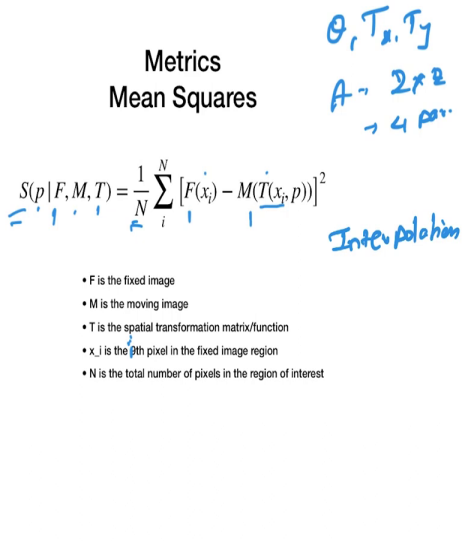
Medical Image Analysis
Professor Ganapathy Krishnamurthi
Department Of Engineering Design
Indian Institute Of Technology, Madras

Lecture 17
Metrics

(Refer Slide Time: 00:14)



Hello and welcome back. So, in this video we are going look at some of the metrics that has that are optimized in order to estimate the parameters of the transformation correct. So, in the last video, we saw a few simple transformations that are typically applied. So, for instance translations and then rotations, we also have looked at the affine transform, which had, N×N matrix along with the translation.

So, the elements of that matrix are typically the parameters of the transformation matrix that is, are the parameters that we typically estimate so, for the rotation matrix, we had the sinθ cosθ matrix, so, that θ is a parameter then this T_x, T_y is a translations that we estimate and for affine transform we had an A matrix for instance, which had n×n parameters.

So, in this case it will be 2×2 or 4 parameters or a_{ij}. So, there will be 4 parameters. So, these parameters have to be estimated based on optimizing a metric. And there are several metrics that are used. But we are going to look at image intensity based metrics typically that are often used, especially in the context of rigid body registration.

So, the first one we are going to look at is the mean squares metric. You can also call it the least squares metric wherein the formula is given here, let me just simplify the rotation S, refers to the metric of course, we are treating is a function of p which are nothing but the parameters of the transformation that we are trying to estimate. F is the fixed image M is the moving image T is the transformation itself.

So, we will see how it comes into the picture. So, the general if we have total number of N pixels, once again, this is some abuse of notation, because we are using capital N, in the previous lecture where we looked at N refers to the dimensions of the image. But here, in this case, we are only N refers to the number of pixels in the image.

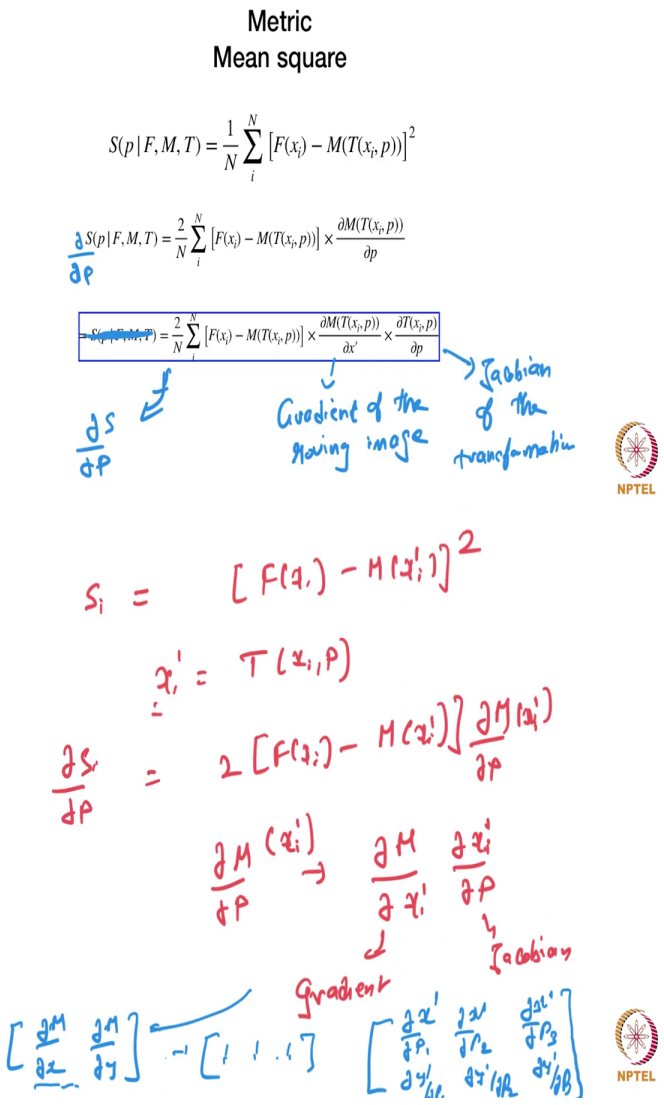
So, or all the pixels, we are going to look at the least squares difference between the fixed image and the transformed moving image. So, every pixel in so we are mapping every pixel in x_i, with this transform T, and estimating the moving image at those points and see after estimation, whether, the images match.

