied twice

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Color Median Filter}


Sparse noise, 32\% coverage in each band


Jim Woodring - A Warm Shoulder www.jimwoodring.com

\title{
EECE/CS 253 Image Processing
}

\section*{Vanderbilt University School of Engineering}

\section*{Color Median Filter}

Absolute differences displayed as negatives to enhance visibility

( \(3 \times 3 \mathrm{CMF}^{2}\) of noisy) - original

\(\left(3 \times 3 \mathrm{CMF}^{2}\right.\) of noisy \()-\left(3 \times 3 \mathrm{CMF}^{2}\right.\) of original \()\)

\section*{CMF vs. Standard Median on Individual Bands}

A color median filter has to compute the distances between all the color vectors in the neighborhood of each pixel. That's expensive computationally.

Q: Why not simply take the 1-band median of each color band individually?
A: The result at a pixel could be a color that did not exist in the pixel's neighborhood in the input image. The result is not the median of the colors - it is the median of the intensities of each color band treated independently.

Q: Is that a problem?
A: Maybe. Maybe not. It depends on the application. It may make little difference visually. If the colors need to be preserved, it could be problematic.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{CMF vs. Standard Median on Individual Bands}


Jim Woodring - A Warm Shoulder www. jimwoodring.com


Sparse noise, \(32 \%\) coverage in each band

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{CMF vs. Standard Median on Individual Bands}

\(3 \times 3\) color median filter applied once

\(3 \times 3\) color median filter applied twice

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{CMF vs. Standard Median on Individual Bands}

\(3 \times 3\) median filter applied to each band once

\(3 \times 3\) median filter applied to each band twice

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{CMF vs. Standard Median on Individual Bands}


Sparse noise, 32\% coverage in each band


Jim Woodring - A Warm Shoulder www.jimwoodring.com

\section*{CMF vs. Standard Median on Individual Bands}


Fraction of pixels in CMF² noisy image identical to original: 0.29


Fraction of pixels in CMF² noisy image identical to \(\mathrm{CMF}^{2}\) original: 0.43


Fraction of pixels in \(\mathrm{MF}^{2}\) noisy image identical to original: 0.14


Fraction of pixels in \(\mathrm{MF}^{2}\) noisy image identical to \(\mathrm{MF}^{2}\) original: 0.28

\title{
EECE\CS 253 Image Processing
}

\section*{Lecture Notes on Mathematical Morphology: Binary Images}

\section*{Richard Alan Peters II}

\title{
Department of Electrical Engineering and \\ Computer Science
}

Fall Semester 2011

\section*{What is Mathematical Morphology?}

\section*{Jt is:}

। nonlinear,
। built on Minkowski set theory,
। part of the theory of finite lattices, for image analysis based on shape, extremely useful, yet not often used.

\section*{Uses of Mathematical Morphology}

। image enhancement
- image segmentation
- image restoration
- edge detection

। texture analysis
। particle analysis
। feature generation
। skeletonization
- shape analysis
, image compression
। component analysis
- curve filling

। general thinning
, feature detection
। noise reduction
। space-time filtering

EECE/CS 253 Image Processing

\section*{Notation and Image Definitions}

An image is a mapping, \(\mathbf{I}\), from a set, \(S_{\mathrm{p}}\), of pixel coordinates to a set, \(G\), of values such that for every coordinate vector, \(\mathbf{p}=(r, c)\) in \(S_{\mathrm{p}}\), there is a value \(\mathrm{I}(\mathbf{p})\) drawn from \(G . S_{\mathrm{P}}\) is also called the image plane.

A binary image has only 2 values. That is, \(\mathrm{G}=\left\{v_{\mathrm{fg}}, v_{\mathrm{bg}}\right\}\), where \(v_{\mathrm{fg}}\), is called the foreground value and \(v_{\mathrm{bg}}\) is called the background value.

Often, the foreground value is \(v_{\mathrm{fg}}=0\), and the background is \(v_{\mathrm{bg}}=-\infty\). Other possibilities are \(\left\{v_{\mathrm{fg}}, v_{\mathrm{bg}}\right\}=\{0, \infty\},\{0,1\},\{1,0\},\{0,255\}\), and \(\{255,0\}\).

In this lecture we assume that \(\left\{v_{\mathrm{fg}}, v_{\mathrm{bg}}\right\}=\{255,0\}\), although the fg is often displayed in different colors for contrast.

\section*{Notation and Image Definitions}

The foreground of binary image \(\mathbf{I}\) is
\[
\operatorname{FG}\{\mathbf{I}\}=\left\{\mathbf{I}(\mathbf{p}), \mathbf{p}=(r, c) \in S_{\mathrm{P}} \mid \mathbf{I}(\mathbf{p})=v_{\mathrm{fg}}\right\},
\]
i.e. the set of locations, \(\mathbf{p}\), where \(\mathrm{I}(\mathbf{p})=v_{\mathrm{fg}}\). Similarly, the background is
\[
\operatorname{BG}\{\mathbf{I}\}==\left\{\mathbf{I}(\mathbf{p}), \mathbf{p}=(r, c) \in S_{\mathrm{P}} \mid \mathbf{I}(\mathbf{p})=v_{\mathrm{bg}}\right\} .
\]

Note that
\[
\operatorname{FG}\{\mathbf{I}\} \cup B G\{\mathbf{I}\}=\mathbf{I} \text { and } \operatorname{FG}\{\mathbf{I}\} \cap \operatorname{BG}\{\mathbf{I}\}=\varnothing,
\]
and that
\[
\operatorname{BG}\{\mathbf{I}\}=\{\operatorname{FG}\{\mathbf{I}\}\}^{\mathrm{C}} \text { and } \operatorname{FG}\{\mathbf{I}\}=\{\operatorname{BG}\{\mathbf{I}\}\}^{\mathrm{C}} .
\]

The background is the complement of the foreground and vice-versa.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{A Binary Image}


This represents a digital image. Each square is one pixel.

\section*{Support of an Image}

The support of a binary image, \(\mathbf{I}\), is
\[
\operatorname{supp}(\mathbf{I})=\left\{\mathbf{p}=(r, c) \in S_{\mathrm{P}} \mid \mathbf{I}(\mathbf{p})=v_{\mathrm{f} g}\right\} .
\]

That is, the support of a binary image is the set of foreground pixel locations within the image plane.

The complement of the support is, therefore, the set of background pixel locations within the image plane.
\[
\{\operatorname{supp}(\mathbf{I})\}^{\mathrm{C}}=\left\{\mathbf{p}=(r, c) \in S_{\mathrm{P}} \mid \mathbf{I}(\mathbf{p})=v_{\mathrm{b} g}\right\} .
\]


\section*{Structuring Element (SE)}

A structuring element is a small image - used as a moving window - whose support delineates pixel neighborhoods in the image plane.


It can be of any shape, size, or connectivity (more than 1 piece, have holes). In the figure the circles mark the location of the structuring element's origin which can be placed anywhere relative to its support.

\section*{Vanderbilt University School of Engineering}

\section*{Structuring Element}

Let \(\mathbf{I}\) be an image and \(\mathbf{Z}\) a SE.
\(\mathbf{Z}+\mathbf{p}\) means that \(\mathbf{Z}\) is moved so that its origin coincides with location \(\mathbf{p}\) in \(S_{\mathrm{p}}\).

\(\mathbf{Z}+\mathbf{p}\) is the translate of \(\mathbf{Z}\) to location \(\mathbf{p}\) in \(S_{\mathrm{P}}\).
\(Z+\vec{p}\) delineates a neighborhood in \(I\) with respect to \(\vec{p}\).


\section*{Reflected Structuring Elements}

Let \(\mathbf{Z}\) be a SE and let \(\boldsymbol{\Sigma}\) be the square of pixel locations that contains the set \(\{(\mathrm{r}, \mathrm{c}),(-\mathrm{r},-\mathrm{c}) \mid(\mathrm{r}, \mathrm{c}) \in \operatorname{supp}(\mathbf{Z})\}\). Then
\(\check{\mathbf{Z}}(\rho, \chi)=\mathbf{Z}(-\rho,-\chi)\) for all \((\rho, \chi) \in \boldsymbol{\Phi}\).
is the reflected structuring element.
\(\overline{\mathbf{Z}}\) is \(\mathbf{Z}\) rotated by \(180^{\circ}\) around its origin.


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Dilation}


\section*{Dilation of Binary Images}

There are a number of equivalent definitions of dilation. Three of them that apply to binary images are:

The set of all pixel locations, \(\mathbf{p}\), in the
\(\mathbf{I} \oplus B=\left\{\mathbf{p} \in S_{\mathrm{P}} \mid[(\breve{\mathbf{Z}}+\mathbf{p}) \cap \mathbf{I}] \neq \varnothing\right\}\). image plane where the intersection of \(\check{\mathbf{Z}}+\mathbf{p}\) with I is not empty.
\[
\mathbf{I} \oplus \mathbf{Z}=\bigcup_{\mathbf{p} \in \operatorname{supp}\{\mathbf{I}\}}(\mathbf{Z}+\mathbf{p}) .
\]
\[
\mathbf{I} \oplus \mathbf{Z}=\bigcup_{\mathbf{p} \in \operatorname{supp}\{\mathbf{Z}\}}(\mathbf{I}+\mathbf{p}) .
\]

The union of copies of the image, one translated to each pixel location in the support of the SE.

