Sparse Image Coding

_ECE367 Problem Set 6 -- Aman Bhargava -- 1005189733__

Using l_1 regularization for image compression.

Problem Background and Formulation

- [x] Read Section 9.6.2 in OptM.
- [x] Read Example 9.19 in OptM.

- $M \in \{0, \dots, 255\}^{n \times n}$ is a matrix representing grayscale images.
- Orthonormal $H \in \mathbb{R}^{n \times n}$ is used for 2D wavelet transform.
- Analogy: $\tilde{y} = A^T y$ -- Wavelet transform coefficients for M are:

$$\tilde{M} = HMH^T$$

- Where $\tilde{M}$ is an $n \times n$ matrix.
  - Values are 'concentrated' -- entries are very small magnitude for the most part, relatively few large coefficients.
- H encodes a bunch of orthogonal Haar Wavelets.

Wavelet Encoding with l_1 Regularization:

$$f(\lambda) = \min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} ||H^T X H - M||_F^2 + \lambda ||X||_1$$

- Optimal X^* leads to optimal approximation of M which is $\hat{M} = H^T X^* H$.
- $||X||_1$ is just the l_1 norm of the vectorized version of X.
- $||\cdot||_F^2$ is just l_2 norm squared of vectorized $\cdot$.

Part A

Derive expression for inverse wavelet transform. Use H to express M in terms of $\tilde{M}$. Why is it important to use specifically designed transform matrix H rather than random orthonormal matrix?

Starting equation: $\tilde{M} = HMH^T$

$$H^T \tilde{M} = MH^T$$

$$H^T \tilde{M} H = M$$

$$\therefore M = H^T \tilde{M} H$$

It is important to use a specifically designed transform matrix H because H is composed of orthonormal discrete Haar wavelet vectors. When images are projected onto the vectors, they tend to be already somewhat sparse. It turns out that images and signals in general can be well-approximated by relatively few (i.e. a sparse set of) wavelets. A randomly selected orthonormal matrix would not necessarily yield an accurate sparse representation.

Part B

Show that $f(\lambda)$ is a separable problem. I.e., it reduces to a set of single-variable problems. Example of separable in OptM Ex. 9.19. Write down the solution to $f(\lambda)$.

Useful Matrix Properties

- $||V||_F^2 = \text{trace}(VV^T)$
- $\text{trace}(ABC) = \text{trace}(CAB) = \text{trace}(BCA)$

$$f(\lambda) = \min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} ||H^T X H - M||_F^2 + \lambda ||X||_1$$

$$f(\lambda) = \min_{X_{ij} \in \mathbb{R}} X_{ij}^2 - [H^T X H]_{ij} M_{ij} - M_{ij} [H^T X H]_{ij} + M_{ij}^2 + \lambda |X_{ij}|$$

Solution:

$$X_{ij}^* = \begin{cases} 0 & \text{if } |\tilde{M}_{ij}| \leq \lambda \\ \tilde{M}_{ij} - \lambda \text{sgn}(\tilde{M}_{ij}) & \text{, otherwise} \end{cases}$$

Part C

- [x] Produce histogram of $M, \tilde{M}$ values.
- [x] Produce histogram of $\tilde{M}$ zoomed into the origin.
- [x] Indicate the number of non-zero coefficients in M, non-zero coefficients in $\tilde{M}$.

From `Part C -- Code & Figures` :

```
There are 65536 non-zero coefficients in M
There are 65325 non-zero coefficients in M̃
```

Histograms can be found below.

Part D

- [x] Compute optimal X^* for $\lambda = 30$.
- [x] Compute compression factor (ratio of non-zero components of X^* to size of M).
- [x] Compute inverse wavelet transform to get $\hat{M}$ from X^*.

- [x] Histogram of non-zero components of X^*.
- [x] Close-up histogram of non-zero components of X^* around origin.

Comment on Zoomed Histogram: There are far fewer non-zero values very close to zero. This indicates that the 'unimportant' coefficients are being thrown out by the l_1 regularization.

Part E

- [x] Re-do part D with $\lambda = 10$, $\lambda = 90$.

Differences in $\hat{M}$:

- $\lambda = 10$: $\hat{M}$ is even closer to the original image than before in terms of MSE (decreased from ~195 to 48.34)
- $\lambda = 90$: $\hat{M}$ is quite different from the original image in terms of MSE (increase from ~195 to ~494).

Differences in histograms:

- $\lambda = 10$: The histogram for non-zero X^* has more values close to zero than for $\lambda = 30$ but still fewer values than the non-regularized encoding. This indicates that the image is being less compressed (fewer discarded values).
- $\lambda = 90$: THe histogram for non-zero X^* has very few values close to zero compared to the $\lambda = 30$ case. This indicates that many low values are being discarded (i.e. set to zero), resulting in substantial compression.

Differences in images:

- $\lambda = 10$ is very close to the original image and retains more details than the $\lambda = 30$ case.
- $\lambda = 90$ is extremely distorted and retains very few details compared to the $\lambda = 30$ case.

