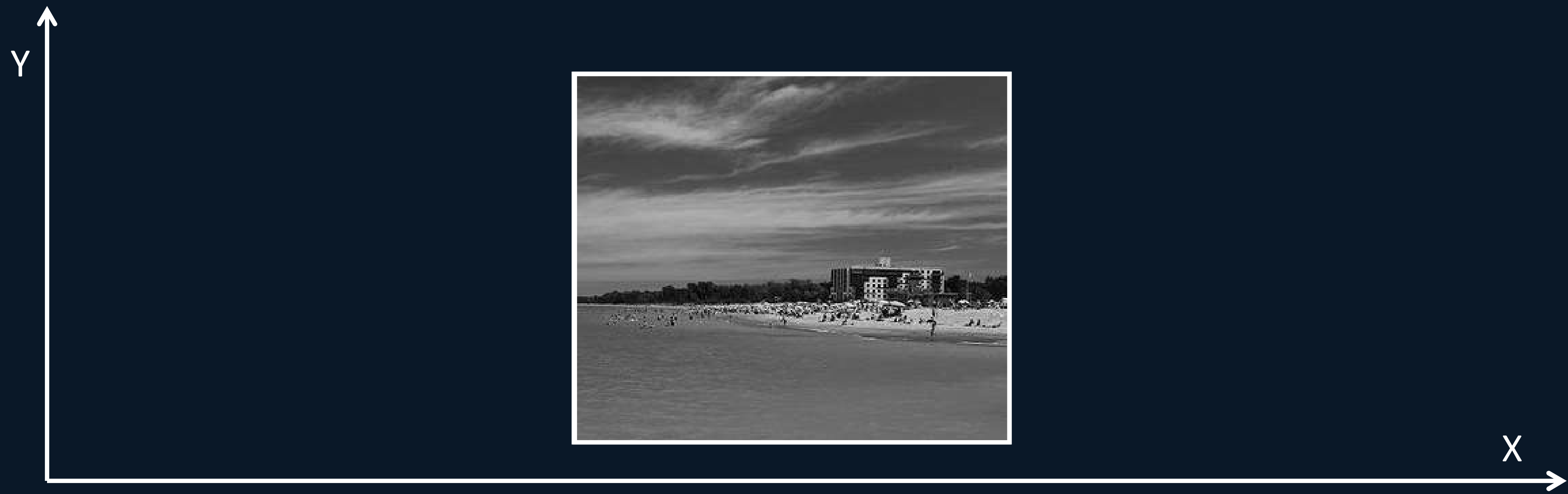
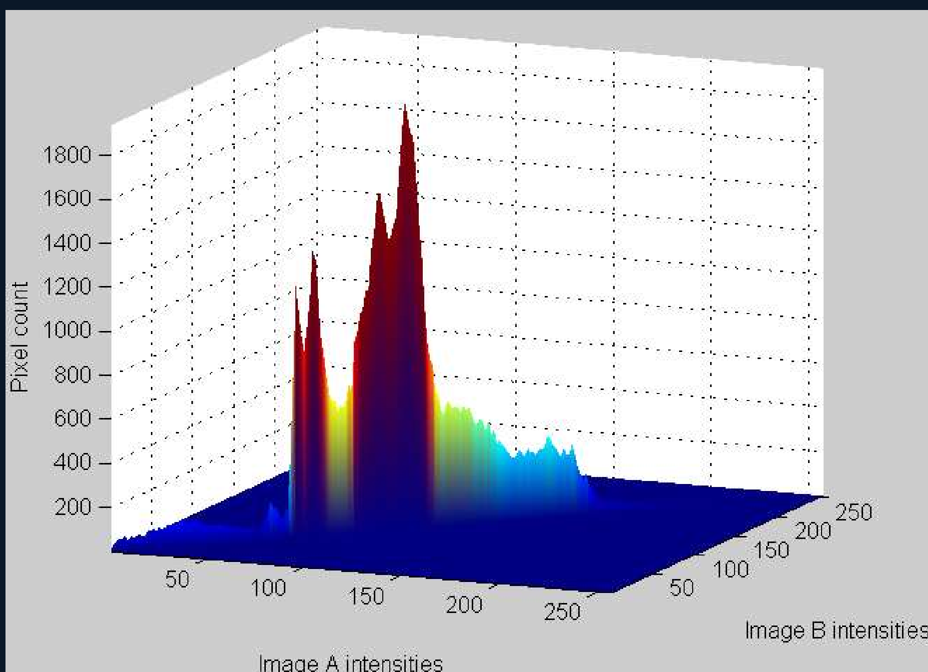
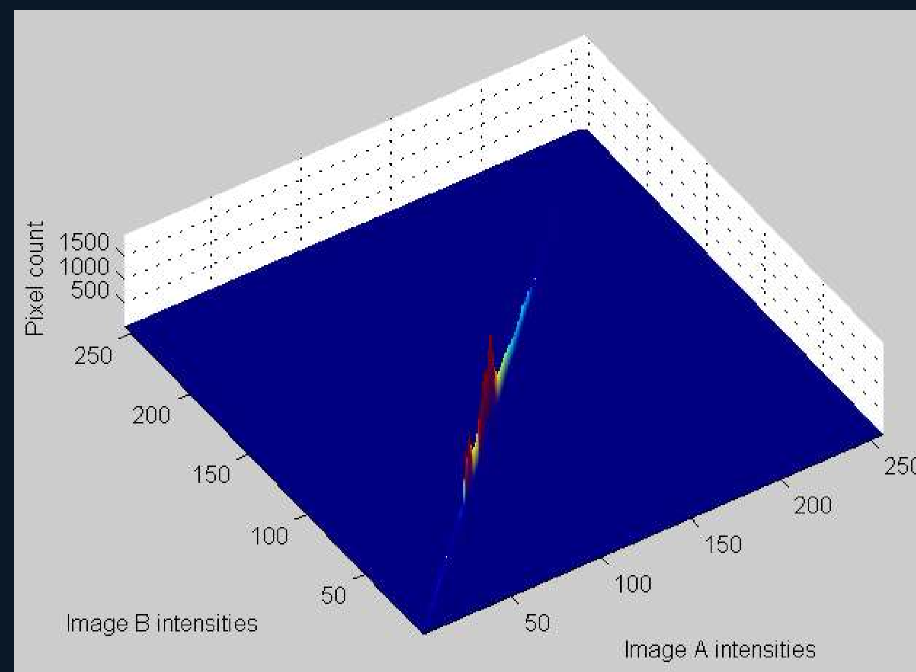
Let’s look at the accumulator table for real images.

These images have 256 different gray levels so to show this as a table we’d need a 256 x 256 table. This is easier to see as a 2D histogram, where large numbers are shown as large peaks, and small numbers as small peaks.



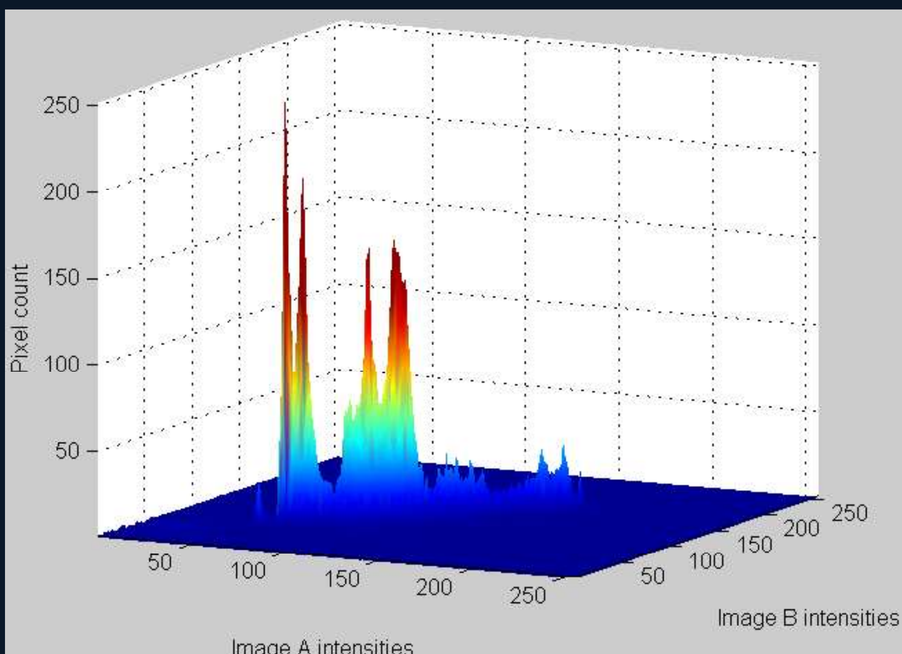
We begin with the situation where the images are perfectly aligned.

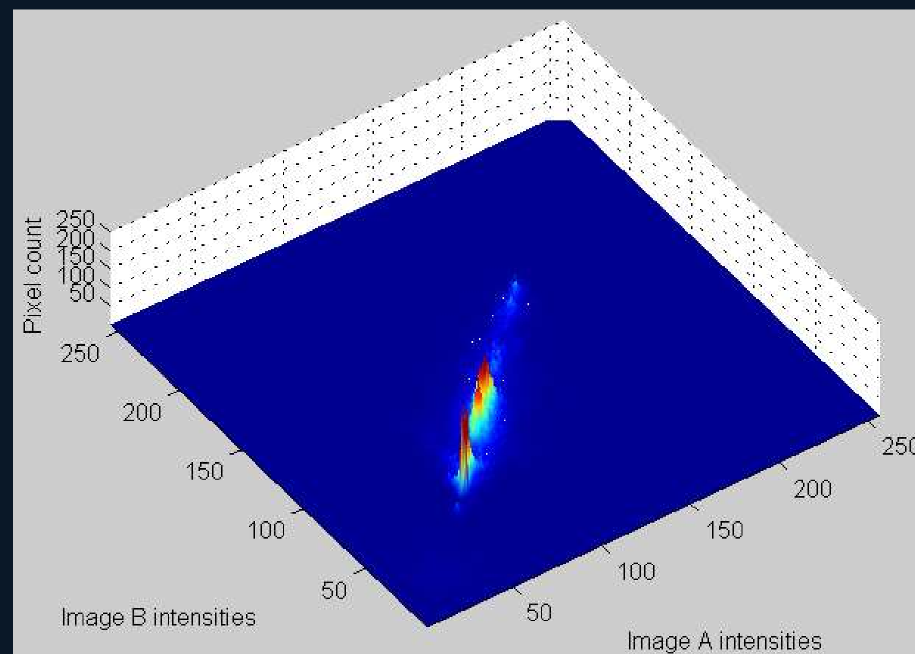




Two views of the accumulator – note the diagonal "ridge".

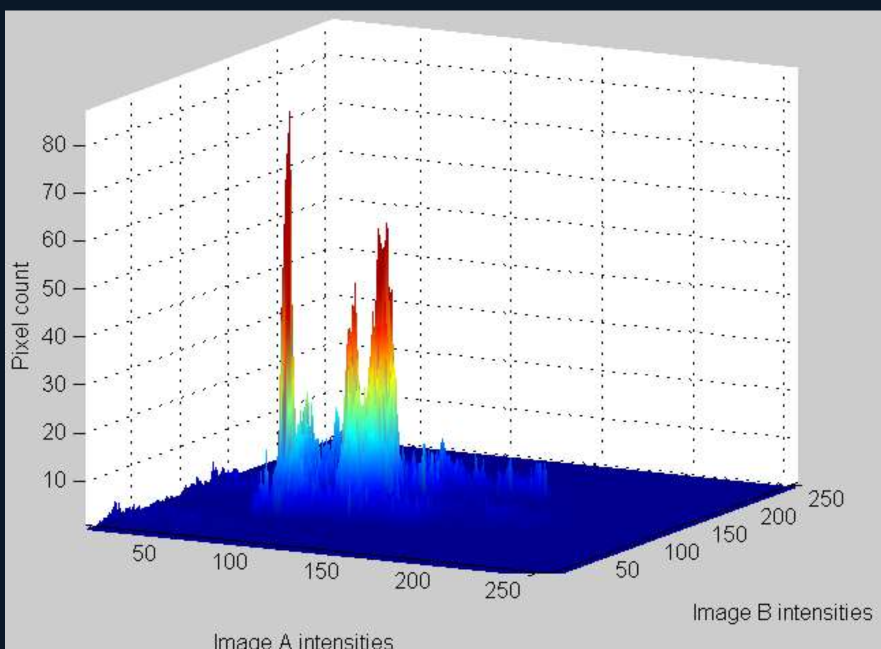
Now let’s move the images apart a bit and look at the accumulator table only in the region where they overlap.

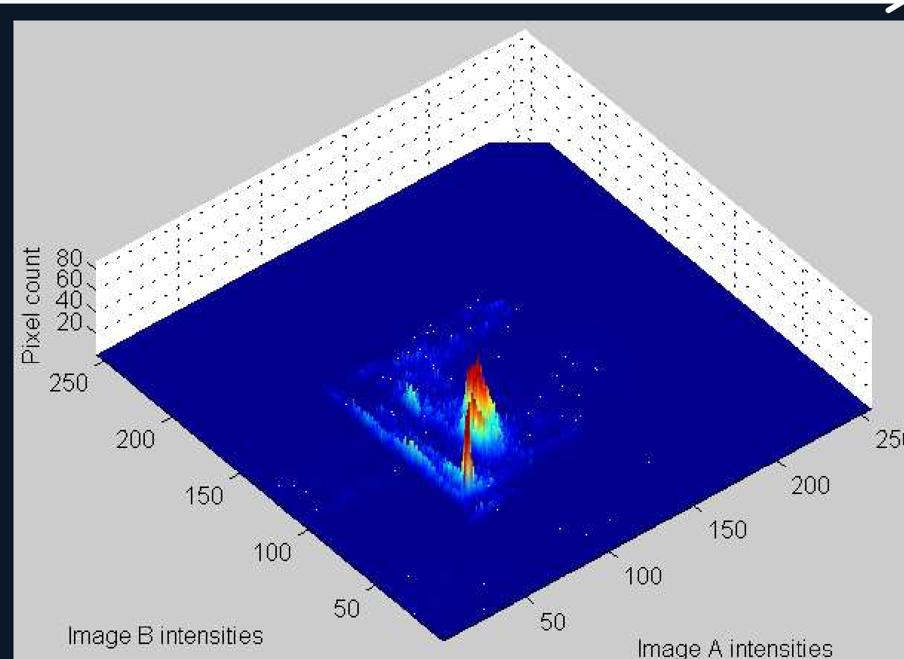




Notice that the ridge is starting to flatten/"smear".

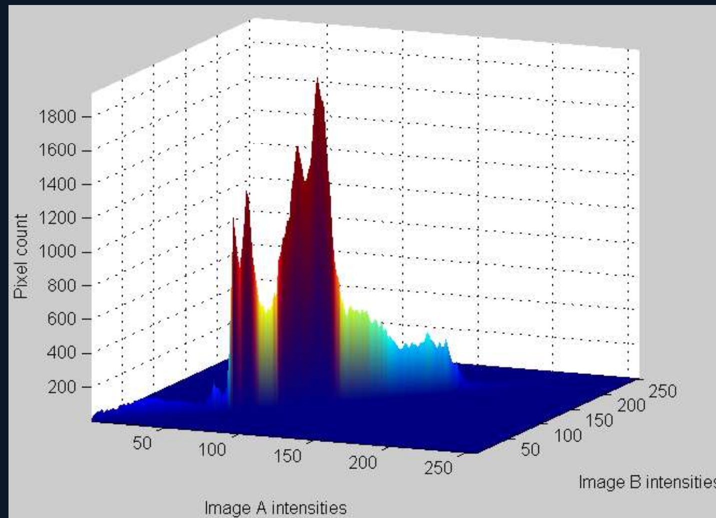
Now let's move the images apart so they only overlap by about 50% and look at the accumulator table only in the region where they overlap.

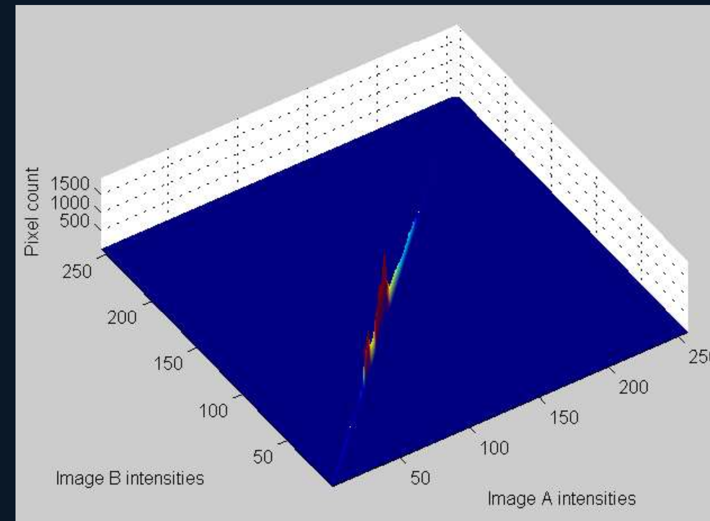




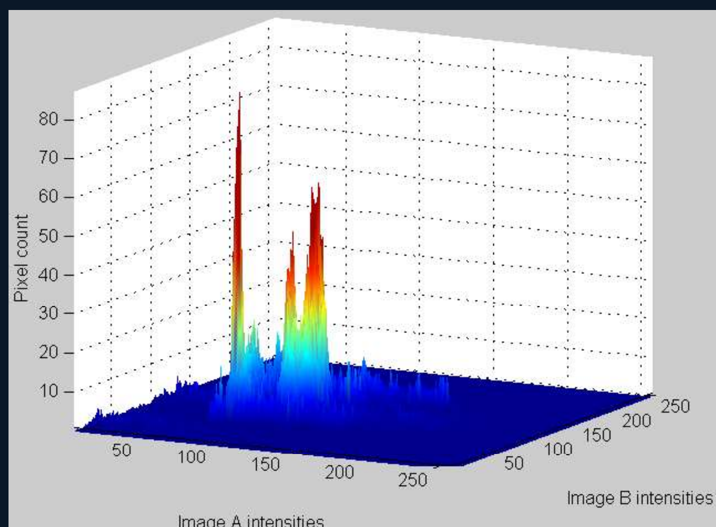
The accumulator table values have become much more distributed, and the heights of the peaks are lower.

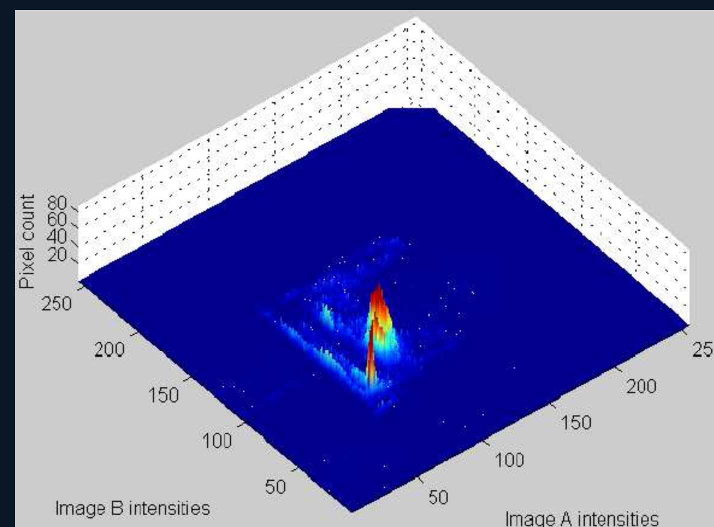
Good alignment





Bad alignment





Key observation:

Good alignment = histogram with narrow, tall peaks.

Bad alignment = flatter histogram.

Image similarity metric: Mean-squared error

$$MSE(I1, I2) = \frac{1}{N} \sum_{x} \sum_{y} \left(I1(x,y) - I2(x,y) \right)^2$$

N: # pixels/voxels
$I1(x,y)$: intensity of pixel in $I1$ at location (x,y)
$I2(x,y)$: intensity of pixel in $I2$ at location (x,y)

The ideal value of MSE is 0; thus gradient *descent* could be used to optimize this.