\section*{Vanderbilt University School of Engineering}

\section*{Dilation}

The locus of pixels \(\mathbf{p} \in S_{\mathrm{P}}\) such that \((\breve{\mathbf{Z}}+\mathbf{p}) \cap \mathrm{I} \neq \varnothing\).

dilated image

original / dilation

original image
\[
\mathrm{SE}=\mathrm{Z}_{8}
\]

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Dilation using a Reflected SE}


\section*{Fast Computation of Dilation}

The fastest way to compute binary dilation is to use the union-of-translates-of-the-image definition. That is, use
\[
\mathbf{J}=\mathbf{I} \oplus \mathbf{Z}=\bigcup_{\mathbf{q} \in \mathbf{Z}} \mathbf{I}+\mathbf{q} .
\]

Assume the dimensions of \(\mathbf{I}\) are \(R \times C\), the dimensions of \(\mathbf{Z}\) are \(N \times M\), and Z's origin is offset from the upper left hand corner (ULHC) by \(\rho\) rows and \(\chi\) columns. Allocate a scratch image, \(\mathbf{T}\), that is \((R+N-1) \times(C+M-1)\) and initialized to zeros. Then, for each FG pixel loc ( \(v, u\) ) in \(\mathbf{Z}\) (measured from the ULHC of \(\mathbf{Z}\) ) perform a logical OR between \(\mathbf{I}+(v, u)\) and \(\mathbf{T}\). Put the results in \(\mathbf{T}\). When done, copy to \(\mathbf{J}\) the \(R \times C\) subarray of T starting at ( \(\rho, \chi\) ).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dilation through Image Shifting}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Dilation through Image Shifting}


The red outlines indicate the positions of the features in the original images.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Erosion}


\section*{Erosion of Binary Images}

There are a number of equivalent definitions of erosion. Three of them that apply to binary images are:
\[
\mathbf{I} \ominus \mathbf{Z}=\left\{\mathbf{p} \in S_{\mathbf{p}} \mid \mathbf{Z}+\mathbf{p} \subset \mathbf{I}\right\}
\]

The set of all pixel locations, \(\mathbf{p}\), in the image plane where \(\mathbf{Z}+\mathbf{p}\) is contained in I.
\[
\mathbf{I} \ominus \mathbf{Z}=\bigcap_{\mathbf{p} \in \sup \{\mathbf{I}\}}(\breve{\mathbf{Z}}+\mathbf{p}) . \begin{aligned}
& \text { The intersection of copies of the refl. SE, one translated } \\
& \text { to each pixel location in the support of the image. }
\end{aligned}
\]
\[
\mathbf{I} \ominus \mathbf{Z}=\bigcap_{\mathbf{p} \in \operatorname{supp}\{\overleftarrow{\mathbf{z}}\}}(\mathbf{I}+\mathbf{p}) . \quad \begin{aligned}
& \text { The intersection of copies of the image, one translated } \\
& \text { to each pixel location in the support of the refl. SE. }
\end{aligned}
\]

\section*{Erosion}

The locus of pixels \(\mathbf{p} \in S_{\mathrm{P}}\) such that \(\mathbf{Z}+\mathbf{p} \subset \mathrm{I}\).

eroded image

erosion / original

original image
\[
\mathrm{SE}=\mathrm{Z}_{8}
\]

This is a piece of a larger image. Boundary effects are not apparent

\section*{Fast Computation of Erosion}

The fastest way to compute binary erosion is to use the intersection-of-translates-of-the-image definition. That is, use
\[
\mathbf{J}=\mathbf{I} \ominus \mathbf{Z}=\bigcap_{\mathbf{q} \in \mathbf{Z}} \mathbf{I}+\mathbf{q} .
\]

Assume the dimensions of \(\mathbf{I}\) are \(R \times C\), the dimensions of \(\mathbf{Z}\) are \(N \times M\), and Z's origin is offset from the upper left hand corner (ULHC) by \(\rho\) rows and \(\chi\) columns. Allocate a scratch image, \(\mathbf{T}\), that is \((R+N-1) \times(C+M-1)\) and initialized to \(\mathbf{I}\). Rotate \({ }^{1} \mathbf{Z}\) by \(180^{\circ}\). Then, for each FG pixel loc ( \(v, u\) ) in \(\mathbf{Z}\) (measured from the ULHC of \(\check{\mathbf{Z}}\) ) perform a logical AND between \(\mathbf{I}+(v, u)\) and \(\mathbf{T}\). Put the results in \(\mathbf{T}\). When done, copy to \(\mathbf{J}\) the \(R \times C\) subarray of T starting at ( \(N-\rho, M-\chi\) ).

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Comparison of Erosion and Dilation}
original contains erosion

erosion / original

erosion / original / dilation
dilation contains original

original / dilation
\[
\mathrm{SE}=\mathrm{Z}_{8}
\]

EECE/CS 253 Image Processing

\section*{Erosion from Dilation / Dilation from Erosion}

Dilation and erosion are duals of each other with respect to complementation:
\[
\mathbf{I}^{c} \oplus \breve{\mathbf{Z}}=\{\mathbf{I} \ominus \mathbf{Z}\}^{c} \quad \text { and } \quad \mathbf{I}^{c} \ominus \breve{\mathbf{Z}}=\{\mathbf{I} \oplus \mathbf{Z}\}^{c} .
\]

That is, dilation with the reflected SE of the complement of a binary image is the complement of the erosion. Erosion with the reflected SE of the complement of the image is the complement of the dilation. It follows that,
\[
\mathbf{I} \ominus \mathbf{Z}=\left\{\mathbf{I}^{c} \oplus \breve{\mathbf{Z}}\right\}^{c} \quad \text { and } \quad \mathbf{I} \oplus \mathbf{Z}=\left\{\mathbf{I}^{c} \ominus \breve{\mathbf{Z}}\right\}^{c},
\]
erosion can be performed with dilation and vice versa. That implies that only one or the other must be implemented directly.

\section*{Opening and Closing}

Opening is erosion by \(\mathbf{Z}\) followed by dilation by \(\mathbf{Z}\).
\[
\mathbf{I} \circ \mathbf{Z}=(\mathbf{I} \ominus \mathbf{Z}) \oplus \mathbf{Z}
\]

The opening is the best approximation of the image FG that can be made from copies of the SE, given that the opening is contained in the original. \(\mathbf{I} \circ \mathbf{Z}\) contains no FG features that are smaller than the SE.

Closing is dilation by \(\check{\mathbf{Z}}\) followed by erosion by \(\check{\mathbf{Z}}\).
\[
\mathbf{I} \bullet \mathbf{Z}=(\mathbf{I} \oplus \breve{\mathbf{Z}}) \ominus \breve{\mathbf{Z}}
\]

The closing is the best approximation of the image BG that can be made from copies of the SE, given that the closing is contained in the image BG. \(\mathbf{I} \cdot \mathbf{Z}\) contains no BG features that are smaller than the SE .

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Opening}


\section*{Opening is Erosion Followed by Dilation}
erode the original

erosion / original
dilate the erosion

erosion / opening
dilated erosion

opening / original
SE \(=\mathrm{Z}_{8}\)

\section*{Opening is Erosion Followed by Dilation}
original image

original
\(\mathrm{SE}=\mathrm{Z}_{8}\)
This is a piece of a larger image. Boundary effects are not apparent

\section*{Opening}

The union of translates of \(\mathbf{Z}\) such that \(\mathbf{Z}+\mathbf{p} \subset \mathbf{I}\).

open image

opening / original

original image

The opening of \(\mathbf{I}\) by \(\mathbf{Z}\) is the best approximation of \(\mathbf{I}\) that can be made by taking the union of translated copies of \(\mathbf{Z}\), subject to the constraint that the opening be contained by the original image.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Closing}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Closing is Dilation Followed by Erosion \({ }^{1}\)}
original image

original / dilation
\(\mathrm{SE}=\mathrm{Z}_{8}\)
erode the dilation

closing / dilation
to get the closing

closing / original
\({ }^{1}\) using the reflected SE, Ž

\section*{Closing is Dilation Followed by Erosion \({ }^{1}\)}
original image

original
\(\mathrm{SE}=\mathrm{Z}_{8}\)
dilated image

dilation
eroded dilation

closing
\({ }^{1}\) using the reflected SE, Ž

\section*{Duality Relationships}

Erosion in terms of dilation:
Dilation in terms of erosion:
Opening in terms of closing:
Closing in terms of opening:
\[
\begin{aligned}
& \mathbf{I} \ominus \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \oplus \breve{\mathbf{Z}}\right]^{\mathrm{C}} \\
& \mathbf{I} \oplus \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \ominus \breve{\mathbf{Z}}\right]^{\mathrm{C}} \\
& \mathbf{I} \circ \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \bullet \mathbf{Z}\right]^{\mathrm{C}} \\
& \mathbf{I} \bullet \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \circ \mathbf{Z}\right]^{\mathrm{C}}
\end{aligned}
\]
\(\mathbf{I}^{\mathbf{C}}\) is the complement of \(\mathbf{I}\) and \(\check{\mathbf{Z}}\) is the reflected SE.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Binary Ops with Asymmetric SEs}

"L" shaped SE
O marks origin
 Background: black pixels

Cross-hatched pixels are indeterminate.

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Border Effects}

Erosion \& Dilation
Since morph. ops. are neighborhood ops., there is a band of pixels around the border of the resultant image where the values are indeterminate.

