Principal component analysis and image compression

Zhicong Chu

December 3, 2015

1. Introduction

Principal Component Analysis (PCA) is a mathematic formulation widely used for data reduction and interpretation. The PCA technique allows the reduction of the dimension of data sets without much loss of information, making it well suited for digital image compression. Here, we briefly test the performance of PCA on image compression by exploring the compression ratios as well as the quality of reconstructed images from **principal components (PCs)**.

2. Data

The data set here is a matrix containing pixel values of a grayscale image. We first load the test image into R and get a 1024 by 574 by 3 array. The third dimension index 3 represents the 3 RGB channels. Then with the help of a transformation formula, we are able to convert RGB color space into single-channel grayscale space. And the resulting object from the conversion is a 2-dimensional 1024 by 574 array also called a matrix. So that matrix is our data set to be analyzed. Its correspondent grayscale image is defined as the original image.

Figure 1: The original image

The data has a structure of 1024 rows and 574 columns. The columns are considered as variables. So the data contains 1024 observations in 574 variables. Since the image is imported via the function readJPEG( ), the pixel values are reals between 0 and 1. After the conversion from 3 channels (RGB) to 1 channel (Grayscale), the scale of the values remain the same. Having all the variables in the same scale, we can conduct PCA directly using the covariance matrix instead of the correlation matrix from the standardized data. The histograms of the first 16 columns namely the first 16 variables of the data matrix, reveal the distribution of the pixel values. As expected, they all fall in the range of 0 to 1. Though the distributions are highly skewed with most of the values close to 0 and a decreasing portion of values as it gets closer to 1, it will not affect our PCA.

