SepFuentes

January 1, 2015

1 Introducción

El objetivo de esta práctica es la aplicación de las técnicas de Análisis por Componentes Principales (*PCA, del inglés*) y de Análisis por Componentes Independientes (*ICA, del inglés*), para separas tres imágenes que se encuetran mezcladas para extraer las componentes y la matríz de mezcla.

2 Fundamento Teórico

2.1 Problema Propuesto

Se han provisto tres imágenes de 512×512 píxeles cada una conteniendo una mezcla lineal de tres imágenes originales. Como se mostrará en la siguiente sección, dos de las imágenes originales pueden distinguirse claramente (una mujer joven y un gato). Sin embargo, la tercera imágen no es claramente identificable por lo que está oculta.

Las imágenes a utilizar no contienen ningún componente de ruido. El modelo de mezcla utilizado se puede describir como:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

en donde $\mathbf{y}$ son la imágenes mezcladas, $\mathbf{x}$ las imágenes originales y $\mathbf{A}$ es la matrix de mezcla (dimensión 3×3). Se desconoce tanto la matriz de mezcla $\mathbf{A}$ como las componentes originales $\mathbf{x}$. Este problema, en general, carece de solución determinada. Sin embargo, en este trabajo se utilizaran las técnicas deAnálisis de Componentes Principales y Análisis por Componentes Independientes para estimar ambos elementos bajo ciertas condiciones impuestas. La aplicación de estos métodos también tiene como objetivo la identificación de la tercera imagen original que se encuentra oculta.

2.2 Análisis de Componentes Principales

El Análisis de Componentes Principales (PCA, del inglés), es un prodedimiento estadístico para transformar un conjunto de observaciones mediante una transformación ortogonal a otro conjunto de componentes que esten descorrelacionadas, denominadas componentes principales.

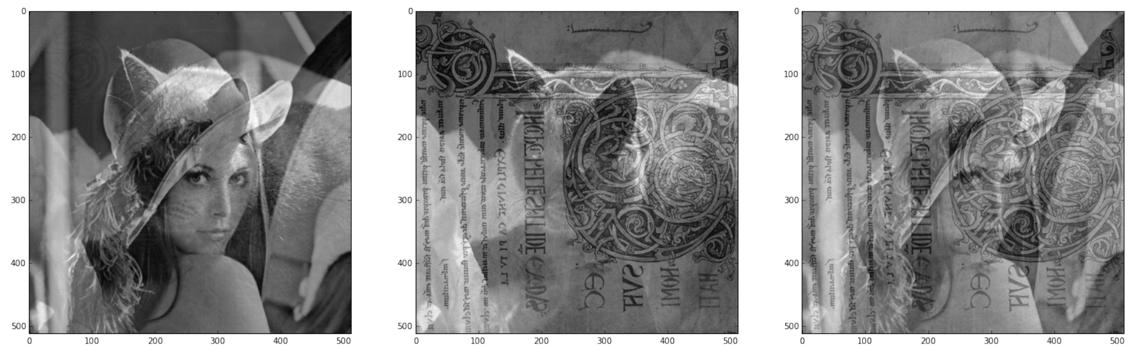
Esta tranformación se define de tal modo que cada componente tenga la máxima varianza posible bajo la condición de que la transfromación se ortogonal a las componentes anteriores. El procedimiento permite, de forma no paramétrica, la compresión o representación en un sistema de coordenadas que explique mejor la varianza de los datos.

In [2]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from sklearn import decomposition
```

In [3]:
```
img_1 = np.genfromtxt('imagen_mezclada_uno.dat', dtype='float64')
img_2 = np.genfromtxt('imagen_mexclada_dos.dat', dtype='float64')
img_3 = np.genfromtxt('imagen_mexclada_tres.dat', dtype='float64')
```

