Question 1

a)

1D-impulse function:

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{else} \end{cases}$$

2D-impulse function:

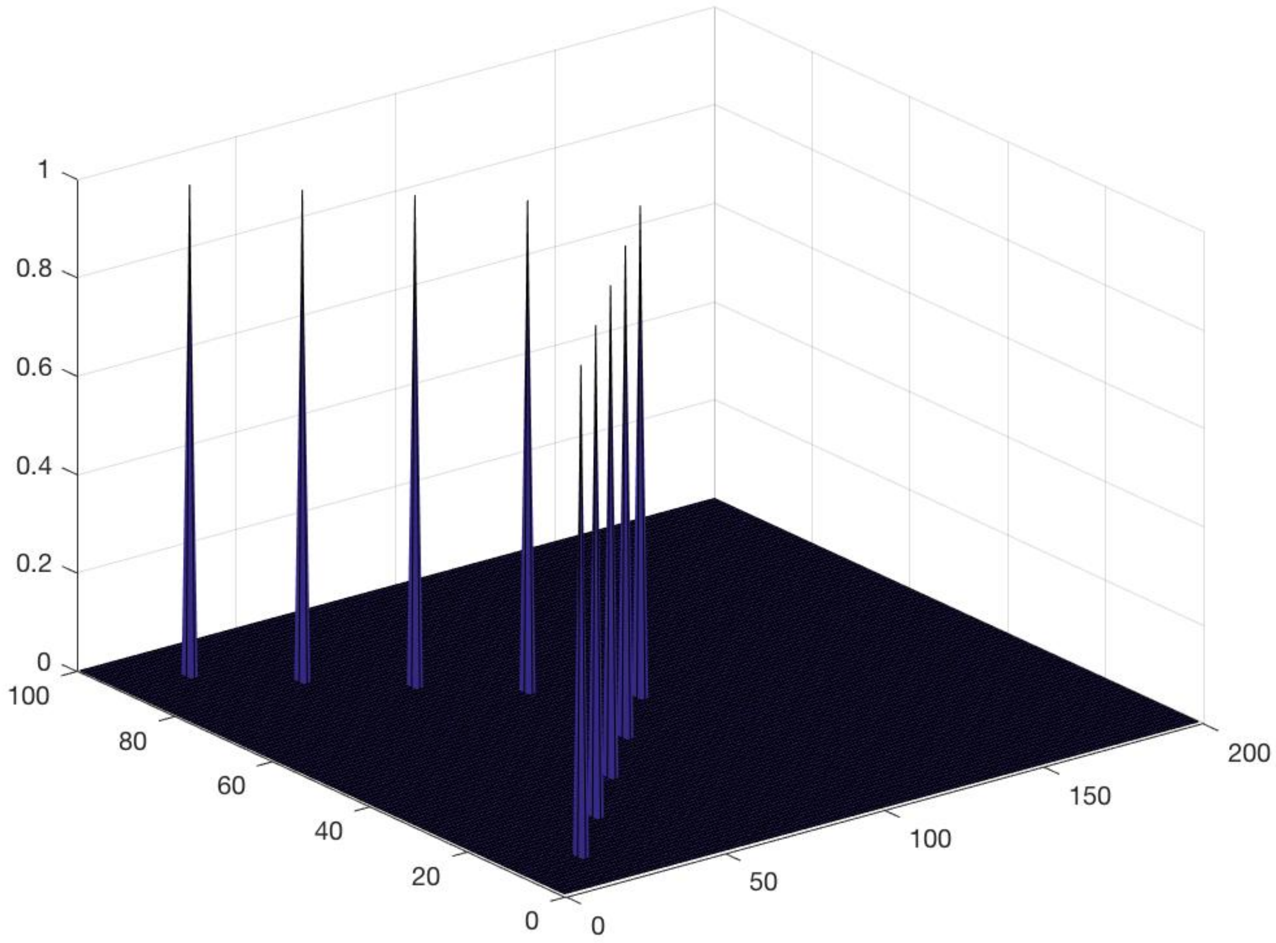
$$\delta(u, v) = \begin{cases} 1 & \text{if } u = 0 \text{ and } v = 0 \\ 0 & \text{else} \end{cases}$$

b)

$$\delta(u - m, v - n) = \begin{cases} 1 & \text{if } u = m \text{ and } v = n \\ 0 & \text{else} \end{cases}$$

c)

```
img = zeros(100, 200);
    xs = [10, 20, 30, 40, 50, 60, 70, 80, 90];
    ys = [20, 40, 60, 80, 100, 80, 60, 40, 20];
    for i = 1 : 9
        img(xs(i), ys(i)) = 1;
    end
    surf(img);
```



d)

$$f(m,n) = \sum_{j=1}^{N} \sum_{i=1}^{M} f(i,j) \cdot \delta(m-i, u-j)$$

Question 2

a)

