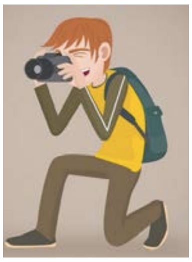
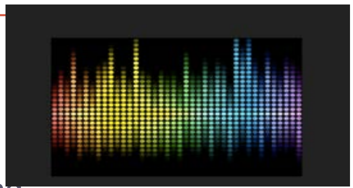
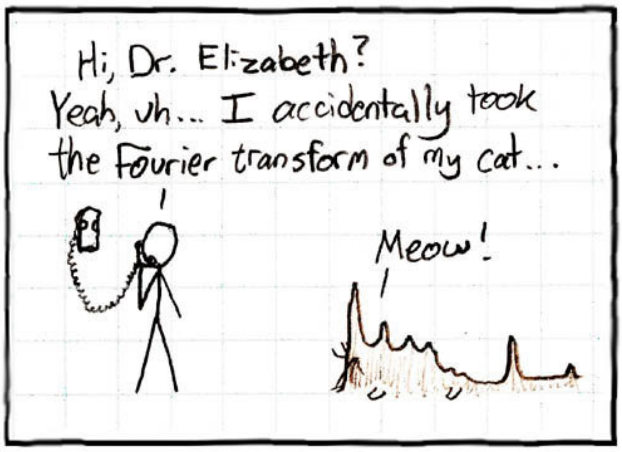
DIGITAL IMAGE PROCESSING





Lecture 4
Frequency domain
Tammy Riklin Raviv
Electrical and Computer Engineering
Ben-Gurion University of the Negev

Fourier Domain



2D Fourier transform and its applications

- Fourier transforms and spatial frequencies in 2D
  - Definition and meaning
- The Convolution Theorem
  - Applications to spatial filtering
- The Sampling Theorem and Aliasing

Much of this material is a straightforward generalization of the 1D Fourier analysis with which you are familiar.

A. Zisserman

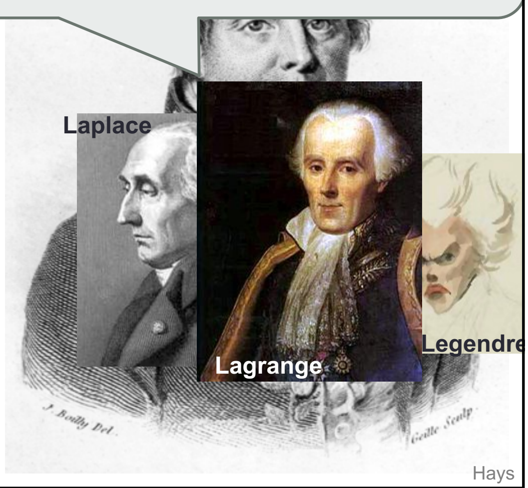
Jean Baptiste Joseph Fourier (1768-1830)

A bold idea (1807):

***Any** univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

- Don't believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it's (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions

...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.



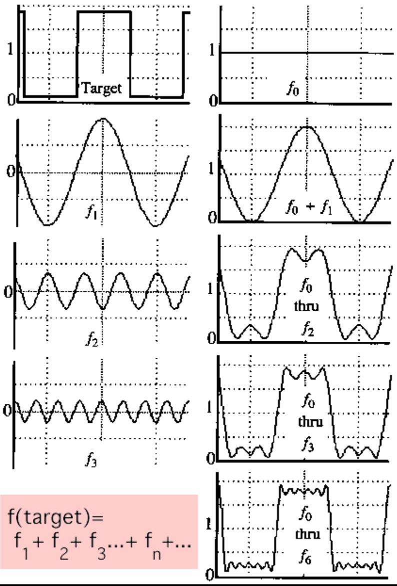
Hays

A sum of sines and cosines

Our building block:

$$A\sin(\omega x) + B\cos(\omega x)$$

Add enough of them to get any signal *g(x)* you want!

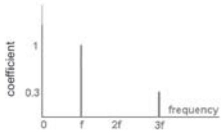


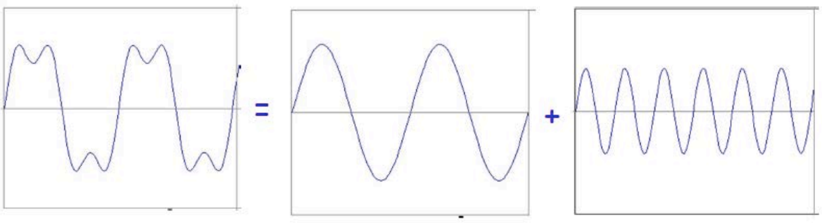
f(target)= $f_1 + f_2 + f_3 ... + f_n + ...$

Hays

Reminder: 1D Fourier Series

Example

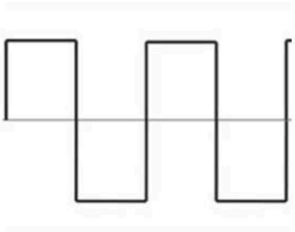




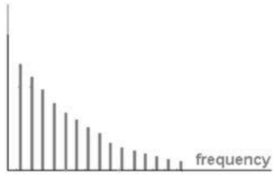
$$f(x) = \sin x + \frac{1}{3}\sin 3x + \ldots$$

A. Zisserman

Fourier Series of a Square Wave



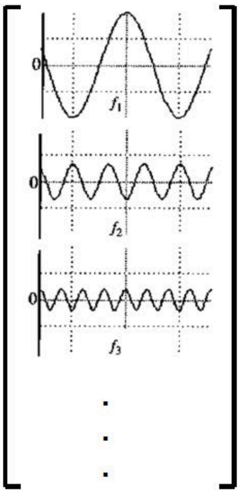
= $f(x) = \sum_{n=1,3,5,\ldots} \frac{1}{n} \sin nx$



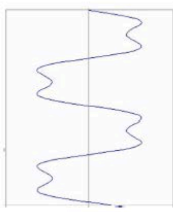
A. Zisserman

Fourier Series: Just a change of basis

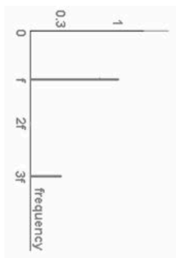
M $f(x) = F(\omega)$



.



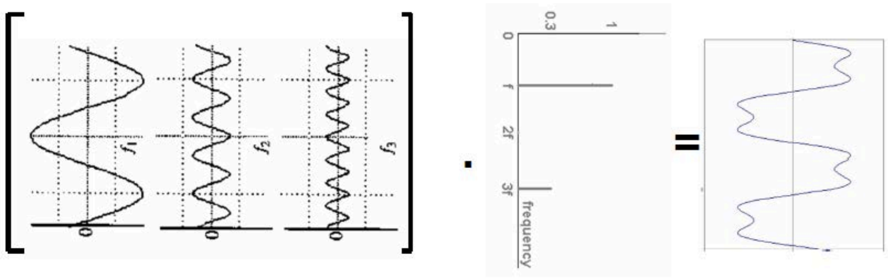
=



A. Zisserman

Inverse FT: Just a change of basis

$$M^{-1}\, F(\omega) = f(x)$$



A. Zisserman

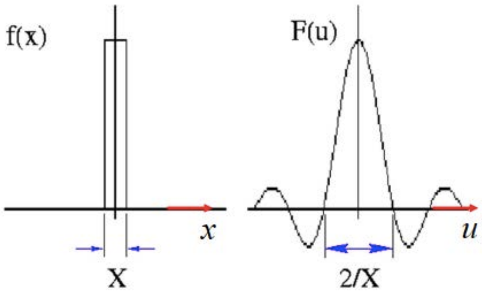
1D Fourier Transform

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux}\, dx,$$
$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux}\, du$$

A. Zisserman

1D Fourier Transform

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux}\, dx,$$
$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux}\, du$$

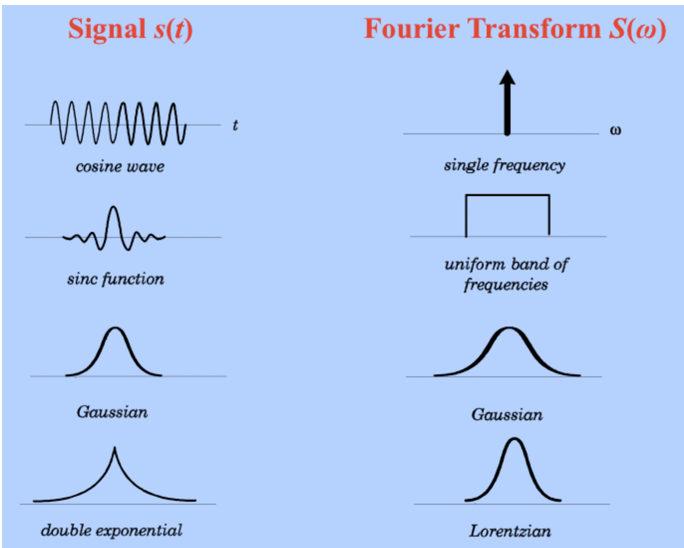


Example

$$f(x) = \begin{cases} 1, & |x| < \frac{X}{2}, \\ 0, & |x| \geq \frac{X}{2}. \end{cases}$$

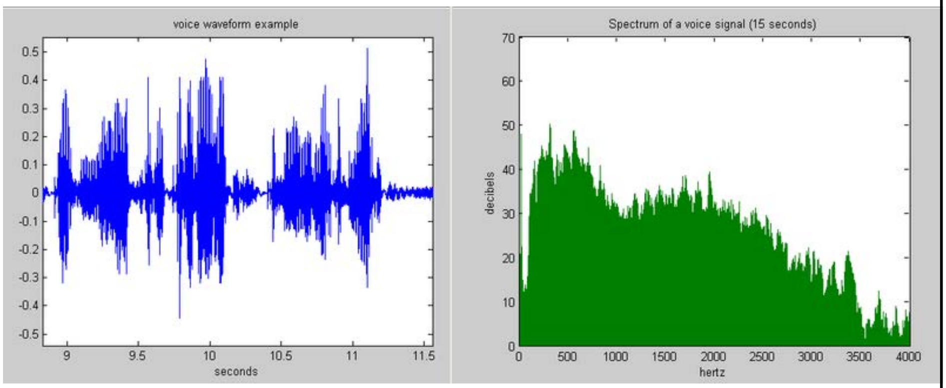
$$\begin{aligned} F(u) &= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \\ &= \int_{-X/2}^{X/2} e^{-j2\pi ux} dx \\ &= \frac{1}{-j2\pi u}[e^{-j2\pi uX/2} - e^{j2\pi uX/2}] \\ &= X \frac{\sin(\pi X u)}{(\pi X u)} = X \mathrm{sinc}(\pi \mathrm{X} \mathrm{u}). \end{aligned}$$

Fourier Transform



Example: Music

- We think of music in terms of frequencies at different magnitudes

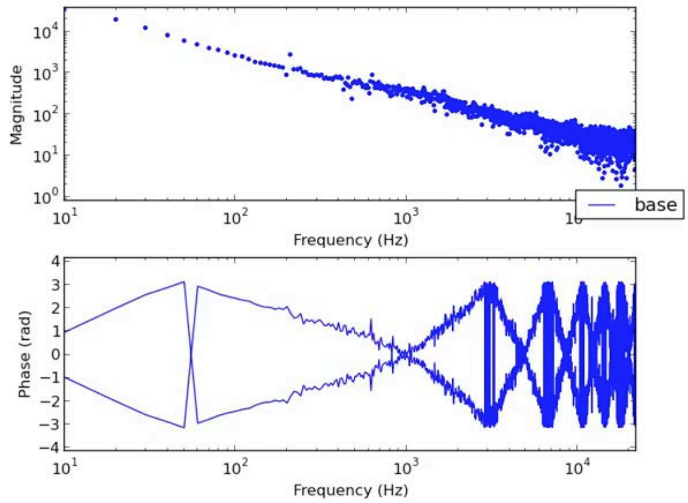


Slide: Hoiem

Fourier Analysis of a Piano

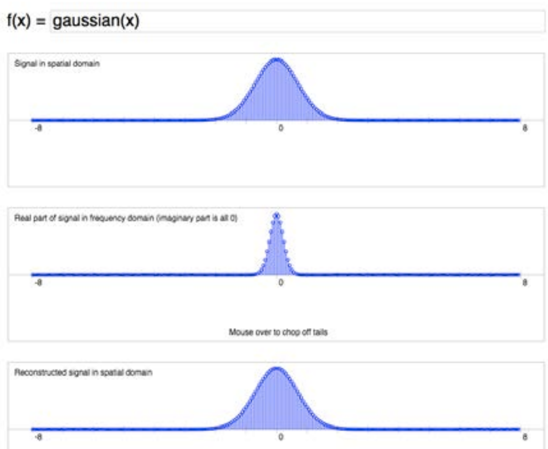
https://www.youtube.com/watch?v=6SR81Wh2cqU