In [155…

```
# IMPORT BOX #
using Images, Plots
using ImageShow
using LinearAlgebra
using MAT
```

In [156…

```
# DATA IMPORT BOX #
vars = matread("sparseCoding.mat")
M = vars["M"];
H = vars["H"];
```

Part C -- Code & Figures

In [157…

```
# showing the image
```

```
println("Original Image");
Gray.(M/255)
```

Out[157…

Original Image

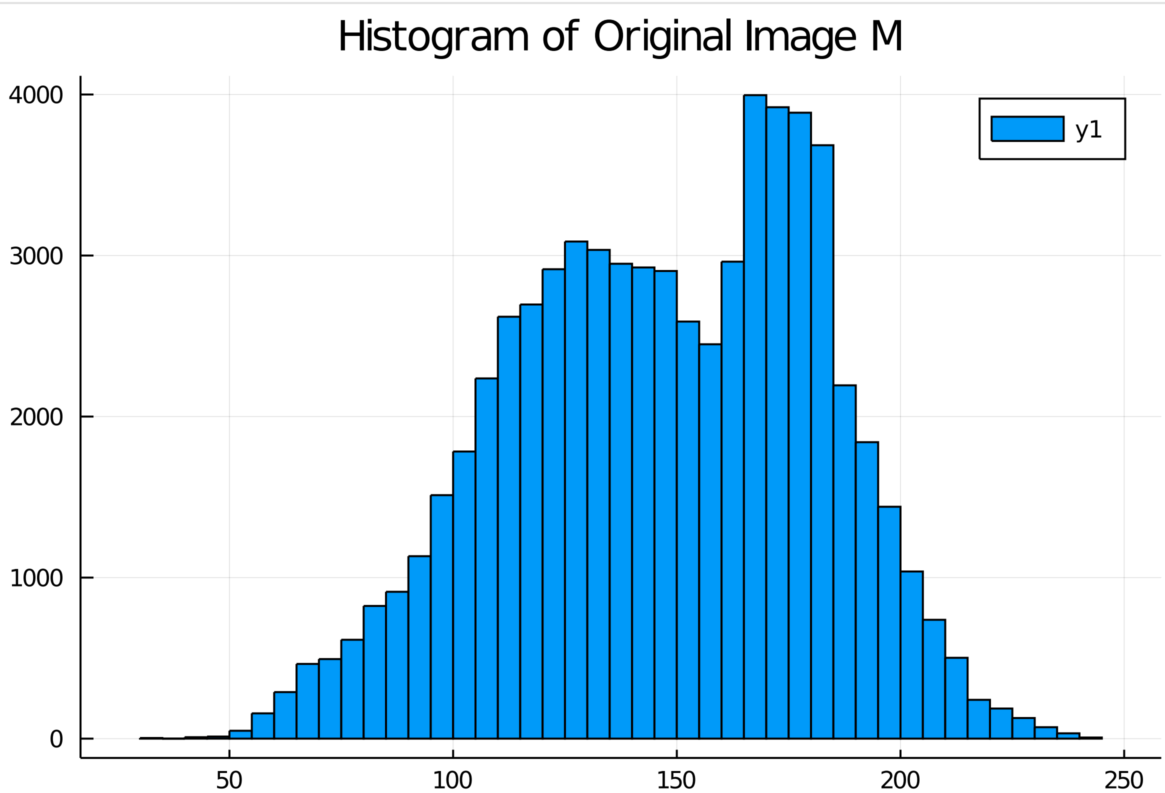
In [158…

```
M̃ = H*M*transpose(H); # Solving for DWT of image.
```

In [159…

```
# Plotting histogram of M:
hist = histogram(vec(M));
plot(hist)
title!("Histogram of Original Image M")
```