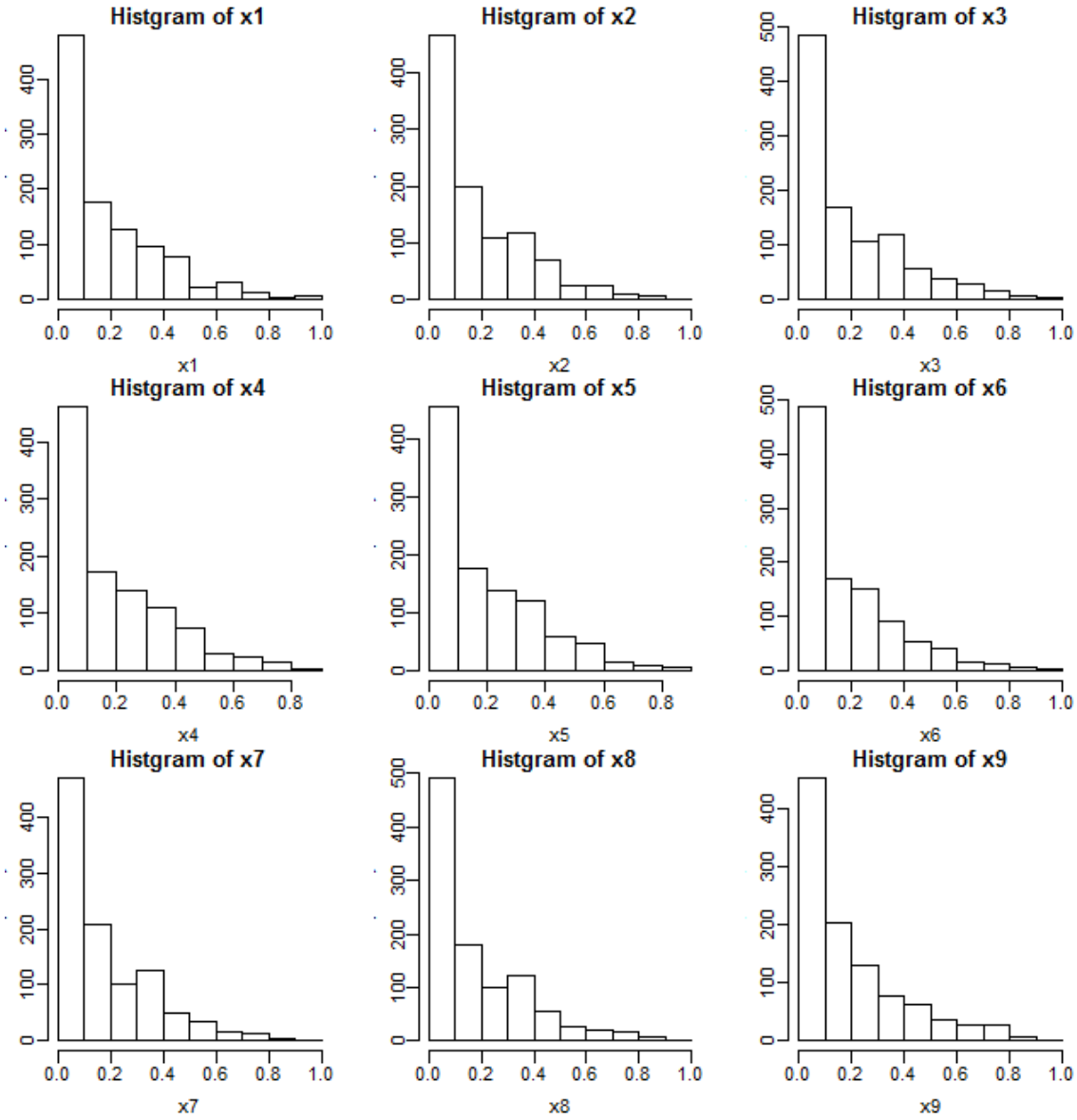

Figure 2: Histograms of the first 16 variables of the data set

3. Theory

a) Assumption

To conduct the PCA directly from covariance matrix $\sum$, we assume that all the variables are approximately in the same scale. As we mentioned in the Data Section, all the entries of the data set are reals between 0 and 1. The assumption is verified.

b) Principal components

Let random n by p matrix $X = [X_1, X_2, \dots, X_p]$ have the covariance matrix $\sum$ with eigenvalue-eigenvector pairs (e_1, λ_1), (e_2, λ_2), $\dots$, (e_p, λ_p). Then the ith principal component is given by

$$Y_i = \boldsymbol{e_i}'\boldsymbol{X}' = e_{i1}X_1 + e_{i2}X_2 + \cdots + e_{ip}X_p, \text{ i=1,2,...,p}$$

And we put the first m PCs in a m by n matrix Y_M so that

$$Y_M{}' = [Y_1, Y_2, \dots, Y_m]$$

c) Image reconstruction

We use PCs to reconstruct the image. If the original image matrix X has n rows and p columns, then the reconstructed image will have the same dimension of n by p. The formula of approximation of X from the first m PCs is as follows:

$$\widetilde{\mathrm{X}} = (\boldsymbol{e_M} Y_M)' = (\boldsymbol{e_M}\boldsymbol{e_M}'\boldsymbol{X}')'$$

Y_M is a m by n matrix which has the first m PCs in rows.

$\boldsymbol{e_M} = \begin{bmatrix} e_{11} & \cdots & e_{m1} \\ \vdots & \ddots & \vdots \\ e_{1p} & \cdots & e_{mp} \end{bmatrix}$ is an p by m matrix which has the first m eigenvectors in columns.

d) Compression ratio

The size of the original picture is determined by the image matrix X. To display the compressed image, we need to get the approximation matrix $\tilde{X}$ from the PCs. $\tilde{X}$ has the same dimension with X, so

$$S_{\tilde{X}} = S_{\mathrm{X}} = S_o$$

$S_{\tilde{X}}$ and S_{X} denote the size to store the matrix $\tilde{X}$ and X respectively. And S_o denotes the size of the original image.

However, the reconstruction is conducted through the multiplication of two matrices e_M and Y_M. So the size of the compressed image is actually determined by these two matrices.

$$S_c = S_{e_M} + S_{Y_M}$$

S_c denotes the size required to store the information of the compressed image, S_{e_M} and S_{Y_M} denote the size to store the matrix e_M and Y_M respectively. And, then the compression ratio R is given by

$$\mathrm{R} = \frac{S_o}{S_c} = \frac{S_o}{S_{e_M} + S_{Y_M}}$$

R is usually represented by the value R or the ratio R:1. When $S_c < S_o$, R should be a positive value greater than 1. R with a greater value indicates better compression efficiency.