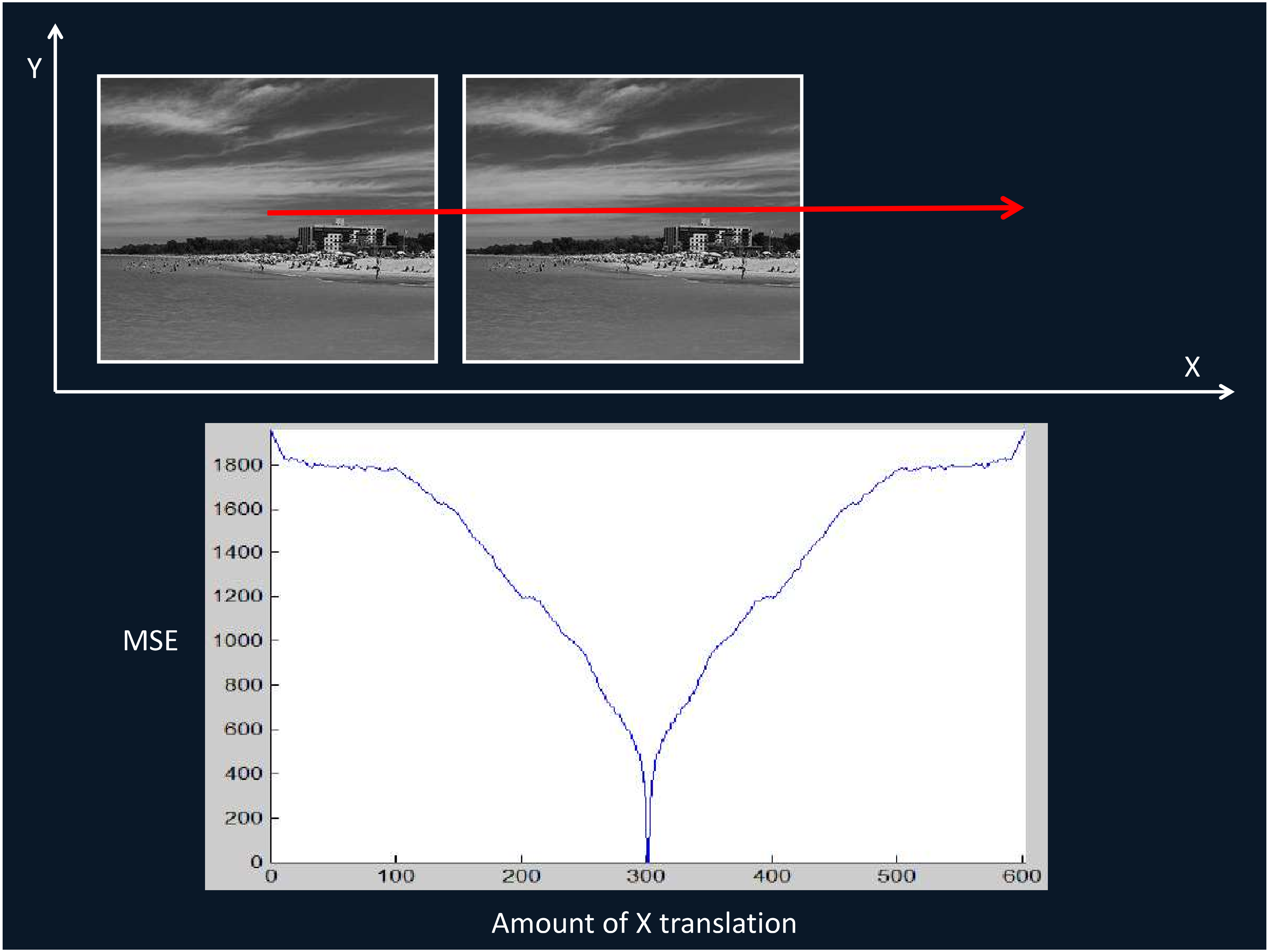
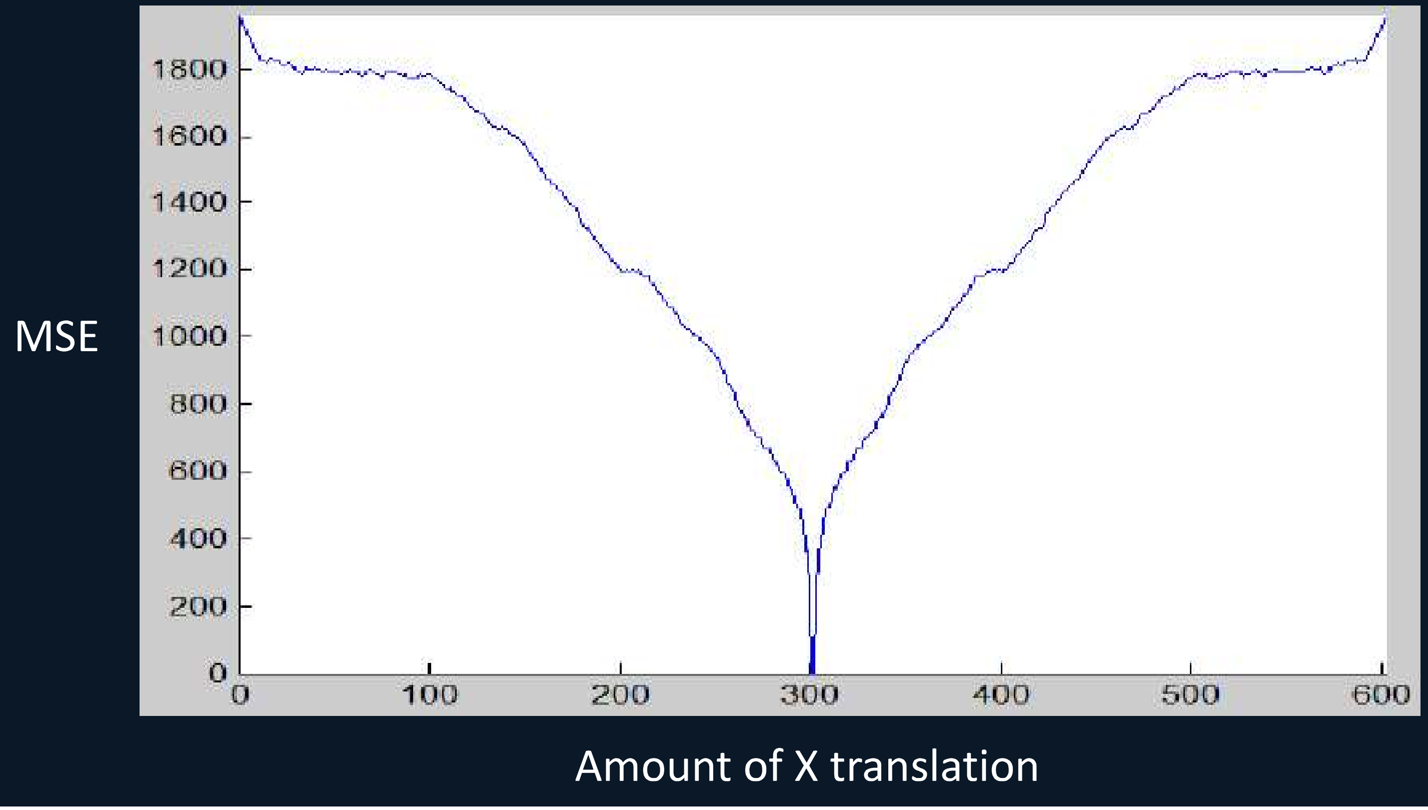

Y
X
MSE
1800
1600
1400
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
Amount of X translation

Image similarity metric: Mean-squared error

This is good behaviour for an image similarity metric. There are few local minima in which a gradient descent optimizer could get stuck, and there is a downward slope toward the correct answer even from far away.



In this example, the intensity of the moving image has been scaled by 150%. This could be analogous to two patient MRIs taken on different days with different scanners.

Let’s observe the behaviour of the MSE image similarity metric for this situation.

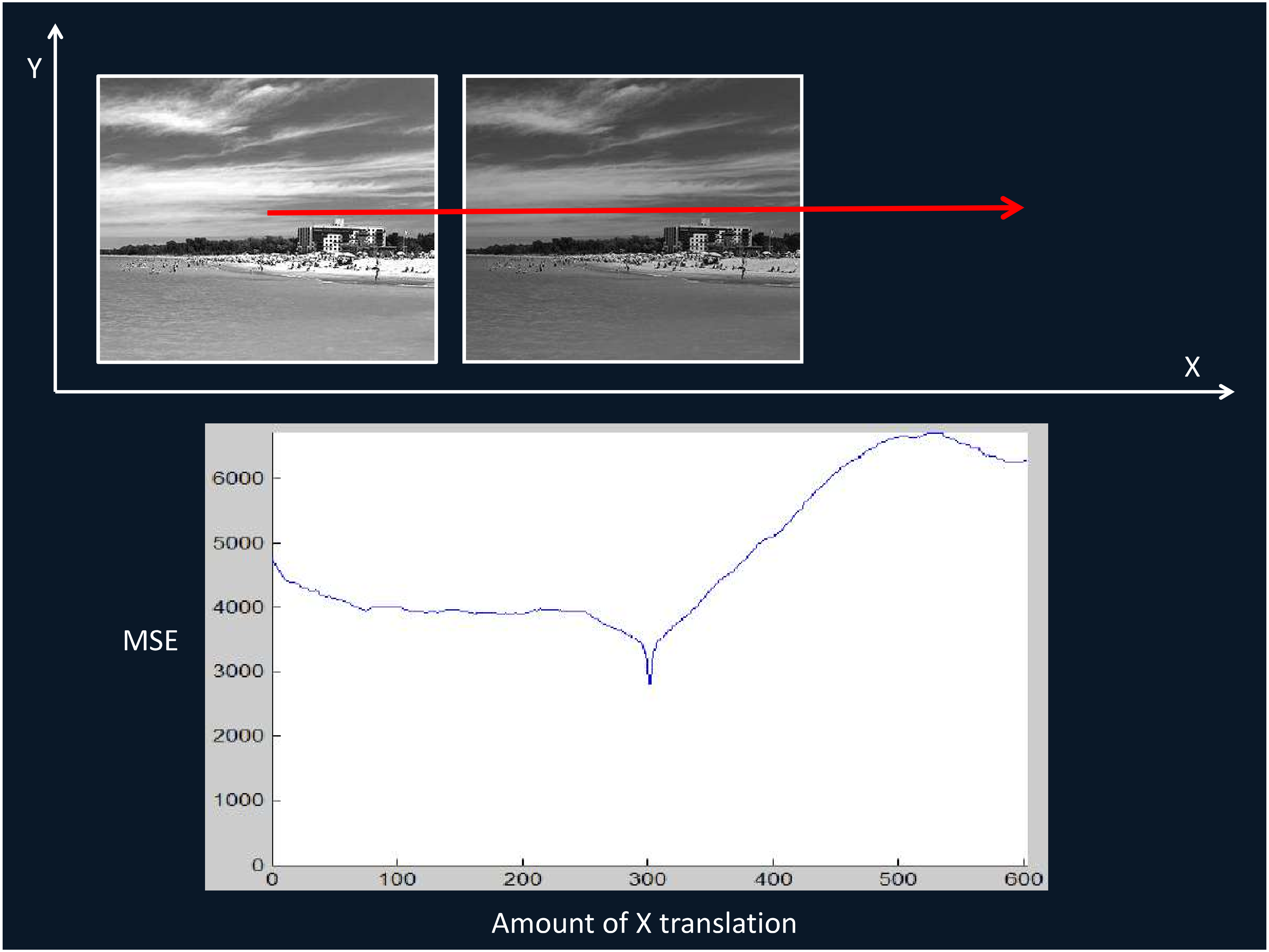
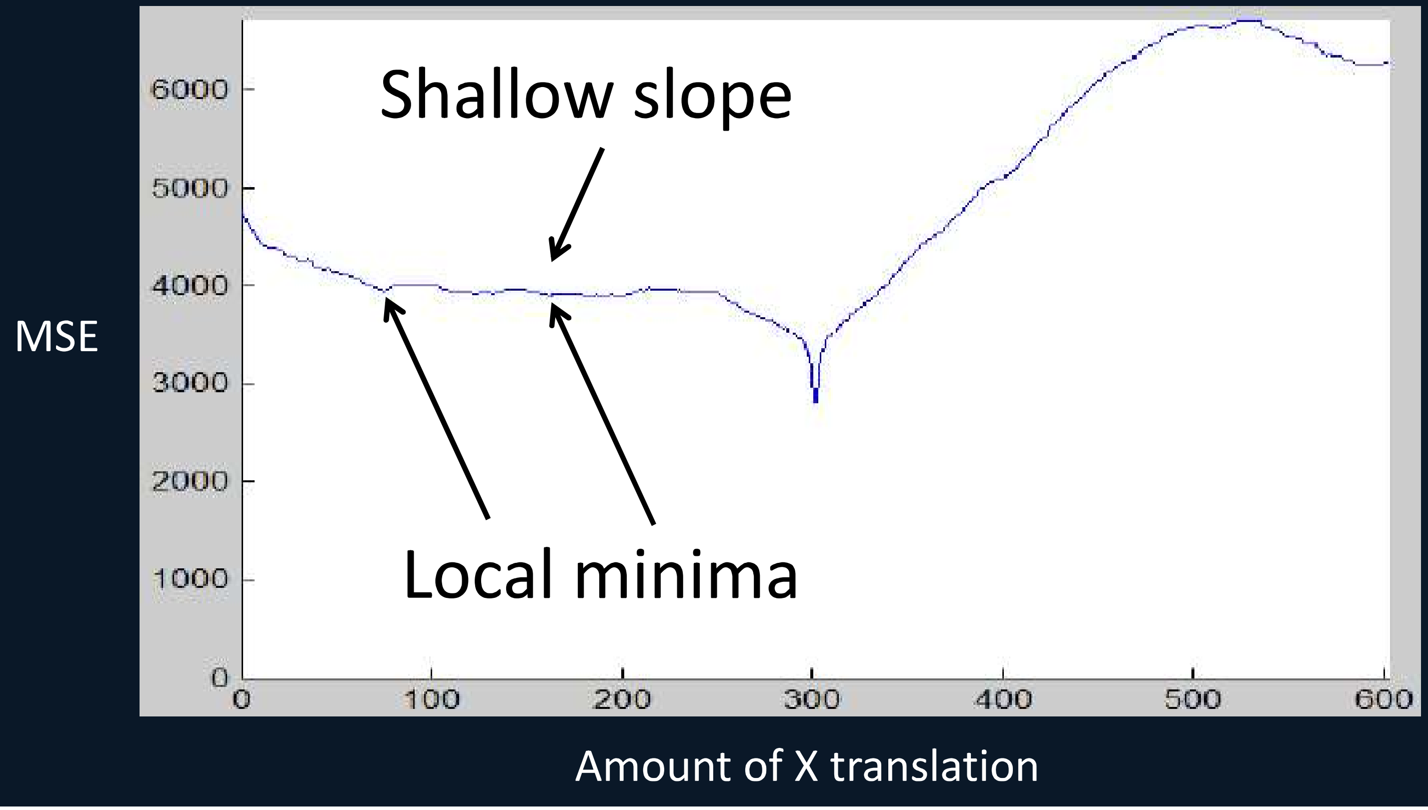
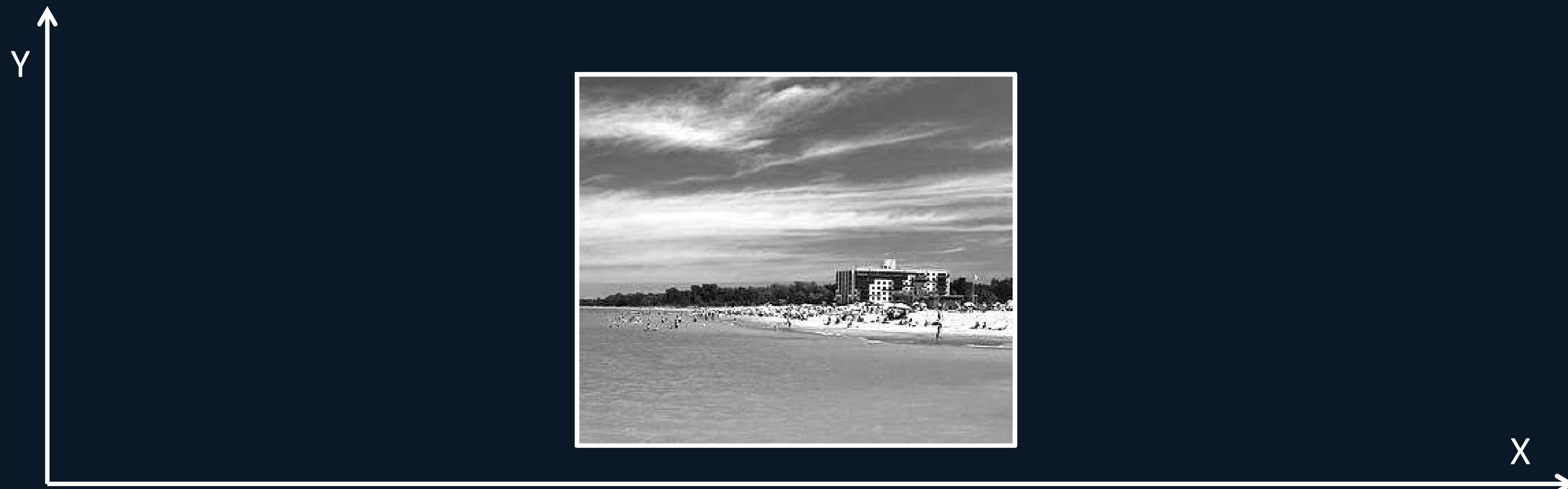

Y
X
MSE
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
Amount of X translation

Image similarity metric: Mean-squared error

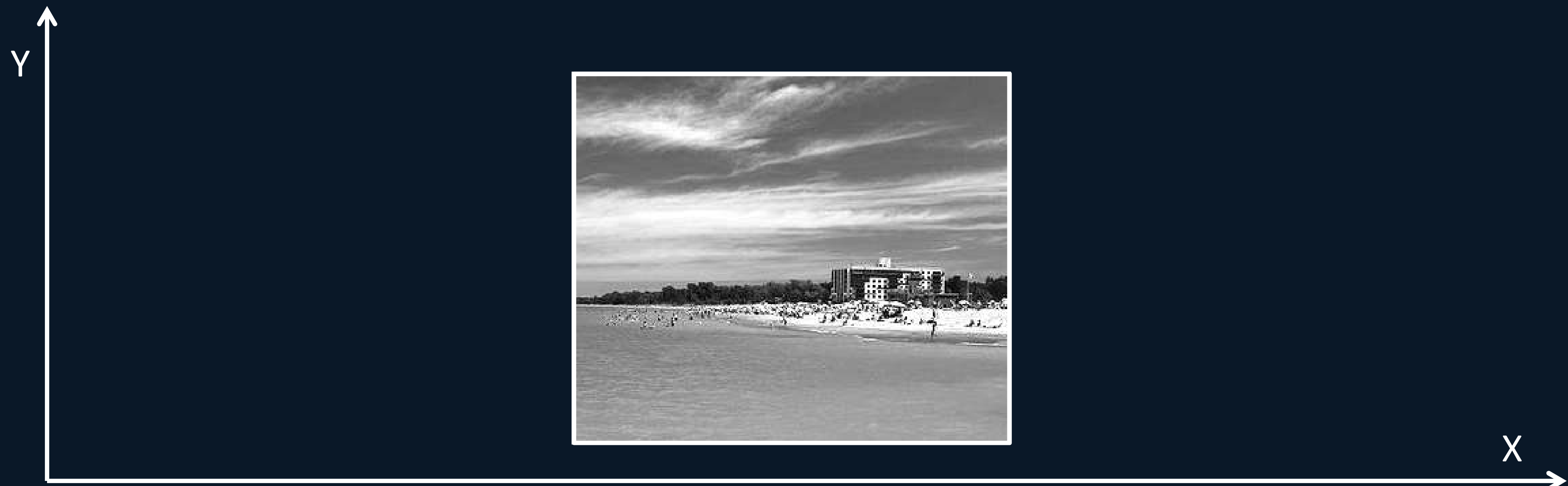
Bad behaviour: local minima, optimum value is not zero, large pleateau where we would prefer a downward slope toward the correct solution, slope depends on the initialization.

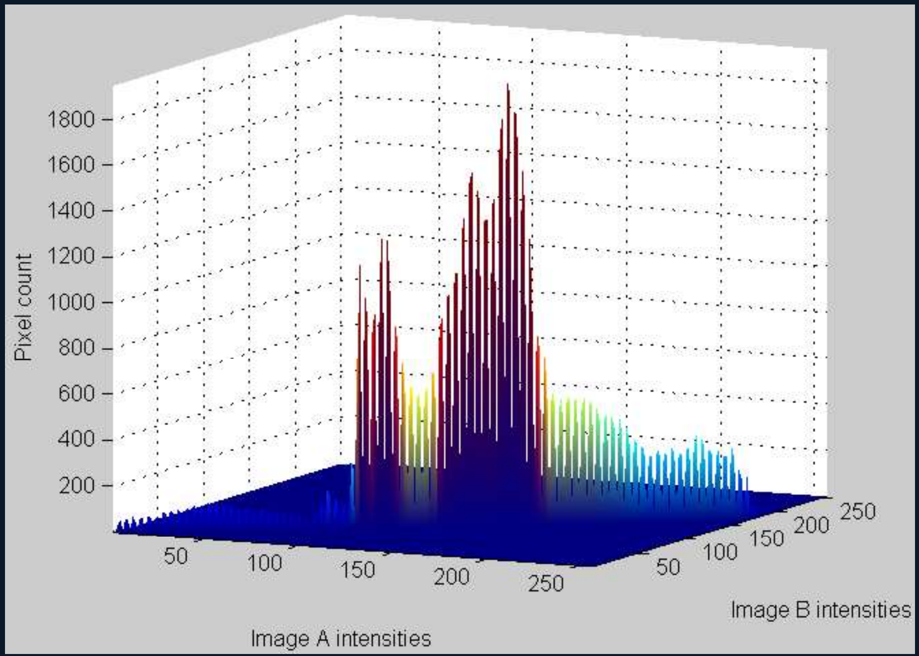


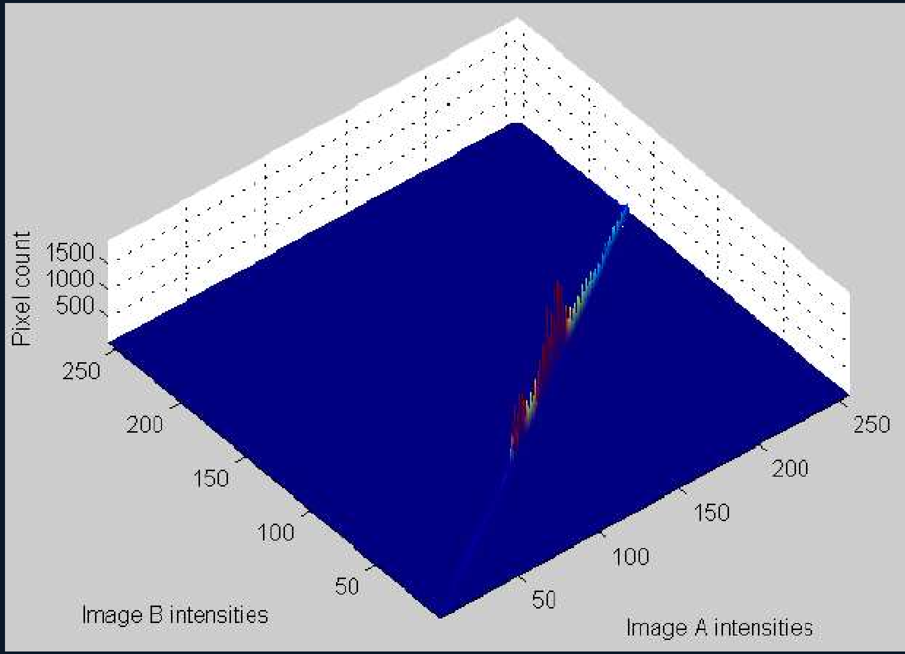


Let’s look again at the joint intensity histograms to understand the situation better.