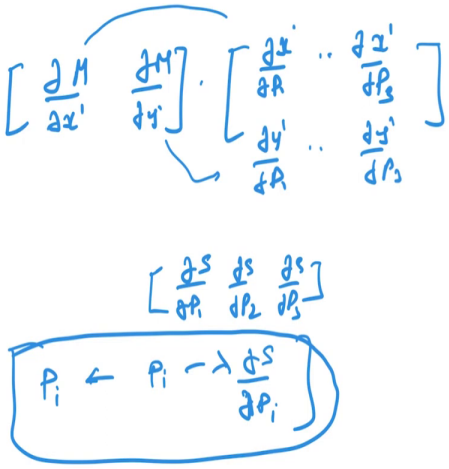
But here actually, I have skipped a step in the sense that the interpolation step, I have not done that, maybe we will go back in we will look at this interpolation step in the maybe in a later slide later lecture, so not later slide. So, here, what happens is once you do the transformation on x_i, with this T, it will give you a physical coordinate space.

But physical coordinate will not exactly map to a pixel index, so there will be some error there. And so, you will have to do an interpolation to estimate your the pixel intensity values at $T(x_i , p)$ after the transformation. So, up to this part is where interpolation comes in typically we use linear or binary interpolation.

So we would not get into the details, but we can talk about it later in a later lecture. So, this is the just to clarify the notation is the there is a spelling mistake here, it is the i^{th} pixel, in the image, x of i is the i^{th} pixel, not the 9^{th} pixel N capital is the total number of pixels. So, you from this formula, how are you going to estimate so we are going to have to minimize this loss function, which means this loss in order to estimate the parameters p of the transform.

(Refer Slide Time: 04:05)





So, if you do that, so for instance, of here, again, there is some error here, it is actually $\frac{\partial S}{\partial p}$ is given by this formula. So, I will clarify the formula here. So, you have the ∂S, this is not correct. Sorry about the error. So, this is actually your $\frac{\partial S}{\partial p}$. It is given by this expression.

So, it is actually the difference between the intensities pixel intensities of the fixed on the moving image after transformation. There is the $\frac{\partial M}{\partial x'}$, this is nothing but the gradient numerical gradient we saw which evaluated of the moving image and this is the Jacobian of the transformation. So, how do we get here?

So, we will just look at one we will just see how this works out when one second. So, we will write this down this the least squares loss function for 1 pixel. So, this S_i is basically for 1 pixel will be $S_i = [F(x_i) - M(x_i')]^2$ this is x_i' writing is nothing but this transform coordinate.