4. Analysis and results

The first several PCs have much greater eigenvalues thus explain large portions of the total variance. This is verified in both the scree plot and the plot of the cumulative variance. Concretely, the first 5 PCs explain 46.2% of the total variance. The first 20 PCs can explain more than 65% of the variance. And the first 100 principal components explain as high as 92.9% of the total variance.

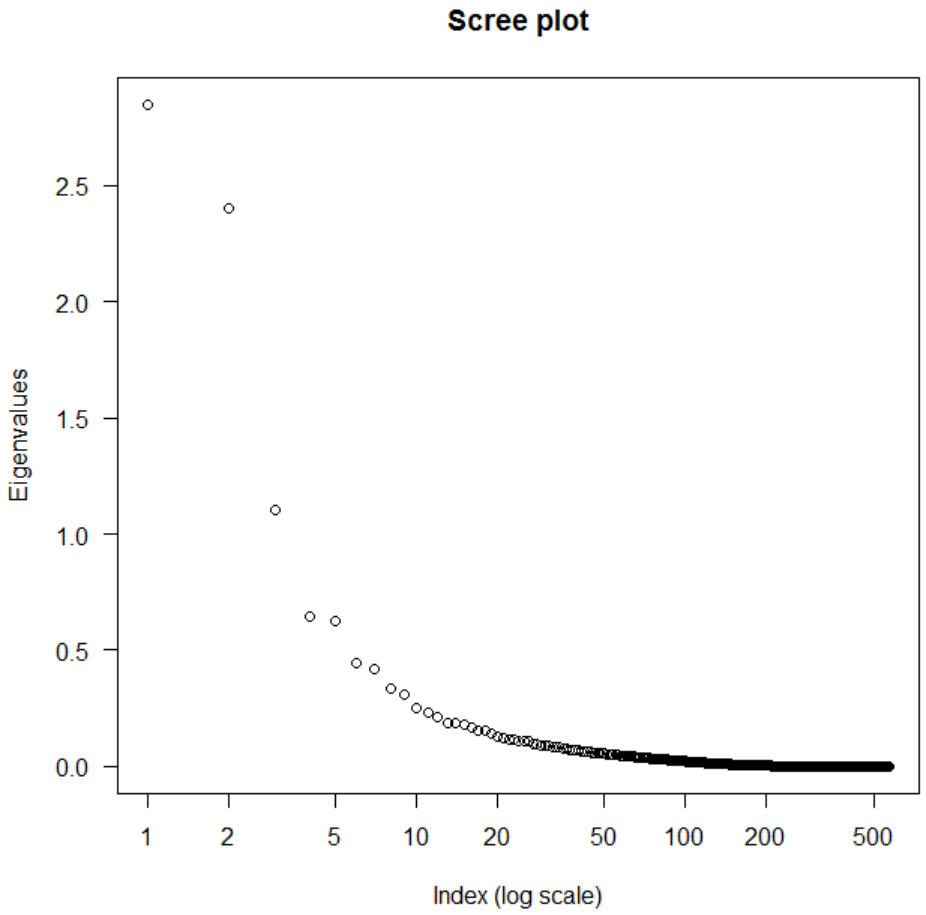

Figure 3: Scree plot

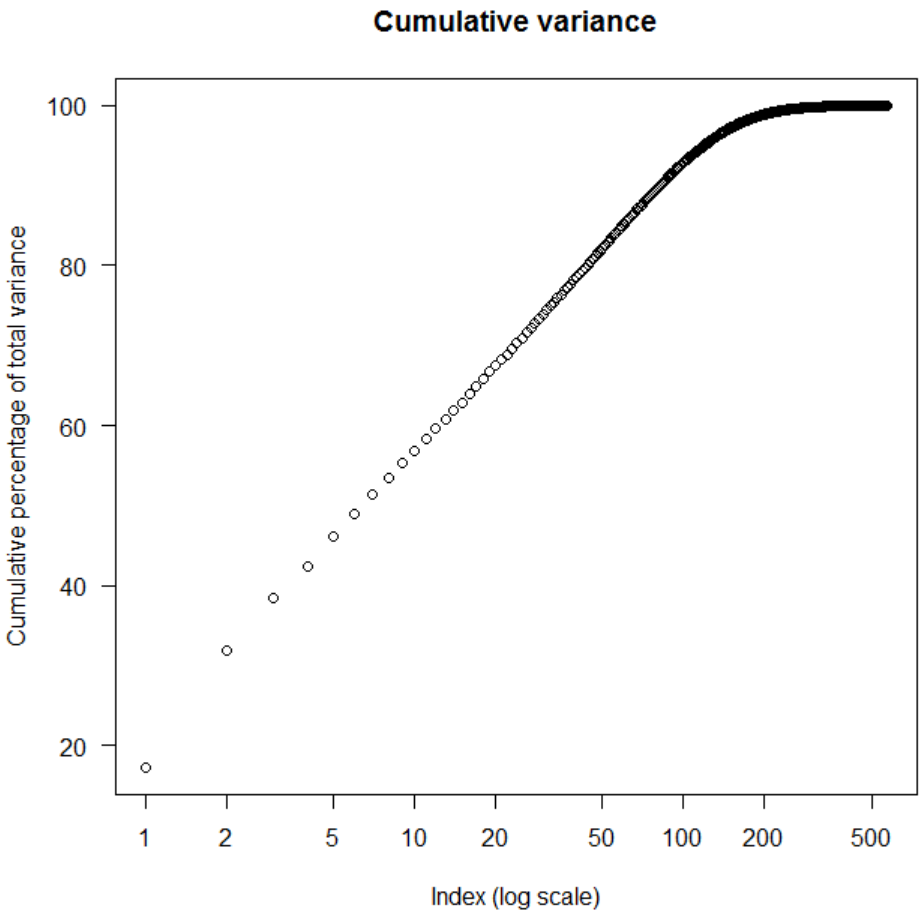

Figure 4: Plot of cumulative variance explained by PCs

We denote m as the number of PCs we use to reconstruct the compressed image, and $0<m\leq 574$. In order to display the compressed image, reconstruction is conducted using a series of m with different values. The image quality improved with the increase of m. When m is less or equal than 20, the images are terribly blurry. When m is between 25 and 70, the general outline of the picture can be seen and become clearer as m increases. When m is greater than 100, the resolution is relatively good and the difference between the compressed image and the original image is minor.

It is obvious that the compression efficiency drop dramatically with the increase of m. When m equals 5, the compression ratio is 73.1. When m increases to 20, the ratio drops to 18.4. When m reaches 100, the compression ratio becomes 3.7. Interestingly, when m equals 400, the compression ratio is as low as 0.9, less than 1. In this case, the space required to store the matrices e_M and Y_M is even greater than the space

required to store the original image matrix X. Thus for this data set and image, it makes no sense to use more than 400 PCs for image compression.