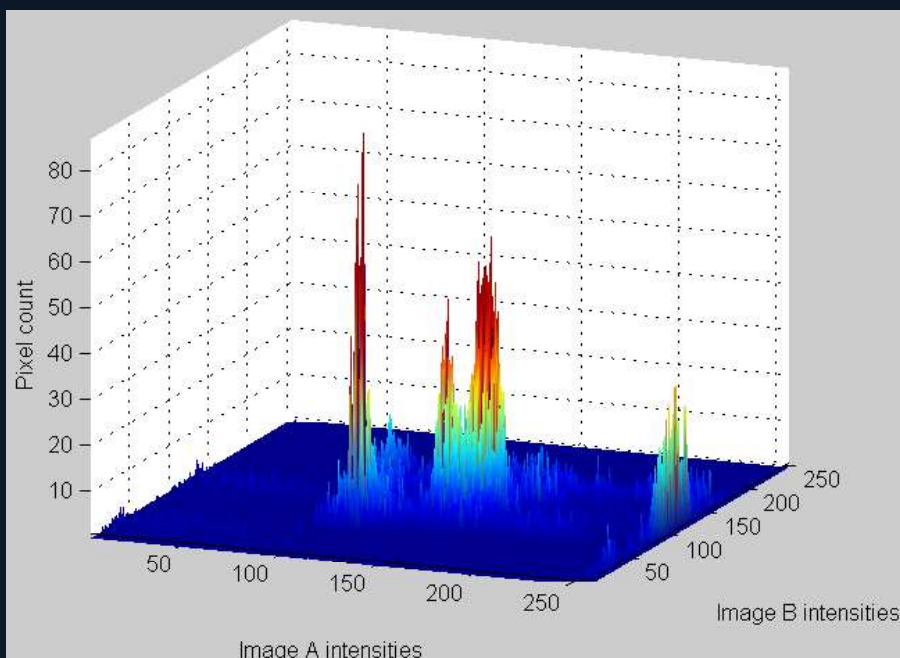
We begin with the situation where the images are perfectly aligned.

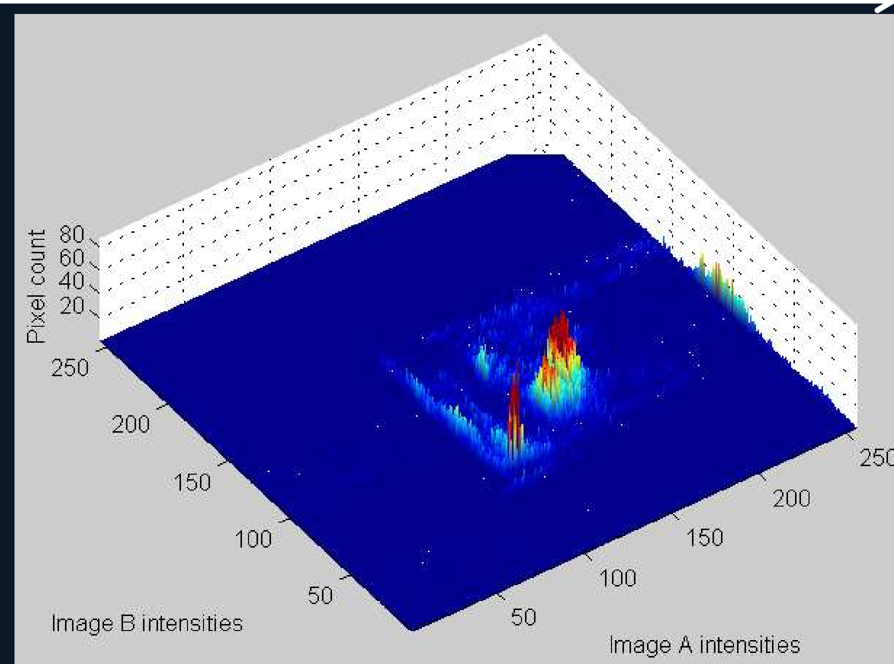




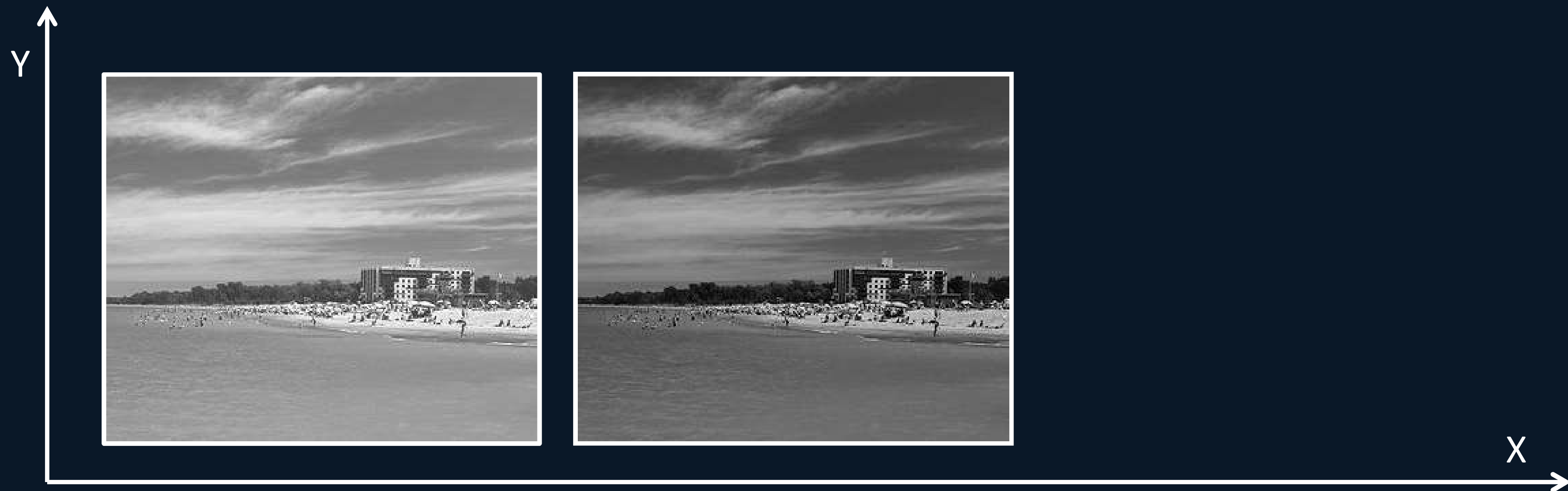


We still have a ridge, but its angle is different due to the relative scale of the intensity values.

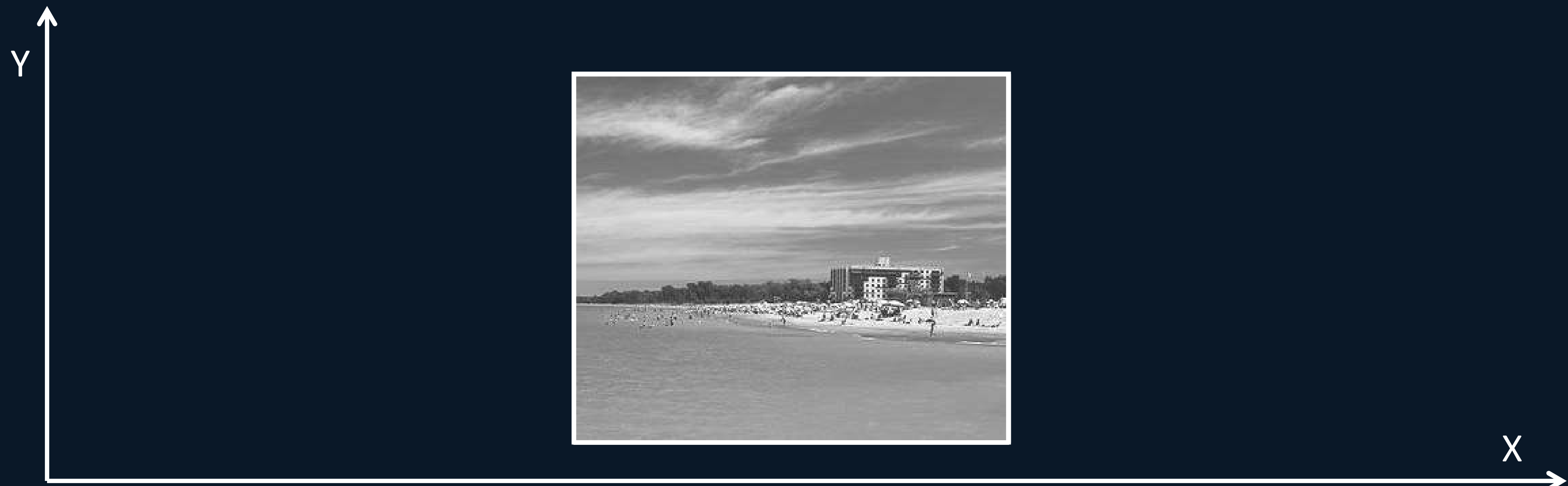


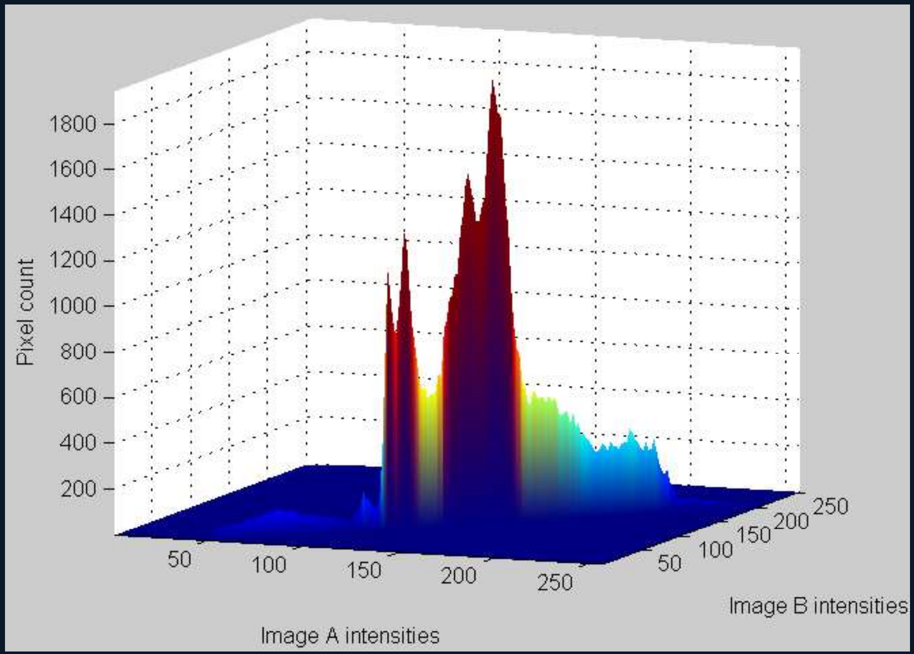


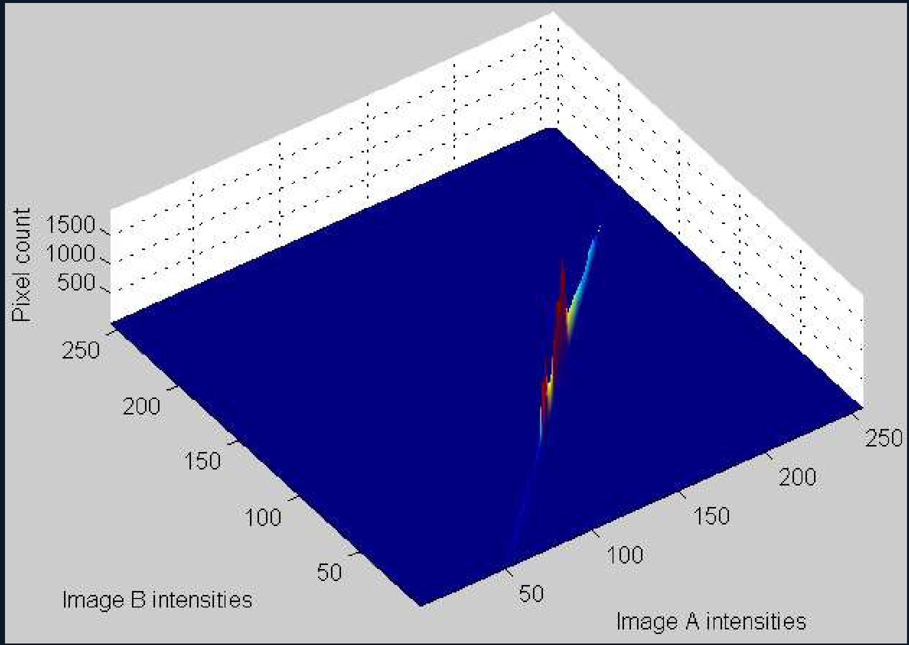
The histogram flattening effect still happens when we move the images out of alignment.



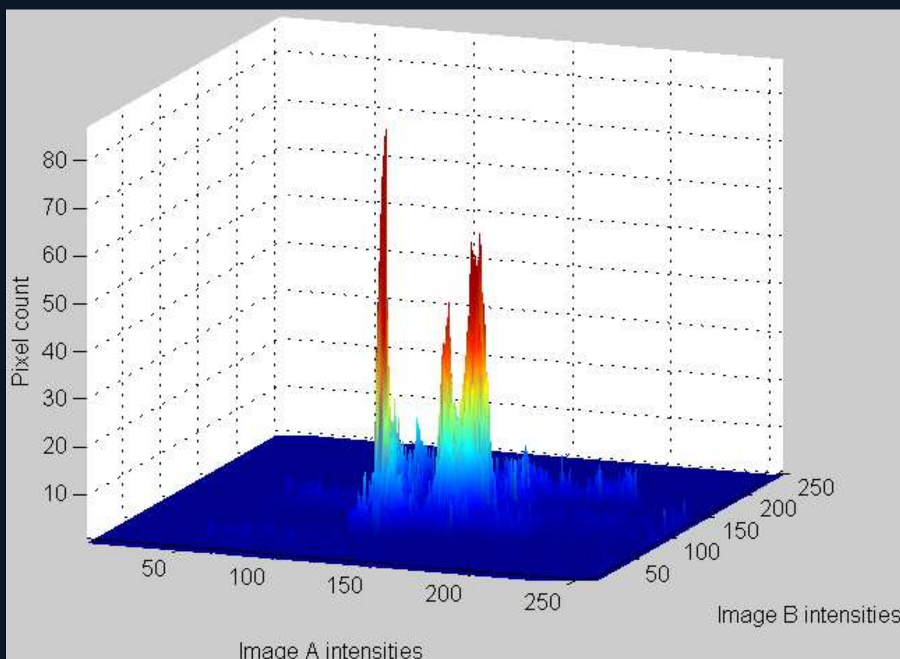
A similar situation can arise as in this example, where the moving image's intensities are shifted (upward, in this case, by 50). That is, at each pixel, the moving image's intensity is the fixed image's intensity, plus 50.

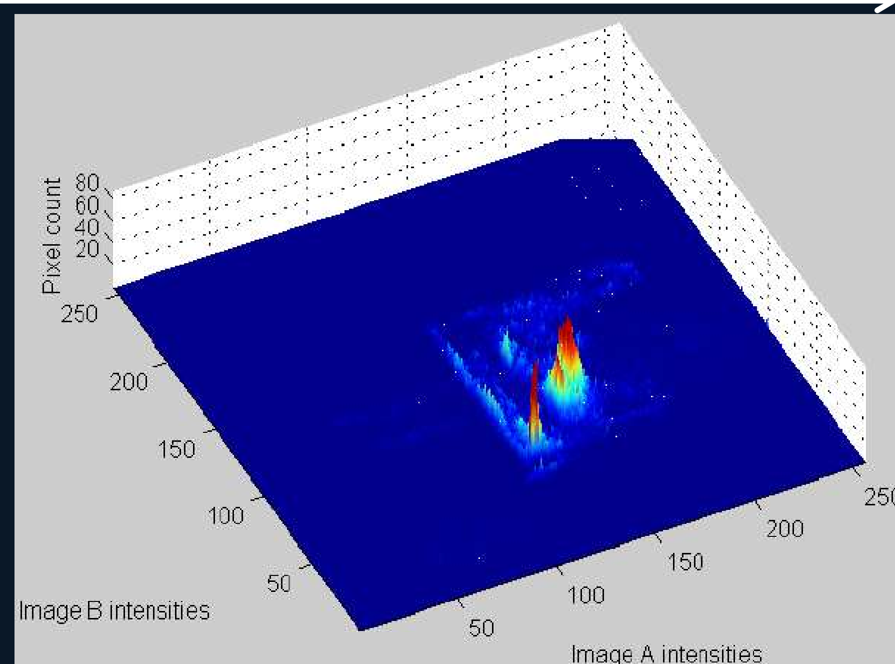






Now the ridge is shifted by 50 intensity values.

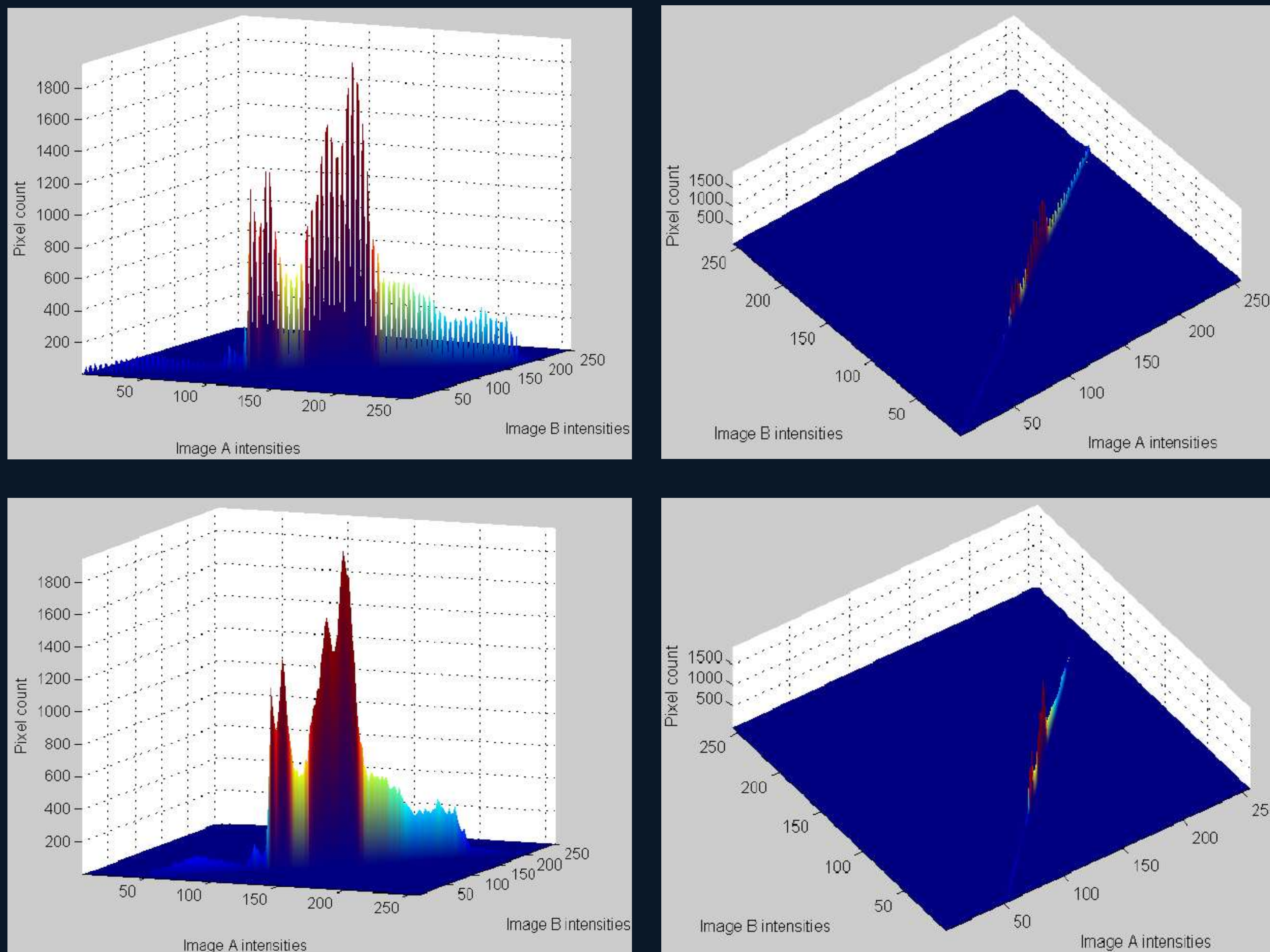




The histogram flattening effect still happens when we move the images out of alignment.

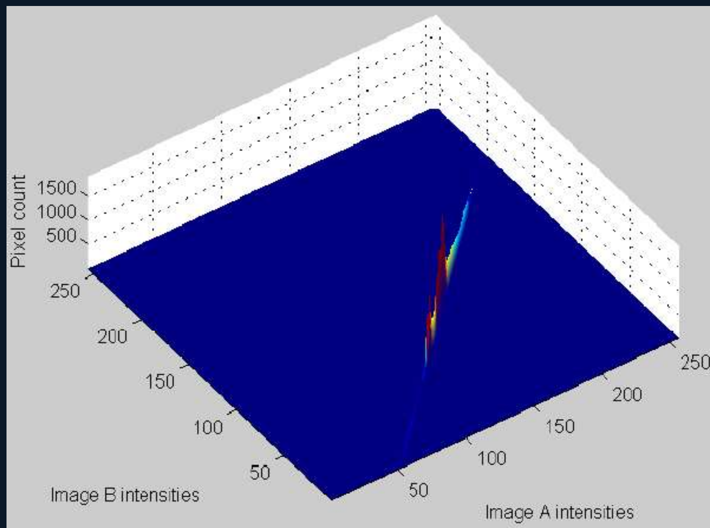
Good alignment (scaled intensities)

Good alignment (shifted intensities)

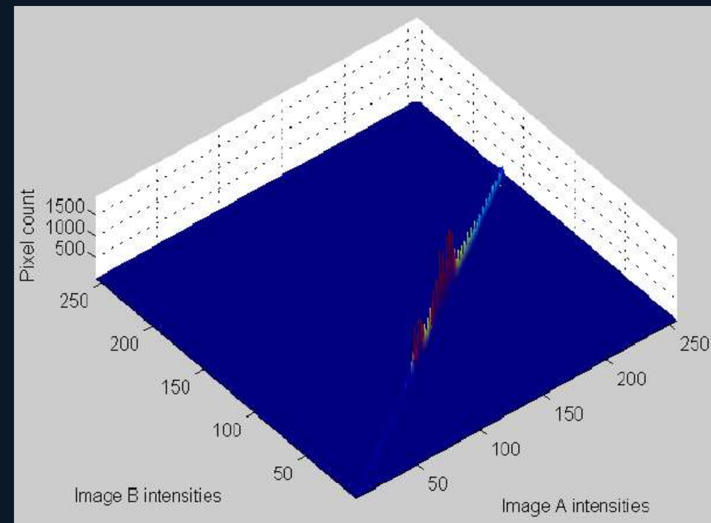


Good alignment still gives a narrow, linear ridge, but it’s not necessarily in the diagonal position due to intensity scale + shift.