So, now, if I do $\frac{\partial S_i}{\partial p} = 2[F(x_i) - M(x_i')]\frac{\partial M(x_i')}{\partial p}$. i does not come out well most of the time. So, this is the formula. So, now, if we look at the $\frac{\partial M(x_i')}{\partial p} \rightarrow \frac{\partial M}{\partial x_i'}\frac{\partial x_i'}{\partial p}$ plus so prime is a function of P because we have seen this here times delta xi prime delta P this is your gradient this is the Jacobian of the transformation. Now, we look at when we say gradient what do we mean there will be typically be so many components, so, let us say 2 dimensional image, you will have delta M by delta x delta M by delta y.

And what about the Jacobian, Jacobian will have 2 rows and as many columns as there are parameters. So, let us say we are considering 2 parameters P_1 and P_2, so or, let us say 3 parameters. So, you will get $\frac{\partial x'}{\partial p_1}$ $\frac{\partial x'}{\partial p_2}$ $\frac{\partial x'}{\partial p_3}$ and you have similar thing for y. So, I will just let me just write this out and rewrite another side so erase this.

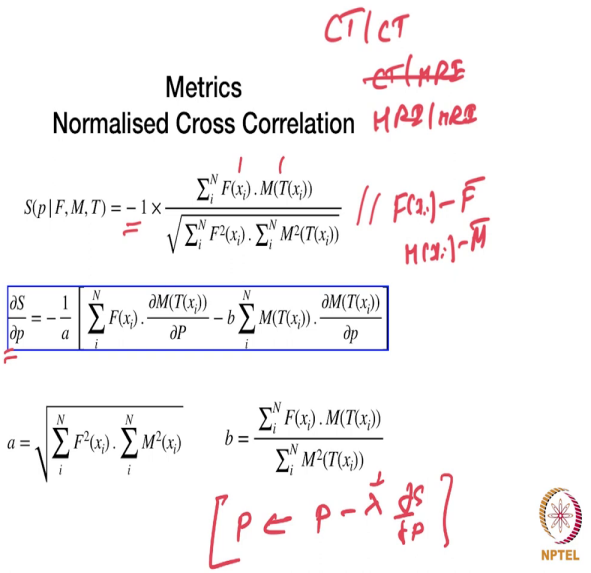
So, this is the gradient so, let us just for the gradient for the Jacobian you will have in this case and it is to 2D. So, you have $\frac{\partial x'}{\partial p_1}$ $\frac{\partial x'}{\partial p_2}$ $\frac{\partial x'}{\partial p_3}$. Similarly, $\frac{\partial y'}{\partial p_1}$ $\frac{\partial y'}{\partial p_2}$ $\frac{\partial y'}{\partial p_3}$. So, you have so, then you what you do is you take dot product with each, this will be dot product with each column.

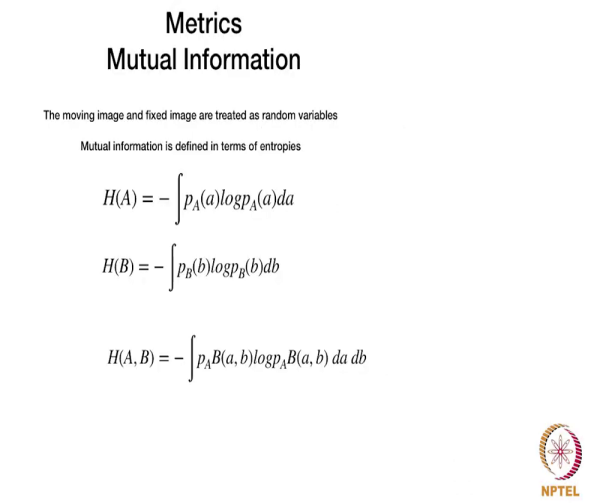
So, we will end up with a 3 vector 3 components each corresponding to the gradient of S with respect to, so these will correspond to $\frac{\partial S}{\partial p_1}$ $\frac{\partial S}{\partial p_2}$ $\frac{\partial S}{\partial p_3}$. So, then what you do with this? So, you would then do gradient descent step. So, that step let me just quickly add.

