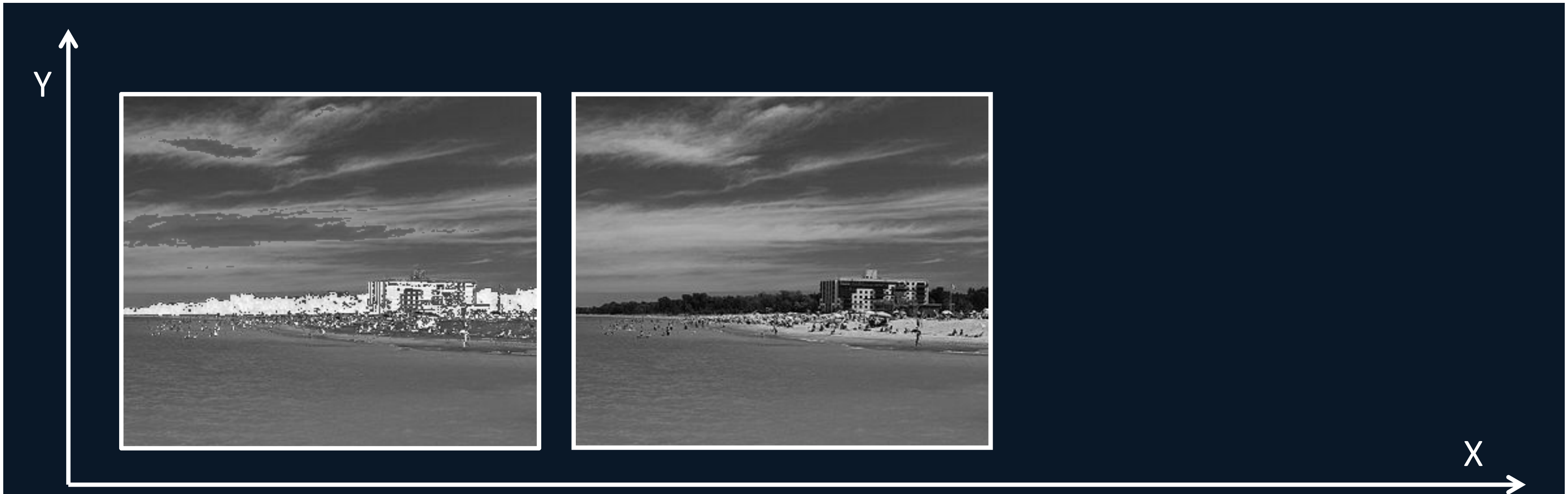
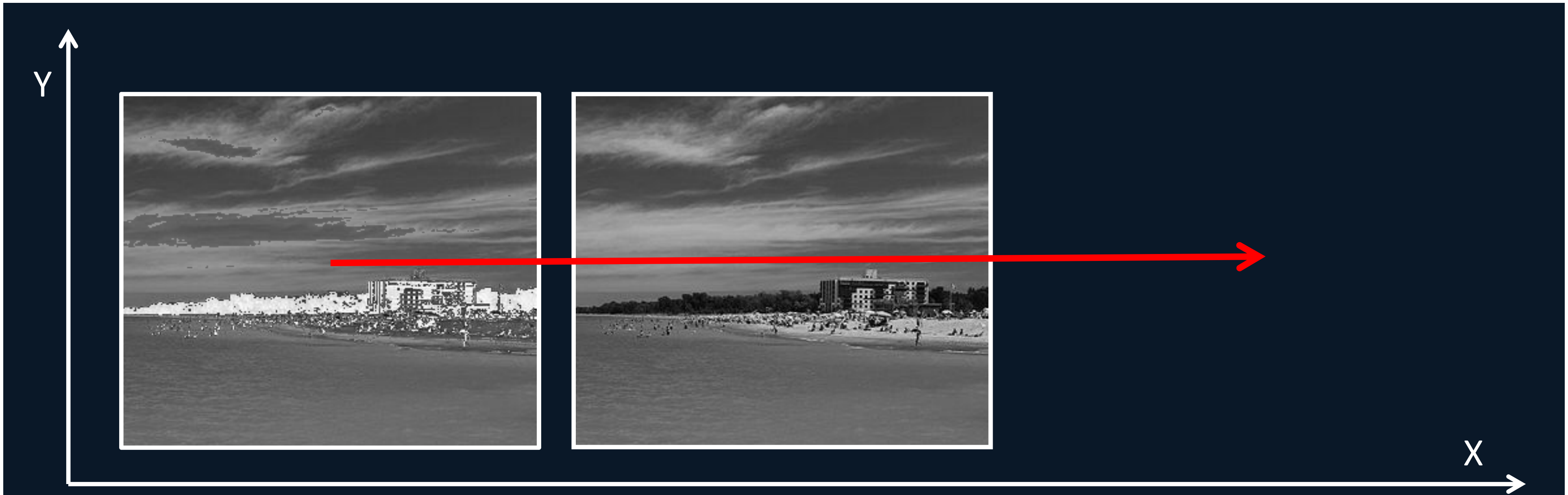
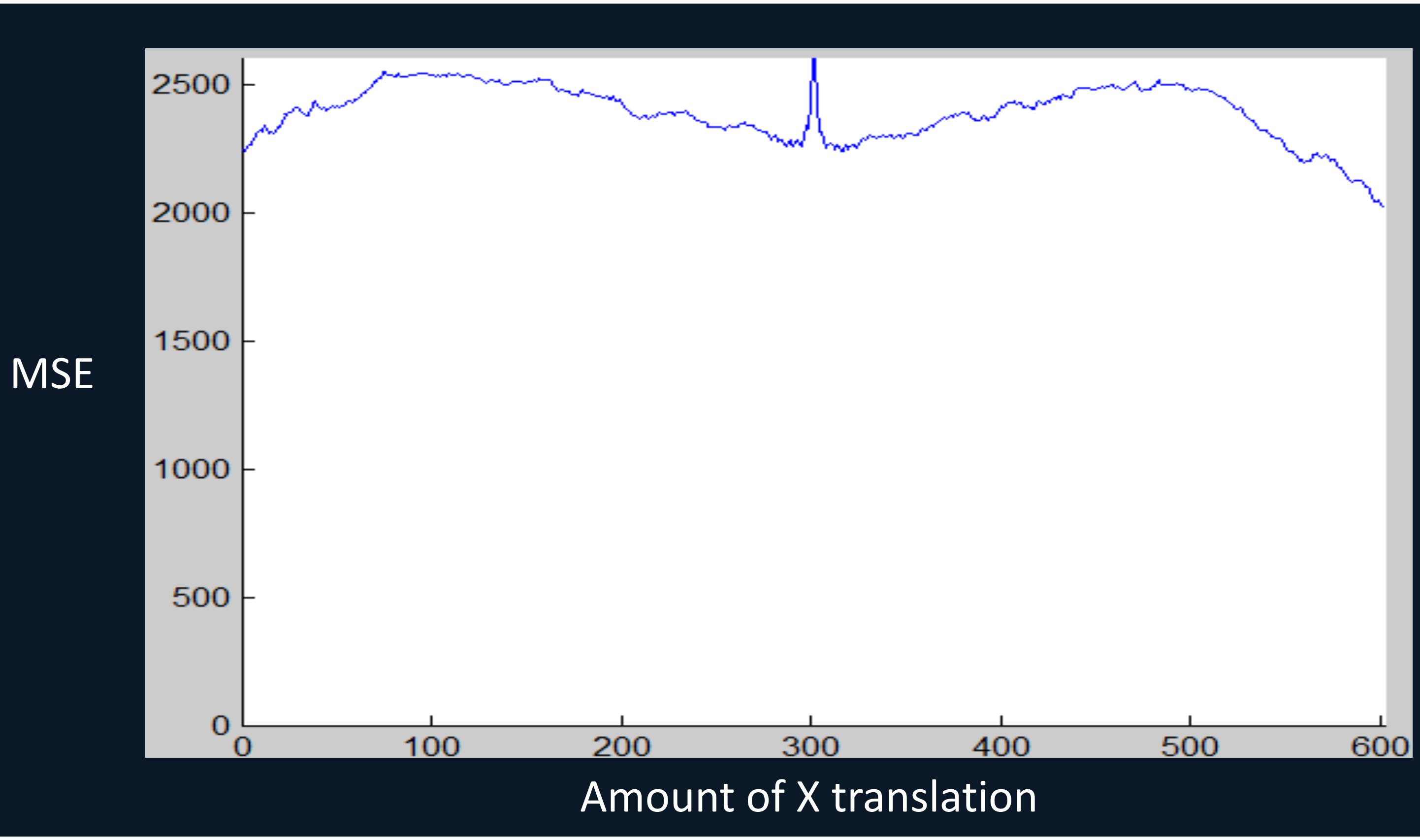
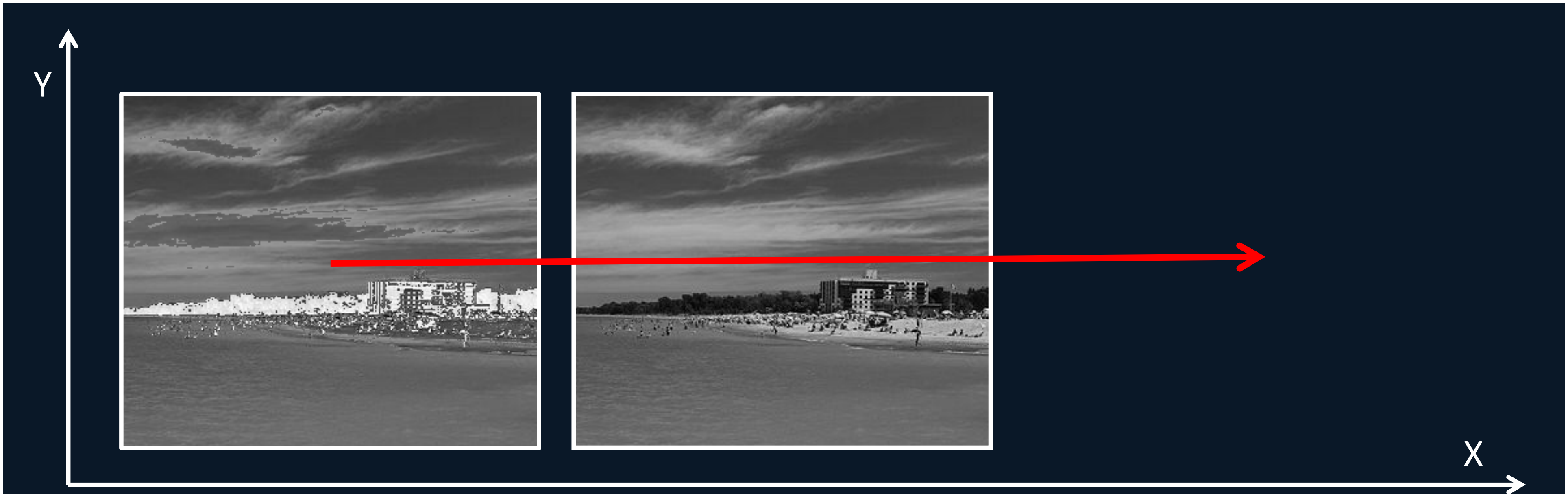


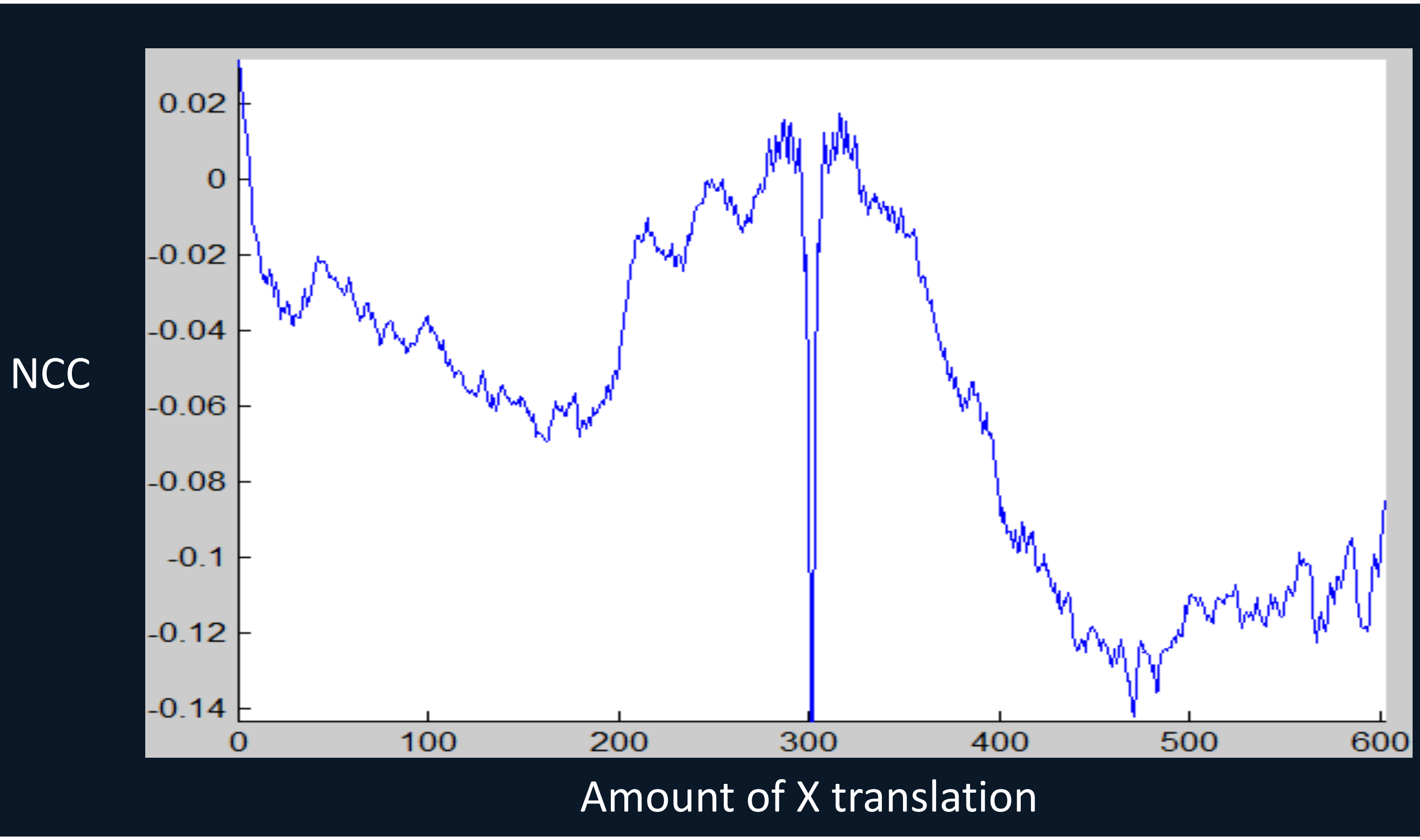
In this example, all the dark pixels (below intensity 50) and all the bright pixels (above intensity 150) have been complemented.

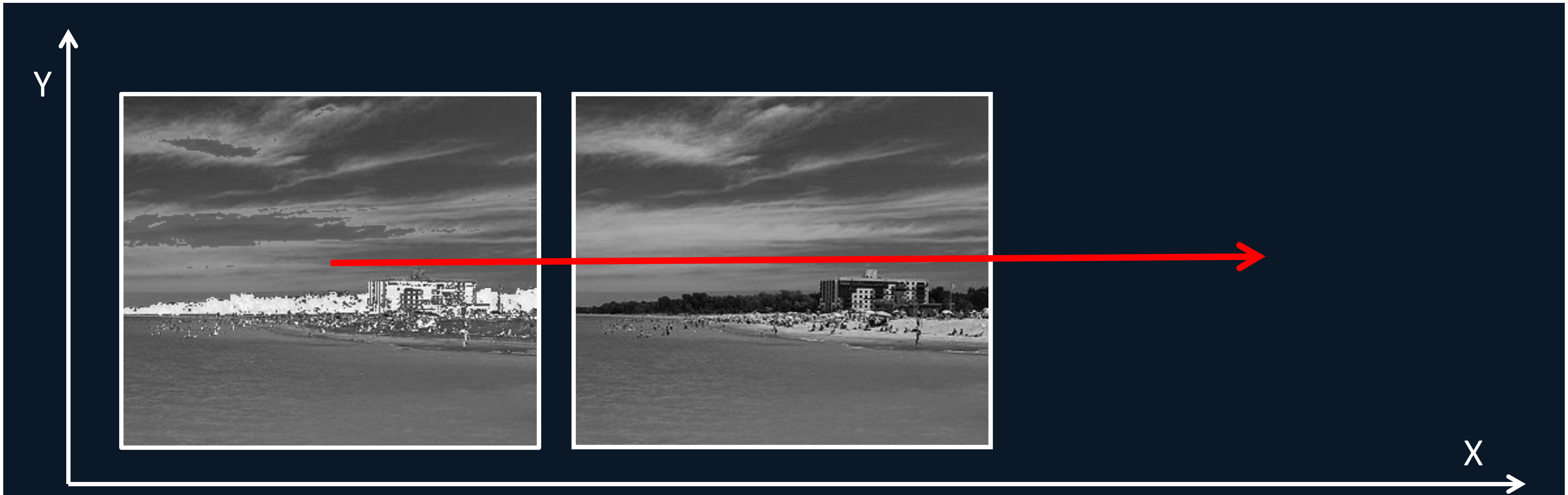
So the dark buildings became bright and the brightest parts of the clouds became darker, but otherwise the image was not modified. This is analogous to CT and MRI registration, where bone is bright on CT and dark on MRI.

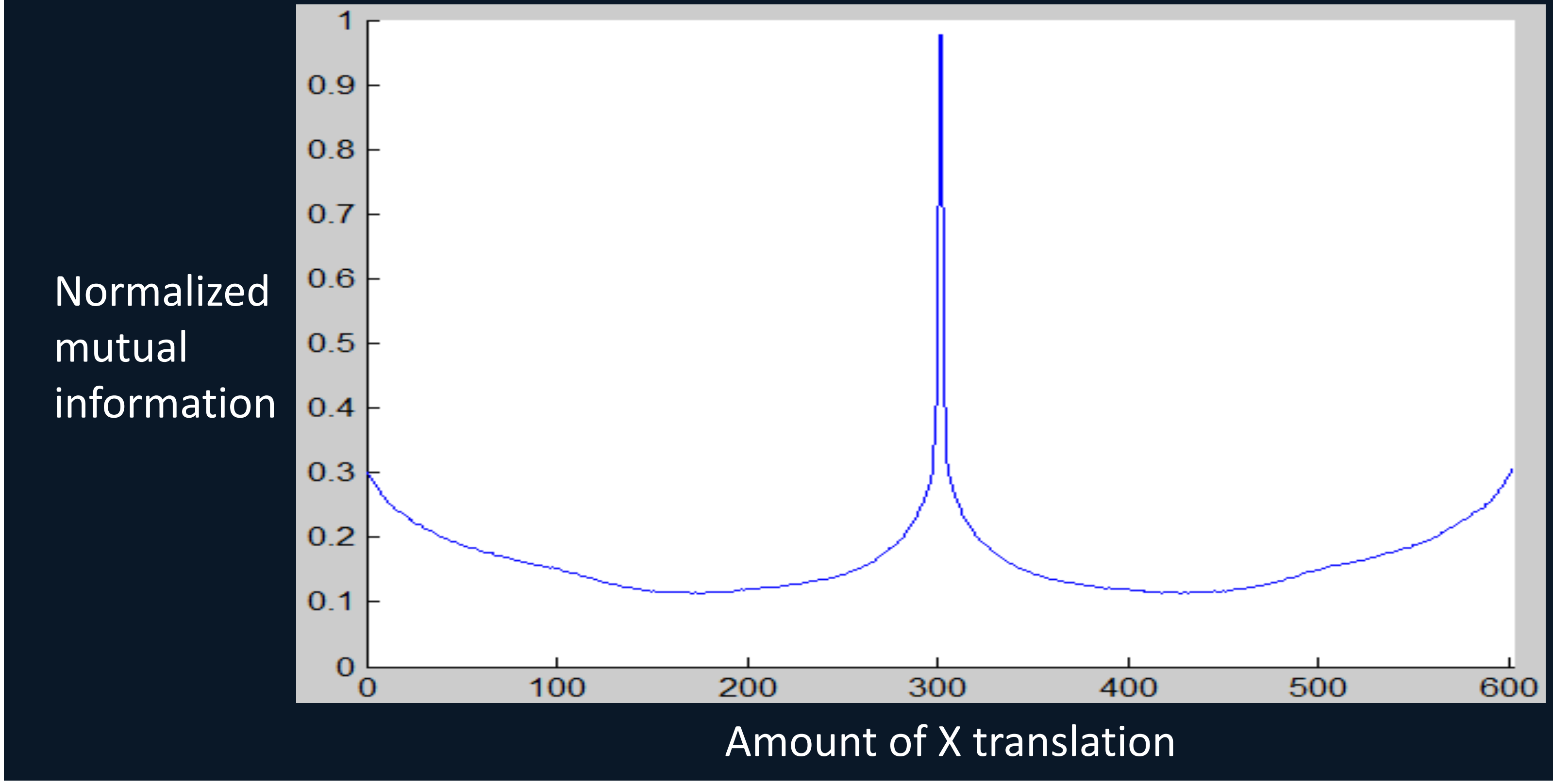

Y
X

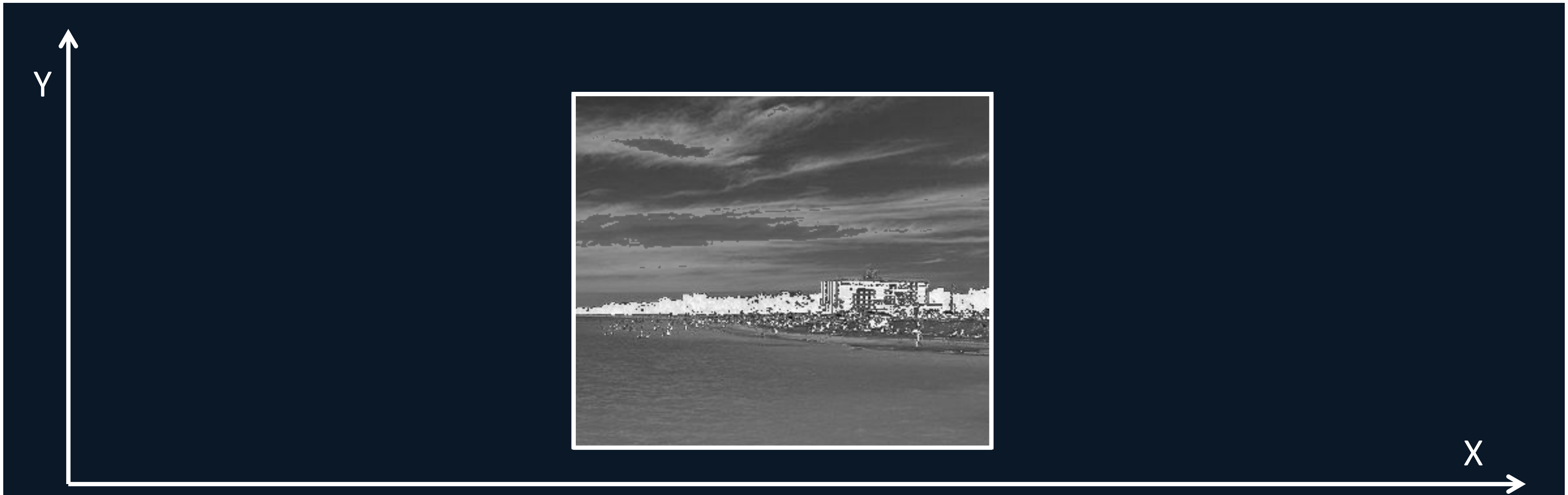

MSE
2500
2000
1500
1000
500
0
0
100
200
300
400
500
600
Amount of X translation


Y
X

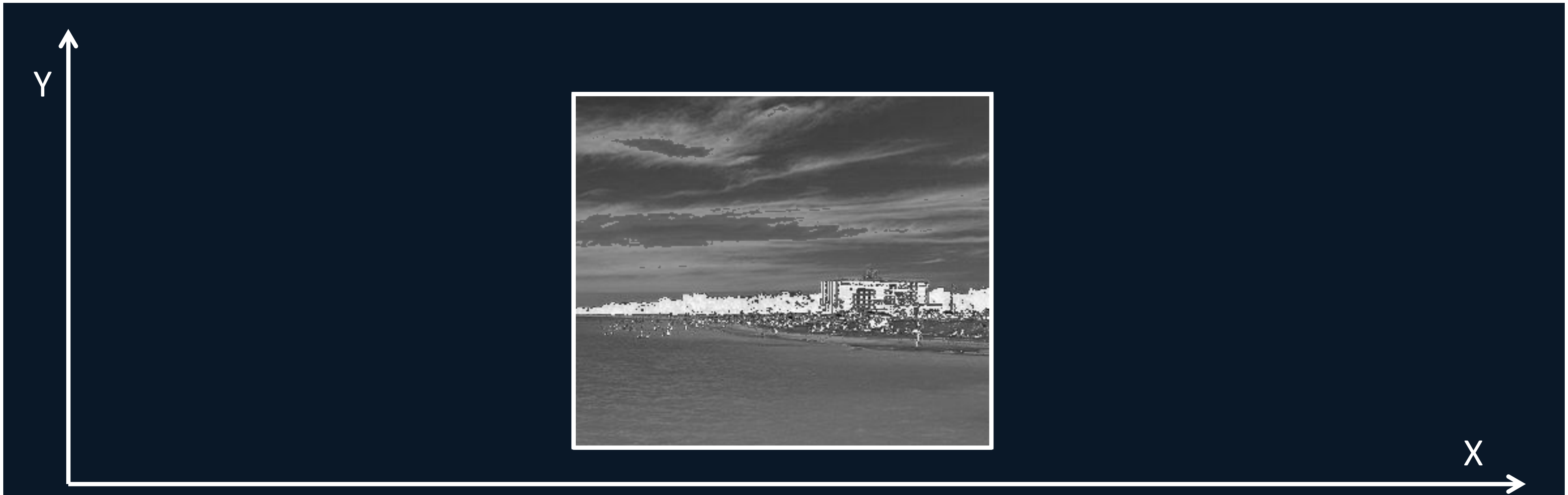

NCC
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
0
100
200
300
400
500
600
Amount of X translation

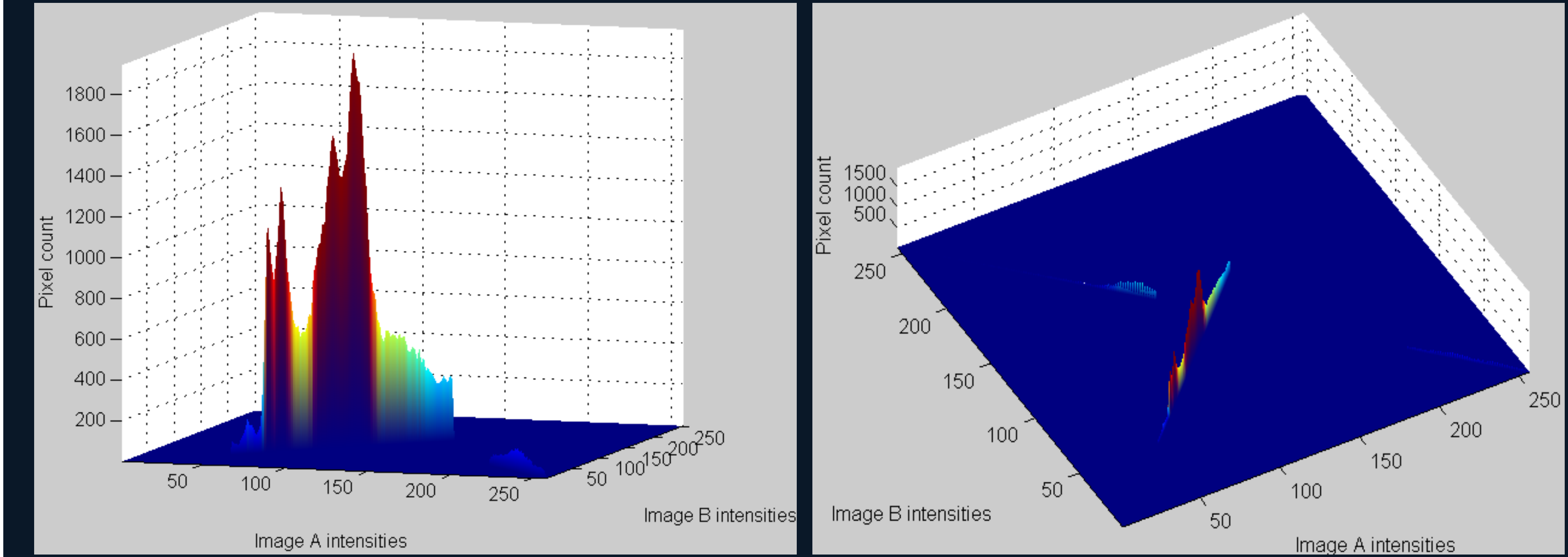

Y
X


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

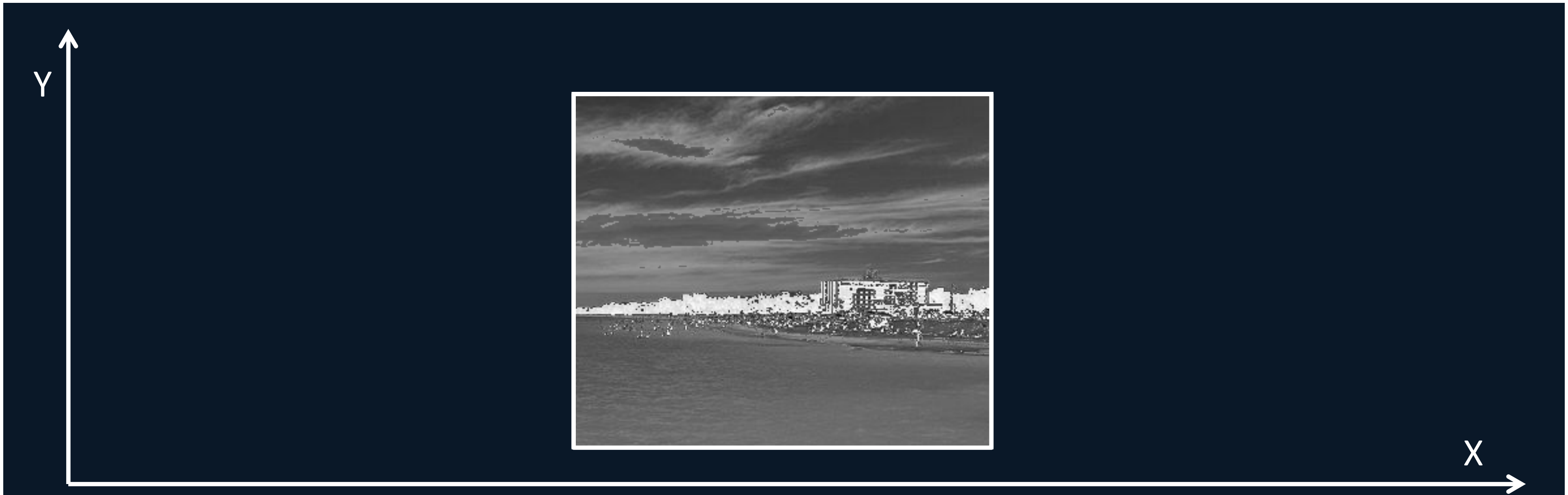


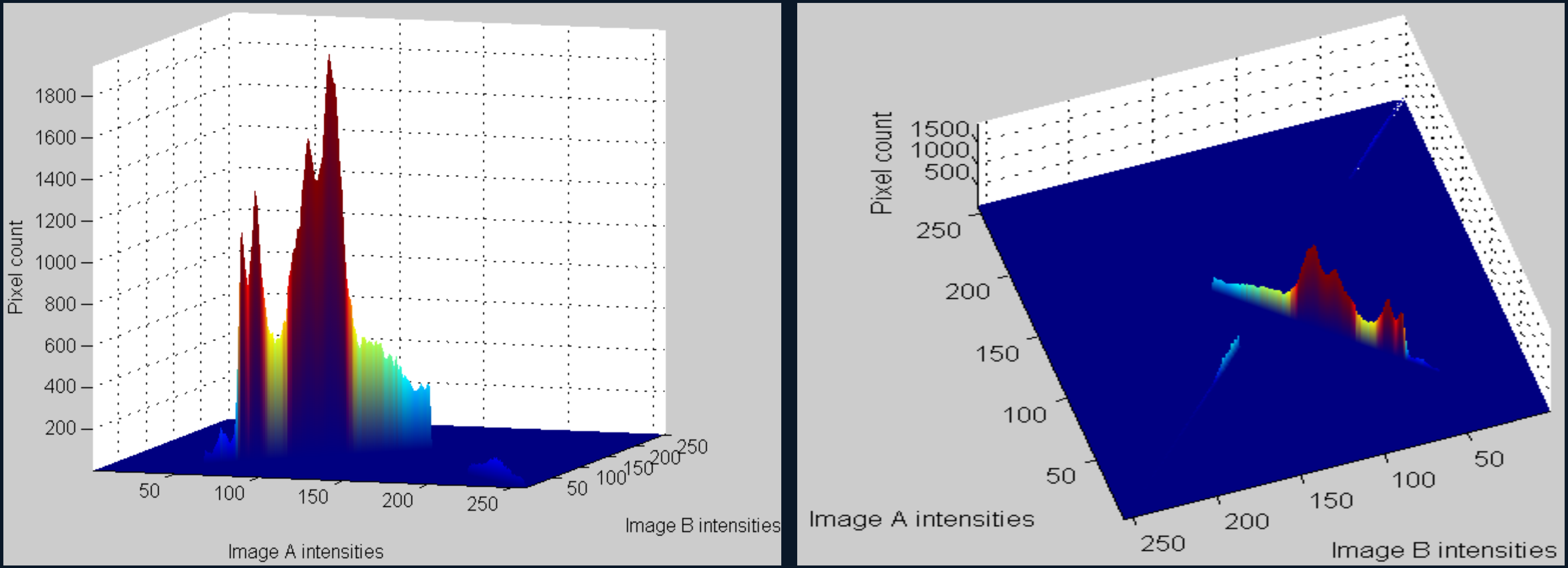
We begin with the situation where the images are perfectly aligned.



