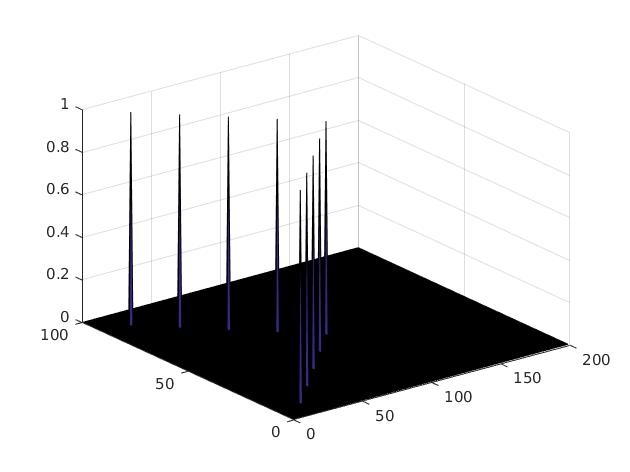
**Q1**

**a):**

**b):**

is located at (m, n)

**c):**

I = zeros(100, 200);

I(10, 20) = 1;

I(20, 40) = 1;

I(30, 60) = 1;

I(40, 80) = 1;

I(50, 100) = 1;

I(60, 80) = 1;

I(70, 60) = 1;

I(80, 40) = 1;

I(90, 20) = 1;

figure;

surf(I);

**d):**

**Q2**

**a):**

**b):**

**c):**

F1 is not separable, since its rank > 1

F2 is separable, F2 = (2 1 2)

**Q3**

**Code:**

**Part 1:**

**%q3 b**

[templates, dimensions] = readInTemplates;

**%q3 c**

img = double(rgb2gray(imread('thermometer.png')));

[height, width] = size(img);

for( i = 1 : 30 )

temp = double(rgb2gray((templates{i})));

originalCorr = normxcorr2(temp, img);

offx = round(dimensions(i).width);

offy = round(dimensions(i).height);

newCorr = originalCorr(offy: offy + height - 1, offx: offx + width - 1);

corrArray(:, :, i) = newCorr;

end

**%q3 d**

[maxCorr, maxIdx] = max(corrArray, [], 3);

**part 2:**

function run(img, maxCorr, maxIdx, corrArray, dimensions, T)

**%display image**

figure;

imagesc(img);

colormap gray;

colorbar; % Turn on color bar on the side

**%q3 e**

idx = 1;

for (row = 1:1268)

for (col = 1:591)

if (maxCorr(row, col) > T)

candidates(idx).row = row;

candidates(idx).col = col;

idx = idx + 1;

end

end

end

**%q3 f**

ans = (size(candidates));

len = ans(2);

for (j = 1:len)

candidate = candidates(j);

x= candidate.col;

y = candidate.row;

templateIndex = maxIdx(y, x);

thisCorr = corrArray(: ,:, templateIndex);

window = thisCorr(y-1:y+1, x-1:x+1);

if(thisCorr(y, x) == max(window(:)))

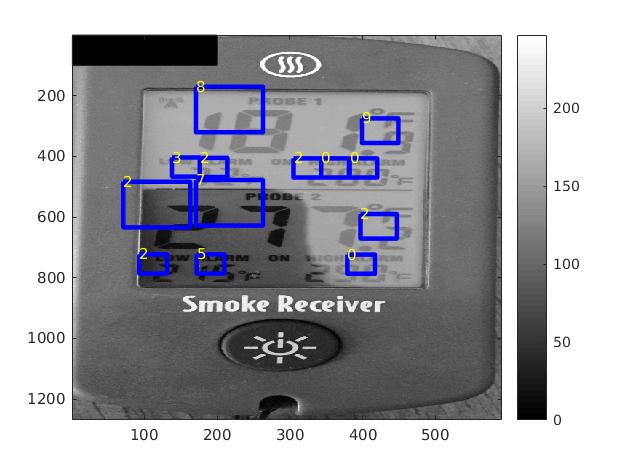
drawAndLabelBox(x, y, templateIndex, dimensions);

drawnow;

end

end

**Result:**



**f) v):**

Threshold: 6.8

Number of correctly labeled digits: 13

**f)vi):**

* Choose templates such that size and shape of digits in templates matches those undetected digits in “thermometer.png”
* Then correlation of chosen templates with “thermometer.png” will output higher intensity at corresponding digits
* Then for the same threshold, those digits that are not pass originally will pass
* Then higher accuracy