COMP 558

Lecture 13

Tracking using histograms

Thurs. Oct. 18, 2018

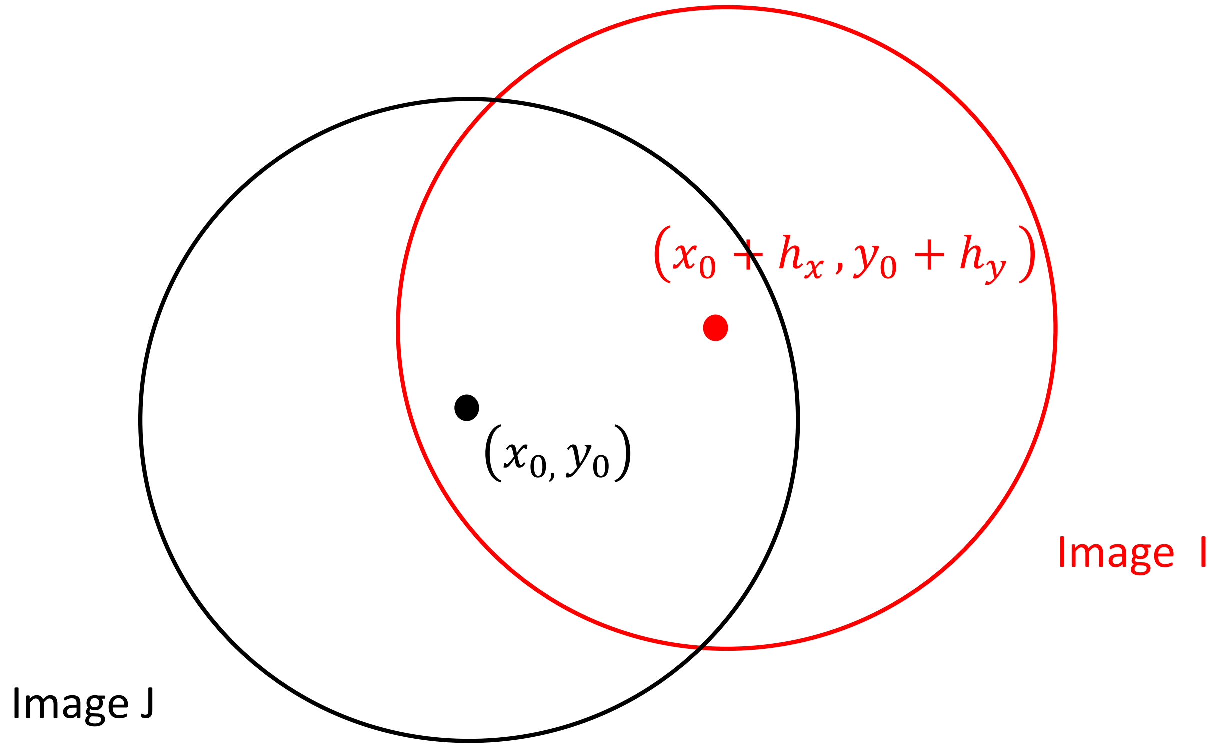
Recall the image registration problem (lecture 11):

For each (x_0, y_0), find the (h_x, h_y) that minimizes:

$$\sum_{(x,y)\in Ngd(x_0,y_0)} \{I(x+h_x, y+h_y) - J(x,y)\}^2$$

We saw the Lucas-Kanade method in lecture 11.
A more general motion model was considered in lecture 12.

For each $(x_{0,} y_0)$, find the $(h_{x,} h_y)$ that minimizes:



Typically one give more weight to the pixels near the center of the window.

$$W(x - x_0, y - y_0)$$

$$\sum_{(x,y)\in Ngd(x_0,y_0)} \{I(x + h_x, y + h_y) - J(x, y)\}^2$$

$W(\ )$ could be a Gaussian shaped function.

Tracking

Estimate the position of something over multiple frames of a video (not just two frames).

Registration-based Tracking

Perform frame-to-frame registration of a local patch: model how its translates and deforms over time.

Registration methods work best in regions that have gradients in multiple directions. So, ensure this! (e.g. condition on eigenvalues of 2^{nd} moment matrix).

Loosely, we think of “image features” such as corners whose position can be uniquely identified.

Registration-based tracking can fail when objects have moving parts e.g. People!