For each pixel, the convolution take m^2 operation. For the total $n \times n$ pixels, it repeats n^2 times. The total operational cost is $O(m^2 n^2)$

b)

If h is separable, filters are separated into sizes of $m \times 1$ and $1 \times m$. For each pixel, applying both filters costs $2 \cdot m$ operations. The total operational cost is $O(mn^2)$

c)

$\mathbf{F_1}$ is not separable.

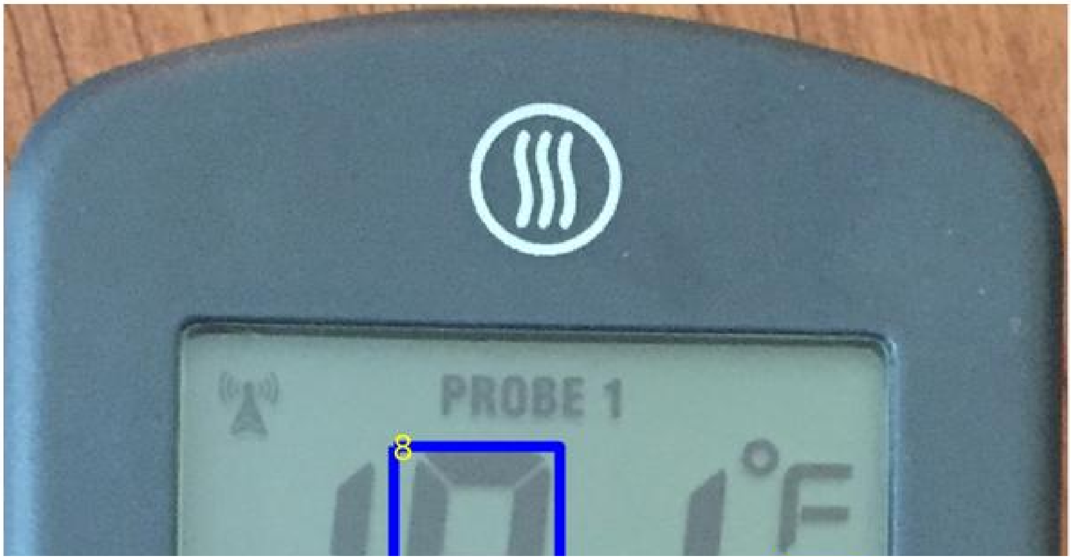
$\mathbf{F_2}$ is separable. With horizontal filter: $\begin{bmatrix} 2 & 1 & 2 \end{bmatrix}$ and vertical filter: $\begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}$