The actual values of pixels in the indet. region depend on the specific algorithm used.


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\title{
Border Effects
}

Opening \& Closing

Since opening \& closing iterate erosion \& dilation, the boundaries of the deterministic region are \(2 x\) as far from the image border as are those of erosion or dilation.


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original

erosion by square

4-conn inside bdry

difference

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original

erosion by plus

8-conn inside bdry

difference

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original

dilation by plus

8-conn outside bdry

difference

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original

dilation by square

4-conn outside bdry

difference

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original
erosion by square is

in erosion by plus

8-conn inside bdry is

in 4-conn inside bdry

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}

original
dilation by plus is

in dilation by square

8 -conn outside bdry is

in 4-conn outside bdry

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Boundary Extraction}
inside boundaries


8-bdry/4-bdry/orig
are disjoint from

all 4 boundaries
outside boundaries

orig/8-bdry/4-bdry

This is a piece of a larger image. Boundary effects are not apparent

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Conditional Dilation}
original image

mask over original

dilation inside a mask
dilated original


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Conditional Dilation}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Connected Component Extraction}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Connected Component Extraction}


EECE/CS 253 Image Processing

\section*{Binary Reconstruction}

Used after opening to grow back pieces of the original image that are connected to the opening.

original

opened

reconstructed

Removes of small regions that are disjoint from larger objects without distorting the small features of the large objects.

\section*{Algorithm for Binary Reconstruction}
\[
\begin{aligned}
& \text { 1. } \mathrm{J}=\mathrm{I} \circ \mathrm{Z} \text {, where } \mathrm{Z} \text { is any } \mathrm{SE} \text {. } \\
& \text { 2. } \mathrm{T}=\mathrm{J}, \\
& \text { 3. } \mathrm{J}=\mathrm{J} \oplus \mathrm{Z}_{\mathrm{k}} \text {, where } \mathrm{k}=4 \text { or } \mathrm{k}=8 \text {, } \\
& \text { 4. } \mathrm{J}=\mathrm{I} \text { AND } \mathrm{J} \text {, [Take only those pixels from } \mathrm{J} \text { that are also in } I \text {.] } \\
& \text { 5. if } \mathrm{J} \neq \mathrm{T} \text { then go to } 2 \text {, } \\
& \text { 6. else stop; [ J is the reconstructed image.] }
\end{aligned}
\]

This is the same as connected component extraction with the opened image, J, containing the tags. The choice of \(Z_{k}\) determines the connectivity of the result.

1. \(J=I O Z\), wher \(\in \quad \mathrm{K}=\operatorname{ReconBin}(\mathrm{I}, \mathrm{J}, \mathrm{z})\);
2. \(\mathrm{T}=\mathrm{J}\),

Then the algorithm starts at step 2.
3. \(\mathrm{J}=\mathrm{J} \oplus \mathrm{Z}_{\mathrm{k}}\), where \(\mathrm{k}=4\) or \(\mathrm{k}=8\),
4. \(\mathbf{J}=\mathrm{I}\) AND \(\mathbf{J}\), [Take only those pixels from J that are also in I.] 5. if \(\mathrm{J} \neq \mathrm{T}\) then go to 2 , 6. else stop; [ \(J\) is the reconstructed image.]

This is the same as connected component extraction with the opened image, J, containing the tags. The choice of \(Z_{k}\) determines the connectivity of the result.

\section*{Skeletonization}

Let \(\operatorname{Skel}(\mathbf{I}, r)\) be the set of pixels in \(\mathbf{I}\) such that if \(\mathbf{p} \subseteq \operatorname{Skel}(\mathbf{I}, r)\) then \(D_{p}(r)\), is a maximal disk of radius \(r\) in \(\mathbf{I}\). That is, \(\operatorname{Skel}(\mathbf{I}, r)\) is the locus of centers of maximal disks of radius \(r\) in \(\mathbf{I}\). Then
\[
S=\bigcup_{r=0}^{\infty} \operatorname{Skel}(\mathbf{I}, r)
\]

That is, the skeleton of \(\mathbf{I}\) is the union of all the sets of centers of maximal disks.

Note that for any actual image \(\mathbf{I}\), the union will not be infinite, since \(\mathbf{I}\) is bounded (not infinite in extent).

\title{
Skeletonization
}

Original shape


Raw skeleton (red)

is the locus of centers of maximal disks.

Pruned and connected

skeleton

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Skeletonization: Maximal Disks}
non maximal "disks"

"disks" are squares
maximal disks (red)

non max disks (blue)
non max \& max disks

over skeleton

The maximal disk at pixel loc \(\mathbf{p}\) is the largest disk in the fg that includes \(\mathbf{p}\).

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Skeletonization: Maximal Disks}
non maximal "disks"

"disks" are squares
maximal disks (red)

non max disks (blue)
non max \& max disks

over skeleton

The maximal disk at pixel loc \(\mathbf{p}\) is the largest disk in the fg that includes \(\mathbf{p}\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Computation of the Skeleton}
- \(\mathrm{SE}=\mathrm{Z}_{8}=\) 膘 \(=\mathrm{Sq}(3)\)
- \(\mathrm{n}=0: \mathrm{SE}=1\) pixel
- \(\mathrm{n}=1: \mathrm{SE}=\mathrm{Sq}(3)\)
- \(\mathrm{n}=2: \mathrm{SE}=\mathrm{Sq}(5)\)
- \(\mathrm{n}=3: \mathrm{SE}=\mathrm{Sq}(7)\)

Note that the result is disconnected and has spurious points.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Skeletonization: Delete Spurious Pixels}

has spurious pixels
def. spurious pixels as

conn. comp. of < 3 pix.
pruned skeleton

raw less spurious

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Skeletonization: Reconnect Components}

pruned skel. dilated.

2 other components

of pruned skel dilated.

Intersection of

dilated components.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Skeletonization}
raw skeleton

pruned skeleton

reconnected skeleton


\section*{EECE\CS 253 Image Processing}

Lecture Notes on Mathematical Morphology: Grayscale Images

\section*{Richard Alan Peters II}

Department of Electrical Engineering and
Computer Science

Fall Semester 2011

\section*{Grayscale Morphology}

Grayscale morphology is a multidimensional generalization of the binary operations. Binary morphology is defined in terms of set-inclusion of pixel sets. So is the grayscale case, but the pixel sets are of higher dimension. In particular, standard \(R \times C\), 1band intensity images and the associated structuring elements are defined as 3-D solids wherein the \(3^{\text {rd }}\) axis is intensity and set-inclusion is volumetric.

(a) binary,
(b) \& (c) grayscale

\section*{Extended Real Numbers}

Let \(\mathbb{R}\) represent the real numbers.
Define the extended real numbers, \(\mathbb{R}^{*}\), as the real numbers plus two symbols, \(-\infty\) and \(\infty\) such that
\[
-\infty<x<\infty
\]
for all numbers \(x \in \mathbb{R}\).
That is if \(x\) is any real number, then \(\infty\) is always greater than \(x\) and \(-\infty\) is always less than \(x\). Moreover,
\[
x+\infty=\infty, \quad x-\infty=-\infty, \quad \infty-\infty=0
\]
for all numbers \(x \in \mathbb{R}\).

\section*{Real Images}

In mathematical morphology a real image, \(\mathbf{I}\), is defined as a function that occupies a volume in a Euclidean vector space. I comprises a set, \(S_{\mathrm{p}}\), of coordinate vectors (or pixel locations), \(\mathbf{p}\), in an \(n\)-dimensional vector space \(\mathbb{R}^{n}\). Associated with each \(\mathbf{p}\) is a value from \(\mathbb{R}^{*}\). The set of pixel locations together with their associated values form the image - a set in \(\mathbb{R}^{n+1}\) :
\[
\mathbf{I}=\left\{[\mathbf{p}, \mathbf{I}(\mathbf{p})] \mid \mathbf{p} \in S_{\mathbf{p}} \subseteq \mathbb{R}^{n}, \mathbf{I}(\mathbf{p}) \in \mathbb{R}^{*}\right\}
\]

Thus, a conventional, 1 -band, \(R \times C\) image is a 3D structure with \(S_{\mathrm{p}} \subset \mathbb{R}^{2}\) and \(\mathbf{I}(\mathbf{p}) \in \mathbb{R}\). By convention in the literature of \(\mathrm{MM}, S_{\mathrm{p}} \equiv \mathbb{R}^{\mathrm{n}}\), a real image is defined over all of \(\mathbb{R}^{\mathrm{n}}\).

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Support of an Image}

The support of a real image, \(\mathbf{I}\), is
\[
\operatorname{supp}(\mathbf{I})=\left\{\mathbf{p} \in \mathbb{R}^{n} \mid \mathbf{I}(\mathbf{p}) \in \mathbb{R}\right\} .
\]

That is, the support of a real image is the set pixel locations in \(\mathbb{R}^{\mathrm{n}}\) such that
\[
\mathbf{I}(\mathbf{p}) \neq-\infty \text { and } \mathbf{I}(\mathbf{p}) \neq \infty
\]

The complement of the support is, therefore, the set of pixel locations in \(\mathbb{R}^{\mathrm{n}}\) where
\[
\mathbf{I}(\mathbf{p})=-\infty \text { or } \mathbf{I}(\mathbf{p})=\infty
\]


EECE/CS 253 Image Processing

\section*{Grayscale Images}

If over its support, \(\mathbf{I}\) takes on more than one real value, then \(\mathbf{I}\) is called grayscale.