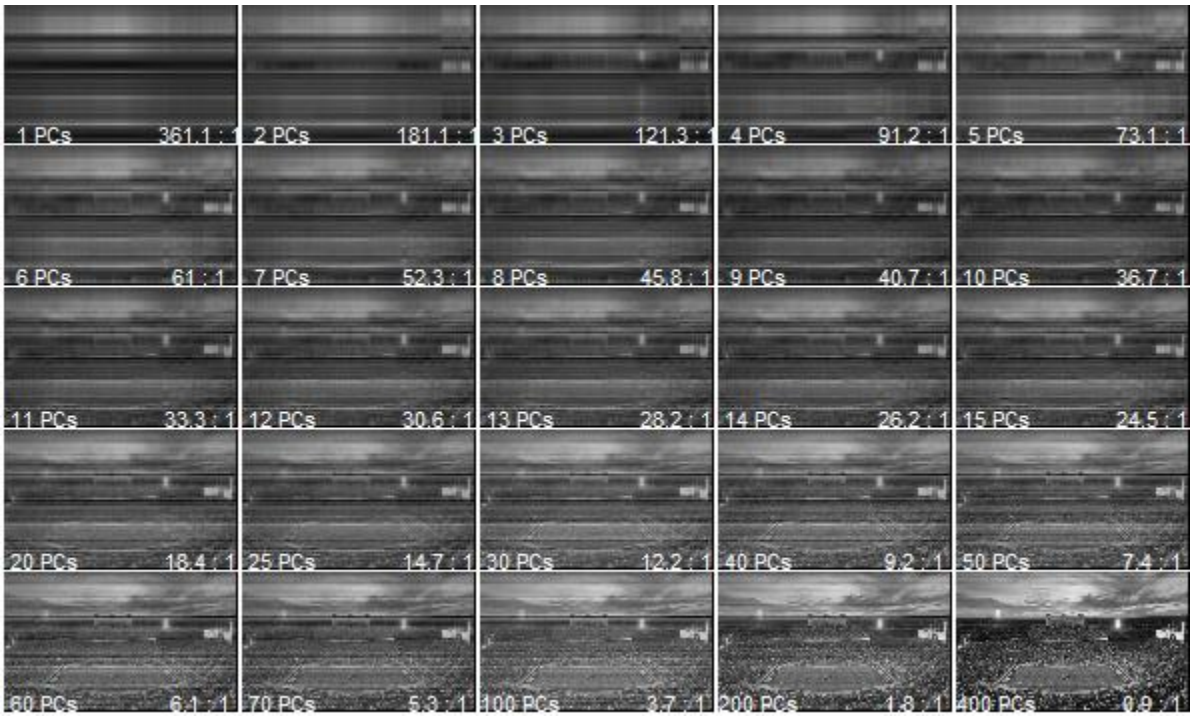

Figure 4: Reconstructions via different numbers of principal components.

Numbers at the bottom left is the value of PCs used for the reconstruction. Ratios at the bottom right are the compression ratio.

number of PCs m	compression ratio in value
1	361.1
2	181.1
3	121.3
4	91.2
5	73.1
6	61
7	52.3

8	45.8
9	40.7
10	36.7
11	33.3
12	30.6
13	28.2
14	26.2
15	24.5
20	18.4
25	14.7
30	12.2
40	9.2
50	7.4
60	6.1
70	5.3
100	3.7
200	1.8
400	0.9

Table 1: Compression ratio from different numbers of principal components.

Appropriate values of m namely the number of PCs used for image compression are usually determined by specific demands on image quality and compression ratio. And the optimal value of m is likely to vary from one image to another. Moreover, no established mathematic theory is found to identify appropriate values of m for image compression. Anyway, in our case, when m equals 100 or 200, the quality of the reconstructed image is quite good while the compression ratio of 3.7 or 1.8 is also acceptable. So roughly speaking, [100, 200] is possibly an appropriate range of m for this specific data set.