Intensity shift



Intensity scale



How to calculate an image similarity metric that is insensitive to intensity scale and shift?

If we *normalize* the intensity values first, maybe we can then use the mean squared error metric.

Image similarity metric: Normalized mean-squared error

$$NMSE(I1, I2) = \frac{1}{N}\sum_{x}\sum_{y}\left(\left(\frac{I1(x,y) - \overline{I1}}{\|I1 - \overline{I1}\|_2}\right) - \left(\frac{I2(x,y) - \overline{I2}}{\|I2 - \overline{I2}\|_2}\right)\right)^2$$

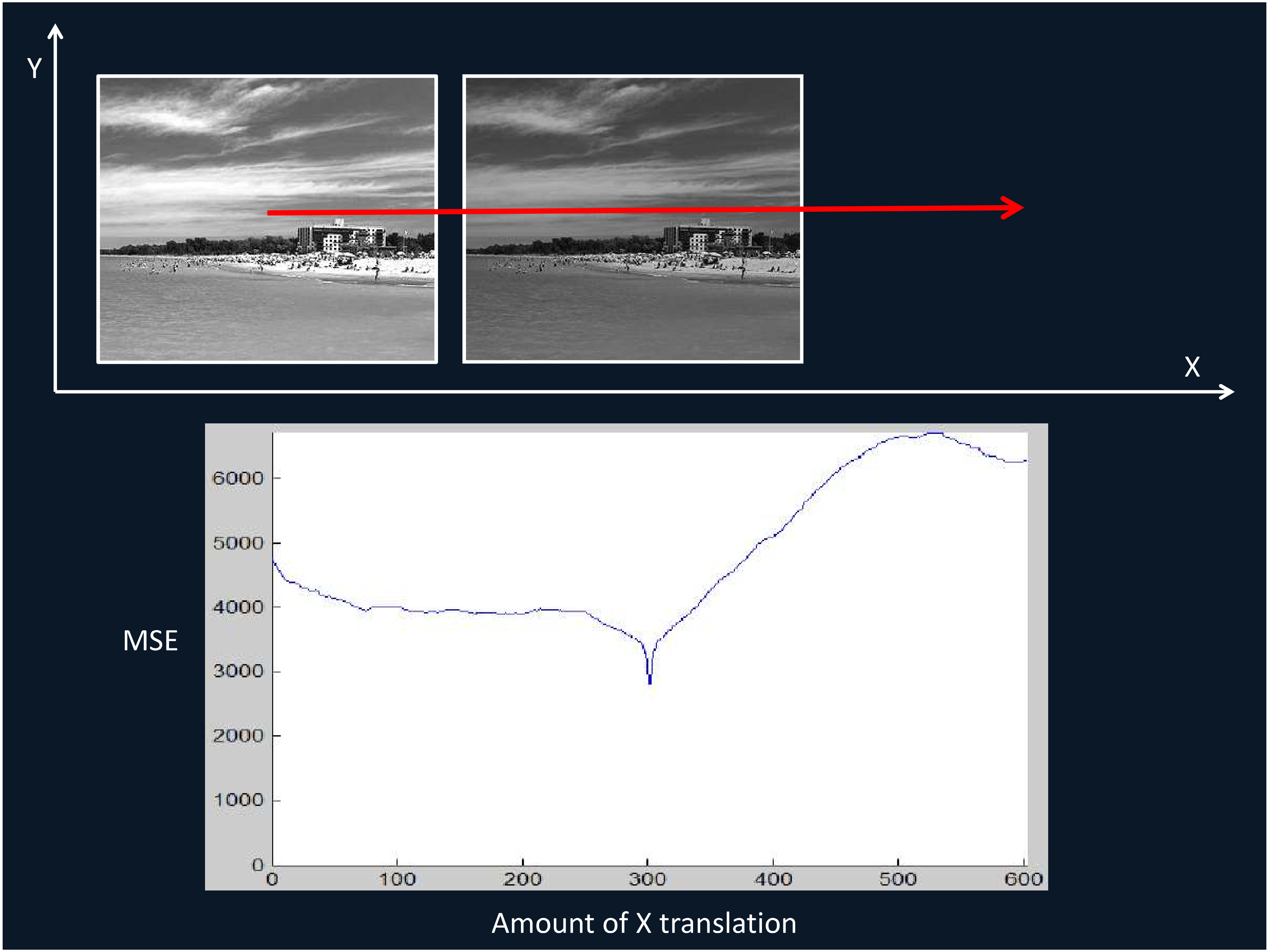
N: # pixels/voxels
$I1(x, y)$: intensity of pixel in $I1$ at location (x, y)
$\overline{I1}$: the mean intensity in $I1$
$\|I1\|_2$: L2-norm of $I1$.

Image similarity metric: Normalized mean-squared error

$$NMSE(I1, I2) = \frac{1}{N}\sum_{x}\sum_{y}\left(\left(\frac{I1(x,y) - \overline{I1}}{\|I1 - \overline{I1}\|_2}\right) - \left(\frac{I2(x,y) - \overline{I2}}{\|I2 - \overline{I2}\|_2}\right)\right)^2$$

Subtracting the mean normalizes for the intensity shift.

Dividing by the L2-norm normalizes for the intensity scale.


Y
X
MSE
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
Amount of X translation

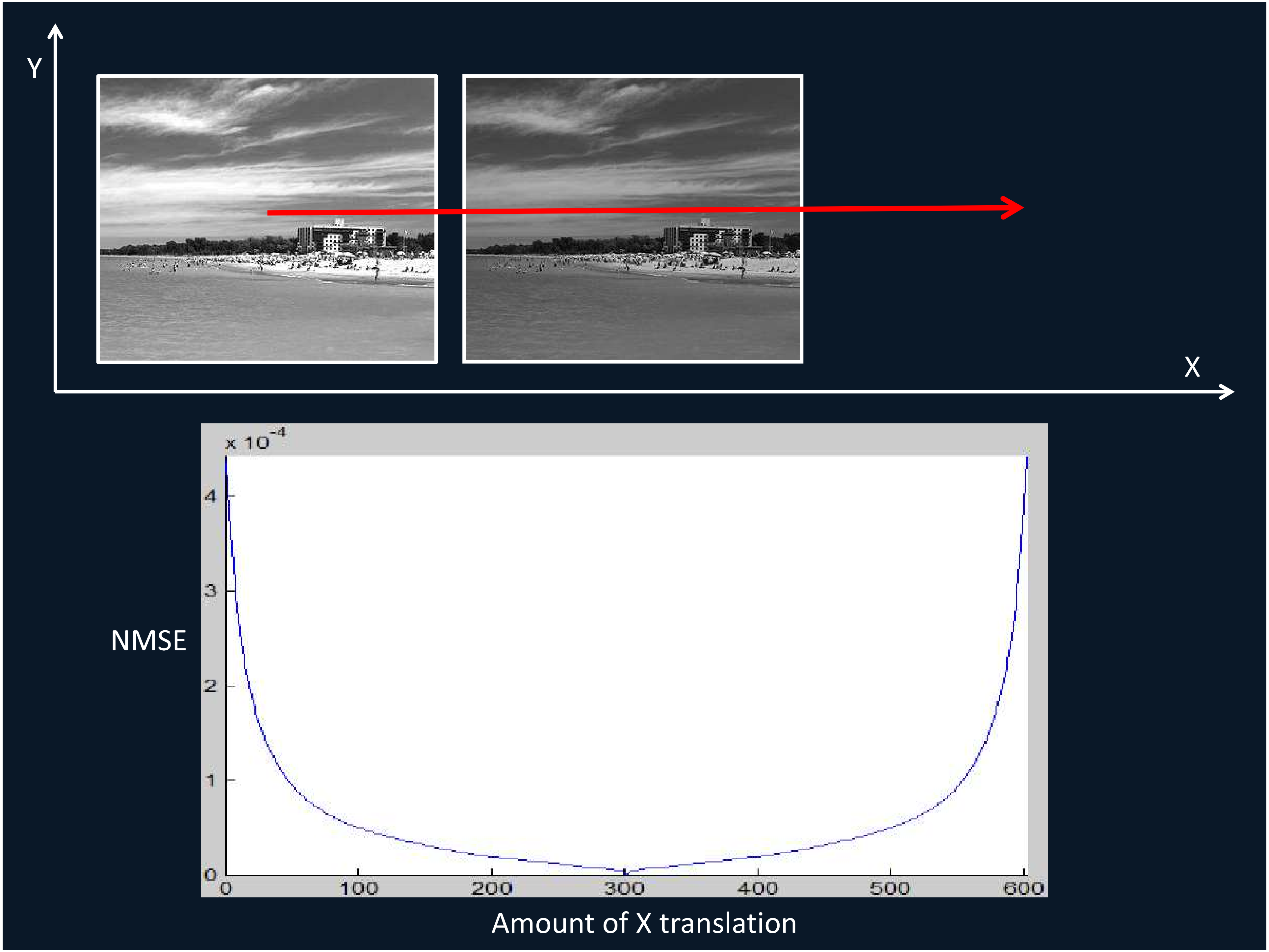

Y
X
x 10^-4
4
3
2
1
0
0
100
200
300
400
500
600
NMSE
Amount of X translation

Image similarity metric: Normalized mean-squared error

No local minima, consistent downslope toward correct answer. But, the gradient becomes small far from the answer; a gradient descent optimizer will converge, but slowly as its steps become small due to the small gradient.

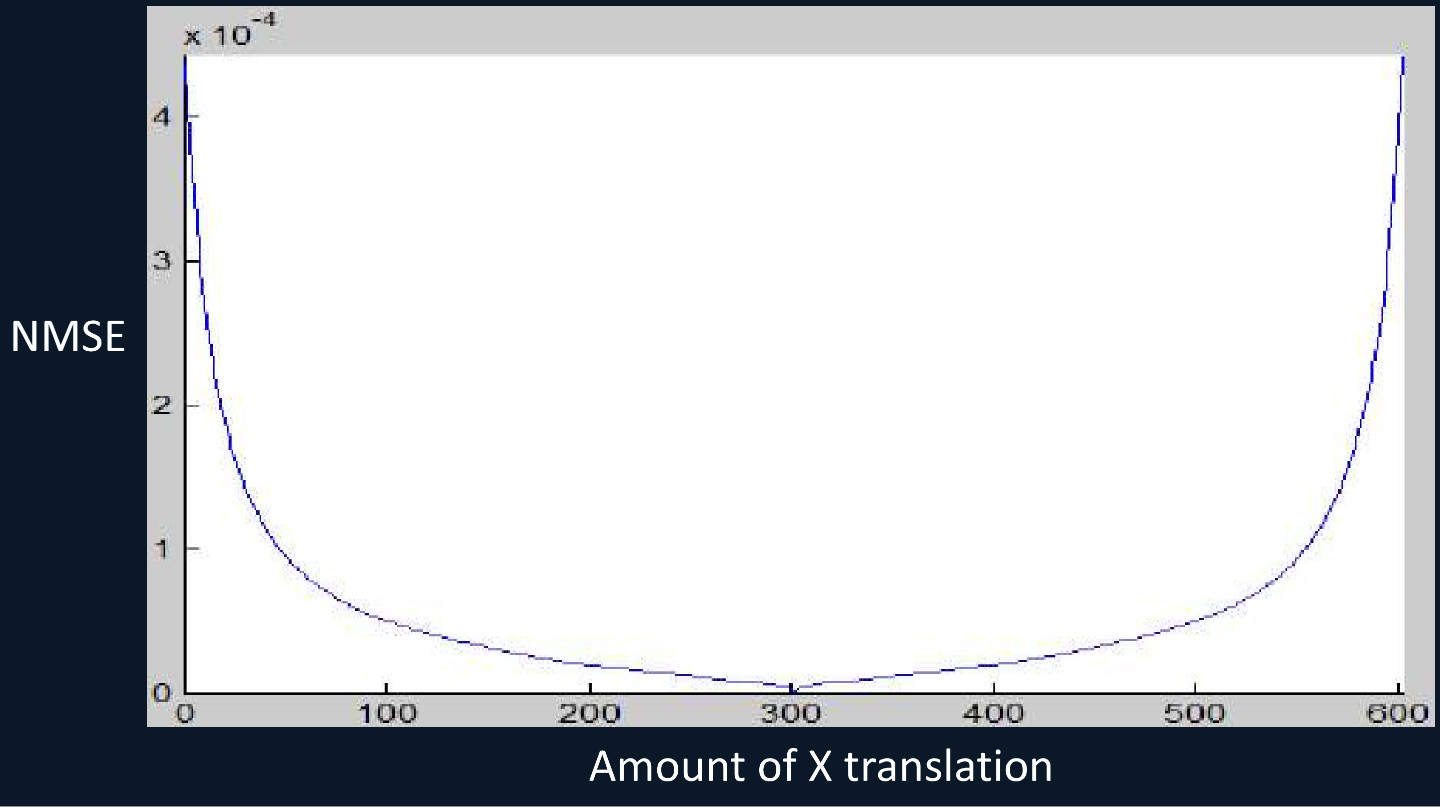


Image similarity metric: Normalized mean-squared error

$$NMSE(I1, I2) = \frac{1}{N}\sum_x \sum_y \left(\left(\frac{I1(x,y) - \overline{I1}}{\|I1 - \overline{I1}\|_2}\right) - \left(\frac{I2(x,y) - \overline{I2}}{\|I2 - \overline{I2}\|_2}\right)\right)^2$$

If you "flatten" these normalized images (by concatenating the rows together) you can see them as vectors.

These vectors are unit vectors due to the normalization.

Image similarity metric: Normalized mean-squared error

5	2
3	1

5	2	3	1

$$\left(\frac{I1(x,y) - \overline{I1}}{\|I1 - \overline{I1}\|_2}\right) \quad \left(\frac{I2(x,y) - \overline{I2}}{\|I2 - \overline{I2}\|_2}\right)$$

If you "flatten" these normalized images (by concatenating the rows together) you can see them as vectors.

These vectors are unit vectors due to the normalization.

Image similarity metric: Normalized cross correlation

$$NCC(I1, I2) = \left(\frac{I1(x,y) - \bar{I1}}{\|I1 - \bar{I1}\|_2}\right) \cdot \left(\frac{I2(x,y) - \bar{I2}}{\|I2 - \bar{I2}\|_2}\right)$$

We can take the dot product of these unit vectors, which yields the cosine of the angle between the vectors.

When the vectors are the same, the angle is zero, and the cosine of the angle is 1.

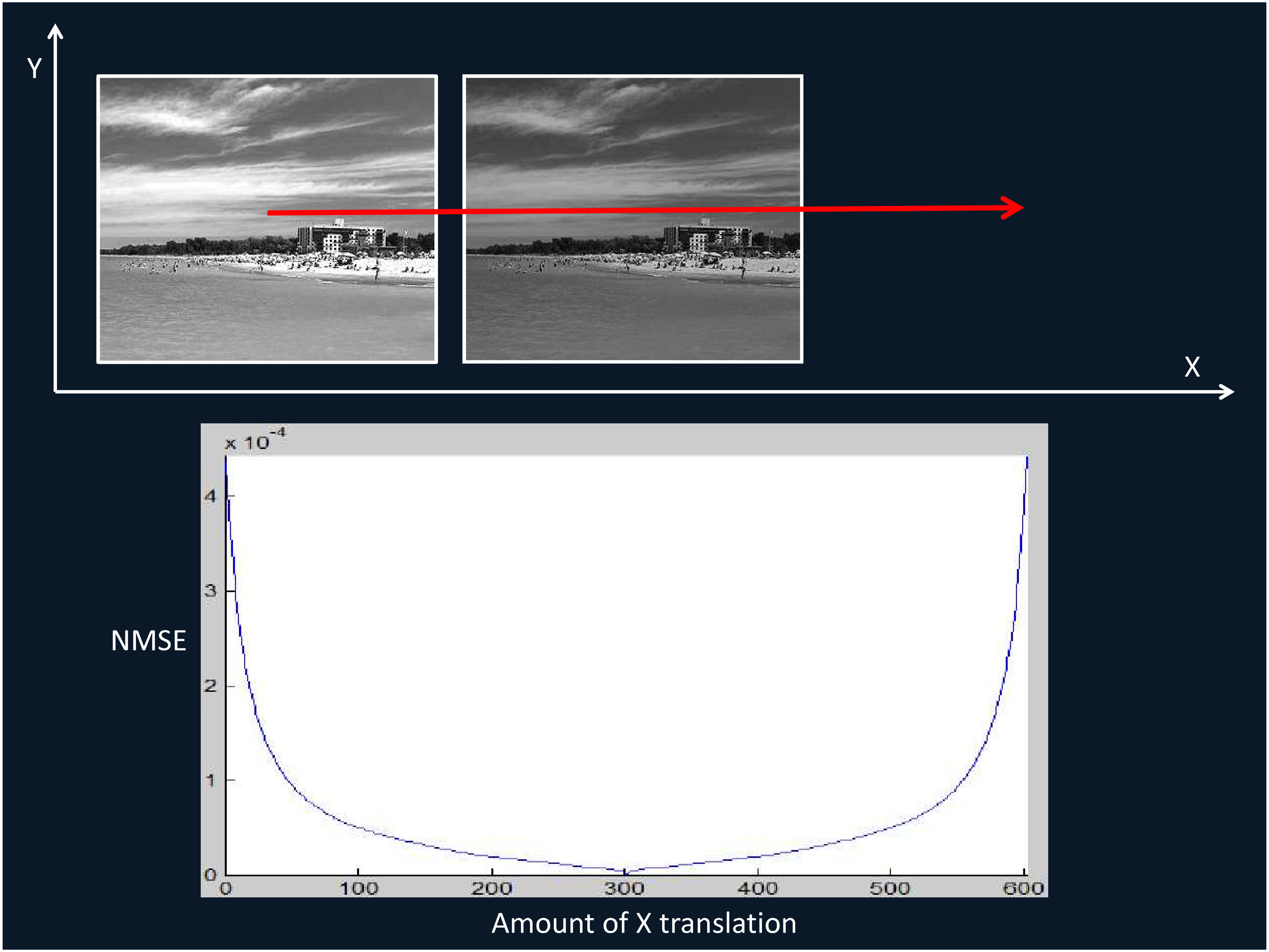
This is the *normalized cross correlation* image similarity metric and it has an ideal value of 1.

Image similarity metric: Normalized cross correlation

$$NCC(I1, I2) = \frac{1}{N-1} \sum_{x} \sum_{y} \frac{(I1(x,y) - \overline{I1})(I2(x,y) - \overline{I2})}{\sigma_{I1}\sigma_{I2}}$$

The dot product is a helpful way to think about this metric, but it is usually written as above.

σ_{I1}: The standard deviation of the intensities in I1.
σ_{I2}: The standard deviation of the intensities in I2.