So, if just to recap what we did there you will get this matrices let me rewrite this so, that you can so, you will have the moving image gradients then, you have a the Jacobian of the transformation $\frac{\partial x'}{\partial p_1}$ $\frac{\partial x'}{\partial p_2}$ $\frac{\partial x'}{\partial p_3}$ then you have $\frac{\partial y'}{\partial p_1}$ $\frac{\partial y'}{\partial p_2}$ $\frac{\partial y'}{\partial p_3}$ and you multiplying row column so, you will have 3 we will end up with 3 which would correspond to $\frac{\partial S}{\partial p_1}$ $\frac{\partial S}{\partial p_2}$ $\frac{\partial S}{\partial p_3}$.

So, then you would do the gradient descent step which is basically $P_i \leftarrow P_i - \lambda\frac{\partial S}{\partial p_i}$. So, you would do the gradient descent to estimate the parameters. So, we will see that this is the recurring case for all the other things that are the other loss functions or metrics that we will use.

(Refer Slide Time: 10:13)





So, the other metric which is basically dependent on the intensity is the normalized cross correlation. So, once again this minus sign is used to make it a minimization problem typically, you want to maximize the correlation between the going and the fixed image after you do the transformation.

But, if you want to make the minimization or gradient descent does you put the minus sign in front the maximum value of this particular expression so, basically you are multiplying them pixel by pixel after transformation and then you are normalizing it with their squares in respect to squares.

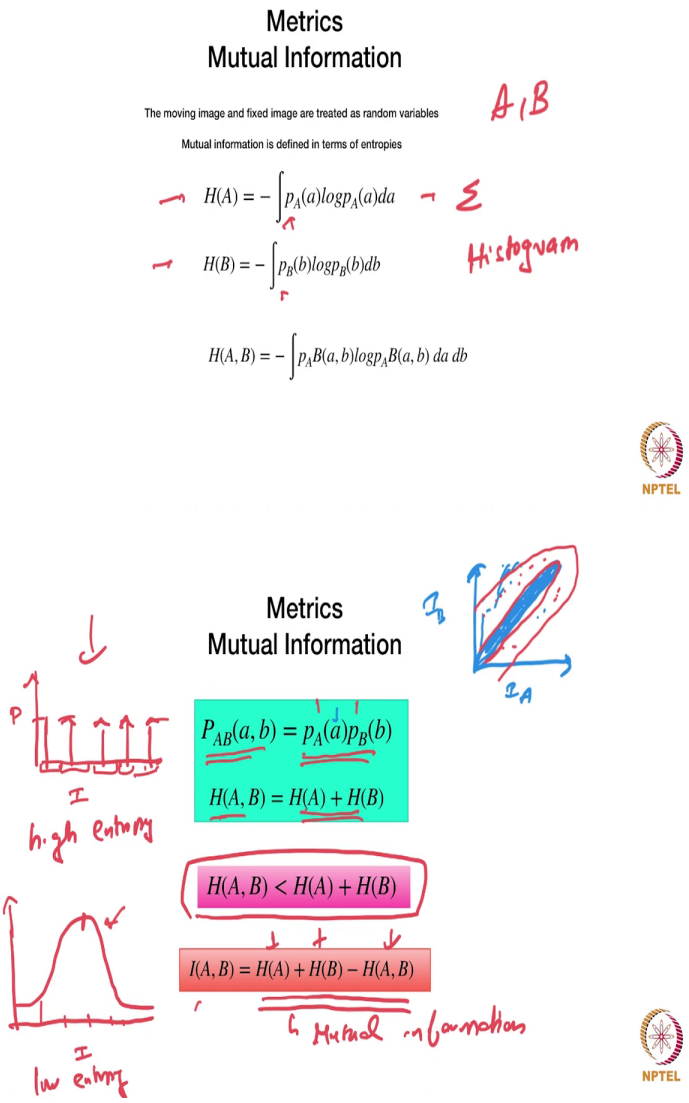
One of the things you would do not shown here is you would subtract the mean so, you would actually $F(x_i)$ - $\bar{F}$ will be done $M(x_i)$ - $\bar{M}$ will be done. So, which $\bar{M}$ and $\bar{F}$ are nothing but the mean of the images. This particular technique makes this metric immune to arbitrary, DC shifts in the images, you can multiply every pixel value by some arbitrary constant or add any arbitrary constant the pixel value and robust to that.