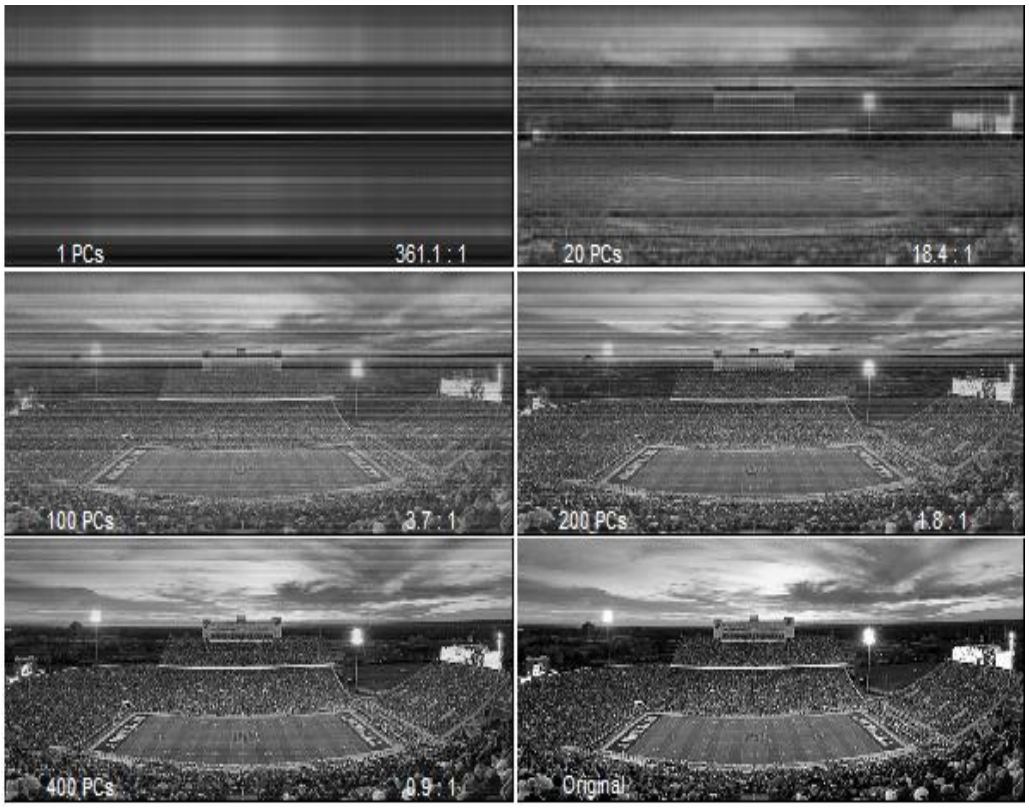

Figure 5: Reconstructions via different numbers of principal components.

Numbers at the bottom left is the value of PCs used for the reconstruction. Ratios at the bottom right are the compression ratio.

Root mean square error (RMSE) of the approximation $\widetilde{X}$ is calculated to provide a quantitative assessment of model fit. The error plot indicates that RMSE drops with the increase of the number of PCs used to reconstruct the image, which is in line with our expectation.