Y
X
x 10^-4
4
3
2
1
0
0
100
200
300
400
500
600
NMSE
Amount of X translation

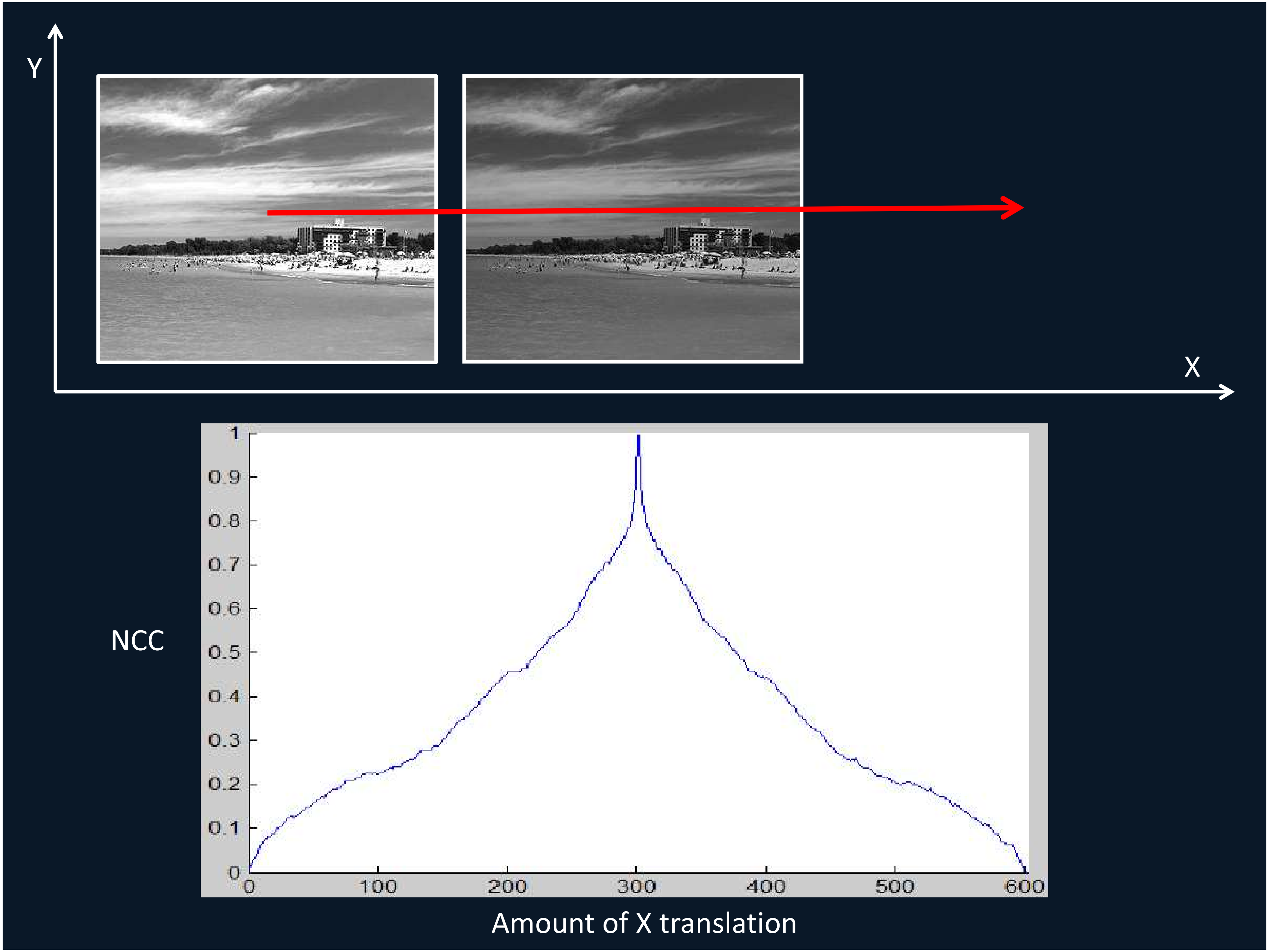
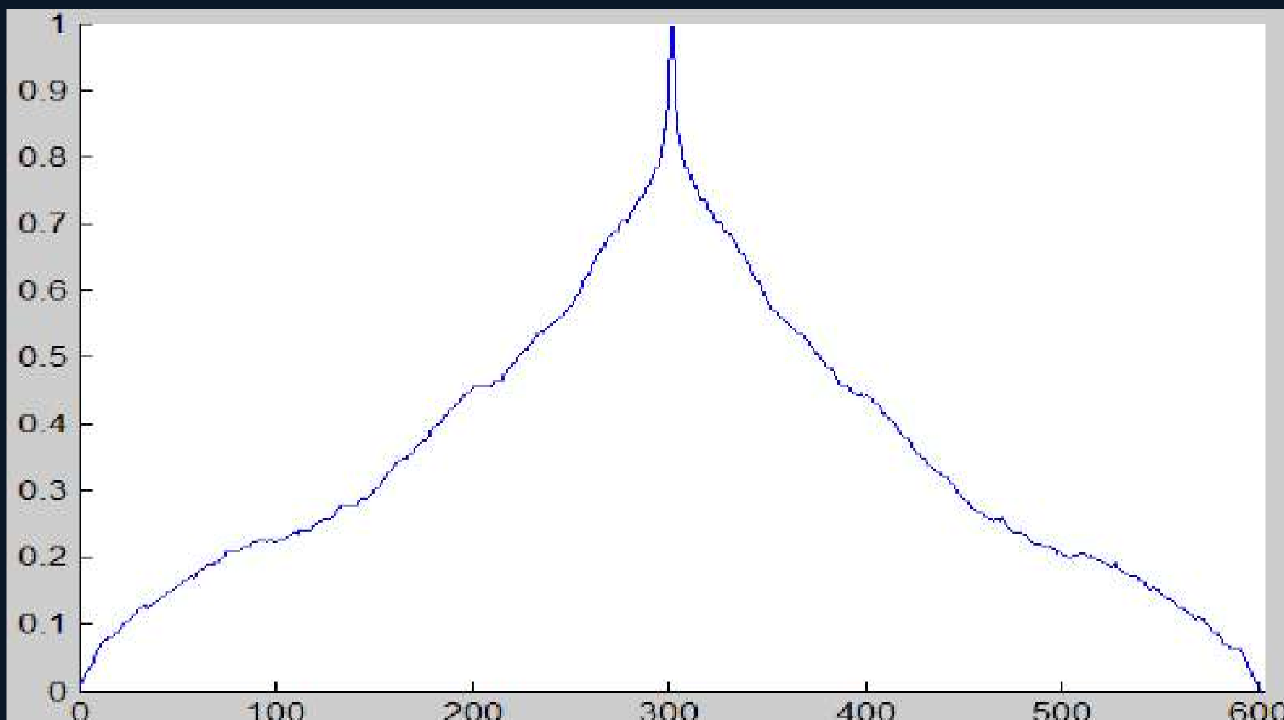

Y
X
NCC
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Amount of X translation

Image similarity metric: Normalized cross correlation

The steep gradient is more favourable for rapid optimization. Gradient ascent could be used for this.

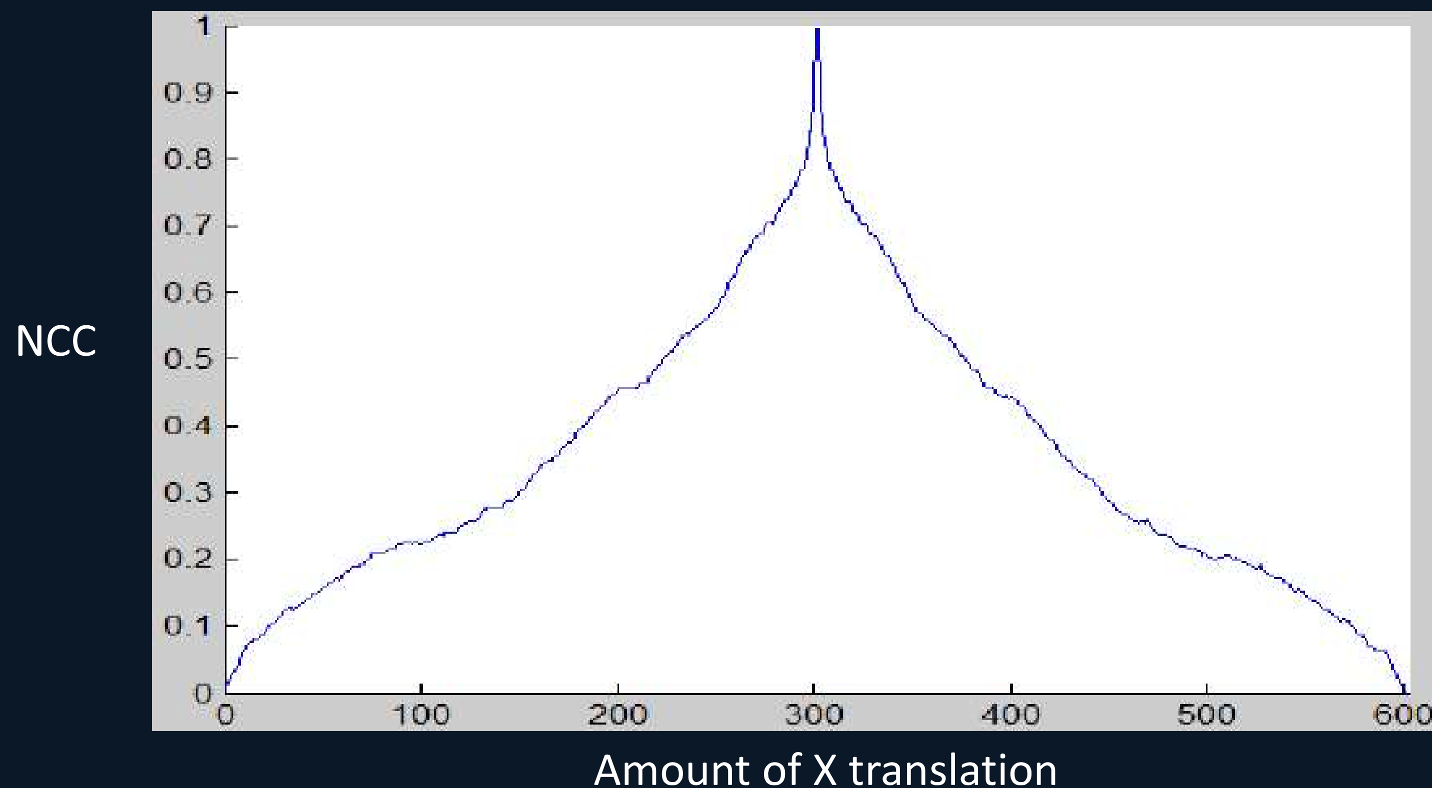
NCC

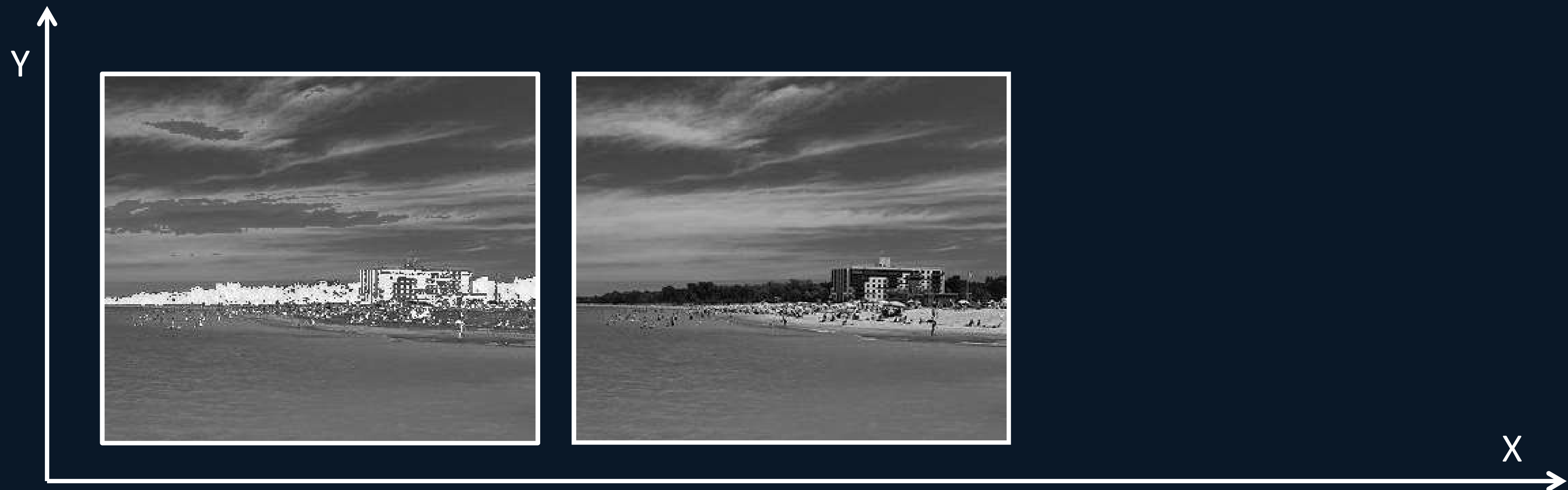


Amount of X translation

Image similarity metric: Normalized cross correlation

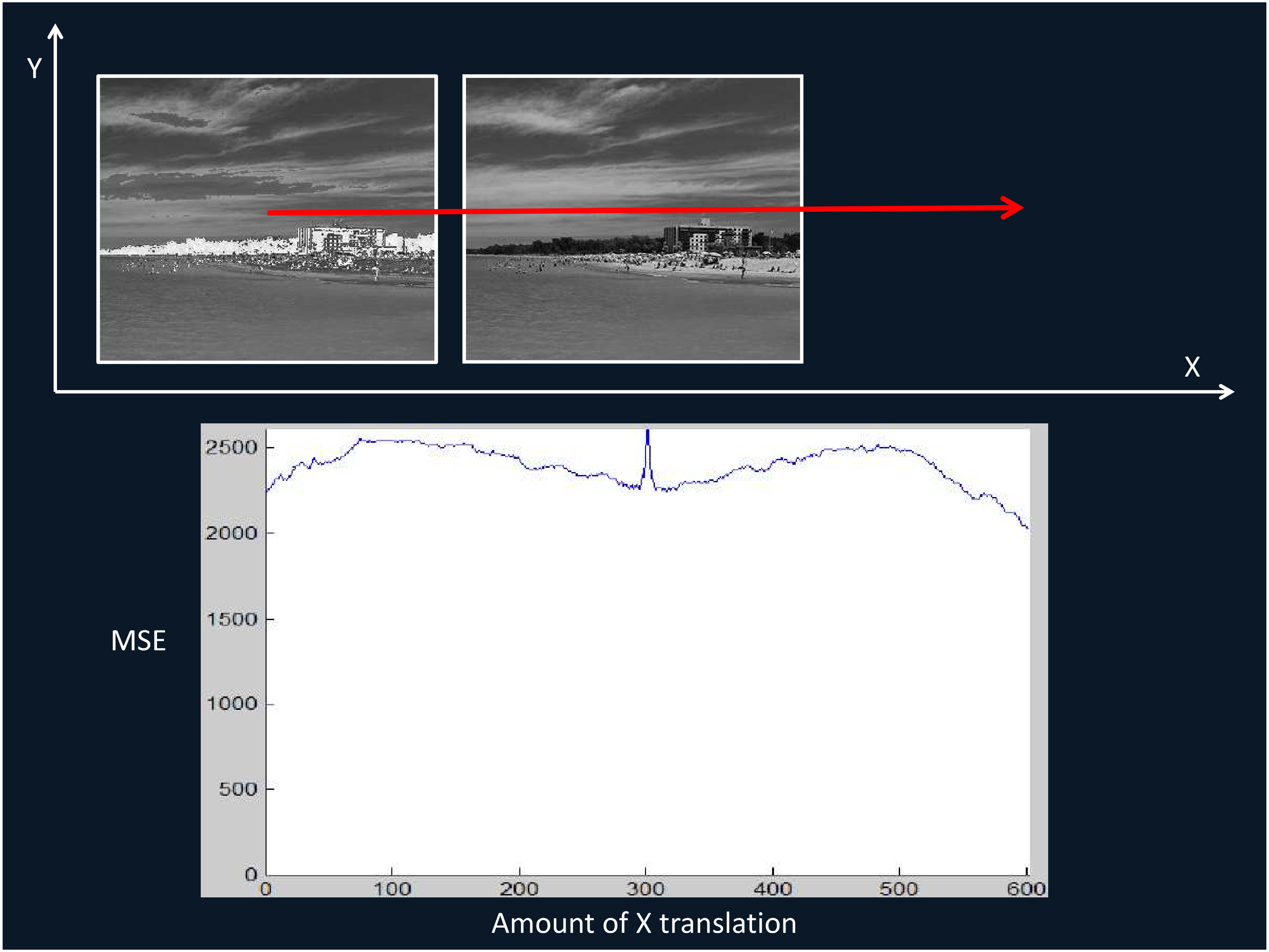
Since you know that the NCC metric is essentially a dot product, you know that it ranges from -1 (worst match) to 1 (best match).

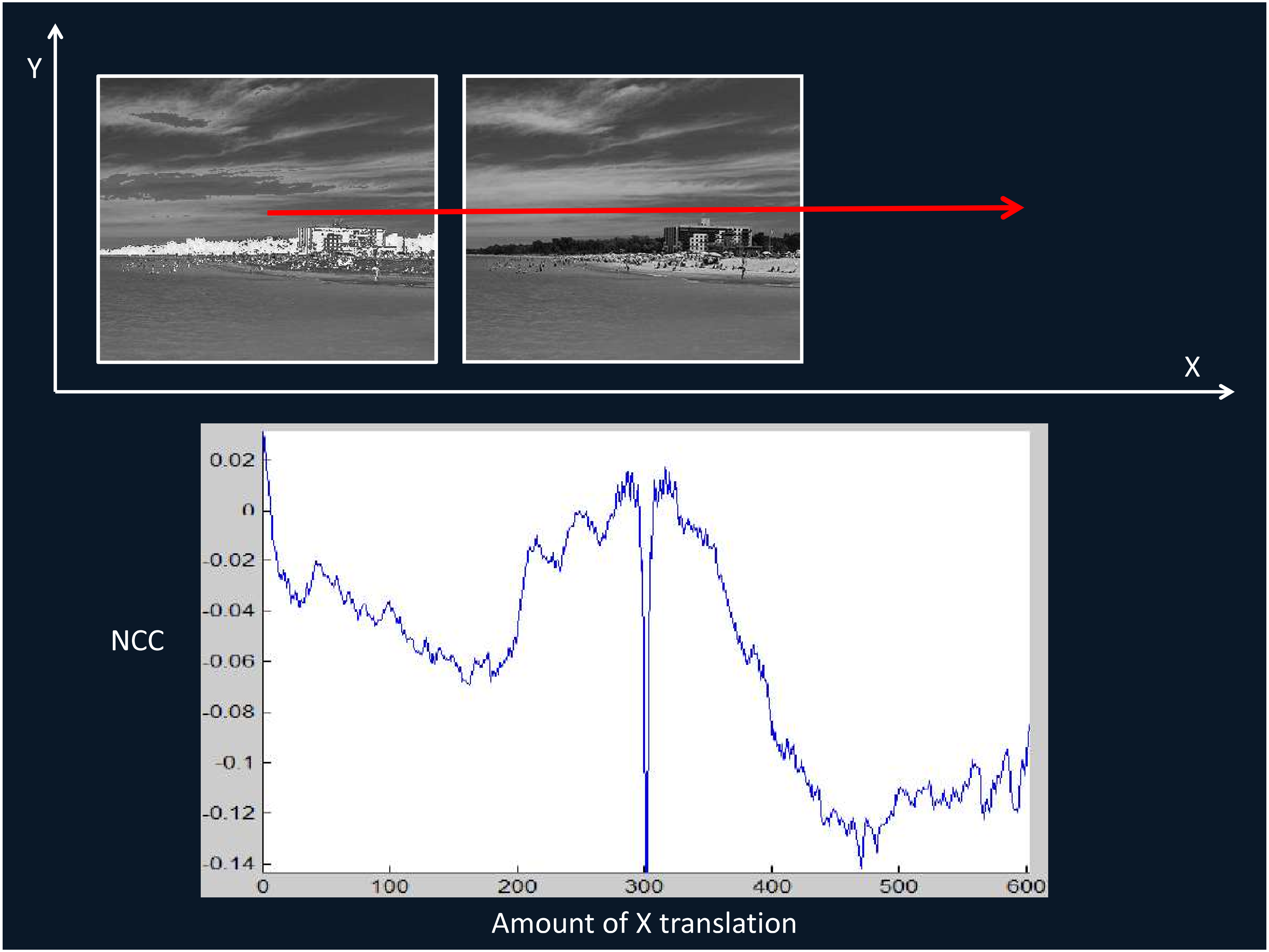


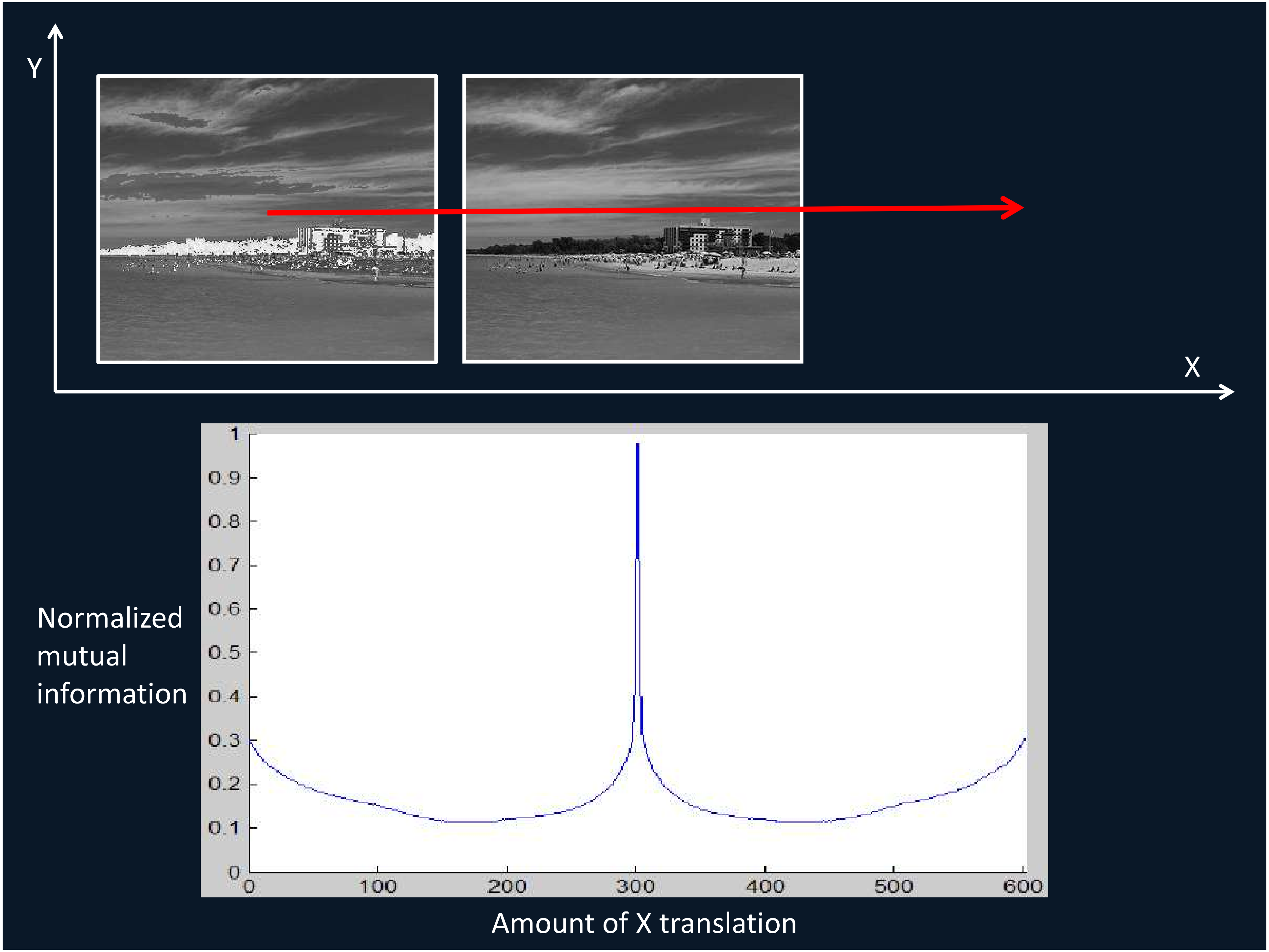


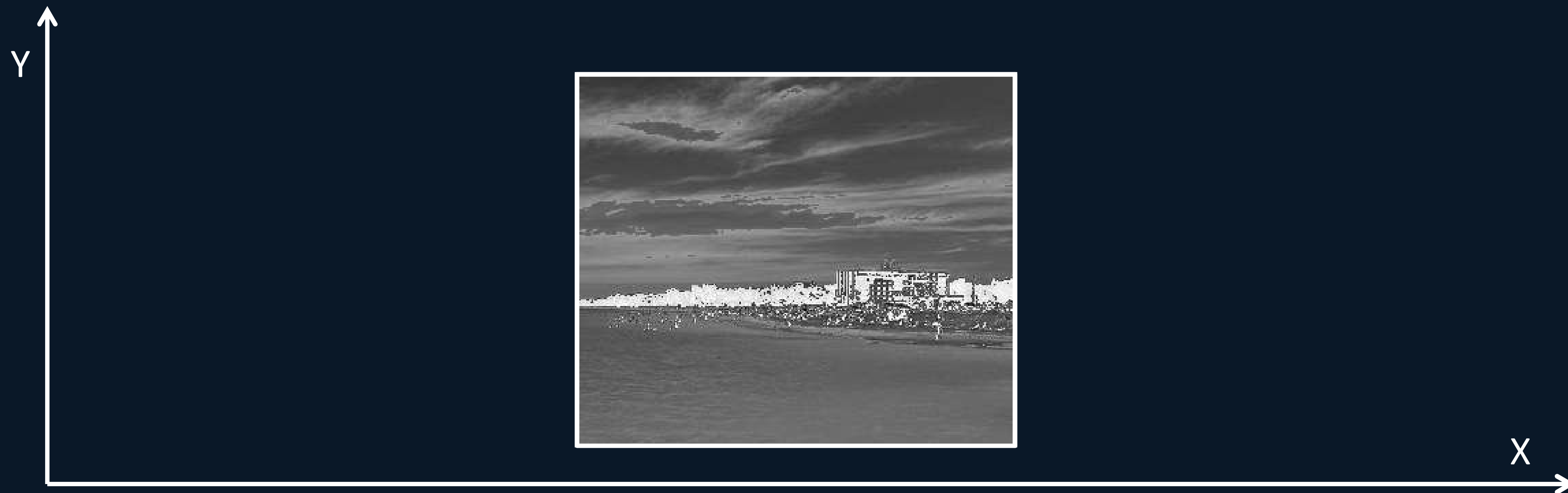
In this example, all the dark pixels (below intensity 50) and all the bright pixels (above intensity 150) have been complemented.

