9: Filtering

1 Announcements

2 Projects

Name	Topic	Data	Algorithms	Prelim Work	
Osman	Banana Wavelets	fprints	x	x	
Ham	News SOM	Gdelt, news feeds	SOM, word frequency count		
Ross	Tennis	OK	Monte carlo,	X	
Catalano	Social Networks	Facebook social netwok data	Scraper		
Booth	Satellite Imagery	x	X		
Silayi	Molecular Dynamics	generate	Verlet Velocity Integration		
Essiaw	Object Isolation				
Zhang	Restaurant Classification	Yelp food image data	Data mining		
Goldfeder	Lungs	Kaggle			
Andrews	Titanium Dioxide	generate	Monte Carlo		
Franklin	Lung				

3 Files to Get

C16correlations.pdf

C19Gabor.pdf

Rolls.png

Hdi-Print-Circuit-Board.jpg

4 Theory

See C16correlations.pdf: Sections 16.1 and 16.2

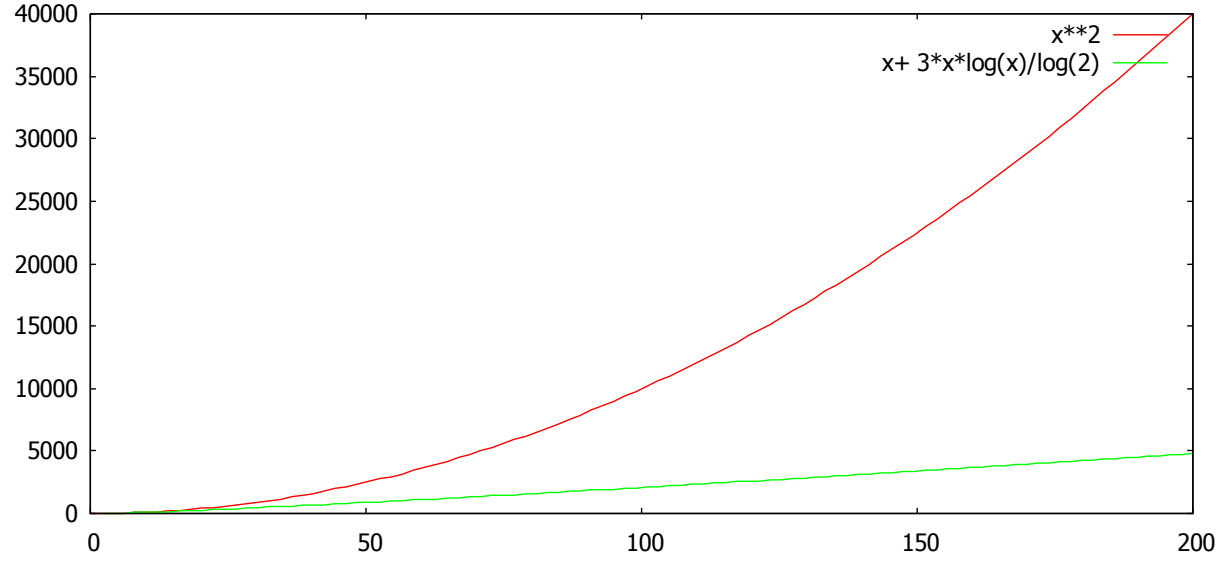
- Correlation
- Math
- Operator notation
- Simple 5x5 kernel. Scipy.signal.correlate2d
-
- Center of correlation should be the dot product.
- Correlations are shifted dot products – Inherently **first order**.

```
>>> a = np.random.ranf( (7,7) )
>>> c = ss.correlate2d( a,a )
>>> c.shape
(13, 13)
>>> import scipy.misc as sm
>>> sm.imsave('dud.png', c )
>>> a = np.random.ranf( (7,7) )-0.5 # zero sum
>>> c = ss.correlate2d( a,a )
>>> sm.imsave('dud.png', c )
```

5 Fourier Transforms and Correlations

See C16correlations.pdf: Sections 16.3

```
gnuplot> plot [0:200] x**2, x+ 3*x*log(x)/log(2)
```



- Theory ONLY: Theory combining Fourier and correlations.

6 Rolls Royce

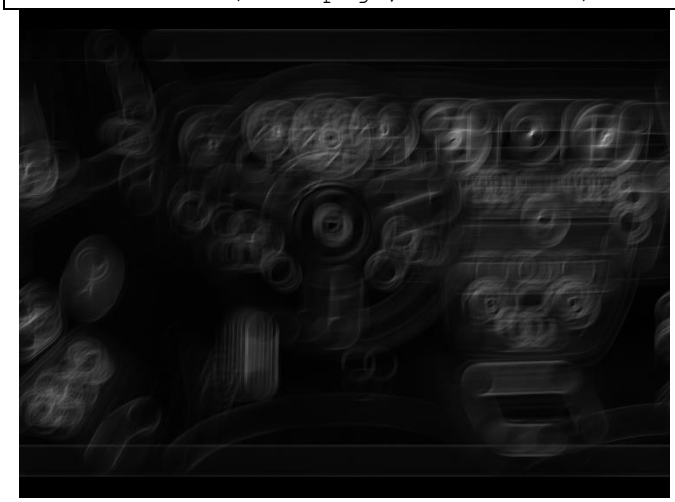
- Measure size of one of the circles.