Fourier Analysis of a Piano



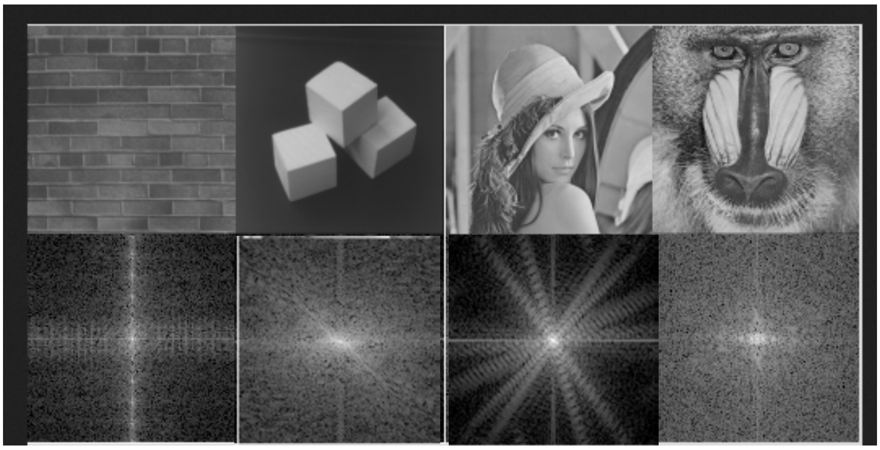
Discrete Fourier Transform Demo

http://madebyevan.com/dft/



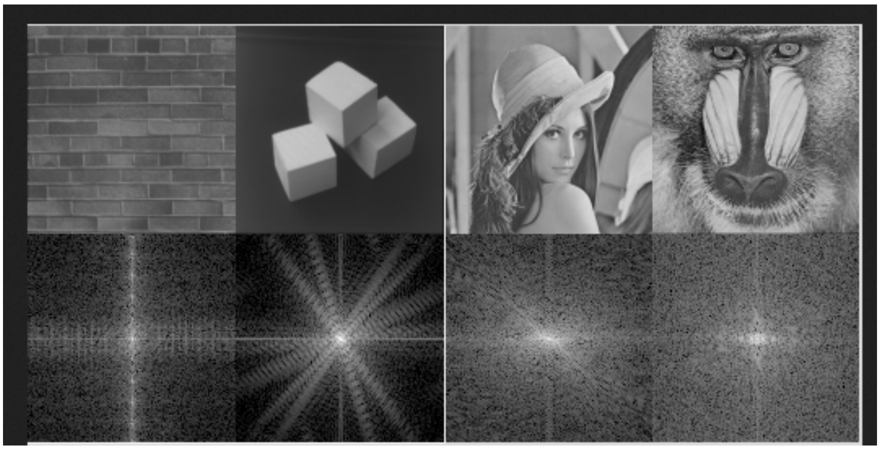
Evan Wallace

2D Fourier Transform



Do not take in for granted! I switched one pair of FT –can you guess which?

2D Fourier Transform



2D Fourier Transform

Definition

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy,$$
$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

where u and v are spatial frequencies.

Also will write FT pairs as $f(x,y) \Leftrightarrow F(u,v)$.

A. Zisserman

2D Fourier Transform

- $F(u,v)$ is complex in general,

$$F(u,v) = F_{\mathrm{R}}(u,v) + jF_{\mathrm{I}}(u,v)$$

- $|F(u,v)|$ is the magnitude spectrum
- $\arctan(F_{\mathrm{I}}(u,v)/F_{\mathrm{R}}(u,v))$ is the phase angle spectrum.
- Conjugacy: $f^*(x,y) \Leftrightarrow F(-u,-v)$
- Symmetry: $f(x,y)$ is even if $f(x,y) = f(-x,-y)$

A. Zisserman

Sinusoidal Waves

In 1D the Fourier transform is based on a decompostion into functions $e^{j2\pi ux} = \cos 2\pi ux + j \sin 2\pi ux$ which form an orthogonal basis. Similarly in 2D

$$e^{j2\pi(ux+vy)} = \cos 2\pi(ux+vy) + j \sin 2\pi(ux+vy)$$

The real and imaginary terms are sinusoids on the x, y plane. The maxima and minima of $\cos 2\pi(ux+vy)$ occur when

$$2\pi(ux+vy) = n\pi$$

Sinusoidal Waves

write $ux + vy$ using vector notation with $\mathbf{u} = (u, v)^\top, \mathbf{x} = (x, y)^\top$ then

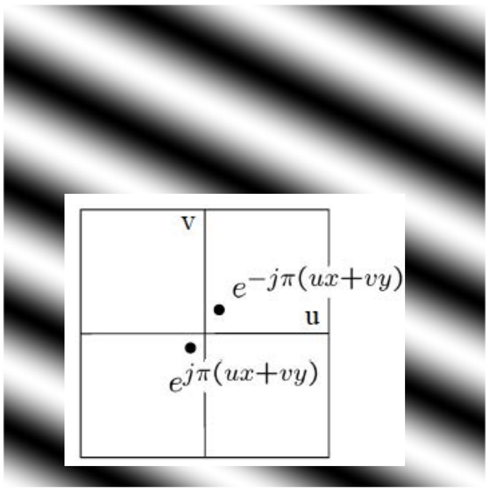
$$2\pi(ux+vy) = 2\pi\mathbf{u}.\mathbf{x} = n\pi$$

are sets of equally spaced parallel lines with normal $\mathbf{u}$ and wavelength $1/\sqrt{u^2+v^2}$.

A. Zisserman

Sinusoidal Waves

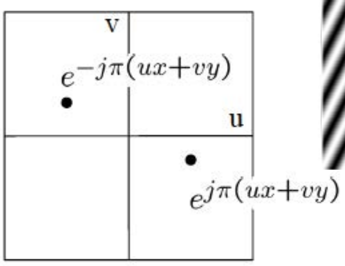
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.



slide: B. Freeman

Sinusoidal Waves

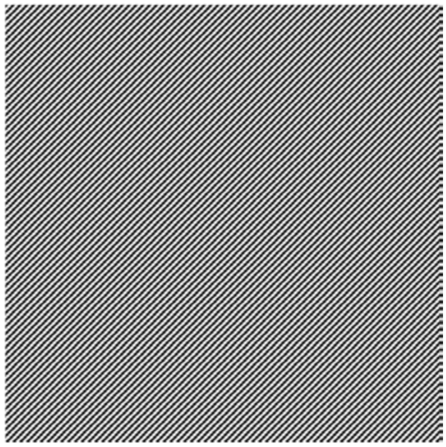
Here u and v are larger than in the previous slide.

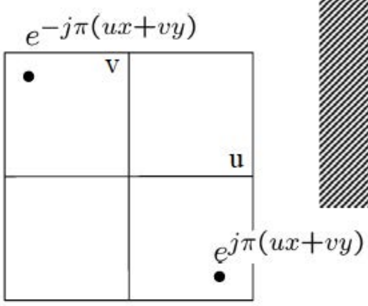


slide: B. Freeman

Sinusoidal Waves

And larger still...

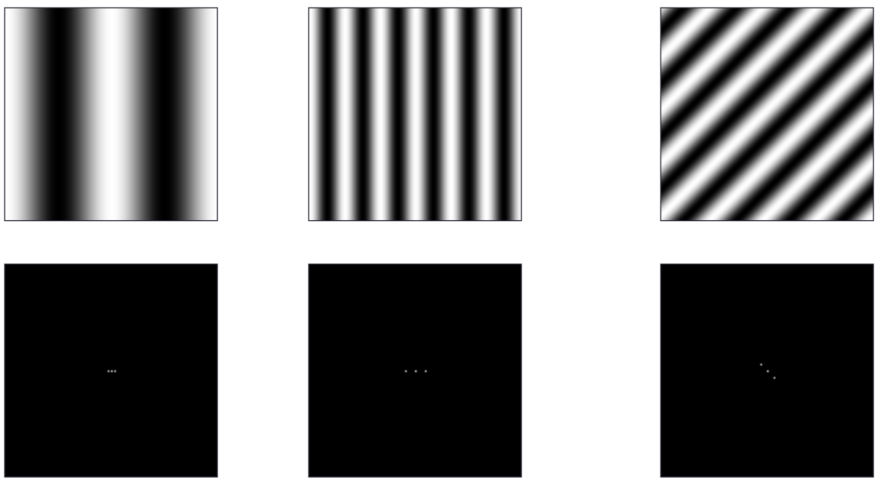




slide: B. Freeman

Fourier analysis in images

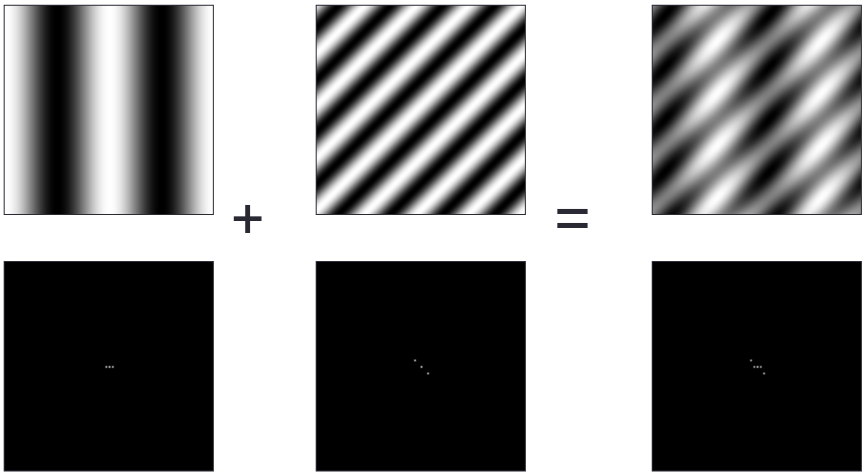
Intensity images



Fourier decomposition images

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed



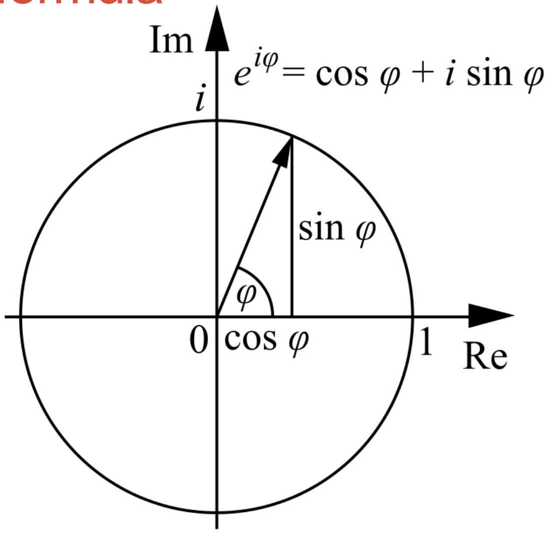
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

Fourier Transform

- Stores the amplitude and phase at each frequency:
  - For mathematical convenience, this is often notated in terms of real and complex numbers
  - Related by Euler's formula

Hays

Euler's formula



Wikipedia

Fourier Transform

- Stores the amplitude and phase at each frequency:
  - For mathematical convenience, this is often notated in terms of real and complex numbers
  - Related by Euler's formula

