Table of Contents

```
%{
Yonatan Carver
ECES 352 - Lab 7

%}
clear; clc; close all
```

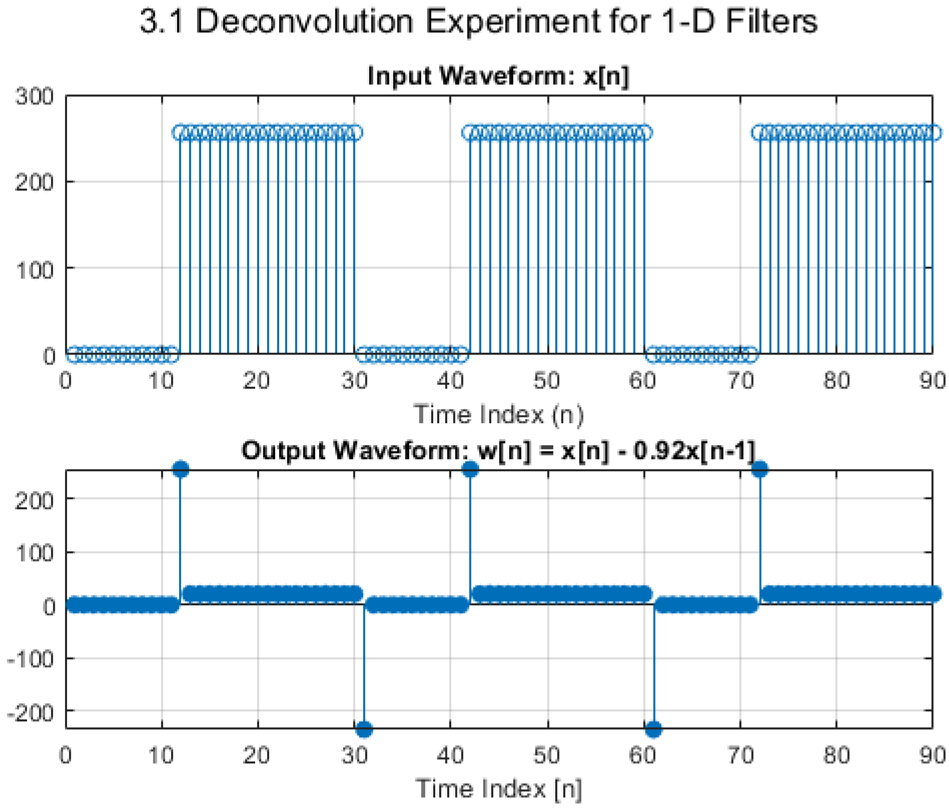
```
load lab6dat % as x1, xtv, x2, h1, h2
```

3 Lab: FIR Filters

3.1 Deconvolution Experiment for 1-D Filters

```
xx = 256 * (rem(0:100, 30) > 10); % x[n]
bb = [1 -0.92];  % impulse response

ww = firfilt(bb, xx);

first = 1;
last = length(xx);
nn = first:last;

figure('Name', '3.1 - Deconvolution Experiment for 1-D Filters')
subplot(2,1,1)
% stem(nn-1, xx(nn))
stem(xx)
grid on
xlim([0 90])
title('Input Waveform: x[n]')
xlabel('Time Index (n)')

subplot(2,1,2)
% stem(nn-1, w(nn), 'filled')
stem(ww, 'filled')
grid on
xlim([0 90])
title('Output Waveform: w[n] = x[n] - 0.92x[n-1]')
```

```
xlabel('Time Index [n]')

sgtitle('3.1 Deconvolution Experiment for 1-D Filters')

%{
The length of x[n] is 101.
The length of w[n] is 102.
If h = 2 elements (n+2-1)
%}
```

