Numerical Computing Final Project

Image Encryption

Wenlin Gong
RIN:661855450
2018 Fall

December 7, 2018

1 Introduction

Nowadays, sending images via Internet is common and convenient in daily life, especially among university students. Yet, information security is an important subject and needs to be looked into. The goal of this project is to create an website (*http://gongw.pythonanywhere.com*) that can be used to do image encryption and decryption (no matter black and white or colorful images). Henon map algorithm, a chaos-based encryption, as well as bicubic imresize function, are used in this project. And it is good combination of safety from brute force attack and complexity.

2 Keywords

Image Encryption, Image Resize, Henon Map, MATLAB, Python, Flask

3 Process

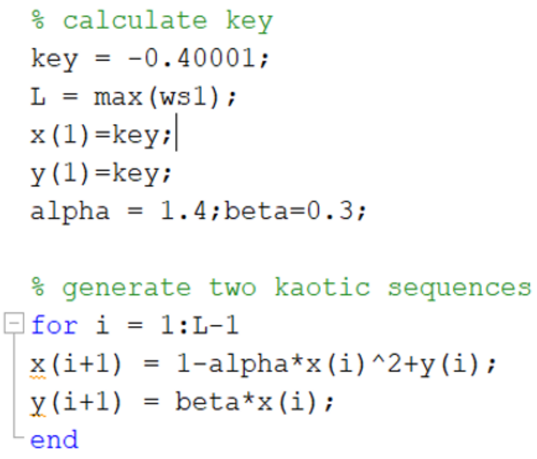
The website can be accessed at *http://gongw.pythonanywhere.com*.

1. Users are asked to browse computer and submit an image to the website
2. Users are asked to input the last four digits of their RIN, which serve as password and later be transformed into two initial values
3. Two images are needed for the encryption process. The image that user uploaded is to be encrypted, and another large landscape picture (set as default) is used to conceal the image
4. After a couple of seconds, the website outputs the encrypted image, which is available for users to download

Encryption process in details: Suppose image A is the image to be encrypted, image B is the image used for concealing encrypted A.

1. For Henon-map algorithm, $\alpha = 1.4$, $\beta = 0.3$, key$= -0.40001$. Here, α and β are set as default. In real cases, they are determined by user' last four digits of their RIN

2. Create two chaotic sequences where key is the starting point for both sequences. Sequences are generated using the formula showed in Figure 1



```
% calculate key
key = -0.40001;
L = max(ws1);
x(1)=key;
y(1)=key;
alpha = 1.4;beta=0.3;

% generate two kaotic sequences
for i = 1:L-1
x(i+1) = 1-alpha*x(i)^2+y(i);
y(i+1) = beta*x(i);
end
```

Figure 1: Henon Map Algorithm

3. Delete some extra elements to ensure that one sequence is the same length of image A width, another sequence is the same length of imgae A height

4. Sort these two chaotic sequences. Use the order generated to rearrange image A pixels in order to achieve encryption effect. Now image A is encrypted

5. The back-end Python code take the largest four digits of image A, which are the most significant digits in terms of visual effect

6. Take the largest four digits of image B and erase least four digits of image B

7. Place the largest four digits of image A into the lowest four digits of image B

8. So right now what human eyes see for image B is still image B, but actually image A is hiding beneath

Decryption process in details:

1. Take out the least four digits in image B
2. Shift these four digits into the largest four digits
3. Use α and β we are able to regenerate the two chaotic sequences. We are also able to sort these two sequences and then recover image A
4. Because we lost the least four digits of image A, there exists a information lost. To minimize the effect of information lost, use bicubic imresize function to improve visual effect of final decrypted image A (before A to be encrypted, resize A to twice larger; similarly when decrypt A, resize A to be 0.5 large).

4 Project Time frame

1. Literature research and determine the encryption method
2. Draft proposal and get approval for this project
3. MATLAB encryption code
4. Transform MATLAB code into Python code. Use Pillow package in Python to process image
5. Use HTML, CSS, SCSS to do front-end of website. Use Python Flask package to do back-end
6. Draft PowerPoint and prepare presentation
7. In-class Presentation
8. Move local website into a domain. I use pythonanywhere.com to achieve this goal. I also tried Google cloud platform yet failed.

5 Challenges

- Encryption Algorithm
  I looked up many ways to do image encryption and chose Henon-Map algorithm in the end. Henon Map is good combination of safety from brute force attack and complexity. Figure 2 are the frequency analysis of pixels in the original image and the encrypted image.
- Transform MATLAB into Python
  MATLAB is very effective in terms of image process as compared to Python. It takes me a while to know how image is represented in Python and the code grammar. The encryption process, which only takes up a few lines in MATLAB, requires Python almost 40 lines to achieve the same goal.