Out[159…

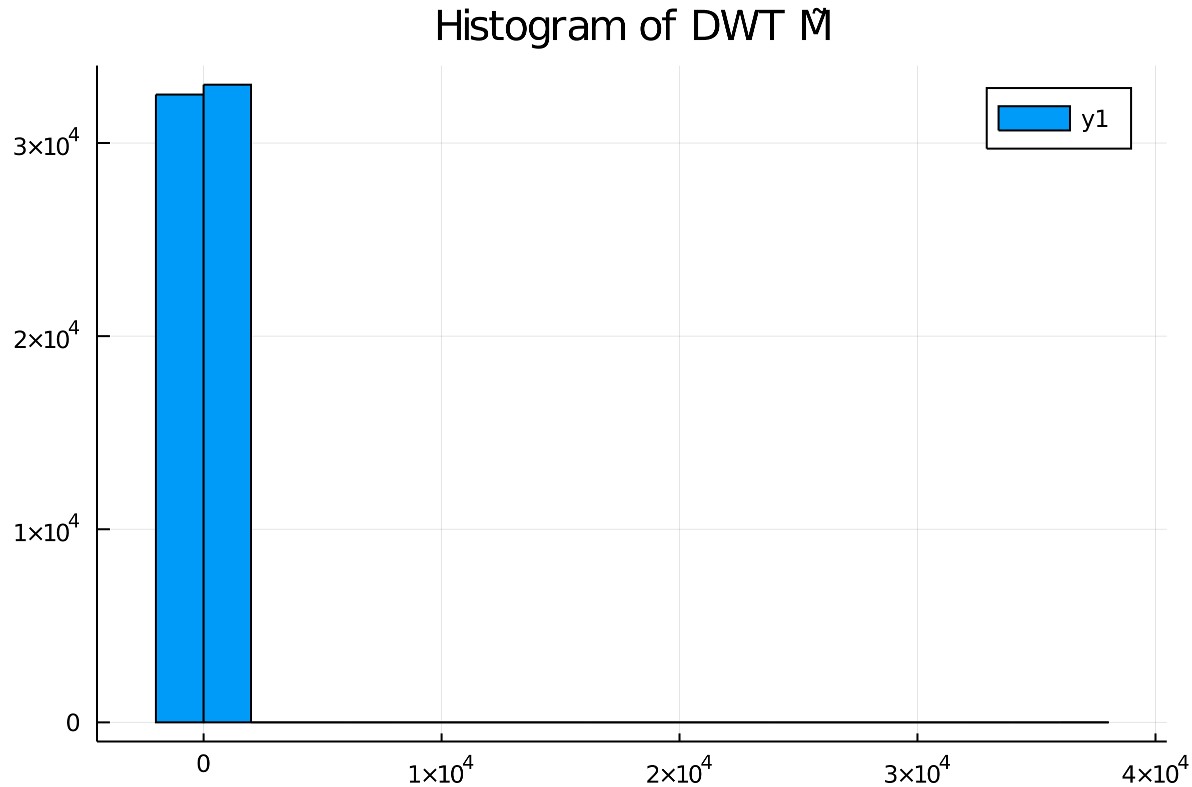


In [160…

```
# Plotting histogram of M̃
hist = histogram(vec(M̃), nbins=20);
```

```
plot(hist)
title!("Histogram of DWT M̃")
```

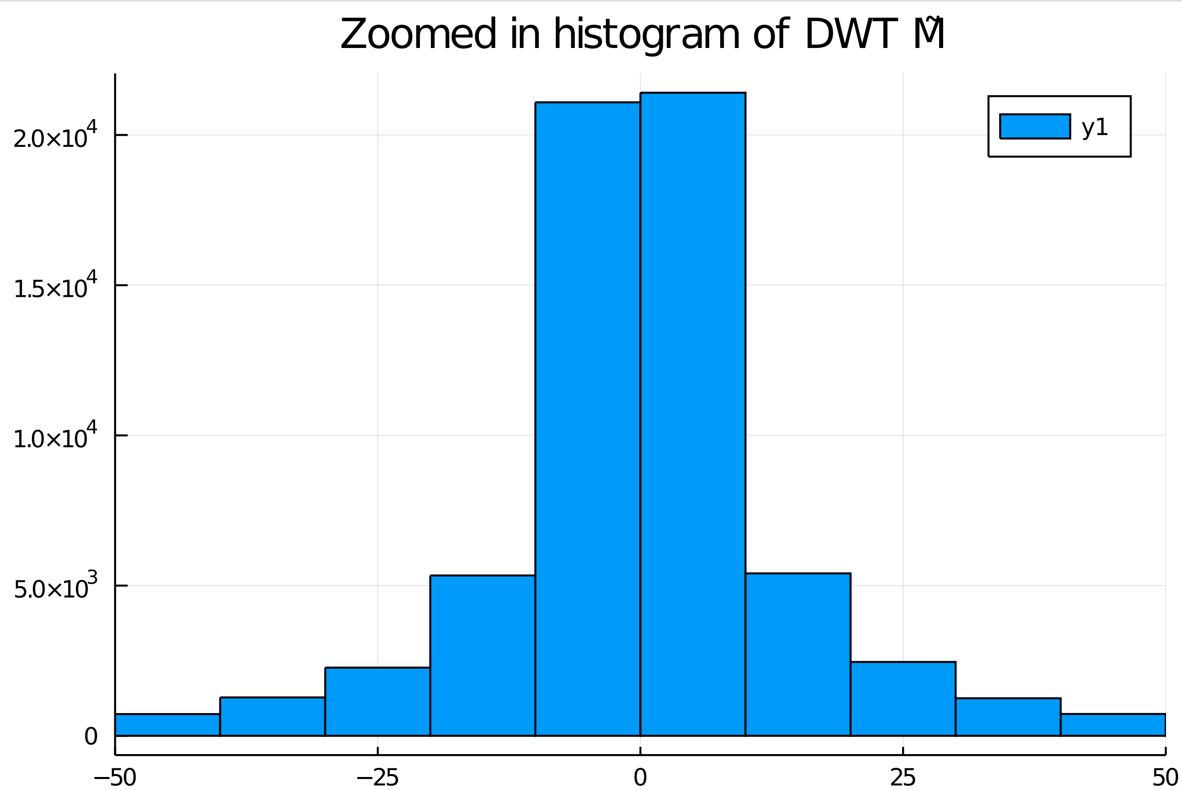
Out[160…



In [161…

```
# Zoomed in histogram of M̃
hist = histogram(vec(M̃), nbins = 5000);
plot(hist, xlims=(-50,50))
title!("Zoomed in histogram of DWT M̃")
```