  - Amplitude encodes how much signal there is at a particular frequency

Amplitude: $$A = \pm\sqrt{\mathrm{Re}(\omega)^2 + \mathrm{Im}(\omega)^2}$$

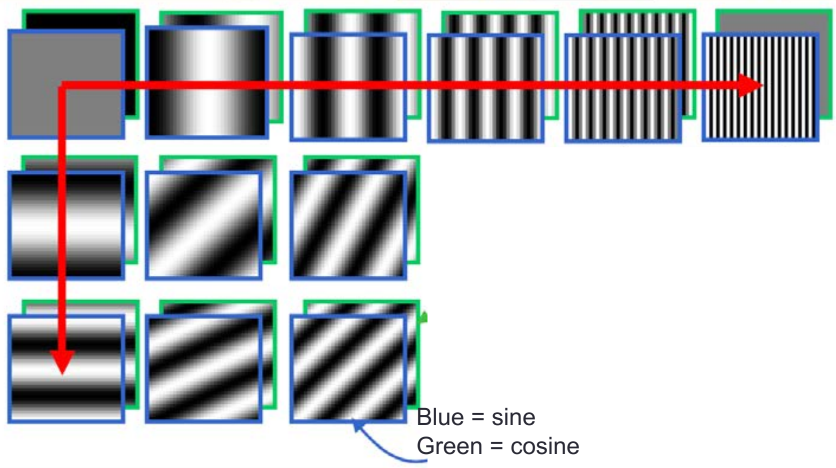
  - Phase encodes spatial information (indirectly)

Phase: $$\phi = \tan^{-1}\frac{\mathrm{Im}(\omega)}{\mathrm{Re}(\omega)}$$

Hays

Fourier Bases

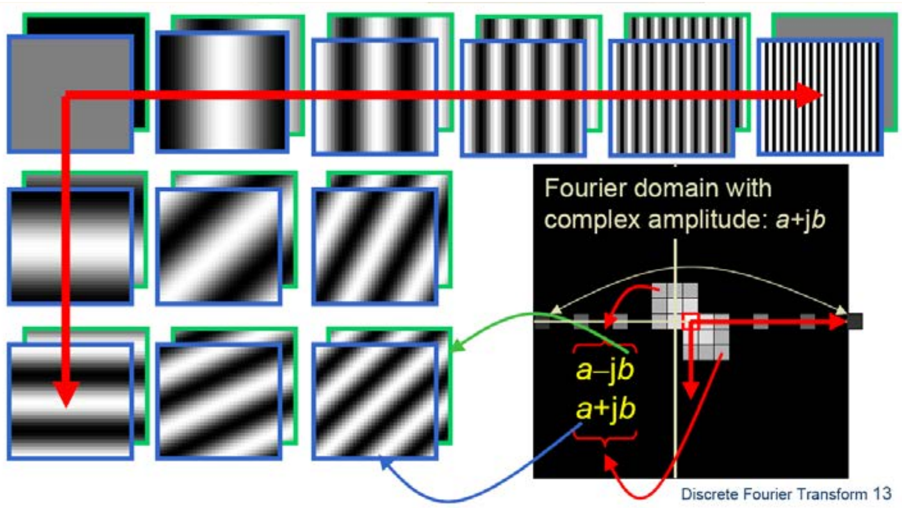
Teases away 'fast vs. slow' changes in the image.



This change of basis is the Fourier Transform

Hays

Fourier Bases



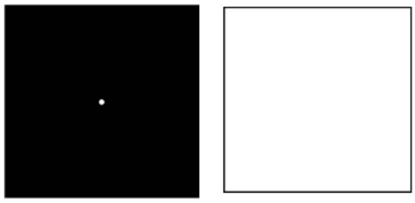
Hays

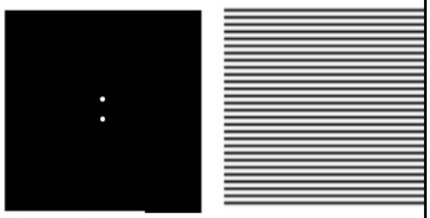
Important Fourier Transform Pairs

$$f(x,y) = \delta(x,y) = \delta(x)\delta(y)$$

$$F(u,v) = \int\int \delta(x,y) e^{-j2\pi(ux+vy)}\, dx dy$$

$$= 1$$





$$f(x,y) = \frac{1}{2}\left(\delta(x, y-a) + \delta(x, y+a)\right)$$

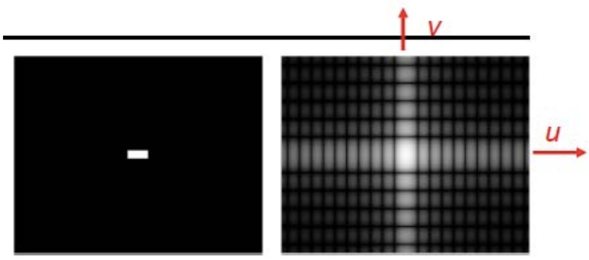
$$F(u,v) = \frac{1}{2}\int\int \left(\delta(x, y-a) + \delta(x, y+a)\right) e^{-j2\pi(ux+vy)}\, dx dy$$

$$= \frac{1}{2}\left(e^{-j2\pi av} + e^{j2\pi av}\right) = \cos 2\pi av$$

A. Zisserman

Important Fourier Transform Pairs

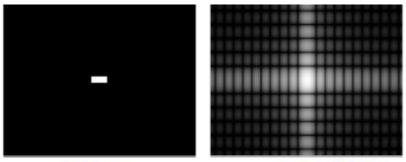
rectangle centred at origin
with sides of length X and Y



$f(x,y)$ $|F(u,v)|$

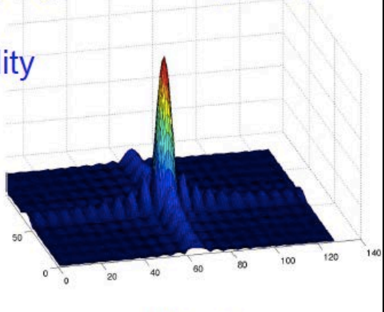
A. Zisserman

Important Fourier Transform Pairs



$f(x,y)$ $|F(u,v)|$

$$
\begin{aligned}
F(u,v) &= \int\int f(x,y)e^{-j2\pi(ux+vy)}dxdy, \\
&= \int_{-X/2}^{X/2} e^{-j2\pi ux}dx \int_{-Y/2}^{Y/2} e^{-j2\pi vy}dy, \quad \text{separability} \\
&= \left[\frac{e^{-j2\pi ux}}{-j2\pi u}\right]_{-X/2}^{X/2} \left[\frac{e^{-j2\pi vy}}{-j2\pi v}\right]_{-Y/2}^{Y/2}, \\
&= \frac{1}{-j2\pi u}\left[e^{-juX} - e^{juX}\right] \frac{1}{-j2\pi v}\left[e^{-jvY} - e^{jvY}\right], \\
&= XY\left[\frac{sin(\pi Xu)}{\pi Xu}\right]\left[\frac{sin(2\pi Yv)}{\pi Yv}\right] \\
&= XY\mathrm{sinc}(\pi \mathrm{Xu})\mathrm{sinc}(\pi \mathrm{Yv}).
\end{aligned}
$$



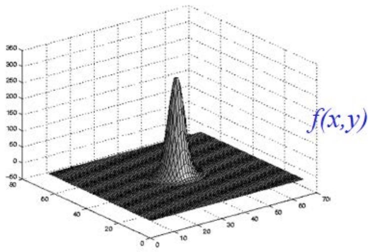
$|F(u,v)|$

Important Fourier Transform Pairs

Gaussian centred on origin

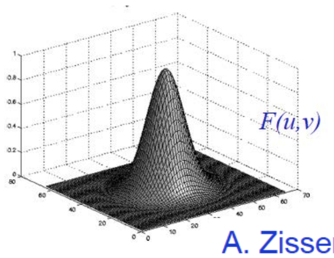
$$f(r) = \frac{1}{2\pi\sigma^2}e^{-r^2/2\sigma^2}$$

where $r^2 = x^2 + y^2$.



$$F(u,v) = F(\rho) = e^{-2\pi^2\rho^2\sigma^2}$$

where $\rho^2 = u^2+v^2$.



- FT of a Gaussian is a Gaussian
- Note inverse scale relation

A. Zisserman

Important Fourier Transform Pairs

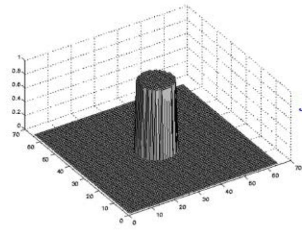
Circular disk unit height and radius *a* centred on origin

$$f(x,y) = \begin{cases} 1, & |r| < a, \\ 0, & |r| \geq a. \end{cases}$$

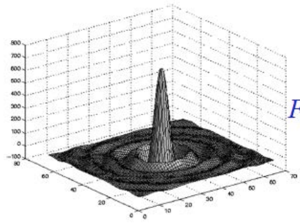
$$F(u,v) = F(\rho) = aJ_1(\pi a \rho)/\rho$$

where $J_1(x)$ is a Bessel function.

- rotational symmetry
- a '2D' version of a sinc



f(x,y)



F(u,v)

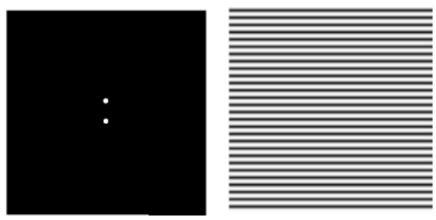
A. Zisserman

Important Fourier Transform Pairs

$$f(x,y) = \delta(x,y) = \delta(x)\delta(y)$$

$$F(u,v) = \int\int \delta(x,y) e^{-j2\pi(ux+vy)}\, dx dy = 1$$

f(x,y) *F(u,v)*



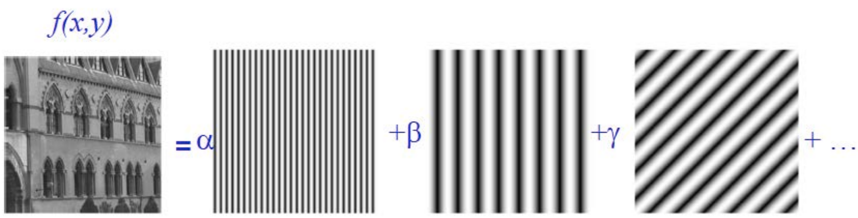
A. Zisserman

2D Fourier Decomposition

The spatial function $f(x, y)$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)}\, du\, dv$$

is decomposed into a weighted sum of 2D orthogonal basis functions in a similar manner to decomposing a vector onto a basis using scalar products.



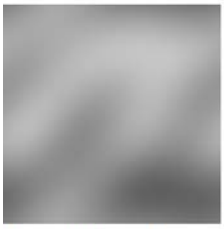
A. Zisserman

Basis reconstruction

Full image

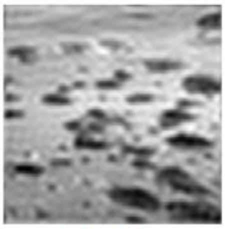
First 1 basis fn

First 4 basis fns



First 9 basis fns

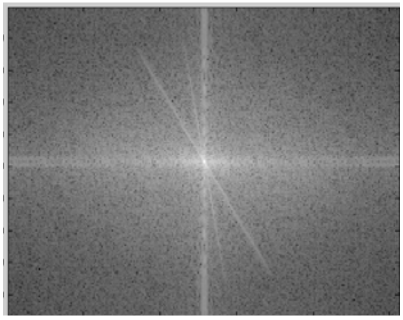
First 16 basis fns



First 400 basis fns

Danny Alexander

2D Fourier Transform of Real Images

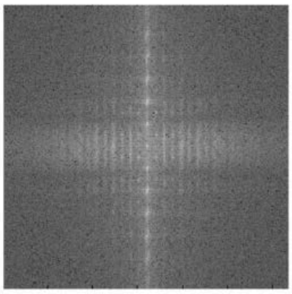


What does it mean to be at pixel x,y?
What does it mean to be more or less bright in the Fourier decomposition image?