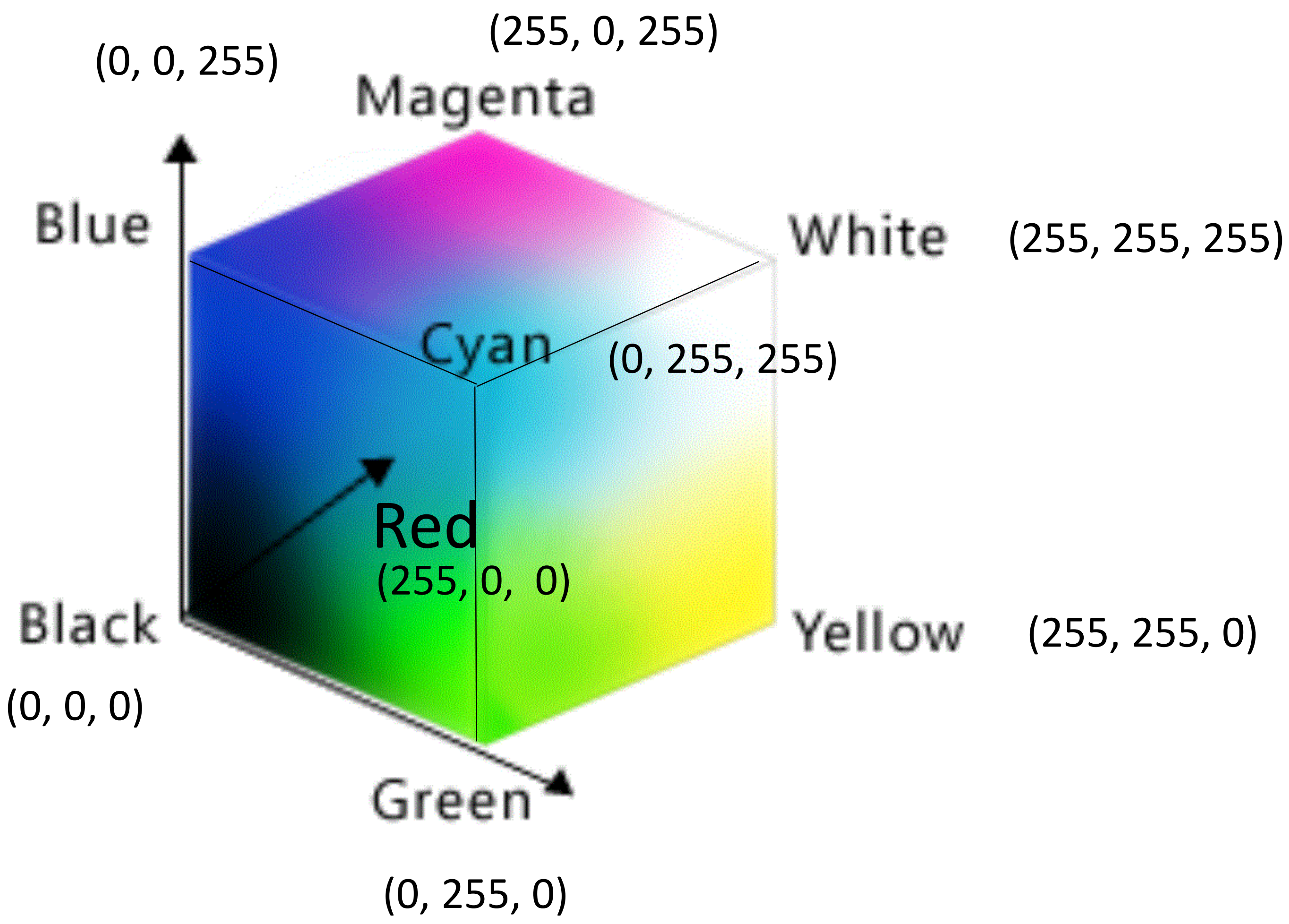
Track this player

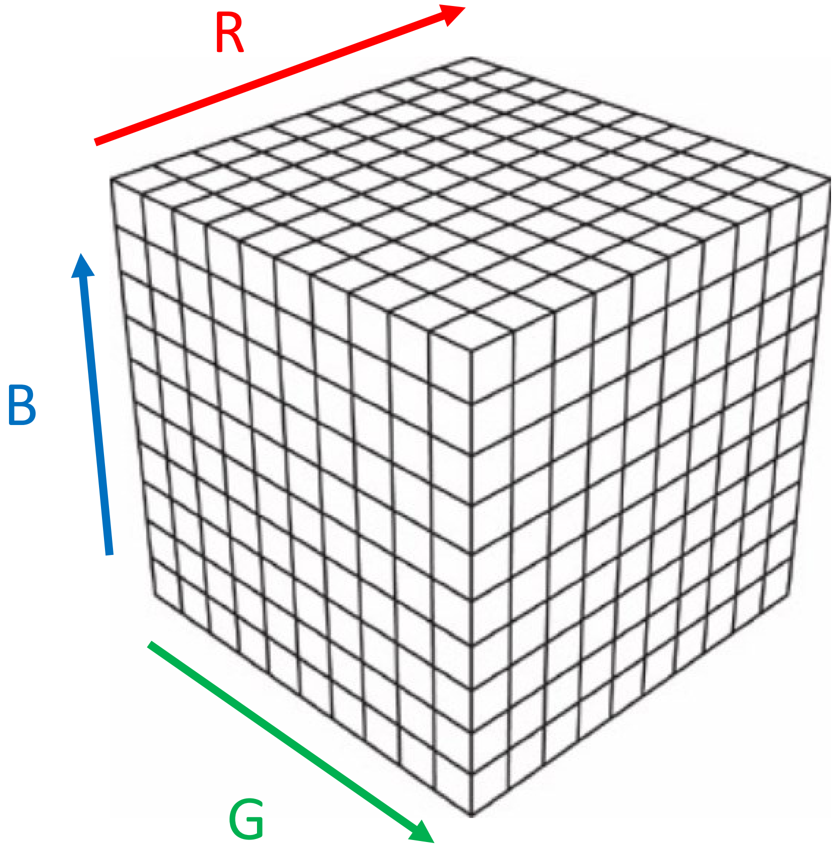
Different approach: Histogram-based tracking

If you just want to track a person's location over multiple frames of a video,   it is often enough to use an RGB histogram.

How to set up the problem ?

RGB values: 0 to 255 (8 bits)





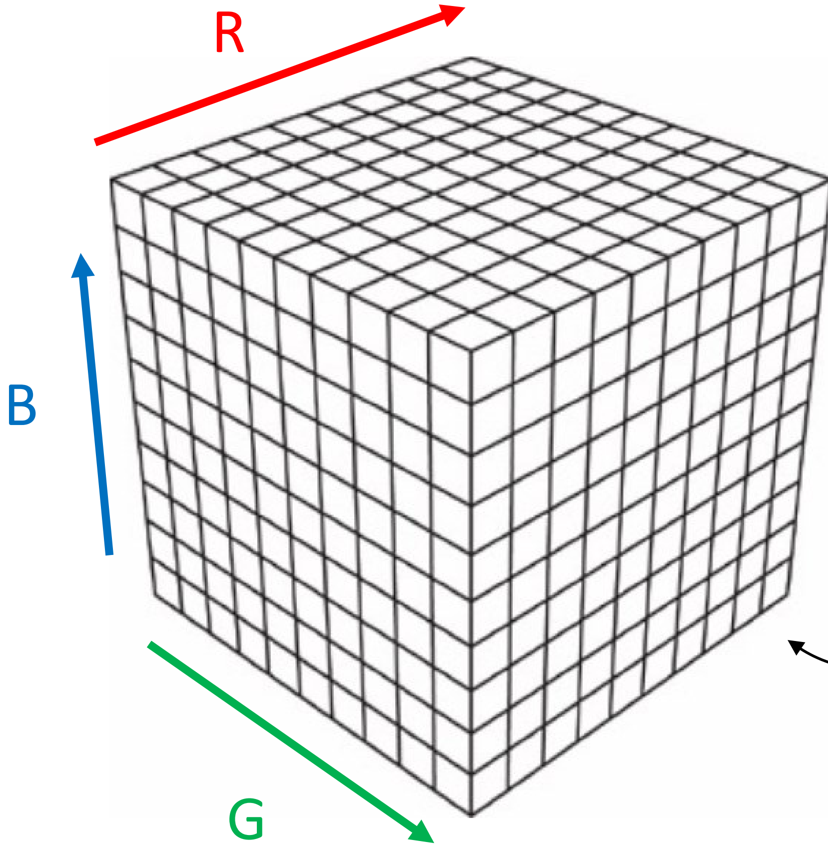
Suppose we partition each axis into 8 levels.
This would give 512 = 8*8*8 bins.
(Sorry the picture is 10*10*10.)

We will call the bins u.

Each bin represents a range of
256/8 = 32 levels of each R, G, B. (crude!)

e.g.

R in [32,63], G in [224, 255], B in [96 ,127].



Suppose we partition each axis into 8 levels.
This would give 512 = 8*8*8 bins.

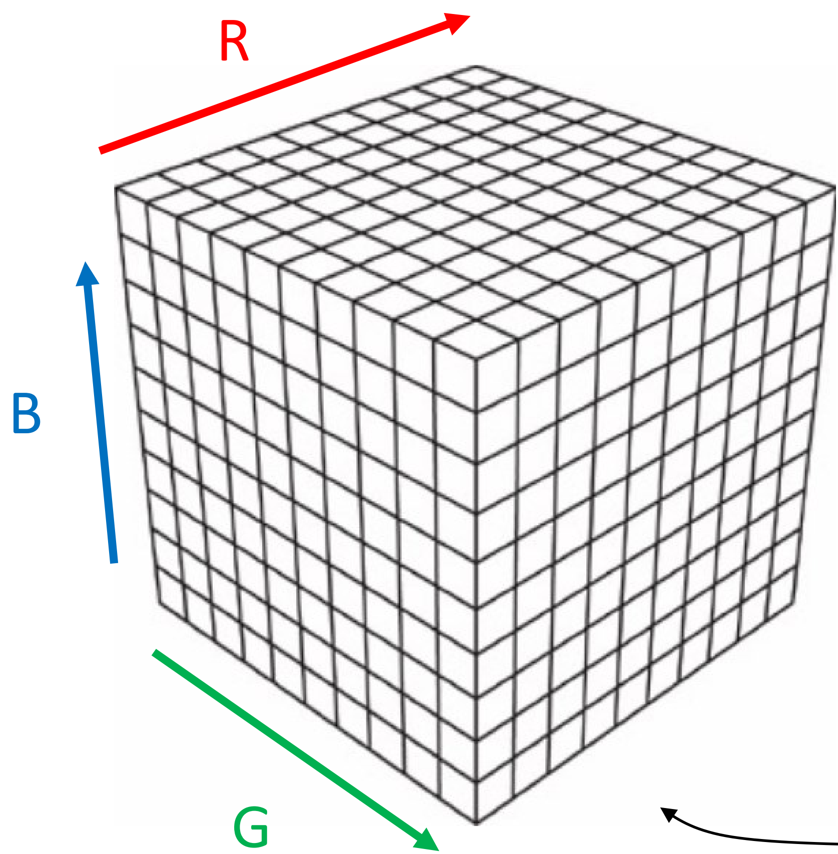
We will call the bins u.