Out[161…



In [162…

```
# Determining number of non-zero coefficients in M, M̃:
M_non_zero = 256*256 - sum(M .== 0)
println("There are $M_non_zero non-zero coefficients in M");

M̃_non_zero = 256*256 - sum(M̃ .== 0)
println("There are $M̃_non_zero non-zero coefficients in M̃");
```

```
There are 65536 non-zero coefficients in M
There are 65325 non-zero coefficients in M̃
```

Part D -- Code & Figures

In [163…

```
λ = 30;
X_star = zeros(256, 256);

# Computing optimal X_star for λ=30
for i = 1:256
    for j = 1:256
        if M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]-λ
        elseif -1*M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]+λ
        end
    end
end
```

In [164…

```
# Computing compresion factor
X_non_zero = 256*256 - sum(X_star .== 0)
comp_ratio = X_non_zero/(256*256)
println("Compression ratio for X_star is $comp_ratio");

# Computing MSE:
# Inverse wavelet transform
M̂ = transpose(H)*X_star*H;
MSE = 1/(256^2)*norm(M̂-M)^2;
println("MSE for compressed image is $MSE");
```

```
Compression ratio for X_star is 0.1153411865234375
MSE for compressed image is 195.3581859869883
```

In [165…

```
# Inverse wavelet transform
M̂ = transpose(H)*X_star*H;
println("Compressed vs. original image for λ=30");
mosaicview(RGB.(M̂/255), RGB.(M/255); nrow=1)
```

```
Compressed vs. original image for λ=30
```

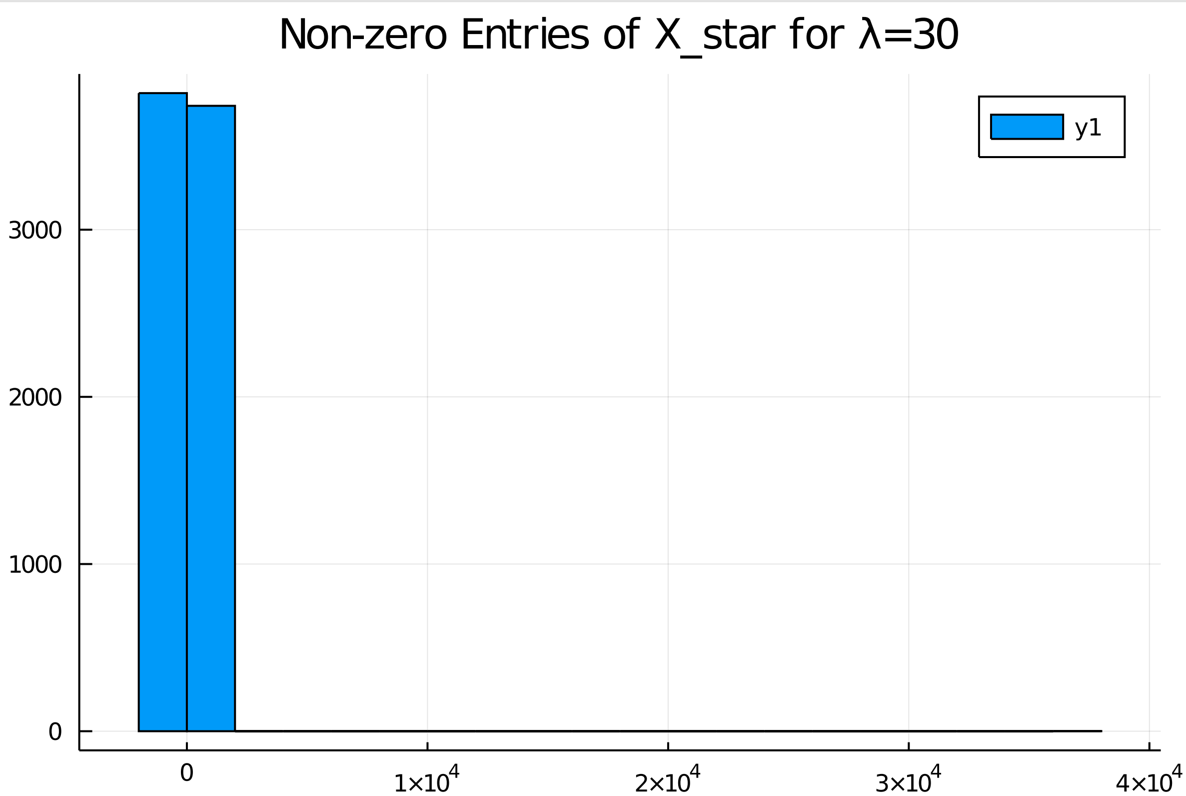
Out[165…

In [166…

```
# Computing histogram for non-zero entries
X_nz = [];
for i = 1:256
    for j = 1:256
        if X_star[i,j] != 0
            append!(X_nz, X_star[i,j])
        end
    end
end

# Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 20);
plot(hist)
title!("Non-zero Entries of X_star for λ=30")
```