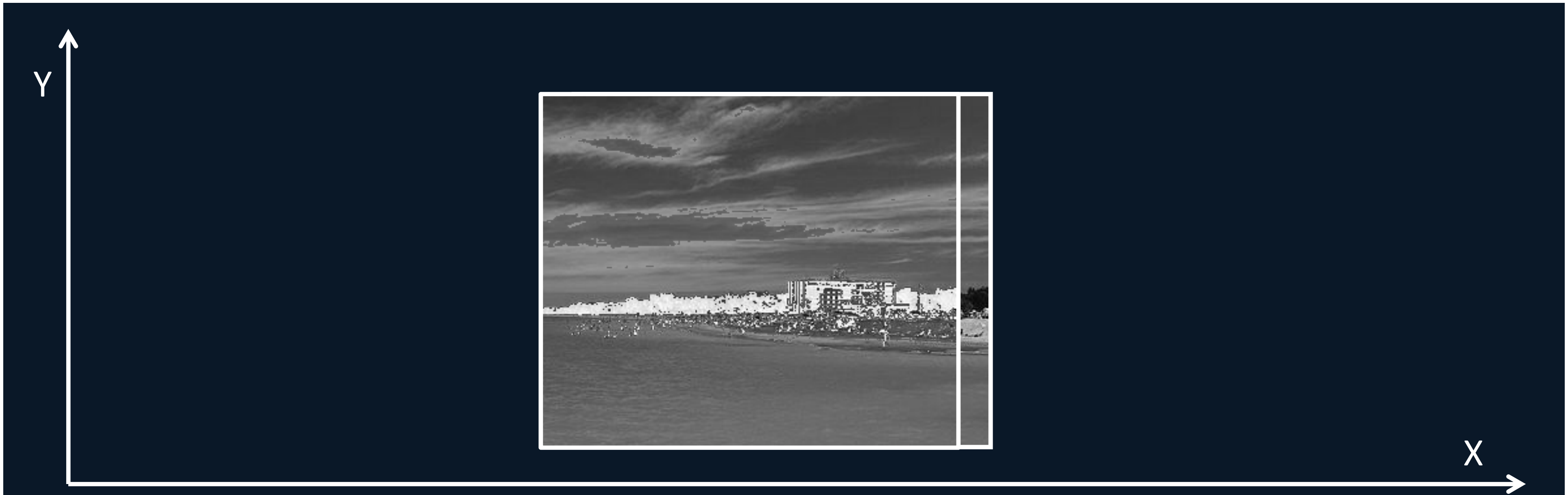


Notice that there are compact ridges, but they do not lie along any one line.

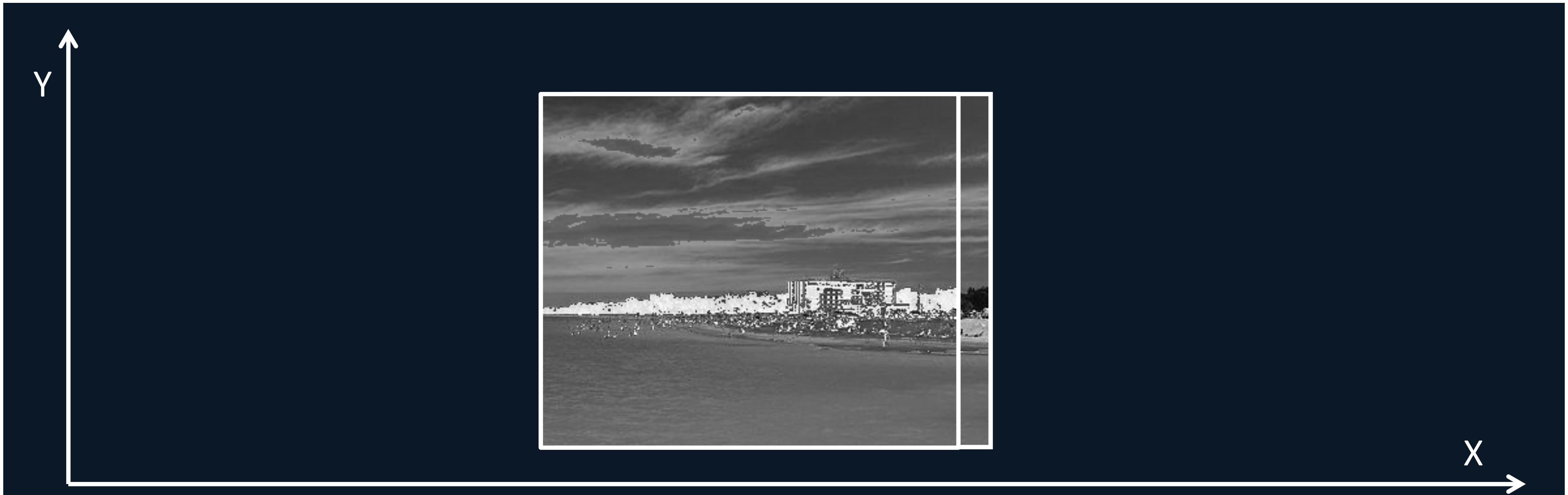


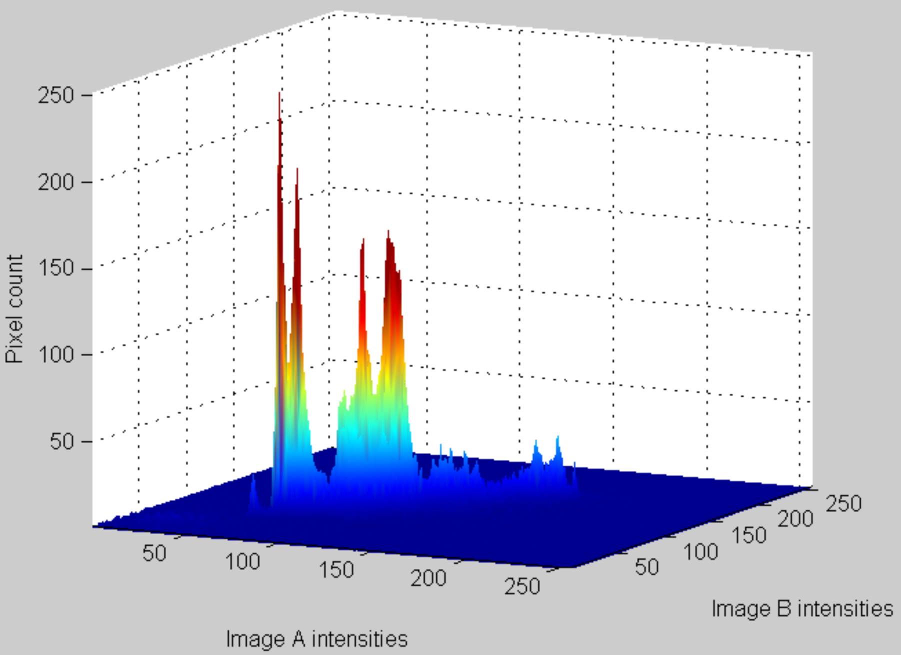


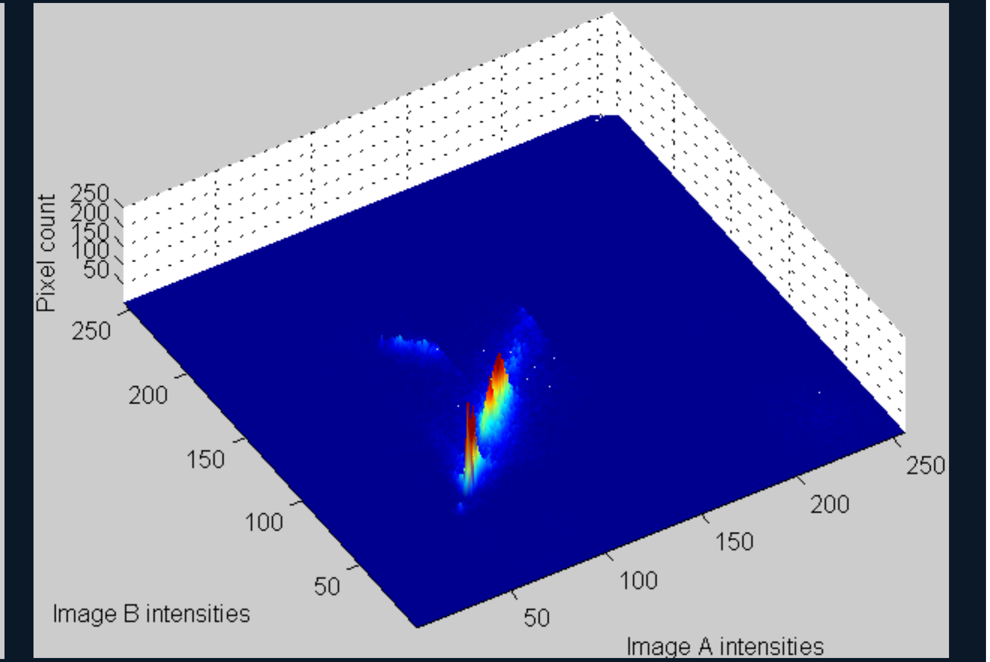
Another view (right).



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

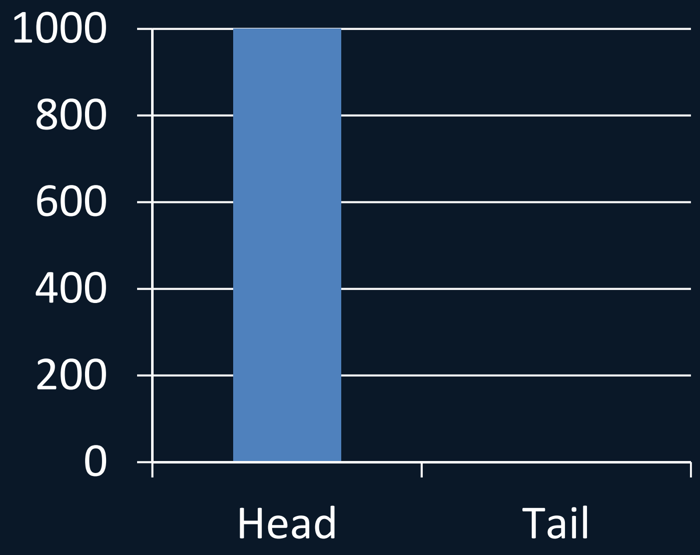






Notice that both ridges are starting to flatten.

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

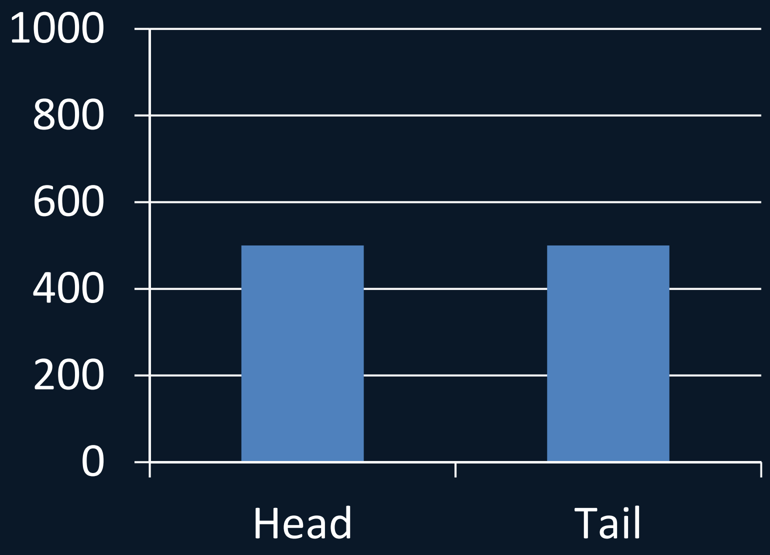


Completely unfair coin:

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?

↓

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

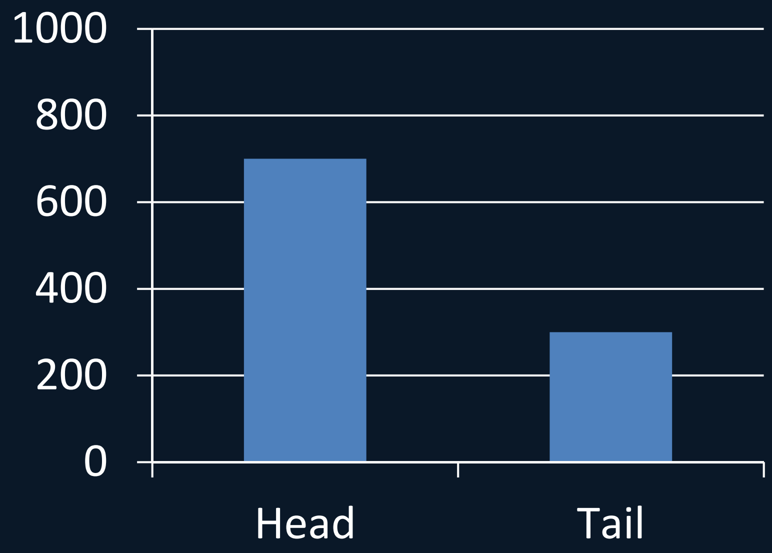
How well can you predict the 1000th coin flip from the first 999 flips?

↓

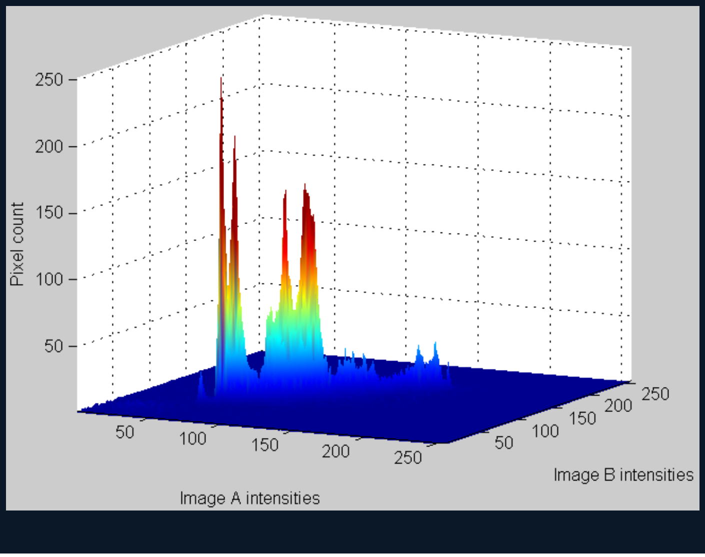
How well can you predict the intensity contents of one image based on those in another aligned image?

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.

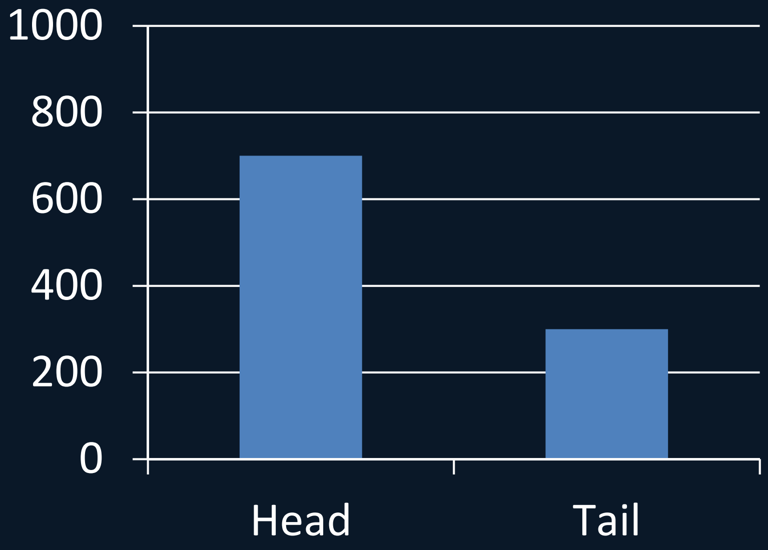


To measure the mutual information contained in a pair of aligned images, we need to measure something about this histogram.

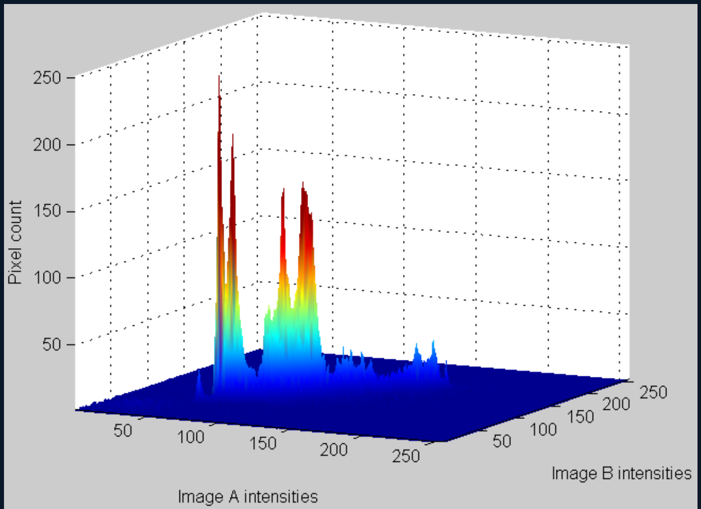


How much “mutual information” is there in a pair of aligned images?

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

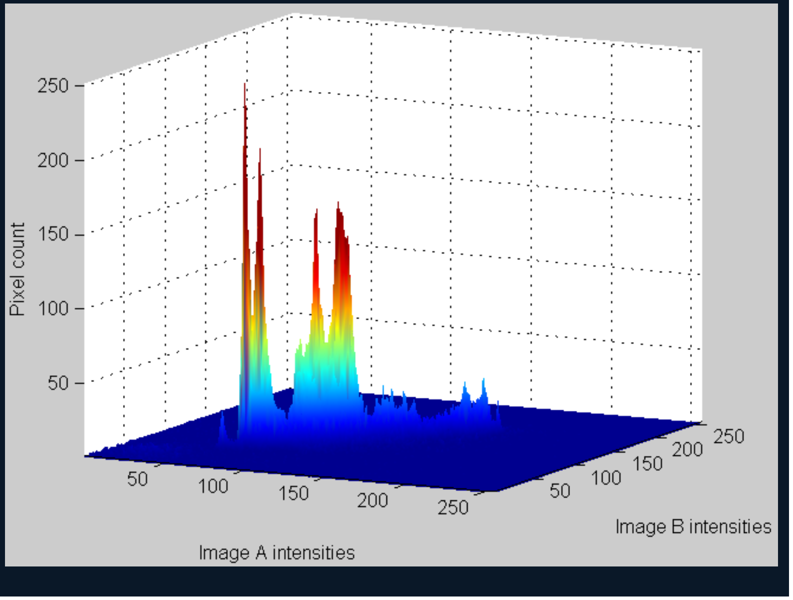
$p(x)$: Probability of x.

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

The 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

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

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

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

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

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

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, $H(A) + H(B) = 0$.

Mutual information

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

Remember that our optimizer will try to *maximize* $MI(A,B)$.

So, the optimizer will be encouraged to find solutions where the images overlap, at least a bit, to get a value of $H(A) + H(B)$ that is larger than 0.

So, why are we subtracting $H(A,B)$ in the above equation?

Mutual information

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

Remember that $H(A,B)$ will be closer to 0 when the joint intensity histogram of the two images is compact with large peaks; i.e. when the images are well aligned.

The more misaligned the images become,
the larger $H(A,B)$ becomes,
decreasing $MI(A,B)$.

This is exactly what we want – a large $MI(A,B)$ should reflect well-aligned images.

Mutual information

$$\boxed{MI(A,B)} = \boxed{H(A) + H(B)} - \boxed{H(A,B)}$$

Optimizer will try to maximize this

Zero if no image overlap, positive if overlap

This penalty is smaller if overlap regions match better

So, the optimizer is encouraged to find a solution where the two images overlap, and where the joint information entropy in the overlap region is minimal.

Describing the terms slightly differently..

Mutual information

$$\boxed{MI(A,B)} = \boxed{H(A) + H(B)} - \boxed{H(A,B)}$$

Optimizer will try to maximize this

Optimizer will overlap images to make this positive

Optimizer will try to get good alignment to minimize the amount subtracted here.

There is a complementary way to look at this to make it easier to understand...

Mutual information

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

$H(A)$: The number of bits you need to transmit image A.

$H(B)$: The number of bits you need to transmit image B.

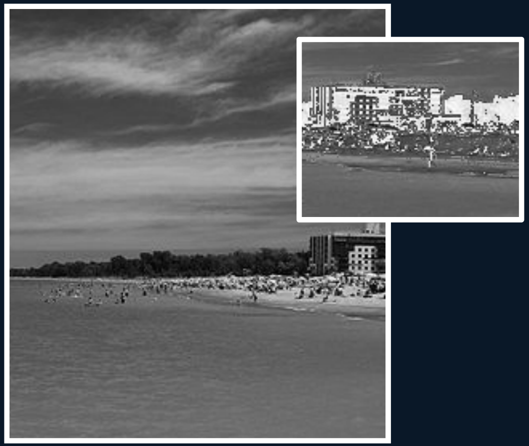
$H(A,B)$: The number of bits you need to transmit images A and B together.

If A and B are registered well, then B can be predicted from A, allowing compact transmission of A and B together with few bits. Thus $H(A,B)$ is small, resulting in large $MI(A,B)$.

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

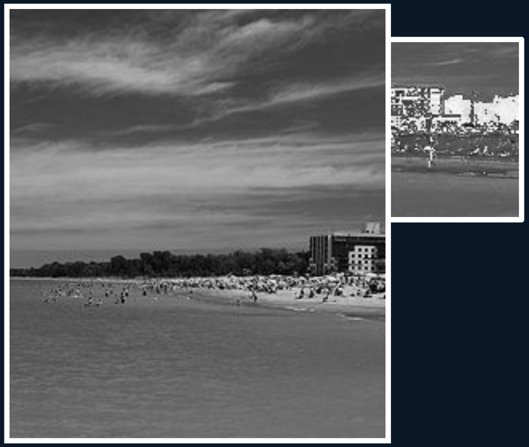


Alignment before registration

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

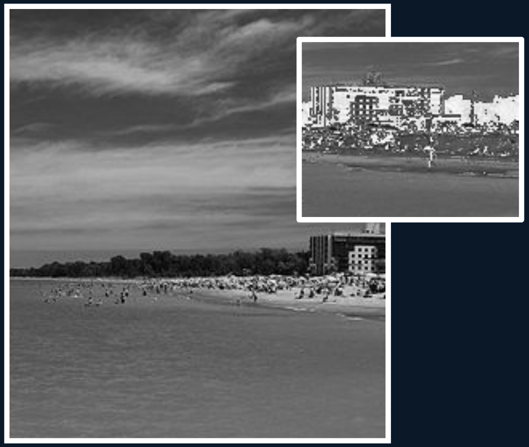


Alignment before registration

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

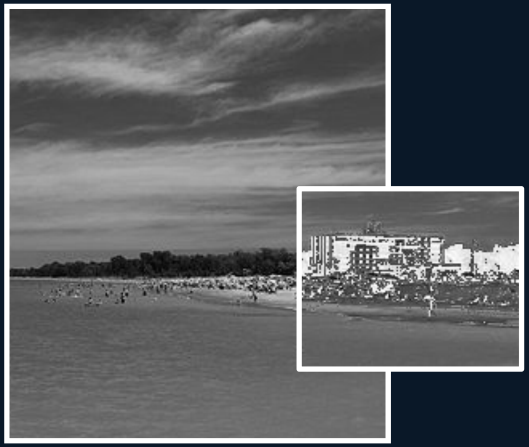


Alignment before registration

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

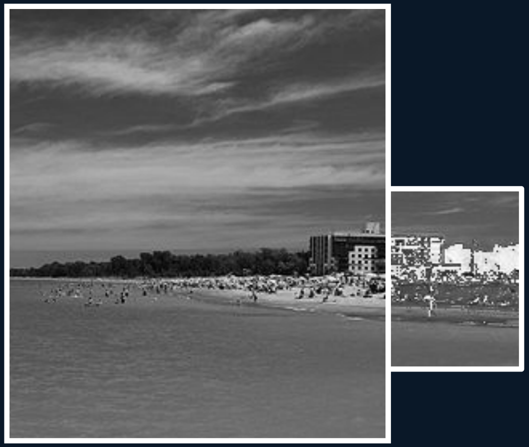


Alignment after registration (desired)

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

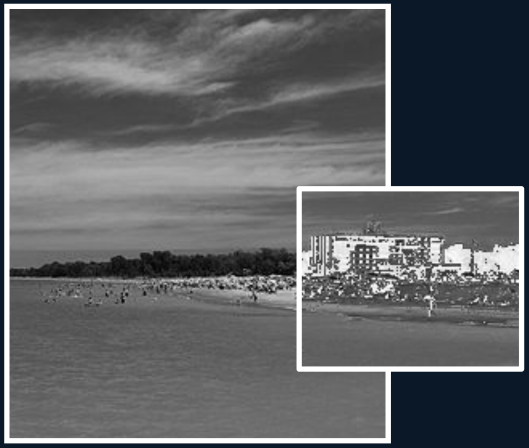


Alignment after registration (desired)

Mutual information

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

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:

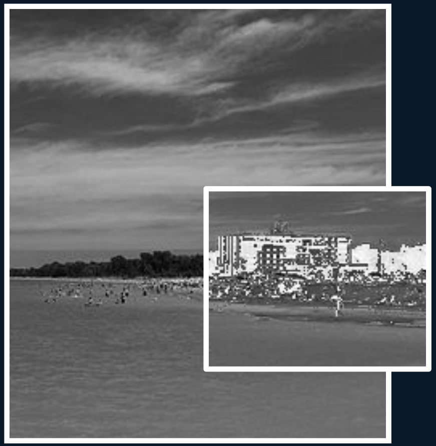


Alignment after registration (desired)

Mutual information

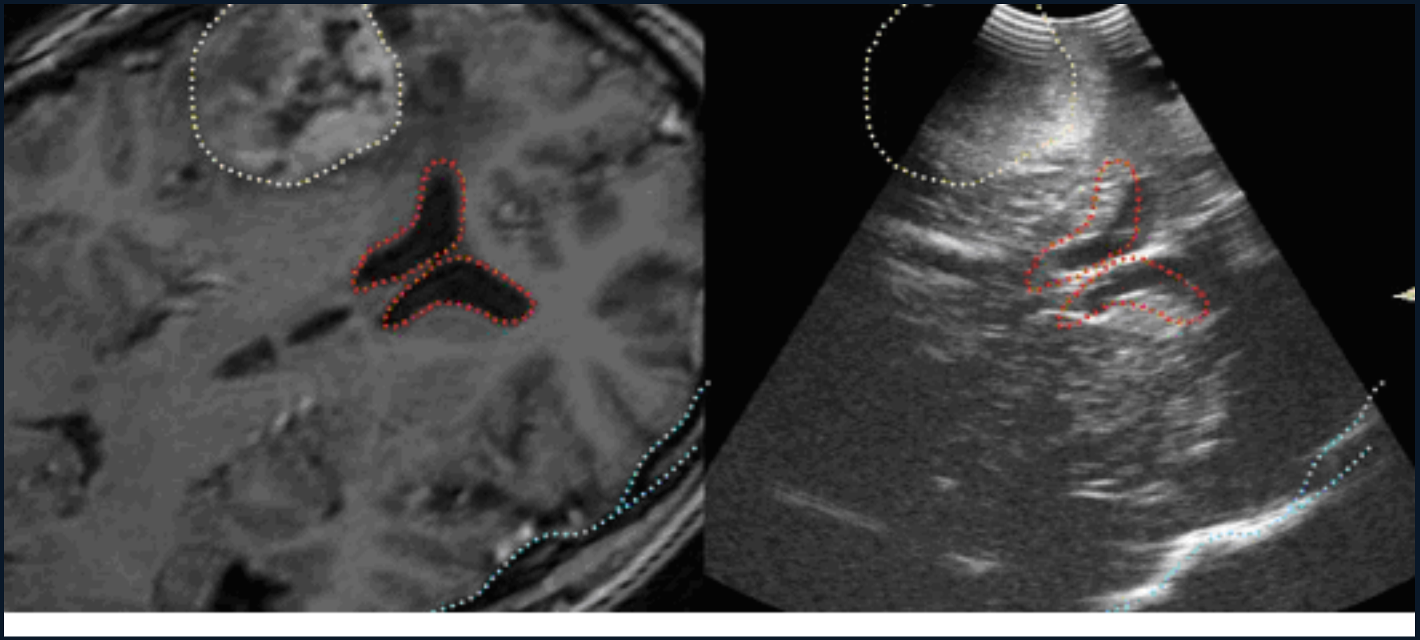
$$MI(A,B) = H(A) + H(B) - H(A,B)$$

It turns out that the MI(A,B) metric wrongly encourages large image overlap in cases where the correct answer involves small image overlap. This is an example of such a case:



Alignment after registration (too much overlap)

Application: Brain surgery



- Pre-op MRI registered to intra-op ultrasound to compensate for brain shift/motion.
- Fields of view from multi-modality imaging often have regions of non-overlap.