So the dark buildings became bright and the brightest parts of the clouds became darker, but otherwise the image was not modified. This is analogous to CT and MRI registration, where bone is bright on CT and dark on MRI.

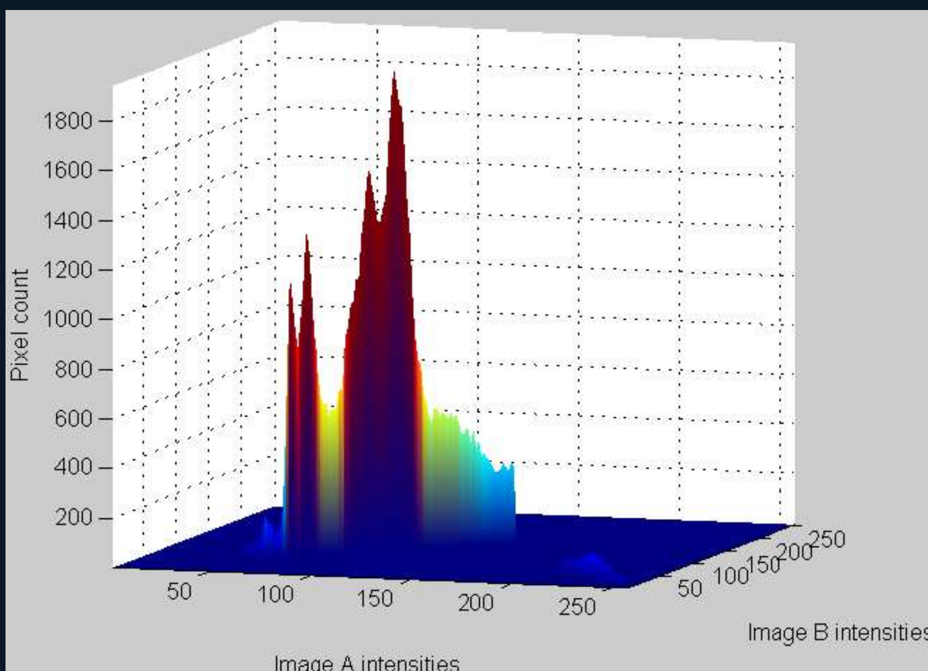

Y
X
MSE
2500
2000
1500
1000
500
0
100
200
300
400
500
600
Amount of X translation

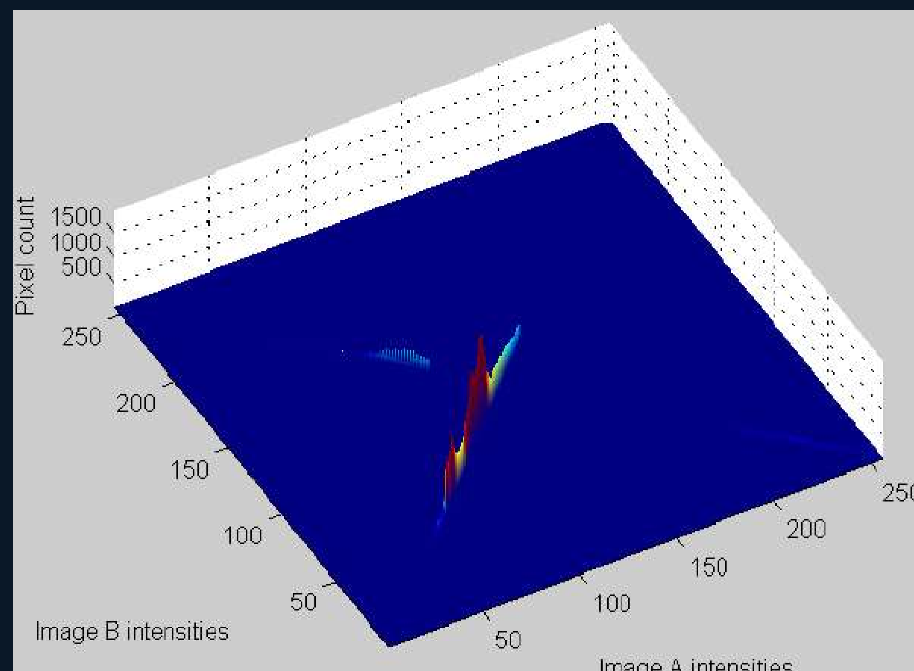

Y
X
NCC
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
0
100
200
300
400
500
600
Amount of X translation


Y
X
Normalized mutual information
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Amount of X translation

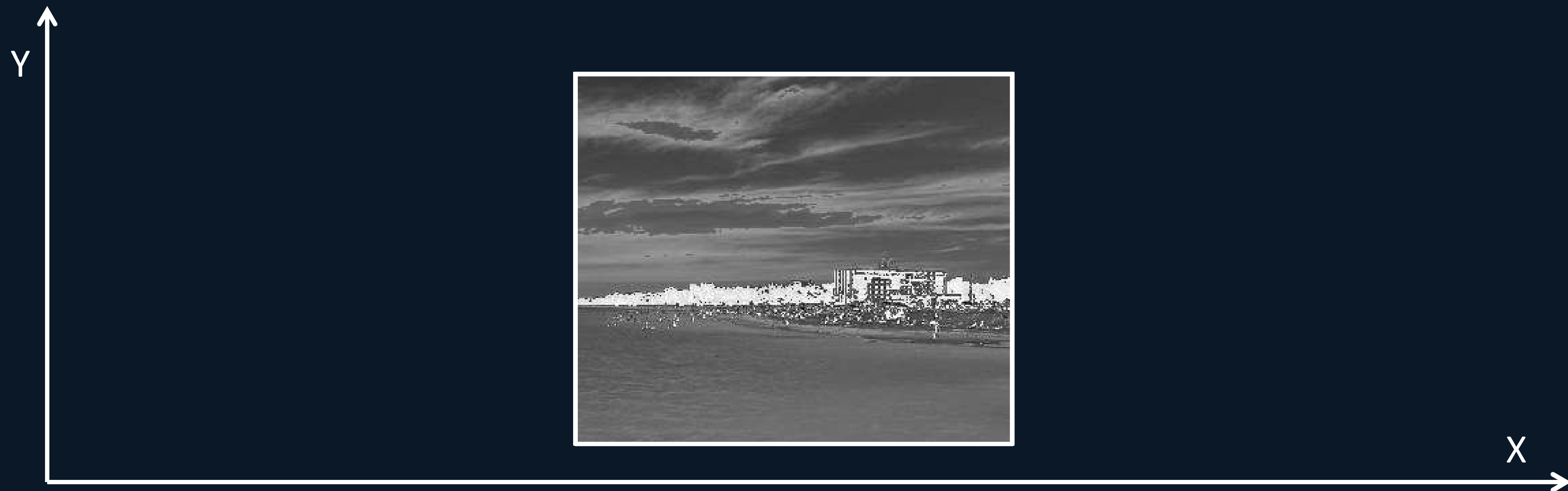


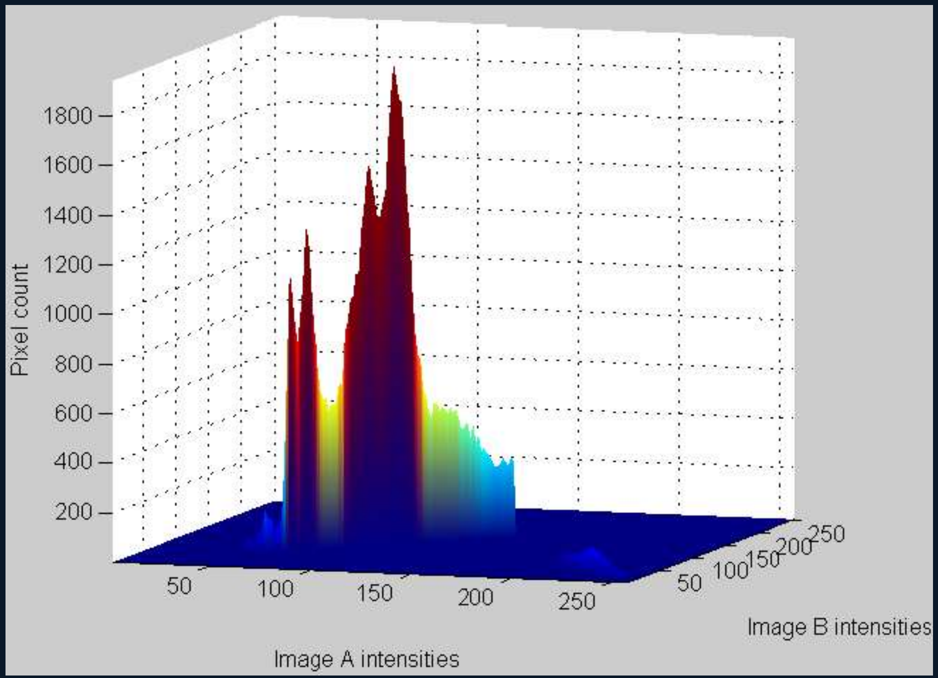
We begin with the situation where the images are perfectly aligned.

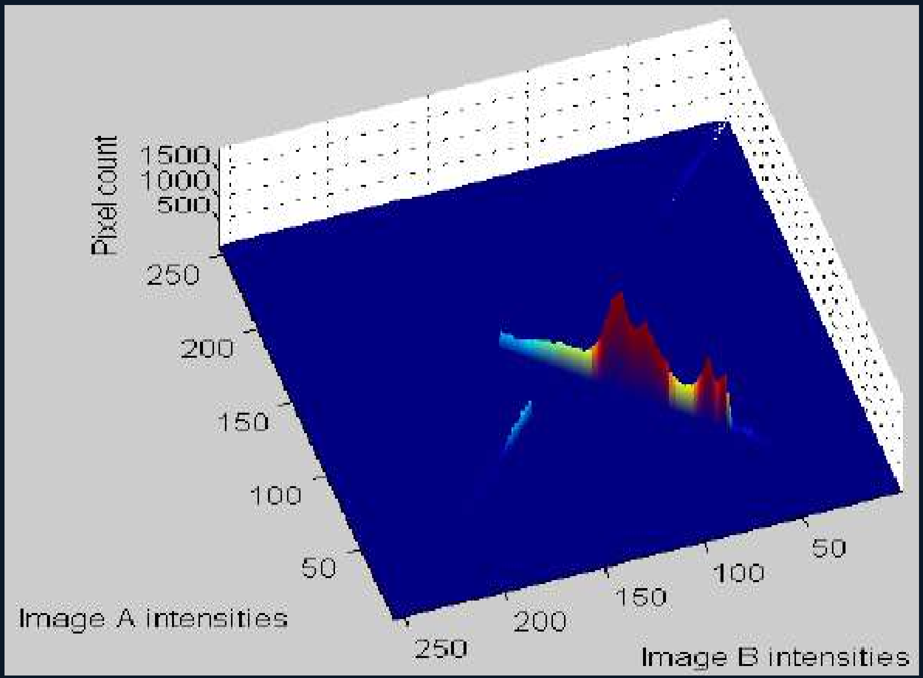




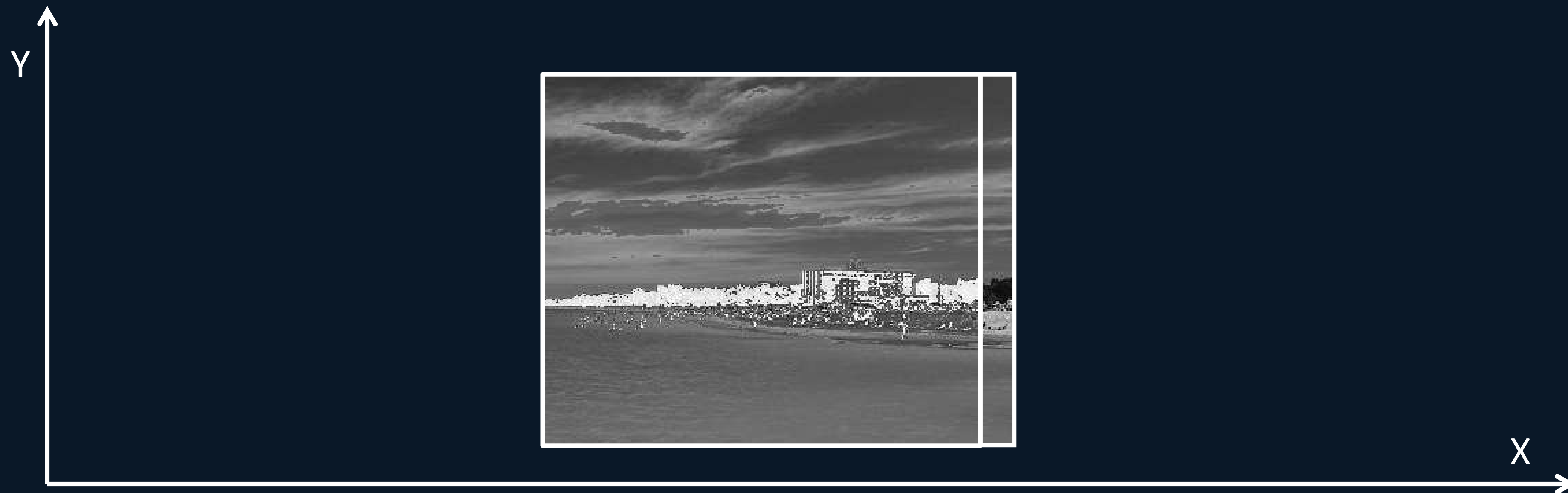
Notice that there are compact ridges, but they do not lie along any one line.



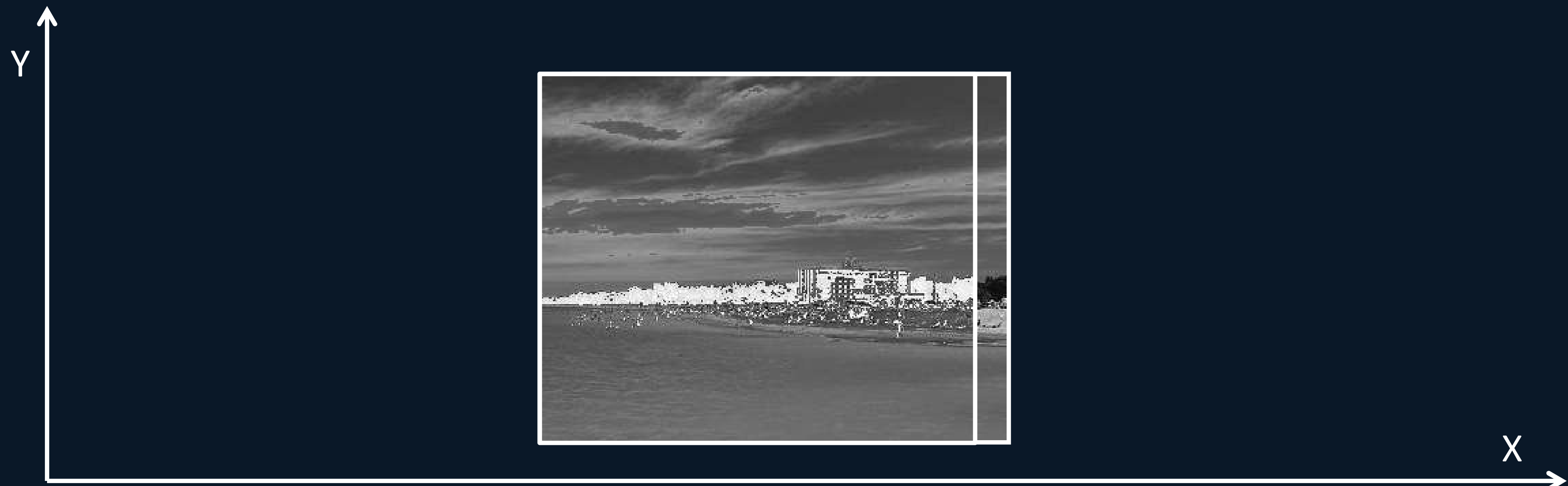


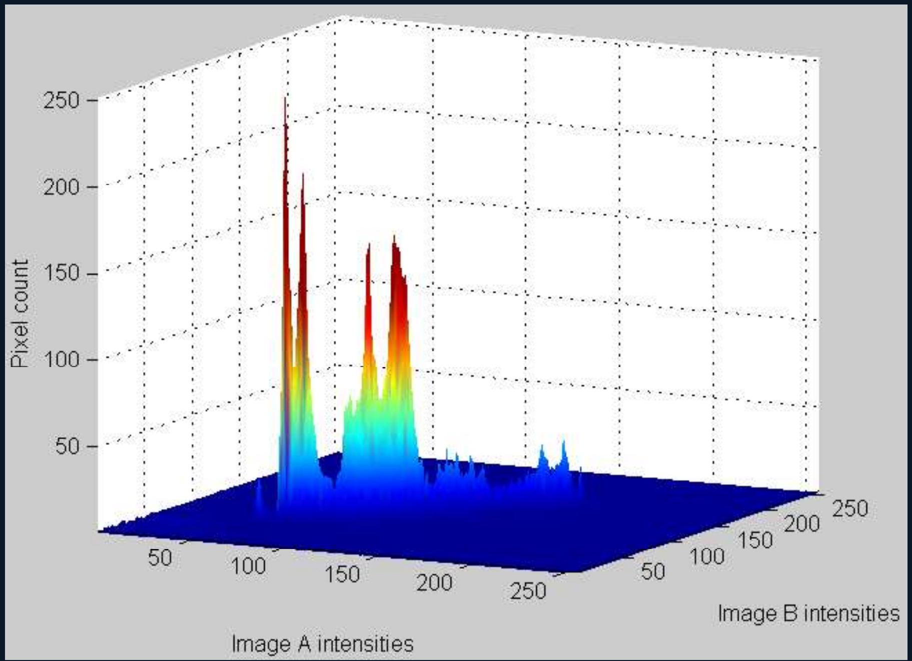


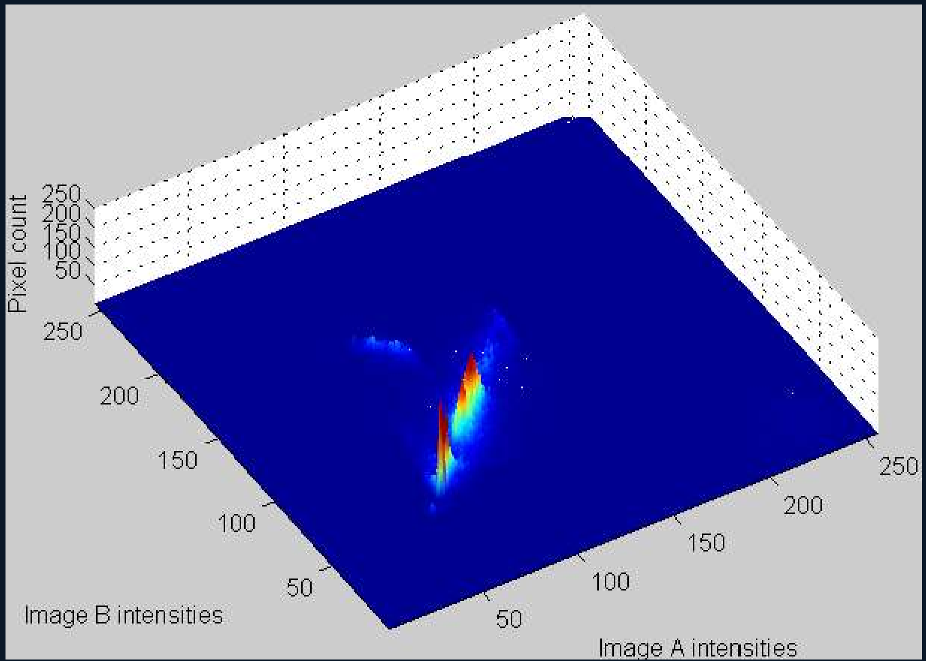
Another view (right).



Now let’s move the images apart a bit and look at the joint intensity histogram only in the region where they overlap.