2D Fourier Transform of Real Images

Image with periodic structure

$f(x,y)$

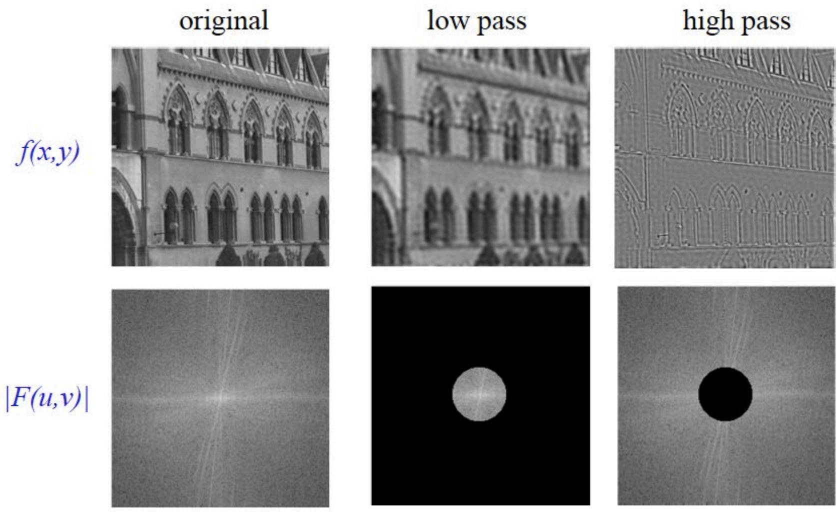


$|F(u,v)|$

FT has peaks at spatial frequencies of repeated texture

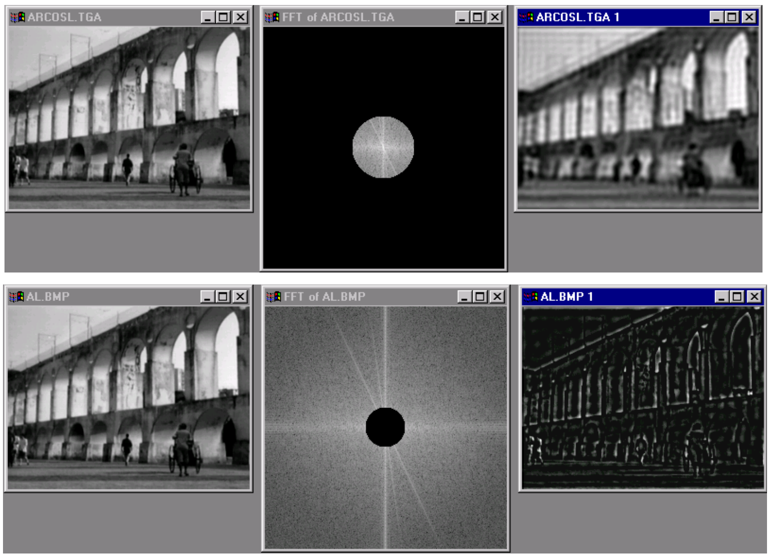
A. Zisserman

Low/High Pass Filters

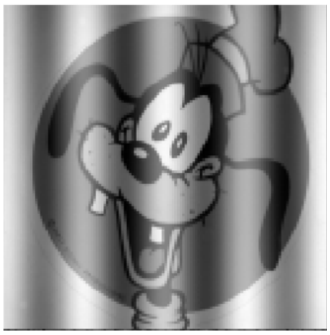


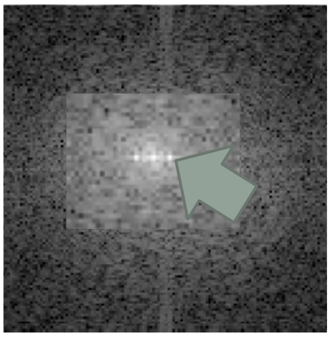
A. Zisserman

Low and High Pass filtering



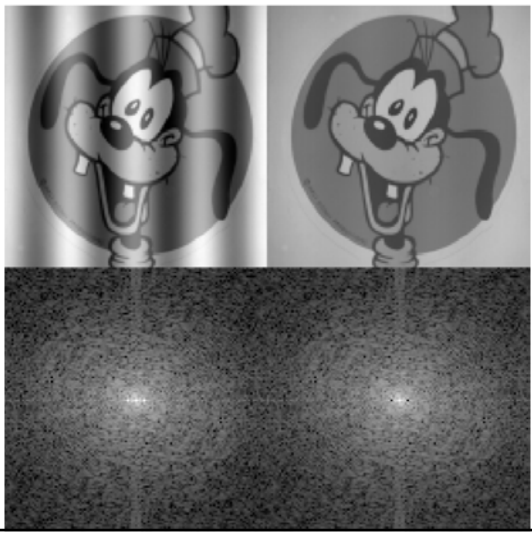
Removing frequency bands





Brayer

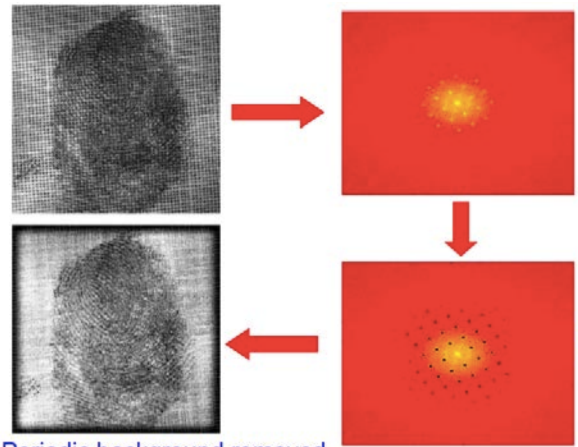
Removing frequency bands



Brayer

Removing frequency bands

Example – Forensic application

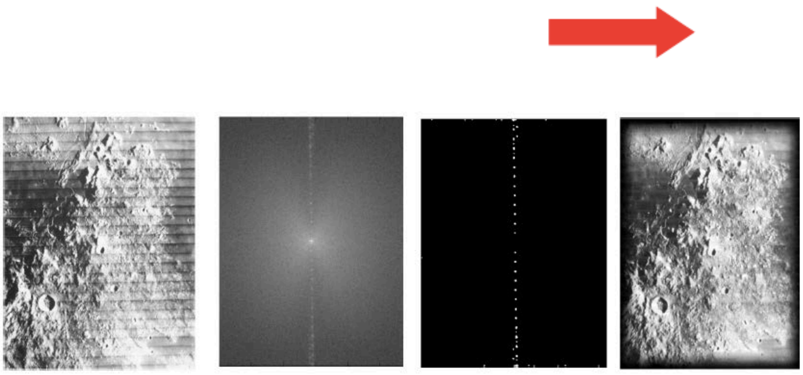


Periodic background removed

A. Zisserman

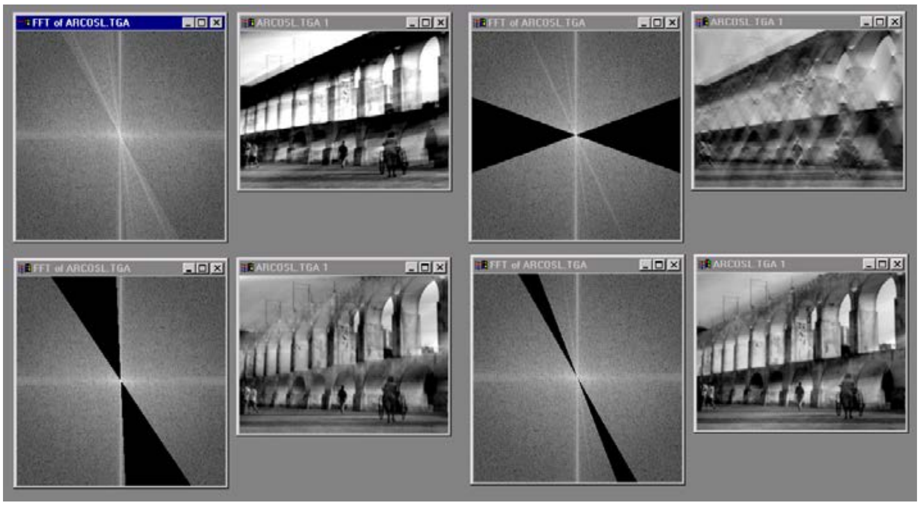
Removing frequency bands

Lunar orbital image (1966)

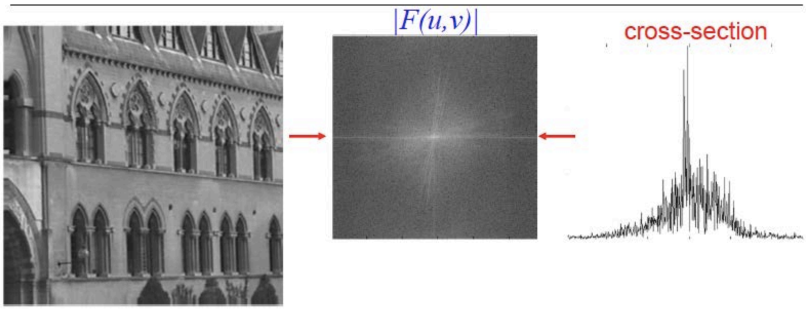


A. Zisserman

Editting frequencies



Magnitude vs. Phase

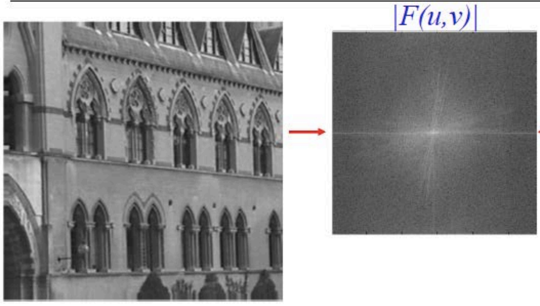


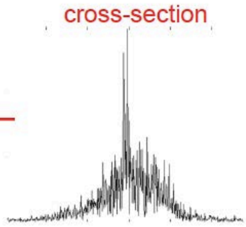
$f(x,y)$

- |f(u,v)| generally decreases with higher spatial frequencies

A. Zisserman

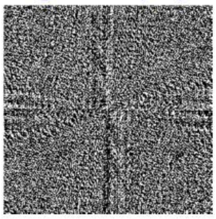
Magnitude vs. Phase





f(x,y)

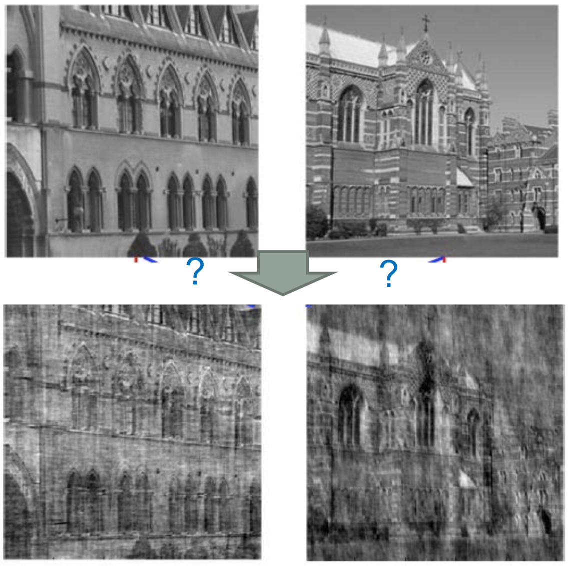
phase F(u,v)



- |f(u,v)| generally decreases with higher spatial frequencies
- phase appears less informative