T. Peters and K. Cleary. Image-Guided Interventions. Springer, 2008.

Normalized mutual information

$$NMI(A,B) = \frac{H(A) + H(B)}{H(A,B)}$$

Dividing by H(A,B) instead alleviates this situation. This metric is widely used and is referred to as *normalized mutual information*.

C. Studholme, D. Hill, D. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment”, Pattern Recognition 32, 1999, 71-86.

Normalized mutual information

$$NMI(A,B) = \frac{H(A) + H(B)}{H(A,B)}$$

This metric ranges from 1 (worst match) to 2 (best match).

In the worst case, we need the same number of bits to transmit both images together as we do to transmit them separately, so the numerator and denominator are the same.

C. Studholme, D. Hill, D. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment”, Pattern Recognition 32, 1999, 71-86.

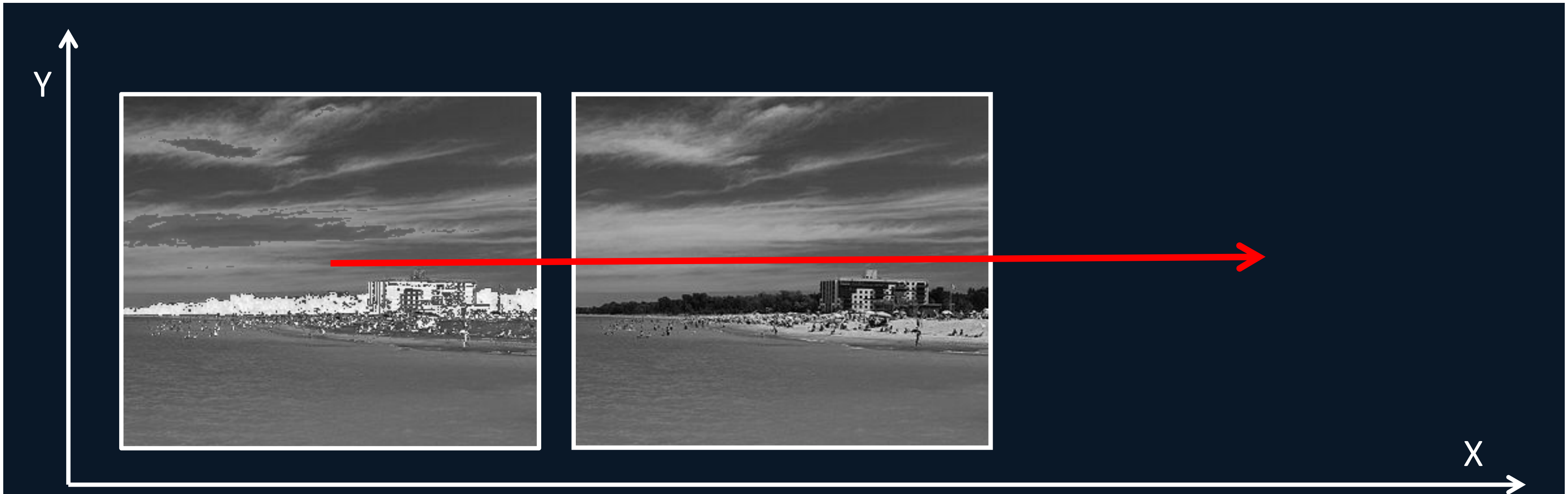
Normalized mutual information

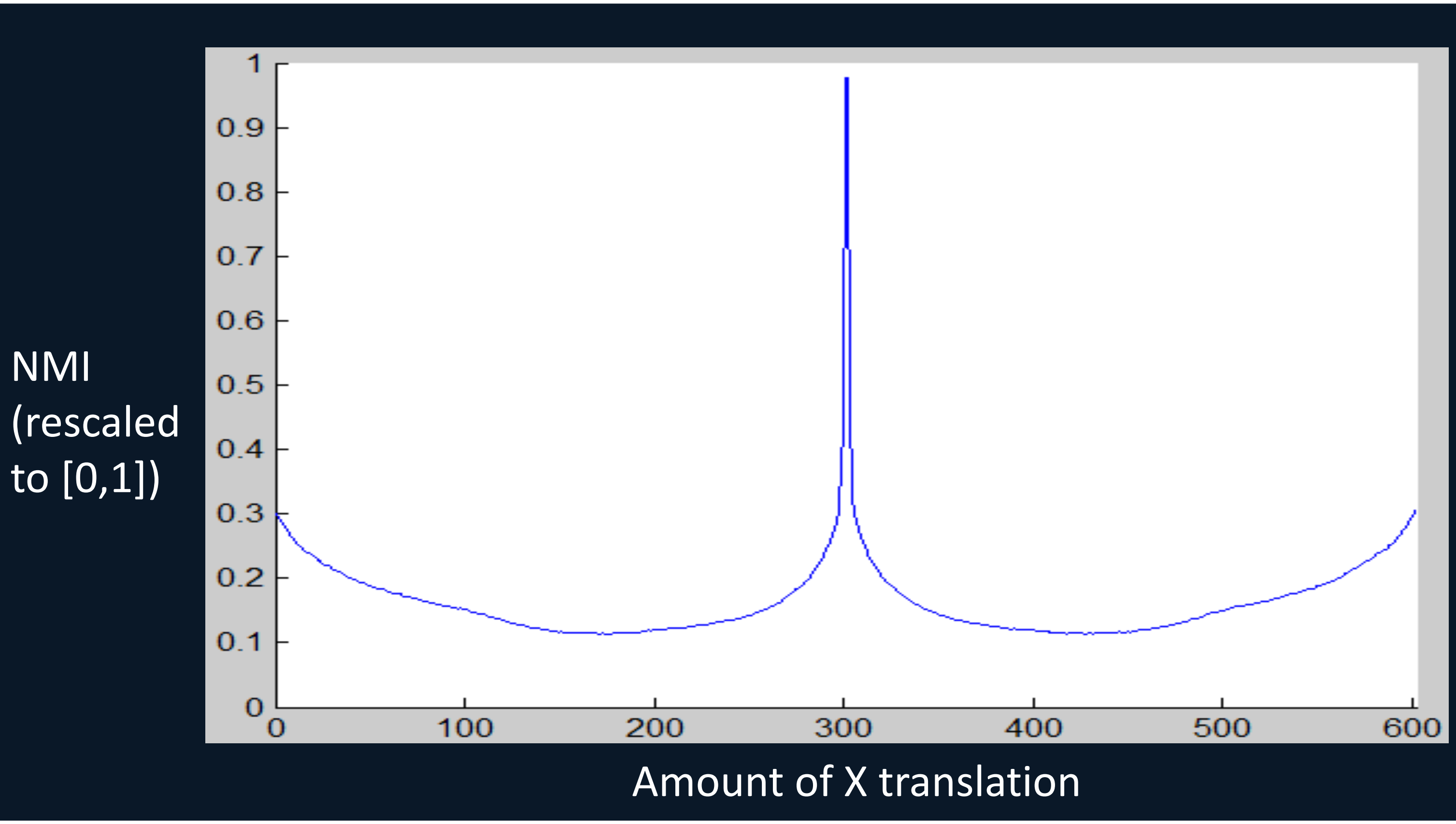
$$NMI(A,B) = \frac{H(A) + H(B)}{H(A,B)}$$

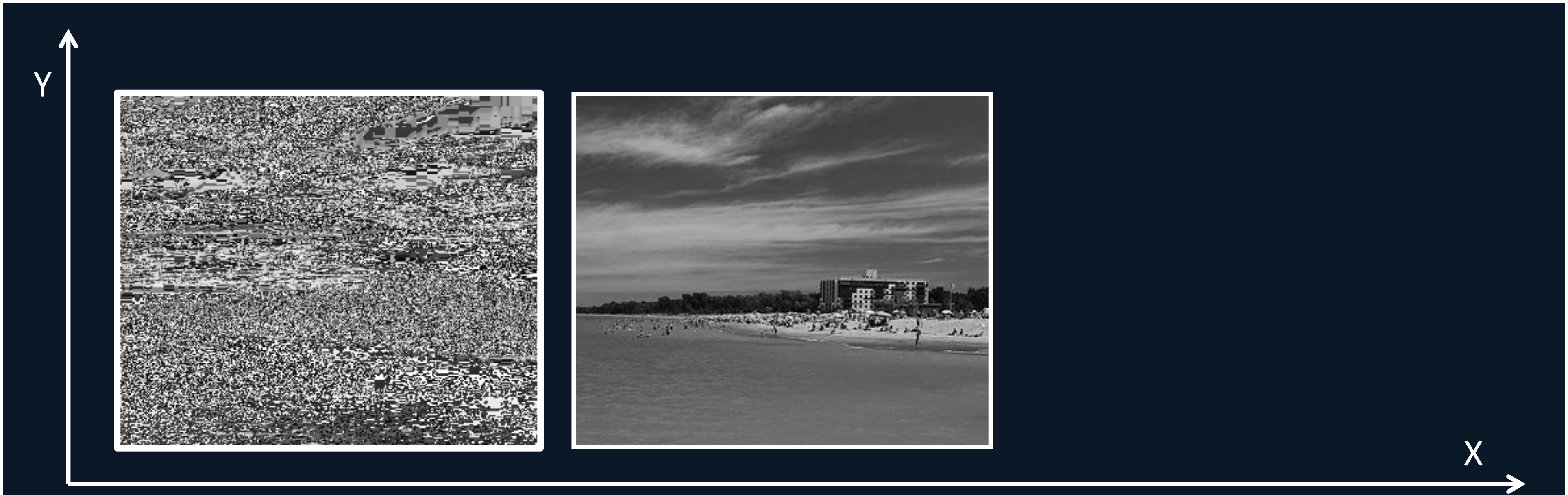
This metric ranges from 1 (worst match) to 2 (best match).

In the best case, it turns out that we need half the number of bits to transmit both images together (because they are a good match and therefore there is information redundancy) than we do to transmit them separately, yielding a best-case value of $NMI(A,B) \ = \ 2$.

C. Studholme, D. Hill, D. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment”, Pattern Recognition 32, 1999, 71-86.


Y
X

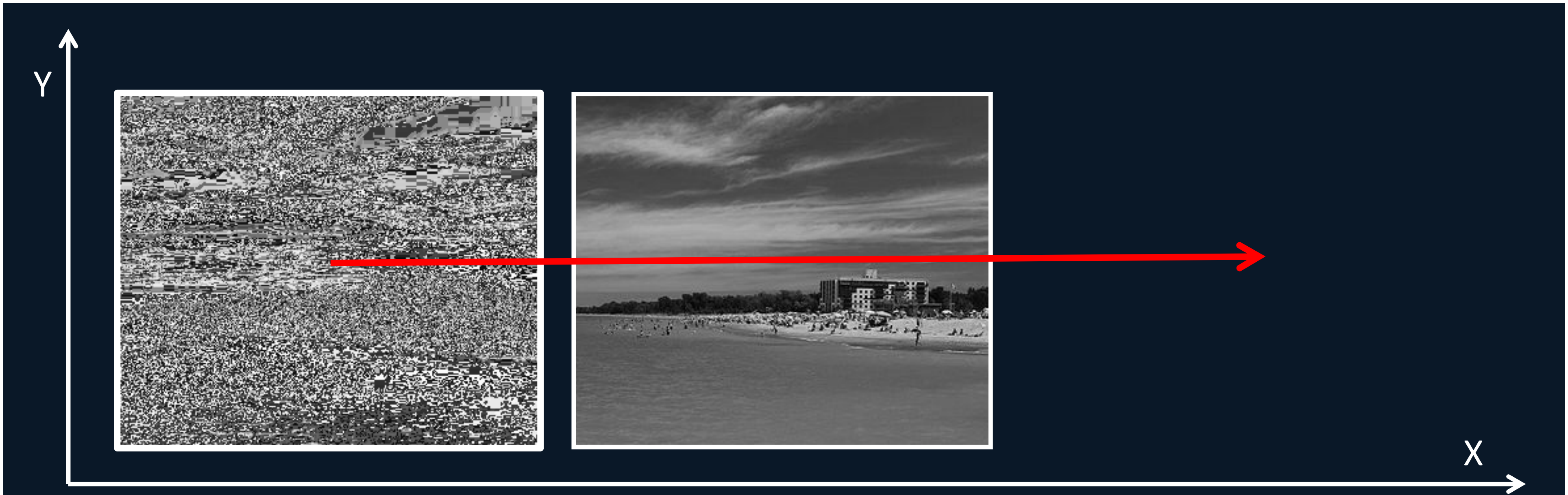

NMI (rescaled to [0,1])
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Amount of X translation

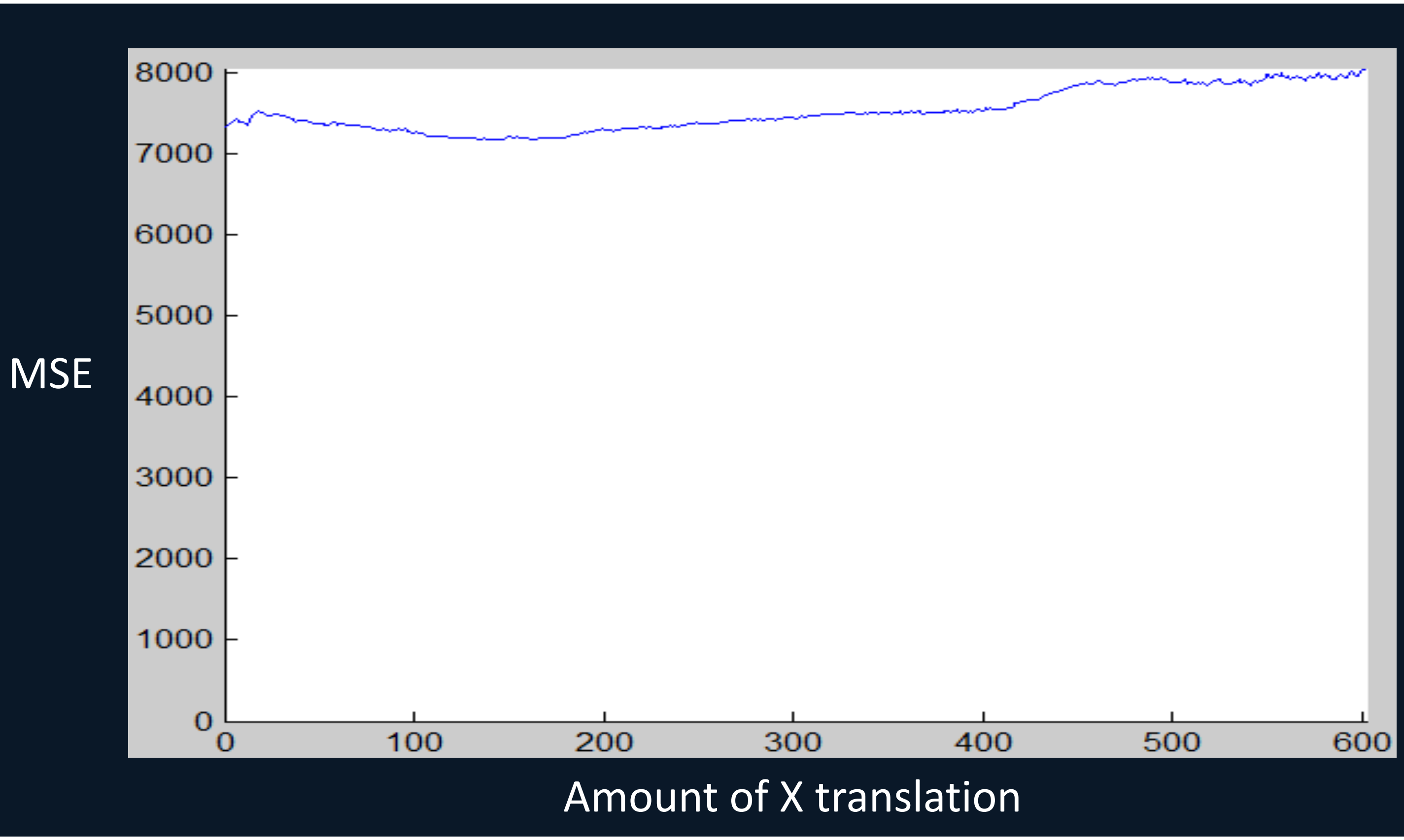


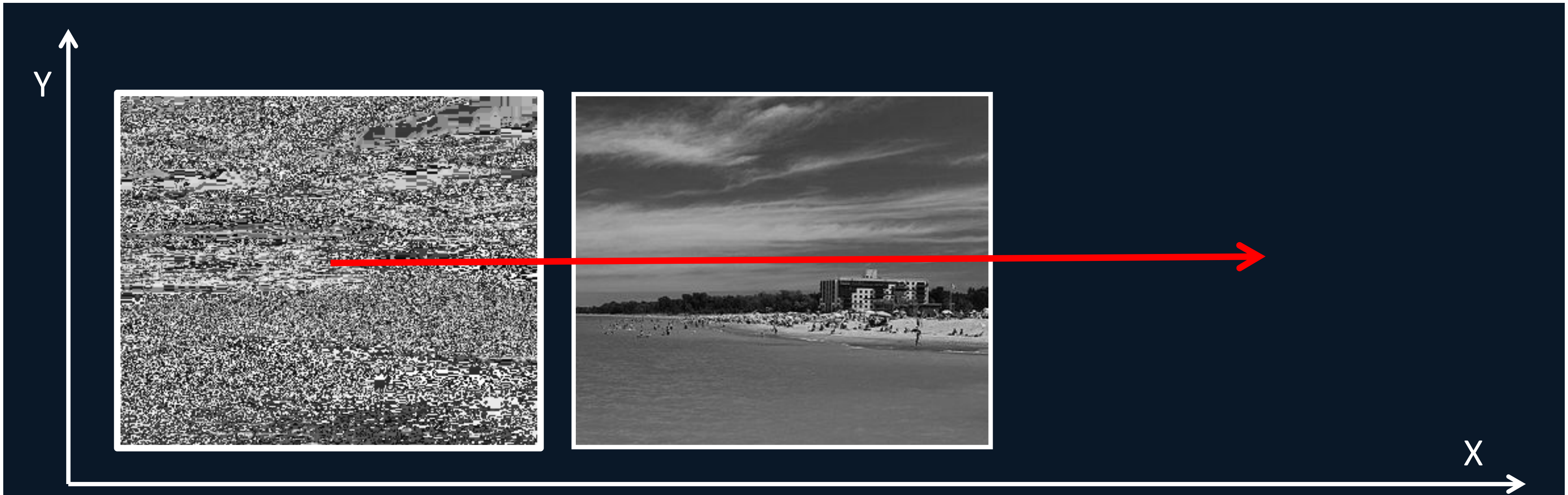
Let’s see how robust NMI is. I created a random intensity map and transformed the original image according to it.

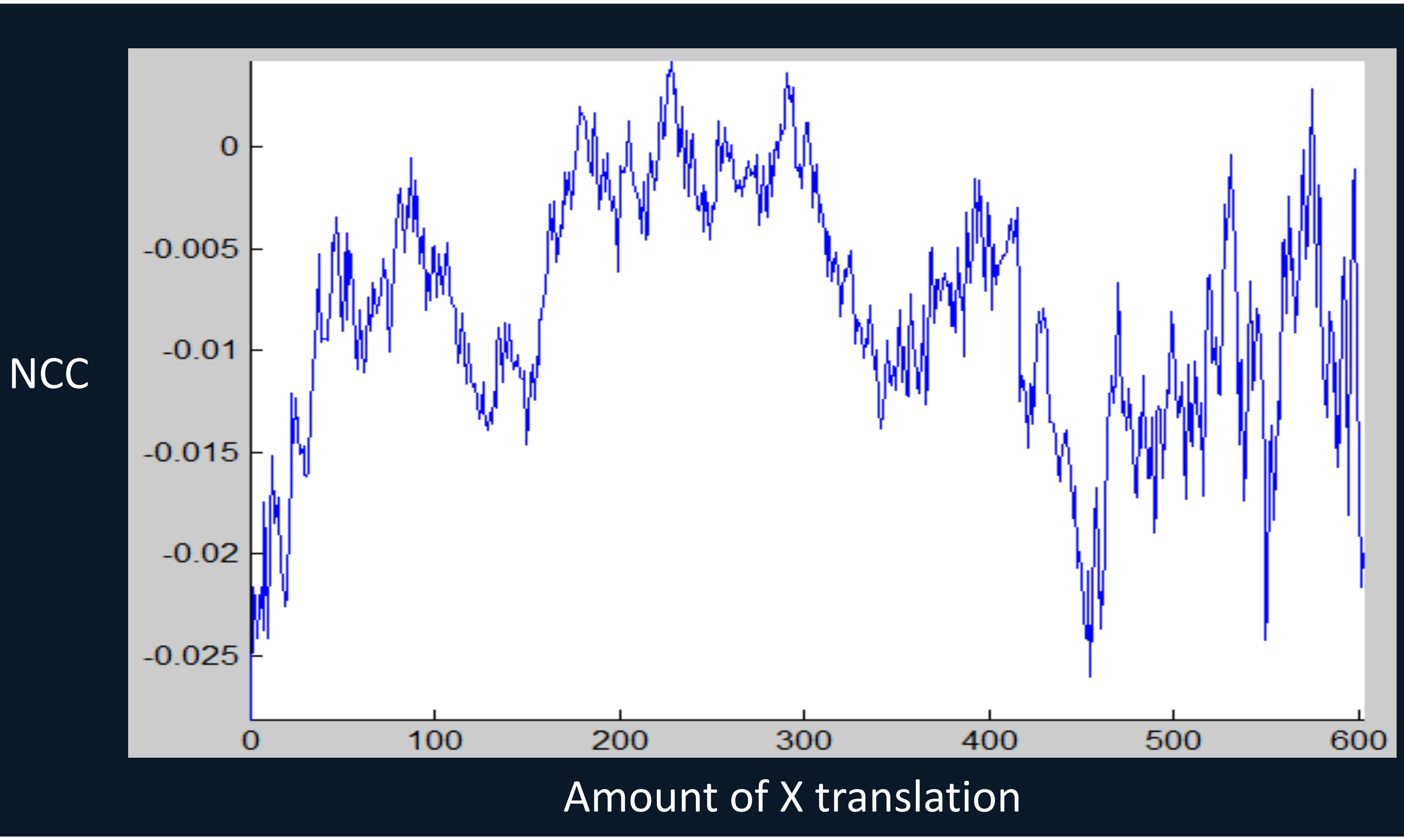
E.g. if the original image had 5 different intensity values, the random intensity map could look like this:
1 → 3, 2 → 5, 3 → 1, 4 → 2, 5 → 4

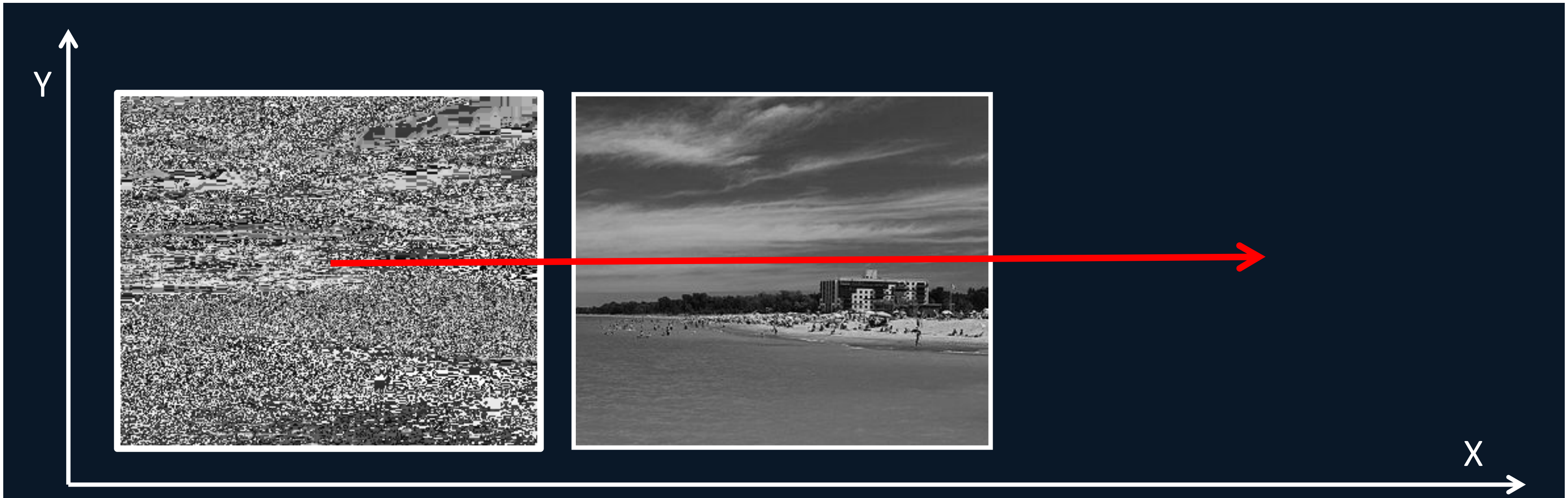
So I would change all pixels with intensity 1 in the original image to have intensity 3, all with intensity 2 to have intensity 5, etc. I did this with a 256-level random intensity map to generate the above.

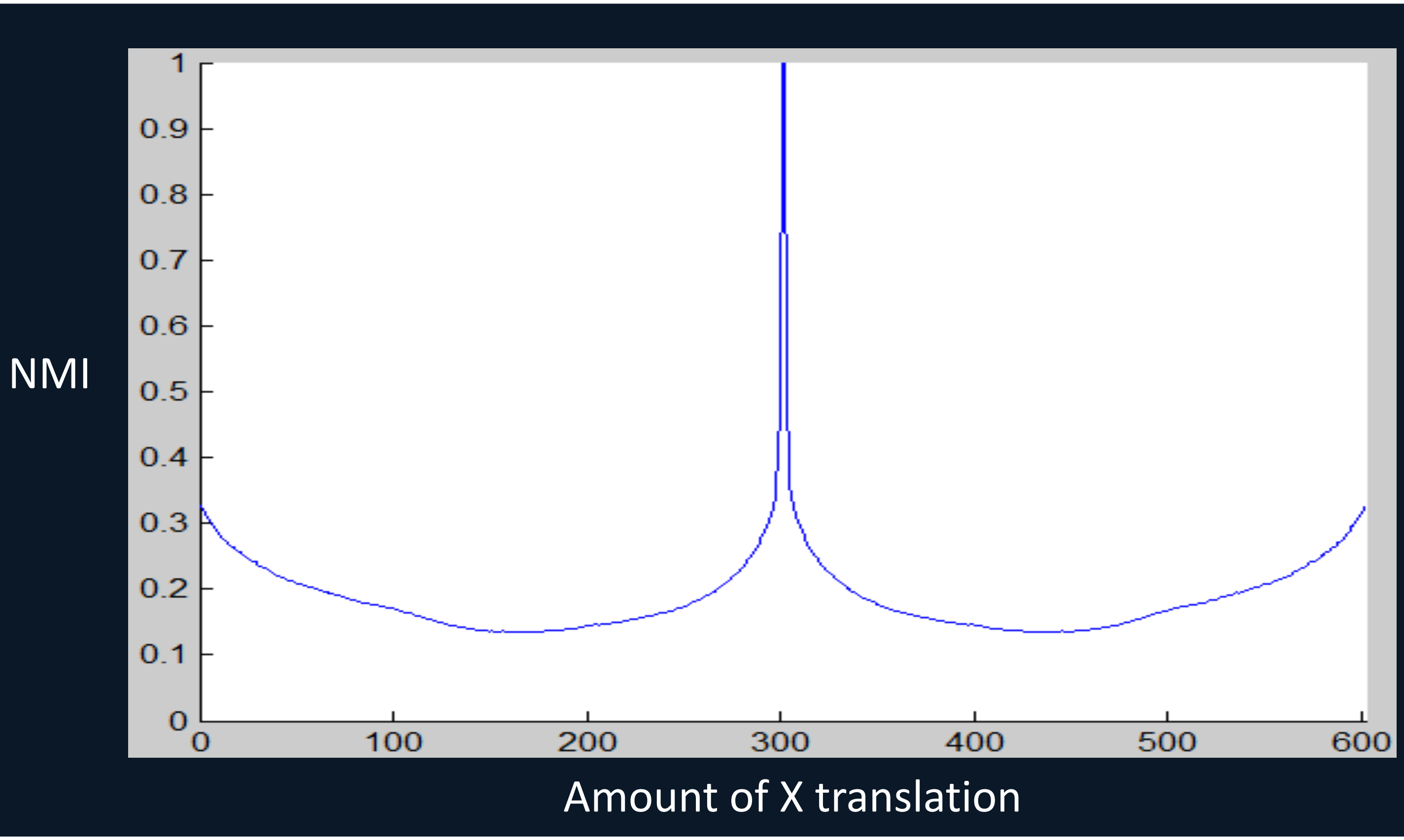

Y
X

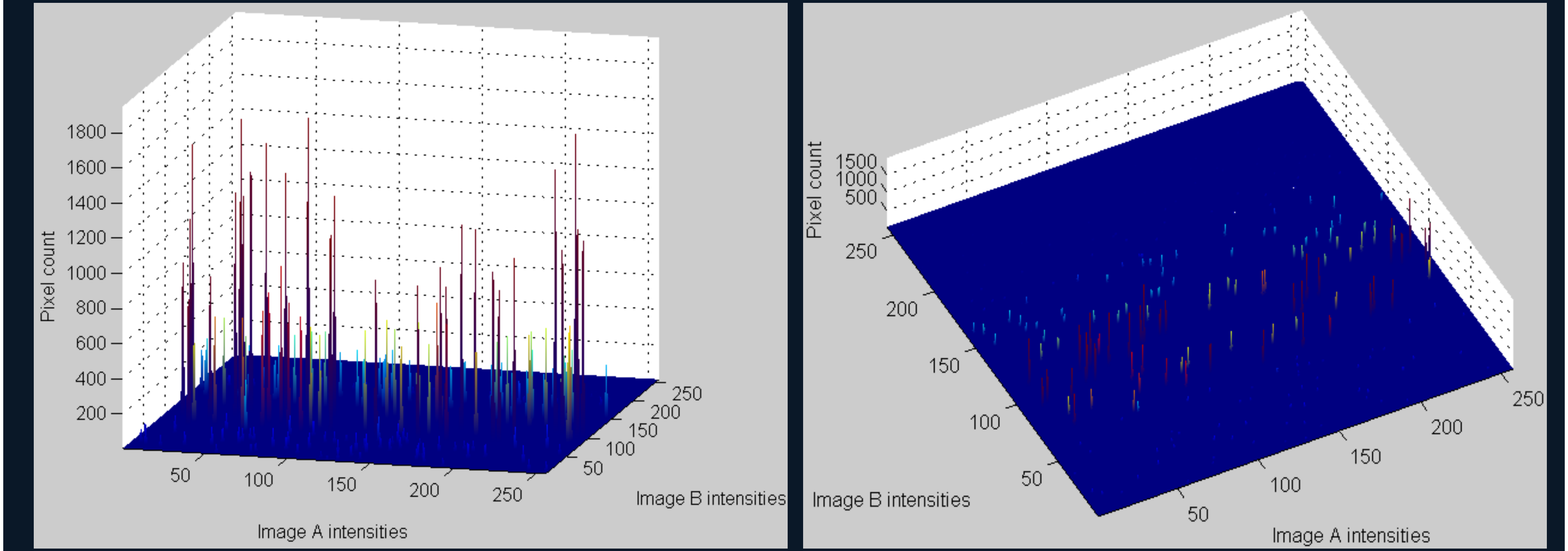

MSE
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
Amount of X translation