The object commonly known as a black and white photograph is a grayscale image that has support in a rectangular subset of \(\mathbb{R}^{2}\). Within that region, the image has gray values that vary between black and white. If the intensity of each pixel is plotted over the support plane, then
\[
\mathbf{I}=\{[\mathbf{p}, \mathbf{I}(\mathbf{p})] \mid \mathbf{p} \in \operatorname{supp}(\mathbf{I})\}
\]
is a volume in \(\mathbb{R}^{3}\). In the abstraction of MM we assume the image does exist outside the support rectangle, but that \(\mathbf{I}(\mathbf{p})=-\infty\) there.

\section*{Grayscale Images}

grayscale image


In MM, a 2D grayscale image is treated as a 3D solid in space - a landscape - whose height above the surface at a point is proportional to the brightness of the corresponding pixel.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Representation of Grayscale Images}

image

landscape

Example: grayscale cones

\section*{Set Inclusion in Grayscale Images}

In grayscale morphology, set inclusion depends on the implicit 3D structure of a 2D image. If \(\mathbf{I}\) and \(\mathbf{J}\) are grayscale images then
\[
\mathbf{J} \subseteq \mathbf{I} \Leftrightarrow \operatorname{supp}(\mathbf{J}) \subseteq \operatorname{supp}(\mathbf{I}) \operatorname{AND}\{\mathbf{J}(\mathbf{p}) \leq \mathbf{I}(\mathbf{p}) \mid \mathbf{p} \in \operatorname{supp}(\mathbf{J})\} .
\]

That is \(\mathbf{J} \subseteq \mathbf{I}\) if and only if the support of \(\mathbf{J}\) is contained in that of \(\mathbf{I}\) and the value of \(\mathbf{J}\) is nowhere greater than the value of \(\mathbf{I}\) on the support of \(\mathbf{J}\).


EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{Recall: Binary Structuring Element (SE)}

Let \(\mathbf{I}\) be an image and \(\mathbf{Z}\) a SE.
\(\mathbf{Z}+\mathbf{p}\) means that \(\mathbf{Z}\) is moved so that its origin coincides with location \(\mathbf{p}\) in \(S_{\mathrm{p}}\).

Image, \(I\).
Origin is marked 0. Origin is marked o.

\(\mathbf{Z}+\mathbf{p}\) is the translate of \(\mathbf{Z}\) to location \(\mathbf{p}\) in \(S_{\mathrm{P}}\).

The set of locations in the image delineated by \(\mathbf{Z}+\mathbf{p}\) is called the Z-neighborhood of \(\mathbf{p}\) in \(\mathbf{I}\) denoted \(\mathbf{N}\{\mathbf{I}, \mathbf{Z}\}(\mathbf{p})\).


EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Grayscale Structuring Elements}

A grayscale structuring element is a small image that delineates a volume at each pixel \([\mathbf{p}, \mathbf{I}(\mathbf{p})]\) through out the image volume.


Translation of a flat SE on its support plane and in gray value.


SE Translation: \(\times\) marks the location of the structuring element origin.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Structuring Elements}


Translation of a flat SE on its support plane and in gray value.

If \(\mathbf{Z}=[\mathbf{p}, \mathbf{Z}(\mathbf{p})]\) is a structuring element and if \(\mathbf{q}=\left[\mathbf{q}_{\mathrm{s}}, \mathrm{q}_{\mathrm{g}}\right]\) is a pixel [location, value] then \(\mathbf{Z}+\mathbf{q}=\left[\mathbf{p}+\mathbf{q}_{\mathrm{s}}, \mathbf{Z}(\mathbf{p})+\mathrm{q}_{\mathrm{g}}\right]\) for all \(\mathbf{p} \in \operatorname{supp}\{\mathbf{Z}\}\).

\section*{Reflected Structuring Elements}


\section*{Grayscale Morphology: Basic Operations}


EECE/CS 253 Image Processing

\section*{Dilation: General Definition}

The dilation of image \(\mathbf{I}\) by structuring element \(\mathbf{Z}\) at coordinate \(\mathbf{p} \in \mathbb{R}^{\mathrm{n}}\) is defined by
\[
[\mathbf{I} \oplus \mathbf{Z}](\mathbf{p})=\max _{\mathbf{q} \in \operatorname{supp}(\mathbf{Z}+\mathbf{p})}\{\mathbf{I}(\mathbf{q})+\mathbf{Z}(\mathbf{p}-\mathbf{q})\}=\max _{\mathbf{q} \in \operatorname{supp}(\overline{\mathbf{Z}}+\mathbf{p})}\{\mathbf{I}(\mathbf{q})-\breve{\mathbf{Z}}(\mathbf{q}-\mathbf{p})\} .
\]

This can be computed as follows:
1. Translate \(\check{\mathbf{Z}}\) to \(\mathbf{p}\).
2. Trace out the \(\check{\mathbf{Z}}\)-neighborhood of \(\mathbf{I}\) at \(\mathbf{p}\).
3. Let \(\mathbf{p}\) be the origin of \(\mathbf{I}\) temporarily during the operation
4. Compute the set of numbers
\[
\mathscr{D}=\{\mathbf{I}(\mathbf{q})+\mathbf{Z}(-\mathbf{q}) \mid \mathbf{q} \in \operatorname{supp}(\breve{\mathbf{Z}})\}=\{\mathbf{I}(\mathbf{q})-\breve{\mathbf{Z}}(\mathbf{q}) \mid \mathbf{q} \in \operatorname{supp}(\breve{\mathbf{Z}})\} .
\]
5. The output value, \([\mathbf{I} \oplus \mathbf{Z}](\mathbf{p})\), is the maximum value in the set, \(\mathfrak{D}\).

\section*{Fast Computation of Dilation}

The fastest way to compute grayscale dilation is to use the translates-of-the-image definition of dilation. That is, use
\[
\mathbf{J}=\mathbf{J} \oplus \mathbf{Z}=\max _{\mathbf{q} \in \operatorname{supp}\{\mathbf{Z}\}}\{[\mathbf{I}+\mathbf{q}]+\mathbf{Z}(\mathbf{q})\} .
\]

Note that if \(\mathbf{Z}\) is flat -- all its foreground elements are 0 -then step (3) is unnecessary. Then it is a maximum filter.

That is,
(1) Make a copy of \(\mathbf{I}\) for each foreground element, \(\mathbf{q}\), in \(\mathbf{Z}\).
(2) Translate the \(\mathbf{q}\) th copy so that its ULHC (origin) is at position \(\mathbf{q}\) in \(\mathbf{Z}\).
(3) Add \(\mathbf{Z}(\mathbf{q})\) to every pixel in the \(\mathbf{q}\) th copy.
(4) Take the pixelwise maximum of the resultant stack of images.
(5) Copy out the result starting at the SE origin in the maximum image.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Grayscale Morphology: Dilation}

dilation

dilation over original
\(S E, Z\), is a flat disk the size of the tops of the truncated cones.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Dilation}

\(S E, Z\), is a flat disk.

\section*{Erosion: General Definition}

The erosion of image \(\mathbf{I}\) by structuring element \(\mathbf{Z}\) at coordinate \(\mathbf{p} \in \mathbb{R}^{\mathrm{n}}\) is defined by
\[
[\mathbf{I} \ominus \mathbf{Z}](\mathbf{p})=\min _{\mathbf{q} \in \operatorname{supp}(\mathbf{Z}+\mathbf{p})}\{\mathbf{I}(\mathbf{q})-\mathbf{Z}(\mathbf{q}-\mathbf{p})\} .
\]

This can be computed as follows:
1. Translate \(\mathbf{Z}\) to \(\mathbf{p}\).
2. Trace out the \(\mathbf{Z}\)-neighborhood of \(\mathbf{I}\) at \(\mathbf{p}\).
3. Let \(\mathbf{p}\) be the origin of \(\mathbf{I}\) temporarily during the operation
4. Compute the set of numbers
\[
\mathcal{E}=\{\mathbf{I}(\mathbf{q})-\mathbf{Z}(\mathbf{q}) \mid \mathbf{q} \in \operatorname{supp}(\mathbf{Z})\} .
\]
5. The output value, \([\mathbf{I} \ominus \mathbf{Z}](\mathbf{p})\), is the minimum value in the set, \(\mathfrak{E}\).

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Grayscale Morphology: Erosion}


SE, \(Z\), is the same flat disk as used for the dilation on page 19.