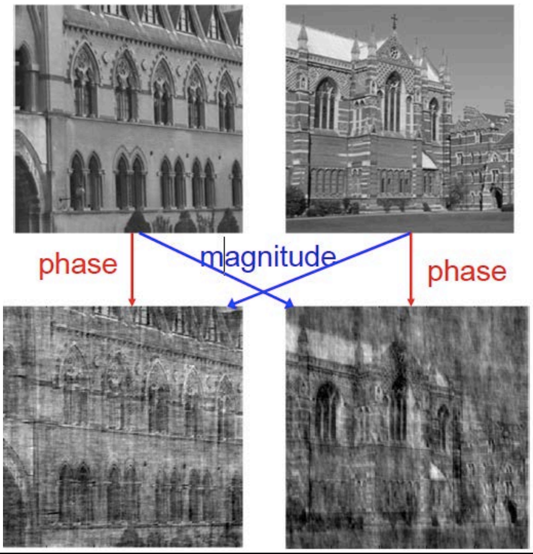
A. Zisserman

The Importance of Phase



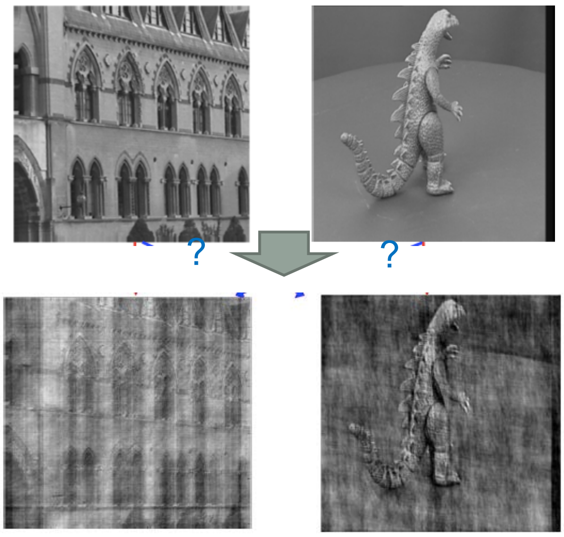
A. Zisserman

The Importance of Phase



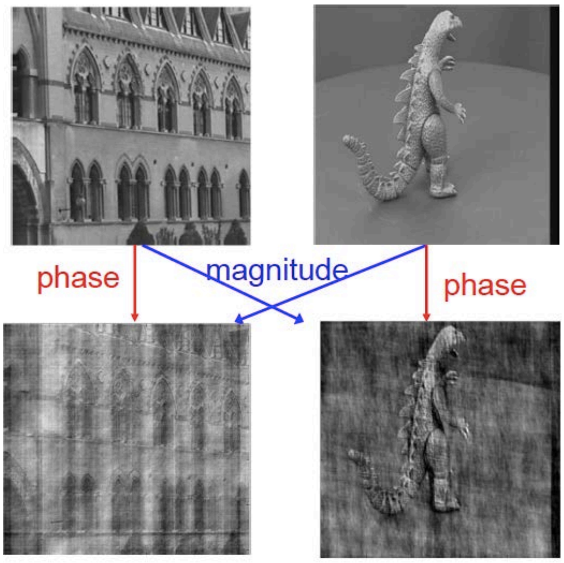
A. Zisserman

Phase and Magnitude- Another Example



A. Zisserman

The Importance of Phase



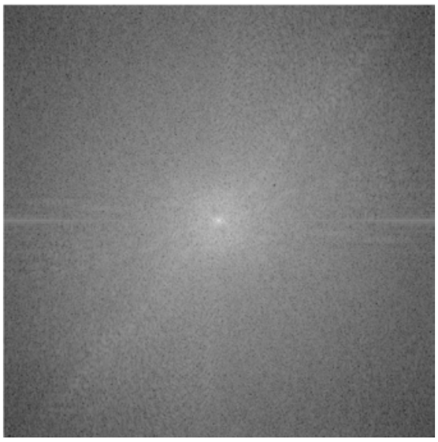
A. Zisserman

Phase and Magnitude- Yet Another Example

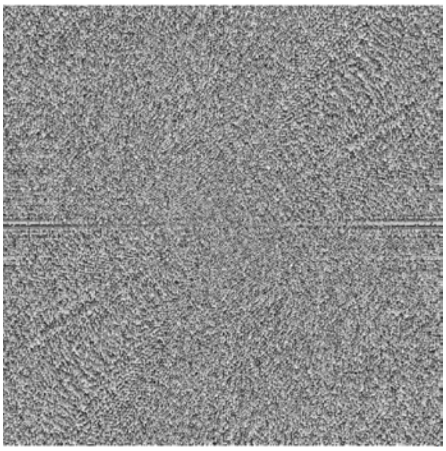
Efros

Phase and Magnitude- Yet Another Example

Amplitude



Phase



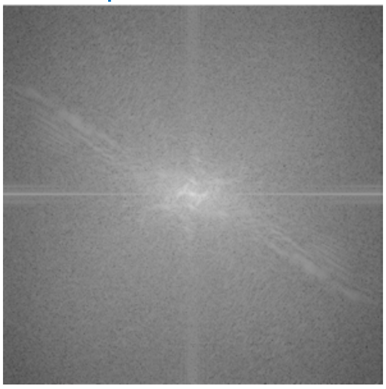
Efros

Phase and Magnitude- Yet Another Example

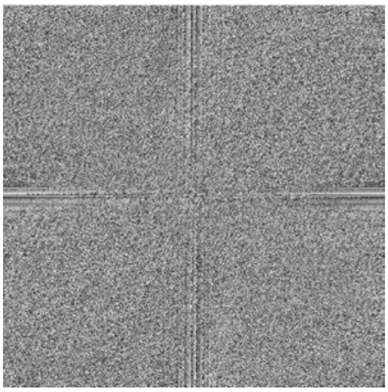
Efros

What about phase?

Amplitude



Phase

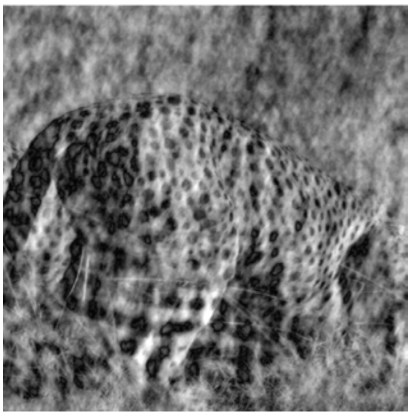


Efros

Cheebra

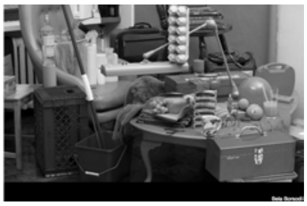
Zebra phase, cheetah amplitude

Cheetah phase, zebra amplitude



Efros

Phase and Frequency



Rotation by

90^0

Efros

Bonus question for next week

- Surprise us with hybrid images that are mixture of phase and amplitude of different images.
- Manipulate phase and frequency (e.g., by rotation) of the same image to generate interesting artifacts.
- There will be a competition!
- Extra bonus point to the winner!

Phase and Frequency

- The frequency amplitude of natural images are quite similar
  - Heavy in low frequencies, falling off in high frequencies
  - Will *any* image be like that, or is it a property of the world we live in?
- Most information in the image is carried in the phase, not the amplitude
  - Not quite clear why

Efros

Properties of the Fourier Transform

As in the 1D case FTs have the following properties

- Linearity

$$\alpha f(x,y) + \beta g(x,y) \Leftrightarrow \alpha F(u,v) + \beta G(u,v).$$

- Similarity

$$f(ax, by) \Leftrightarrow \frac{1}{ab} F\left(\frac{u}{a}, \frac{v}{b}\right).$$

This applies, for example, when an image is scaled

- Shift $$f(x-a, y-b) \Leftrightarrow e^{j2\pi(au+bv)} F(u,v)$$

This might apply, for example, if an object moved.

Properties of Fourier Transforms

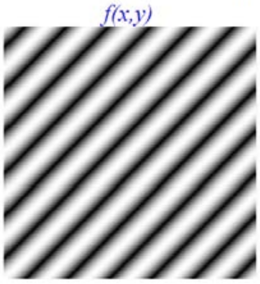
In 2D can also rotate, shear etc

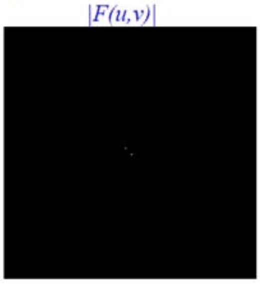
Under an affine transformation: $\mathbf{x} \rightarrow \mathtt{A}\mathbf{x}$ $\quad \mathbf{u} \rightarrow \mathtt{A}^{-\top}\mathbf{u}$

Example

How does $F(u,v)$ transform if $f(x,y)$ is rotated by 45 degrees?

$f(x,y)$ $\quad |F(u,v)|$





If $\mathtt{A} = \mathtt{R}$ then $\mathtt{A}^{-\top} = \mathtt{R}$.

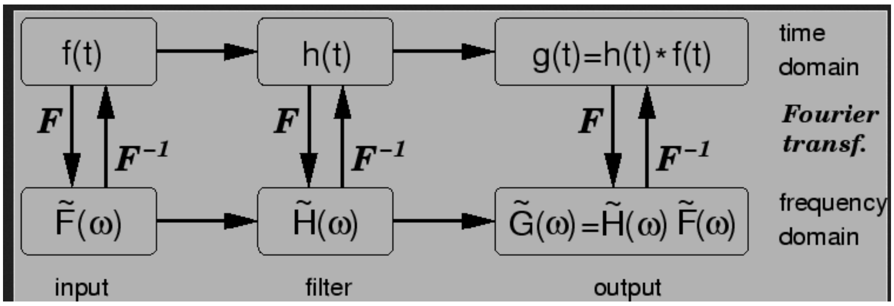
i.e. FT undergoes the same rotation.

Properties of Fourier Transforms

- Fourier transform of a real signal is symmetric about the origin
- The energy of the signal is the same as the energy of its Fourier transform

See Szeliski Book (3.4)

The Convolution Theorem



http://jclahr.com/science/psn/wielandt/node8.html

Filtering Vs. Convolution in 1D

$$g(x) = \sum_i f(x+i)h(i)$$

filtering f(x) with h(x)

f(x)	100	200	100	200	90	80	80	100	100
h(x)	1/4	1/2	1/4						
g(x)		150							

molecule/template/kernel

$$g(x) = \int f(u)h(x-u)\,du$$

convolution of f(x) and h(x)

$$= \int f(x+u')h(-u')\,du'$$

after change of variable $u' = u - x$

$$= \sum f(x+i)h(-i)$$

Filtering Vs. Convolution in 1D

$$g(x) = \sum_i f(x+i)h(i)$$ filtering f(x) with h(x)

$$g(x) = \int f(u)h(x-u)\,du$$ convolution of f(x) and h(x)

$$= \int f(x+u')h(-u')\,du'$$ after change of variable $u' = u - x$

$$= \sum f(x+i)h(-i)$$

Filtering Vs. Convolution in 1D

$$g(x) = \sum_i f(x+i)h(i)$$ filtering f(x) with h(x)

f(x)	100	200	100	200	90	80	80	100	100
h(x)	1/4	1/2	1/4						
g(x)		150							

molecule/template/kernel

$$g(x) = \int f(u)h(x-u)\,du$$ convolution of f(x) and h(x)

$$= \int f(x+u')h(-u')\,du'$$ after change of variable $u' = u - x$

$$= \sum f(x+i)h(-i)$$

Filtering Vs. Convolution in 1D

$$g(x) = \sum_i f(x+i)h(i)$$ filtering f(x) with h(x)

$$g(x) = \int f(u)h(x-u)\,du$$ convolution of f(x) and h(x)

$$= \int f(x+u')h(-u')\,du'$$ after change of variable $u' = u - x$

$$= \sum f(x+i)h(-i)$$

- note negative sign (which is a reflection in x) in convolution
- *h(x)* is often symmetric (even/odd), and then (e.g. for even)