Y
X

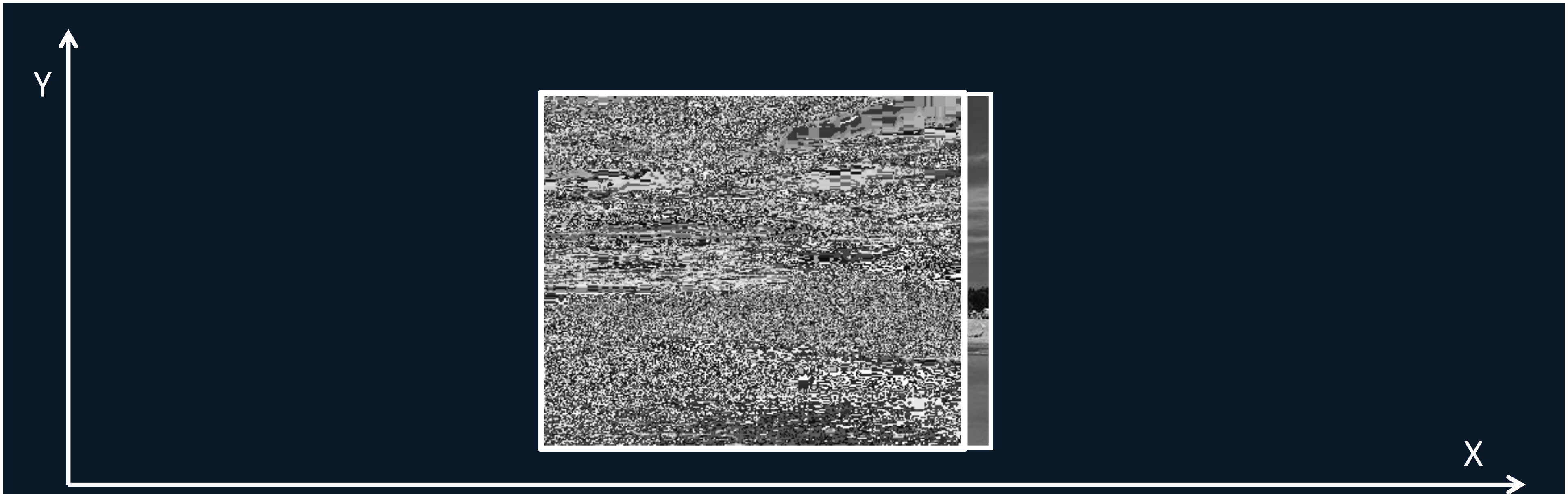

NCC
0
-0.005
-0.01
-0.015
-0.02
-0.025
0
100
200
300
400
500
600
Amount of X translation

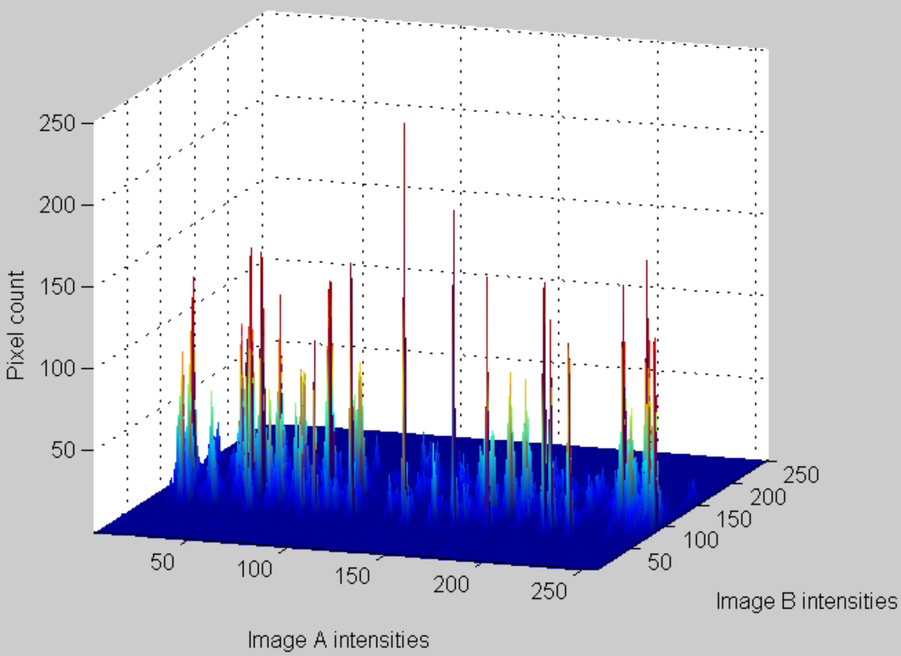

Y
X

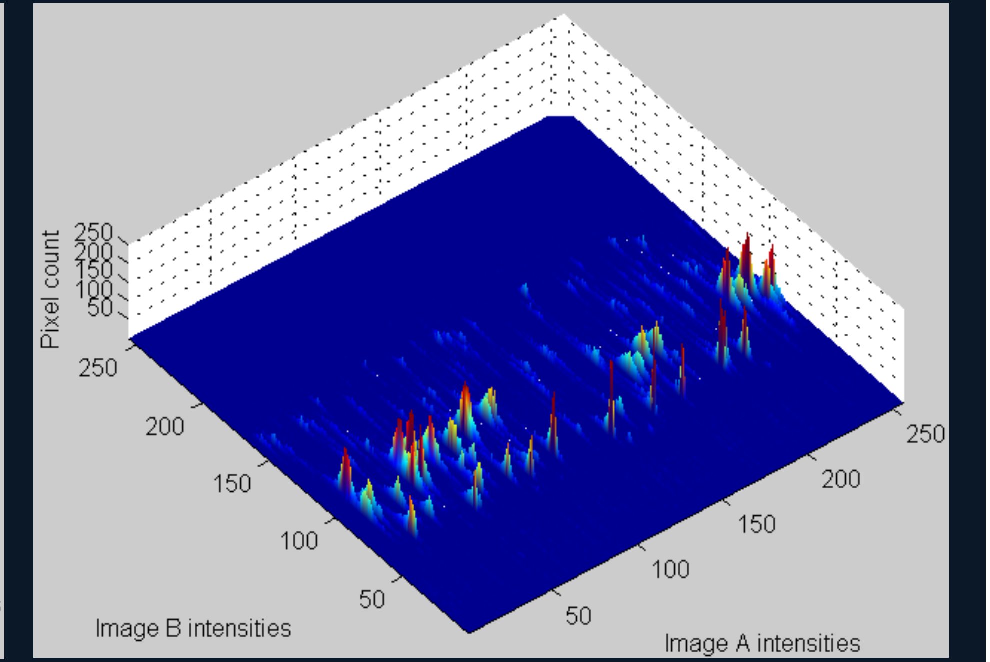

NMI
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
Amount of X translation



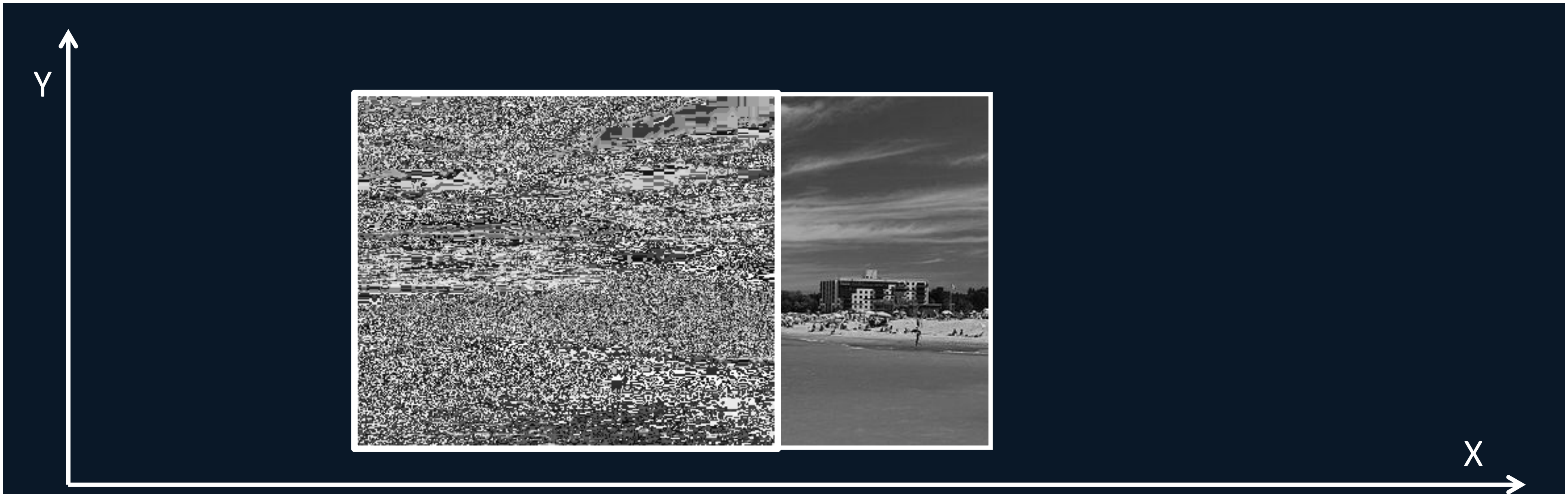
Although the intensity mapping was random, it was consistent, so a perfect alignment produces a histogram with compact tall peaks.

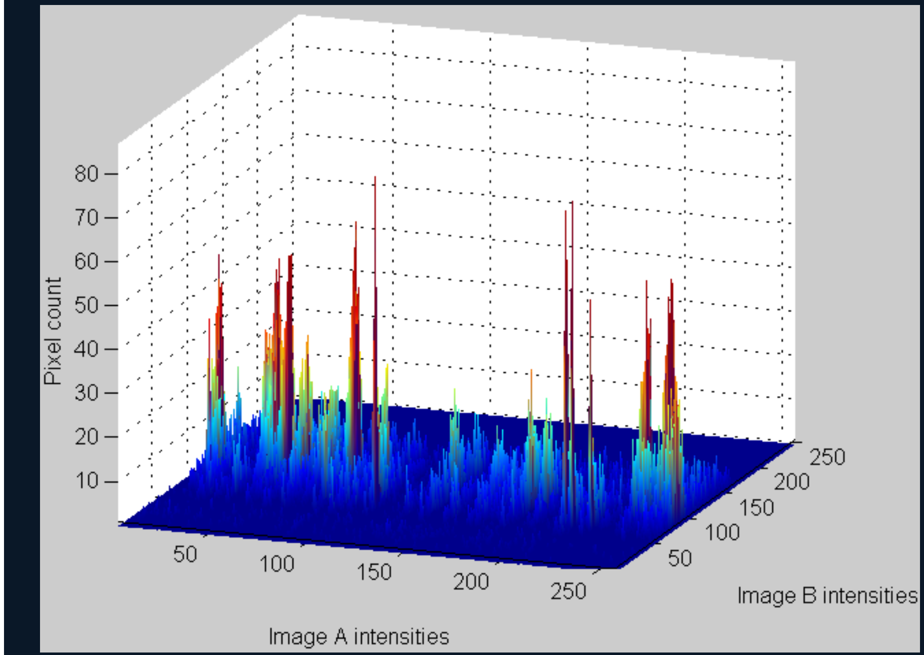


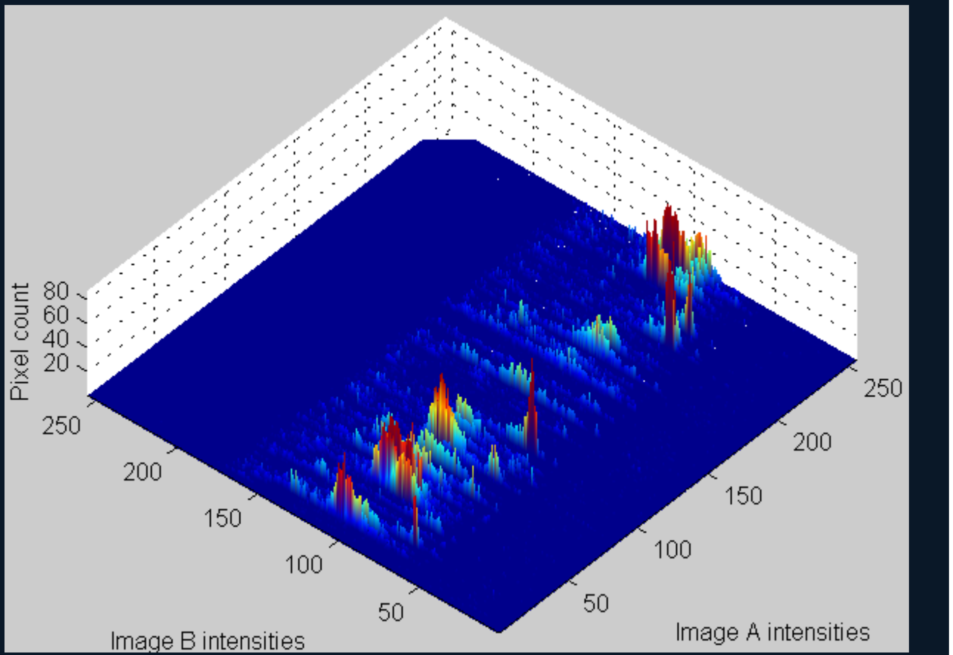




Notice that all the spikes are starting to flatten and widen.

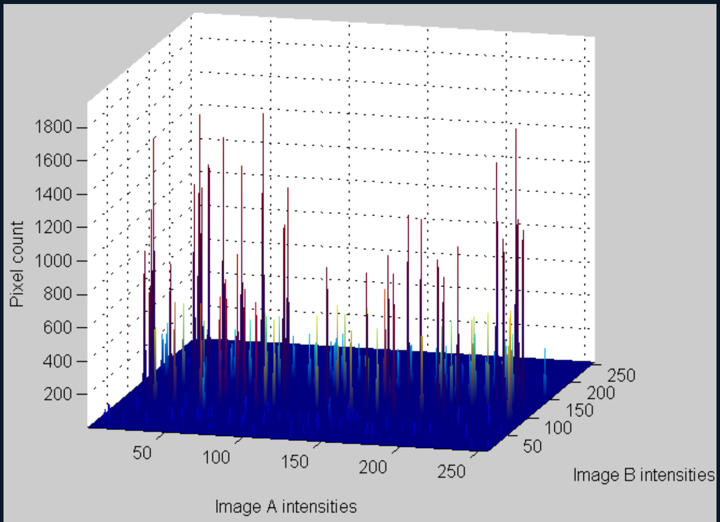


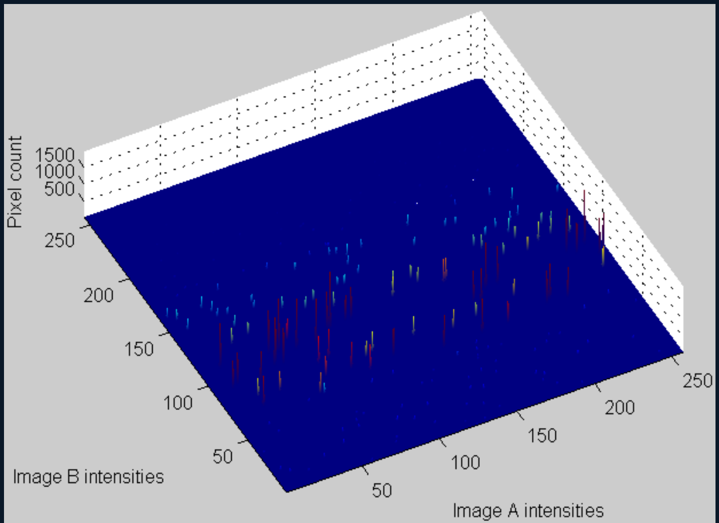




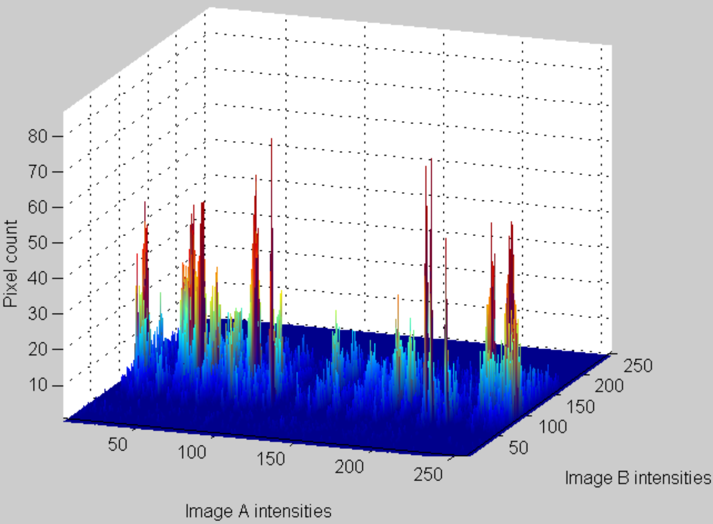
Further flattening with increasing misalignment.

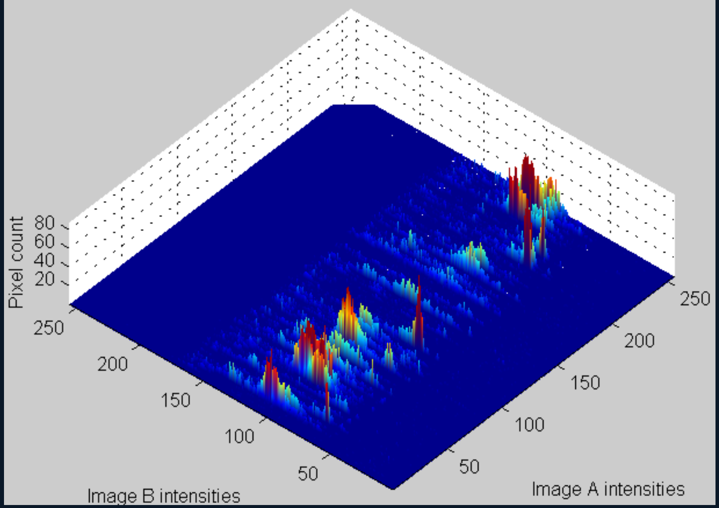
Good alignment



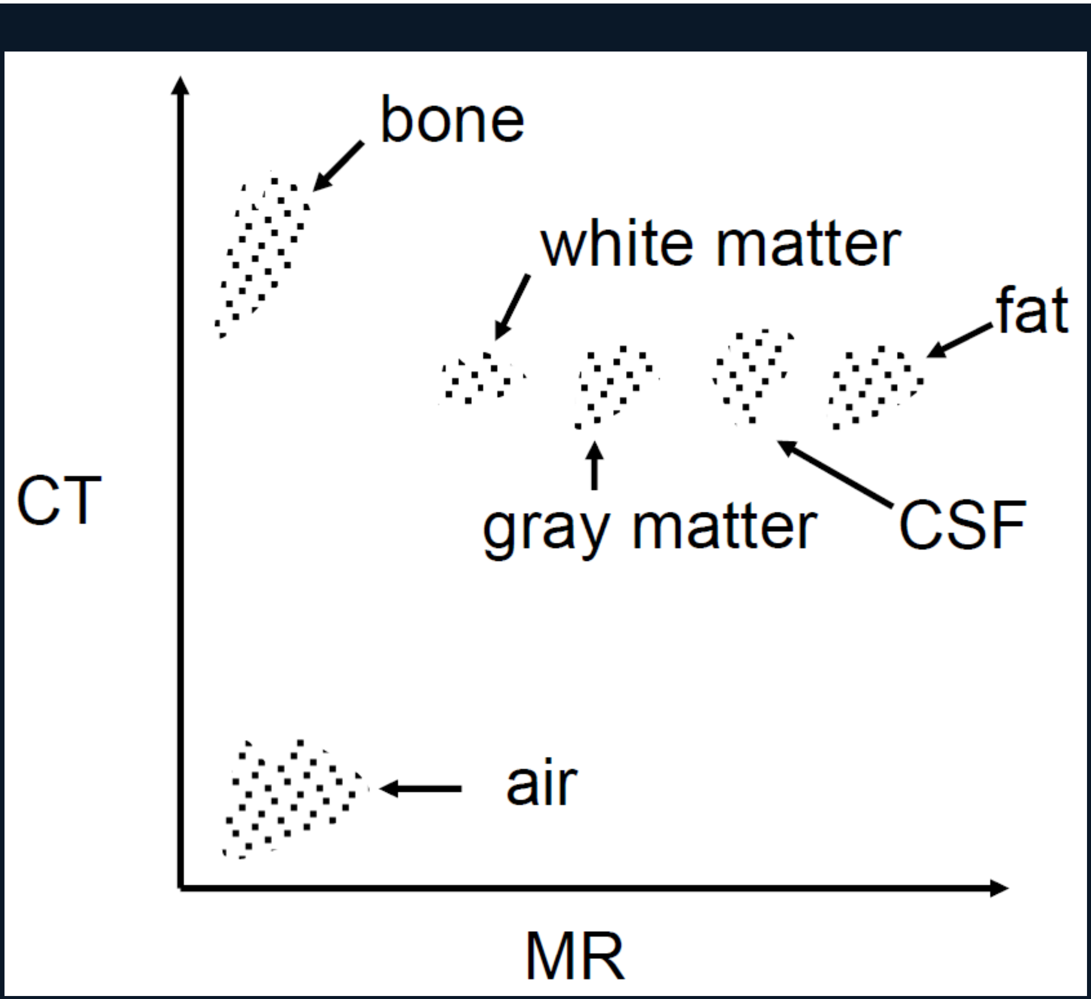


Bad alignment





Key observation: For mutual information to be a useful similarity metric, after alignment, the intensity values of image B must be predictable based on image A.



This is frequently (but not always) the case in medical images of different modalities. Note the complex but somewhat predictable intensity relationships between signal intensities of different tissue types on CT and MRI.

Image A

2	2	2
3	3	1
1	3	3

Image B

2	2	2
3	3	1
1	3	3

Image B intensities

Image A intensities	1	2	3	4	5
1	2	0	0	0	0
2	0	3	0	0	0
3	0	0	4	0	0
4	0	0	0	0	0
5	0	0	0	0	0

To understand this better, let’s return to this view of the joint intensity histogram for two small (3x3) images with intensities ranging from 1 through 5.

In all of the following examples, Image A and Image B are taken to be correctly aligned.

Image A

2	2	2
3	3	1
1	3	3

Image B

2	2	2
3	3	1
1	3	3

Image B intensities

Image A intensities

	1	2	3	4	5
1	2	0	0	0	0
2	0	3	0	0	0
3	0	0	4	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Remember what the situation looks like when the overlaid images are identical – a “ridge” along the diagonal.

I.e. joint intensity histogram is compact with tall peaks – low entropy, high mutual information.

Image A

2	2	2
3	3	1
1	3	3

Image B

5	5	5
1	1	3
3	1	1

	Image B intensities				
Image A intensities	1	2	3	4	5
1	0	0	2	0	0
2	0	0	0	0	3
3	4	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Imagine that my intensity map was:
1 → 3, 2 → 5, 3 → 1, 4 → 2, 5 → 4

This would make Image B as above and changes the joint intensity histogram as shown.

Image A

2	2	2
3	3	1
1	3	3

Image B

5	5	5
1	1	3
3	1	1

Image B intensities

Image A intensities

	1	2	3	4	5
1	0	0	2	0	0
2	0	0	0	0	3
3	4	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Joint intensity histogram is still compact with tall peaks – low entropy, high mutual information.

All that happened is the peaks moved to different places in the histogram.

Image A

2	2	2
3	3	1
1	3	3

Image B

2	1	2
4	3	5
1	2	3

Image B intensities

Image A intensities	1	2	3	4	5
1	1	0	0	0	1
2	1	2	0	0	0
3	0	1	2	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Now, instead, let’s make Image B identical to Image A, except we’ll make every second pixel (in blue above) a random intensity value between 1 and 5.

This is something like adding noise to an image.

Image A

2	2	2
3	3	1
1	3	3

Image B

2	1	2
4	3	5
1	2	3

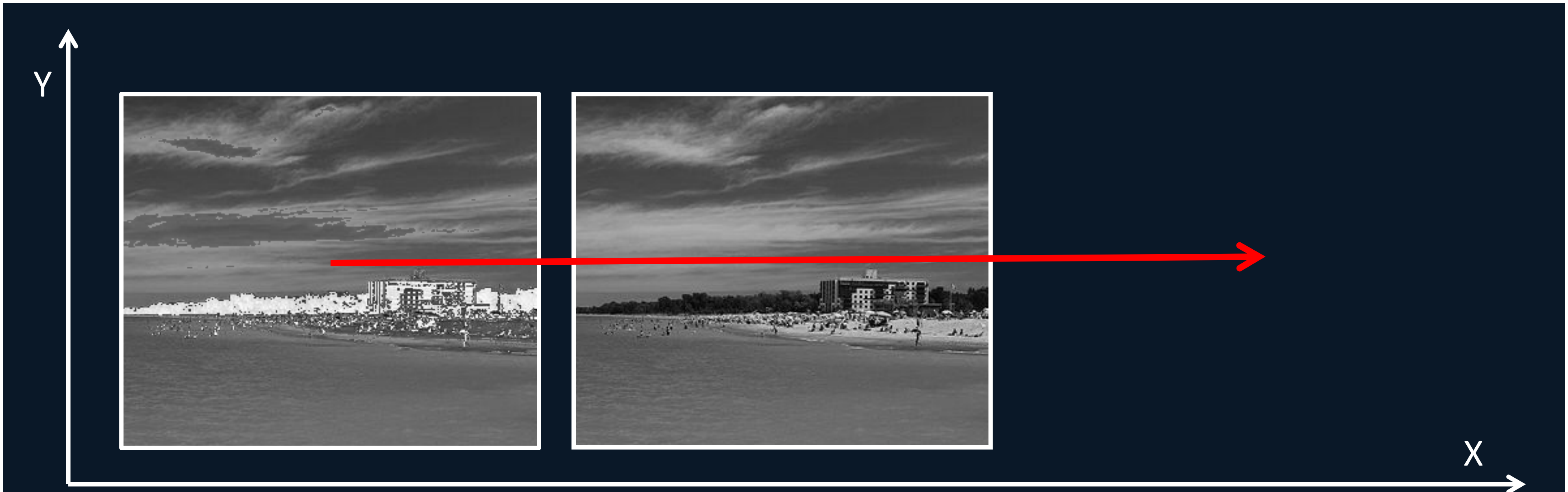
Image B intensities

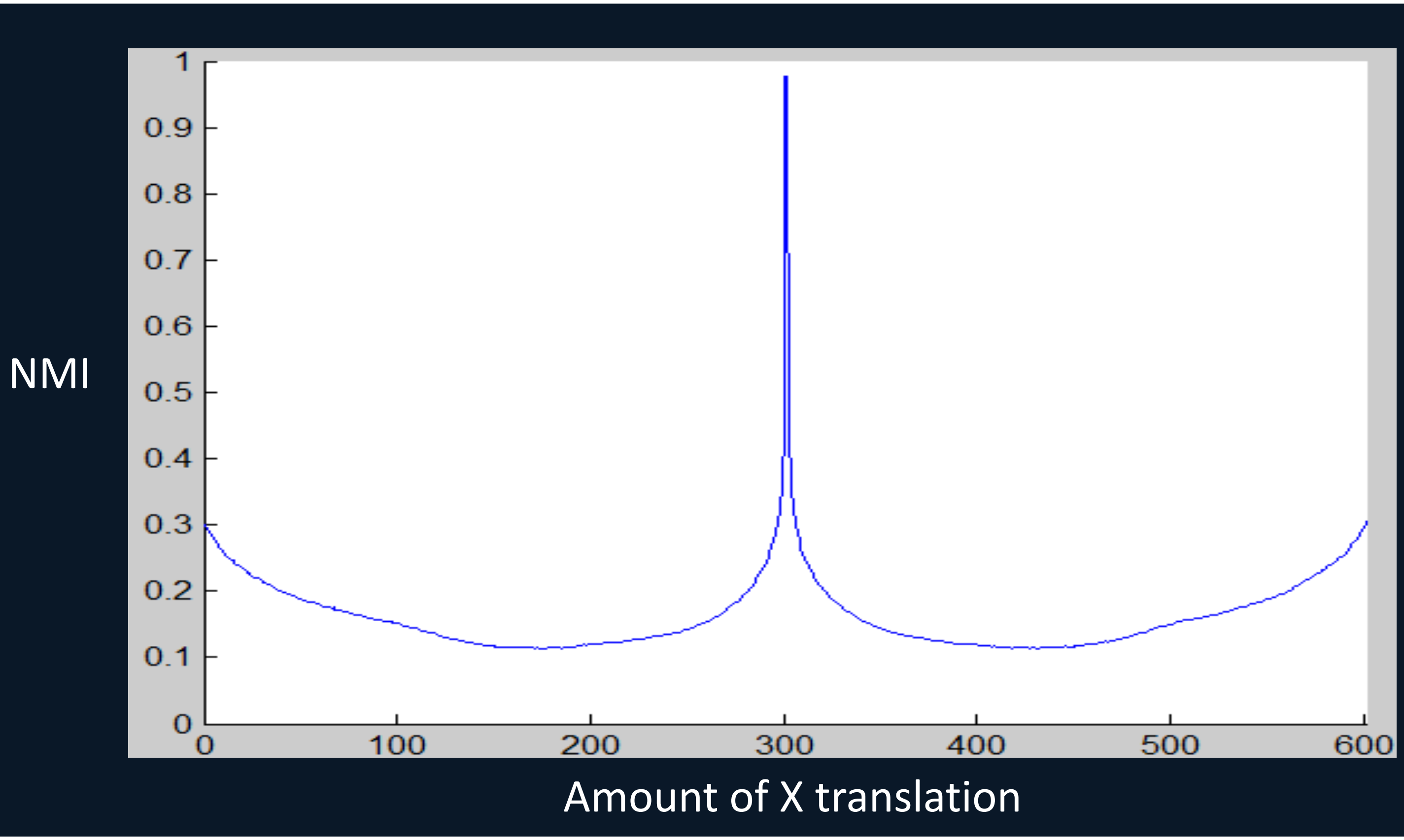
Image A intensities	1	2	3	4	5
1	1	0	0	0	1
2	1	2	0	0	0
3	0	1	2	1	0
4	0	0	0	0	0
5	0	0	0	0	0

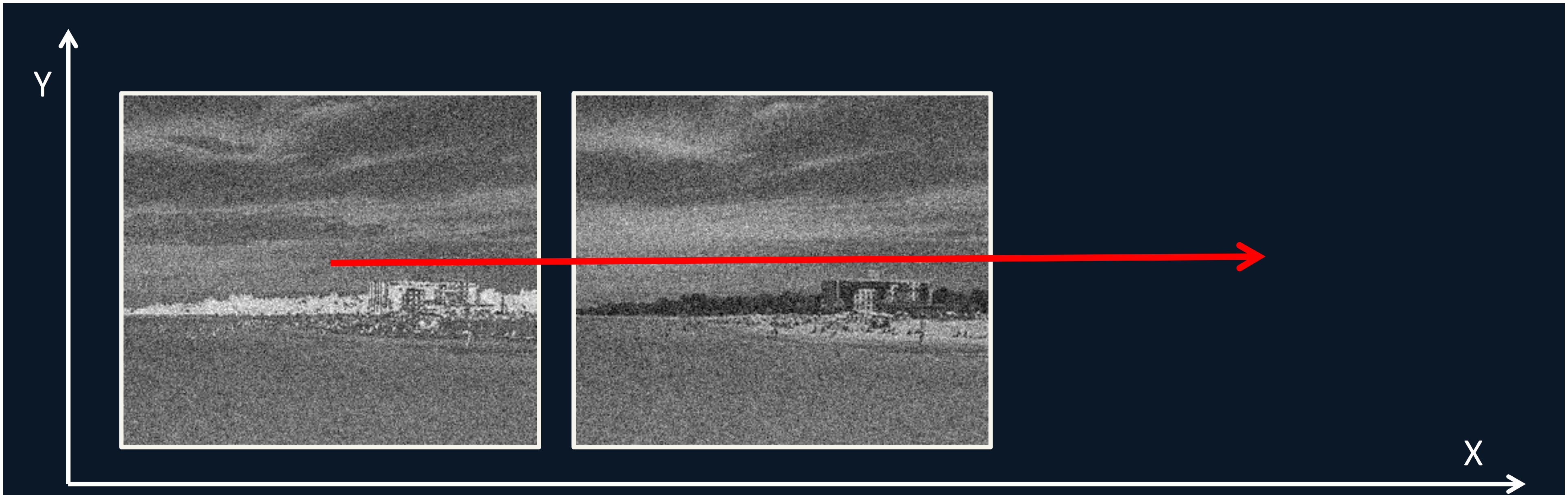
The joint intensity histogram is no longer compact with tall peaks.

