%part 4.2: For parts a-i

%part a

im = imread('HP110v3.png', 'png'); %opens image

[r,c] = size(im);

xx = im(round(r/2), :); %chooses a row in the middle

%part b

bb = [1, -1]; %defines FIR filter

yy = conv2(xx, bb); %filters xx with FIR filter

n1 = 1:length(xx); %creates stem plot of xx and yy...

n2 = 1:length(yy);

subplot(2,1,1);

stem(n1-1, xx(n1))

title('Plot of x[n]')

subplot(2,1,2);

stem(n2-1,yy(n2),'filled')

xlabel('Time Index (n)')

title('Plot of y[n]')

%part c

T = 210; %defines threshold

dd = [];

for i = 1:length(yy)

if abs(yy(i)) >= T

add = true;

dd = [dd add];

else

add = false;

dd = [dd add];

end

end

loc = find(dd>0);

l = length(loc);

figure

n3 = 1:length(loc);

stem(n3-1, loc(n3))

xlabel('Time Index (n)')

title('Location Signal l[n]')

%part d

deltan = conv(loc, bb);

deltan(end) = []; %removes the large negative number at the end caused by

%the FIR filter

figure

n3 = 1:length(loc);

n4 = 1:length(deltan);

subplot(2,1,1);

stem(n3-1, loc(n3))

title('Location signal l[n]')

subplot(2,1,2);

stem(n4-1, deltan(n4),'filled')

axis([0,70,0,40])

title('Width signal Delta[n]')

xlabel('Time Index (n)')

%part f

list = [];

for i = 1:length(deltan)

if i + 58 <= length(deltan) %checks for a valid subset

vec = deltan(i:i+58); %extracts that subset

addme = sum(vec) / 95; %finds theta for one subset

list = [list addme]; %adds that theta to a list

else

break

end

end

theta = mean(list); %the estimated theta was the average of the thetas from

%the subsets

%part g

newdeltan = round(deltan/theta); %divide deltan by estimated theta and round up

%part h

vecin = newdeltan(7:65); %picks a vector of length 59

out = decodeUPC(vecin); %decodes that vector

%part 4.2: For part j. Identical to code above, except changing inputs to imread, which row is examined, and which subset of the integer deltan values are used.

%part a

im = imread('OFFv3', 'png'); %opens image

xx = im(86, :); %choose a row in the middle

%part b

bb = [1, -1]; %defines FIR filter

yy = conv2(xx, bb); %filters xx with FIR filter

n1 = 1:length(xx); %creates stem plot of xx and yy...

n2 = 1:length(yy);

subplot(2,1,1);

stem(n1-1, xx(n1))

subplot(2,1,2);

stem(n2-1,yy(n2),'filled')

xlabel('Time Index (n)')

%part c

T = 200; %defines threshold

dd = [];

for i = 1:length(yy)

if abs(yy(i)) >= T

add = true;

dd = [dd add];

else

add = false;

dd = [dd add];

end

end

loc = find(dd>0);

l = length(loc);

figure

n3 = 1:length(loc);

stem(n3-1, loc(n3))

xlabel('Time Index (n)')

title('Location Signal l[n]')

%part d

deltan = conv(loc, bb);

deltan(end) = []; %removes the large negative number at the end caused by

%the FIR filter

figure

n3 = 1:length(loc);

n4 = 1:length(deltan);

subplot(2,1,1);

stem(n3-1, loc(n3))

title('Location signal l[n]')

subplot(2,1,2);

stem(n4-1, deltan(n4),'filled')

axis([0,70,0,40])

title('Width signal Delta[n]')

xlabel('Time Index (n)')

%part f

list = [];

for i = 1:length(deltan)

if i + 58 <= length(deltan) %checks for a valid subset

vec = deltan(i:i+58); %extracts that subset

addme = sum(vec) / 95; %finds theta for one subset

list = [list addme]; %adds that theta to a list

else

break

end

end

theta = mean(list); %the estimated theta was the average of the thetas from

%the subsets

%part g

newdeltan = round(deltan/theta); %divide deltan by estimated theta and round up

%part h

vecin = newdeltan(5:63); %picks a vector of length 59

out = decodeUPC(vecin); %decodes that vector