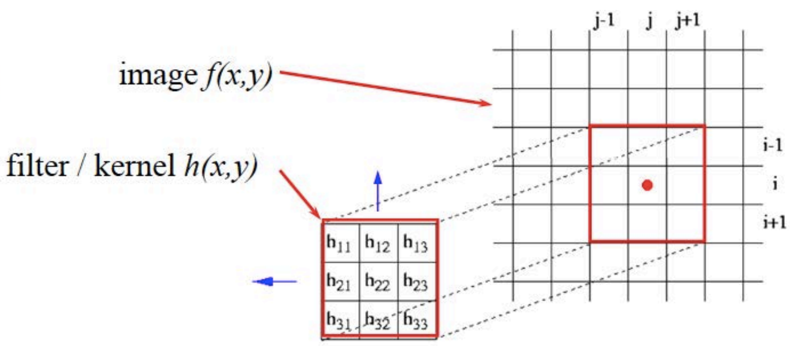
$$g(x) = \sum_i f(x+i)h(i)$$

Filtering Vs. Convolution in 2D

convolution

$$g(x,y) = h(x,y) * f(x,y) = f(x,y) * h(x,y)$$
$$= \int\int f(u,v)h(x-u,y-v)\,du\,dv$$

filtering



Filtering Vs. Convolution in 2D

convolution

$$
\begin{aligned}
g(x,y) &= h(x,y) * f(x,y) = f(x,y) * h(x,y) \\
&= \int\int f(u,v)h(x-u, y-v)\,du\,dv
\end{aligned}
$$

filtering

$$
\begin{aligned}
g(x,y) = \; & h_{11}\, f(i-1, j-1) + h_{12}\, f(i-1, j) + h_{13}\, f(i-1, j+1) + \\
& h_{21}\, f(i, j-1) + h_{22}\, f(i, j) + h_{23}\, f(i, j+1) + \\
& h_{31}\, f(i+1, j-1) + h_{32}\, f(i+1, j) + h_{33}\, f(i+1, j+1)
\end{aligned}
$$

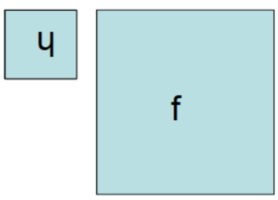
for convolution, reflect filter in x and y axes

Convolution

- Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)

$$
g[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] f[i-u, j-v]
$$

convolution with h



slide: K. Grauman

Filtering vs. Convolution in 2D Matlab

2D filtering

- `g=filter2(h,f);`

f=image
h=filter

$$g[m,n]=\sum_{k,l} h[k,l]\, f[m+k,n+l]$$

2D convolution

- `g=conv2(h,f);`

$$g[m,n]=\sum_{k,l} h[k,l]\, f[m-k,n-l]$$

Convolution Theorem

$$f(x,y) * h(x,y) \quad \Leftrightarrow \quad F(u,v)H(u,v)$$

Space convolution = frequency multiplication

In words: the Fourier transform of the convolution of two functions is the product of their individual Fourier transforms

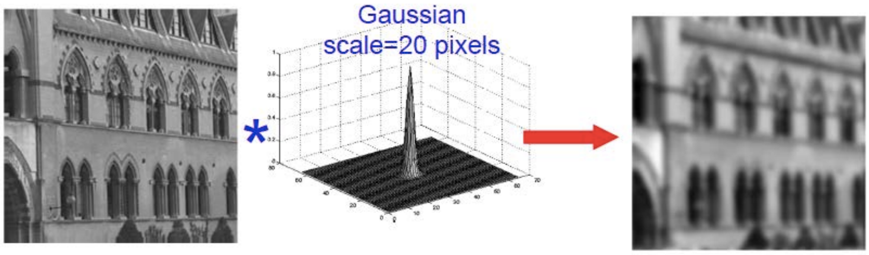
Why is this so important?

Because linear filtering operations can be carried out by simple multiplications in the Fourier domain

The Importance of Convolution Theorem

It establishes the link between operations in the frequency domain and the action of linear spatial filters

Example smooth an image with a Gaussian spatial filter

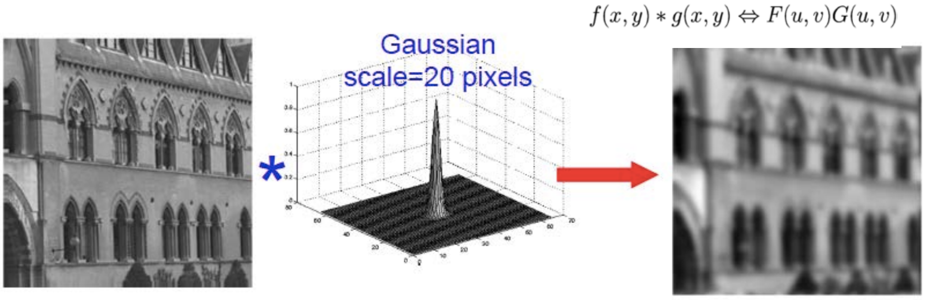


$$f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

The Importance of Convolution Theorem

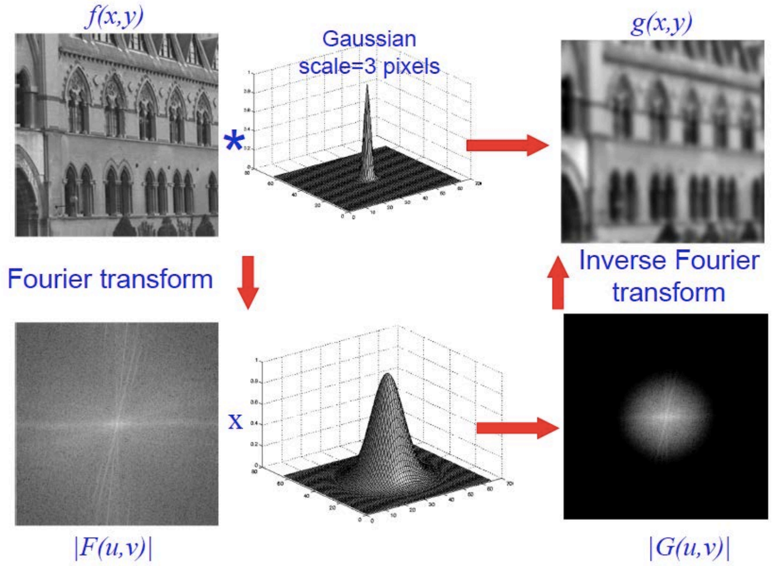
Example smooth an image with a Gaussian spatial filter

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

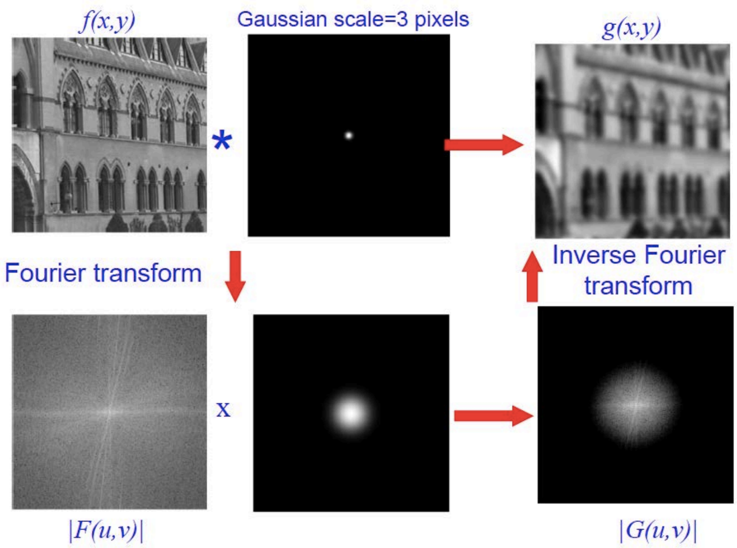


1. Compute FT of image and FT of Gaussian
2. Multiply FT's
3. Compute inverse FT of the result.

The Importance of Convolution Theorem



The Importance of Convolution Theorem



Filtering: Spatial Domain vs. Frequency Domain

There are two equivalent ways of carrying out linear spatial filtering operations:

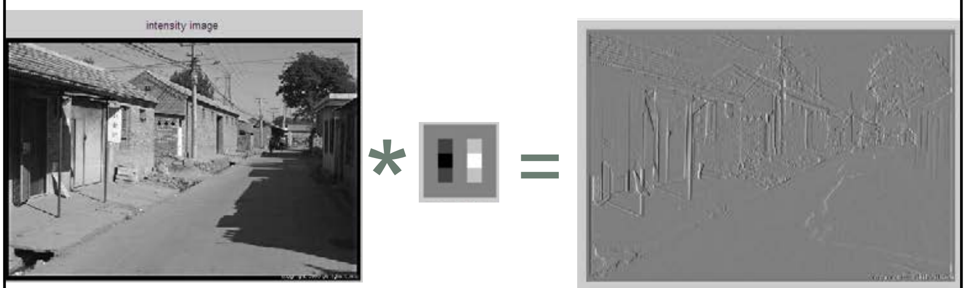
1. Spatial domain: convolution with a spatial operator
2. Frequency domain: multiply FT of signal and filter, and compute inverse FT of product

Why choose one over the other ?

- The filter may be simpler to specify or compute in one of the domains
- Computational cost

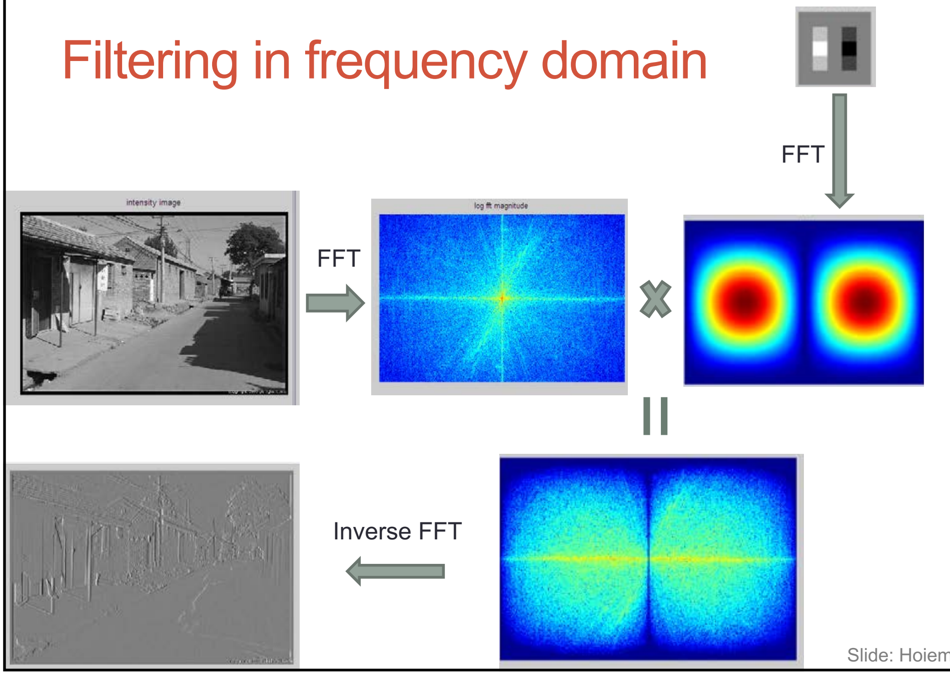
More on Filtering in spatial domain

1	0	-1
2	0	-2
1	0	-1



Hays

Filtering in frequency domain



Slide: Hoiem

Fast Fourier Transform in Matlab

```
>> I = rgb2gray(im);
>> I = double(I)/255;
>> figure;imshow(I)
>> [w,h] = size(I)

w =

   526


h =

   764
```