It has higher entropy and lower mutual information.

I.e. mutual information may not perform well as a similarity metric!


Y
X


NMI
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

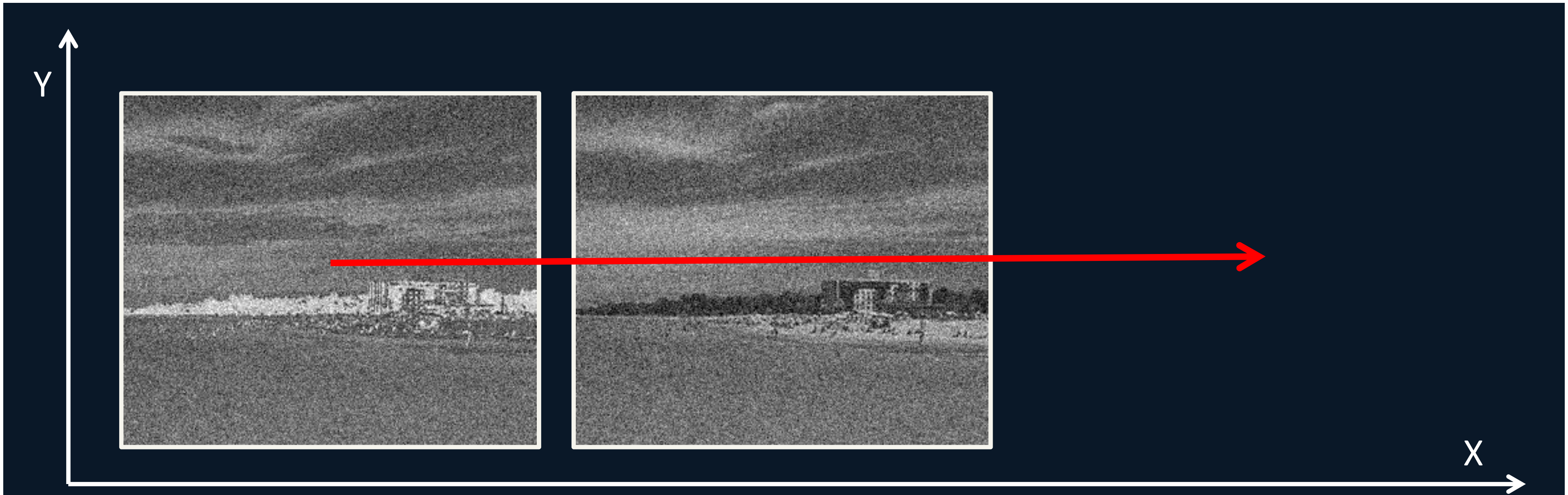


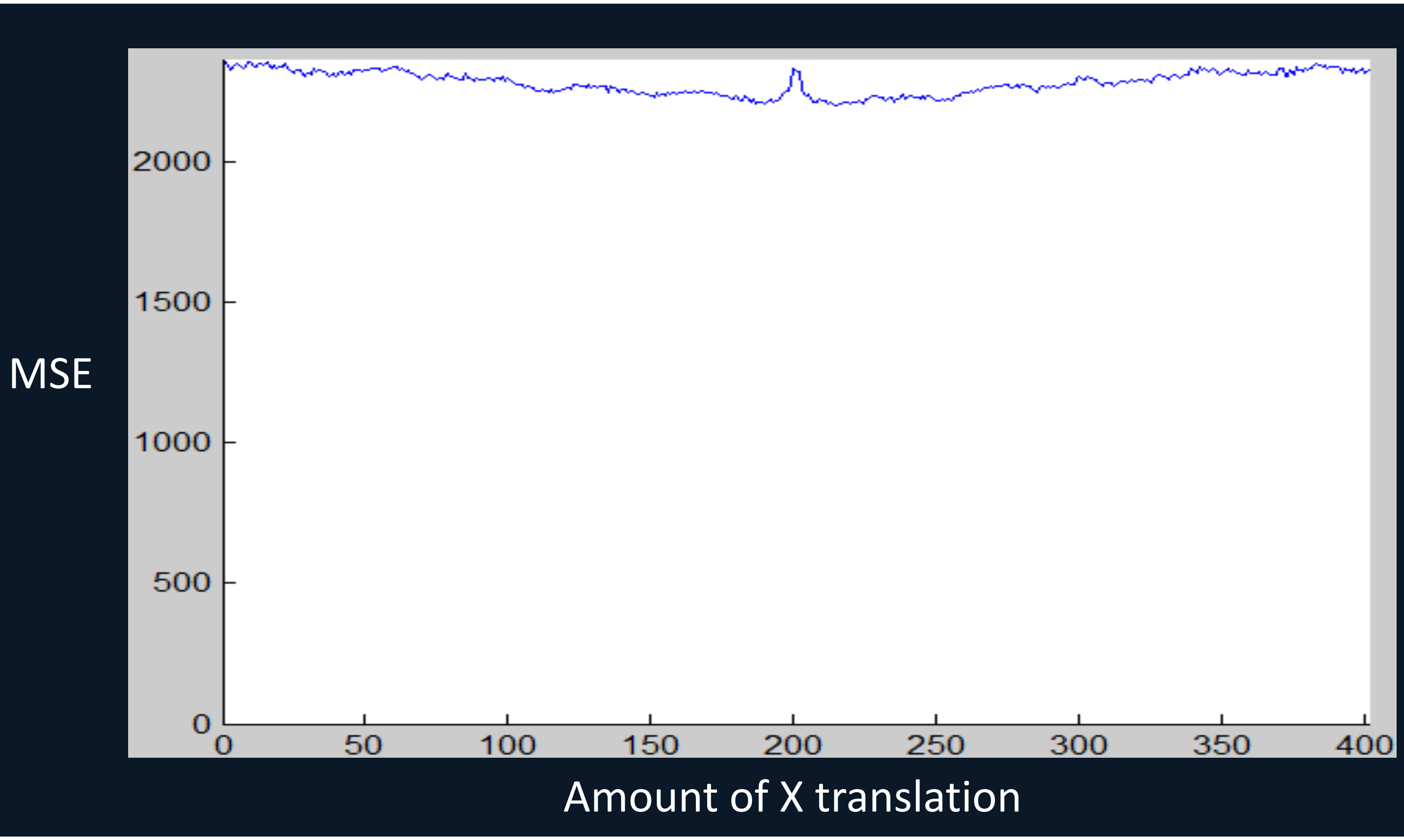
Same images, but with some Gaussian noise added to each pixel.

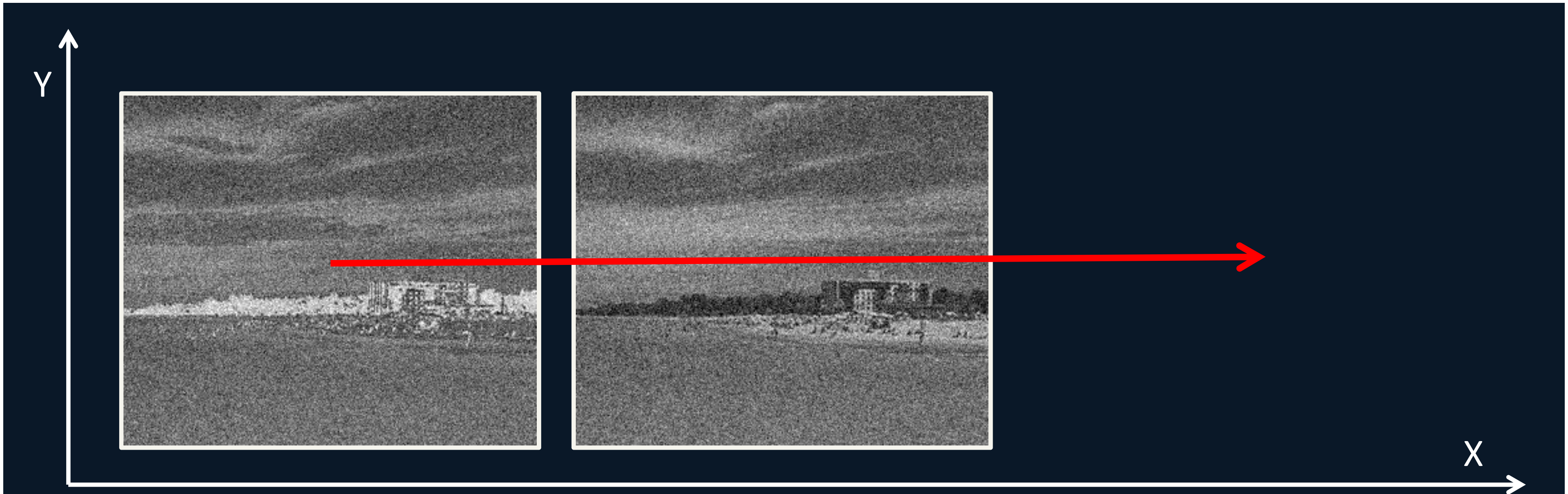
The noise has corrupted the images, but not to the point were we can’t see where the water, beach, buildings, sky, clouds, etc. are located.

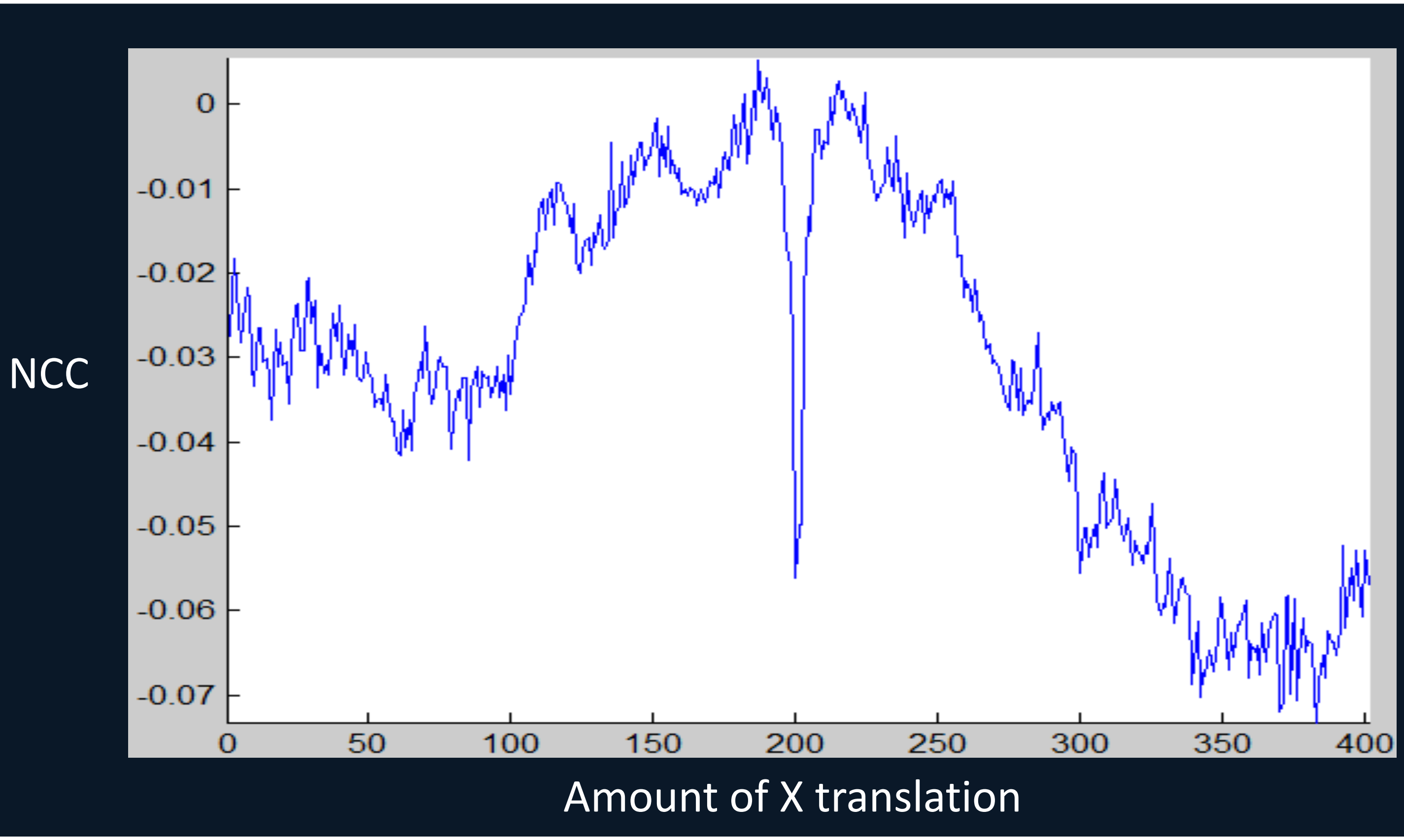
How well will the MSE, NCC, and NMI metrics work?

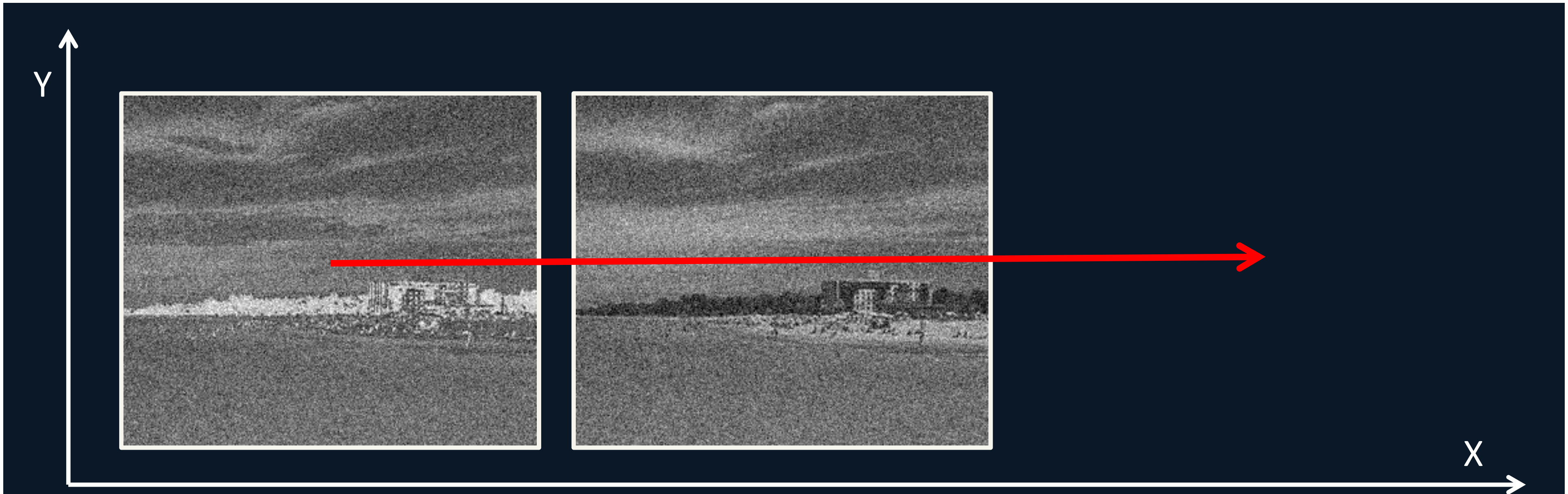
Let’s test each and find out. Remember that an optimizer will *minimize* MSE and *maximize* NCC and NMI.

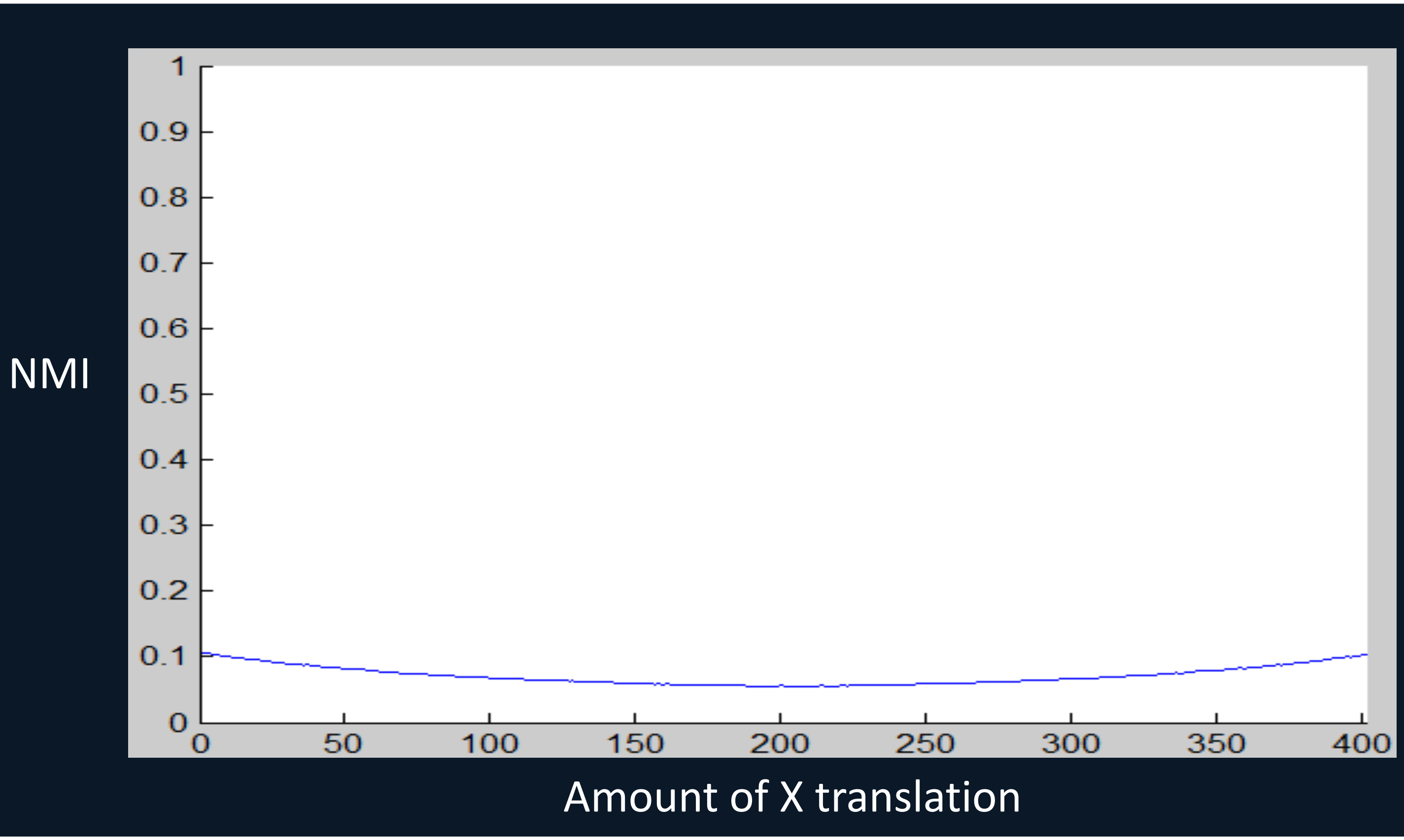

Y
X

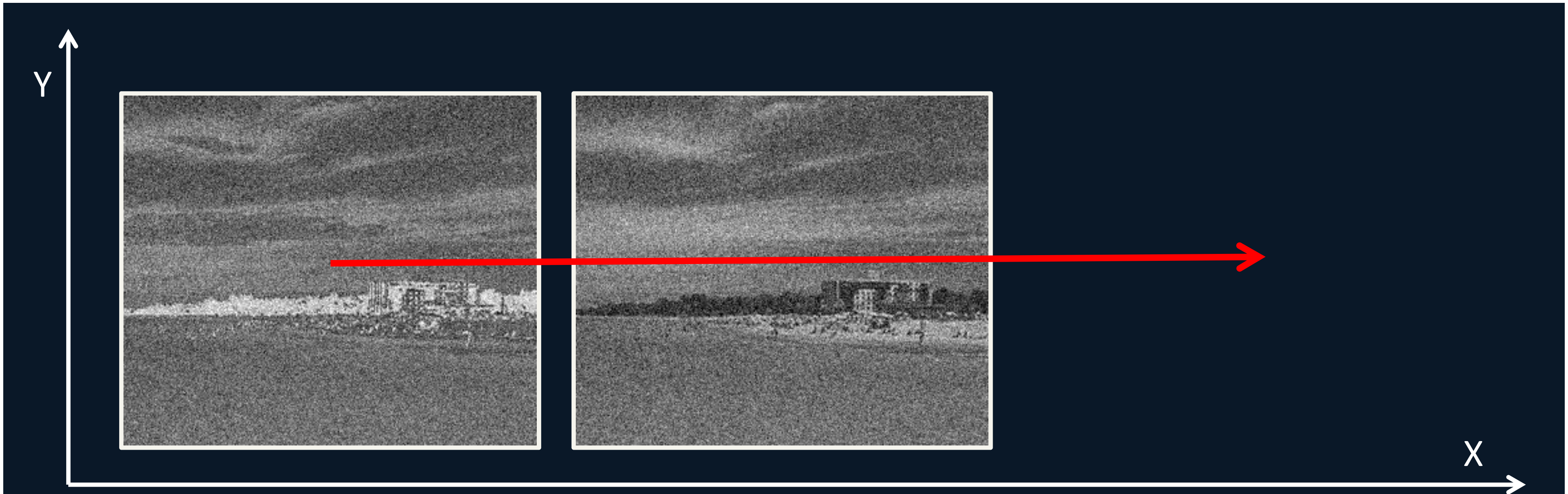

MSE
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400
Amount of X translation


Y
X


NCC
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
0
50
100
150
200
250
300
350
400
Amount of X translation


Y
X


NMI
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350
400
Amount of X translation

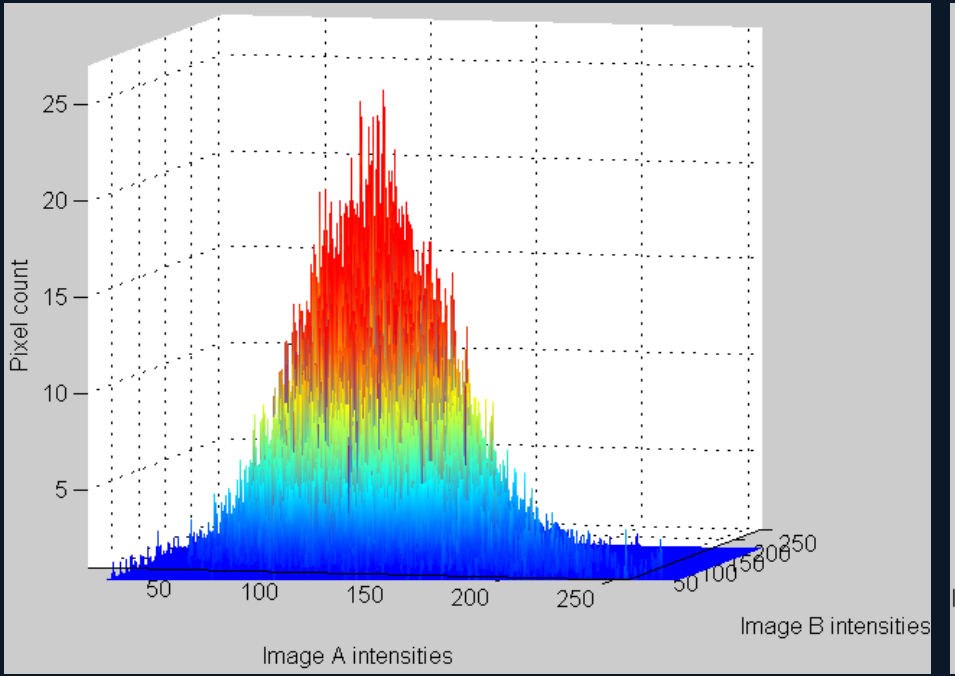


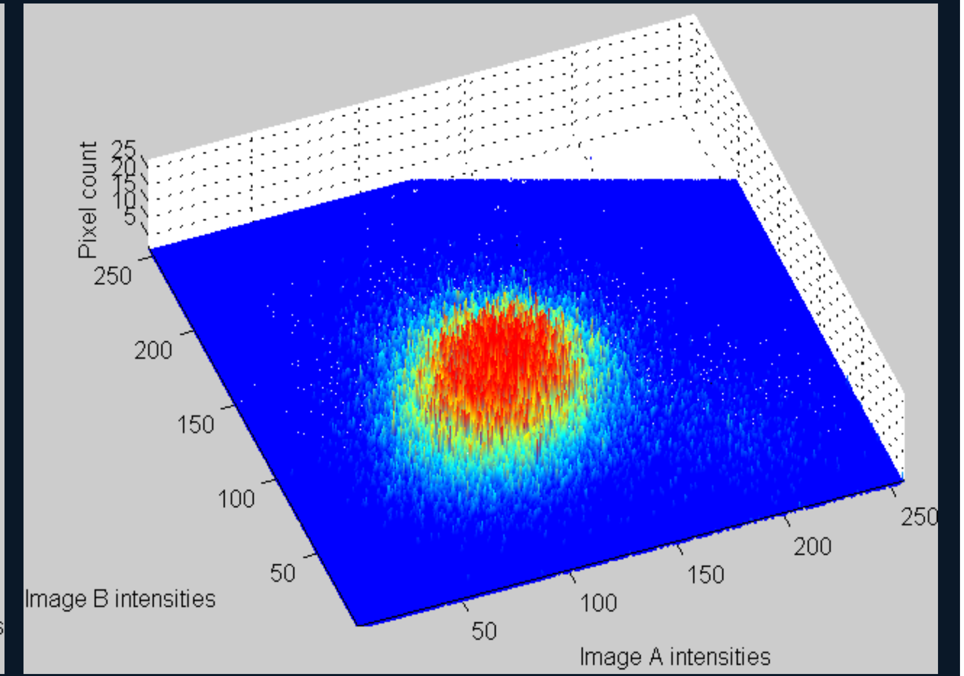
None of the metrics correctly quantifies the alignment of these two images.

MSE gives a maximum at the correct answer when it should give a minimum.

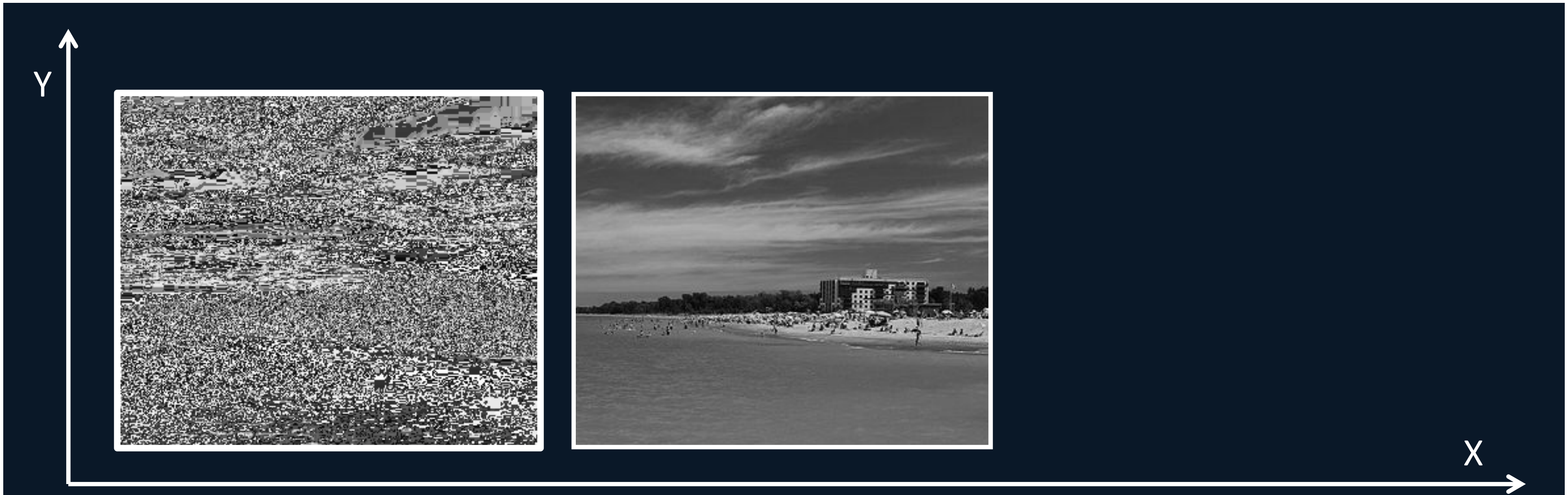
NCC and NMI give minima at the correct answer when they should give maxima.

Why is this happening?



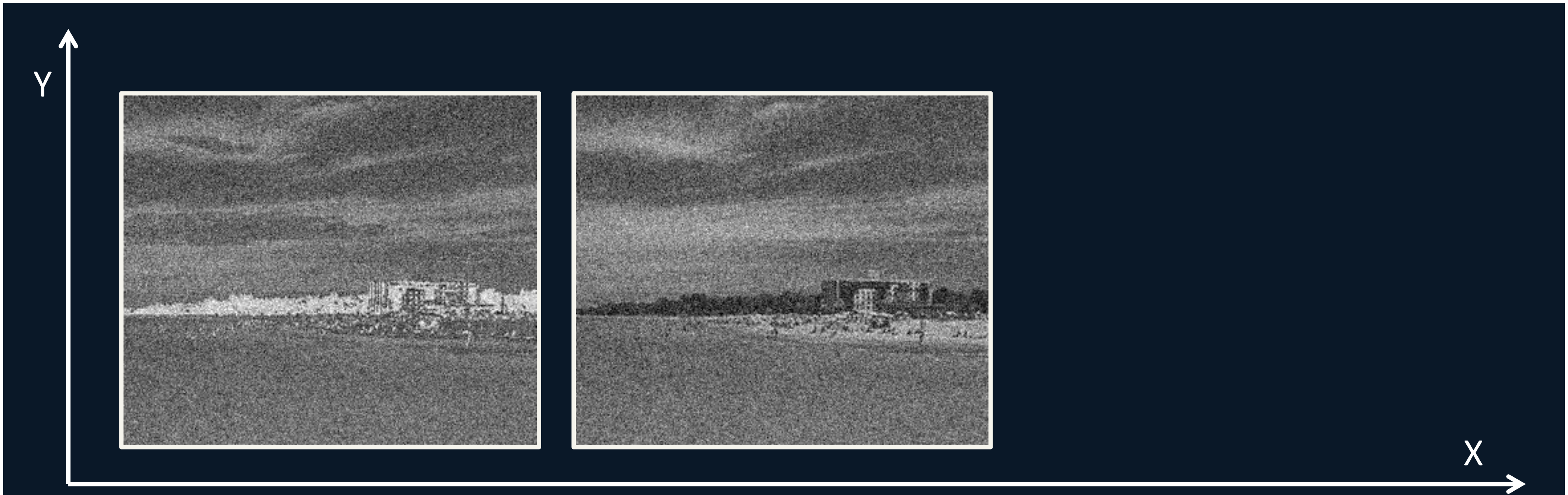


When the images are aligned, the joint intensity histogram is diffuse – the Gaussian noise has introduced entropy.