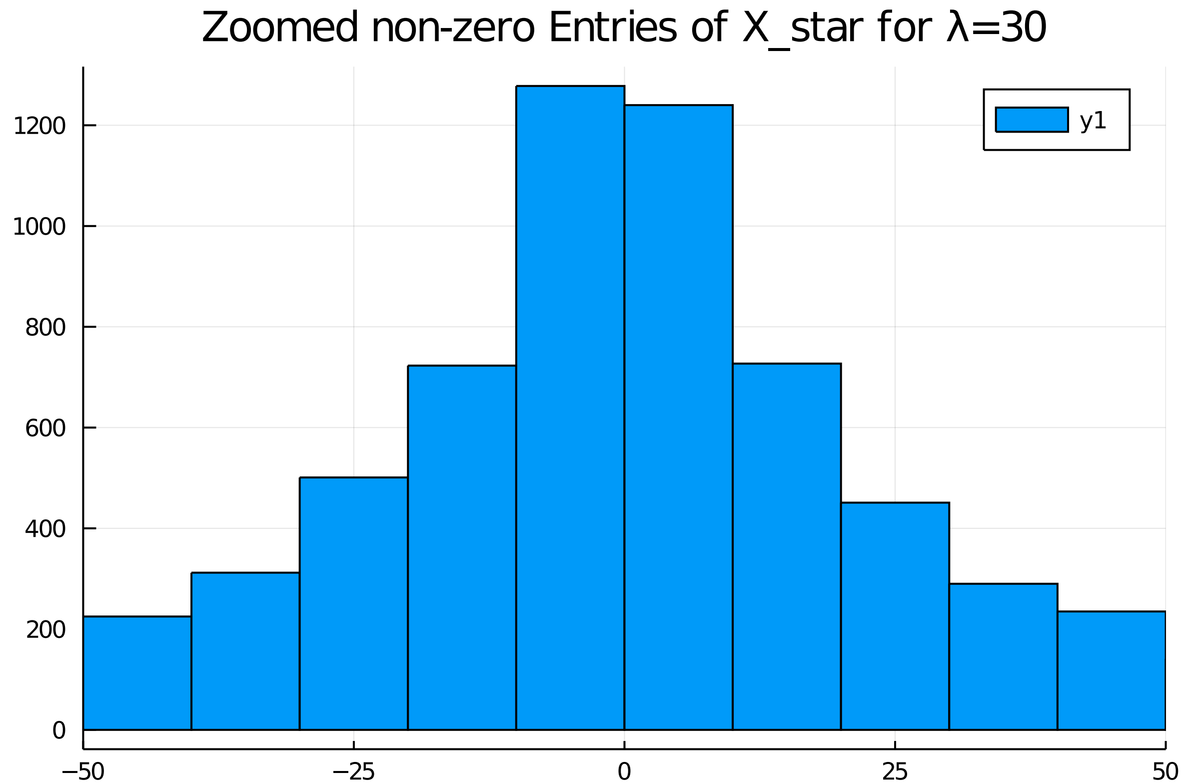
Out[166…



In [167…

```
# Zoomed Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 5000);
plot(hist, xlims=(-50,50))
title!("Zoomed non-zero Entries of X_star for λ=30")
```

Out[167…



Part E -- Code & Figures

$$\lambda = 10$$

In [168…

```
λ = 10;
X_star = zeros(256, 256);

for i = 1:256
    for j = 1:256
        if M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]-λ
        elseif -1*M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]+λ
        end
    end
end
```

In [169…

```
# Computing compresion factor
X_non_zero = 256*256 - sum(X_star .== 0)
comp_ratio = X_non_zero/(256*256)
println("Compression ratio for X_star is $comp_ratio");


# Computing MSE:
```

```
# Inverse wavelet transform
M̂ = transpose(H)*X_star*H;
MSE = 1/(256^2)*norm(M̂-M)^2;
println("MSE for compressed image is $MSE");
```

```
Compression ratio for X_star is 0.3509979248046875
MSE for compressed image is 48.34250795468691
```

In [170…

```
println("Compressed vs. original image");
mosaicview(RGB.(M̂/255), RGB.(M/255); nrow=1)
```