- Make filter for that size.
- Correlate.

```
>>> exec(open('gogabor.py').read())
>>> fname = 'images/rolls.png'
>>> corr =Rolls( fname )
```

-
- Comment on how bad this idea worked.
- Edge Enhance the target.
    - Why does this work better?

```
>>> corr = Rolls2( fname )
>>> sm.imsave('dud.png', corr.real )
```



7 Boats

Use C16correlations.pdf

8 Invariances

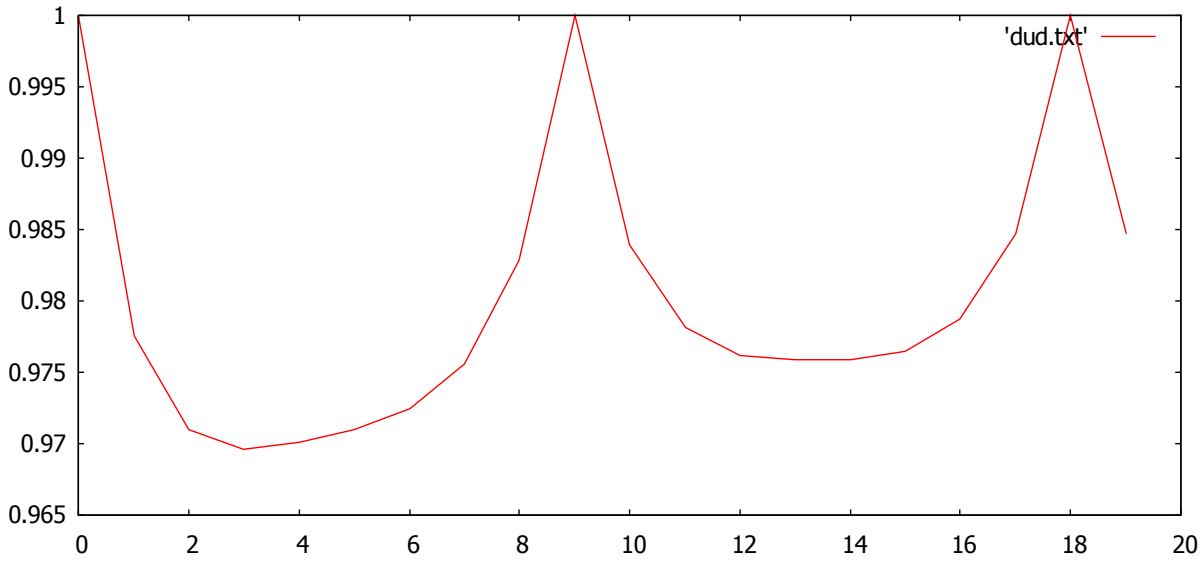
- Shift
- Rotation
    - Rotate the filter
    - Rpolar – smooth horizontally – IRPolar
    - Sensitivity: dependent on the shape. Pencils are more sensitive than circles.
- Scale
- Others: Deformation, out of plane rotation, occlusion, noise

9 Composite Filtering

C16correlations.pdf section; 16.4

- Theory. Include pseudo inverse.
- What is Q?
- Tach example
- Limitations: Dimension of Q
- Only the center is constrained. Large sidelobes possible.
- Change values of the constraint vector (-1,+1) or (theta)
- MACE
- FPF
- Dual FPF
- Circuit board: http://image.made-in-china.com/2f0j00SejtQbvnwMcw/Hdi-Print-Circuit-Board.jpg
    - Resize clean, grayscale
    - Create 2 FPFs.
    - Correlate with 0-degrees to confirm.
    - Correlate with another angle.
    - Correlate with many angles to get telephone poles.
    - Optimize alpha
    - Random testing orientation

```
>>> a = LoadBoard( 'images/board.jpg')
>>> sm.imsave('dud.png', a  )
>>> for i in range( 20 ):
      a = nd.rotate( data, i/3., reshape=False )
      v[i]= (filt2*a).sum().real*V*H
>>> filt = filt.reshape( (V,H) )
>>> filt2 = ft.ifft2( filt )
>>> sm.imsave('dud.png', filt2.real )
>>> filta = filt + 0
>>> filtb = Filter2( data )
>>> filt3 = ft.ifft( filtb.reshape(V,H))
>>> sm.imsave('dud.png', filt3.real )
>>> exec(open('gogabor.py').read())
>>> filtb = Filter2( data )
>>> filt3 = ft.ifft2( filtb.reshape(V,H))
```



10 Gabor Filters

C19Gabor.pdf

Osman's project will extend this.

```
>>> a = gabor.GaborCos( (256,256), .2, 0, 1,1 )
>>> sm.imsave('dud.png', a )
>>> A = ft.fft2( a )
```

11 Noise

- Additive

```
>>> sm.imsave('dud.png', rolls )
>>> V,H = rolls.shape
>>> noise = np.random.ranf( (V,H) ) * 75
>>> sm.imsave('dud.png', rolls + noise )
```

- Multiplicative.
- Compare FFT of image with FFT of image with additive noise. It's high frequency.
- Colored noise

```
>>> Rolls = ft.fft2( rolls )
>>> A = Rolls*noise
>>> a = ft.ifft2( A )
>>> sm.imsave('dud.png', a.real )
```