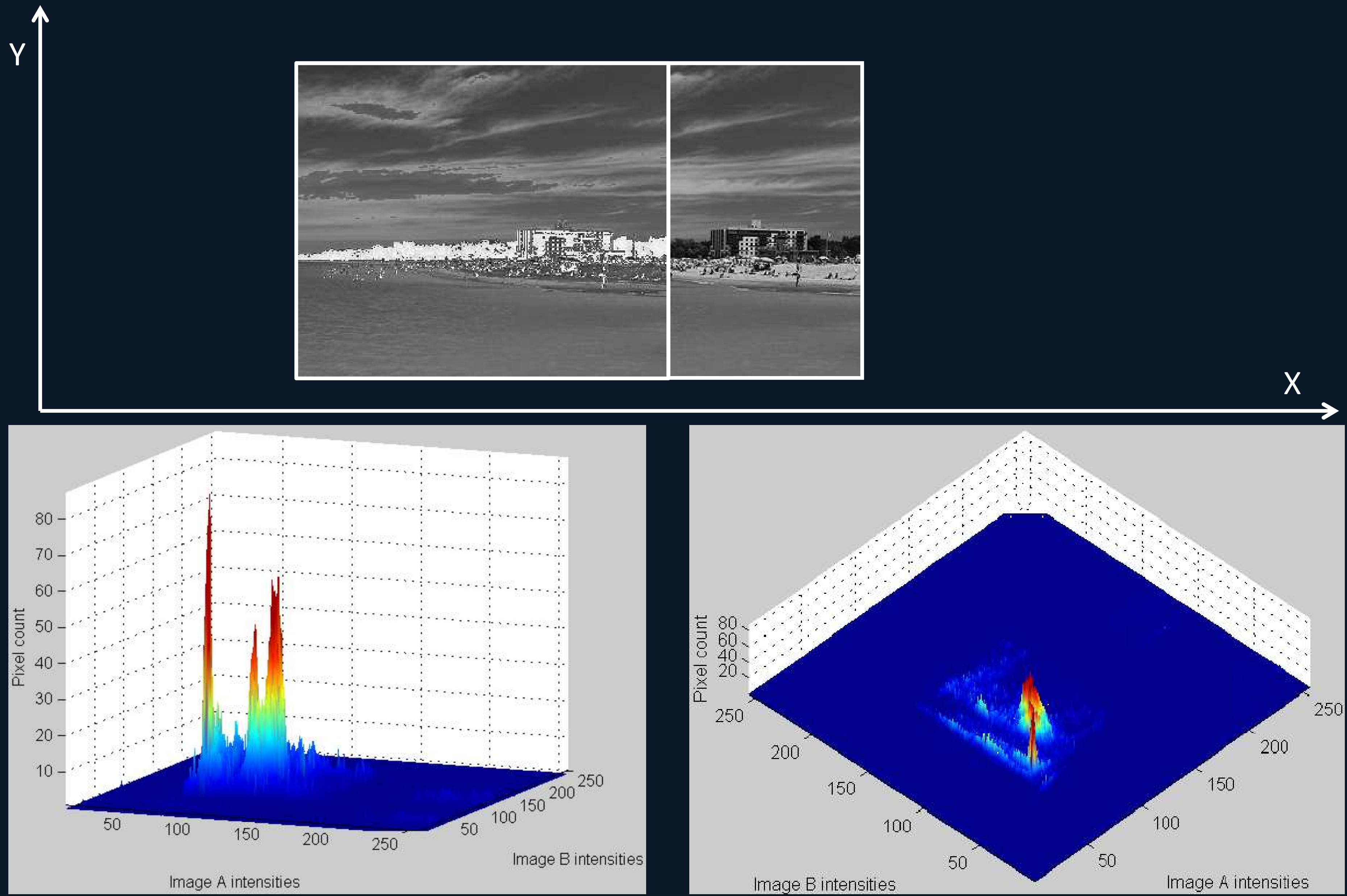






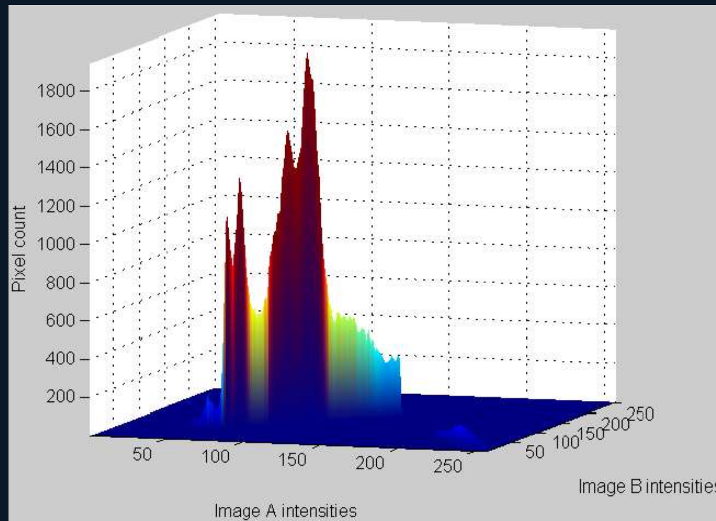
Notice that both ridges are starting to flatten.

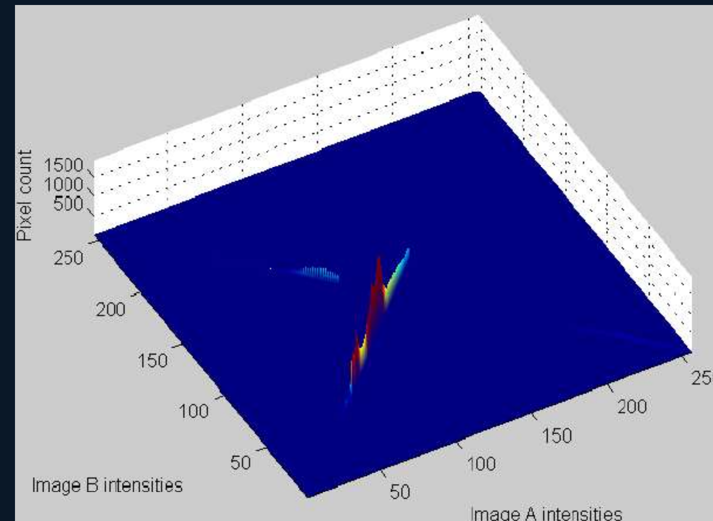
Now let's move the images apart so they only overlap by about 50% and look at the joint intensity histogram only in the region where they overlap.



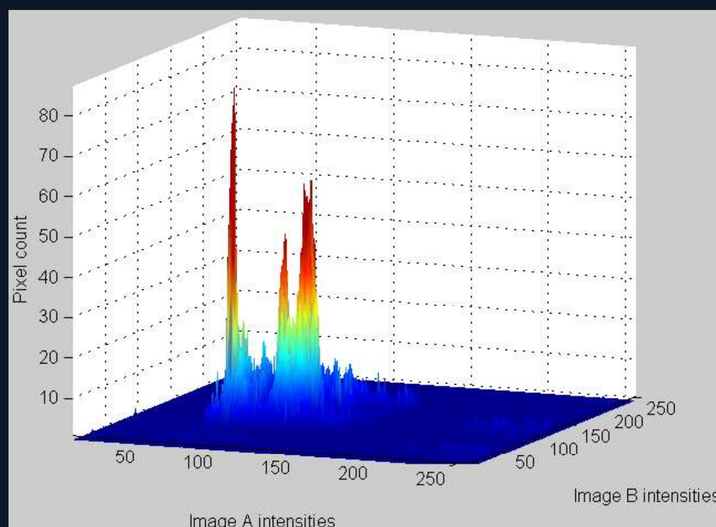
The joint intensity histogram values have become much more distributed, and the heights of the peaks are lower.

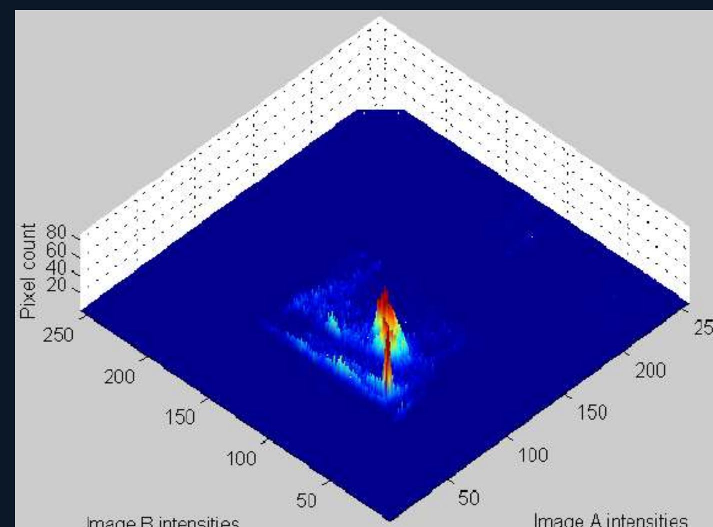
Good alignment





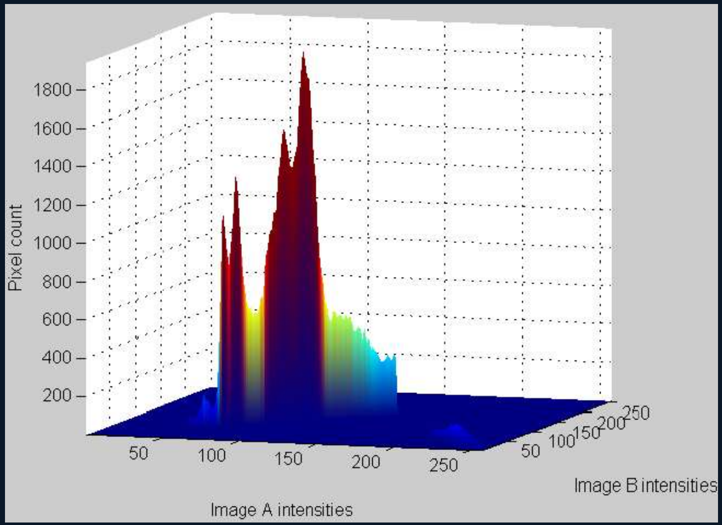
Bad alignment

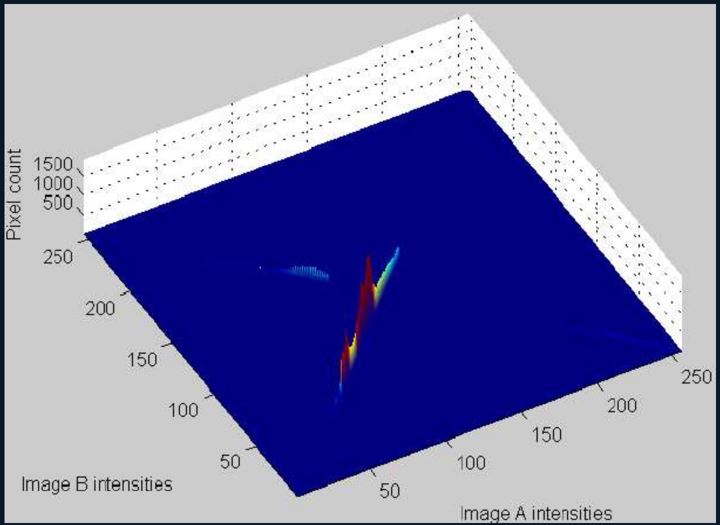




Key observation:
Good alignment = histogram with narrow, tall peaks.
Bad alignment = flatter histogram.

Good alignment





How to calculate an image similarity metric (scalar "goodness of match" score) based on this observation?

To do this, we need to learn a little information theory.

First, a thought experiment.

How much “information” do you get from a flipped coin?

Scenario 1:

You have a completely unfair coin. Every time you flip it, it comes up as “head”.

You flip the coin 999 times, and every time, it comes up as “head”.

Do you get any new information from the 1000th flip?

“Tail”

“Head”

How much "information" do you get from a flipped coin?

Scenario 1:

Do you get any new information from the 1000th flip?

No: You already know what it's going to be.

"Tail"

"Head"

How much “information” do you get from a flipped coin?

Scenario 2:

You have a completely fair coin. Every time you flip it, there is a 50% probability it will come up as “head”, and 50% probability it will come up as “tail”.

You flip the coin 999 times.

Do you get any new information from the 1000^{th} flip?

“Tail”

“Head”

How much "information" do you get from a flipped coin?

Scenario 2:

Do you get any new information from the 1000th flip?

Yes: You have no idea what it is going to be.

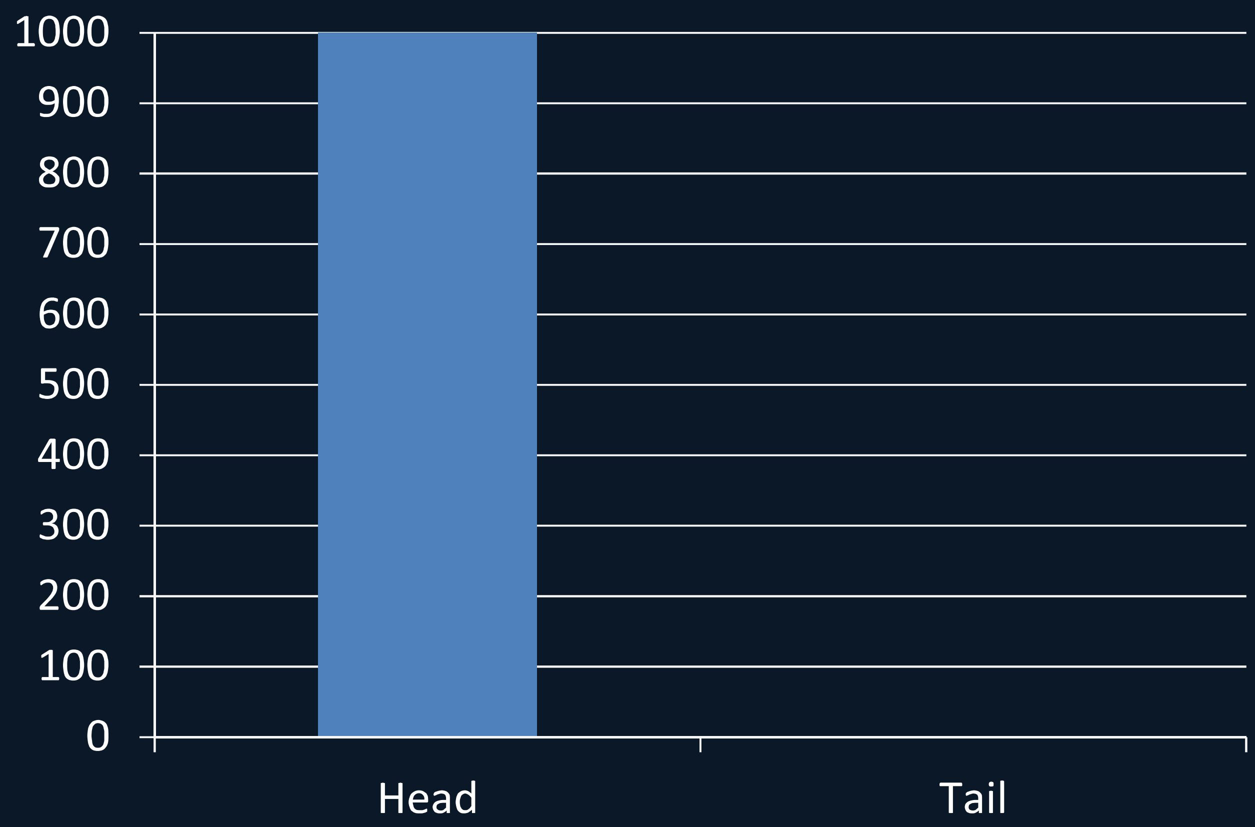
You get 1 bit of completely new information: either it's a tail (0) or a head (1).

"Tail"

"Head"

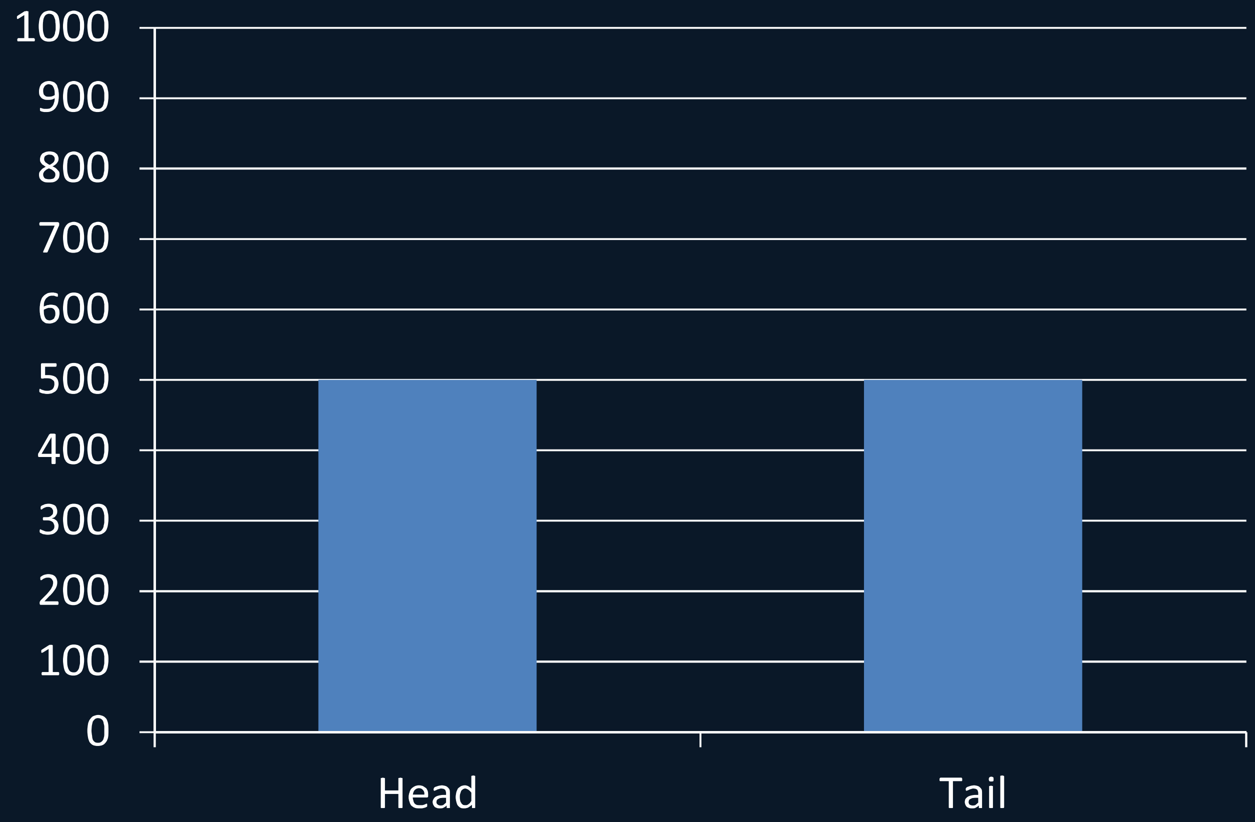
How much “information” do you get from a flipped coin?

Scenario 1: Unfair coin – histogram of flips

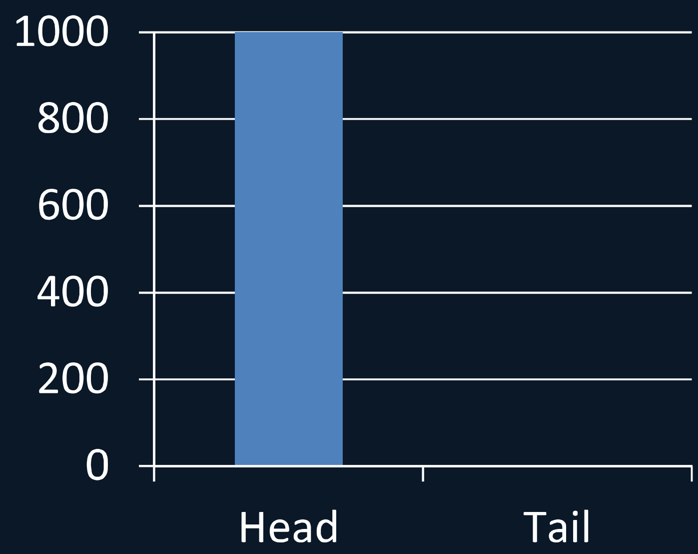


How much “information” do you get from a flipped coin?

Scenario 2: Fair coin – histogram of flips



How much "information" do you get from a flipped coin?

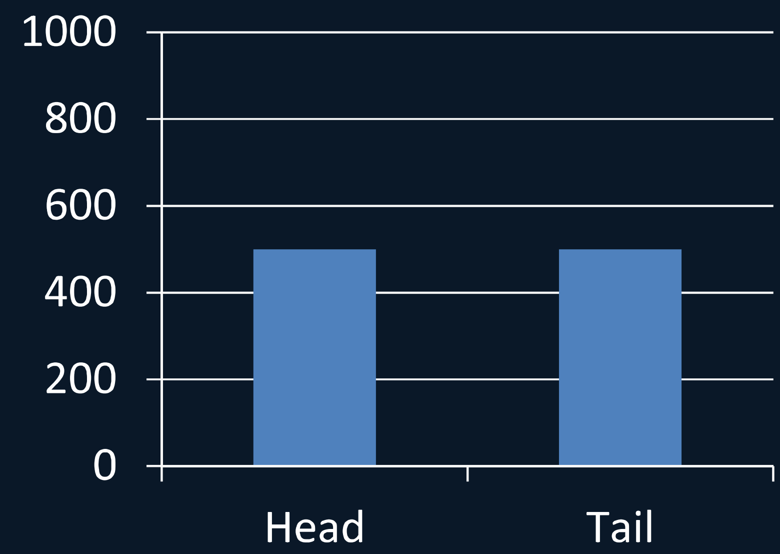


Completely unfair coin:

Can predict result of next flip from previous flips.

New flips **do not** provide additional information.

Histogram is **compact** with a **tall peak**.



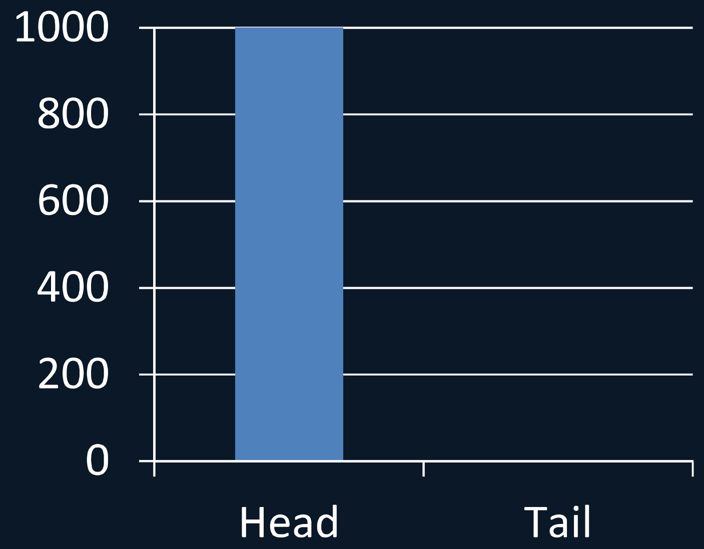
Completely fair coin:

Cannot predict result of next flip from previous flips.