So, once again, you can go through and do the calculation ∂S or ∂p like I said, this is an important step because once you figure out $\frac{\partial S}{\partial p}$ all you have to do is $P_i \leftarrow P_i - \lambda\frac{\partial S}{\partial p_i}$, this is the gradient descent step. So, once you take this step, then you once again evaluate this metric calculate how much you have to move.

You also have to parameterize this λ this hyper parameter λ you have to adjust based on the problem**.** So, it is normalized cross correlation is also used both the least squares metric and normalized cross correlation, very good to use if you are looking at same time. So, for instance, if you are going to do CT to CT, or MRI to MRI, or basically you are trying to align images from the same imaging modality to CT MRI.

And in fact, intra subject is even better. So, that is a very good these two are very good metrics to use when you are trying to do this kind of alignment. Now, the problem starts when you are trying to CT to MRI, CT to pet MR to pet or CT to ultrasound, MR to ultrasound, they are the pixel intensities are or what intensities are completely different. And these metrics might not work as you might expect. So, in that case, what you do so that is when you have this particular metric called mutual information.

(Refer Slide Time: 12:54)



The mutual information is defined in terms of entropy of an image. So, what is the definition of entropy is given here, let us say you have two images, in this case A and B, the entropy of each of these images are given by these formulas. Of course, this is in the continuous domain for the, for a discrete domain, like in an image, this converges to a summation over the pixel values.

Or the various bins in an image. So, how do we estimate this probability density function, this P_A and P_B are the probability density functions of an image. So, these are typically estimated using the histogram. Very simple ways to construct a name for a histogram there, of course, there are some kernel based methods.

You would do to actually estimate this, make it into a continuous function, but histogram is a pretty good way of doing it. So, what this mutual information that we want to maximize mutual information and how does this work? So, you might ask, what is the idea behind maximizing mutual information is the question that we are trying, we will try to understand here.

So, what we are assuming is that, let us say we have this case, where these two images we assume are independent. Assume that both these images are independent. So, then, let us just go before we go here, so let us assume that these two images are independence in the joint probability of these two images.

The joint property of these two images given that they are independent, have nothing to do with each other is basically, or that they are not correlated at all, is that it is the product of the individual probability distributions. So, if the images are not correlated, or the this here if you say more specifically I should say the images are independent of each other, you cannot predict anything about one image using another image correct.

So, you have fixed on moving image and assume that you can predict nothing about the moving image using the fixed image, then you have the joint probability of those two images, pixel values of those two images is given by the product of the individual probabilities here, these are the probability density functions corresponding to the pixel intensity values.

So, what is the probability of a certain pixel having a value 100 that is what this shows? So, this A and B's are nothing but pixel intensity values. So, we are treating them like random variables. So, the joint so, if you, look at the, what I can call the joint entropy? And the joint entropy H(B) in this H(A,B) is given by the sum of them.

Now, let us assume that in the case that they are not independent there are they are not independent, they are not independent, then we can easily write this down, we can write this in this form with there is an extra term that comes in which excluded. So, $H(A,B) < H(A) + H(B)$. Now, what we want to do is look at this difference.

This difference with this is this mutual this is what we call mutual information. And we want to maximize this difference. So, because what we want you to be able to do following registration is that, we should be able to say something about the moving image based on the fixed image.

That is what we ideally wanted that would, that would suggest a very good registration. So, then the idea is to for to make this difference bigger. So, we want to be as far away from independence as possible. That is the goal. So, then we want to individually maximize the centerpiece while minimizing the centerpiece.

So, this the joint, so we are to minimize this joint entropy while maximizing the individual interface. So, from the point of view of, understanding what the image entropy means. So, if you consider, let us say, it is a distribution of pixels, which are, I am going to just draw a very simple picture here.