\section*{Fast Computation of Erosion}

The fastest way to grayscale erosion is to create a stack of images translated to minus the values of the reflected SE then take the pixelwise minimum:
\[
\begin{aligned}
& \mathbf{J}=\mathbf{I} \ominus \mathbf{Z}=\min _{\mathbf{q} \in \mathbf{Z}}\{[\mathbf{I}+\mathbf{q}]+\breve{\mathbf{Z}}(\mathbf{q})\} \\
& \breve{\mathbf{Z}}=\left\{-\mathbf{Z}(-\mathbf{q}) \mid \mathbf{q} \in \mathbb{R}^{2}\right\}
\end{aligned}
\]
```

Note that if Z is
symmetric and if all
the foreground
elements are 0, then
Z=Z and step (3) is
unnecessary. Then it
is a minimum filter

```

That is, (1) make a copy of \(\mathbf{I}\) for each foreground element, \(\mathbf{q}\), in \(\check{\mathbf{Z}}\). (Note that if \(\mathbf{q}\) is a foreground element in \(\check{\mathbf{Z}}\) then - \(\mathbf{q}\) is a foreground element in \(\mathbf{Z}\).) (2) Translate each copy so that its ULHC (origin) is at position \(\mathbf{q}\) in \(\check{\mathbf{Z}}\) (or - \(\mathbf{q}\) in \(\mathbf{Z}\) ). (3) Then add \(\check{\mathbf{Z}}(\mathbf{q})\) (or subtract \(\mathbf{Z}(-\mathbf{q})\) ) to every pixel in the \(\mathbf{q}\) th copy. Finally, (4) take the pixelwise minimum of the resultant stack of images.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Erosion}

\(S E, Z\), is a flat disk.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Opening}

opening: erosion then dilation

opened \& original
\(S E, Z\), is a flat disk
the size of the tops of the truncated cones.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Grayscale Morphology: Opening}

erosion \& opening

erosion \& opening \& original
\(S E, Z\), is a flat disk
the size of the tops of the truncated cones.

\section*{Opening and Closing}

Opening is erosion by \(\mathbf{Z}\) followed by dilation by \(\mathbf{Z}\).
\[
\mathbf{I} \odot \mathbf{Z}=(\mathbf{I} \ominus \mathbf{Z}) \oplus \mathbf{Z}
\]

The opening is the best approximation of the image FG that can be made from copies of the SE, given that the opening is contained in the original. \(\mathbf{I} \circ \mathbf{Z}\) contains no FG features that are smaller than the SE.

Closing is dilation by \(\check{\mathbf{Z}}\) followed by erosion by \(\check{\mathbf{Z}}\).
\[
\mathbf{I} \bullet \mathbf{Z}=(\mathbf{I} \oplus \breve{\mathbf{Z}}) \ominus \breve{\mathbf{Z}}
\]

The closing is the best approximation of the image BG that can be made from copies of the SE, given that the closing is contained in the image BG. \(\mathbf{I} \cdot \mathbf{Z}\) contains no BG features that are smaller than the SE.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Opening}

\(S E, Z\), is a flat disk.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Closing}

closing: dilation then erosion

closing \& original

SE, \(Z\), is the same flat disk as used for the dilation on page 19.

EECE/CS 253 Image Processing

\author{
Vanderbilt University School of Engineering
}

\section*{Grayscale Morphology: Closing}

dilation over closing

dilation \& closing \& original
\(S E, Z\), is a flat disk the size of the tops of the truncated cones.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Closing}

\(S E, Z\), is a flat disk.

\section*{Duality Relationships}

Erosion in terms of dilation:
Dilation in terms of erosion:
\[
\mathbf{I} \ominus \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \oplus \breve{\mathbf{Z}}\right]^{\mathrm{C}}
\]

Opening in terms of closing:
Closing in terms of opening:
\[
\mathbf{I} \oplus \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \ominus \breve{\mathbf{Z}}\right]^{\mathrm{C}}
\]
\[
\mathbf{I} \cdot \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \cdot \mathbf{Z}\right]^{\mathrm{C}}
\]
\[
\mathbf{I} \bullet \mathbf{Z}=\left[\mathbf{I}^{\mathrm{C}} \mathbf{Z}_{\mathbf{Z}}\right]^{\mathrm{C}}
\]
\(\mathbf{I}^{\mathrm{C}}\) is the complement of \(\mathbf{I}\) and \(\check{\mathbf{Z}}\) is the reflected SE.

\section*{Duality Relationships}
\(S E\), Ž, operates on \(I^{C}\) as if it were Z operating on I.

\(S E, Z\), operates on \(\mathbf{I}^{\mathrm{C}}\) as if it were Ž operating on I.

\section*{Gray Ops with Asymmetric SEs}

"L" shaped SE
O marks origin Foreground: white pixels Background: black pixels
\(\cdots\)\begin{tabular}{l} 
Cross-hatched \\
pixels are \\
indeterminate.
\end{tabular}

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Tophat}

tophat + opened = original

tophat: original - opening

SE, \(Z\), is the same flat disk as used for the dilation on page 19.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Tophat}

- \(S E, Z\), is a flat disk.

2 December 2011
© 1999-2011 by Richard Alan Peters II

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Bothat}

region added by dilation

superimposed on original

SE, \(Z\), is the same flat disk as used for the dilation on page 19.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Bothat}

region added by dilation


Bothat: closing - original

SE, \(Z\), is the same flat disk as used for the dilation on page 19.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Bothat}

- \(S E, Z\), is a flat disk.

\footnotetext{
2 December 2011
}
© 1999-2011 by Richard Alan Peters II

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Tophat and Bothat}

- SE, Z, is a flat disk.

2 December 2011
©(1999-2011 by Richard Alan Peters II

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Morphology: Small Feature Detection}

- SE, Z, is a flat disk.

2 December 2011
© 1999-2011 by Richard Alan Peters II

\section*{Algorithm for Grayscale Reconstruction}
1. \(\mathbf{J}=\mathbf{I} \circ \mathbf{Z}\), where \(\mathbf{Z}\) is any \(S E\).
2. \(\mathbf{T}=\mathbf{J}\),
3. \(\mathbf{J}=\mathbf{J} \oplus \mathbf{Z}_{k}\), where \(k=4\) or \(k=8\),
4. \(\mathbf{J}=\min \{\mathbf{I}, \mathbf{J}\}\), [pixelwise minimum of \(\mathbf{I}\) and \(\mathbf{J}\).]
5. if \(\mathbf{J} \neq \mathbf{T}\) then go to 2 ,
6. else stop; [ J is the reconstructed image. ]

This is the same as binary reconstruction but for grayscale images \(\mathbf{J}(r, c) \in \mathbf{I}\) if and only if \(\mathbf{J}(r, c) \leq \mathbf{I}(r, c)\).

Algoritnn for Graysc will take both \(J\) and \(I\) as inputs. E.g.
1. \(\mathbf{J}=\mathbf{I} \circ \mathbf{Z}\), where \(\mathbf{Z}\)

K = ReconGray(I,J,Z);
Then the algorithm starts at step 2.
2. \(\mathbf{T}=\mathbf{J}\),
3. \(\mathbf{J}=\mathbf{J} \oplus \mathbf{Z}_{k}\), where \(k=4\) or \(k=8\),
4. \(\mathbf{J}=\min \{\mathbf{I}, \mathbf{J}\}\), [pixelwise minimum of I and J .]
5. if \(\mathbf{J} \neq \mathbf{T}\) then go to 2 ,
6. else stop; [ J is the reconstructed image. ]

This is the same as binary reconstruction but for grayscale images \(\mathbf{J}(r, c) \in \mathbf{I}\) if and only if \(\mathbf{J}(r, c) \leq \mathbf{I}(r, c)\).

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Reconstruction}

opened image

opened image \& original
\(S E, Z\), is a flat disk the size of the tops of the truncated cones.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Reconstruction}

opened \& recon. image

opened, recon., \& original
\(S E, Z\), is a flat disk
the size of the tops of the truncated cones.

\section*{Grayscale Morphology: Reconstruction}

original

orig - opened

opened

orig - recon

reconstructed

- \(S E, Z\), is a flat disk.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Reconstruction}

- \(S E, Z\), is a flat disk.

\footnotetext{
2 December 2011
}
© 1999-2011 by Richard Alan Peters II

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Reconstruction}

- \(S E, Z\), is a flat disk.

2 December 2011
© 1999-2011 by Richard Alan Peters II

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Grayscale Reconstruction}

- \(S E, Z\), is a flat disk.
2 December 2011
© 1999-2011 by Richard Alan Peters II

\title{
EECE/CS 253 Image Processing
}

\author{
Lecture Notes: JPEG Image Compression
}

\author{
Richard Alan Peters II \\ Department of Electrical Engineering and \\ Computer Science \\ Fall Semester 2011
}

\section*{From the Article by Wallace:}

Title: The JPEG Still Picture Compression Standard
Source: Communications of the ACM archive
Volume 34, Issue 4 (April 1991)
Special issue on digital multimedia systems
Pages: 30-44
Year: 1991
ISSN: 0001-0782
Author: Gregory K. Wallace
Digital Equipment Corp., Maynard, MA
Publisher: ACM Press New York, NY, USA

\section*{The JPEG Still Picture Compression Standard}
, Joint Photographic Experts Group
1 A standards committee set up by the CCITT and the ISO.
Tasked in the late 1980's to generate a generalpurpose standard for compression of almost all continuous tone and still-image applications.
Published Standard: "Digital Compression and Coding of Continuous-tone Still Images - Requirements and Guidelines," ISO/IEC 10918-1:1993(E)

\section*{JPEG’s Goals for the Standard}
, Be at or near the state of the art in compression rate and image fidelity.
User decides on the trade-off between image fidelity and compression ratio.
Be applicable to any kind of continuous-tone digital image source (unrestricted with respect to content, complexity, color-range, statistics, etc.)
Have tractable computational complexity for implementation on a wide range of computational hardware.