New flips **do** provide additional information.

Histogram is **flat** with **no peak**.

How much "information" do you get from a flipped coin?

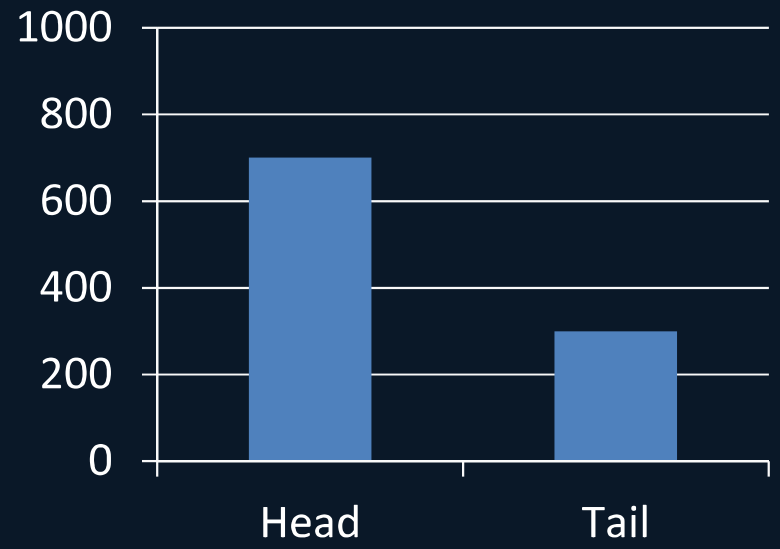


Completely unfair coin:

Can predict result of next flip from previous flips.

New flips **do not** provide additional information.

Histogram is **compact** with a **tall peak**.



Somewhat fair coin:

Can predict result of next flip from previous flips, but not perfectly.

New flips provide **some** additional info.

Histogram is **somewhat flat** with a **short peak**.

How much “information” do you get from a flipped coin?

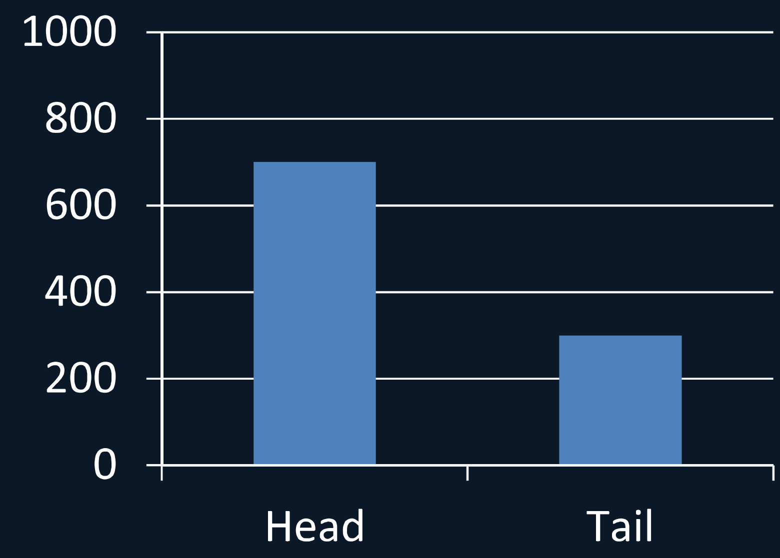
How much “mutual information” is there in a pair of aligned images?

How well can you predict the 1000^{th} coin flip from the first 999 flips?

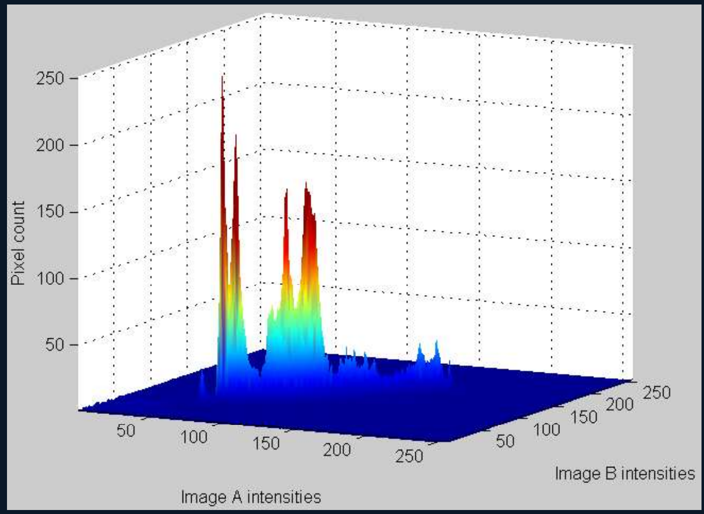
How well can you predict the intensity contents of one image based on those in another aligned image?

How much "mutual information" is there in a pair of aligned images?

To measure the amount of information contained in the coin flips, we needed to measure something about this histogram.

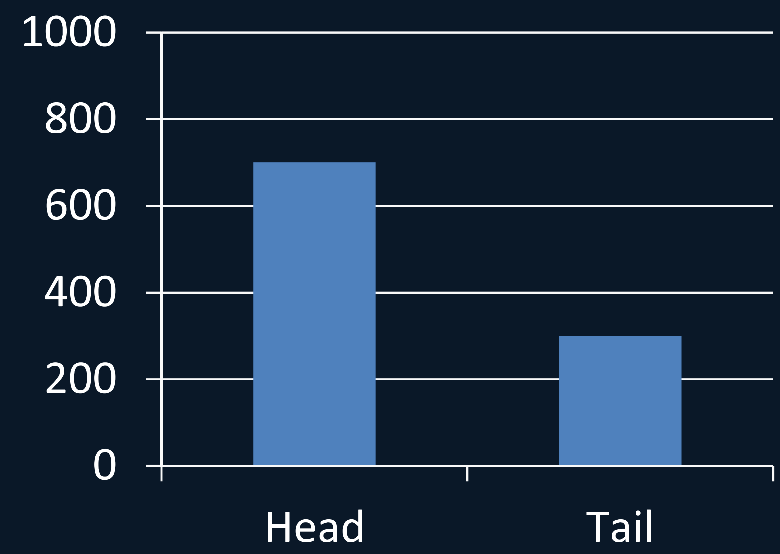


To measure the mutual information contained in a pair of aligned images, we need to measure something about this histogram.

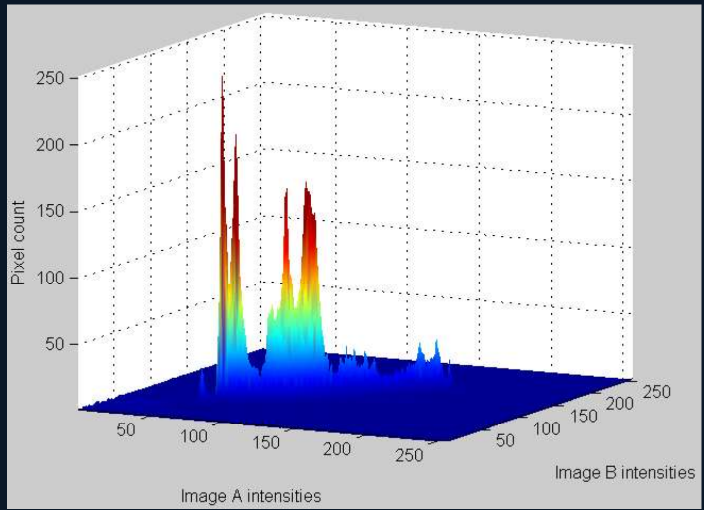


How much "mutual information" is there in a pair of aligned images?

The basic measurement that we will use is the same for both: information entropy.



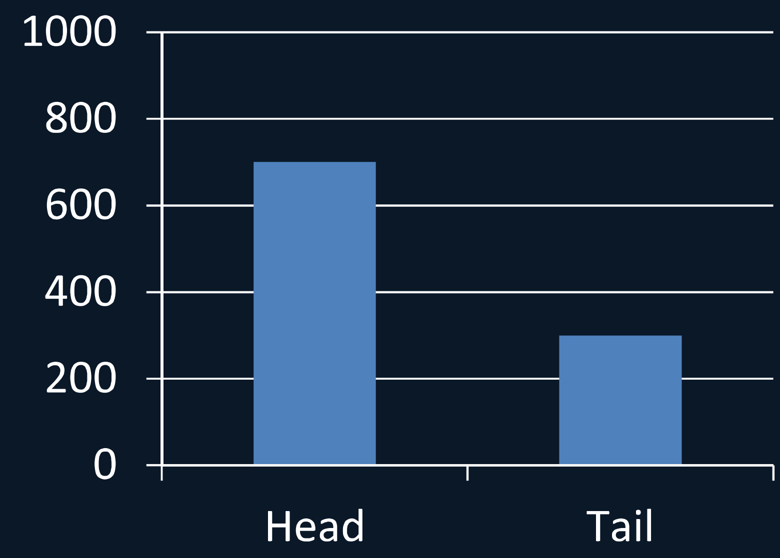
$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$



X: Random variable (intensity/coin flip result)
x: Specific value of random variable
$p(x)$: Probability of x.

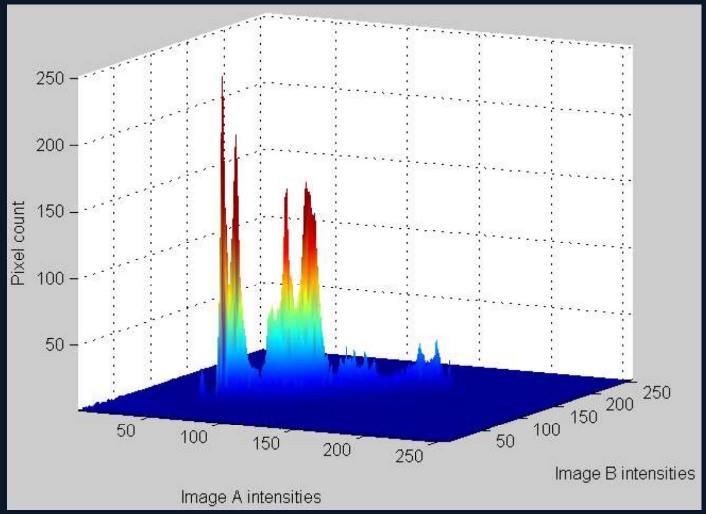
How much "mutual information" is there in a pair of aligned images?

The basic measurement that we will use is the same for both: information entropy.



$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

X: Random variable
(a variable whose possible values are numerical outcomes of a random phenomenon)

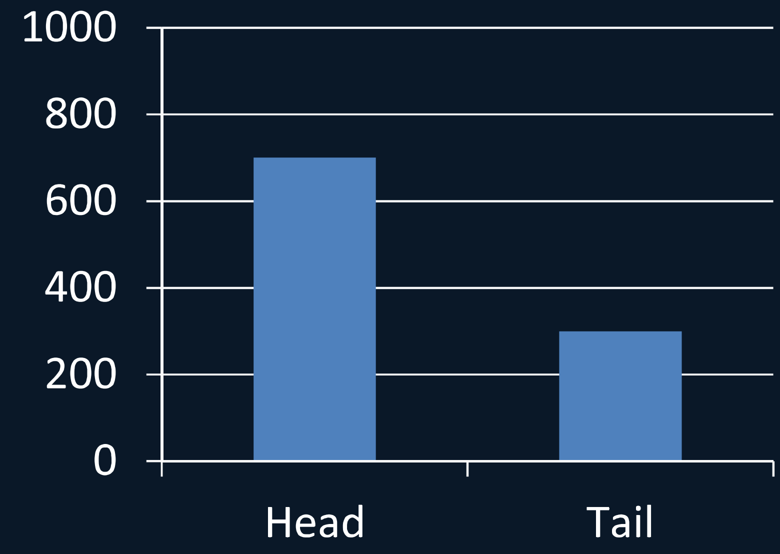


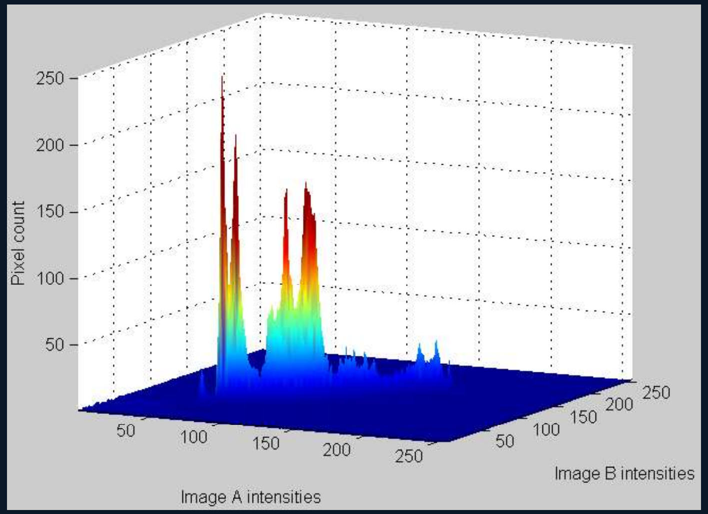
How much "mutual information" is there in a pair of aligned images?

The basic measurement that we will use is the same for both: information entropy.

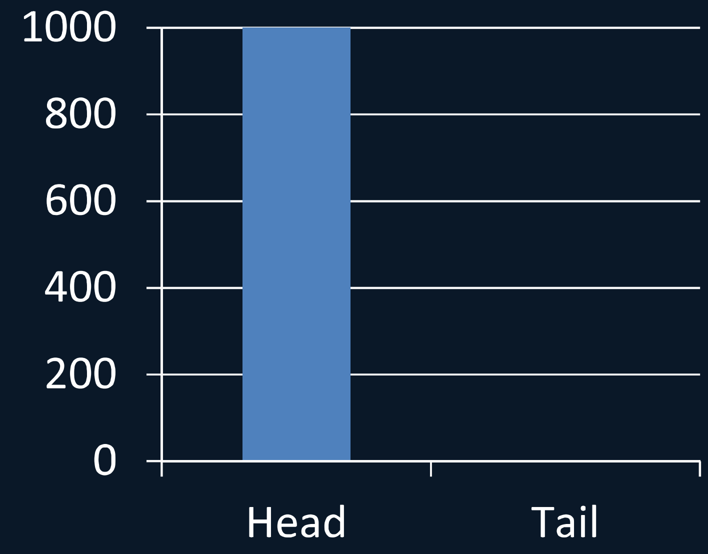
$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

X: Random variable (intensity/coin flip result)
x: Specific value of random variable
$p(x)$: Probability of x.





How much "mutual information" is there in a pair of aligned images?

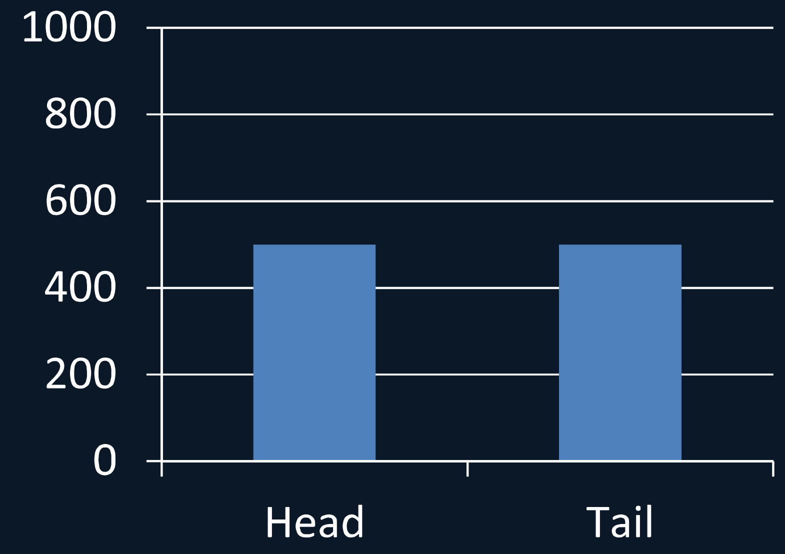


$$H(X) = -\sum_x p(x) \log_2 p(x)$$

$$H(X) = -(1 \times \log_2(1) + 0 \times \log_2(0))$$
$$H(X) = 0$$

I.e., each new coin flip gives 0 bits of information.

How much "mutual information" is there in a pair of aligned images?

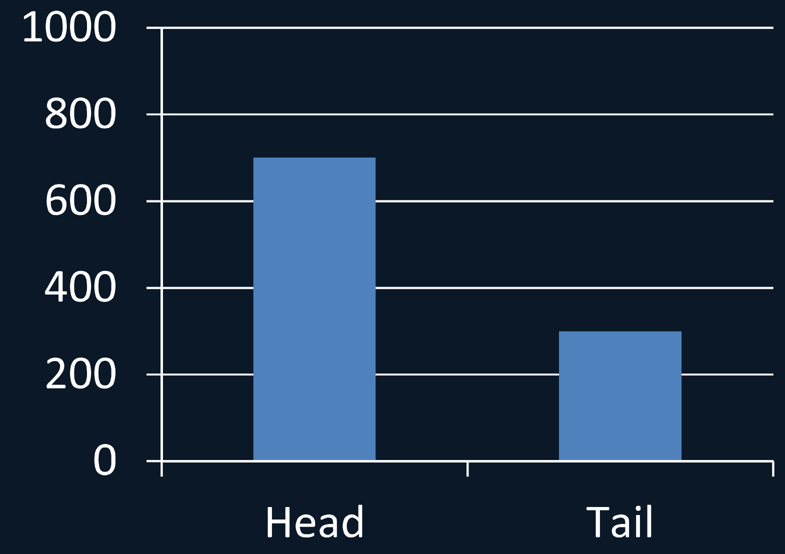


$$H(X) = -\sum_x p(x) \log_2 p(x)$$

$$H(X) = -(0.5 \times \log_2(0.5) + 0.5 \times \log_2(0.5))$$
$$H(X) = 1$$

I.e., each new coin flip gives 1 bit of information. To transmit a message describing 1000 coin flips, you'd need a minimum of 1000 bits.

How much "mutual information" is there in a pair of aligned images?



$$H(X) = -\sum_x p(x) \log_2 p(x)$$

$$H(X) = -(0.75 \times \log_2(0.75) + 0.25 \times \log_2(0.25))$$
$$H(X) = 0.8113$$

I.e., each new coin flip gives 0.8113 bits of information. To transmit a message describing 1000 flips, you'd need 812 bits at minimum no matter how cleverly efficient you are.

How much “mutual information” is there in a pair of aligned images?

$$H(X) = -\sum_x p(x)\log_2 p(x)$$

This last point is worth emphasizing. When $H(X)$ is higher, you need more bits to transmit information about coin tosses.

E.g. you can tell your friend about your 1,000 unfair coin tosses by simply saying “all heads”. But for the fair coin, you have to tell her each of the 1,000 heads or tails you got.

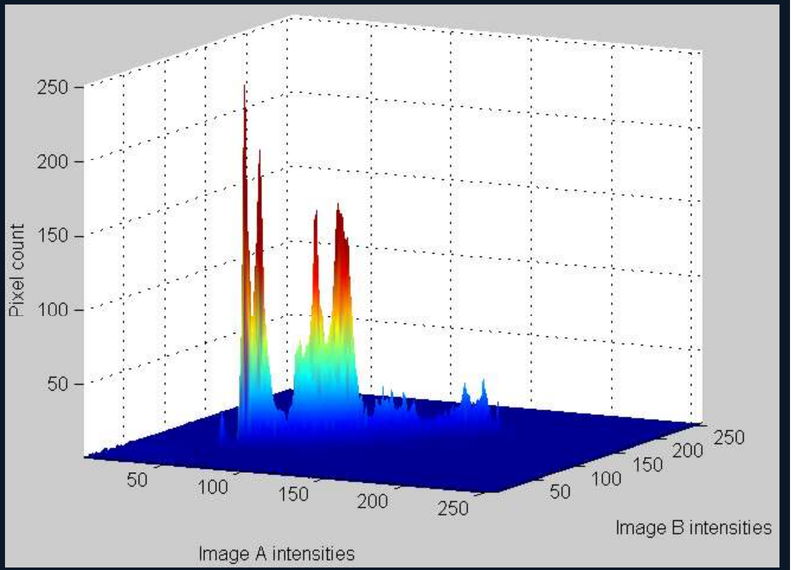
Larger $H(X)$ means that there is more information in signal X.

How much "mutual information" is there in a pair of aligned images?

The generalization to the 2D histogram:

$$H(A,B) = -\sum_{a}\sum_{b} p(a,b)\log_2 p(a,b)$$

A, B: Random variables (intensity)
a, b: Specific intensity values
$p(x)$: Probability of x.



Mutual information

$$H(A,B) = -\sum_a \sum_b p(a,b) \log_2 p(a,b)$$

$$MI(A,B) = -H(A,B) \text{ ?}$$

Could we just use $-H(A,B)$ as our image similarity metric?

When the histogram is compact with tall peaks (good image match), $-H(A,B)$ tends to 0.

When the histogram is totally flat, $-H(A,B)$ is -1.

Seems like a suitable metric to maximize.

Mutual information

$$H(A,B) = -\sum_a \sum_b p(a,b) \log_2 p(a,b)$$
$$MI(A,B) = -H(A,B) \text{ ?}$$

Could we just use $-H(A,B)$ as our image similarity metric?

One problem: if the images don't overlap at all, then all the $p(a,b)$ values go to 0, so $MI(A,B)$ goes to 0. This is called a *degenerate solution*.

We need to do something to encourage the optimizer to avoid converging to this degenerate solution.

Mutual information

$$MI(A,B) = H(A) + H(B) - H(A,B)$$

How about this instead?

We have added the independent information entropy values of the two images to this function.

$H(A)$ and $H(B)$ are computed only within the overlap region of the two images.

So, if the images don't overlap at all, $H(A) + H(B) = 0$.