Compressed vs. original image

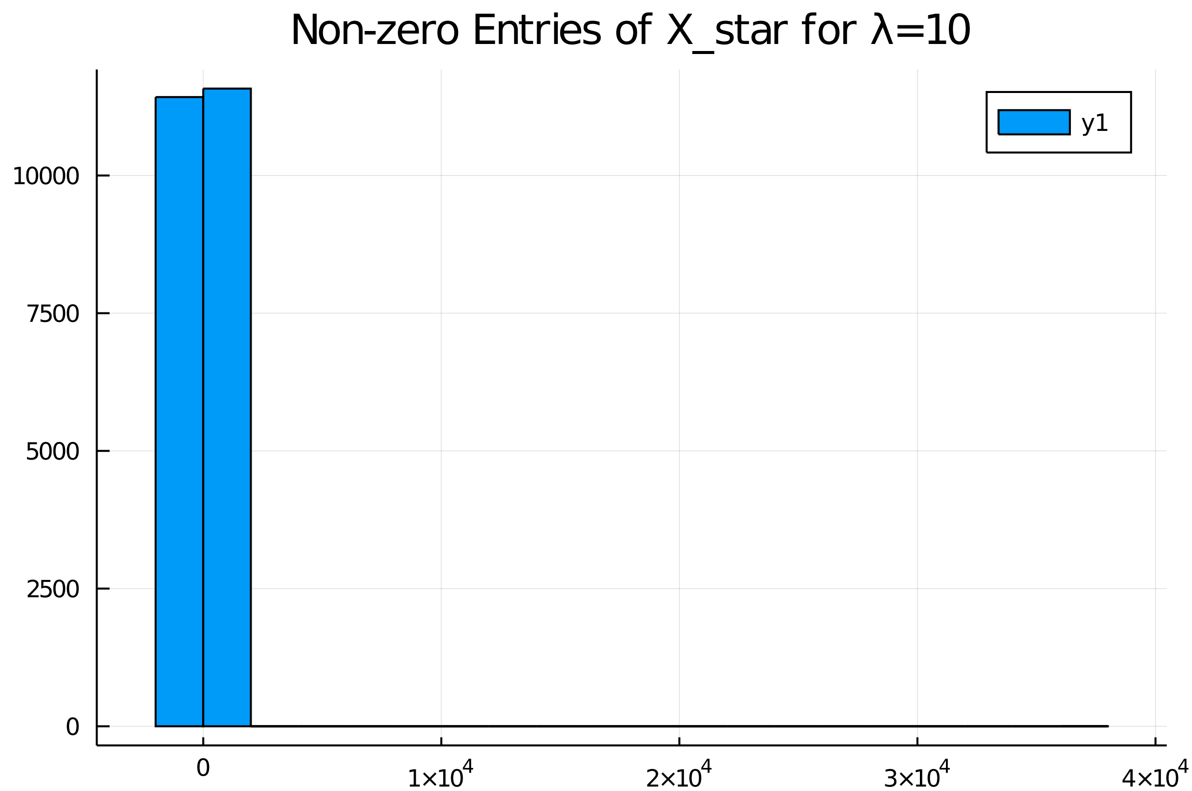
Out[170…

In [171…

```
# Computing histogram for non-zero entries
X_nz = [];
for i = 1:256
    for j = 1:256
        if X_star[i,j] != 0
            append!(X_nz, X_star[i,j])
        end
    end
end

# Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 20);
plot(hist)
title!("Non-zero Entries of X_star for λ=10")
```

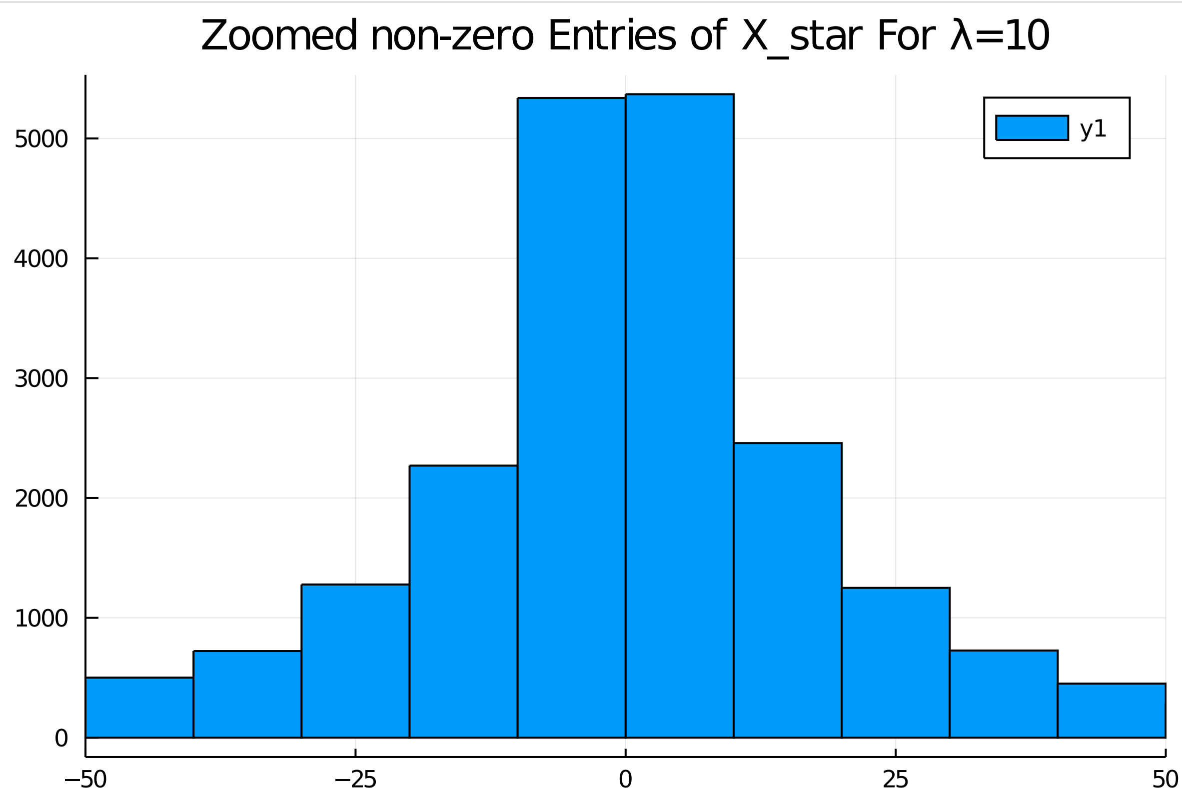
Out[171…



In [172…

```
# Zoomed Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 5000);
plot(hist, xlims=(-50,50))
title!("Zoomed non-zero Entries of X_star For λ=10")
```