\section*{JPEG's Modes of Operation}
1. Sequential encoding: each component (band) encoded in a single raster scan
2. Progressive encoding: progressive coarse-to-fine encoding and decoding of entire image
3. Lossless encoding: exact recovery of original image
4. Hierarchical encoding: multiresolution compression with independently retrievable lower-resolution versions

\title{
EECE/CS 253 Image Processing
}

Vanderbilt University School of Engineering

\section*{JPEG Codec}


\section*{JPEG Baseline Process}
, DCT-based process
, Source image: 8-bit samples within each component Sequential Huffman coding: 2 AC and 2 DC tables
Decoders process scans with 1, 2, 3, or 4 components
। Interleaved and non-interleaved scans

\section*{EECE/CS 253 Image Processing}

\author{
Vanderbilt University School of Engineering
}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{We'll use the image in some of the following slides.}


\section*{Color Conversion: RGB \(\rightarrow\) YCbCr}


RGB


Y


Cb
\[
\left[\begin{array}{c}
\mathrm{Y} \\
\mathrm{Cb} \\
\mathrm{Cr}
\end{array}\right]=\left[\begin{array}{ccc}
0.25678824 & 0.50412941 & 0.09790588 \\
-0.14822290 & -0.29099279 & 0.43921596 \\
0.43921569 & -0.36778831 & -0.07142737
\end{array}\right]\left[\begin{array}{l}
\mathrm{R} \\
\mathrm{G} \\
\mathrm{~B}
\end{array}\right]+\left[\begin{array}{c}
16 \\
128 \\
128
\end{array}\right]
\]

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Downsample Color Bands}


Y


Cb
by a factor of 2

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Partition Each Band into \(8 \times 8\) Blocks}


\section*{Prep for Forward Discrete Cosine Transform}

I
, Each \(8 \times 8\) image block is operated on independently.
- Shift all image values from \(\mathrm{I} \in\left[0,2^{P}-1\right]\) to \(\mathrm{I} \in\left[-2^{P-1}, 2^{P-1}-1\right]\). E.g. [0, 255] \(\rightarrow[-128,127]\).

The FDCT decomposes each block into a set of coefficients with respect to the 64 orthogonal basis functions shown on the next slide.

\section*{DCT Functions and Variables}
\(\mathscr{F}\{\mathbf{I}\}(v, u ; r, c)-\) DCT of \(8 \times 8\) block from \(\mathbf{I}\) starting at \((r, c)\)
\(\Lambda(\xi)\) - Normalization Factor
\(\phi(v, u ; \rho, \chi)-2 D\) Cosine Basis Function, \((v, u)\)
\(r\) - image row index (vertical, increasing down)
\(c\) - image column index (horizontal, increasing right)
\(\rho\) - DCT row index (horizontal wave fronts, vertical propagation down)
\(\chi\) - DCT column index (vertical wave fronts, horizontal propagation right)
\(v\) - DCT row frequency index (vertical, increasing down)
\(u\) - DCT column frequency index (horizontal, increasing right)

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{DCT Basis Functions}


\section*{Forward Discrete Cosine Transform}
\[
\begin{aligned}
& \mathscr{F}\{\mathbf{I}\}(v, u ; r, c)=\sum_{\rho=0}^{7} \sum_{\chi=0}^{7} \frac{1}{4} \Lambda(v) \Lambda(u) \phi(v, u ; \rho, \chi) \mathbf{I}(r+\rho, c+\chi) \\
& \Lambda(\xi)=\left\{\begin{array}{cc}
\frac{1}{\sqrt{2}} & \text { for } \xi=0 \\
1 & \text { otherwise }
\end{array}\right. \\
&(r, c) \in\{0,8,16, \ldots, \mathrm{R}\} \times\{0,8,16, \ldots, c\}, \\
& \phi(v, u ; \rho, \chi)=\cos \left[\frac{1}{16}(2 \rho+1) \pi v\right] \cos \left[\frac{1}{16}(2 \chi+1) \pi u\right]
\end{aligned}
\]

\section*{Forward Discrete Cosine Transform}
\(F\{I\}(0,0 ; r, c)\) is the DC component of the \(8 \times 8\) block from \(I\) at ( \(r, c\) ) ...

...the others are the AC components, e.g. \(F\{I\}(4,4 ; r, c)\).


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Forward Discrete Cosine Transform}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Forward Discrete Cosine Transform}

Each value, \(F(v, u ; r, c)\), in the \(8 \times 8\) output is the sum of the pixel-wise product of the \(8 \times 8\) block that starts at \((r, c)\) in the image and the cosine basis function at row \(v\), column \(u\) in the table.


\section*{Example FDCT Quantization Tables}

Luminance Quantization Table
\begin{tabular}{|r|r|r|r|r|r|r|r|}
\hline 2 & 2 & 3 & 4 & 5 & 6 & 8 & 11 \\
\hline 2 & 2 & 2 & 4 & 5 & 7 & 9 & 11 \\
\hline 3 & 2 & 3 & 5 & 7 & 9 & 11 & 12 \\
\hline 4 & 4 & 5 & 7 & 9 & 11 & 12 & 12 \\
\hline 5 & 5 & 7 & 9 & 11 & 12 & 12 & 12 \\
\hline 6 & 7 & 9 & 11 & 12 & 12 & 12 & 12 \\
\hline 8 & 9 & 11 & 12 & 12 & 12 & 12 & 12 \\
\hline 11 & 11 & 12 & 12 & 12 & 12 & 12 & 12 \\
\hline
\end{tabular}

Precision: 8 bits
Approximate quality factor: 91.64
Scaling: 16.71 Variance: 22.54

Chrominance Quantization Table
\begin{tabular}{|r|r|r|r|r|r|r|r|}
\hline 3 & 3 & 7 & 13 & 15 & 15 & 15 & 15 \\
\hline 3 & 4 & 7 & 13 & 14 & 12 & 12 & 12 \\
\hline 7 & 7 & 13 & 14 & 12 & 12 & 12 & 12 \\
\hline 13 & 13 & 14 & 12 & 12 & 12 & 12 & 12 \\
\hline 15 & 14 & 12 & 12 & 12 & 12 & 12 & 12 \\
\hline 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\
\hline 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\
\hline 15 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\
\hline
\end{tabular}

\author{
Precision: 8 bits
}

Approximate quality factor: 92.57
Scaling: 14.85 Variance: 23.00

\section*{FDCT Quantization Procedure}
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline 255 & 23 & 34 & 13 & 44 & 11 & 44 & 6 \\
\hline 19 & 4 & 19 & 12 & 18 & 9 & 16 & 15 \\
\hline 58 & 10 & 11 & 2 & 14 & 6 & 12 & 7 \\
\hline 22 & 18 & 9 & 12 & 7 & 11 & 12 & 9 \\
\hline 42 & 26 & 19 & 9 & 23 & 15 & 16 & 6 \\
\hline 9 & 6 & 23 & 14 & 20 & 10 & 19 & 21 \\
\hline 48 & 13 & 13 & 11 & 15 & 12 & 18 & 10 \\
\hline 3 & 0 & 13 & 18 & 16 & 4 & 11 & 14 \\
\hline
\end{tabular}

Output of FDCT, \(\mathscr{F}(u, v)\), at image pixel location \((r, c)\).
\((r, c) \in\)
\(\{0,8,16, \ldots, R\} \times\{0,8,16, \ldots, C\}\),
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline 1 & 1 & 1 & 2 & 3 & 6 & 8 & 10 \\
\hline 1 & 1 & 2 & 3 & 4 & 8 & 9 & 8 \\
\hline 2 & 2 & 2 & 3 & 6 & 8 & 10 & 8 \\
\hline 2 & 2 & 3 & 4 & 7 & 12 & 11 & 9 \\
\hline 3 & 3 & 8 & 11 & 10 & 16 & 15 & 11 \\
\hline 3 & 5 & 8 & 10 & 12 & 15 & 16 & 13 \\
\hline 7 & 10 & 11 & 12 & 15 & 17 & 17 & 14 \\
\hline 14 & 13 & 13 & 15 & 15 & 14 & 14 & 14 \\
\hline
\end{tabular}

Quantization Table, \(\mathrm{Q}(u, v)\)
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline 255 & 23 & 34 & 7 & 15 & 2 & 6 & 1 \\
\hline 19 & 4 & 10 & 4 & 5 & 1 & 2 & 2 \\
\hline 29 & 5 & 6 & 1 & 2 & 1 & 1 & 1 \\
\hline 11 & 9 & 3 & 3 & 1 & 1 & 1 & 1 \\
\hline 14 & 9 & 2 & 1 & 2 & 1 & 1 & 1 \\
\hline 3 & 1 & 3 & 1 & 2 & 1 & 1 & 2 \\
\hline 7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
\hline
\end{tabular}

Quantized Result, \(\mathscr{E C}(u, v)\) at image pixel location \((r, c)\).
\[
\mathscr{F}^{Q}(u, v)=\text { round }\left(\frac{\mathscr{F}(u, v)}{Q(u, v)}\right)
\]

This \(Q(u, v)\) is yet another quantization table.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Image DC Components ( \(8 \times 8\) constant blocks)}


\section*{DC Component Coding (DPCM)}


DC components from 32 blocks


Differential PCM coding of same

DC components are coded separately from the AC components.

