Principle Components for Real World Data

Presentation Outline

- Show real world data.
- Calculate mean and covariance.
- Show code.
- Calculate SVD.
- Show principle components.
- Show 2 decomposed images.
- Conclusion

OUR DATA:

715,244 black and white jpegs of nature

64 x 64 pixels

$x_k \in \mathbb{R}^{4096}$

Calculate Mean and Covariance

$$\text{Mean } = \mu \in \mathbb{R}^d,\ \mu = \frac{1}{L}\sum_{i=1}^{L} x_i,\ L = 715,244\ \ d = 4096$$

$$\text{Covariance } = \Sigma \in \mathbb{R}^{d\times d},\ \sigma_{i,j} = \frac{1}{L}\sum_{k=1}^{L}\left(\left(x_k\right)_i - \mu_i\right)\left(\left(x_k\right)_j - \mu_j\right) \quad \text{element wise}$$

$$\Sigma(:,j) = \frac{1}{L}\sum_{k=1}^{L}\left(x_k - \mu\right)\left(\left(x_k\right)_j - \mu_j\right) \quad \text{column wise}$$

$$\Sigma = \tfrac{1}{L} XX^T - \mu\mu^T \quad \text{entire matrix}$$

The Code

```
n=400300;
avg=avg(:);
y=zeros(4096,4096);
ct=0;
for i = 1:n
  if exist(strcat('./pics64GG/imageGG64_',num2str(i),'.jpg'))
    x=cast(imread(strcat('./pics64GG/imageGG64_',num2str(i),'.jpg'),'jpg'),'double');
    x=x(:);
    ct=ct+1;
    for k=1:4096
      y(:,k)=y(:,k)+(x-avg)*(x(k)-avg(k));
    end
  end
  if (i~=314647 && i~=328594 && i~=400246) %bad images
    if exist(strcat('./pics64/image64_',num2str(i),'.jpg'))
      x=cast(imread(strcat('./pics64/image64_',num2str(i),'.jpg'),'jpg'),'double');
      x=x(:);
      ct=ct+1;
      for k=1:4096
        y(:,k)=y(:,k)+(x-avg)*(x(k)-avg(k));
      end
    end
  end
  if (mod(i,10)==0) i
  end
end
covar=y/ct;
```

Calculating the covariance column wise proved fastest in Matlab.

Calculate SVD

$$\Sigma = PDP^T$$

$P, D \in \mathbb{R}^{4096\times 4096}$

P is orthonormal ($i.e.\ PP^T = I$)