Take-home message:

Even in extreme cases where the human eye cannot evaluate the alignment of two images, NMI may be suitable if the intensities in the fixed image can be predicted based on the intensities in the correctly aligned moving image.



Take-home message:

There are cases where the human can evaluate the alignment of two images but the images are corrupted by randomness that reduces the utility of image similarity metrics (even NMI) to evaluate their alignment.

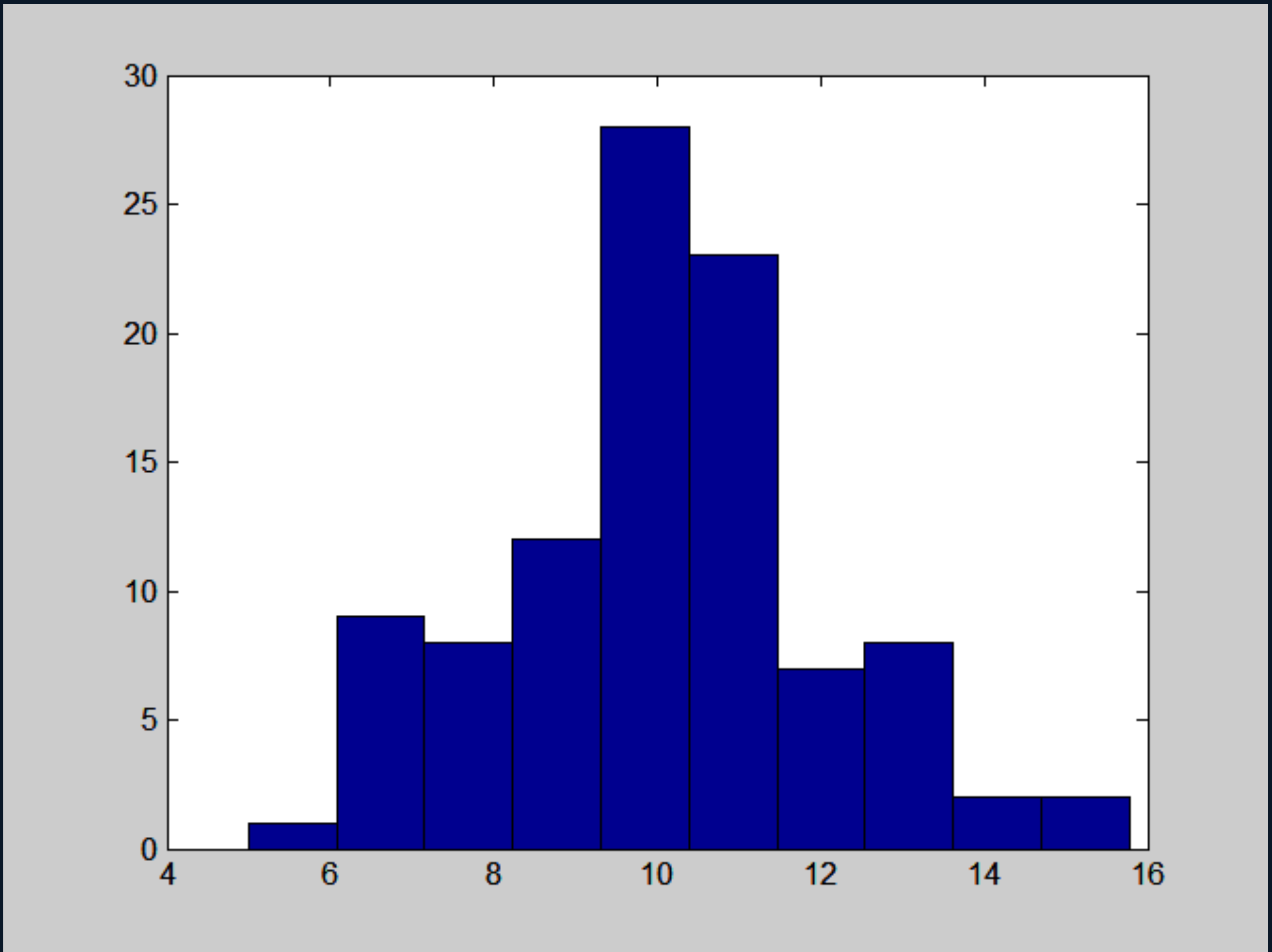
Does the number of histogram bins matter for MI?

To explore this, we’ll use a Gaussian distribution of 100 numbers with mean of 10 and standard deviation of 2.

We’ll plot histograms of these numbers with different numbers of bins.

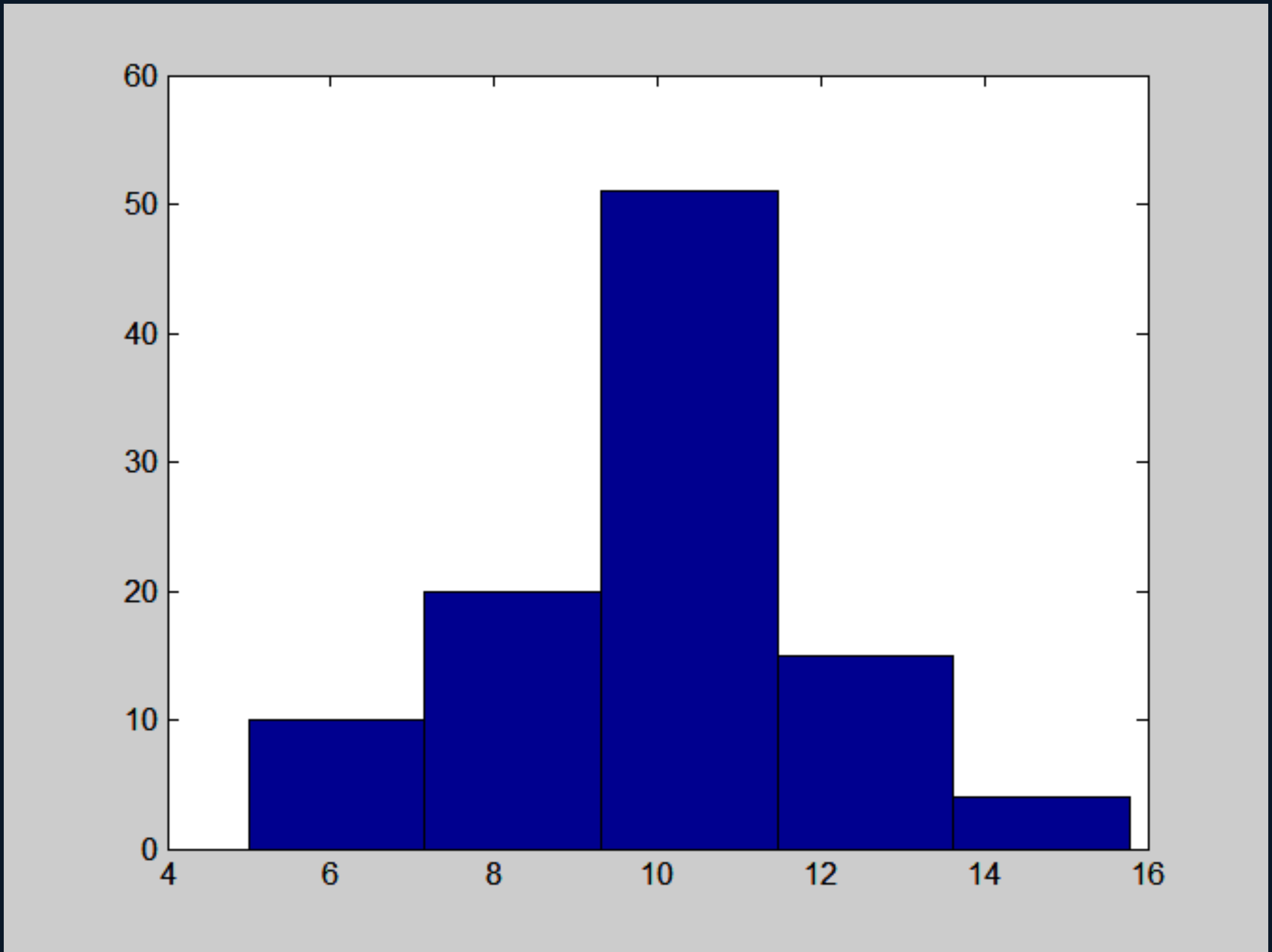
Observe what happens to the histogram peaks as the number of bins varies, and consider the impact on measurement of information entropy in the mutual information calculation.

Does the number of histogram bins matter?



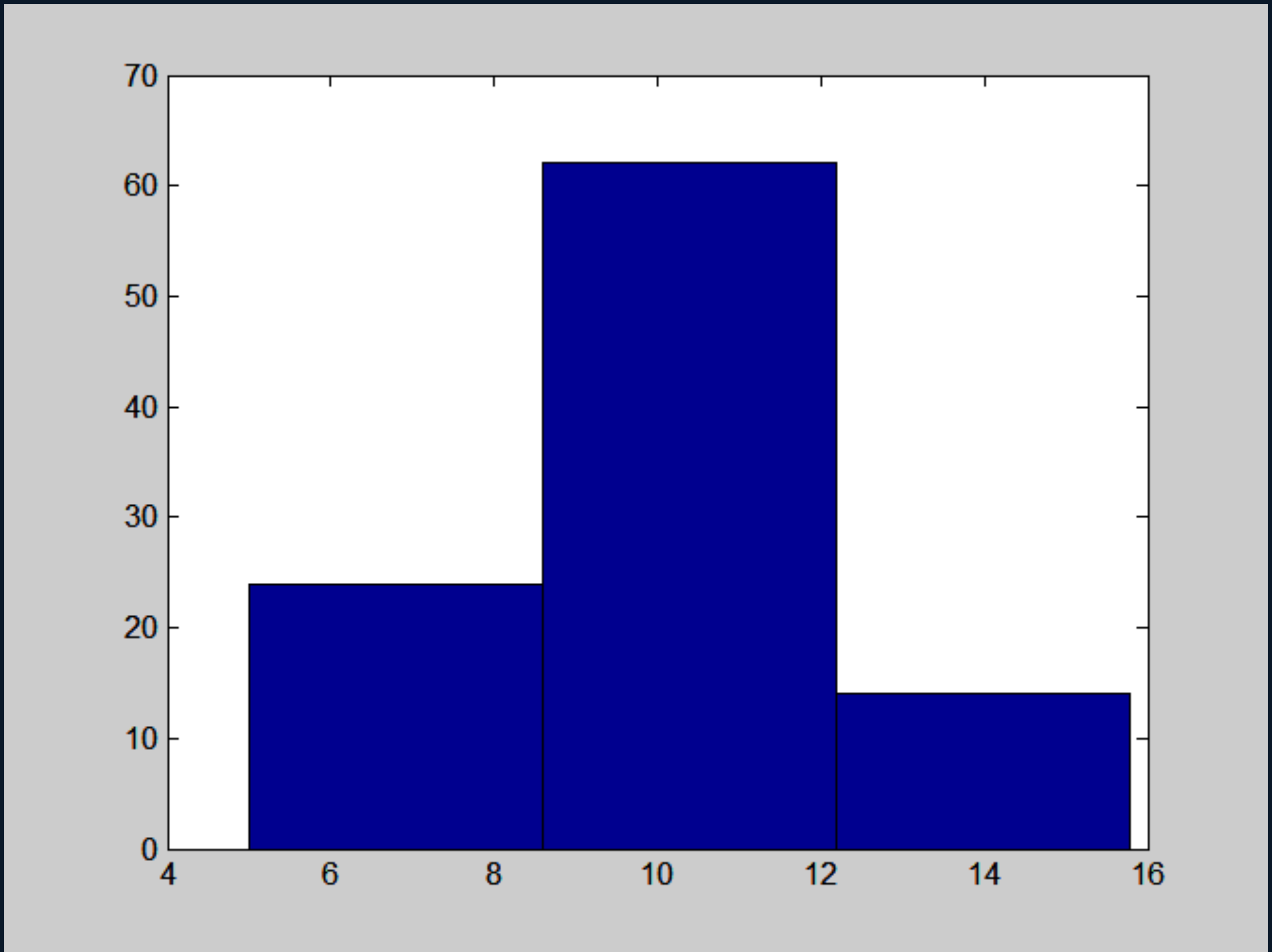
10-bin histogram of 100 values

Does the number of histogram bins matter?



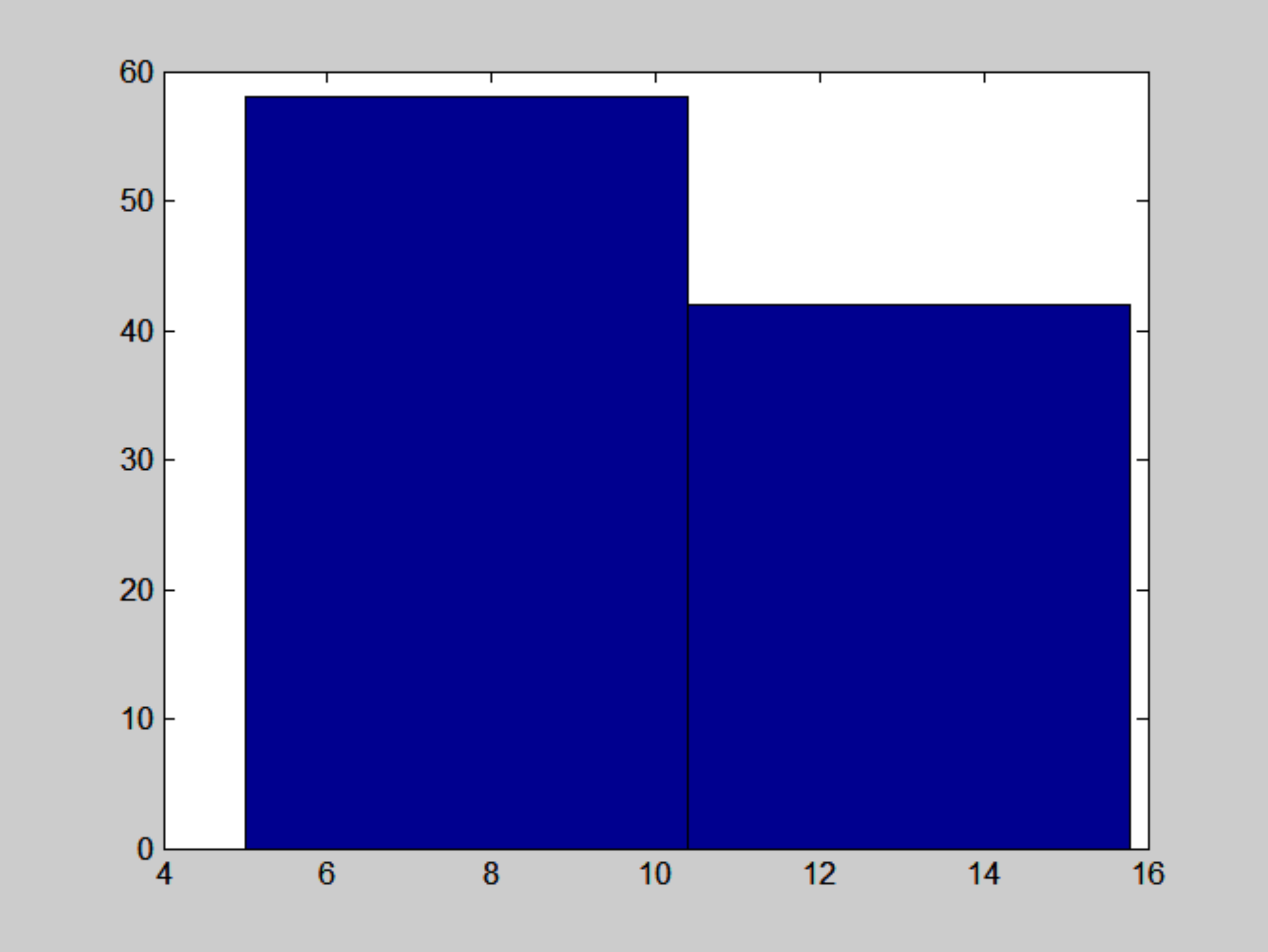
5-bin histogram of same 100 values

Does the number of histogram bins matter?



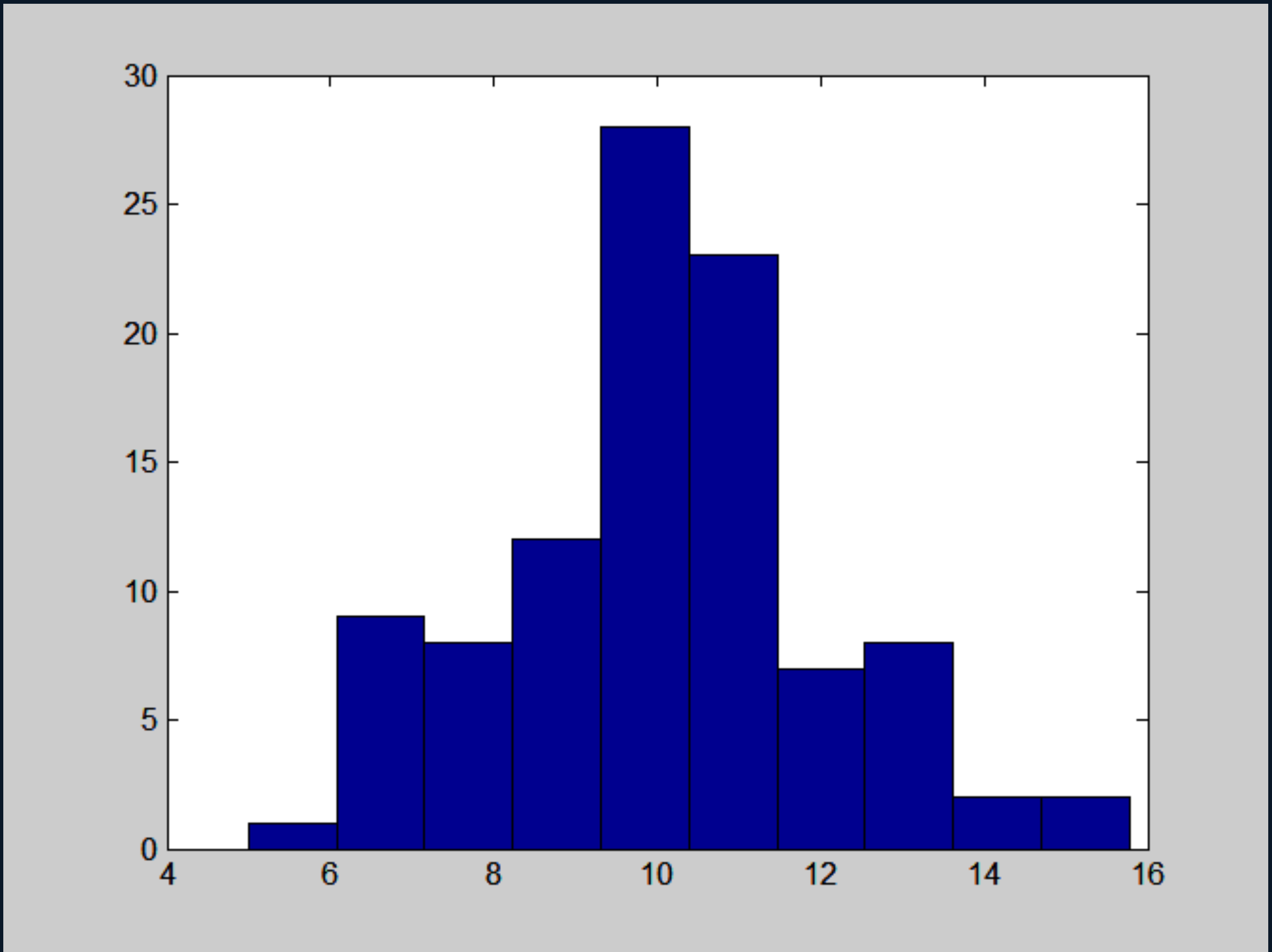
3-bin histogram of same 100 values

Does the number of histogram bins matter?



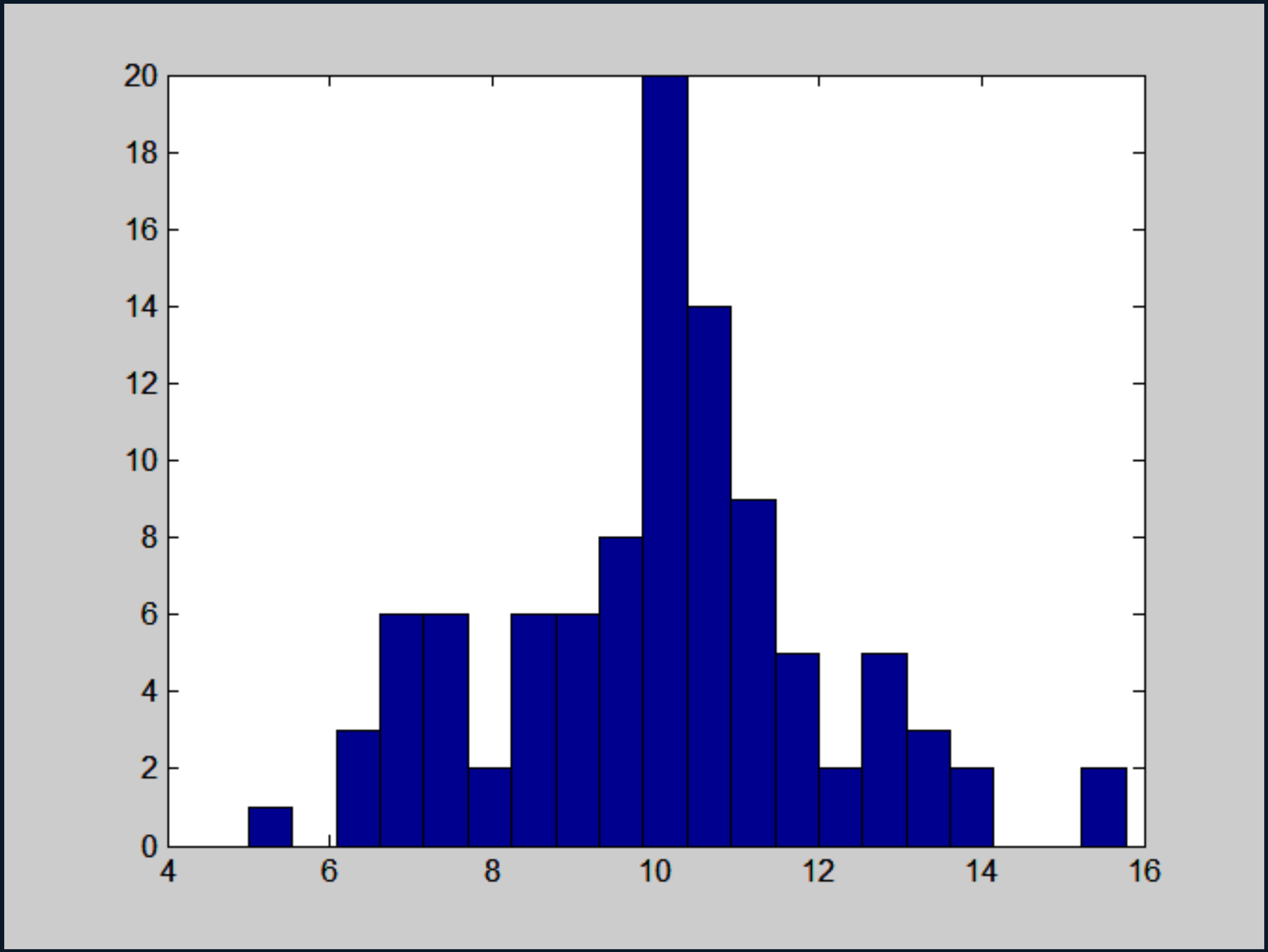
2-bin histogram of same 100 values

Does the number of histogram bins matter?



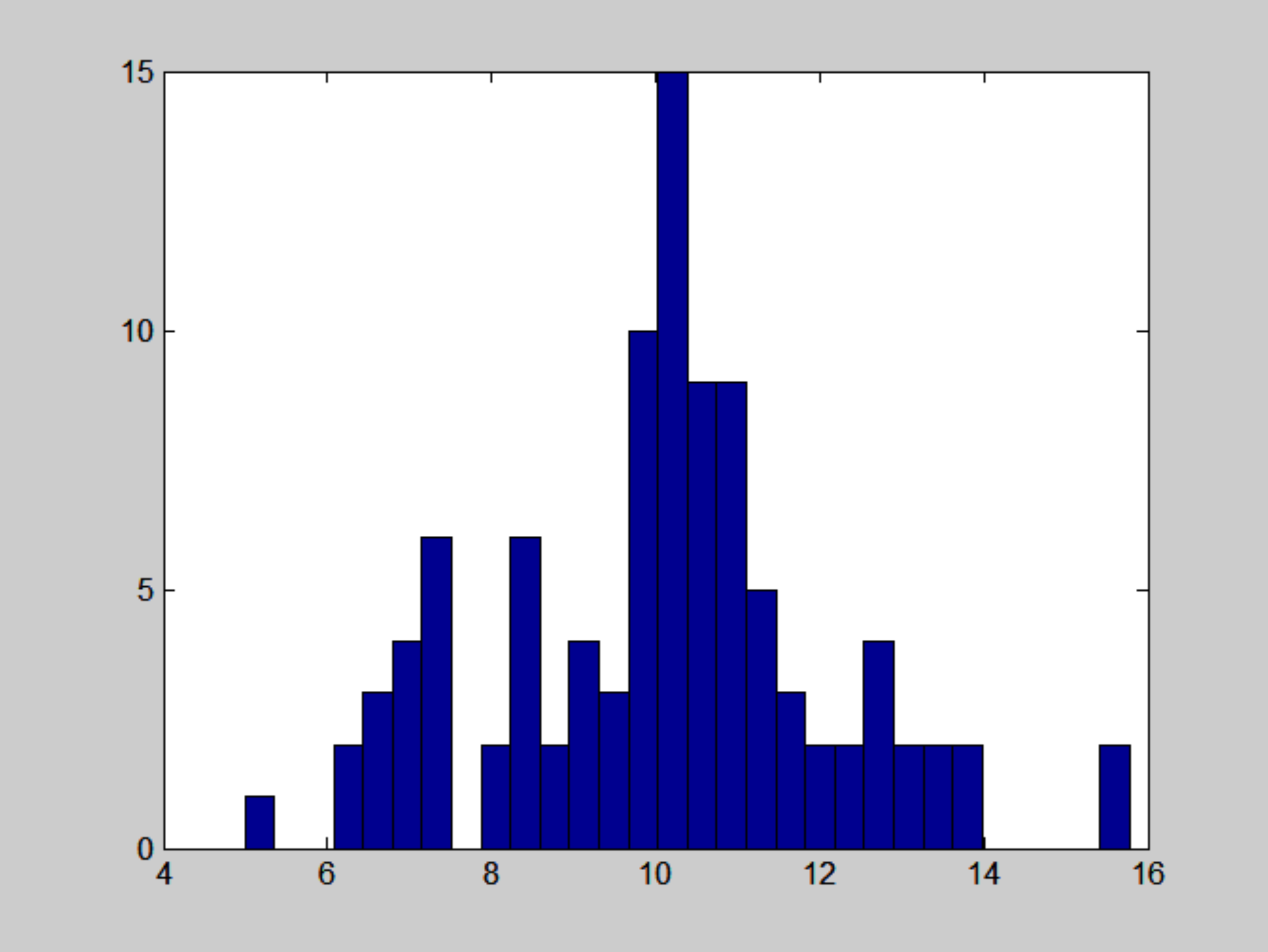
10-bin histogram of 100 values

Does the number of histogram bins matter?



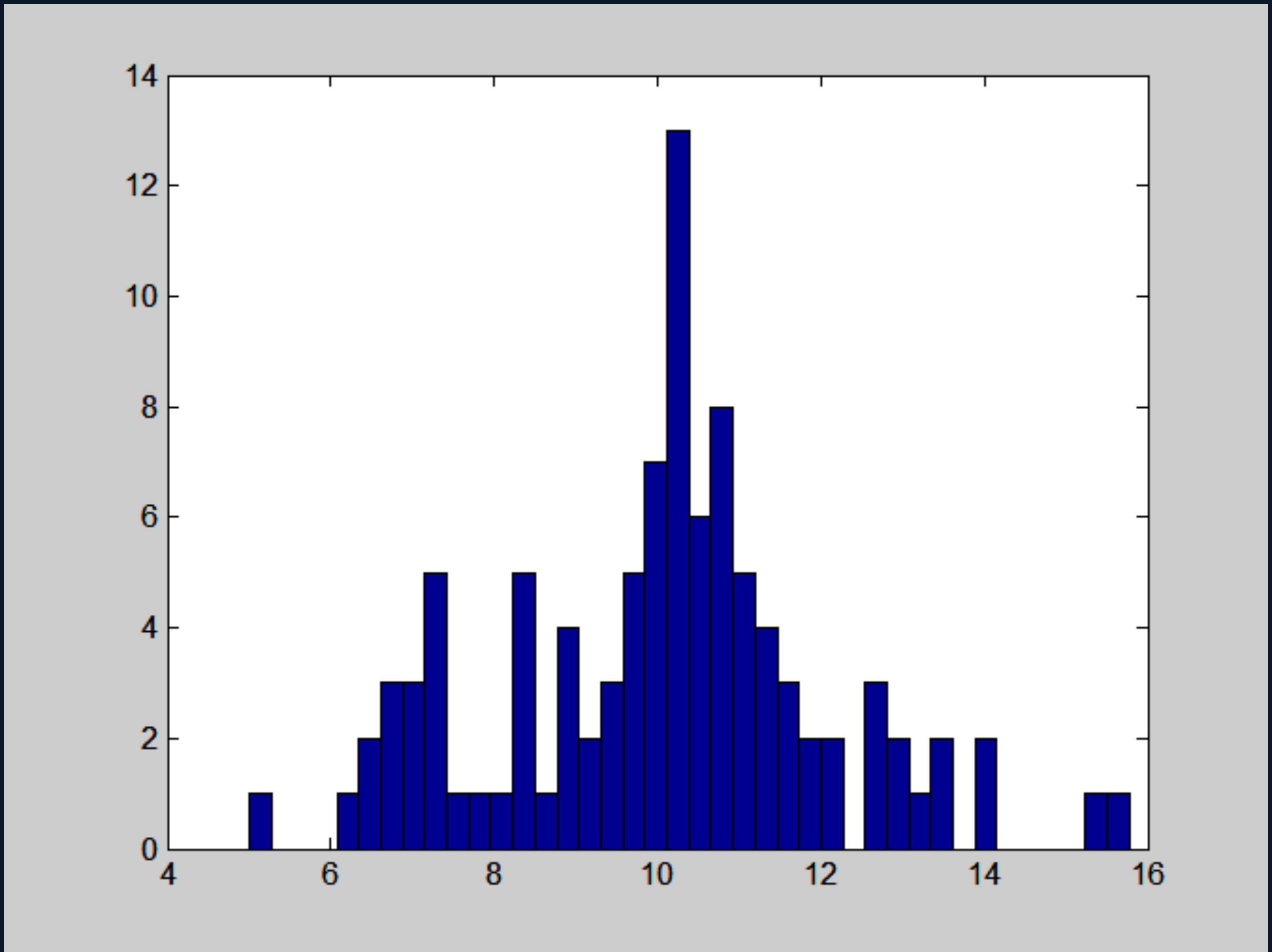
20-bin histogram of same 100 values

Does the number of histogram bins matter?