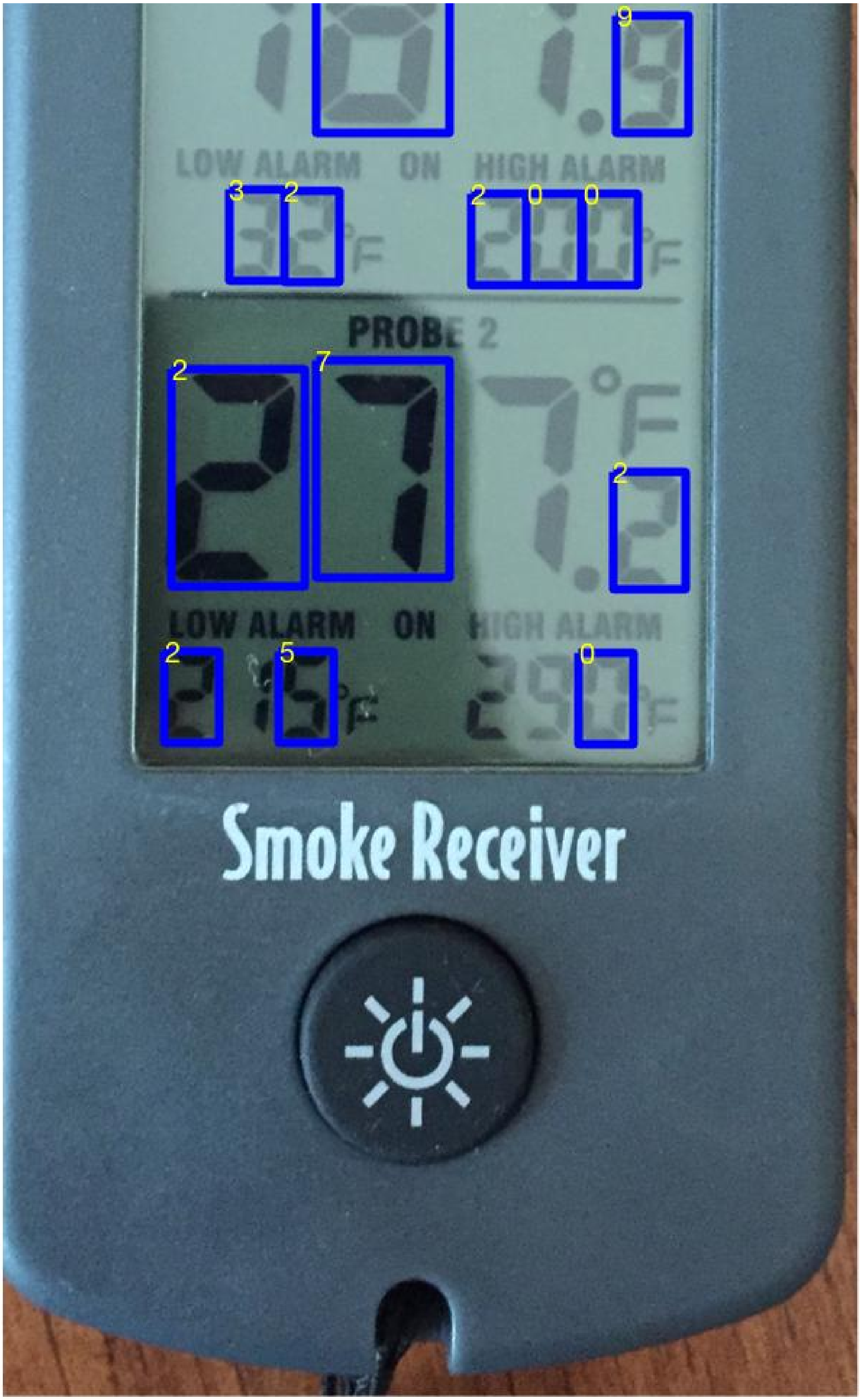
Question 3

```
% Q3
    % b)
    [templates, dimensions] = readInTemplates();

    % c)
    img = imread('thermometer.png');
    imshow(img);
    drawnow();
    grayImg = im2double(rgb2gray(img));
    [M, N] = size(grayImg);
    corrArray = zeros(M, N, length(templates));
    for i = 1 : length(templates)
        rgbTemp = rgb2gray(cell2mat(templates(i)));
        grayTemp = im2double(rgbTemp);
        result = normxcorr2(grayTemp, grayImg);
        [tH, tW] = size(grayTemp);
        offSetX = round(tW/2);
        offSetY = round(tH/2);
        corrArray(:,:,i) = result(offSetY : offSetY + M - 1, offSetX : offSetX + N-1);
    end

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

    % e)
    T = 0.5;
    [candY, candX] = find(maxCorr > T);

    % f)
    for i = 1 : length(candX)
        % f) i.
        templateIndex = maxIdx(candY(i), candX(i));
        % f) ii.
        thisCorr = corrArray(:,:, templateIndex);
        % f) iv.
        if isLocalMaximum(candX(i), candY(i), thisCorr)
            drawAndLabelBox(candX(i), candY(i), templateIndex, dimensions);
            drawnow();
        end
    end

    % f) iii.
    function result = isLocalMaximum(x, y, thisCorr)
        x1 = x - 1;
        x2 = x + 1;
        y1 = y - 1;
        y2 = y + 1;
        [M, N] = size(thisCorr);
        if x1 < 1
            x1 = 1;
            x2 = 2;
        end
        if x2 > N
            x1 = N-1;
            x2 = N;
        end
        if y1 < 1
            y1 = 1;
            y2 = 2;
        end
        if y2 > M
            y1 = M-1;
            y2 = M;
        end
        result = max(thisCorr(y1:y2,x1:x2)) <= thisCorr(y, x);
    end
```

f) v.

With Threshold $T = 0.5$




9
LOW ALARM ON HIGH ALARM
3 2
2 0 0
°F
PROBE 2
2 7
°F
2
LOW ALARM ON HIGH ALARM
2 5
0
°F
Smoke Receiver

f) vi.

Since all the templates are cropped from the image, then for each template in the image there must be a pixel where correlation is 1 (the center pixel of the template). This improves the performance.