```
f, (ax1,ax2,ax3) = plt.subplots(1,3)
f.set_size_inches((24,40))
ax1.imshow(img_1, cmap=plt.cm.gray)
ax2.imshow(img_2, cmap=plt.cm.gray)
ax3.imshow(img_3, cmap=plt.cm.gray)
```

Out[3]: <matplotlib.image.AxesImage at 0x107dd1a50>



In [4]:
```
# each image is flattened and all three are stacked together
X = np.vstack((img_1.flatten(),img_2.flatten(),img_3.flatten()))
X.shape
```

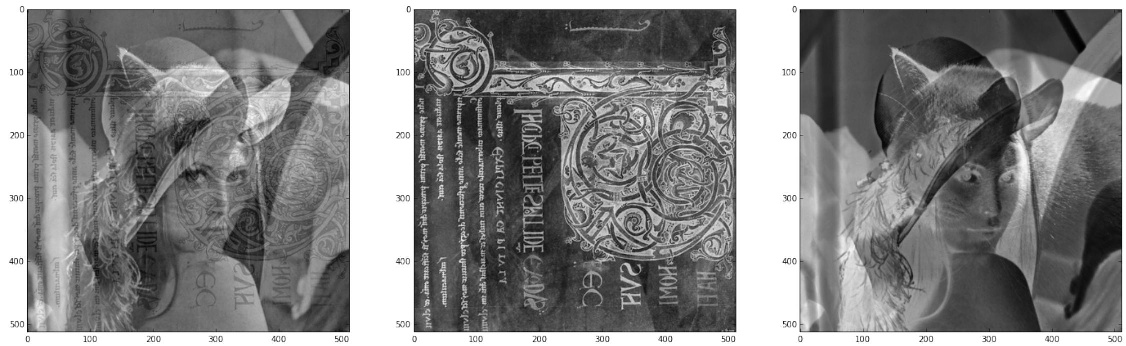
Out[4]: (3, 262144)

In [24]:
```
pca = decomposition.PCA(n_components=3)
img_pca= pca.fit_transform(X.T)
f, (ax1,ax2,ax3) = plt.subplots(1,3)

print "Explained variance:\n", pca.explained_variance_
print "Explained variance ratio:\n", pca.explained_variance_ratio_
print "Inverse of Mixing Matrix:\n", pca.components_
print "Mixing Matrix:\n", np.linalg.inv(pca.components_)
f.set_size_inches((24,40))
ax1.imshow(img_pca[:,0].reshape((512,512)), cmap=plt.cm.gray)
ax2.imshow(img_pca[:,1].reshape((512,512)), cmap=plt.cm.gray)
ax3.imshow(img_pca[:,2].reshape((512,512)), cmap=plt.cm.gray)
```

```
Explained variance:
[ 5.72141859  0.90953297  0.189357  ]
Explained variance ratio:
[ 0.83887973  0.13335657  0.0277637 ]
Inverse of Mixing Matrix:
[[ 0.65909172  0.14263253  0.73841321]
 [ 0.74315636 -0.27417719 -0.61036505]
 [ 0.11539815  0.95104302 -0.28670619]]
Mixing Matrix:
[[ 0.65909172  0.74315636  0.11539815]
 [ 0.14263253 -0.27417719  0.95104302]
 [ 0.73841321 -0.61036505 -0.28670619]]
```

Out[24]: <matplotlib.image.AxesImage at 0x114d65d50>



In [25]:
```
np.random.seed(20)

ica = decomposition.FastICA(n_components=3)
img_ica= ica.fit_transform(X.T)

print "Mixing Matrix:\n",ica.mixing_
print "Unmixing Matrix:\n",ica.components_

f, (ax1,ax2,ax3) = plt.subplots(1,3)
f.set_size_inches((24,40))
ax1.imshow(img_ica[:,0].reshape((512,512)), cmap=plt.cm.gray)
ax2.imshow(img_ica[:,1].reshape((512,512)), cmap=plt.cm.gray)
ax3.imshow(img_ica[:,2].reshape((512,512)), cmap=plt.cm.gray)
```

```
Mixing Matrix:
[[-567.29852888  675.60652116  -74.83530837]
 [-241.26704961  -37.39529646  183.64472666]
 [-419.33633585  630.17696537  581.14653985]]
Unmixing Matrix:
[[-0.00103173 -0.00330088  0.00091023]
 [ 0.00047438 -0.00271003  0.00091747]
 [-0.00125886  0.00055686  0.00138266]]
```

Out[25]: <matplotlib.image.AxesImage at 0x10e350910>

```
In [105]:
```

```
In []:
```