D is ordered positive diagonal

Note: If $XX^T = P\bar{D}P^T$ then

$\Sigma = \frac{1}{L}XX^T - \mu\mu^T$

$= P\left(\frac{1}{L}\bar{D}\right)P^T - P\left(P^T\mu\mu^T P\right)P^T$

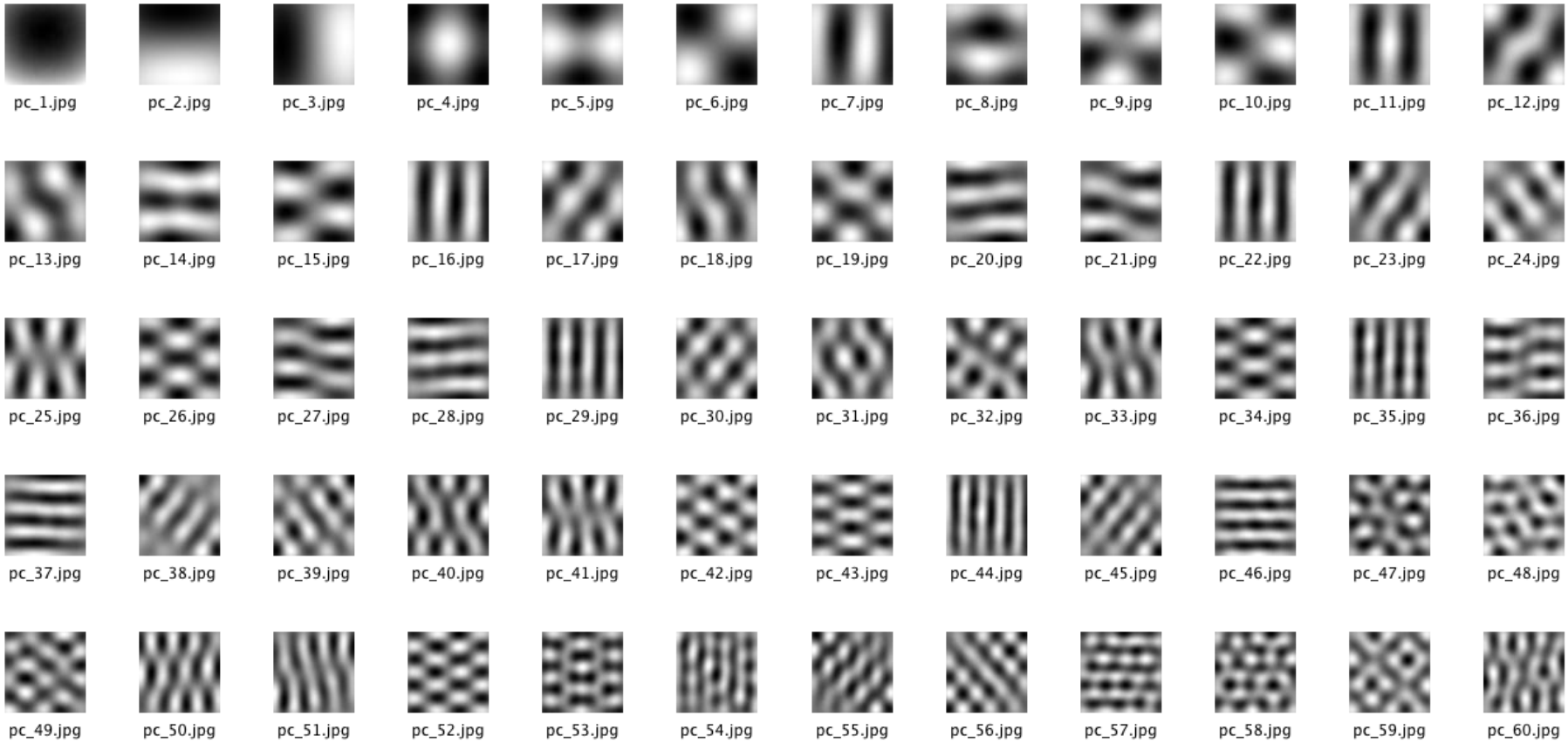
$= P\left(\frac{1}{L}\bar{D} - qq^T\right)P^T$ where $q = P^T\mu$

so, $D = \frac{1}{L}\bar{D} - qq^T$

```
D(1,1) = 4036644.859
D(100,100) = 2782.439406
D(200,200) = 1197.804021
D(300,300) = 685.186182
D(400,400) = 447.401744
D(500,500) = 314.774249
D(600,600) = 232.873262
D(700,700) = 177.366133
D(800,800) = 139.750178
D(900,900) = 111.909932
D(1000,1000) = 91.102820
D(1100,1100) = 74.875001
D(1200,1200) = 62.283436
D(1300,1300) = 52.581428
D(1400,1400) = 45.019721
D(1500,1500) = 38.745885
D(1600,1600) = 34.088773
D(1700,1700) = 29.828843
D(1800,1800) = 26.409134
D(1900,1900) = 23.052920
D(2000,2000) = 20.132907
```

```
D(2100,2100) = 17.267350
D(2200,2200) = 14.789627
D(2300,2300) = 12.857765
D(2400,2400) = 11.563492
D(2500,2500) = 10.332990
D(2600,2600) = 7.842264
D(2700,2700) = 6.353235
D(2800,2800) = 5.557154
D(2900,2900) = 4.752576
D(3000,3000) = 3.615194
D(3100,3100) = 2.562477
D(3200,3200) = 2.097061
D(3300,3300) = 1.740729
D(3400,3400) = 1.373363
D(3500,3500) = 1.026713
D(3600,3600) = 0.778342
D(3700,3700) = 0.549040
D(3800,3800) = 0.448699
D(3900,3900) = 0.370568
D(4000,4000) = 0.301966
D(4096,4096) = 0.227236
```

The Principle Components



The palette for these vectors is black for -1, white for 1, and grayscale applied linearly for in between values.