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Quantized coefficient encoding order}
\begin{tabular}{|rrrrrrrr|}
\hline 255 & 23 & 34 & 7 & 15 & 2 & 6 & 1 \\
\hline 19 & 4 & 10 & 4 & 5 & 1 & 2 & 2 \\
29 & 5 & 6 & 1 & 2 & 1 & 1 & 1 \\
11 & 9 & 3 & 3 & 1 & 1 & 1 & 1 \\
14 & 9 & 2 & 1 & 2 & 1 & 1 & 1 \\
3 & 1 & 3 & 1 & 2 & 1 & 1 & 2 \\
7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
\hline
\end{tabular}


AC: 23192943471051114964152513937123216121113100112112111211111

\section*{Entropy Encoding of AC Coefficients}

Two step process:
1. Convert zigzag sequence of quantized coefficients into a sequence of symbols.
2. Convert symbols to a data stream of variable length codes.

> Symbol 1 = (Runlength, Size) Symbol 2 = Amplitude

\section*{Entropy Encoding of Coefficients}

Symbol 1: (Runlength, Size);
Symbol 2: Amplitude
AC:
Runlength: number of consecutive zero values (0-15)
Size: number of bits used to encode amplitude (1-10)
Amplitude: quantized value
DC:
Runlength: not included
Size: number of bits used to encode amplitude (1-11)
Amplitude: quantized value

\section*{Variable Length Entropy Encoding}


EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Effects of Quality Settings}


Image: http://en.wikipedia.org/wiki/File:Quality_comparison_jpg_vs_saveforweb.jpg

\section*{JFIF Information Display Program}

\section*{JPEGsnoop - JPEG File Decoding Utility}
by Calvin Hass © 2010

From the web page:
Every digital photo contains a wealth of hidden information -- JPEGsnoop was written to expose these details to those who are curious. Not only can one determine the various settings that were used in the digital camera in taking the photo (EXIF metadata, IPTC), but one can also extract information that indicates the quality and nature of the JPEG image compression used by the camera in saving the file. Digital cameras specify compression quality levels, many of them wildly different, leading to the fact that some cameras produce far better JPEG images than others.
http://www.impulseadventure.com/photo/

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Example: JPEG/JFIF Encoded Image}



EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{Data Markers from the JPEG File Interchange Format (JFIF)}
\begin{tabular}{|c|c|c|c|c|}
\hline Short name & Code & Payload & Name & Comments \\
\hline SOI & 0xFFD8 & none & Start Of Image & \\
\hline SOFO & 0xFFC0 & variable size & Start Of Frame (Baseline) & Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0). \\
\hline SOF2 & 0xFFC2 & variable size & Start Of Frame (Progressive) & Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0). \\
\hline DHT & 0xFFC4 & variable size & Define Huffman Table(s) & Specifies one or more Huffman tables. \\
\hline DQT & 0xFFDB & variable size & \begin{tabular}{l}
Define \\
Quantization \\
Table(s)
\end{tabular} & Specifies one or more quantization tables. \\
\hline DRI & 0xFFDD & 2 bytes & Define Restart Interval & Specifies the interval between RSTn markers, in macroblocks. This marker is followed by two bytes indicating the fixed size so it can be treated like any other variable size segment. \\
\hline SOS & 0xFFDA & variable size & Start Of Scan & Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Progressive DCT JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data. \\
\hline RSTn & 0xFFDD0...D7 & none & Restart & Inserted every r macroblocks, where r is the restart interval set by a DRI marker. Not used if there was no DRI marker. The low 3 bits of the marker code cycle in value from 0 to 7 . \\
\hline APPn & 0xFFEn & variable size & Applicationspecific & For example, an Exif JPEG file uses an APP1 marker to store metadata, laid out in a structure based closely on TIFF. \\
\hline COM & 0xFFFE & variable size & Comment & Contains a text comment. \\
\hline EOI & 0xFFD9 & none & End Of Image & \\
\hline
\end{tabular}

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

Filename: [Saoirse Ronan - Ember.jpg]
Filesize: [3156857] Bytes
Start Offset: 0x00000000
*** Marker: SOI (xFFD8) ***
OFFSET: 0x00000000
*** Marker: APP13 (xFFED) ***
OFFSET: 0x00000002
length =28
Identifier = [Photoshop 3.0]
8BIM: [0x0404] Name=[] Len=[0x0000]
IPTC [0xFFE1:002] ? size=12392
*** Marker: APP1 (xFFE1) ***
OFFSET: 0x00000020
length = 560
Identifier = [http://ns.adobe.com/xap/1.0/]
XMP =
*** Marker: APPO (xFFE0) ***
OFFSET: 0x00000252
length = 16
identifier = [JFIF]
version = [1.2]
density = 300 x 300 DPI (dots per inch)
thumbnail = 0 < 0

```
```

*** Marker: APP2 (xFFE2) ***
OFFSET: 0x00000264
length = 576
Identifier = [ICC_PROFILE]
ICC Profile:
Marker Number = 1 of 1
Profile Size : 560 bytes
Preferred CMM Type : 'ADBE' (0x41444245)
Profile Version :0.2.1.0 (0x02100000)
Profile Device/Class : Display Device profile ('mntr' (0x6D6E7472))
Data Colour Space : rgbData ('RGB ' (0x52474220))
Profile connection space (PCS) : 'XYZ ' (0x58595A20)
Profile creation date : 1999-06-03 00:00:00
Profile file signature : 'acsp' (0x61637370)
Primary platform : Apple Computer, Inc. ('APPL' (0x4150504C))
Profile flags :0x00000000
Profile flags > Profile not embedded
Profile flags
> Profile can't be used independently of embedded
: 'none' (0x6E6F6E65)
Device Manufacturer
Device Model
: '....' (0x00000000)
Device attributes
: 0x00000000_00000000
Device attributes
> Reflective
Device attributes
> Glossy
Device attributes
> Media polarity = negative
Device attributes
> Black \& white media
Rendering intent
: Perceptual
Profile creator
: 'ADBE' (0x41444245)
Profile ID
: 0x00000000_00000000_00000000

```
: 'ADBE' (0x41444245)
Profile ID

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

*** Marker: DQT (xFFDB) ***
Define a Quantization Table.
OFFSET: 0x000004B6
Table length = 132
Precision=8 bits
Destination ID=0 (Luminance)
DQT, Row \#0: }20.2\mp@code{2
DQT,Row \#1: }
DQT,Row \#2: 3 2 2 3 5 5 7 9 11 12
DQT, Row \#3: 4 4 4 5 7 7 9 11 12 12
DQT, Row \#4: 5 5 5 7 9 9 11 12 12 12
DQT,Row \#5: 6 7 9 11 12 12 12 12
DQT,Row \#6: 8 9 11 12 12 12 12 12
DQT,Row \#7: 11 11 12 12 12 12 12 12
Approx quality factor = 91.64 (scaling=16.71 variance=22.54)
- ---
Precision=8 bits
Destination ID=1 (Chrominance)
DQT, Row \#0: 3 3 3 7 131515 15 15
DQT,Row \#1: 3 4 7 131412 12 12
DQT, Row \#2: 7 7 13 1412 12 12 12
DQT,Row \#3: 13 1314 12 12 12 12 12

```
```

*** Marker: DRI (Restart Interval) (xFFDD) ***
length = 4
*** Marker: SOF0 (Baseline DCT) (xFFC0) ***
OFFSET: 0x0000053C
Frame header length = 17
Precision = 8
Number of Lines = 2592
Samples per Line = 3872
Image Size = 3872 x 2592
Raw Image Orientation = Landscape
Number of Img components = 3
Comp[1]: ID=0x01, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x00 (Lum: Y)
Comp[2]: ID=0x02, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x01 (Chrom: Cb)
Comp[3]: ID=0x03, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x01 (Chrom: Cr)
OFFSET: 0x0000054F
interval = 484

```

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

*** Marker: DHT (Define Huffman Table) (xFFC4) ***
OFFSET: 0x00000555
Huffman table length =418
Destination ID = 0
Class = 0 (DC / Lossless Table)
Codes of length 01 bits (000 total):
Codes of length 02 bits (000 total):
Codes of length 03 bits (007 total): 04050302060100
Codes of length 04 bits (001 total): 07
Codes of length 05 bits (001 total): 08
Codes of length 06 bits (001 total): 09
Codes of length 07 bits (001 total): 0A
Codes of length 08 bits (001 total): 0B
Codes of length 09 bits (000 total):
Codes of length }10\mathrm{ bits (000 total):
Codes of length }11\mathrm{ bits (000 total):
Codes of length }12\mathrm{ bits (000 total):
Codes of length }13\mathrm{ bits (000 total):
Codes of length }14\mathrm{ bits (000 total):
Codes of length }15\mathrm{ bits (000 total):
Codes of length }16\mathrm{ bits (000 total):
Total number of codes: 012

```

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

----
Destination ID = 0
Class = 1 (AC Table)
Codes of length 01 bits (000 total):
Codes of length 02 bits (002 total): 01 02
Codes of length 03 bits (001 total): 03
Codes of length 04 bits (003 total): }11040
Codes of length 05 bits (003 total): 05 21 12
Codes of length 06 bits (002 total): }314
Codes of length 07 bits (004 total): 51 06 1361
Codes of length 08 bits (002 total): }227
Codes of length 09 bits (006 total): }81143291 A1 07
Codes of length 10 bits (007 total): 15 B1 42 23 C1 52 D1
Codes of length }11\mathrm{ bits (003 total): E1 33 16
Codes of length 12 bits (004 total): 62 F0 }247
Codes of length }13\mathrm{ bits (002 total): }82\mathrm{ F1
Codes of length 14 bits (006 total): }2543345392 A
Codes of length }15\mathrm{ bits (002 total): B2 }6
Codes of length }16\mathrm{ bits (115 total):
73 C2 354427 93 A3 B3 36 1754 64 74 C3 D2 E2
08268309 0A 18 1984944546 A4 B4 56 D3 55
28 1A F2 E3 F3 C4 D4 E4 F4 }65758595 A5 B5 C5
D5 E5 F5 66 76 86 96 A6 B6 C6 D6 E6 F6 374757
67778797 A7 B7 C7 D7 E7 F7 38 48 58 6878 88
98 A8 B8 C8 D8 E8 F8 29 394959697989 99 A9
B9 C9 D9 E9 F9 2A 3A 4A 5A 6A 7A 8A 9A AA BA CA
DA EA FA
Total number of codes: 162