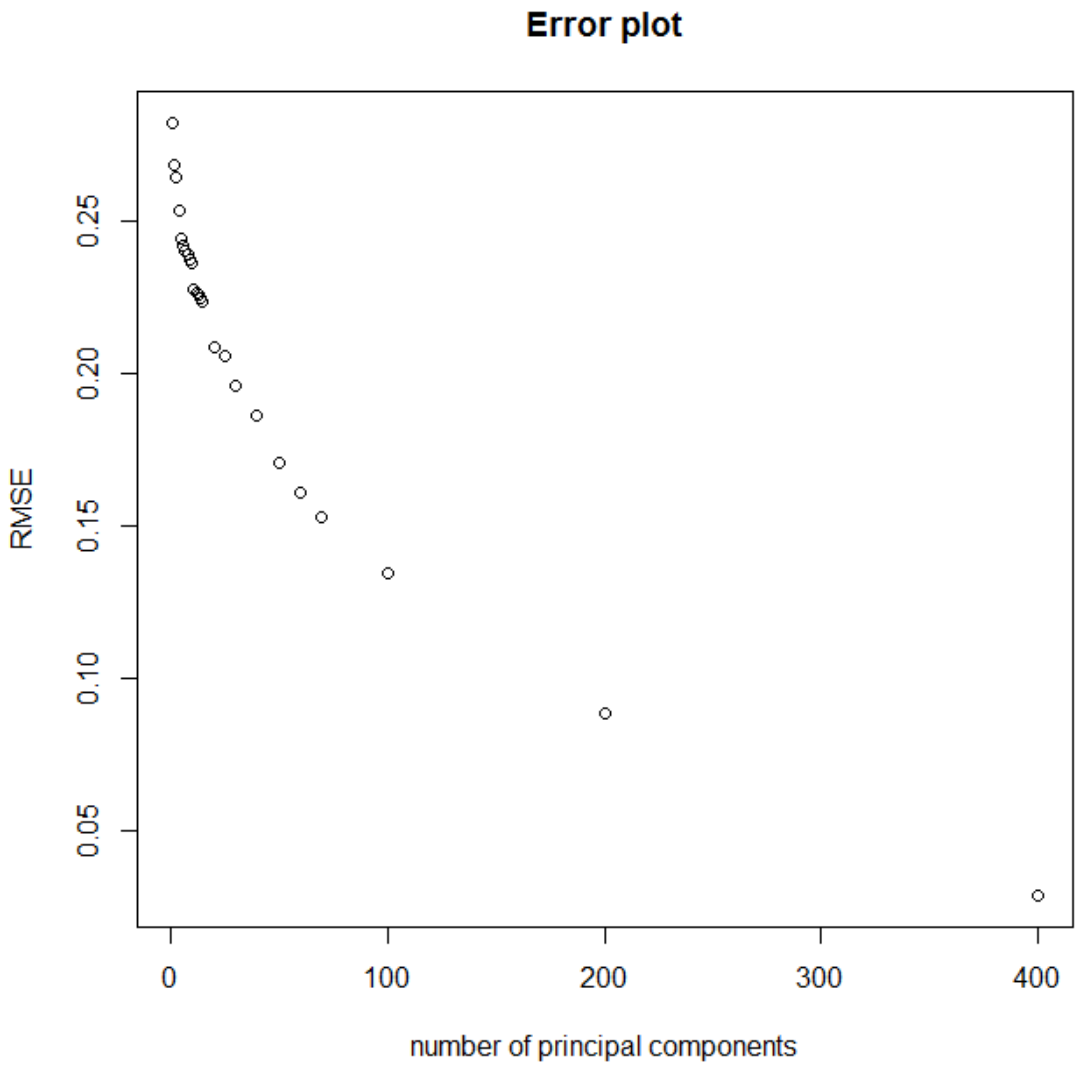

Figure 6: Error plot of RMSE.

And I also tested the same analysis on another image represented by a 620 by 434 matrix. The results are quite similar.

