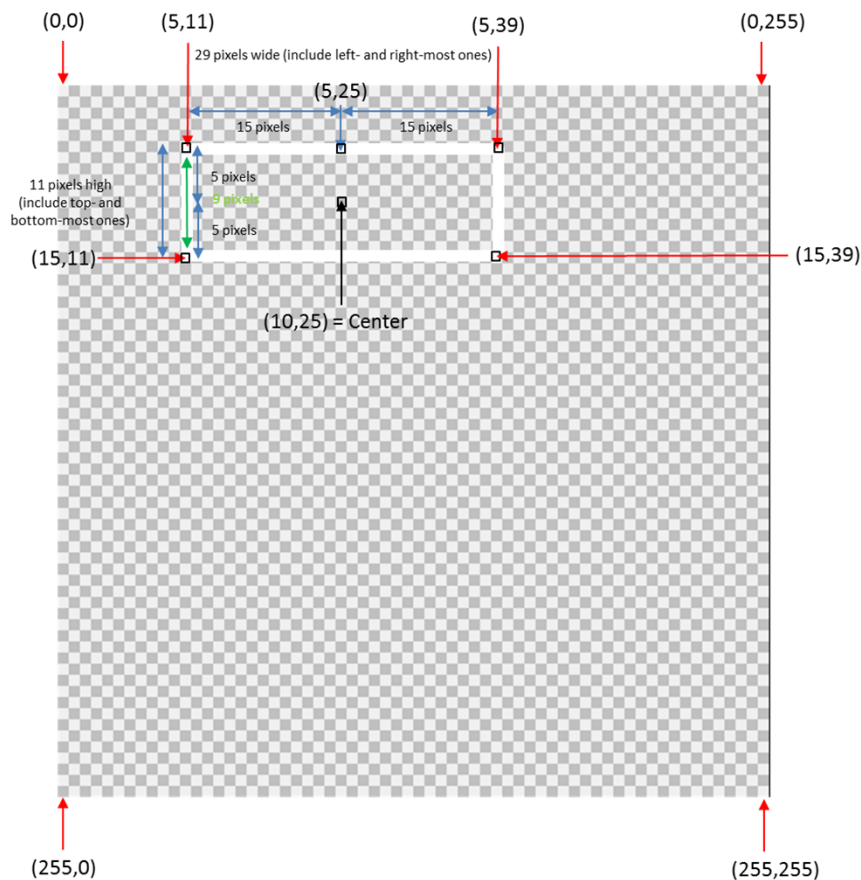


Image Processing: White Box on Image (Drawing Images)

Preamble: In some applications (e.g. tracking an object in the image), one wishes to draw an object (e.g. a white rectangle) on the image. In grey-scale images, recall that a pixel value of 0 is BLACK and 255 is WHITE. Thus, to overlay a white box on an image, one has to assign 255 to those pixels that represent the rectangle.

Example: Desire a rectangle centered at (10,25) with 14 pixels to left and right, and 5 pixels up and down



Q1. What is coordinate of top-left corner?

Row should be 5
 Say $10 - (11/2) = 4.5$ (round up to 5)
 Or, say $10 - (9/2) = 5.5$ (round down to 5)

Column should be 11
 Say $25 - (29/2) = 10.5$ (round up to 11)
 Or, say $25 - (27/2) = 11.5$ (round down to 11)

Q2. What is coordinate of top-right corner?

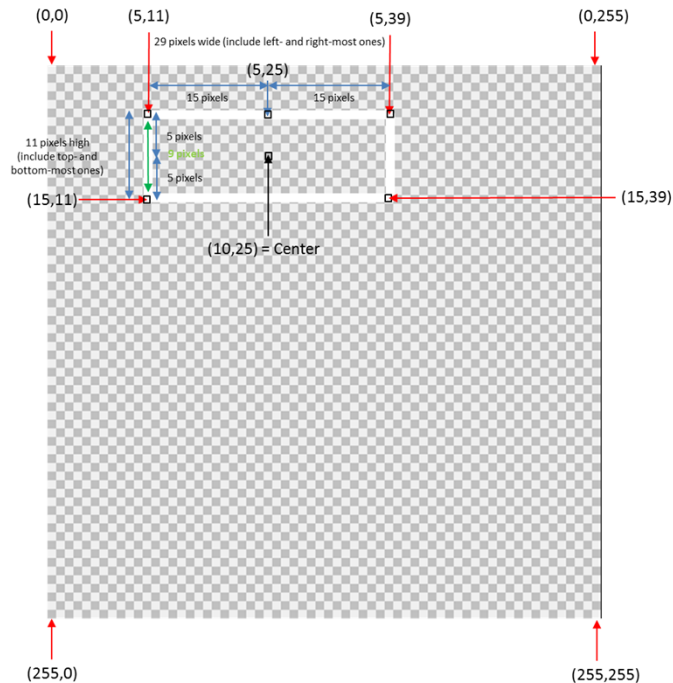
Row should be 5
 Say $10 - (11/2) = 10.5$ (round up to 5)
 Or say $10 - (9/2) = 5.5$ (round down to 5)

Column should be 39
 Say $25 + (29/2) = 39.5$ (round down to 39)
 Or say, $25 + (27/2) = 38.5$ (round up to 39)

Q3 What is coordinate of bottom-left corner? (15,11)

Row should be 15
 Say $10 + (5) = 15$

Column should be 11
 Say $25 - (14) = 11$



Q4 What is coordinate of bottom-right corner? (15,39)

Row should be 15
Say $10 + (5) = 15$

Column should be 39
Say $25 + (14) = 39$

14 pixels to left and right means: $14 + 14 + \text{center} = 29$ pixels wide
5 pixels up and down means: $5 + 5 + \text{center} = 11$ pixels high

Thus, the algorithm begins by defining the rectangle's height, width and desired location in the image that this rectangle's center should be. These are used to calculate the rectangle's corners. Nested for-loop checks location (e.g. top or bottom of rectangle and left or right side of rectangle)

```

for(i=0; i<In->Rows; ++i) {
    for(j=0; j<In->Cols; ++j) {
        val = *(In->Data + i*In->Rows + j);
        if( (i==boxTopLeftRowCorner || i==boxBottomLeftRowCorner) ) {
            // OK, we're on box's top or bottom row
            if( (j>=boxTopLeftColCorner && j<=boxTopRightColCorner) || (j>=boxBottomLeftColCorner && j<=boxBottomRightColCorner) ){
                // Draw top OR bottom line
                val = 255; // make row white between left and right side
            }; // otherwise just keep the original value of val
        }; // end if that checks for box's top or bottom row

        if( j==boxTopLeftColCorner || j==boxTopRightColCorner ) {
            // OK, we're on left or right side
            if( (i>=boxTopLeftRowCorner && i<=boxBottomLeftRowCorner) || (i>=boxTopRightRowCorner && i<=boxBottomRightRowCorner) ) {
                // Draw left OR right line
                val = 255; // make column white between top and bottom row
            };
        }; // end if that checks for box's left or right side
        tmp = Out->Data + i*Out->Rows + j;
        *tmp = (unsigned char)val;
    };
};

```