Suppose we have an RGB image $I(x)$ where x is a pixel.

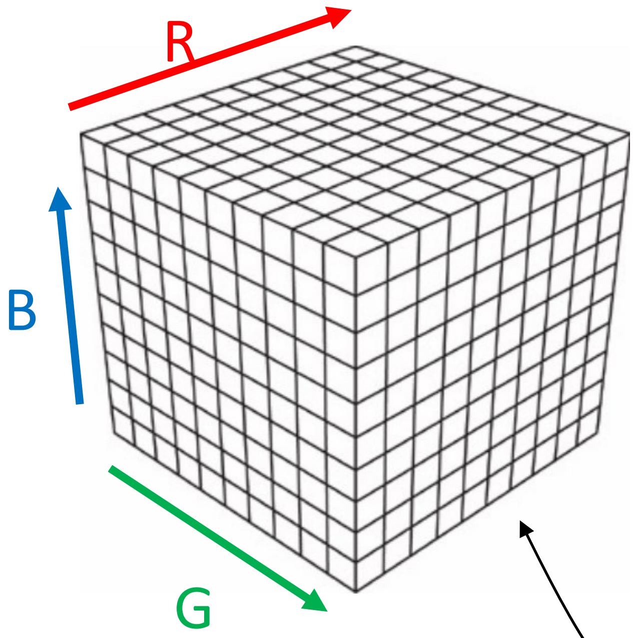
$$u = bin(I(x))$$

Maps pixel x to bin u.

A histogram counts the number of pixels that map to each bin in RGB space.



$$hist(u) \equiv \sum_x \delta(u - bin(I(x)))$$



Today we will define histograms over a region of interest (ROI), centered at some pixel location y.

How many pixels have RGB value in bin u ?

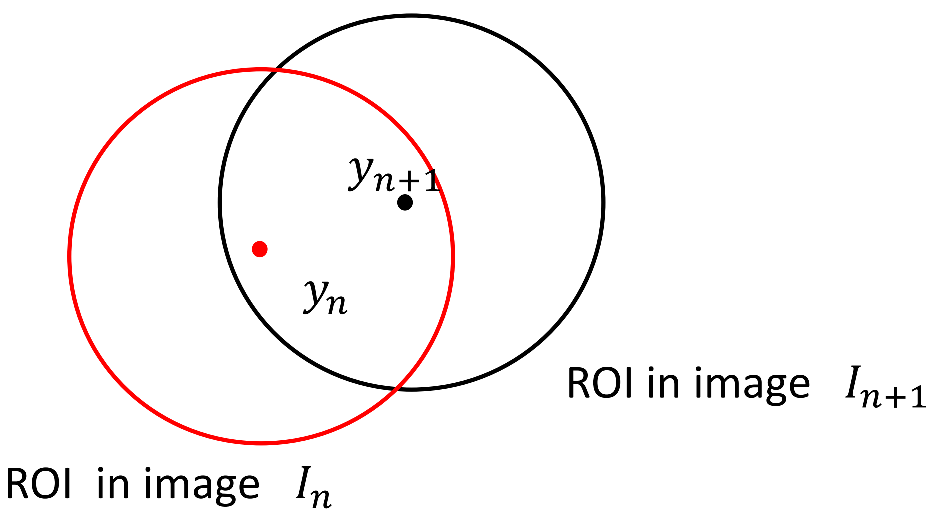
$$hist(u;y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I(x_i)))$$

We want to track the object over many frames.

Let the image be $I_1, I_2, \dots, I_n, I_{n+1}, \dots\dots$

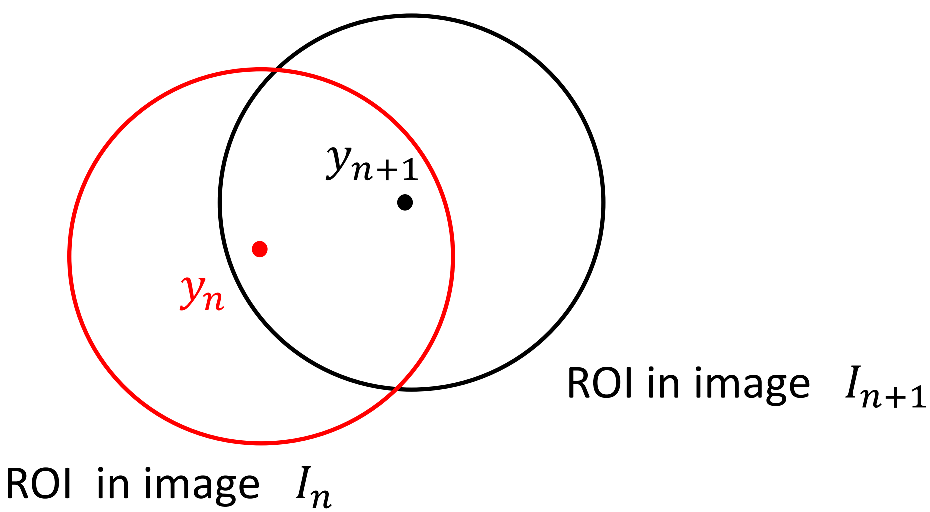
Suppose we initialize a ROI at position y_1 in image 1

Given position y_n, estimate position y_{n+1}.



Given position y_n centered at ROI in frame I_n , find the nearby position y_{n+1} in frame I_{n+1} that *maximizes the similarity of the histograms*.

How do we define this ?



For simplicity, think of 1D images positions (x) and 1D RGB bins (u).

How do we define histograms ?