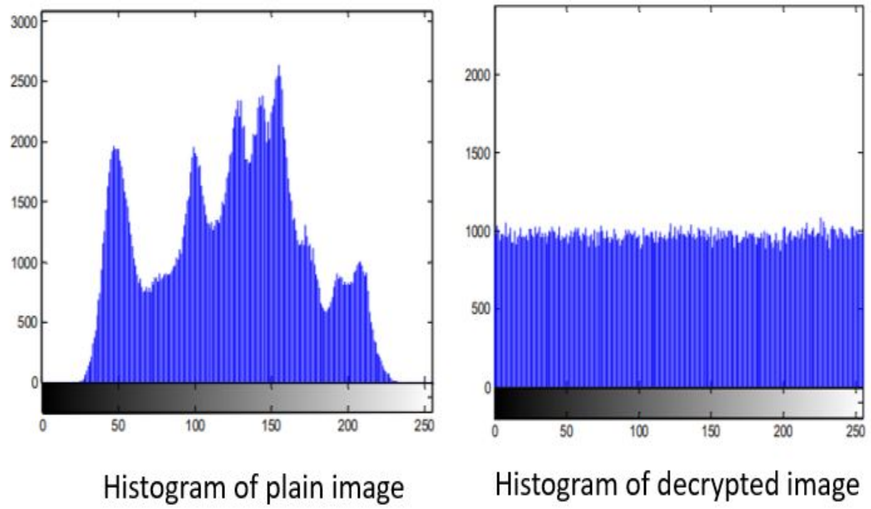


Figure 2: Frequency Analysis of pixels

- Minimize information lost
  How to represent the decrypted image as similar as to the ogrinal image is a challenging problem in this project. While bicubic resize method is very good at representing different layers of colors, it is highly computational and a bit slow in run time.

6 Conclusion and Discussion

This project is a small example of image encryption and numerical computing's application in daily life. Students who interested could make use of this website to encrypt their numerical computing homework. It is a interesting project and I want to improve the website to have more features in the future. For example, right now the website could only be used to do encryption. I want implement a decryption site at the same time. Also, I realize bicubic resize is not the ideal interpolation method due to this high computation cost. Maybe a better interpolation method should be created to fit in this context.

7 Examples

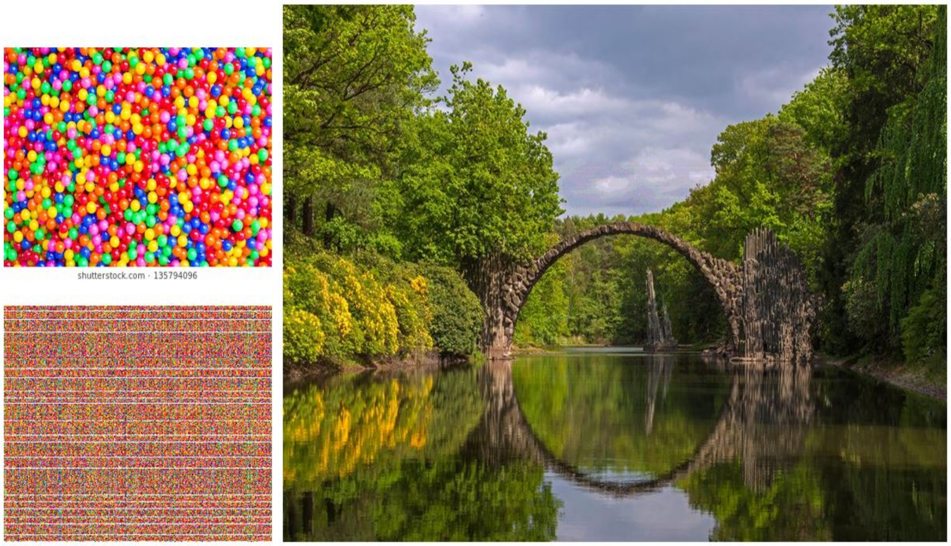
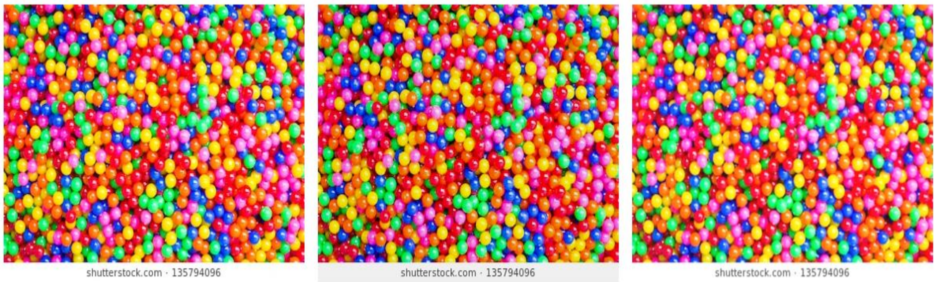