```
```

Destination ID = 1
Class = 1(AC Table)
Codes of length 01 bits (000 total):
Codes of length 02 bits (002 total): 01 00
Codes of length 03 bits (002 total): 02 11
Codes of length 04 bits (001 total): 03
Codes of length 05 bits (002 total): 04 21
Codes of length 06 bits (003 total): }12314
Codes of length 07 bits (005 total): 0551136122
Codes of length 08 bits (005 total): 06 71819132
Codes of length 09 bits (004 total): A1 B1 F0 }1
Codes of length }10\mathrm{ bits (005 total): C1 D1 E1 }234
Codes of length }11\mathrm{ bits (006 total): 15 52 62 72 F1 33
Codes of length }12\mathrm{ bits (004 total): 24 344382
Codes of length 13 bits (008 total): }16925325 A2 63 B2 C2
Codes of length }14\mathrm{ bits (003 total): 07 73 D2
Codes of length }15\mathrm{ bits (003 total): 35 E2 44
Codes of length }16\mathrm{ bits (109 total):
831754930809 0A 18 1926 3645 1A 27 64 74
55 37 F2 A3 B3 C3 28 29 D3 E3 F3 84 94 A4 B4 C4
D4 E4 F4 65 75 85 95 A5 B5 C5 D5 E5 F5 46 56 66
76 86 96 A6 B6 C6 D6 E6 F6 47 57 67 77 87 97 A7
B7 C7 D7 E7 F7 3848586878 8898 A8 B8 C8 D8
E8 F8 39 495969 79 89 99 A9 B9 C9 D9 E9 F9 2A
3A 4A 5A 6A 7A 8A 9A AA BA CA DA EA FA
Total number of codes: }16

```

EECE/CS 253 Image Processing

\section*{Vanderbilt University School of Engineering}

\section*{JPEG / JFIF Information (via JPEGsnoop)}

\author{
*** Marker: SOS (Start of Scan) (xFFDA) *** \\ OFFSET: 0x000006F9 \\ Scan header length \(=12\) \\ Number of img components = 3 \\ Component[1]: selector \(=0 \times 01\), table \(=0 \times 00\) \\ Component[2]: selector=0x02, table=0x11 \\ Component[3]: selector \(=0 \times 03\), table \(=0 \times 11\) \\ Spectral selection = 0 .. 63 \\ Successive approximation \(=0 x 00\) \\ *** Decoding SCAN Data *** \\ OFFSET: 0x00000707 \\ Scan Decode Mode: No IDCT (DC only) \\ NOTE: Low-resolution DC component shown. \\ Scan Data encountered marker \\ 0xFFD9 @ 0x00302B77.0 \\ Compression stats: \\ Compression Ratio: 3028.43:1
}
\begin{tabular}{lc} 
Huffman code histogram stats: \\
Huffman Table: (Dest ID: 0, Class: DC) \\
\# codes of length 01 bits: & \(0(0 \%)\) \\
\# codes of length 02 bits: & \(0(0 \%)\) \\
\# codes of length 03 bits: & \(142702(91 \%)\) \\
\# codes of length 04 bits: & \(8490(5 \%)\) \\
\# codes of length 05 bits: & \(3974(3 \%)\) \\
\# codes of length 06 bits: & \(1649(1 \%)\) \\
\# codes of length 07 bits: & \(1(0 \%)\) \\
\# codes of length 08 bits: & \(0(0 \%)\) \\
\# codes of length 09 bits: & \(0(0 \%)\) \\
\# codes of length 10 bits: & \(0(0 \%)\) \\
\# codes of length 11 bits: & \(0(0 \%)\) \\
\# codes of length 12 bits: & \(0(0 \%)\) \\
\# codes of length 13 bits: & \(0(0 \%)\) \\
\# codes of length 14 bits: & \(0(0 \%)\) \\
\# codes of length 15 bits: & \(0(0 \%)\) \\
\# codes of length 16 bits: & \(0(0 \%)\)
\end{tabular}

Huffman Table: (Dest ID: 1, Class: DC)
\# codes of length 01 bits: 0 ( \(0 \%\) )
\# codes of length 02 bits: 162056 ( 52\%) \# codes of length 03 bits: 128895 ( 41\%)
\# codes of length 04 bits: 22673( 7\%)
\# codes of length 05 bits: 8 ( \(0 \%\) ) \# codes of length 06 bits: 0 ( 0\%)
\# codes of length 07 bits: \# codes of length 08 bits: 0 ( 0\%) 0 ( 0\%) \# codes of length 09 bits: \# codes of length 10 bits: \# codes of length 11 bits: 0 (0\%)
0 ( 0\%) 0 ( 0\%) \# codes of length 12 bits: 0 ( 0\%) \# codes of length 13 bits: \# codes of length 14 bits: \# codes of length 15 bits: \# codes of length 16 bits:

EECE/CS 253 Image Processing
Vanderbilt University School of Engineering

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

Huffman Table: (Dest ID: 0, Class: AC)

# codes of length 01 bits: 0( 0%)

# codes of length 02 bits: }2103874\mathrm{ ( 53%)

# codes of length 03 bits: }393151\mathrm{ ( 10%)

# codes of length 04 bits: 712367 (18%)

# codes of length 05 bits: 368952( 9%)

# codes of length 06 bits: 148211 ( 4%)

# codes of length 07 bits: 116694 ( 3%)

# codes of length 08 bits: 52460( 1%)

# codes of length 09 bits: 49772 ( 1%)

# codes of length 10 bits: 19533(0%)

# codes of length 11 bits: 2118(0%)

# codes of length }12\mathrm{ bits: 2566(0%)

# codes of length }13\mathrm{ bits: 726(0%)

# codes of length 14 bits: }367\mathrm{ ( 0%)

# codes of length 15 bits: 24 ( 0%)

# codes of length 16 bits: 1861(0%)

```
```

Huffman Table: (Dest ID: 1, Class: AC)

# codes of length 01 bits: 0(0%)

# codes of length 02 bits: 580346 ( 62%)

# codes of length 03 bits: 209085 ( 22%)

# codes of length 04 bits: 25344( 3%)

# codes of length 05 bits: 50682( 5%)

# codes of length 06 bits: 50395(5%)

# codes of length 07 bits: 10862( 1%)

# codes of length 08 bits: 2482(0%)

# codes of length 09 bits: 2378(0%)

# codes of length 10 bits: 941(0%)

# codes of length 11 bits: 273(0%)

# codes of length 12 bits: 0 ( 0%)

# codes of length 13 bits: 1 ( 0%)

# codes of length 14 bits: 0 ( 0%)

# codes of length 15 bits: 0(0%)

# codes of length 16 bits: 0 ( 0%)

```

YCC clipping in DC:
Y component: \([<0=0][>255=0]\)
Cb component: \([<0=0][>255=0]\)
Cr component: \([<0=0][>255=0]\)
RGB clipping in DC:
R component: \([<0=0][>255=0]\)
G component: \([<0=0][>255=0]\)
B component: \([<0=0][>255=0]\)
Average Pixel Luminance ( Y ):
\(\mathrm{Y}=[86]\) (range: \(0 . .255\) )
Brightest Pixel Search:
YCC=[ 1016, 0, 0]
RGB=[255,255,255] @ MCU[ 94, 1]
Finished Decoding SCAN Data
Number of RESTART markers decoded: 323
Next position in scan buffer:
Offset 0x00302B75.1

\section*{JPEG / JFIF Information (via JPEGsnoop)}
```

*** Marker: EOI (End of Image) (xFFD9) ***
OFFSET: 0x00302B77
*** Searching Compression Signatures ***

| Signature: | 0166B0BC0B82C8233430BF67FA31C829 |
| :--- | :--- |
| Signature (Rotated): | $0166 B 0 B C 0 B 82 C 8233430 B F 67 F A 31 C 829$ |
| File Offset: | 0 bytes |
| Chroma subsampling: | $1 \times 1$ |
| EXIF Make/Model: | NONE |
| EXIF Makernotes: | NONE |
| EXIF Software: | NONE |

    Searching Compression Signatures: (3327 built-in, 0 user(*) )
    | EXIF.Make / Software | EXIF.Model | Quality | Subsamp Match? |
| :---: | :---: | :---: | :---: |
| SW :[Adobe Photoshop] |  | ve As 10] |  |

Based on the analysis of compression characteristics and EXIF metadata:
ASSESSMENT: Class 1 - Image is processed/edited
Position Marked @ MCU=[177, 52](0,0) YCC=[ -468, -60, 30]

```
```