bin u

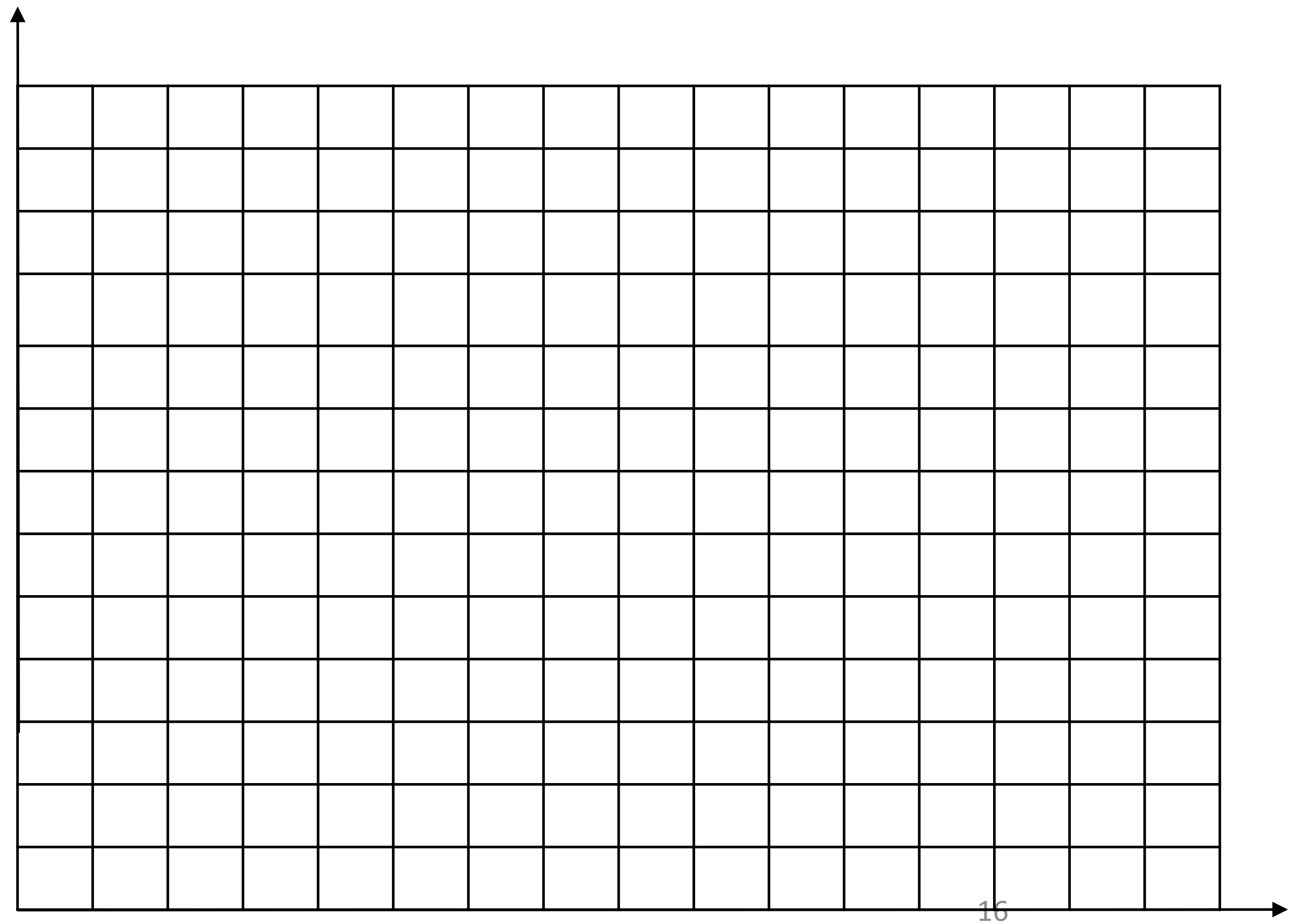


Image position x

For each image position, there is an RGB value, and so there is one bin value u. If $u = bin\big(I(x)\big)$, then we put a value 1 in that bin.

Note: each column sums to 1, but rows typically do not sum to 1.

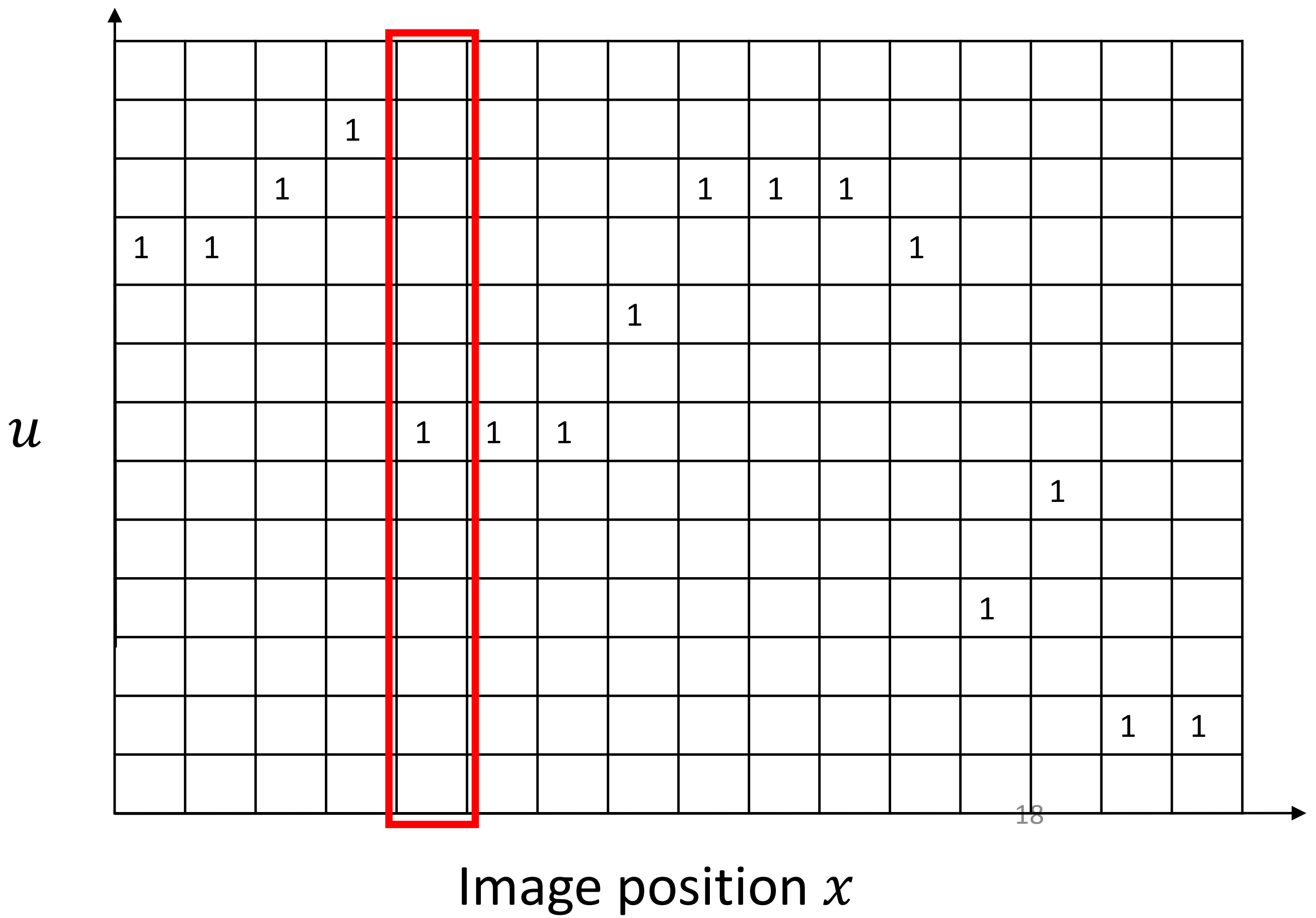
u

			1												
		1						1	1	1					
1	1										1				
							1								
				1	1	1									
													1		
												1			
														1	1

Image position x

For each x, we have a histogram:

$$hist(u; x) = \delta\Big(u - bin\big(I(x)\big)\Big).$$



Histogram for whole image.

u

			1												
		1						1	1	1					
1	1										1				
							1								
				1	1	1									
													1		
												1			
														1	1

Image position x

0
1
4
3
1
0
3
1
0
1
0
2
0

How do we define histograms on ROI's ?