Figure 3: Example 1


shutterstock.com · 135794096
shutterstock.com · 135794096
shutterstock.com · 135794096

Original

Decrypted-no interpolation

Interpolation

Figure 4: Example 1 - Outcome

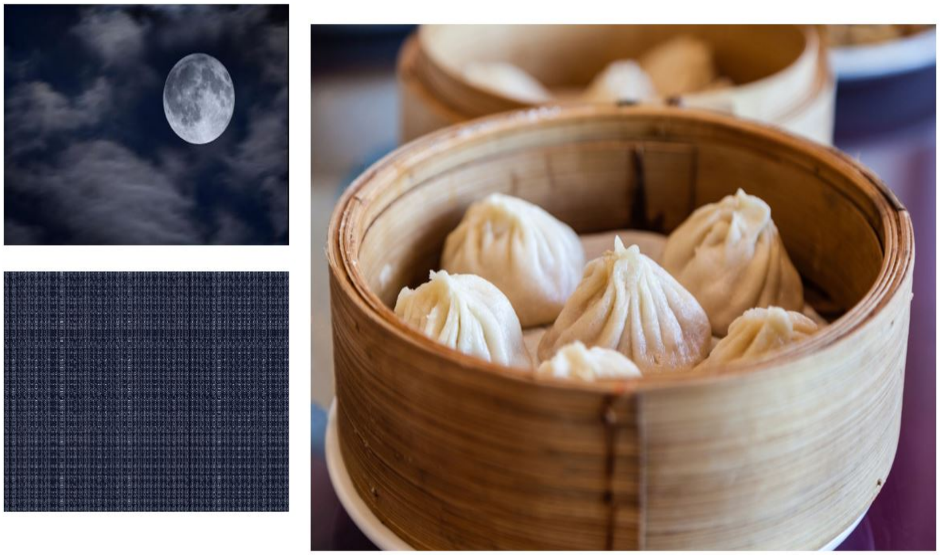
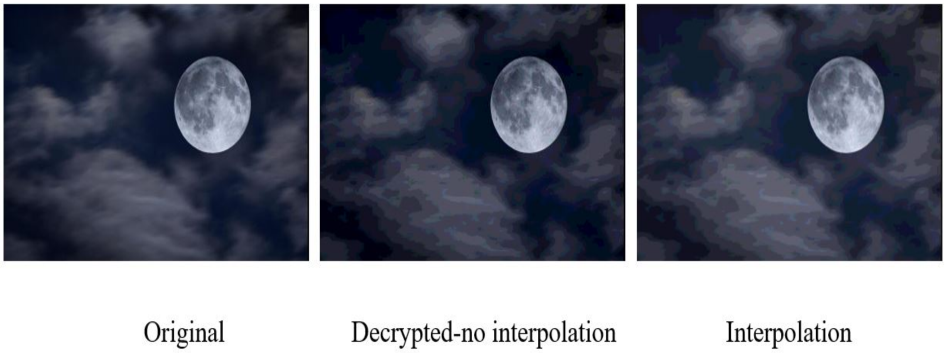


Figure 5: Example 2

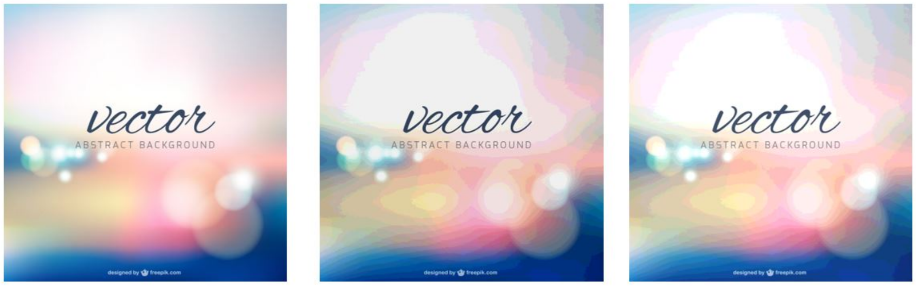


Original

Decrypted-no interpolation

Interpolation

Figure 6: Example 2 - Outcome



Original Decrypted-no interpolation Interpolation

Figure 7: Example 3: Information Lost