# Integral Image Generation

Since the Haar classifier needs to access multiple integral image values at a time, integral image buffer needs to be stored at registers. We use the architecture described in [paper Cho] as it retains neighboring pixel values so we don’t have to provide every pixel value when we shift the window.

The architecture consists of image line buffer, image window buffer, and integral image buffer. The image line buffer stores every incoming pixel values in a series of BRAM with its x-coordinate as its address. The number of BRAMs used is the same as the row of the image window with each BRAM has depth same as the width of the image. When a pixel is inputted, the current values at the specific x-coordinate of the incoming pixel is shifted to its next BRAM at the same address so each column of the line buffer stores a sorted pixel values at its address. This way, every pixel needed when the image window shifts is ready to be inputted so the wait time is significantly reduced.

When a pixel is incoming to the buffer, its x-coordinate is used as the address to read and write from the line buffer’s BRAMs. These read values are inputted to the image window buffer, which consists of a 2n × n array of registers, where n is the image window row and column size. The values from the line buffer are inputted to the right-most registers of the window buffer. Previous values are shifted to the left, and at some specific cells it sums two values from its right cells. Since each cell is a register, each values can be read immediately and be used to compute integral image buffer values.

The integral image buffer stores integral pixel values of the image window at a coordinate. The window shifts when a pixel is inputted to the line buffer, and the integral pixel values are computed from its previous values and values from the window buffer. For the integral image value of position of the window, the value of the image window buffer at position , the next value of is the following.

With this configuration, whenever a pixel is inputted the left-most components from the window buffer is removed from integral image buffer so the computed values represents a shifted window.

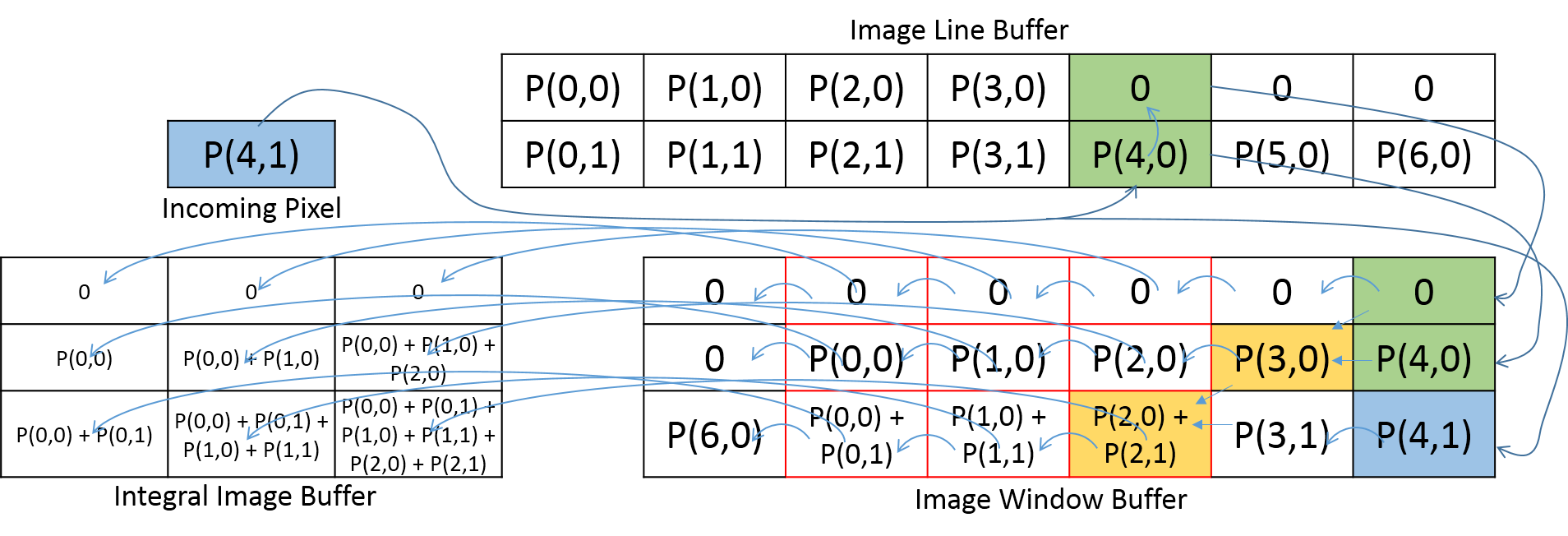


Figure 1 Integral image generation of an image with width 7 pixels and window of 3 × 3. Green shading shows current active address of image line BRAMs.

By using this architecture, the integral image buffer will represent integral pixel values of the window at position after pixel values inserted, where width of the image and size of the window. Therefore, the control unit needs to count number of pixels that has been inserted to the buffer and the classifier needs to wait until the integral image buffer is filled. Since the integral image buffer represents integral values of window at after pixel values inserted, the integral image buffer is values late from the incoming pixel coordinate. In figure 1, it is showed by the bottom-right-most integral pixel value of P(2,1) against incoming pixel value of P(4,1). Because of this, the control unit needs to insert dummy values to shift the window after the last pixel value is inserted.