bin u

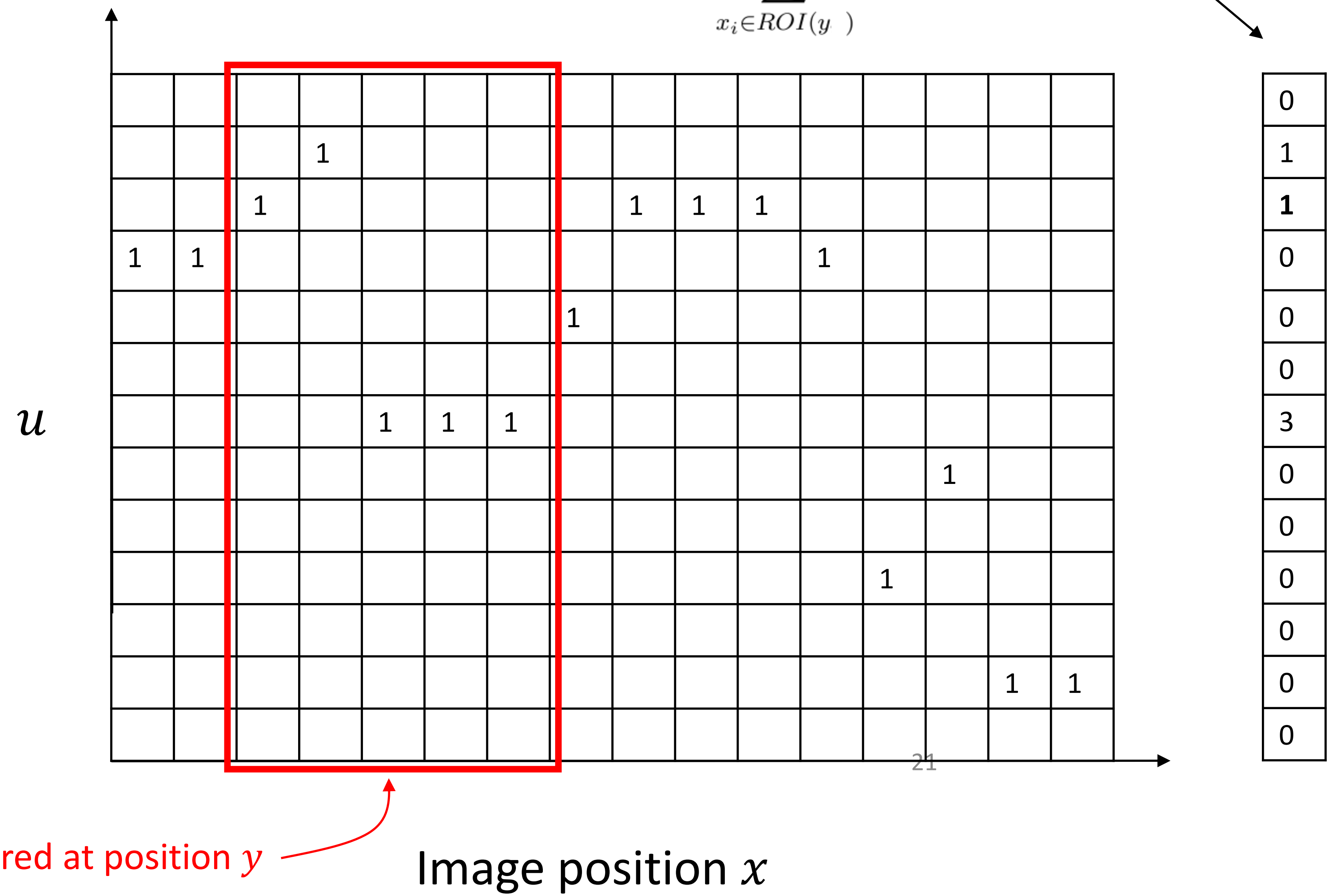
			1												
		1						1	1	1					
1	1										1				
							1								
				1	1	1									
													1		
												1			
														1	1

20

Image position x

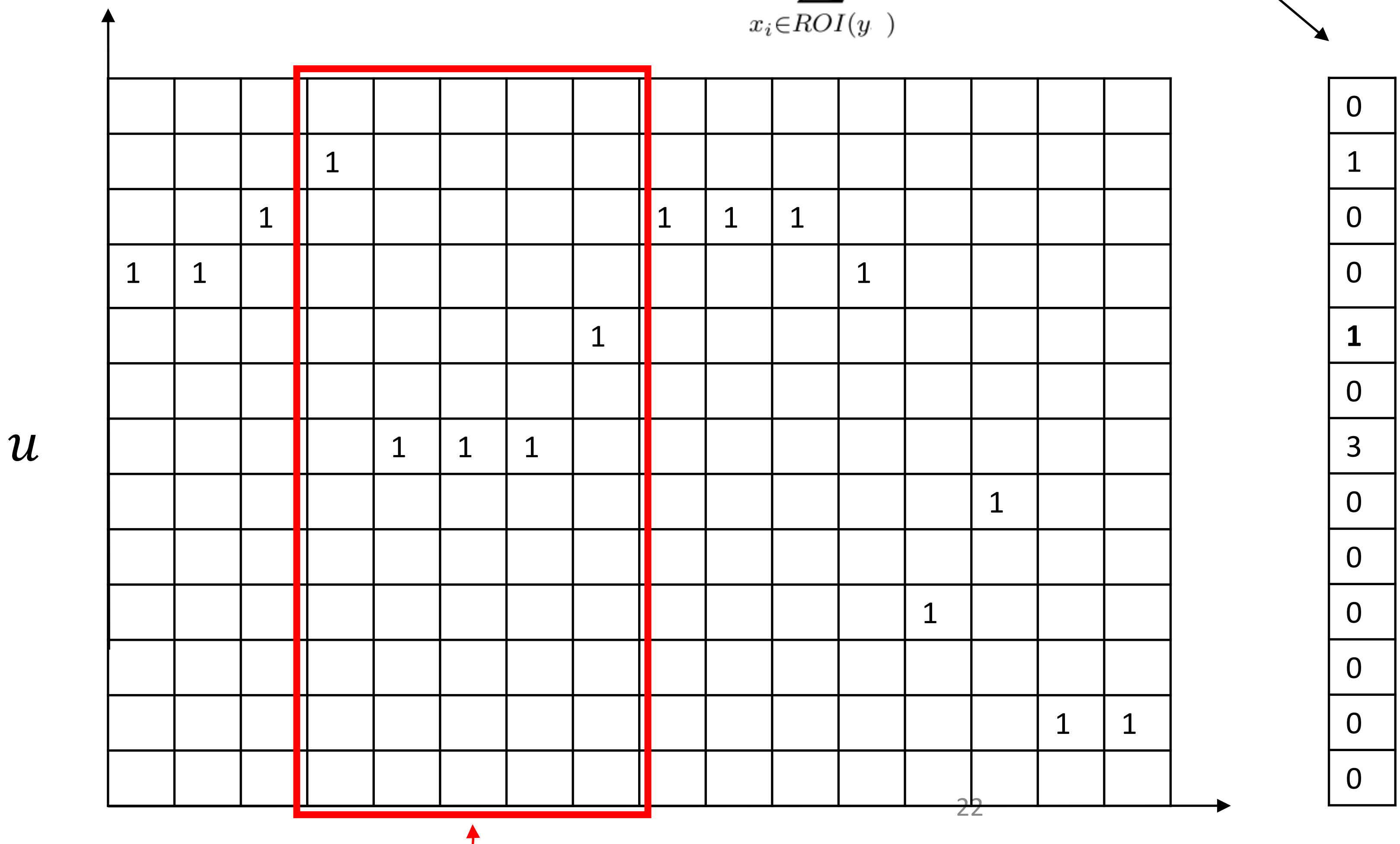
Histogram for the ROI centered at location y.

$$hist_n(u; y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I(x_i)))$$



Histogram for the ROI centered at location y.