Fast Fourier Transform in Matlab

```
>>
>> fftsize = 1024;
>> % should be of order of 2 (for speed) and include padding

>> fs = 50; % filter half-size

>> fil = fspecial('gaussian', fs*2+1,10);
```

row &cols

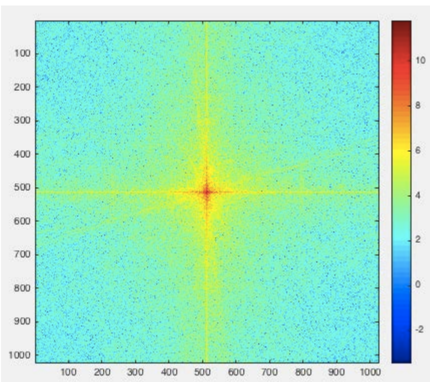
Standard Deviation

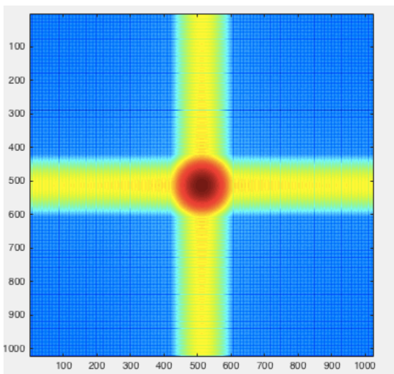
```
>>
>> im_fft = fft2(I,fftsize,fftsize);
>> fil_fft = fft2(fil,fftsize,fftsize);

>> im_fil_fft = im_fft.*fil_fft;
```

pointwise multiplication

Fast Fourier Transform in Matlab

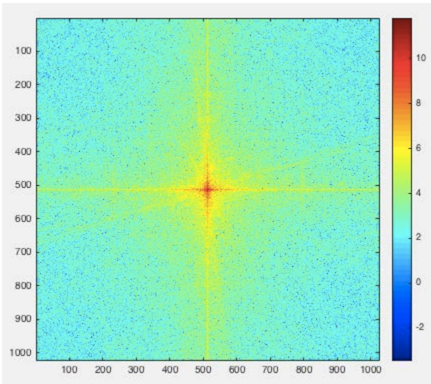


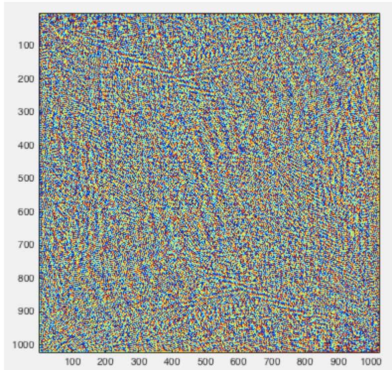



```
>> figure; imagesc(log(abs(fftshift(im_fft)))); axis image, colormap jet
>> colorbar

>> figure; imagesc(log(abs(fftshift(fil_fft)))); axis image, colormap jet
```

Fast Fourier Transform in Matlab






```
>> figure; imagesc(log(abs(fftshift(im_fft)))); axis image, colormap jet
>> colorbar
>>
>> figure; imagesc((angle((im_fft)))); axis image, colormap jet
>>
```

Fast Fourier Transform in Matlab

```
>> im_fil = ifft2(im_fil_fft);

>> im_fil_no_pad = im_fil(1+fs:size(I,1)+fs,1+fs:size(I,2)+fs);
```

Fast Fourier Transform in Matlab

```
>> figure;imshow(im_fil_no_pad);
```

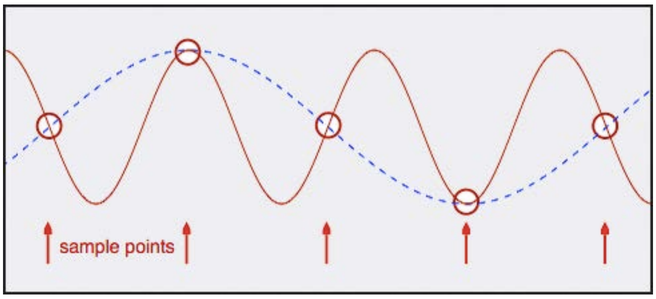
Class Work

- Read cameraman image:
  I = imread('cameraman.tif');

- Calculate its frequency spectrum with fft2

- Display the absolute value of its spectrum with and w/o fftshift
- It is recommended to present the spectral image using logarithmic scale.

Sampling Theorem

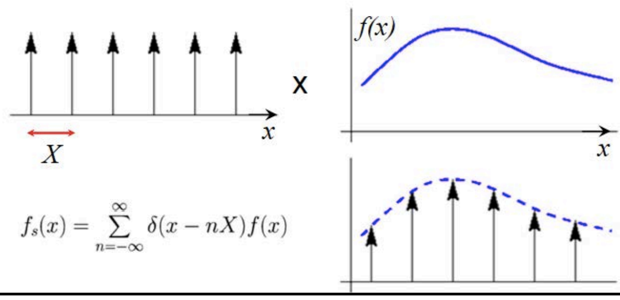


1D Sampling

In 1D model the image as a set of point samples obtained my multi-plying $f(x)$ by the comb function

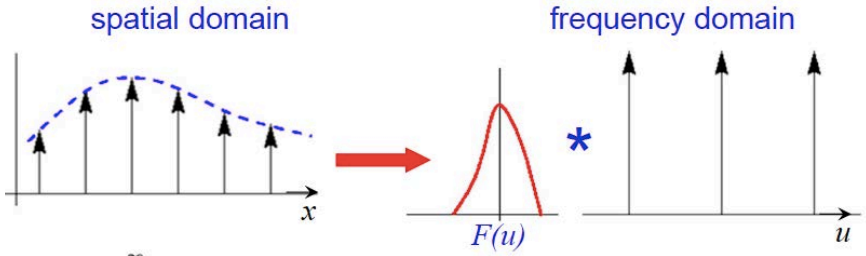
$$comb(x) = \sum_{n=-\infty}^{\infty} \delta(x - nX)$$

an infinite set of delta functions spaced by X.



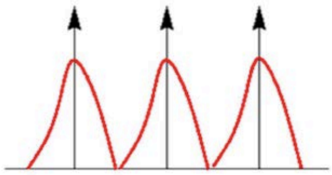
$$f_s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nX) f(x)$$

1D Sampling



$$f_s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nX) f(x)$$
$$= \sum_{n=-\infty}^{\infty} f(nX)\delta(x - nX)$$

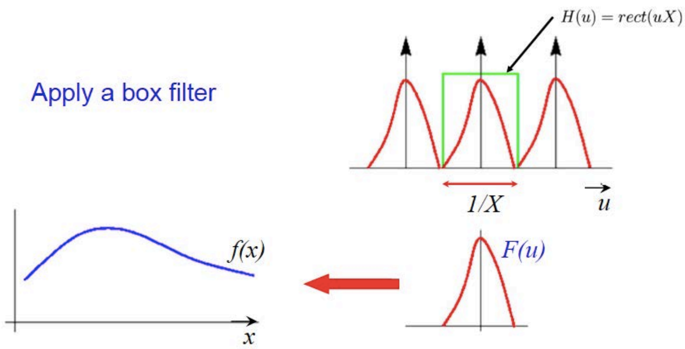
$$F_s(u) = \frac{1}{X}\sum_{n=-\infty}^{\infty} \delta(u - n/X) * F(u) = \frac{1}{X}\sum_{n=-\infty}^{\infty} F(u - n/X)$$



replicated copies of $F(u)$

1D Sampling

Apply a box filter



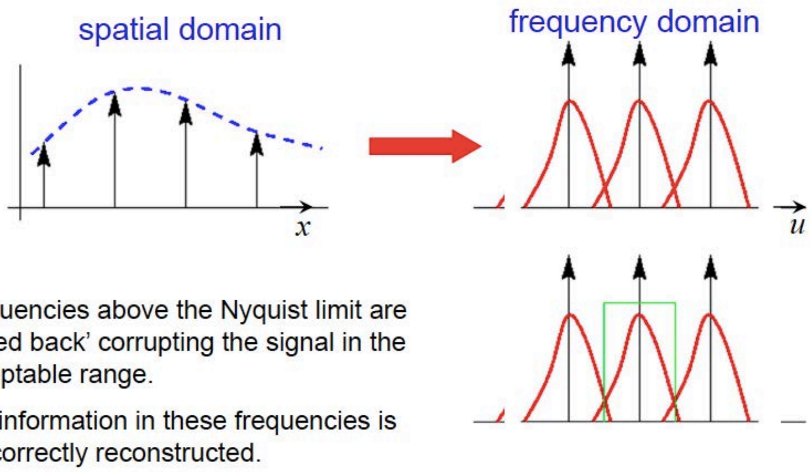
$$F(u) = F_s(u)H(u)$$

$$f(x) = \sum_{n=-\infty}^{\infty} f(nX)\delta(x - nX) * \mathrm{sinc}\frac{\pi x}{X}$$
$$= \sum_{n=-\infty}^{\infty} f(nX)\mathrm{sinc}\frac{\pi}{X}(x - nX)$$

The original continuous function $f(x)$ is completely recovered from the samples provided the sampling frequency ($1/X$) exceeds twice the greatest frequency of the band-limited signal. (Nyquist sampling limit)

The Sampling Theorem and Aliasing

if sampling frequency is reduced …



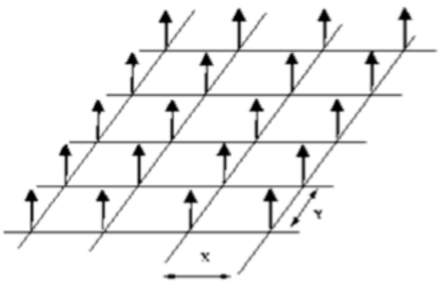
Frequencies above the Nyquist limit are 'folded back' corrupting the signal in the acceptable range.

The information in these frequencies is not correctly reconstructed.

2D Sampling

In 2D the equivalent of a comb is a bed-of-nails function

$$\sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta(x - nX)\delta(y - mY)$$



2D Sampling

In 2D the equivalent of a comb is a bed-of-nails function

$$\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta(x-nX)\delta(y-mY)$$

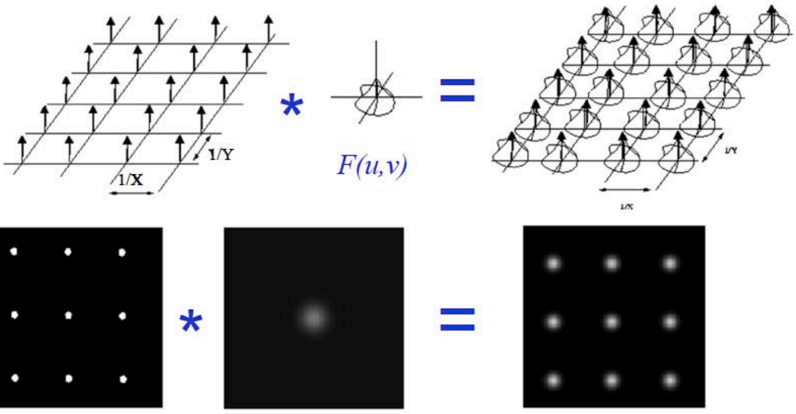
Fourier transform pairs

$$\sum_{n=-\infty}^{\infty}\delta(x-nX) \leftrightarrow \frac{1}{X}\sum_{n=-\infty}^{\infty}\delta(u-n/X)$$

$$\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta(x-nX)\delta(y-mY) \leftrightarrow \frac{1}{XY}\sum_{n=-\infty}^{\infty}\delta(u-n/X)\sum_{m=-\infty}^{\infty}\delta(v-n/Y)$$

Sampling Theorem in 2D

frequency domain



$H(u,v) = rect(uX)rect(vY)$

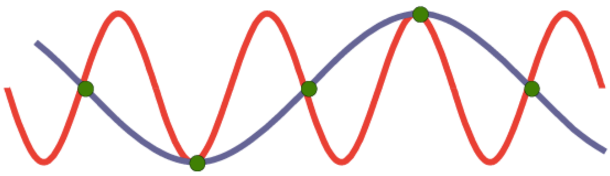
$$f(x,y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} f(nX,mY)\,\mathrm{sinc}\frac{\pi}{X}(x-nX)\,\mathrm{sinc}\frac{\pi}{Y}(y-nY)$$

Sampling Theorem in 2D

If the Fourier transform of a function $f(x,y)$ is zero for all frequencies beyond u_b and v_b,i.e. if the Fourier transform is *band-limited*, then the continuous function $f(x,y)$ can be completely reconstructed from its samples as long as the sampling distances w and h along the x and y directions are such that $w \leq \frac{1}{2u_b}$ and $h \leq \frac{1}{2v_b}$