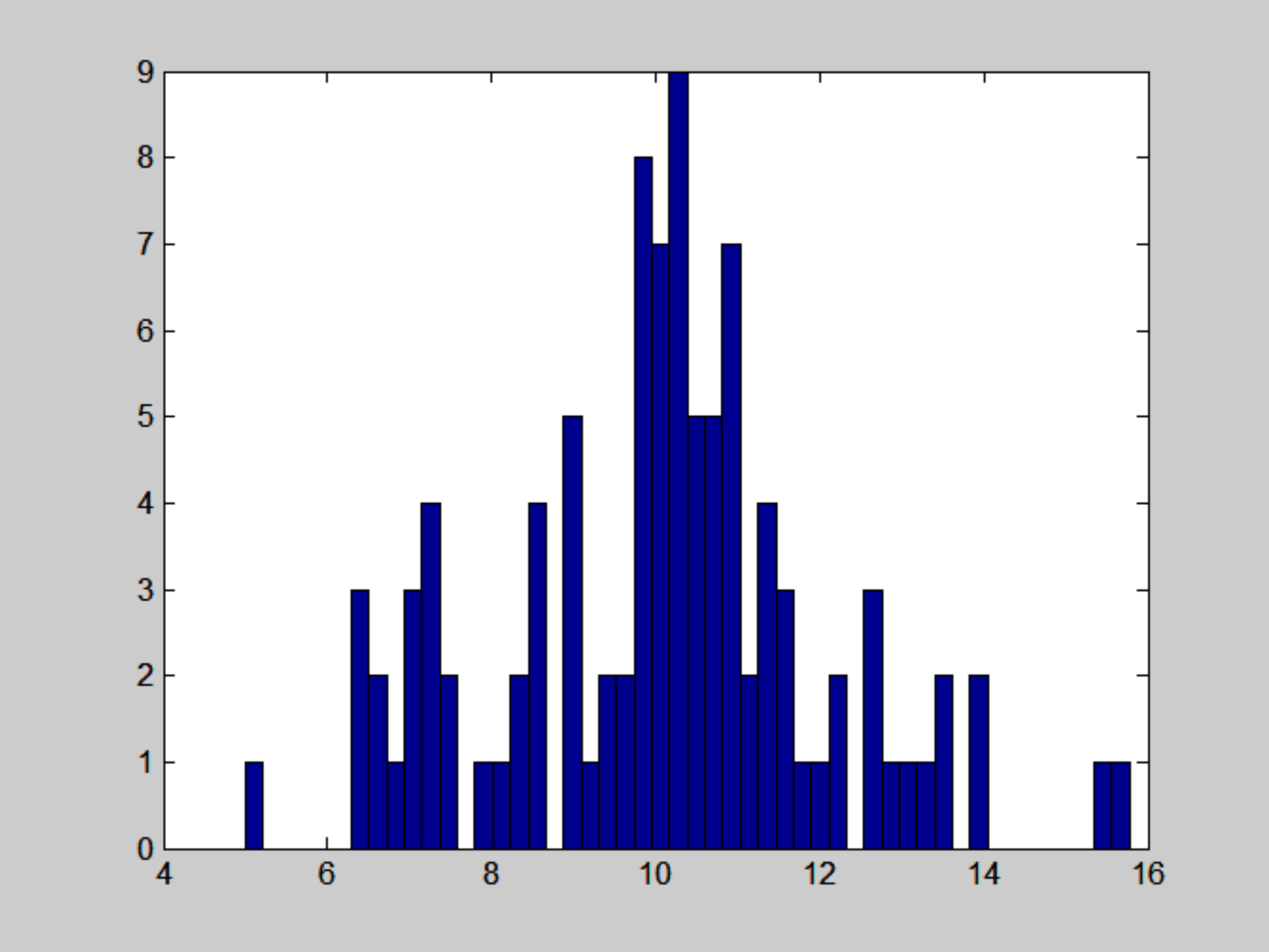
30-bin histogram of same 100 values

Does the number of histogram bins matter?



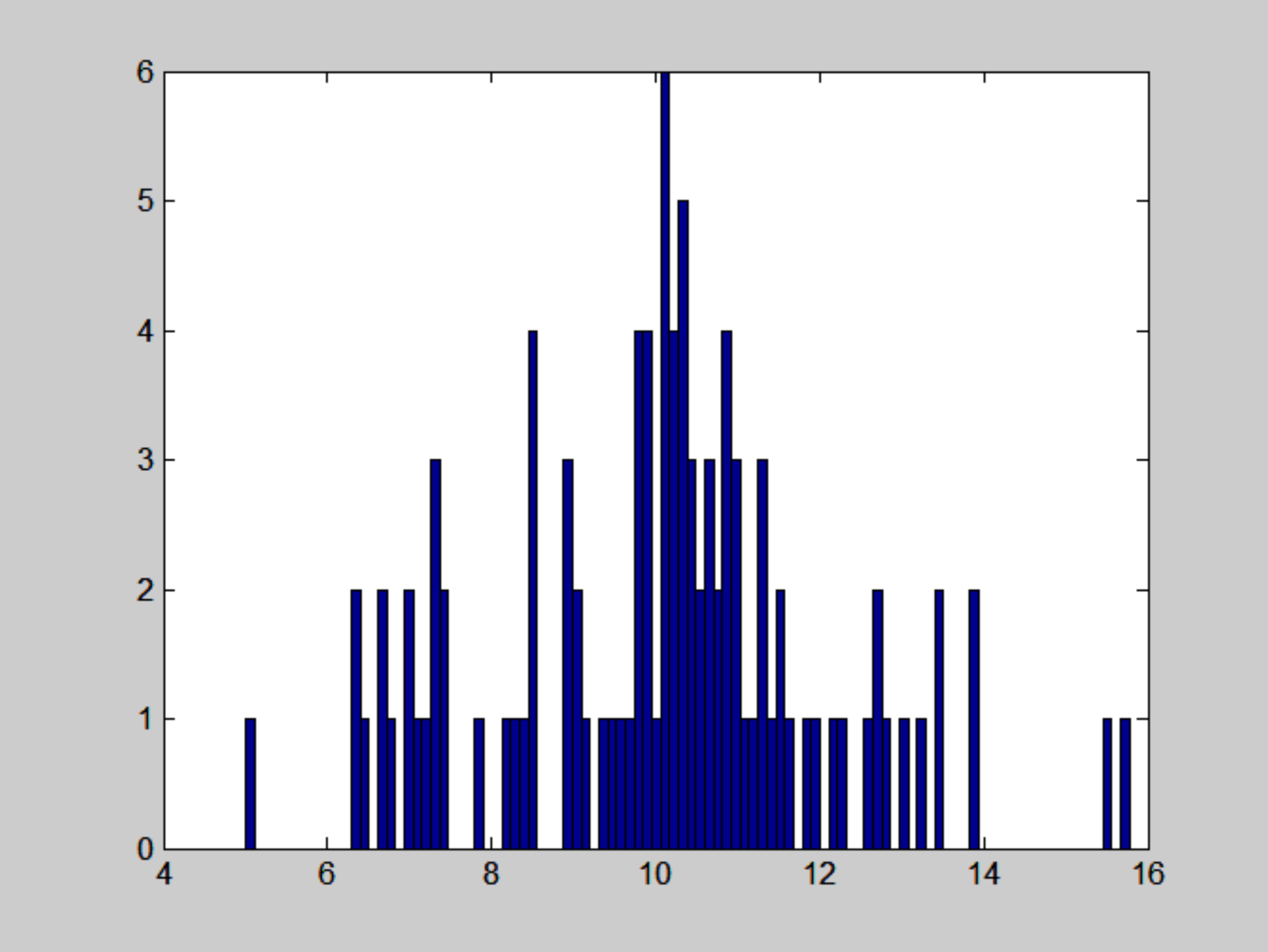
40-bin histogram of same 100 values

Does the number of histogram bins matter?



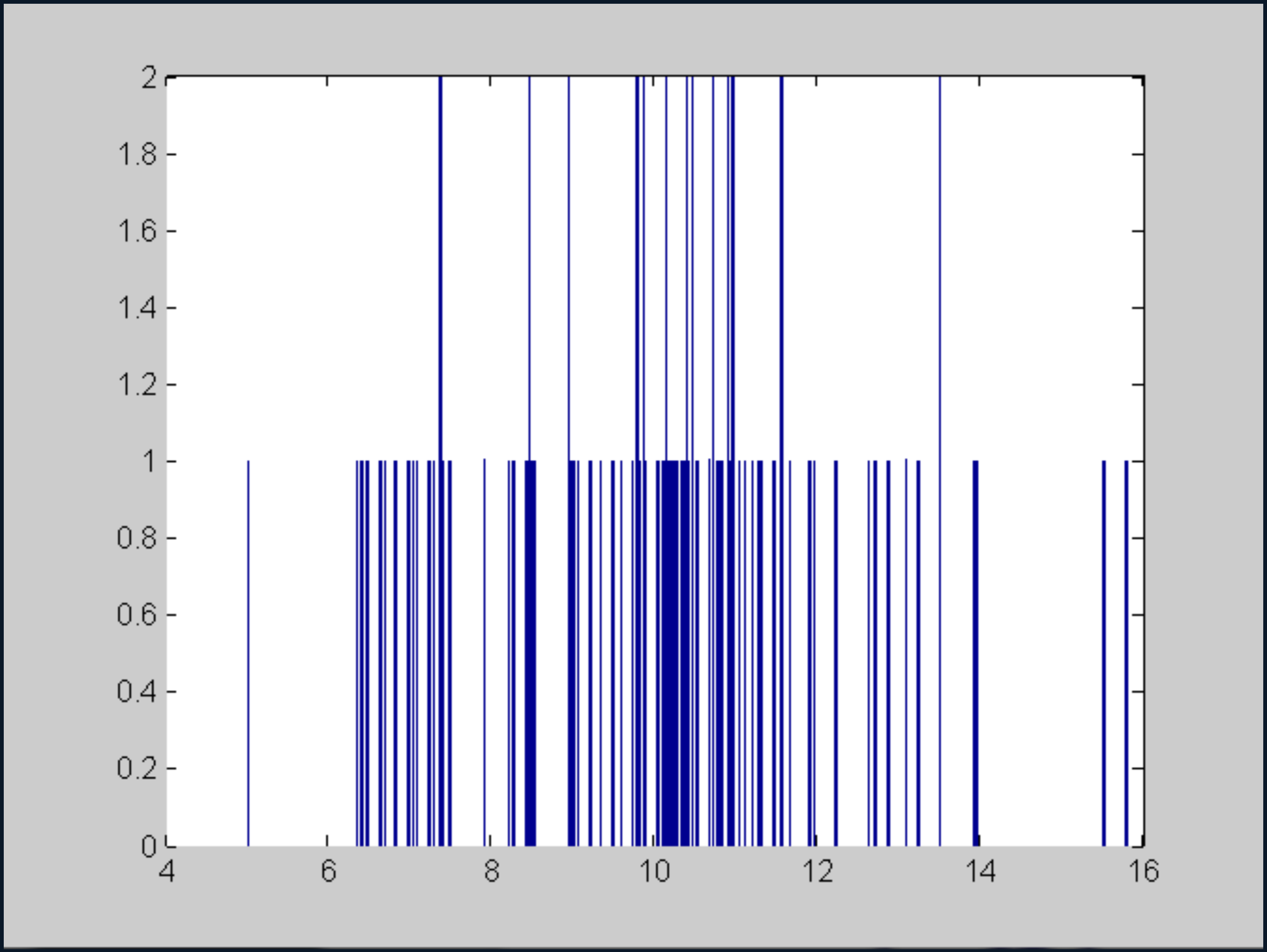
50-bin histogram of same 100 values

Does the number of histogram bins matter?



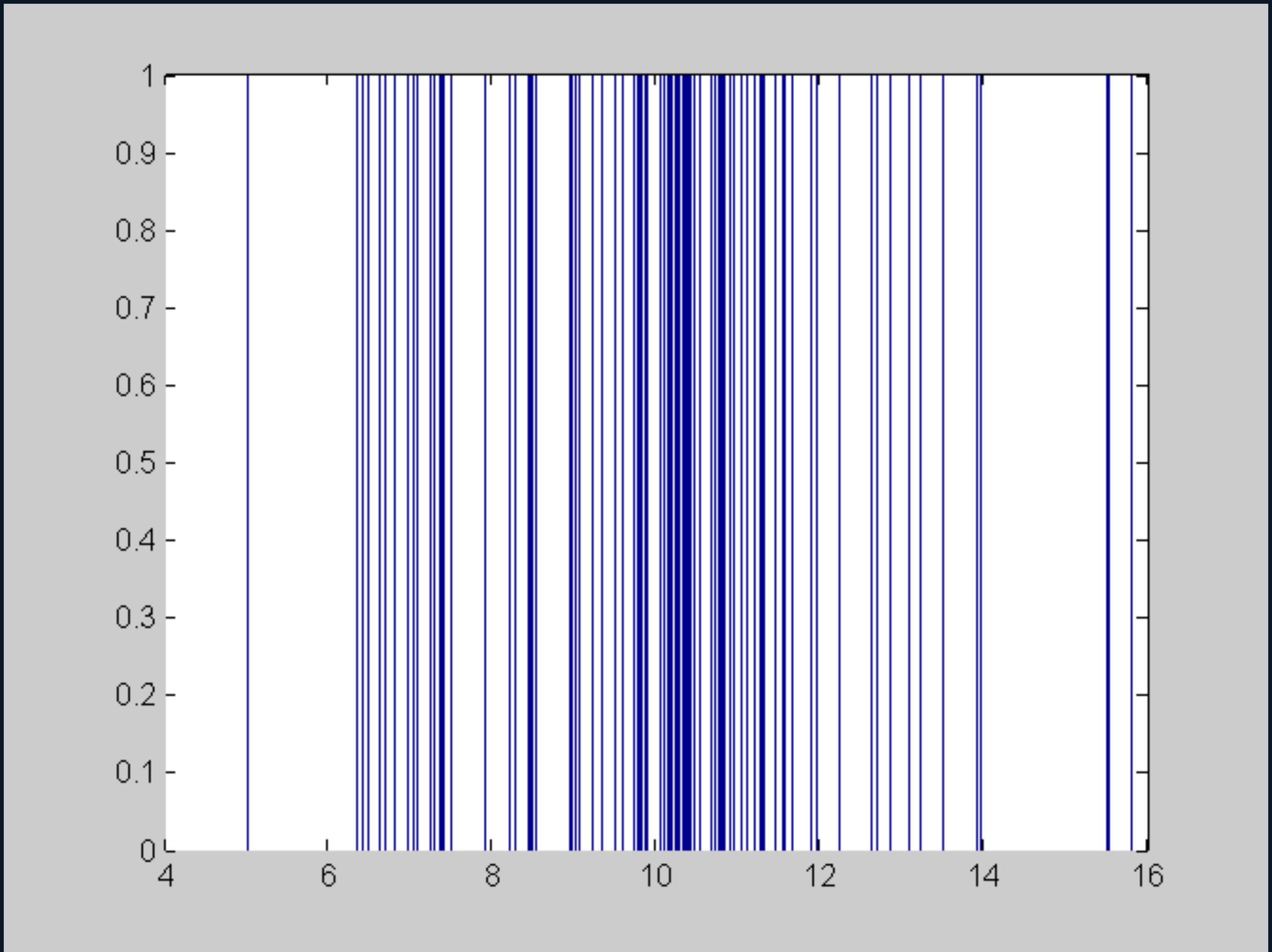
100-bin histogram of same 100 values

Does the number of histogram bins matter?



1000-bin histogram of same 100 values

Does the number of histogram bins matter?



10000-bin histogram of same 100 values

Why does the number of histogram bins matter for MI?

Clearly the nature of a discrete histogram (in terms of being flat or peaked) is affected by the choice of the number of bins.

Since NMI depends on a measure of these histograms, the calculated NMI value is sensitive to the choice of the number of bins.

In practice, this is a parameter that is tuned/optimized empirically for each registration problem.

Algorithm specification

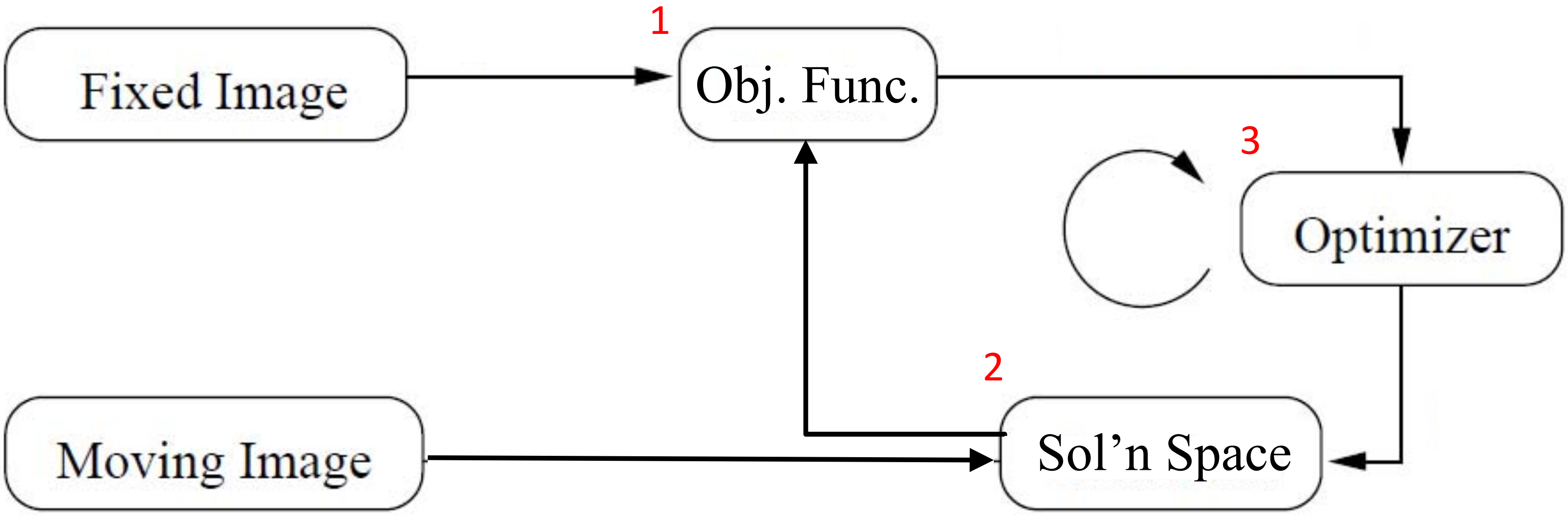
Recall that the description of an algorithm (e.g. for image registration) can have three components:

1. The *objective function*.

2. The *solution space*.

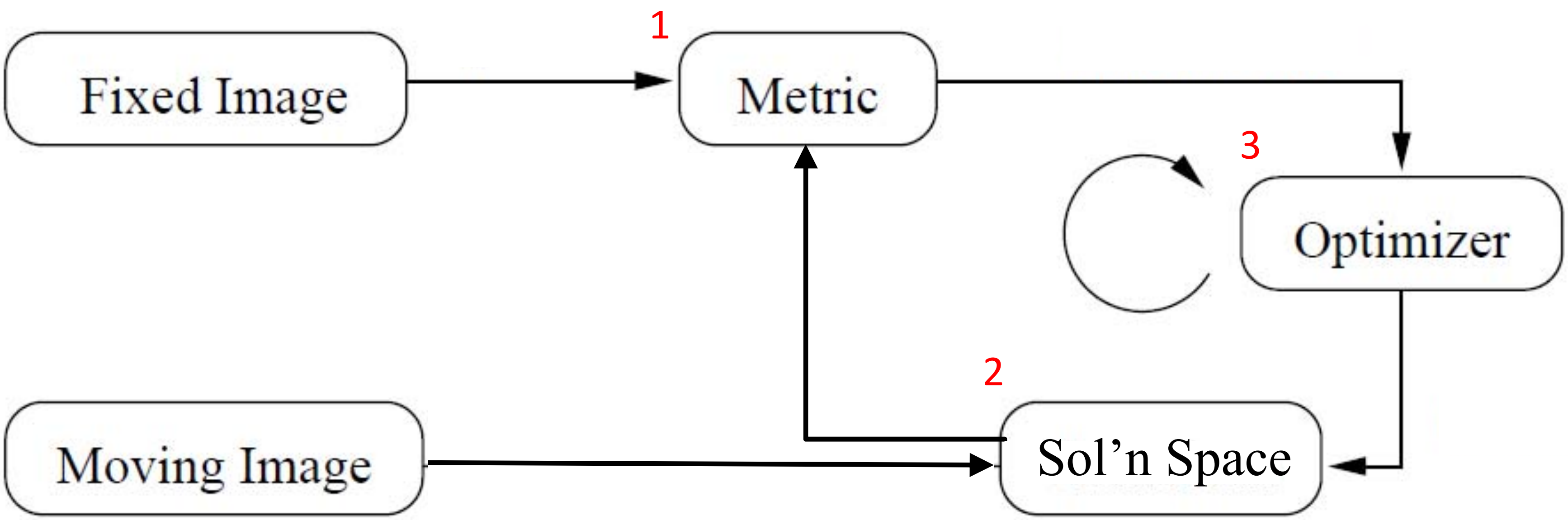
3. The *optimizer*.

An image registration framework



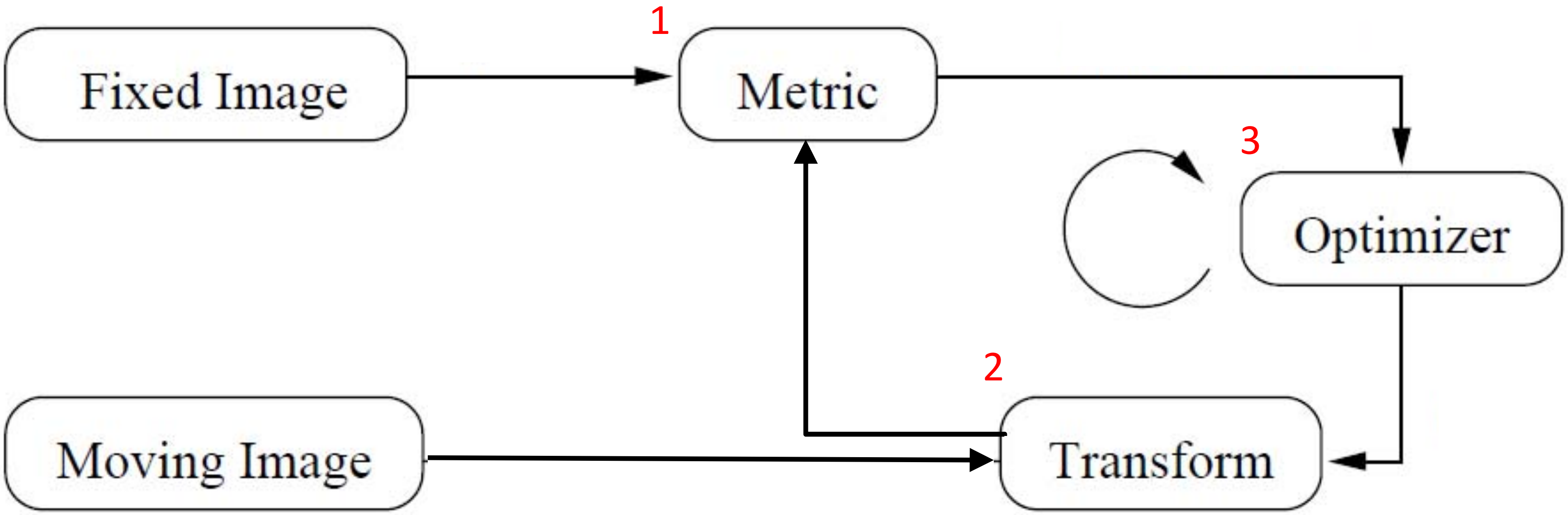
For image registration, we could arrange these three components in a loop, like this.

An image registration framework



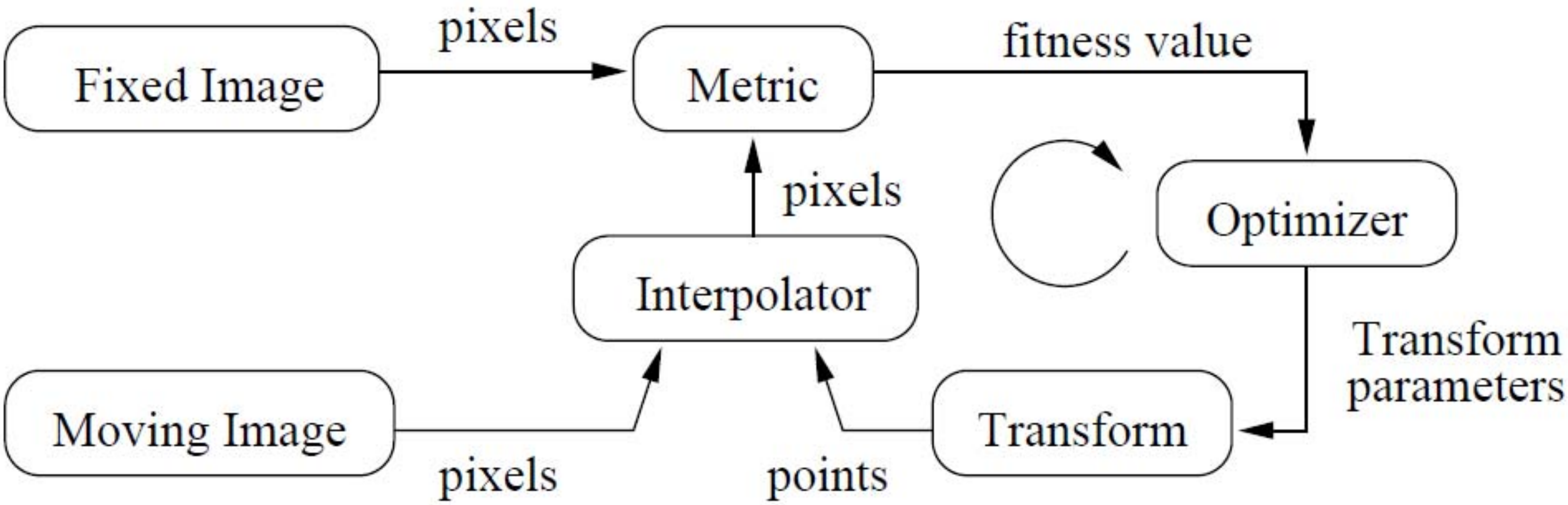
In registration, the objective function can be the image similarity metric....

An image registration framework



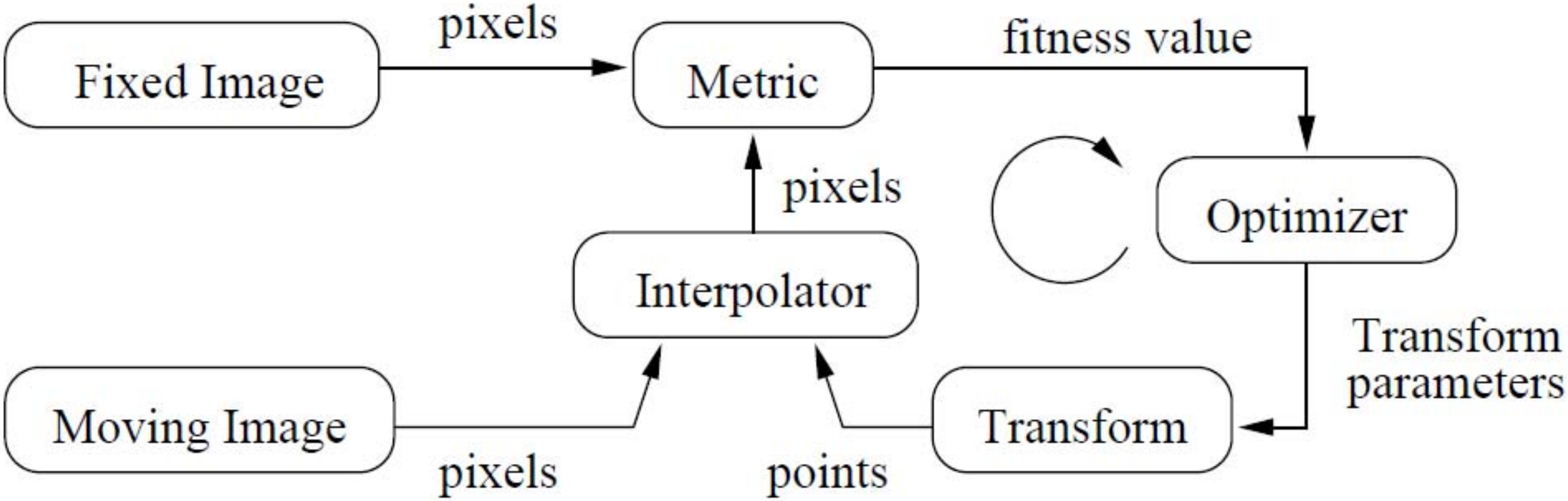
…and the solution space can be the space of possible spatial transformations of the moving image.

An image registration framework



In a practical implementation, each transformation of the moving image requires a resampling step, and a specific type of interpolator needs to be used (e.g. nearest neighbour, linear, cubic, etc.)