Out[172…



$\lambda = 90$

In [173…

```
λ = 90;
X_star = zeros(256, 256);

for i = 1:256
    for j = 1:256
        if M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]-λ
        elseif -1*M̃[i,j] >= λ
            X_star[i,j] = M̃[i,j]+λ
        end
    end
end
```

In [174…

```
# Computing compresion factor
X_non_zero = 256*256 - sum(X_star .== 0)
comp_ratio = X_non_zero/(256*256)
println("Compression ratio for X_star is $comp_ratio");
# Computing MSE:
# Inverse wavelet transform
M̂ = transpose(H)*X_star*H;
MSE = 1/(256^2)*norm(M̂-M)^2;
println("MSE for compressed image is $MSE");
```

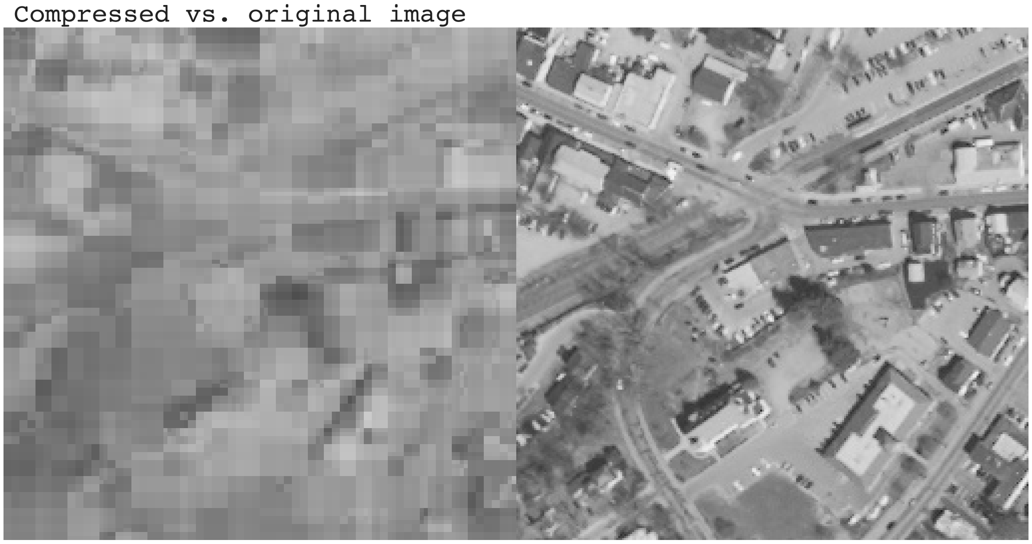
```
Compression ratio for X_star is 0.01959228515625
MSE for compressed image is 494.0171799985692
```

In [175…

```
println("Compressed vs. original image");
mosaicview(RGB.(M̂/255), RGB.(M/255); nrow=1)
```

```
Compressed vs. original image
```

