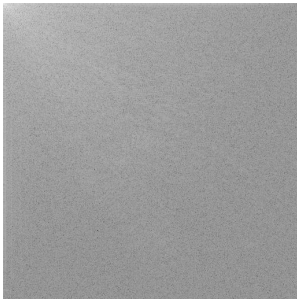


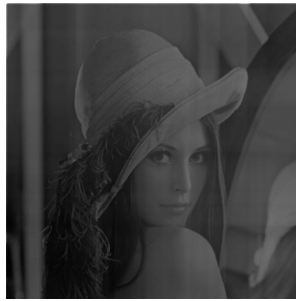
## Homework 3: Image Processing

*Instructor: Siheng Chen***Zhou Haoquan****Q1. Image Compression****P1.  $1/4$  Case**

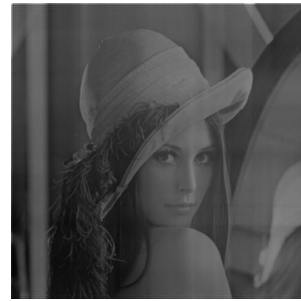
The frequency domain image is shown in the following figure. We can see that there is no significant



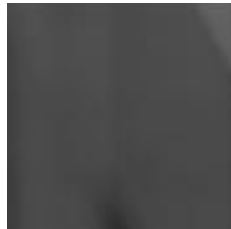
(a) frequency domain



(b) recovered

(c) compressed recovered with  $1/4$  scale

pattern in the frequency domain picture. And we see lots of lines in the compressed picture. The difference can be easily found in following figures.



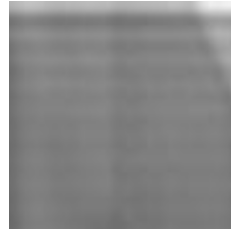
(d) recovered detail

(e) compressed detail with  $1/4$  scale**P2  $1/16$  Case**

We find that the lines in the  $1/16$  scale recovered picture is much more thicker than those in  $1/4$  scale recovered picture.



(f) compressed recovered with 1/16



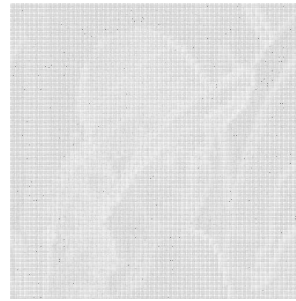
(g) compressed detail with 1/16 scale

### P3. Blockwise Compression

The blockwise compression result is shown in the following pictures Here we see that there are clear



(h) compressed recovered with blockwise compression



(i) frequency domain of blockwise compression

meshes between blocks. On the other hand, we find that the frequency picture shows the high frequency component more clearly than global compression picture. It contains more information.

### Q3. Edge Detection

**Write-up:** As shown in the following figures, edge detection shows a good result. Almost all the edges have been detected. Note that I have met some issues of being divided by 0 when calculating the angle of gradient  $\theta = \arctan \frac{G_y}{G_x}$ . The solution is adding a small noise to  $G_x = 0$ .



(j) original figure



(k) gradient



(l) nms



(m) edge

## Q4. Line Detection

We first do Hough Transformation to **img01.jpg** and **img02.jpg**, the result is shown in the following figures. Note that the response is skewed to the very left of the figures and thus may not be easily found. We show the Line detection results of both **img01.jpg** and **img02.jpg** in the following

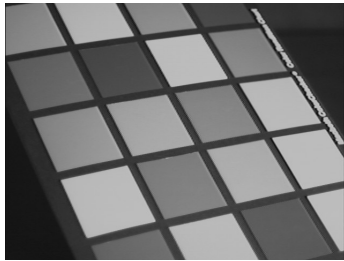


(n) Hough Transformation of img01.jpg

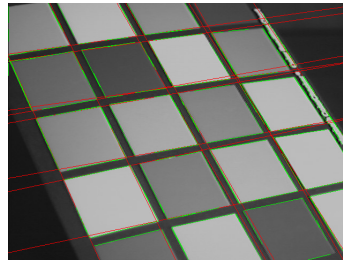


(o) Hough Transformation of img02.jpg

figures. From the comparison between red lines (my results) and green lines (opencv results), we can say that the results are fundamentally correct.



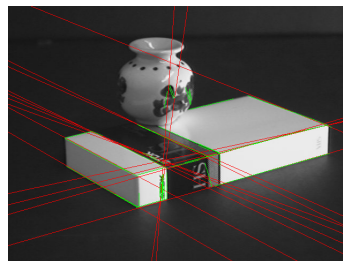
(p) Original **img01.jpg**



(q) Line Detection result of **img01.jpg**



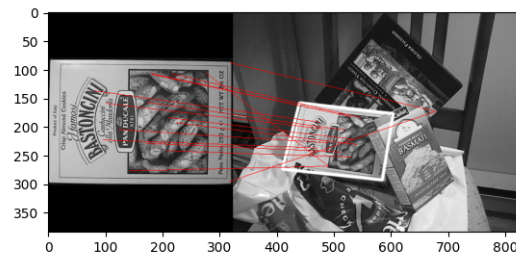
(r) Original **img02.jpg**



(s) Line Detection result of **img02.jpg**

## Q5. Feature Extraction

The result of SIFT is shown in the following figure. We can see that the features are matched in a generally correct way. But there are some mis-matches, which need to be improved.



(t) Result of SIFT