Decomposition of image64_1.jpg

$$x = x^{(4096)} = \sum_{i=1}^{4096} \alpha_i p_i$$

$$x^{(k)} = x^{(4096)} \text{ for } 3910 \le k \le 4096$$

$$\text{but } x_{19,19}^{(3909)} = 112 \neq 111 = x_{19,19}^{(4096)}$$


x10.jpg

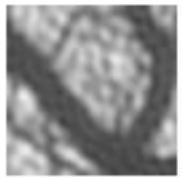

x20.jpg


x30.jpg


x50.jpg

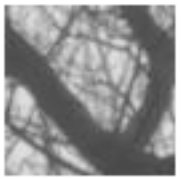

x100.jpg


x200.jpg


x500.jpg


x1000.jpg


x2000.jpg

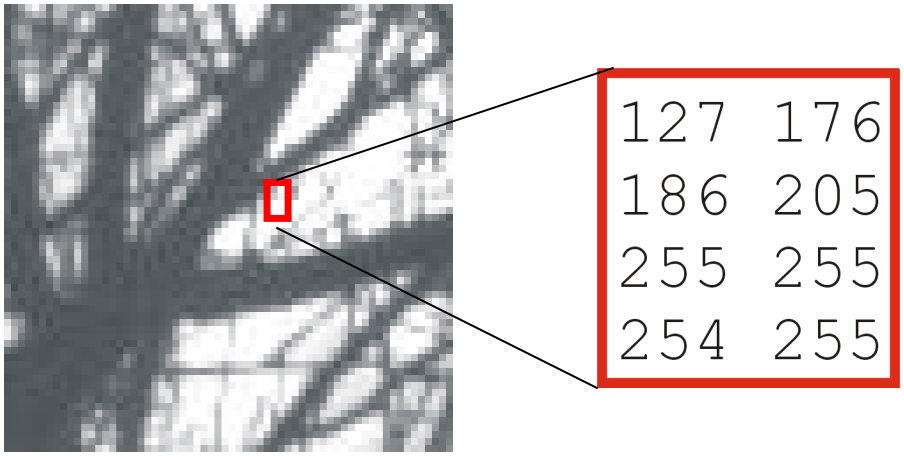

x3000.jpg


x3909.jpg


x4096.jpg

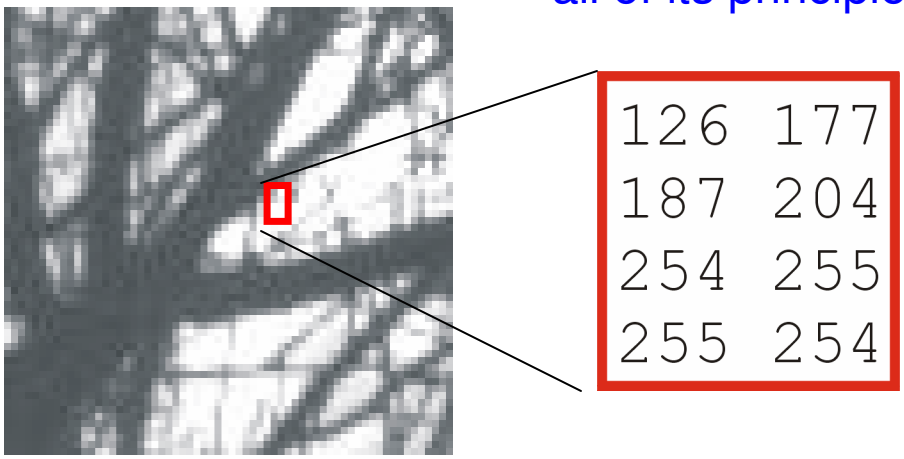
This image can be perfectly represented with limited principle components

Decomposition of image64_326.jpg



$$x = x^{(4096)} = \sum_{i=1}^{4096} \alpha_i p_i$$

This image can only be perfectly represented with all of its principle components as shown here.



$$y = x^{(4095)} = \sum_{i=1}^{4095} \alpha_i p_i$$

Results and Conclusions

- This procedure allows us to approximate and/or represent perfectly our data with less principle components than dimensions.
- Half of the images can be perfectly represented with only 3304 principle components (determined by analyzing every image)
- 2.7% of the images need all of the principle components for perfect representation (determined by analyzing every image)
- I suspect if the data were less varied we could approximate all our data exactly with a limited number of principle components.
- It would be fun and enlightening to redo this project with images that are more similar like photographs of faces.