$$hist_n(u; y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I(x_i)))$$

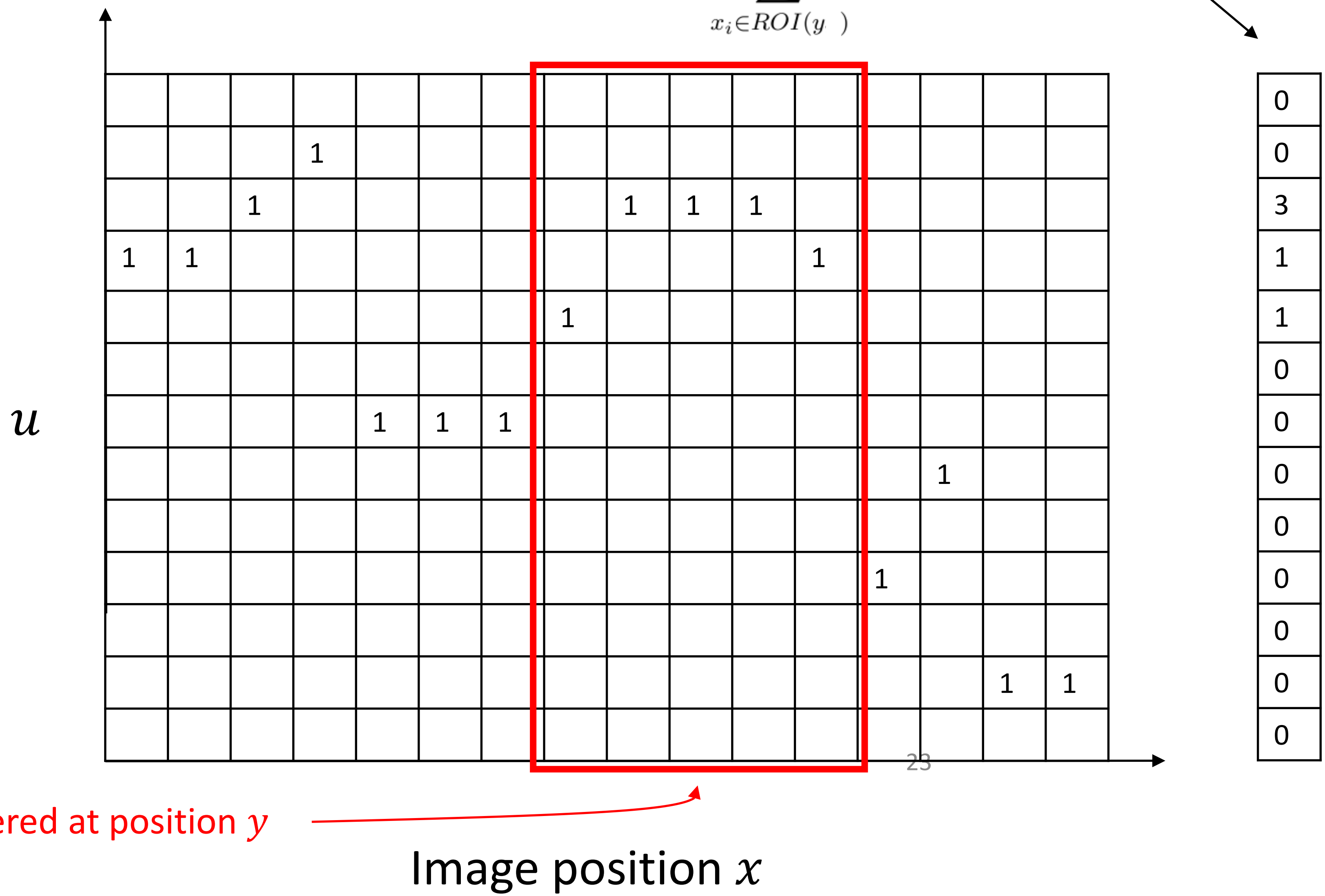


ROI centered at position y

Image position x

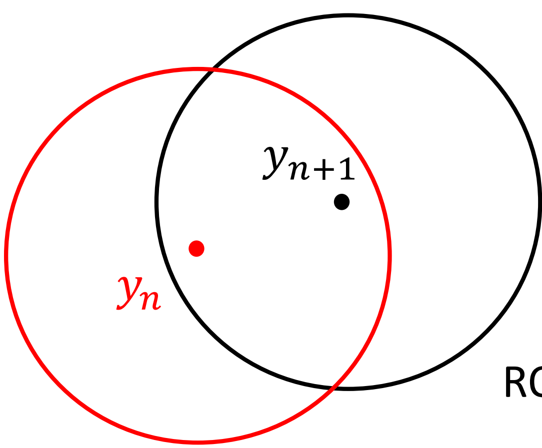
Histogram for the ROI centered at location y.

$$hist_n(u; y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I(x_i)))$$



Given position y_n centered at ROI in frame I_n , find the nearby position y_{n+1} in frame I_{n+1} that *maximizes the similarity* of the ROI histograms.

How do we define this ?



ROI in image I_{n+1}

ROI in image I_n

Brute force tracking with ROI histogram comparison.

Histogram for ROI centered at y_n in frame I_n.

$$hist_n(u; y_n) = \sum_{x_i \in ROI(y_n)} \delta(u - bin(I(x_i)))$$

Histogram for ROI centered at y in frame I_{n+1}.

$$hist_{n+1}(u; y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I_{n+1}(x_i)))$$

Let y_{n+1} be the position y in frame I_{n+1} that *maximizes the similarity* of the histograms.

Histogram for ROI centered at y_n in frame I_n.

$$hist_n(u; y_n) = \sum_{x_i \in ROI(y_n)} \delta(u - bin(I(x_i)))$$

Histogram for ROI centered at y in frame I_{n+1}.

$$hist_{n+1}(u; y) = \sum_{x_i \in ROI(y)} \delta(u - bin(I_{n+1}(x_i)))$$

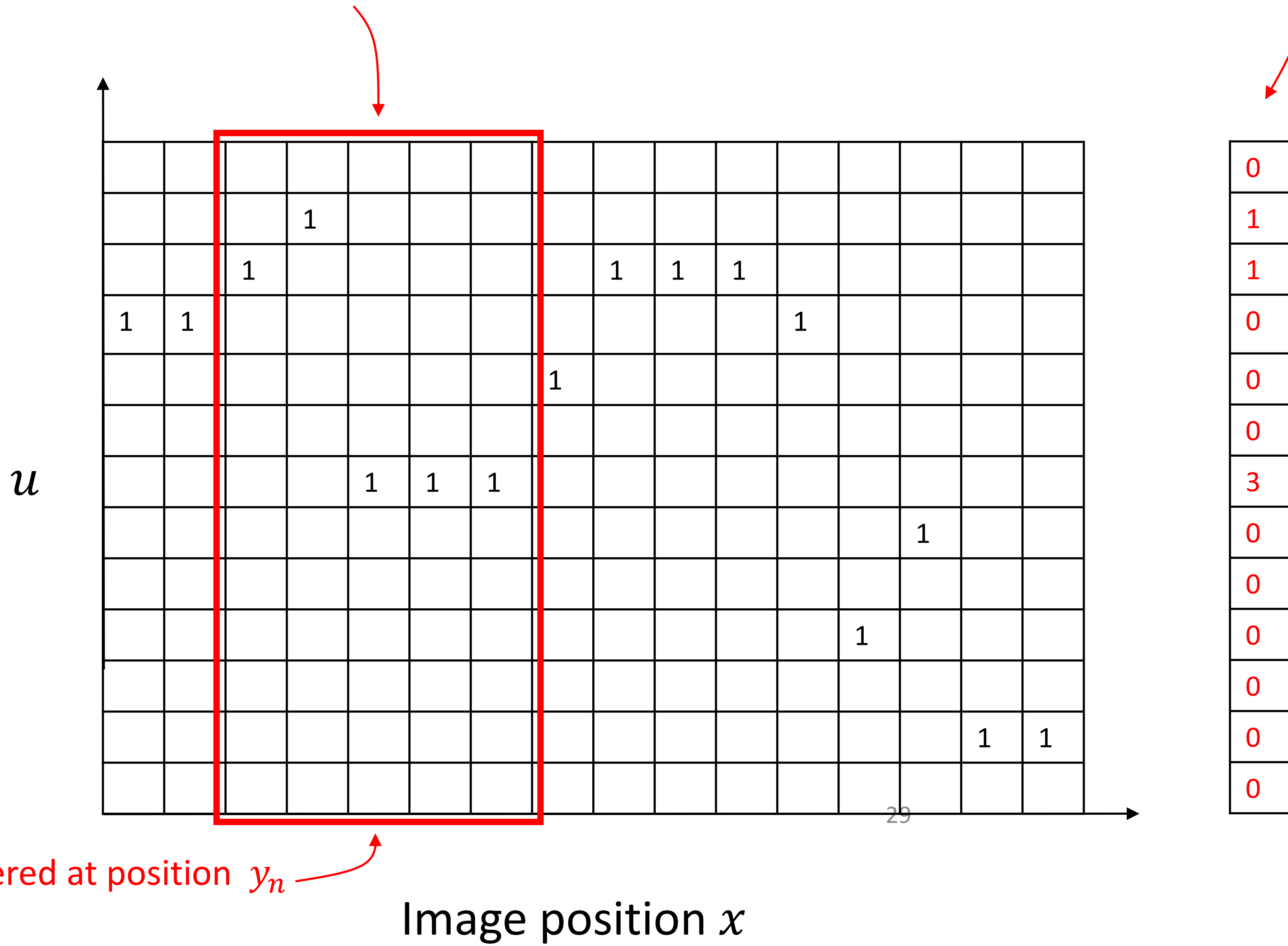
Let y_{n+1} be the position y in frame I_{n+1} that *minimizes the difference of the histograms:*