Aliasing: 1D Example

If the signal has frequencies above the Nyquist limit …



Insufficient samples to distinguish the high and low frequency

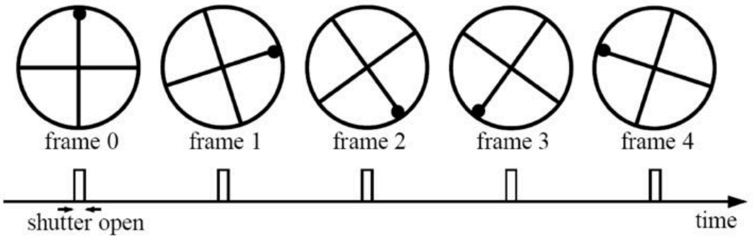
aliasing: signals "travelling in disguise" as other frequencies

Aliasing in video

Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise).
Mark wheel with dot so we can see what's happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



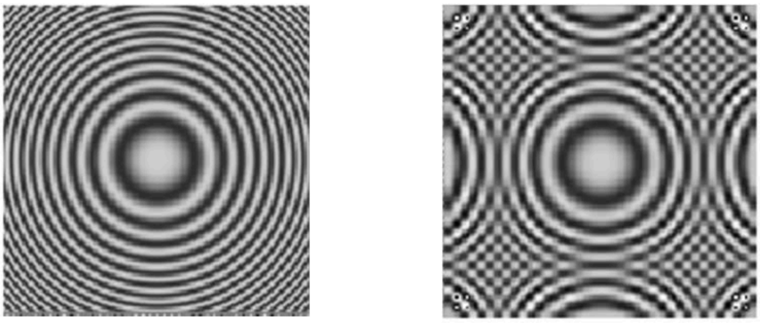
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Slide by Steve Seitz

Aliasing in 2D: under-sampling reconstruction

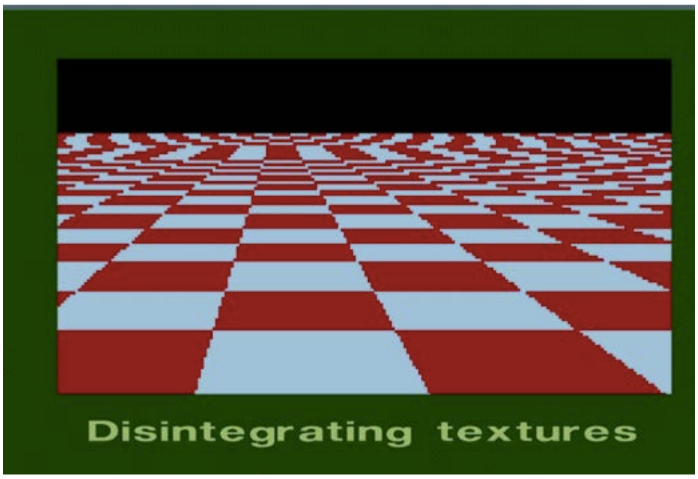
original

reconstruction



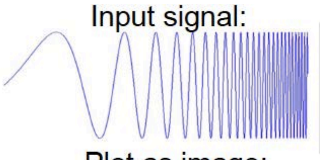
signal has frequencies above Nyquist limit

Aliasing in Images

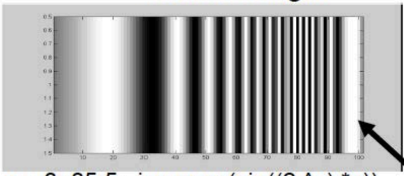


What's happening

Input signal:

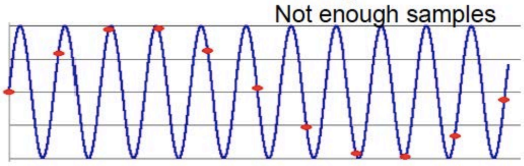


Plot as image:



x = 0:.05:5; imagesc(sin((2.^x).*x))

Aliasing
Not enough samples

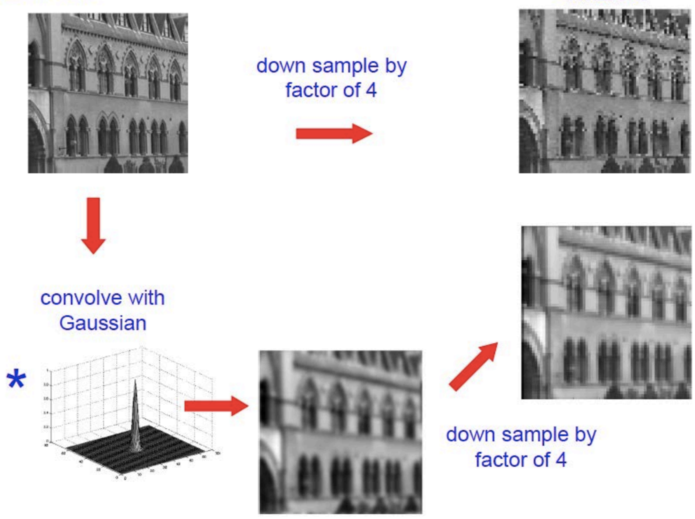


Anti-aliasing

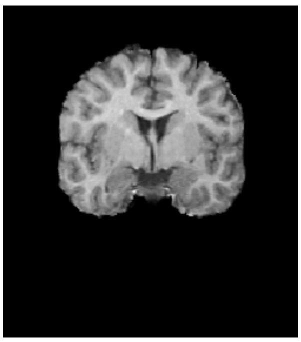
- Increase sampling frequency
  - e.g. in graphics rendering cast 4 rays per pixel
- Reduce maximum frequency to below Nyquist limit
  - e.g. low pass filter before sampling

Anti-aliasing

Example

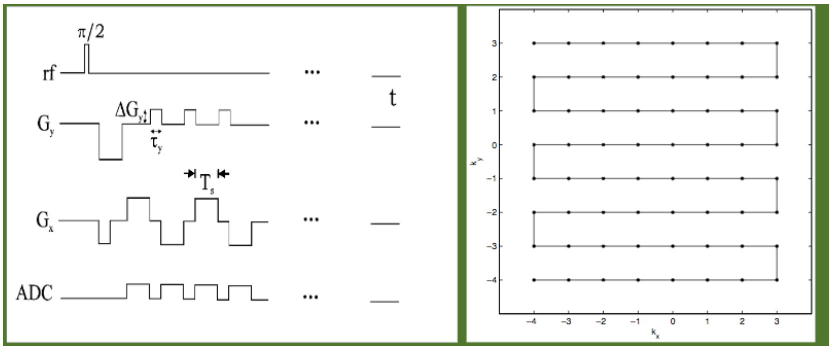


Aliasing in MRI



MRI

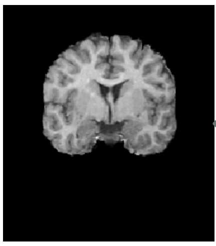
Aliasing in MRI



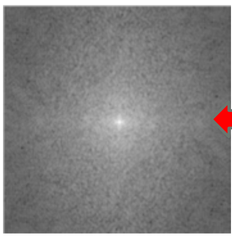
(a) Gradient Echo-EPI Pulse Sequence (b) *k*-Space Trajectory

STAT692, Wisconsin

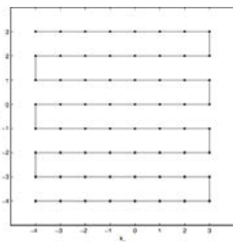
Aliasing in MRI



MRI

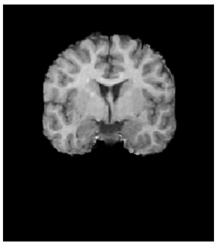


K-Space

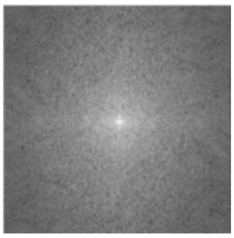


K-Space

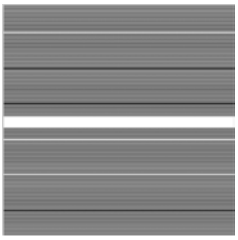
Aliasing in MRI



MRI

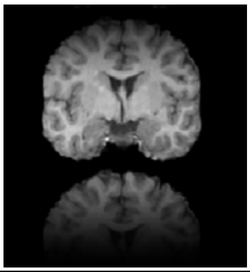


K-Space



Mask

Reconstructed
MRI



Ideas for final Projects

Solitaire Recognition

Chess recognition

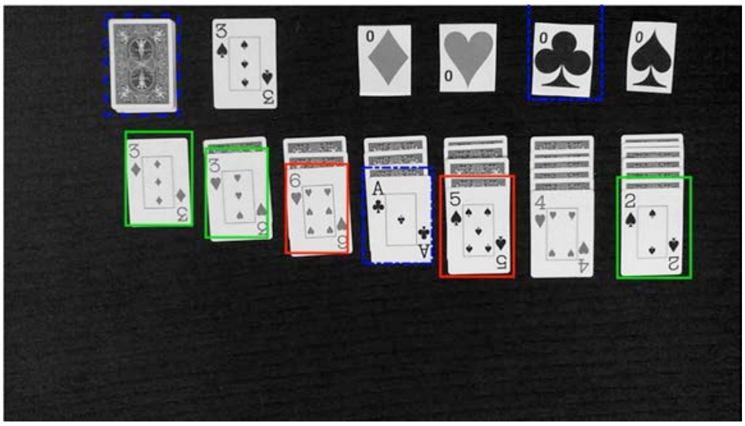
Real time Human Activity recognition

Face Recognition

Flaying Object Detection

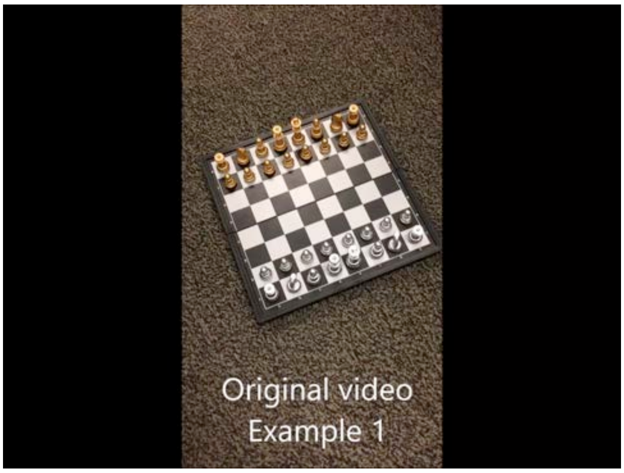
Ball detection (in soccer game)

Ideas for final Projects



EENG 512/CSCI 512 - Final Projects
Hoch, Garrett, *Solitaire Recognition*

Ideas for Final Projects

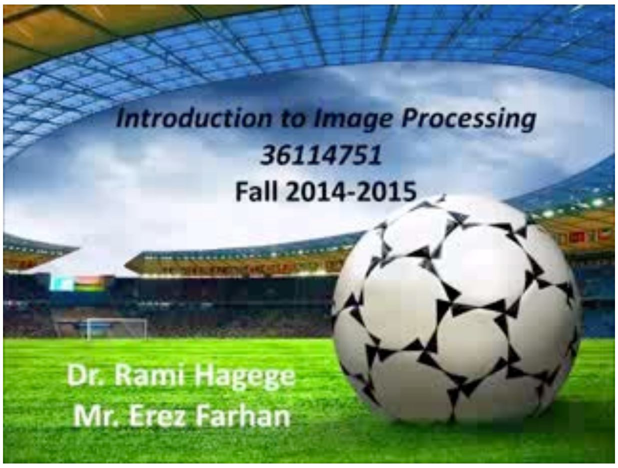


EENG 512/CSCI 512 - Final Projects
Xiao, Ke, *Chess Recognition*

Ideas for Final Projects

BGU – ECE 2013: Topaz, Ohad, Tsachi, Nadav

Ideas for Final Projects



BGU – ECE 2015:Doron, Boris, Alex

Ideas for Final Projects

BGU – ECE 2015:Nir, Tal & Shay – Interactive temple Run

Ideas for Final Projects

BGU – ECE 2013:Ariel, Tomer, Oren – Virtual Keyboard