We have, let us say, 10 pixel levels, a 10 pixel levels, I will draw 10 of them, but then even some n pixel levels, and all of them have the same probability. So, this is, let us say, the problem, the axis this is the intensity, this is a random variable, and its corresponding probability. So, I am just putting one arrow there just to see they are all the same.

So, this has very high entropy. Because it is very difficult for you to predict, if you point to a particular pixel, you cannot predict its value with a great conference, because it is all pixel values are equally probable. On the other hand, let us consider a pixel distribution, which looks like this very sharply peak.

Once again, this is the pixel intensity bins, the probability you see that, we can with some degree of certainty predict the pixel value, this has low entropy. So, we actually want to increase the entropy of these individual images that are being registered, especially in the region of overlap, because this high entropy means that you have a very heterogeneous region in the image region.

That is what this high entropy translates to a low entropy translates to a very uniform smooth region. So, we want to be overlapping in both the cases the region of overlap, for both A and B are fixed and moving image should correspond to regions of high entropy, which means they have some structure there. That is the one we want to be matching.

On the other hand, when you consider the joint probability distribution, joint probability distribution, meaning, what is the probability that certain a pair of pixel intensities co occur, the co-occurrence of pixel intensities, that is what we typically would mean by joint by this joint entropy and sorry the joint distribution of these pixel values.

And you want them to be sharply peaked for maximum alignment, correct. So, if we are if they are aligned properly, then you have you will see that the 2-D histogram so we are looking at a 2-D histogram will be a very sharp histogram. So, what do I mean by a sharp histogram? Let me just draw this quickly.

So, if you have the joint histogram, which means that you are looking at probability of co-occurrence of pixel intensities. So, here, when you say $P_A(a)$, we are just looking at a particular pixel value A and its probability of occurrence, you just count the number of times it means intensity occurs.

Or the other ways of also doing this, if we have a range of a continuous range of intensity values, but we would not get into those details. But as far as the joint histogram concern, you have, let us say, two axis, it is a 2-D histogram. So, you are looking at, let us say, in this case, intensity along of A intensity values from A, intensity values from B.

Here, and then you are looking at a histogram of these pairs of values occurring, and we count the number of times a pair of values occur together divided by the total number of pixels. Now, in that case, you want a sharply peaked histogram, when I say sharply peaked histogram. So, if you look at this, in 2-D, they have two sets, two axis, corresponding intensities from each of the images.

So, you it would lie the, the high probability regions should be here. So, it is a sharply peak, because this is the high probability region, everywhere else it will be very sparse in the sense will be close to 0 here. So, you can think of we have to think of it the 3-D where the this shaded portion is basically pricing out of the plane of the screen.

So, this requires, this implies, very bright, very large values of probability. And, in other regions, it is very low values of probability correct. So, this is what I mean by sharply peaked histogram in 2 dimensions, so will be a thin line, but which vary, the probability here in this region is very high probability outside decisions is very low.

And if the images are not properly aligned, then you will have that this this becomes slightly worse, in a sense, you will have a much, much larger region with, higher probabilities or in this case, the probability spread all over the place, like a uniform probability it like look that way.

So, this is the, this situation this particular situation I am talking about this blue band is similar to what I have shown here, for one image. And this other situation where I have shown this red region where your most a lot of pics co-occurrence of these pixel intensities is equally probable that corresponds to a poor alignment, this has been observed empirically also.

So, for optimizing this I(A,B), you would expect H(A) and H(B) to be high and H(A,B) to be much lower. So, this concludes our look at different metrics used for image registration, especially rigid station, when we are looking at, just rigid body transformations, rotations translations, maybe some affine transforms.

We are looking at cases where we are doing interest subject registration, for the same patient with the same measuring modalities, wherein, we can use normalized cross correlation or just sum of squares, least squares. In this case, if you have different imaging modalities, then you can use the mutual information as a metric for image registration. Thank you.