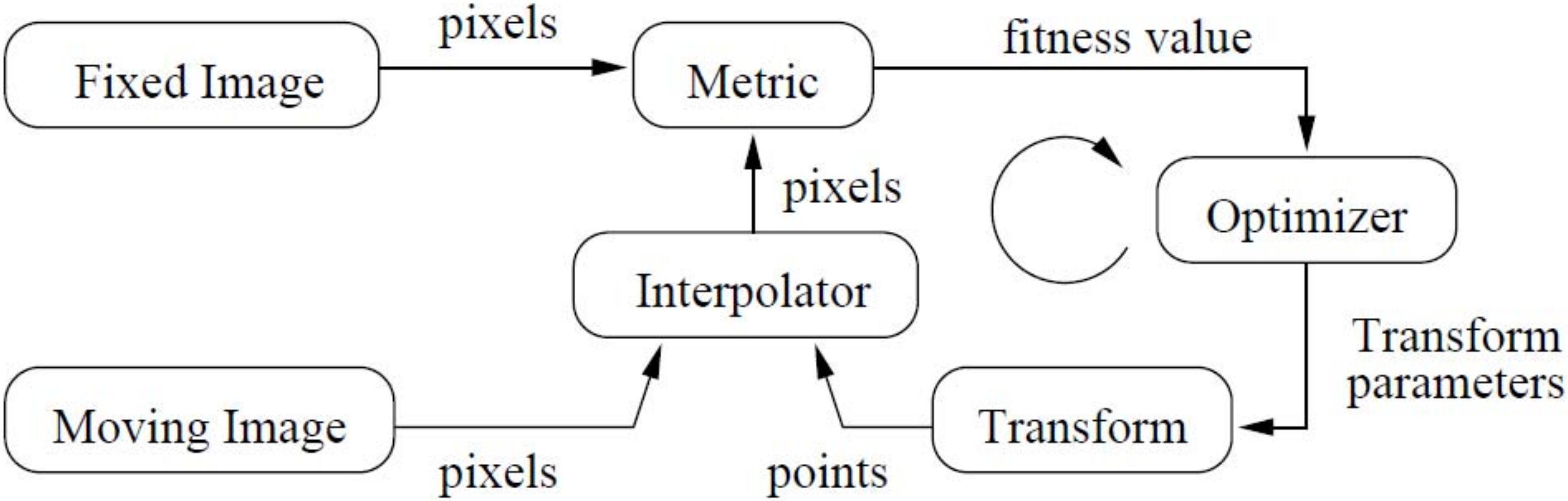
ITK registration framework



Let’s look at the ITK implementation of each of these elements in conjunction with example:

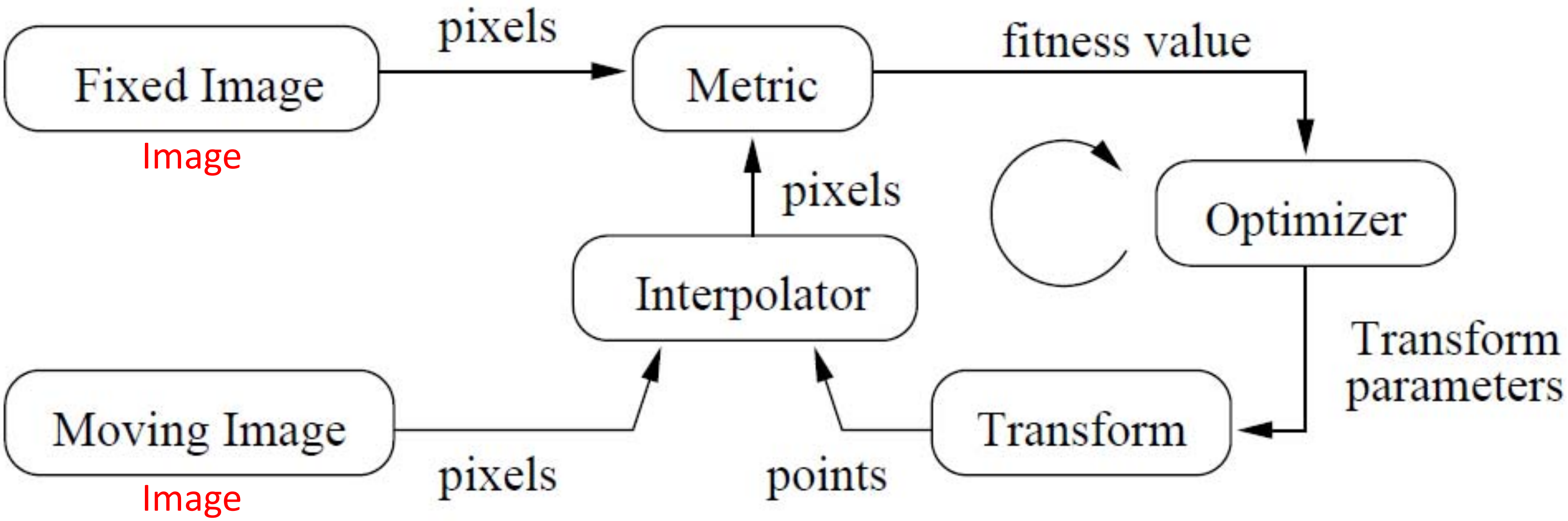
6_Translation_Registration

ITK registration framework



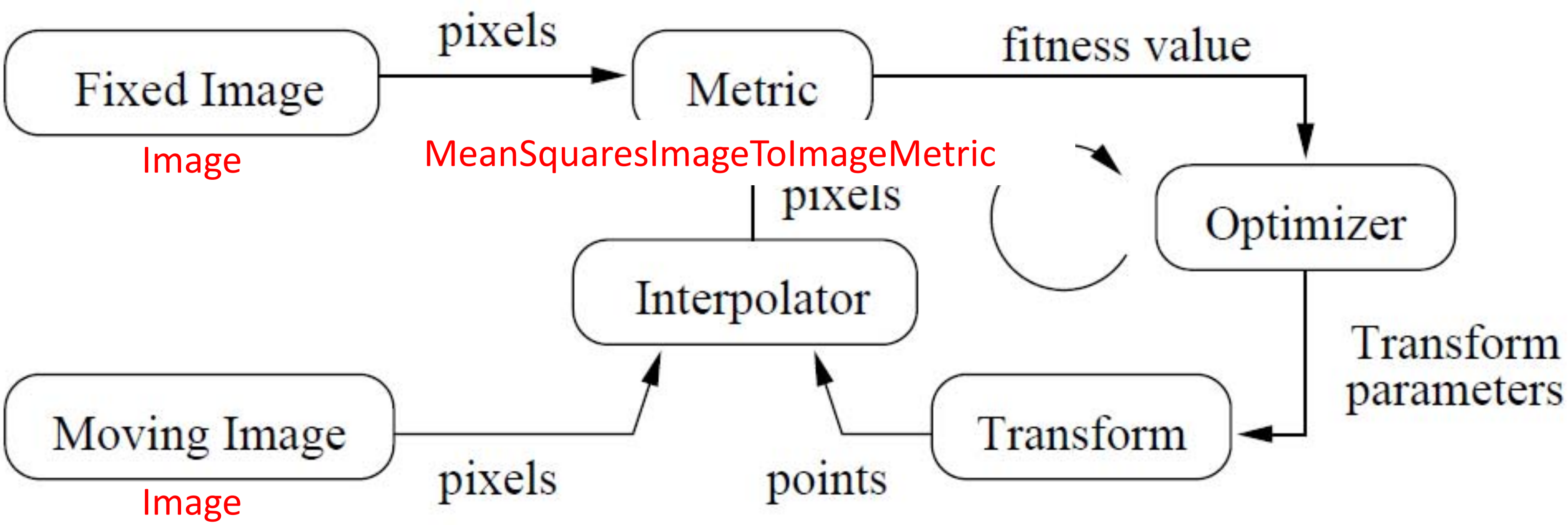
What you are seeing here is the “ITK registration loop” verbatim from the ITK software guide. We can follow this diagram like a recipe to make an ITK-based image registration tool.

ITK registration framework



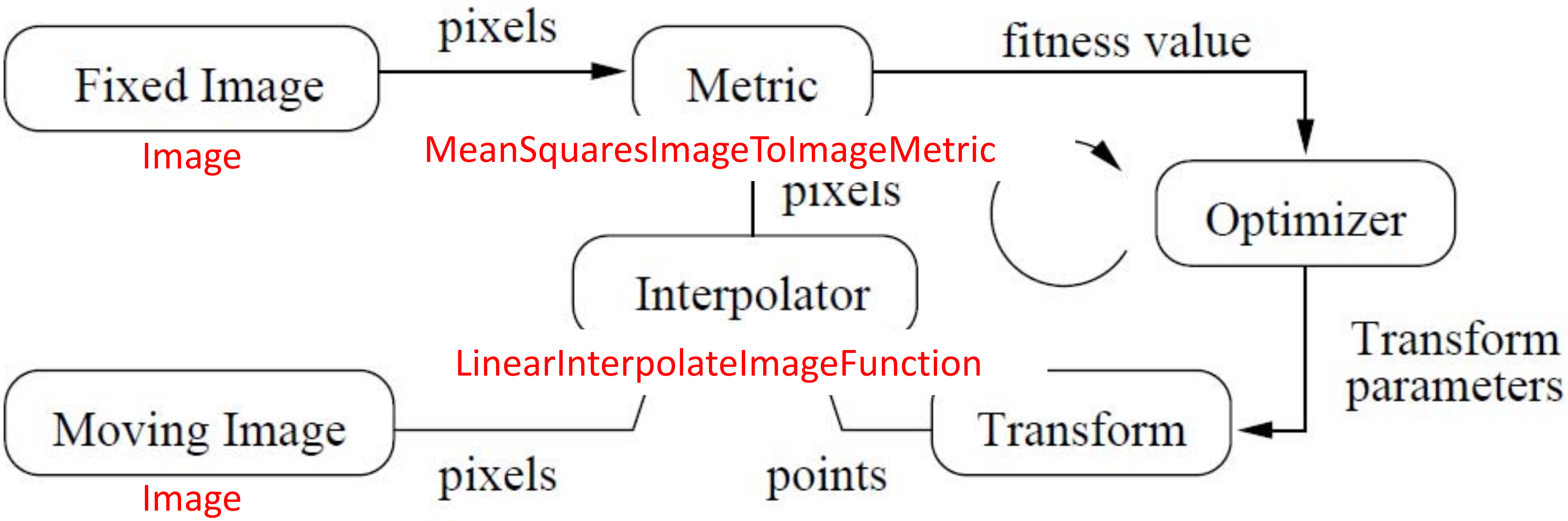
Each of the elements of this framework is implemented using a specific ITK class. The two images are represented by itk::Image.

ITK registration framework



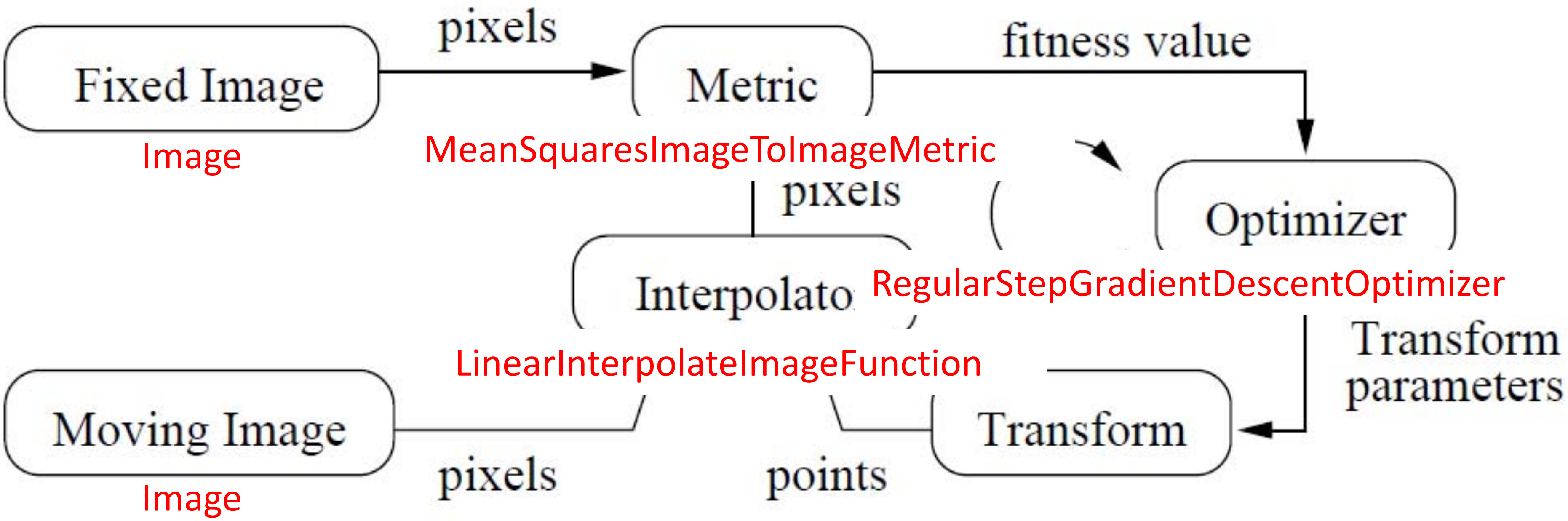
There are several image similarity metrics in ITK. We’ll start with ITK’s implementation of MSE, which is in itk::MeanSquaresImageToImageMetric.

ITK registration framework



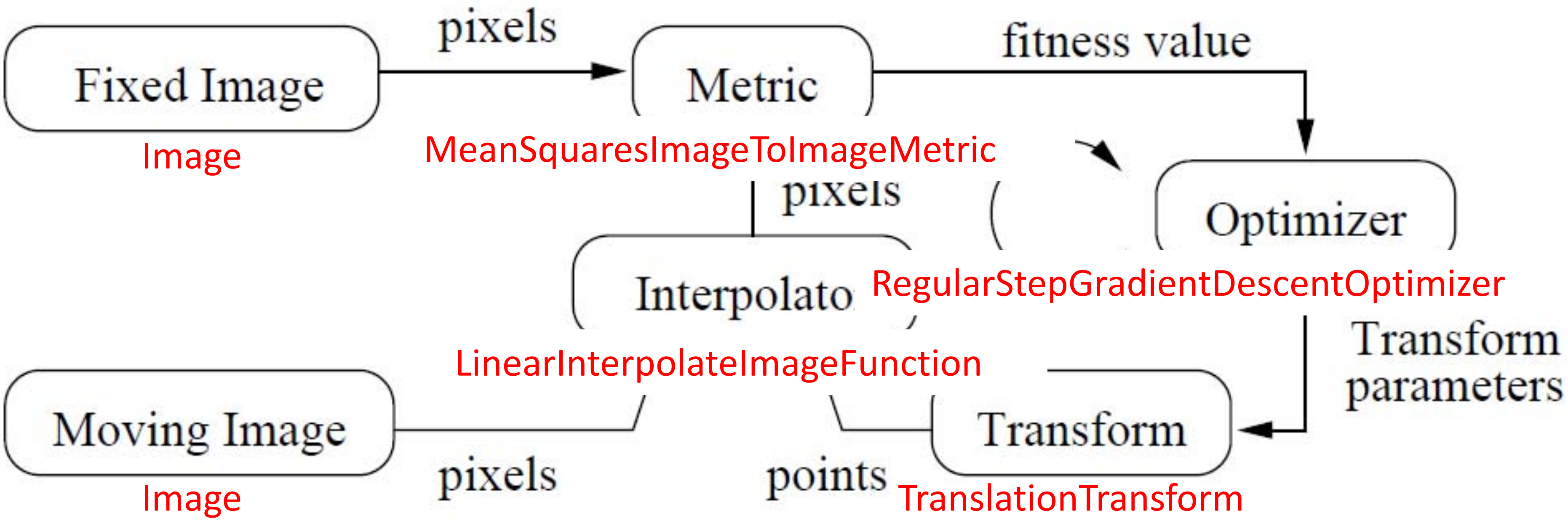
We will resample the transformed moving image using linear interpolation, implemented in itk::LinearInterpolateImageFunction.

ITK registration framework



We will use gradient descent optimization, implemented in itk::RegularStepGradientDescentOptimizer.

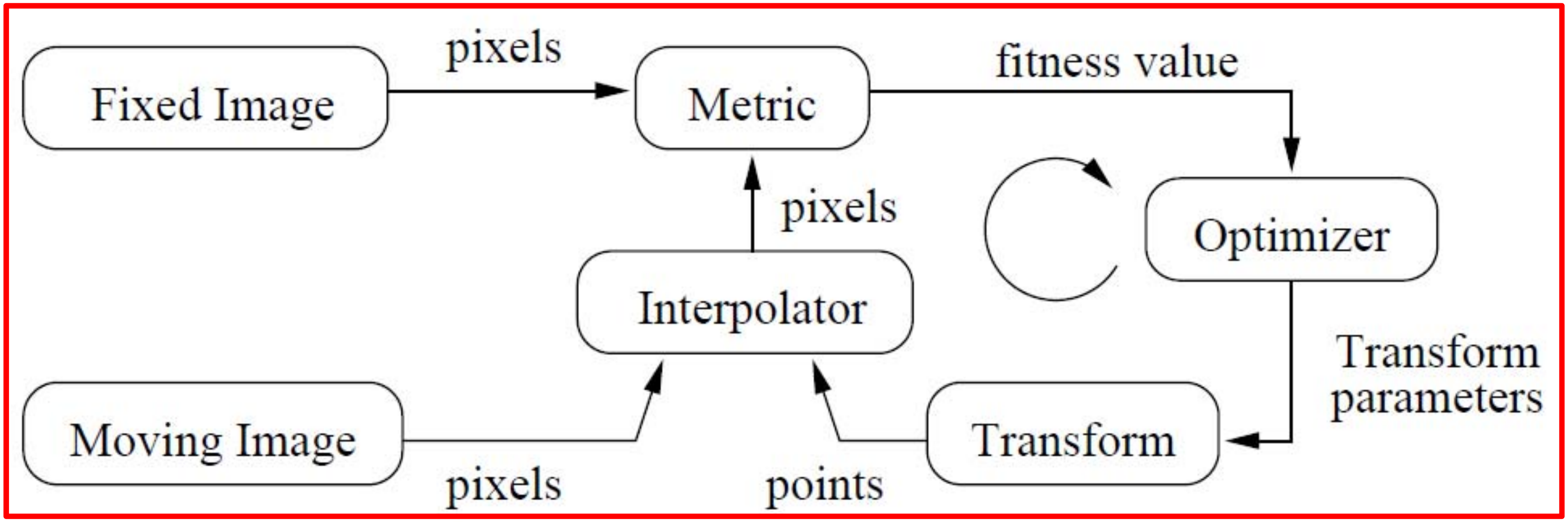
ITK registration framework



And lastly, for our first example, we'll implement X- and Y-translation only, which is implemented in itk::TranslationTransform.

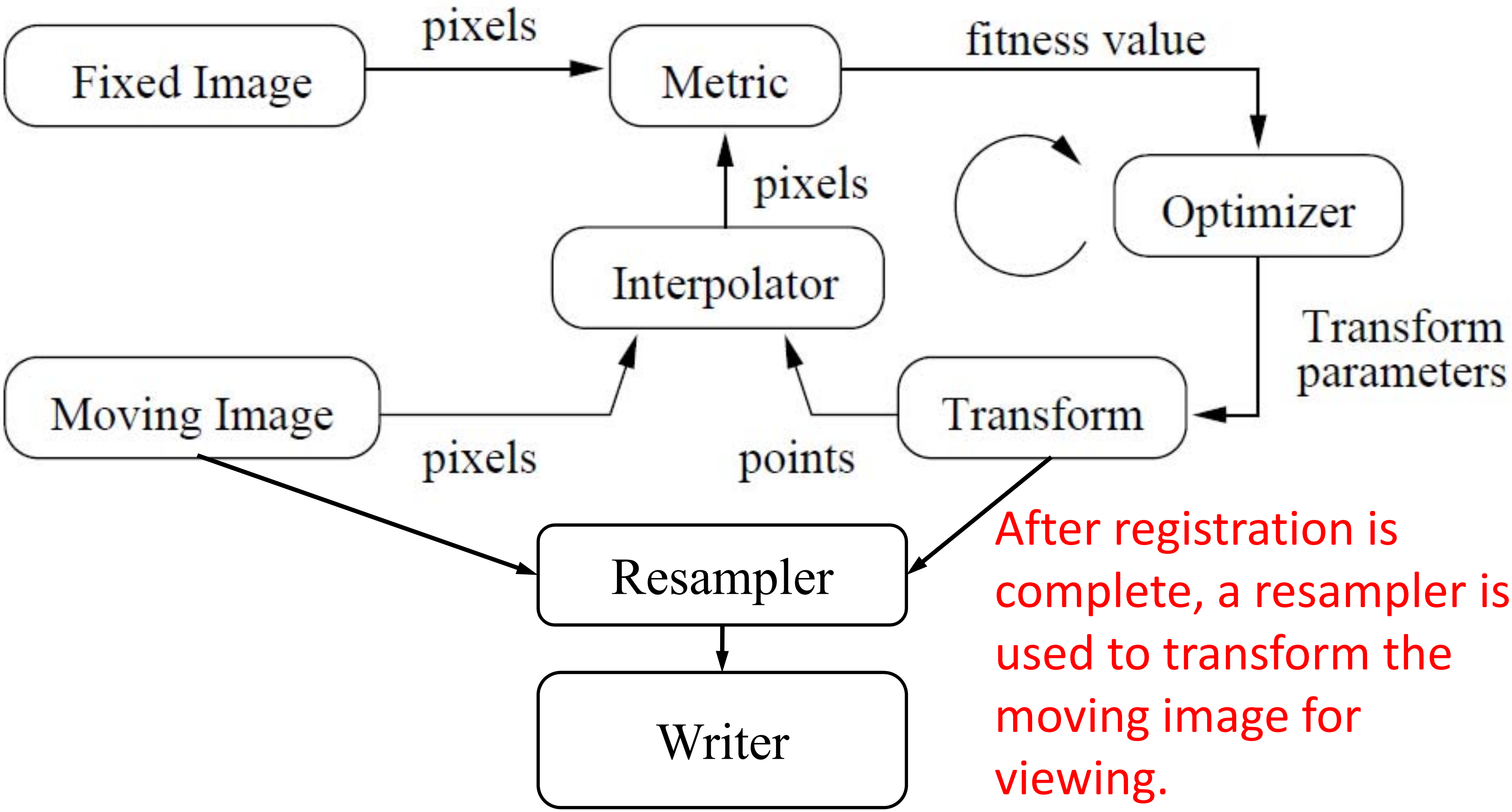
ITK registration framework

ImageRegistrationMethod



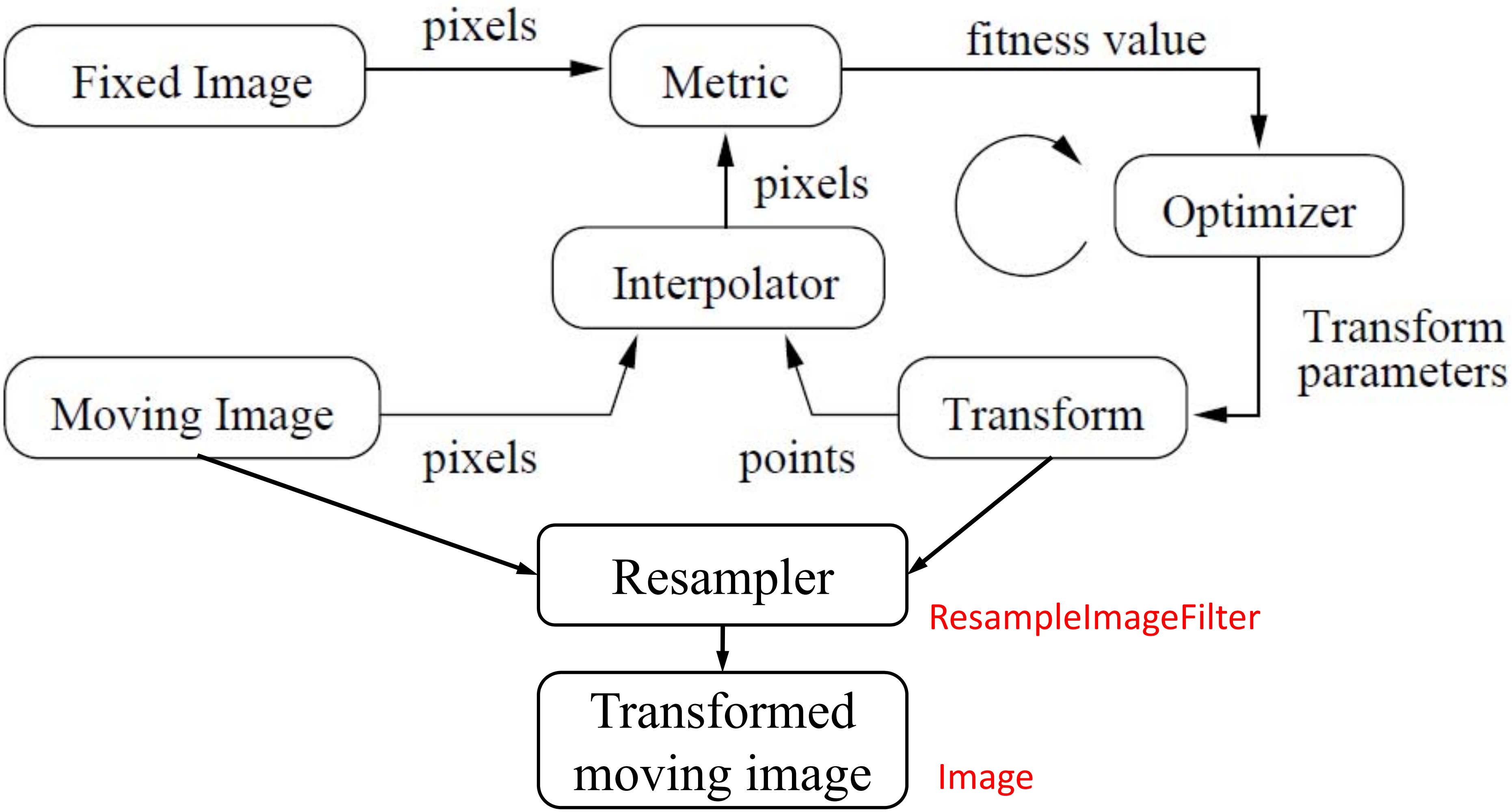
In ITK, all of these “ingredients” are plugged together on a “framework” class called itk::ImageRegistrationMethod.

ITK registration framework



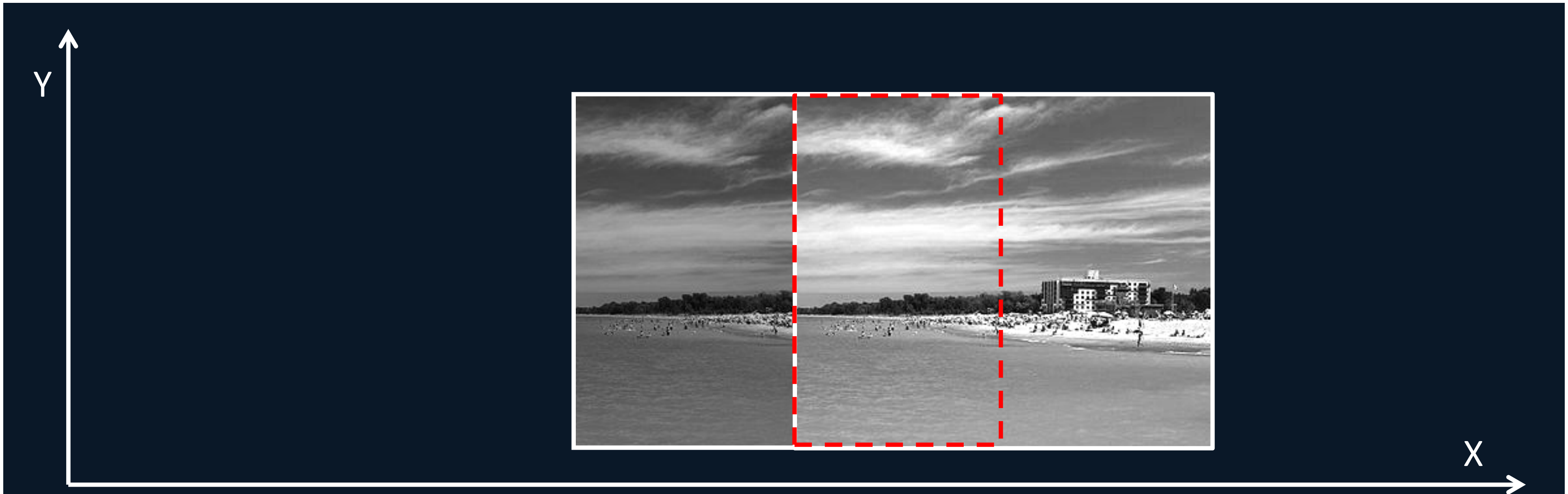
After registration is complete, a resampler is used to transform the moving image for viewing.

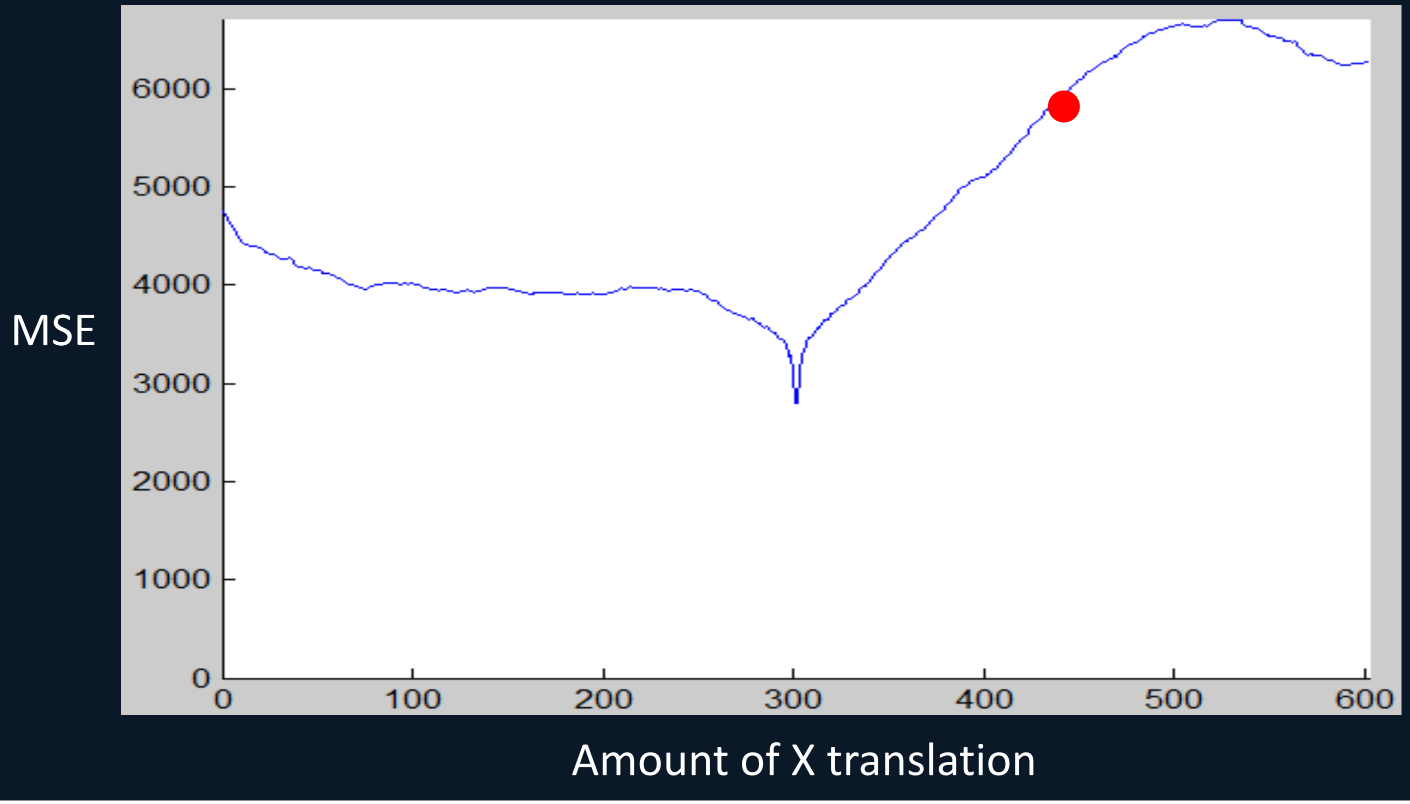
ITK registration framework



ResampleImageFilter

Image


Y
X


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