$$\sum_u |hist_{n+1}(u; y) - hist_n(u; y_n)|$$

To say it again in pictures….

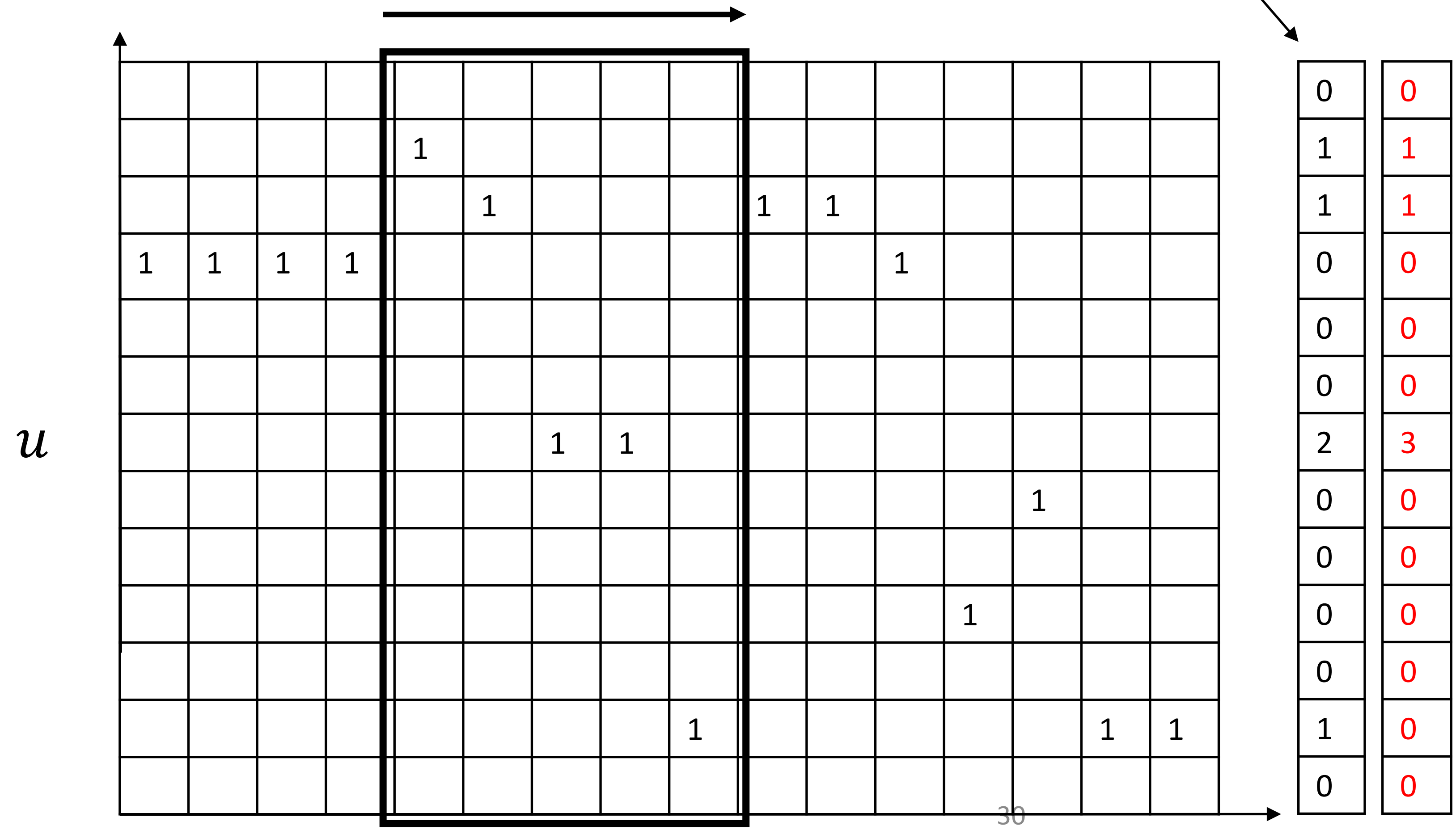
Suppose we have an ROI in image I_n centered at position y_n.

Histogram for the ROI in image I_n centered at y_n.



Find the position y_{n+1} in image I_{n+1} whose ROI histogram (right) is most similar to histogram from previous slide.

Histogram for the ROI in image I_{n+1} centered at y_{n+1}



ROI centered at position y_{n+1}

Image position x

Two problems with brute force tracking with ROI histogram comparison.