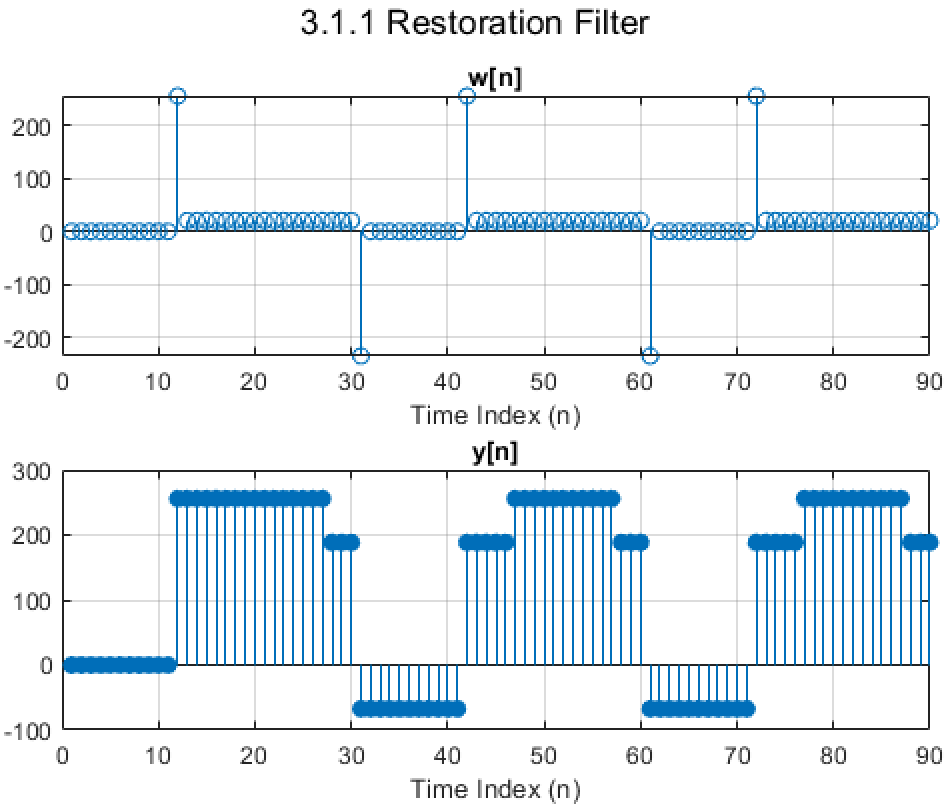


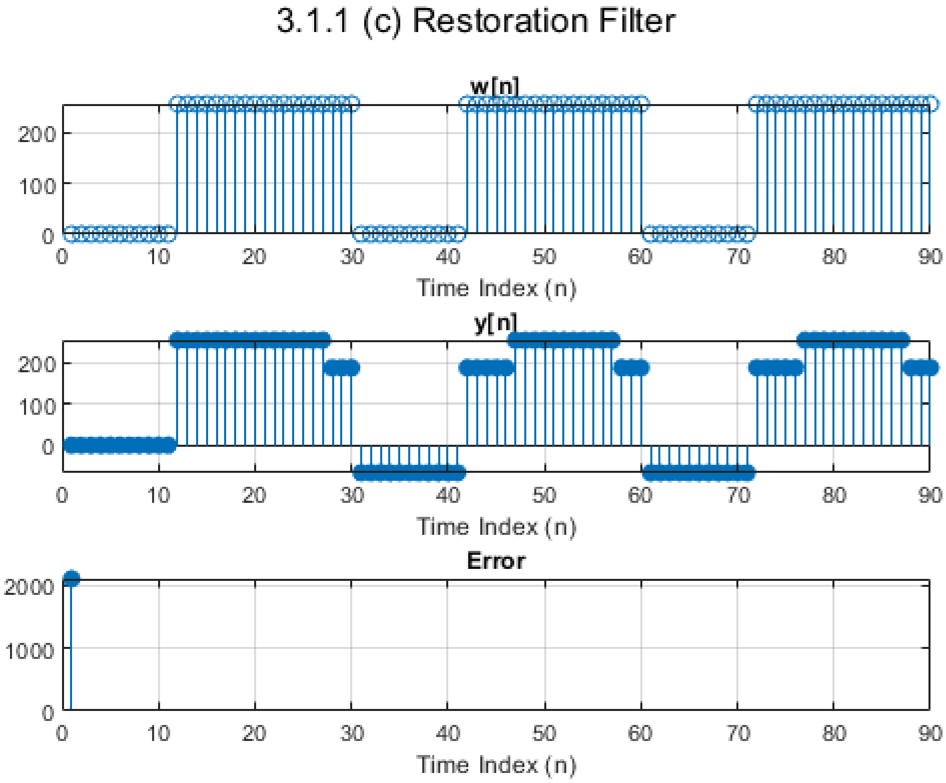
```
% l=1;
% for n = l:1:length(xx)
%  ww(n) = xx(n) - 0.92*xx(n);
% end
%
% stem(ww)
```