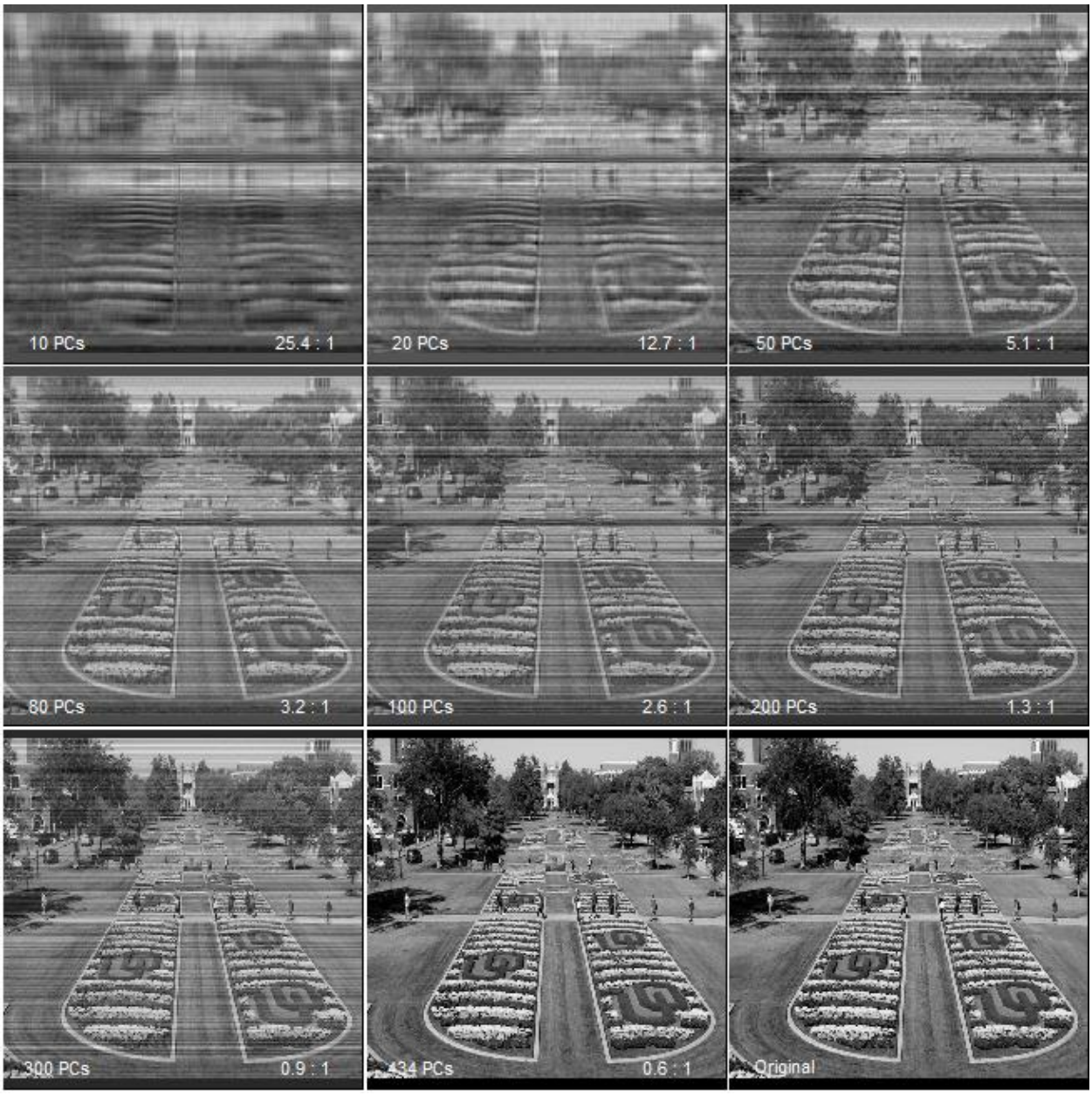

Figure 7: Reconstructions of another image via principal components.

Numbers at the bottom left is the value of PCs used for the reconstruction. Ratios at the bottom right are the compression ratio.

5. Discussion

In this study the performance of PCA on image compression was examined. Although we have the compression ratio as an indicator of compression efficiency, we do lack a quantitative assessment on the quality of the compressed image. Due to the fact that we are evaluating the resolution of reconstructed images subjectively, it is difficult to find the balance between image quality and compression efficiency. In order to find the optimal value of m for compression, objective and quantitative parameters as well as criteria should be introduced in further study.

In this study, all the variables are in the same scale, which enables me to run the PCA without standardization on the data set. Though as comparison, I also conducted the compression via PCA on the correlation matrix which is the covariance matrix of the standardized data. The PCs from covariance and correlation matrices are different. But their performance on compression are almost the same. However, there is one potential problem of PCA on the correlation matrix (and it is encountered when I conduct PCA on the correlation matrix of Picture 2). That is sometimes the pixel values in one variable, namely one column, are exactly the same, making its standard deviation equal zero. Consequently, it leads to NAs in the correlation matrix which is problematic for the subsequent calculation. One way to get around this problem is to check the correlation first before the analysis is conducted. Once NA is identified in the correlation matrix, transpose the data matrix and define the new columns as variables which are the rows of the original data set. Using this strategy, we can get around most of problems of this type. Yet it cannot solve the problem when there are standard deviations of zero in both columns and rows. Better solutions remain to be investigated.

6. Conclusion

PCA is a feasible method for image compression. But due to the limitation of the study, I failed to identify the optimal value of the number of PCs to use for compression.

Follow-up work

Image compression of RGB (3 channel) jpg image files into determined compression ratio via PCA.

Figure f1: Original RGB image

```
> JPEGcomp("Thunder.jpeg",6.2,"Thunder_compressed1.Rda","Thunder_compressed2.
Rda")    #compress image with the compression ratio of 6.2


> #check compression ratio
> check_ratio   #should be around 6.2
[1] 6.175492
```

Figure f2: Reconstructed RGB image