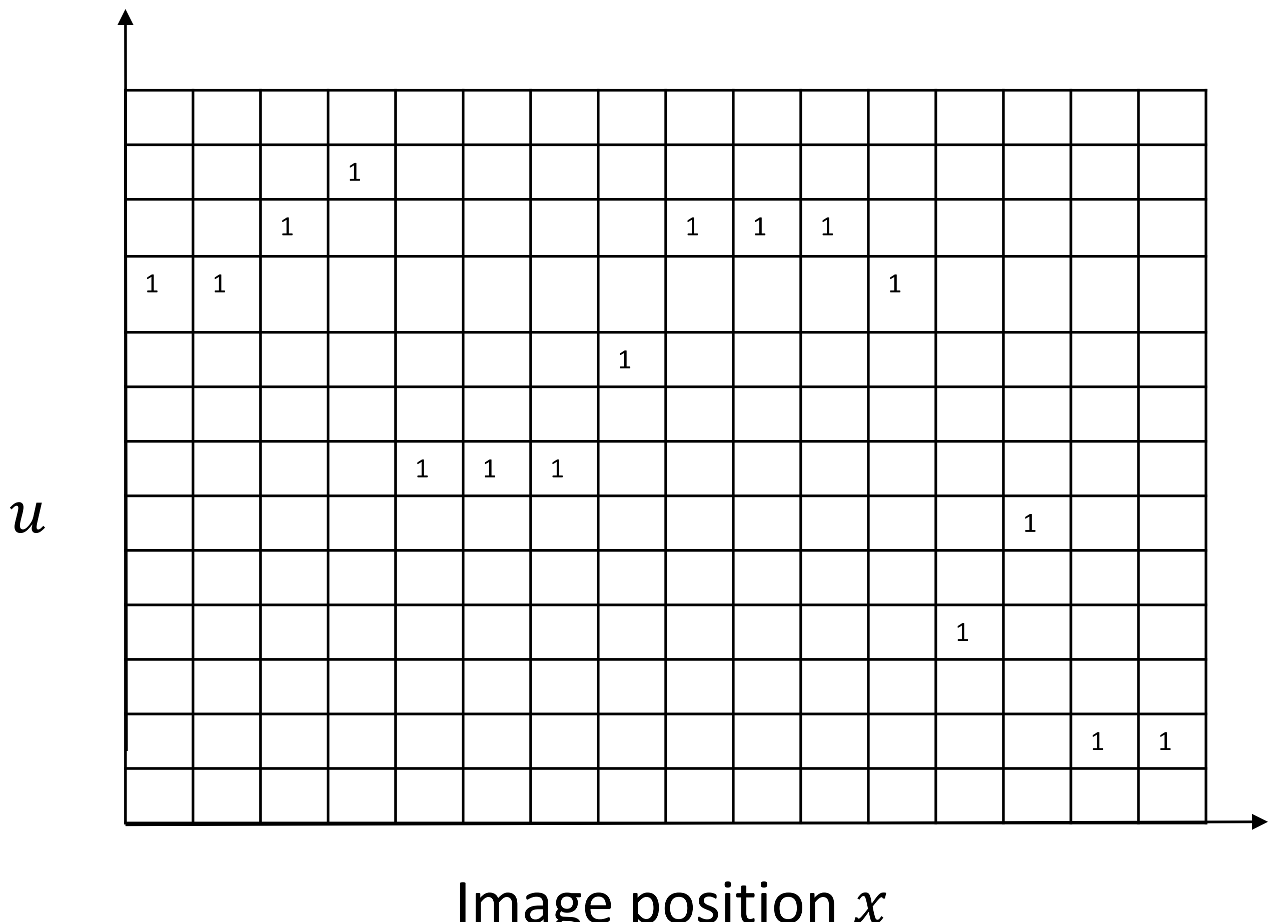
1) One should give more weight to the pixels near the center of the ROI. Somehow use a Gaussian window.

2) Bruce force search is inefficient.

We saw similar issues with Lucas-Kanade at start of lecture.

How to give more weight to the pixels near the center of the ROI ?
Define a symmetric kernel $K(x)$, typically a Gaussian.

Convolve each row u with kernel $K(x)$. See result on next slide.



$$p(u;y) = \sum_x K(y-x)\ \delta(u - bin(I(x)))$$

For this example, $K(x)$ = (.2, .6, .2).

u

		.2	.6	.2											
	.2	.6	.2				.2	.8	1	.8	.2				
.8	.8	.2								.2	.6	.2			
						.2	.6	.2							
			.2	.8	1	.8	.2								
												.2	.6	.2	
											.2	.6	.2		
.2															.2
													.2	.8	.8

Image position y

$$p(u;y) = \sum_x K(y-x)\ \delta(u - bin(I(x)))$$

I will refer to each column as a “weighted histogram”.

For this example, $K(x)$ = (.2, .6, .2).

u

		.2	.6	.2											
	.2	.6	.2				.2	.8	1	.8	.2				
.8	.8	.2								.2	.6	.2			
						.2	.6	.2							
			.2	.8	1	.8	.2								
												.2	.6	.2	
											.2	.6	.2		
.2															.2
													.2	.8	.8

Image position y

$$p(u; y) = \sum_x K(y - x)\ \delta(u - bin(I(x)))$$

One can show (see lecture notes) that the weighted histogram is a probability function for each image position y. That is, each column sums to 1.

u

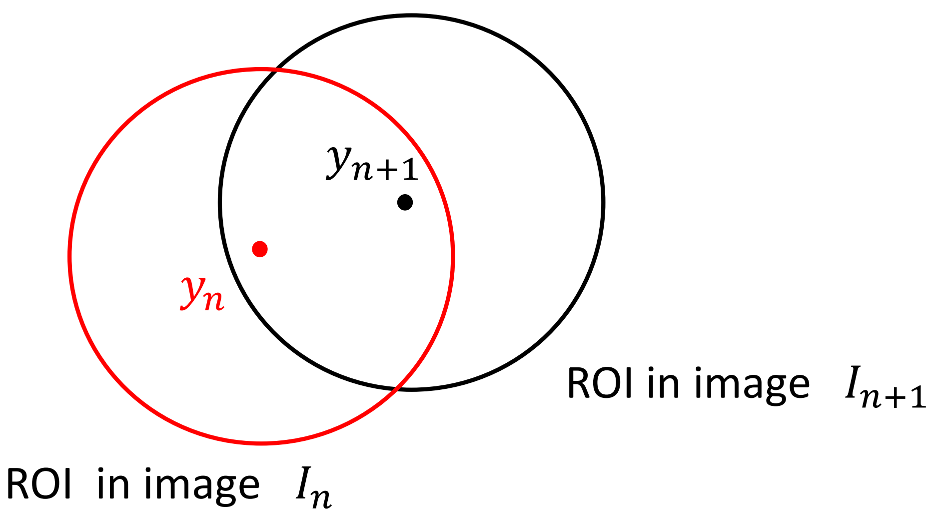
		.2	.6	.2											
	.2	.6	.2				.2	.8	1	.8	.2				
.8	.8	.2								.2	.6	.2			
						.2	.6	.2							
			.2	.8	1	.8	.2								
												.2	.6	.2	
											.2	.6	.2		
.2															.2
													.2	.8	.8

Image position y

Given position y_n centered at ROI in frame I_n,
find the nearby position y_{n+1} in frame I_{n+1} that
maximizes the similarity of the *weighted* ROI histograms.

(Note: we are neither blurring the image, nor blurring the ROI histograms.)

How do we define this similarity ?



How to define the similarity of two probability functions?

Let $p(u)$ and $q(u)$ be two probability functions.

The *Bhattacharya coefficient* is defined as:

$$BC(p, q) = \sum_u \sqrt{p(u)\, q(u)}$$

What is its value when $p(u) = q(u)$ for all u ?

What is its value when $p(u) * q(u)$ = 0 for all u ?

Note: Kaleem discussed the Bhattacharya *distance* in a lecture 9, which is closely related.

Weighted histogram for ROI centered at y_n in frame I_n.

$$p_n(u; y_n) = \sum_x K(y_n - x)\ \delta(u - bin(I_n(x)))$$

Weighted histogram for ROI centered at y in frame I_{n+1}.

$$p_{n+1}(u; y) = \sum_x K(y - x)\ \delta(u - bin(I_{n+1}(x)))$$

Let $\boldsymbol{y_{n+1}}$ be the position $\boldsymbol{y}$ in frame $\boldsymbol{I_{n+1}}$ that *maximizes the Bhattacharya coefficient.*

$$BC(p_n(u; y_n), p_{n+1}(u; y)) = \sum_u \sqrt{p_n(u; y_n)\ p_{n+1}(u; y)}$$

Two problems with brute force tracking with ROI histogram comparison.

1) One should give more weight to the pixels near the center of the ROI. Somehow use a Gaussian window.

2) Bruce force search is inefficient.

There is an algorithm called “mean shift” which can be used to solves this problem. (Details omitted.)

See Mubarak Shah's video if you are interested:
https://www.youtube.com/watch?v=M8B3RZVqgOo

Summary

- When objects have moving parts, registration methods from last 2 lectures don't work.
- Instead, for any ROI in one frame, find ROI in next frame whose histogram is most similar
- To give more weight to pixels in center of ROI, we use a weighted histogram (which can be defined as a probability function)