3.1.1 Restoration Filter

```
r = 0.92;
M = 15;
cc = r.^[0:M];
yy = conv(cc,ww);
error = immse(yy(1:91),xx(1:91)); % 2.0984e+03

figure('Name', '3.1.1 - Restoration Filter')
```

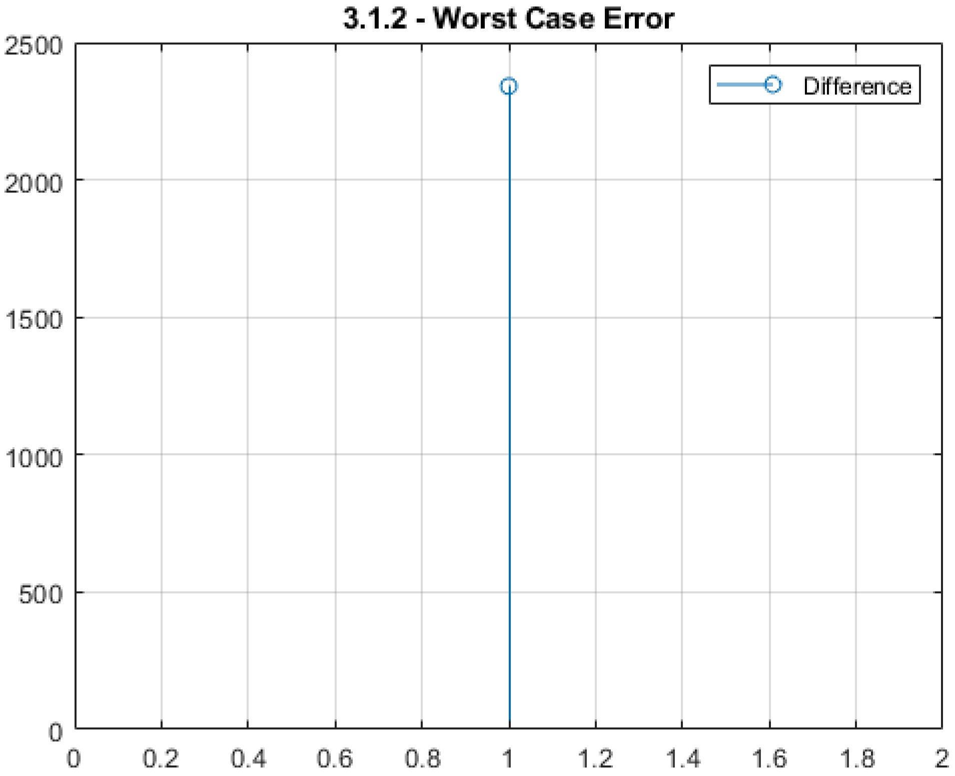
```
subplot(2,1,1)
stem(ww)
grid on
xlim([0 90])
title('w[n]')
xlabel('Time Index (n)')

subplot(2,1,2)
stem(yy, 'filled')
grid on
xlim([0 90])
title('y[n]')
xlabel('Time Index (n)')

sgtitle('3.1.1 Restoration Filter')

% (c)
figure('Name', '3.1.1 (c)')
subplot(3,1,1)
stem(xx)
grid on
xlim([0 90])
title('w[n]')
xlabel('Time Index (n)')

subplot(3,1,2)
stem(yy, 'filled')
grid on
xlim([0 90])
title('y[n]')
xlabel('Time Index (n)')

subplot(3,1,3)
stem(error, 'filled')
grid on
xlim([0 90])
title('Error')
xlabel('Time Index (n)')

sgtitle('3.1.1 (c) Restoration Filter')
```


3.1.1 Restoration Filter
w[n]
200
100
0
-100
-200
0
10
20
30
40
50
60
70
80
90
Time Index (n)
y[n]
300
200
100
0
-100
0
10
20
30
40
50
60
70
80
90
Time Index (n)


3.1.1 (c) Restoration Filter
w[n]
200
100
0
0
10
20
30
40
50
60
70
80
90
Time Index (n)
y[n]
200
100
0
0
10
20
30
40
50
60
70
80
90
Time Index (n)
Error
2000
1000
0
0
10
20
30
40
50
60
70
80
90
Time Index (n)

3.1.2 Worst Case Error

```
diff = max(immse(yy(1:101), xx(1:101)));

figure('Name', '3.1.2 - Worst Case Error')
stem(diff)
hold on; grid on
% stem(yy(1:90))
% stem(xx(1:90))
legend('Difference')%, 'y[n]', 'x[n]')
title('3.1.2 - Worst Case Error')

%{
Error plot & worst case error tells us that it is more difficult to
 reproduce
the original signal.
%}
```



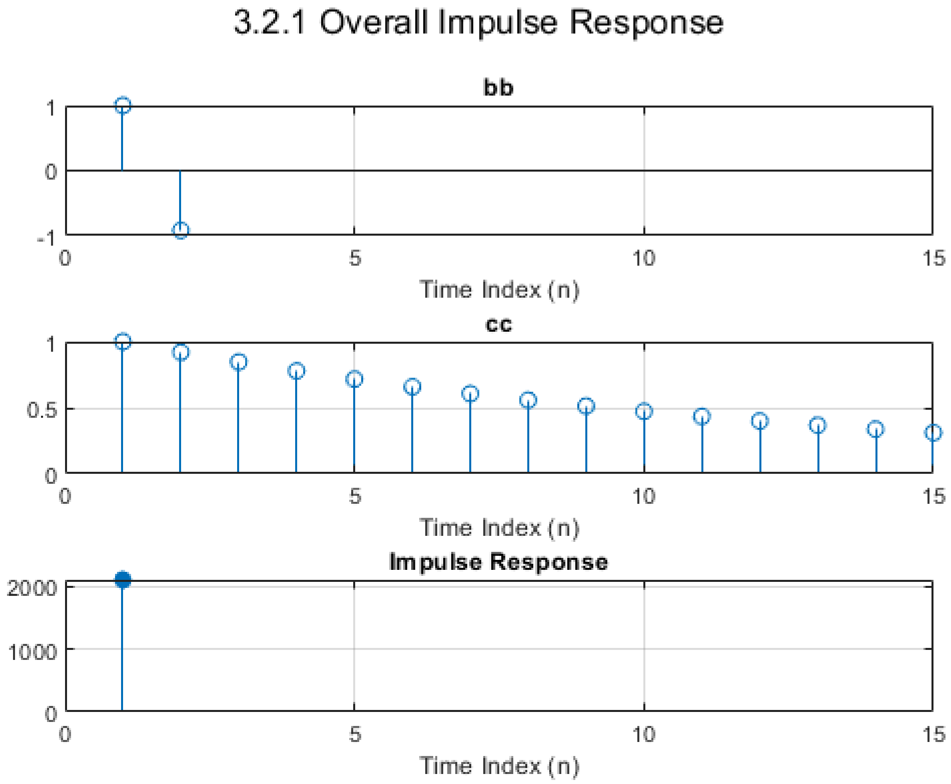
3.1.3 An Echo Filter

```
fs = 8000;
time_delay = 0.2 * fs;
r = 0.9; % 90% of original signal
P = 2000;
```

```
y1n = [1 zeros(1,time_delay-2) r];
x22 = conv(x2, y1n);

% soundsc(y1n)
% soundsc(x2)
soundsc(x22) % new sound with echo

%{
time_delay is 1600 because fs = 8000
magnitude of signal is 90% of original

Coefficients of FIR filter should be r = 0.90.

Convolve the echo filter with the input signal.
%}
```

3.2.1 Overall Impulse Response

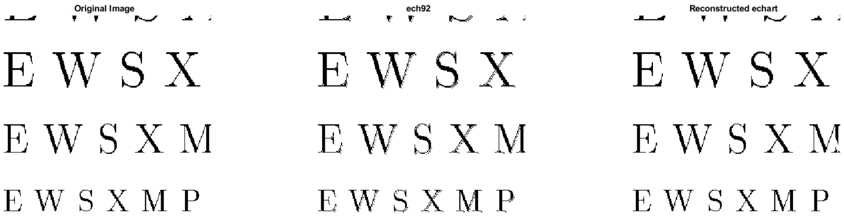
```
yn = conv(cc, ww);
hh = conv(bb, cc);

figure('Name', '3.2.1 - Overall Impulse Response')
subplot(3,1,1)
stem(bb)
grid on
xlim([0 15])
title('bb')
xlabel('Time Index (n)')

subplot(3,1,2)
stem(cc)
grid on
xlim([0 15])
title('cc')
xlabel('Time Index (n)')

subplot(3,1,3)
stem(error, 'filled')
grid on
xlim([0 15])
title('Impulse Response')
xlabel('Time Index (n)')

sgtitle('3.2.1 Overall Impulse Response')
```



3.2.2 Distorting and Restoring Images

```
load echart

% figure('Name', '(a) Show Image: echart')
% imshow(echart)
% title('echart: Original Image')

bb1 = zeros(1,16);
q = 0.92;

for n = 0:15
 bb1(n+1) = q^n;
end

[row, height] = size(echart);
ech92 = zeros(row, height);

for i = 1:row
 w = conv(echart(i,:), bb);
 ech92(i,:) = w(1:height);
end
for i = 1:height
 w = conv(ech92(:,i), bb);
 ech92(:,i) = w(1:row);
```

```
end

reconstructed = zeros(row, height);
for i = 1:row
 w = conv(ech92(i,:), bb1);
 reconstructed(i,:) = w(1:height);
end
for i = 1:height
 w = conv(reconstructed(:,i), bb1);
 reconstructed(:,i) = w(1:row);
end

figure('Name', '3.2.2 - Distorting and Restoring Images', ...
 'units','normalized', 'outerposition', [0 0.50 1 0.5])
subplot(1,3,1)
imshow(echart)
title('Original Image')

subplot(1,3,2)
imshow(ech92)
title('ech92')

subplot(1,3,3)
imshow(reconstructed)
title('Reconstructed echart')

%{
Small "delay" in the image - called: ghosting.
%}
```

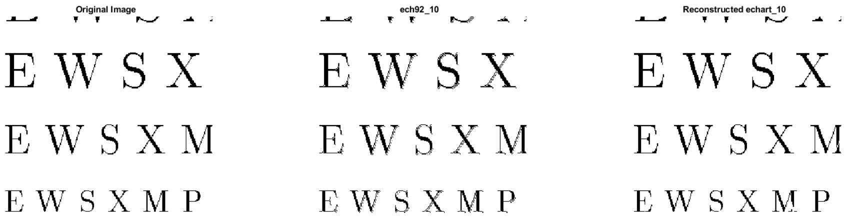


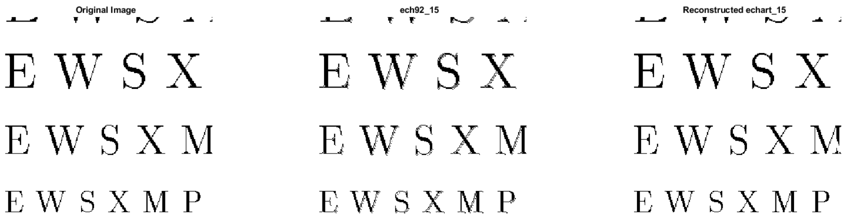
3.2.3 A Second Restoration Experiment

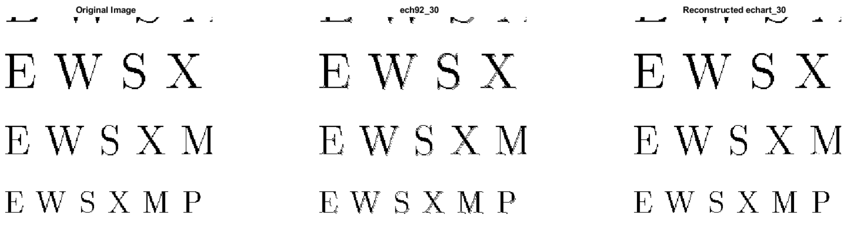
```
% 10
 ===========================================================================
bb2 = zeros(1,11);

for n = 0:10
 bb2(n+1) = q^n;
end

ech92_10 = zeros(row, height);

for i = 1:row
 w = conv(echart(i,:), bb);
 ech92_10(i,:) = w(1:height);
end
for i = 1:height
 w = conv(ech92_10(:,i), bb);
 ech92_10(:,i) = w(1:row);
end

reconstructed_10 = zeros(row, height);
for i = 1:row
 w = conv(ech92_10(i,:), bb2);
 reconstructed_10(i,:) = w(1:height);
end
for i = 1:height
 w = conv(reconstructed_10(:,i), bb2);
 reconstructed_10(:,i) = w(1:row);
end

figure('Name', '3.2.2 - Distorting and Restoring Images: 10', ...
 'units','normalized', 'outerposition', [0 0.50 1 0.5])
subplot(1,3,1)
imshow(echart)
title('Original Image')

subplot(1,3,2)
imshow(ech92_10)
title('ech92\_10')

subplot(1,3,3)
imshow(reconstructed_10)
title('Reconstructed echart\_10')

% 15
 ===========================================================================

bb3 = zeros(1,16);

for n = 0:15
 bb3(n+1) = q^n;
end

ech92_15 = zeros(row, height);

for i = 1:row
 w = conv(echart(i,:), bb);
 ech92_15(i,:) = w(1:height);
end
for i = 1:height
 w = conv(ech92_15(:,i), bb);
 ech92_15(:,i) = w(1:row);
end

reconstructed_15 = zeros(row, height);
for i = 1:row
 w = conv(ech92_15(i,:), bb3);
 reconstructed_15(i,:) = w(1:height);
end
for i = 1:height
 w = conv(reconstructed_15(:,i), bb3);
 reconstructed_15(:,i) = w(1:row);
end

figure('Name', '3.2.2 - Distorting and Restoring Images: 15', ...
 'units','normalized', 'outerposition', [0 0.50 1 0.5])
subplot(1,3,1)
imshow(echart)
title('Original Image')

subplot(1,3,2)
imshow(ech92_15)
title('ech92\_15')

subplot(1,3,3)
imshow(reconstructed_15)
title('Reconstructed echart\_15')

% 30
 ===========================================================================

bb4 = zeros(1,31);

for n = 0:30
 bb4(n+1) = q^n;
end

ech92_30 = zeros(row, height);

for i = 1:row
 w = conv(echart(i,:), bb);
 ech92_30(i,:) = w(1:height);
end
for i = 1:height
 w = conv(ech92_30(:,i), bb);
 ech92_30(:,i) = w(1:row);
end

reconstructed_30 = zeros(row, height);
for i = 1:row
 w = conv(ech92_30(i,:), bb4);
 reconstructed_30(i,:) = w(1:height);
end
for i = 1:height
 w = conv(reconstructed_30(:,i), bb4);
 reconstructed_30(:,i) = w(1:row);
end

figure('Name', '3.2.2 - Distorting and Restoring Images: 30', ...
 'units','normalized', 'outerposition', [0 0.50 1 0.5])
subplot(1,3,1)
```

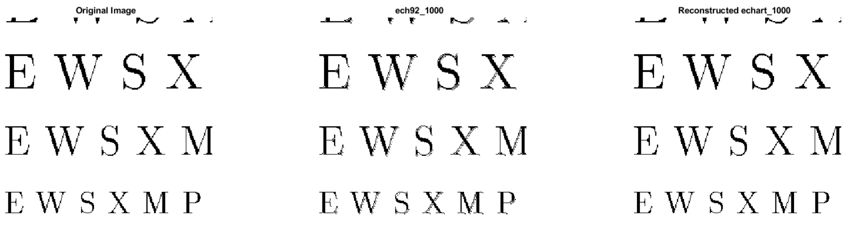
```
imshow(echart)
title('Original Image')

subplot(1,3,2)
imshow(ech92_30)
title('ech92\_30')

subplot(1,3,3)
imshow(reconstructed_30)
title('Reconstructed echart\_30')

% 1000
 =========================================================================

bb5 = zeros(1,1001);

for n = 0:1000
 bb5(n+1) = q^n;
end

ech92_1000 = zeros(row, height);

for i = 1:row
 w = conv(echart(i,:), bb);
 ech92_1000(i,:) = w(1:height);
end
for i = 1:height
 w = conv(ech92_1000(:,i), bb);
 ech92_1000(:,i) = w(1:row);
end

reconstructed_1000 = zeros(row, height);
for i = 1:row
 w = conv(ech92_1000(i,:), bb5);
 reconstructed_1000(i,:) = w(1:height);
end
for i = 1:height
 w = conv(reconstructed_1000(:,i), bb5);
 reconstructed_1000(:,i) = w(1:row);
end

figure('Name', '3.2.2 - Distorting and Restoring Images: 1000', ...
 'units','normalized', 'outerposition', [0 0.50 1 0.5])
subplot(1,3,1)
imshow(echart)
title('Original Image')

subplot(1,3,2)
imshow(ech92_1000)
title('ech92\_1000')

subplot(1,3,3)
imshow(reconstructed_1000)
title('Reconstructed echart\_1000')
```

```
%{
As M increased, the errors approach zero. The higher the M value, the better the
reconstructed image will be.
%}
```









worst case error (wce)

```
wce_10 = max(max(echart - reconstructed_10));
wce_15 = max(max(echart - reconstructed_15));
wce_30 = max(max(echart - reconstructed_30));
wce_1000 = max(max(echart - reconstructed_1000));

fprintf('Worse case error of M = 10: %s\n', num2str(wce_10))
fprintf('Worse case error of M = 15: %s\n', num2str(wce_15))
fprintf('Worse case error of M = 30: %s\n', num2str(wce_30))
fprintf('Worse case error of M = 1000: %s\n', num2str(wce_1000))
```

```
Worse case error of M = 10: 203.8151
Worse case error of M = 15: 134.3307
Worse case error of M = 30: 38.4585
Worse case error of M = 1000: 3.1264e-13
```

Published with MATLAB® R2018b