Out[175…

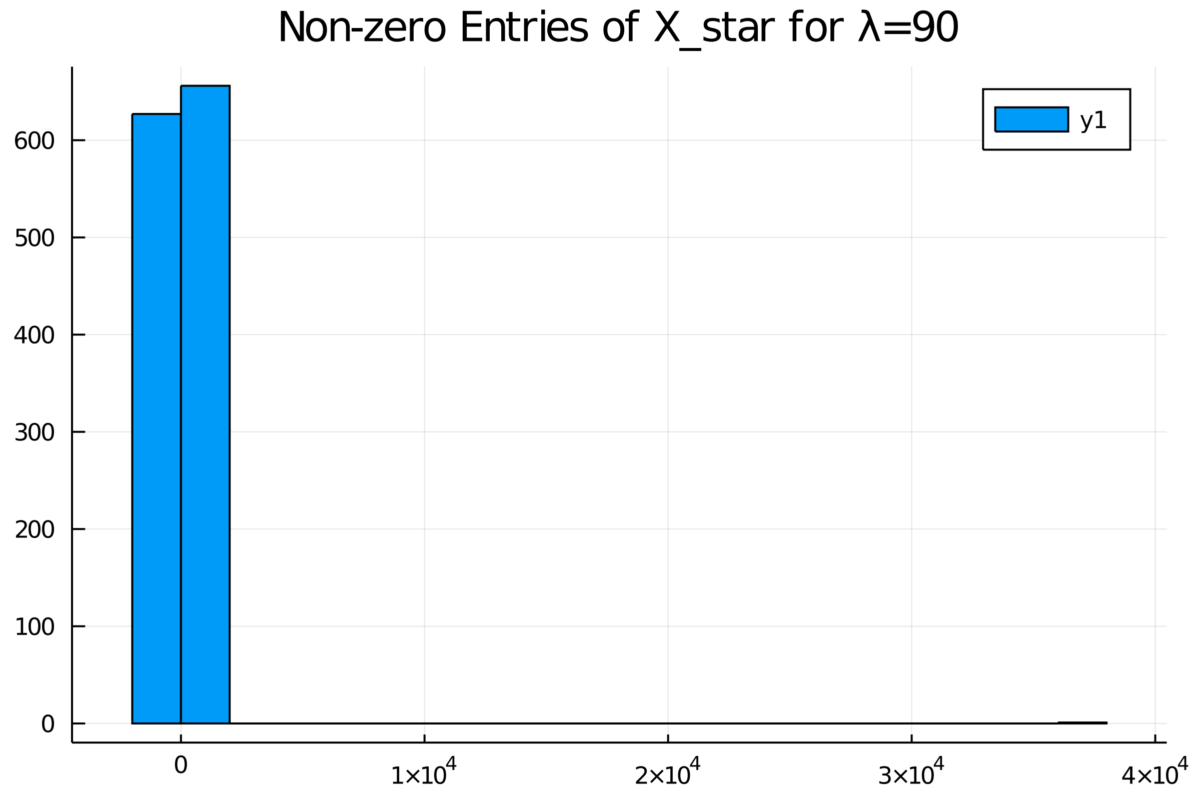


In [176…

```
# Computing histogram for non-zero entries
X_nz = [];
for i = 1:256
    for j = 1:256
        if X_star[i,j] != 0
            append!(X_nz, X_star[i,j])
        end
    end
end
```

```
# Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 20);
plot(hist)
title!("Non-zero Entries of X_star for λ=90")
```

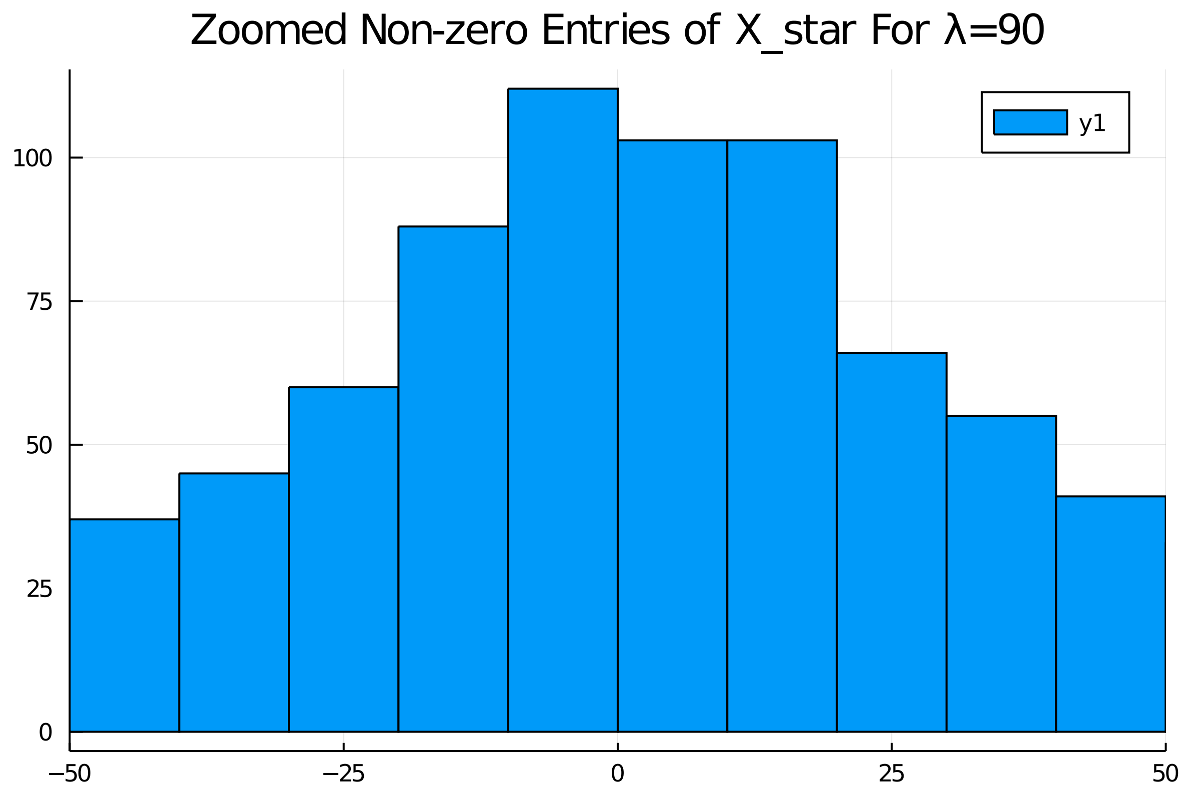
Out[176…



In [177…

```
# Zoomed Histogram of non-zero entries of X_star
hist = histogram(X_nz, nbins = 5000);
plot(hist, xlims=(-50,50))
title!("Zoomed Non-zero Entries of X_star For λ=90")
```

Out[177…



Zoomed Non-zero Entries of X_star For λ=90
y1
100
75
50
25
0
−50
−25
0
25
50

In [ ]: