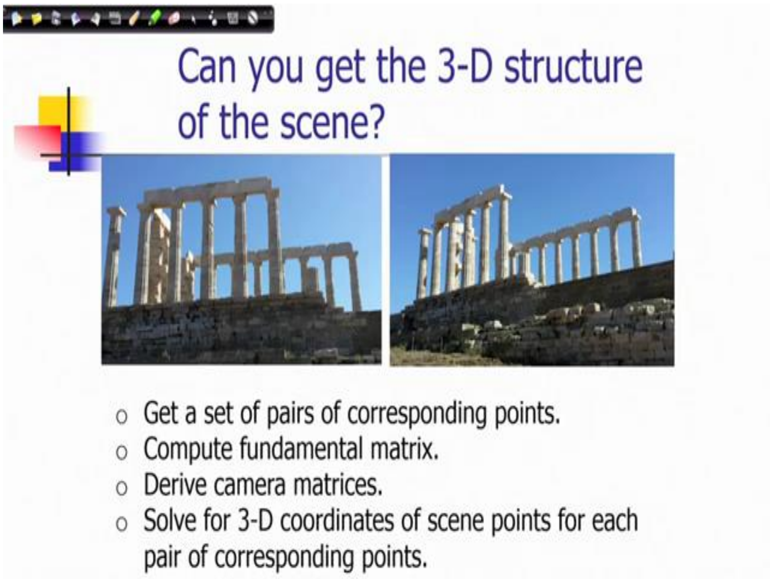
Computer Vision
Prof. Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 29
Feature Matching and Model Fitting – Part I

In this lecture we will start on a new topic on Feature Matching and Model Fitting.

(Refer Slide Time: 00:24)



Let us consider a typical problem where you require this kind of matching of features for example, you would like to compute the 3-D structure of a scene. So, we have already gone through the computational aspects of this particular problem and we can design an algorithm with some computational steps to solve it. So, what we can do? We can get a set of pairs of corresponding points, that is the first thing we should get and in fact, this is a step, where we will see that matching is very much required.

But just to give a holistic view of the solution of this problem. So, you need a set of pairs of corresponding points, then you should compute the fundamental matrix, and then we should derive the camera matrices and then we can solve for 3-D coordinates of scene points for each pair of corresponding points. So, you see that your initial assumption. So, having a set of pairs of corresponding points that itself requires some introspection. So, for in our computations we have assumed they are available either by visually finding out which points corresponds to what. So, let us consider these particular structures.

(Refer Slide Time: 01:51)

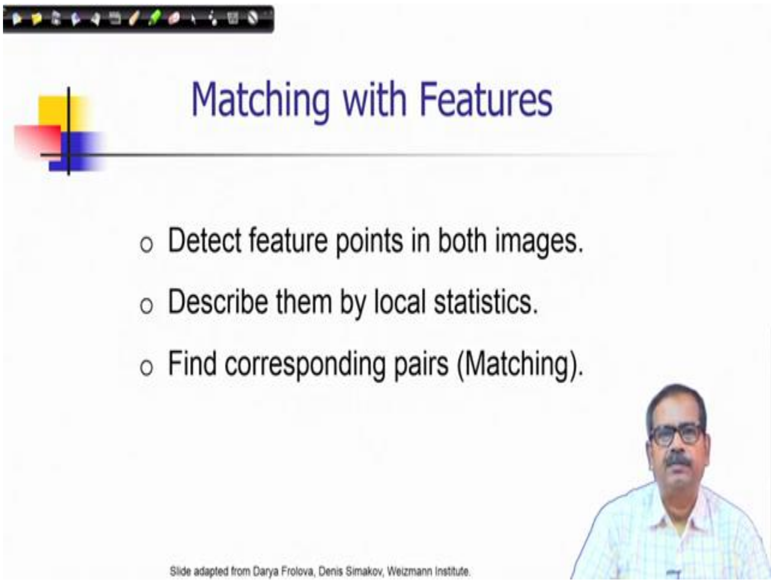
So, what we can do at least if you would like to automate this process, we would like to find out first thing interesting points in those images as the feature detection we have discussed. So, we can apply those transformational invariant feature detectors like SIFT detector, Harris corner detectors and also apply the detections in a scale invariant and transformation invariant manner.

So, consider this particular operations where you define a measure like Harris corner measure and then from there you get the feature locations by finding out the local maxima out of this set of points and those points define your feature points. Similarly for the other view also you perform the similar operations.

So, now since you expect that these points which you have detected their transformation invariant, which means even after the change of view in the imaging still those points will be retained in your detect after detections. So, we could try to find out that what are the correspondences or which point corresponds to which point of the other image, which means that we need to find out this kind of relationships where a point in the left image corresponds to the image another point in the right image and their relation is that they are images of the same point. So, you can check with this particular thing.

So, we are trying to find out for example, by precisely locating the feature points we are trying to get a correspondence and in this way you would like to get the set of corresponding points. So, computation which involves in pairing these points or getting the corresponding pairs of corresponding points that is what is the what is known as feature matching. So, we will be discussing some of these techniques about this feature match.

(Refer Slide Time: 04:27)

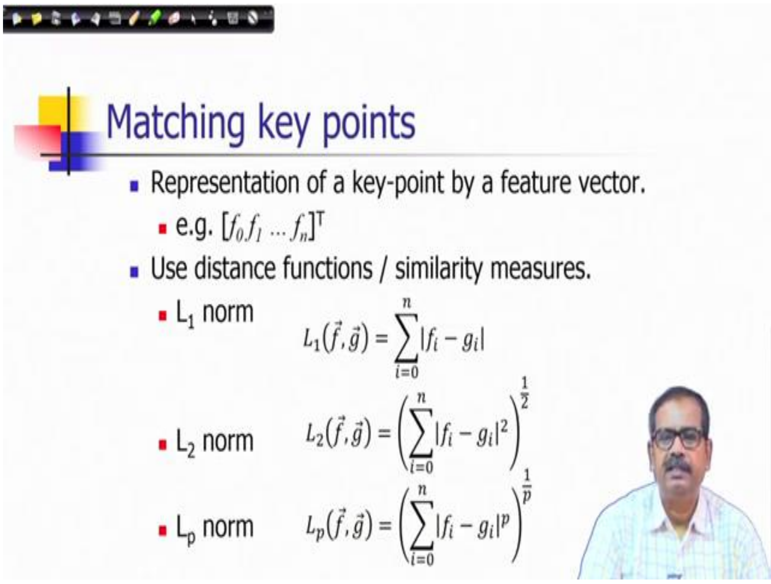


So, this is the summary of that computations that we discussed that first you have to detect feature points in both images and then you should describe them by local statistics. So, this is the way how these points are to be you know matched that just their locations will not give any particular information about the nature of the point.

So, you need to at least look at the neighboring statistics neighboring distribution of intensity values or some functional you know distributions in it is particular neighborhood and which you are expecting that would also remain the similar in the other image of the similar of the same landmark point around that same key point and then exploiting that similarity we are trying to match them.

So, you have to describe them by local statistics and we have discussed about different feature description techniques like shift descriptors or what we different kinds of descriptors and which have the properties of these transformation ingredients and then you have to find corresponding pairs. So, we will be considering that what competition involves suppose we are given those descriptions, then how do you say that this pairs this pair of points are the corresponding points?

(Refer Slide Time: 05:53)



So, we have already discussed in the previous topic also because while explaining the motivations of describing a key point that one of the computations that we consider that matching is in this representation. So, same description or same discussions will be repeating here that, you can represent a key point by a feature vector that you have considered. That suppose we have a feature representation of a vector here it is an n plus 1 dimensional vector $[f_0, f_1 \dots f_n]^T$. And then you can use some distance function to define the proximity or similarity between two vectors, you can other than distance function also directly you can use any similarity measures also.

So, some of the examples of this distance function are given here like you can use L1 norm,

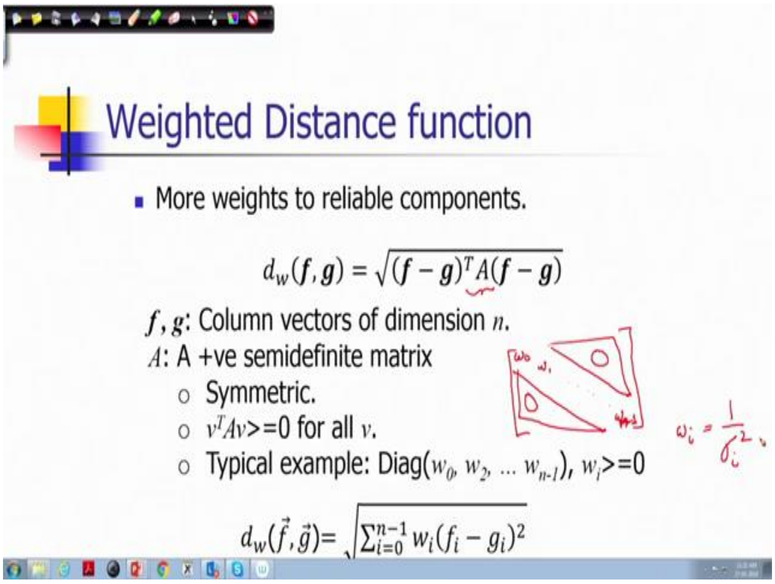
$$L_1(\vec{f},\vec{g}) = \sum_{i=0}^{n} |f_i - g_i|$$

$$L_2(\vec{f},\vec{g}) = (\sum_{i=0}^{n} |f_i - g_i|^2)^{1/2}$$

$$L_p(\vec{f},\vec{g}) = (\sum_{i=0}^{n} |f_i - g_i|^p)^{1/p}$$

So, you can use these distance functions to define the proximities between key points and you can apply certain strategies we will discuss that later on that how to declare that two key points, they correspond to the same scene point for any way for two images.

(Refer Slide Time: 08:21)



Sometimes no distance functions you can have weighted distance functions which means in the in our previous examples of distance functions, we considered every component has uniform weight we did not distinguish that which component is more reliable and which component is less reliable. So, if it is more reliable we give more weight to that differences if it is less contributing to the discrimination of the images reliable means this it is contribution to the discrimination between two images.

$$d_w(f,g) = \sqrt{(f-g)^T A(f-g)}$$

So, if it is less then we give placement. So, this can be represented in this particular mathematical form as you can see that, it is a it is used the column vector representation of a feature vector and using matrix operations, you can compute this distance. So, in this expression particularly note that A is a positive semidefinite matrix and which means that it is a symmetric matrix and if I perform this operation

$$v^T Av \geq 0, for\ all\ v$$

So, its value should be positive with respect to the same vector and typical example of A is given here. So, that this is a diagonal matrix which we can represent in this form that you can write

$$Diag(w_0, w_2, \dots w_{n-1}), w_i \geq 0$$

So, w 0 today in the previous example we have the index from 0 to n which was in n+1 dimension space. So, this is the diagonal matrix and all others are 0. So, this part and this part they are all 0. So, if I multiply if I use this diagonal matrix here in this expression then it is simply this expression will boil down to in this form

$$d_w(\vec{f}, \vec{g}) = \sqrt{\sum_{i=0}^{n-1} w_i (f_i - g_i)^2}$$

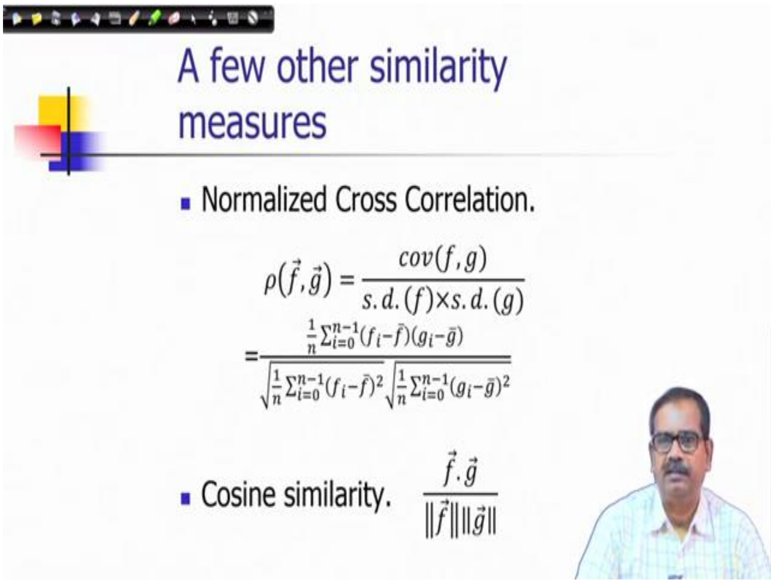
So, one typical example could be that they could be weighted by inverse of the standard deviations of the variations in that ith component of the feature vectors say we can consider or variants rather.

So, if I consider

$$w_i = \frac{1}{\sigma_i^2}$$

Where σ_i is the standard deviation of the ith component of this feature vectors you consider all possible feature vectors. So, what is the variability on the ith component and then if it if it has more variability we give less weight if it has less variability we give more weight to define this distance. So, this could be one such policy while defining this distances one typical example of weighted distance function when we can use $w_i = \frac{1}{\sigma_i^2}$ and σ_i is the standard deviation of the ith component if I consider the statistical distribution of the ith component of these vectors.

(Refer Slide Time: 12:19)



There could be other kinds of similarity measures. So, instead of distances by distance compute the distance between two feature vectors means smaller the distance bit greater is a chance of having them or declaring them similar or the smallest distance between a pairs of feature vector may indicate that they are the matching pairs whereas, for similarity it is just the inverse relationship.

Where in a similarity measure the higher the similarity value higher is the chance that they are matching candidates. So, two such similarity measures which are used here one is called normalized cross correlation measure and the other one is cosine similarity. So, let us consider their mathematical forms.

So, this is the normalized cross correlation measure which you can see that it is defined as the correlation between no two distributions where here distributions are considered in terms of their components. So,

$$\rho(\vec{f},\vec{g}) = \frac{cov(f,g)}{s.d.(f)Xs.d.(g)}$$

$$= \frac{\frac{1}{n}\sum_{i=0}^{n-1}(f_i-\bar{f})(g_i-\bar{g})}{\sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(f_i-\bar{f})^2}\sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(g_i-\bar{g})^2}}$$

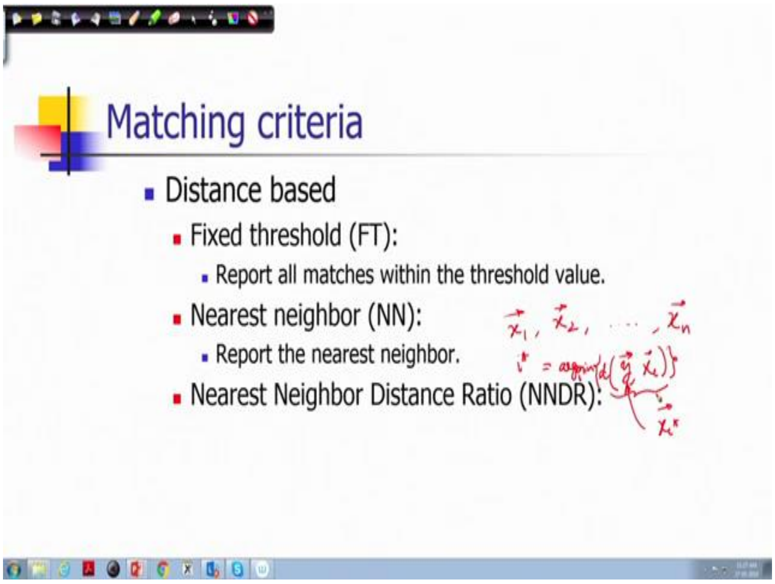
They have been expanded also in my definitions and as I mentioned that, here we are considering the values of components across the vectors and we are comparing those variabilities with respect to two different vectors. And if they are very similar this group cross correlation value should be you know high actually it ranges from [-1, 1].

So, if they are very highly similar then it should be close to one whereas, the other function which is called cosine similarity that is defined in this way.

$$Cosine\ similarity = \frac{\vec{f}.\vec{g}}{\left|\left|\vec{f}\right|\right| \, ||\vec{g}||}$$

So, as you understand here also, if the vectors are very similar these angles should be near to 0. So, which means the cosine of that angle 0 should be equal to 1. So, here also the values range from [-1, 1] and the values which are nearby and which are very similar there the value should be high, it should be close to 1.

(Refer Slide Time: 15:09)



So, let us now consider what kind of different matching criteria could be there. So, I will consider distance based matching criteria mostly here and you can extend this discussion or extend these ideas for using similarity based matching. So, in the distance based matching the minimum is the distance or smaller is the distance better is the you know matching between two vectors, that is a way that this policy is considered. So, one of the policy of distance based matching is that, you can use a fixed threshold which means you can report all matches within that threshold value.

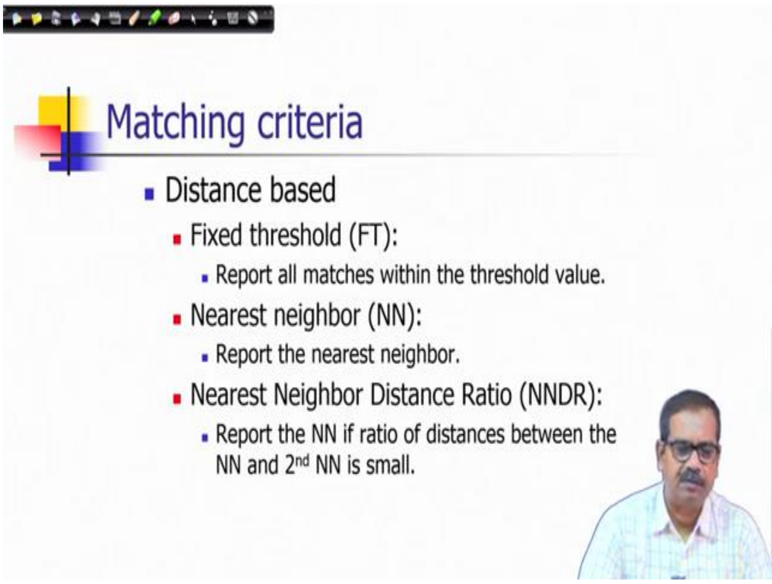
So, it is not declaring just a pair of feature vectors, but given a feature vector it is providing you a candidate feature vectors and then you may have to do some post processing to select which one is the fittest or which one the actual feature vector is corresponding to the query feature. Similarly, but if you would like to precisely define the corresponding feature vector, the nearest neighbor definition is much more useful because here it is not a fixed threshold.

So, it considers which feature vector is nearest to it. The problem is that you need to also consider; that means, it does not consider that whether the the distance between them is greater than certain threshold value or within certain you can combine them of course, but by definition nearest neighbor is the feature vector corresponding to that.

So, if I say that there are feature vectors say [x1, x2, …, xn] those are the candidate feature vectors and you have a feature vector y in another image and this is the feature vectors in the other image. So, what you are doing? You are computing distance between this each feature vectors and which one is the smallest out of them. So, you take the minimum of them that operation called arg min and that would give you the corresponding feature vector; that means, xi* that is a nearest neighbor of y.

But it may happen that the distance is greater than the fixed threshold what you have considered. So, you can apply that consideration also while selecting this nearest neighbor point. So, this is the strategy when you are considering nearest neighbor principle of detecting or matching a pair of feature vectors.

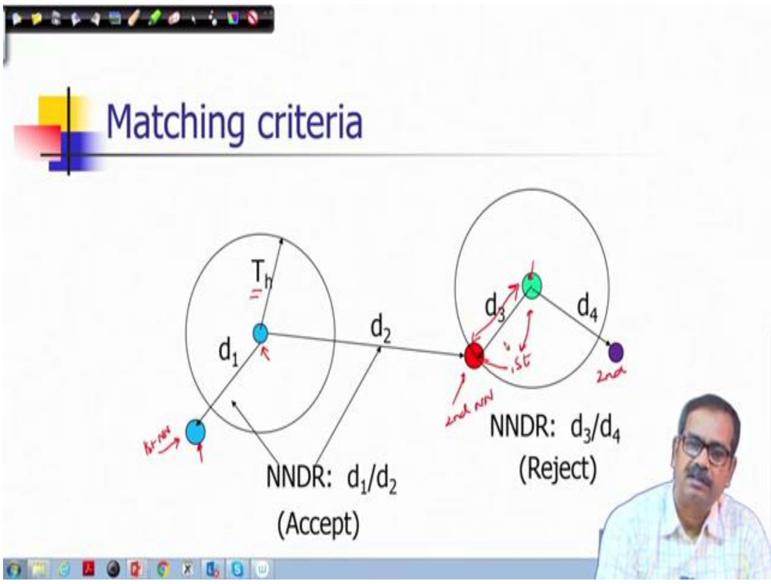
(Refer Slide Time: 18:22)



The other strategy which is found to be more robust is called nearest neighbor distance ratio. So, in this case it considers the ratio of distances between the nearest neighbor and the second nearest neighbor. So, that means, if it is distinctly nearest, the next neighbor is quite far away then we should accept that as a reliable matching between these two feature vectors. So, which means this ratio should be very small it is a distance between the nearest neighbor and the second nearest neighbor.

So, ratio should be very small and if this ratio is very small then you can consider that this is a good matching point. Here also we are not giving an absolute criteria it is a relative it is relative distances what we are comparing. So, relative comparison between two distances one with the one distance is with the nearest neighbor and the other one is with the second nearest neighbor.

(Refer Slide Time: 19:26)



So, there is a figure by which you can understand this particular process. So, in this example you can see that the feature vectors which are actually closer in the space they are colored by the same color. So, for example, if I use a fixed threshold of th, then all the feature vectors within these circular region around it. So, here we are showing a two dimensional space.

So, it could be sphere, if it is a three dimensional feature time feature vector or for an n dimensional feature vector it would be a hyper sphere. So, anyway. So, within this region if there is any feature vector that would have been reported, but what happened the nearest neighbor of this one is just outside of the circle. So, if you use fixed threshold you will be missing this.

If we use nearest neighbor principle then this will be also selected which means you get a true match according to this particular configuration and also it matches with the ground truth as you have considered that they are the two matching points closest neighbor. But if I consider nearest neighbor principle then what happens you see that for this point this is the nearest neighbor.

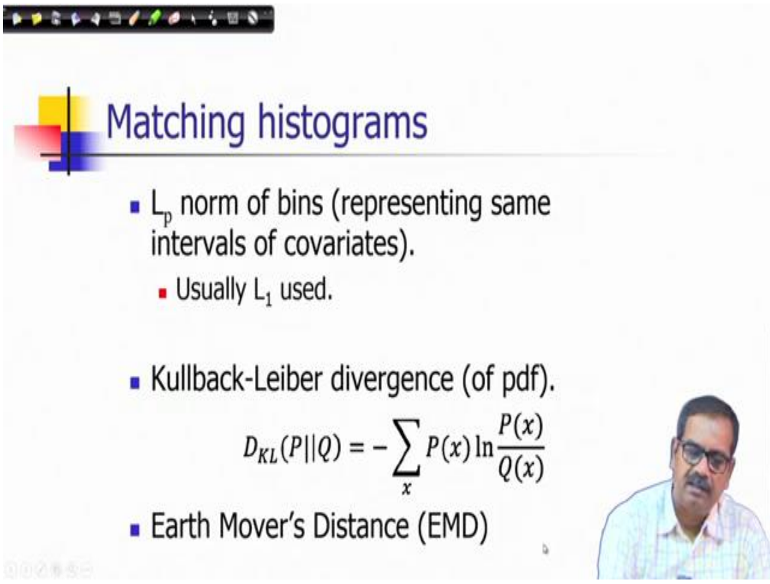
But it is not the corresponding matching point it is showing with the different color in this particular case. So, though it is nearest neighbor, but still it should not be considered. So, that they are the nearest neighbor distance rule comes if I consider the nearest neighbor distance rule then with respect to this particular point, this is the first nearest neighbor which means that this is the nearest neighbor and this is the second nearest neighbor with respect to this point.

So, if I take the ratio of between these two and this ratio is expected to be small according to this diagram and then also we accept it. Which means that even the principle of nearest neighbor distance ratio which is shown here in acronym NNDR, even if you apply them then also it is accepted whereas, if I consider the other situation you can see that here this the with respect to this image this point this one is the first nearest neighbor with respect to this one and this is a second nearest neighbor.

And the second nearest neighbor is also quite close as you can see, because no there is an ambiguity of descriptors in this case. And if I take the ratio d3/d4 then this ratio is expected to be higher and then with the appropriate thresholding you can consider and you can reject them.

So, this policy in NNDR policy it is also rejecting in this case and which is desirable, because the using nearest neighbor principle this will be assigned this list will be matched using nearest neighbor principle, but which is not desirable according to the given data. So, that is why policy of nearest neighbor using distance ratio is found to be more robust than the other policies.

(Refer Slide Time: 23:12)



So, we discussed about matching of key points here using their feature descriptors, what about regions or even the images where you represent a feature description a feature where you have a feature descriptor in the form of histograms of certain measurable quantity so, there you require matching of histograms.

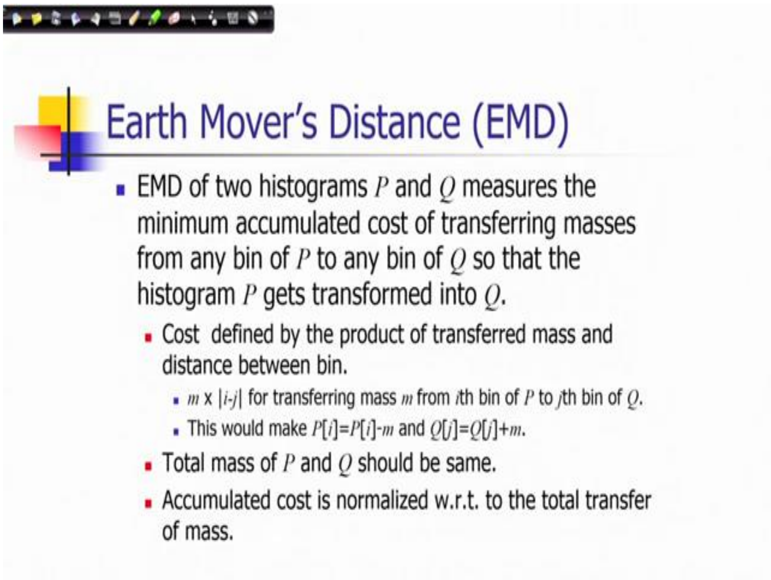
Now histograms also can be considered as a feature vector each bin represents as a component of a feature vector. So, there you can use usual distance functions what we discussed in the previous case also like Lp norms could be used and with respect to the corresponding bin, and usually L1 norm is used to mostly for representing this histograms that has been found, but you can use any other norm. But there are others special measures other special distance functions or measures by which know you can describe the differences between two histograms or similarities between two histograms.

So, one of this measure is called Kullback Leiber divergence measure and it actually it tells you if I give you two probability density functions probability distributions, two different probability distribution how close or how different they are. So, the measure is defined in this way;

$$D_{KL}(P||Q) = -\sum_{x} P(x)\ln(\frac{P(x)}{Q(x)})$$

There is also another distance function which is called earth movers distance function and which I will elaborate a bit more.

(Refer Slide Time: 25:16)



So, in this case the idea is that you have two histograms P and Q and you consider that transforming one histogram P to Q by transferring masses from a bin to any other bin of Q. So, you are assuming that every bin represents every bin has certain amount that is a definition of histogram it is a frequency distribution. So, frequency is considered frequency of that particular quantity that itself each unit is called as a mass.

And if you are going to transfer a portion of that to another bin and of a Q and so, that the histogram P gets transformed into Q. So, this transfer has a cost and this cost could be defined as the product of transferred mass and distance between bin which mathematically I can express in this form that m and the distance between two bins.

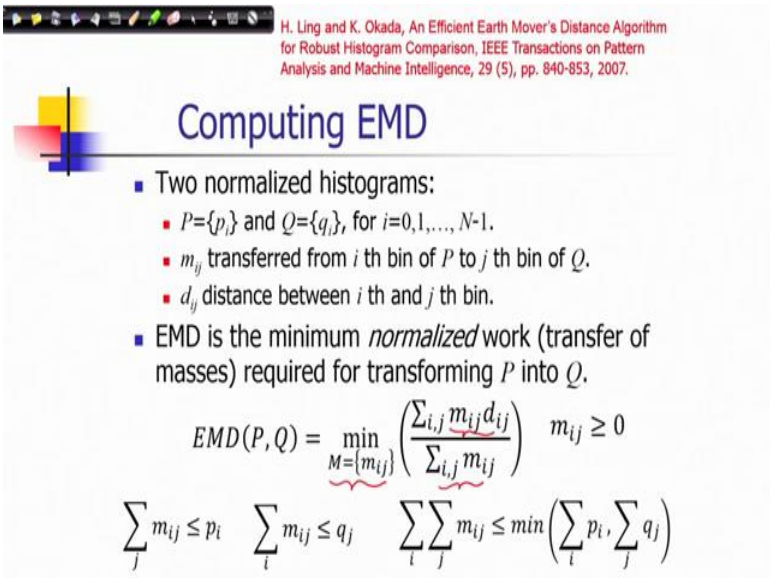
So, each bin say ith bin and jth bin their difference of the bin locations itself could be considered as a distance that absolute differences between bin locations. So, you take the product of these two that is what is the mass you are transferring from from ith bin to jth bin. So, for example,

$$P[i] = P[i] - m \text{ and } Q[j] = Q[j] + m$$

So, this is the transfer operation that would do and by doing this thing what you are trying to do trying to achieve is that transform histogram P into Q. So, we consider in this context the total mass of P and Q should be same.

So, after doing all these transfers you are expecting that no you can convert the distribution of masses in P in the form of P should we now in the form of Q. So, what is the minimum cost operation accumulated cost operation? So, particularly this accumulated cost again it could be it should be normalized with respect to the total transfer of mass that is a measure. So, which what is that minimum no cost and that will give you the distance and that is what is called earth movers distance, because no it relates like transferring mass is something like digging the earth from one part from one place and placing it to the that amount to the another bin.

(Refer Slide Time: 28:26)



So, something from that analogy the name has come and providing you a bit more mathematical formulation of this computational problem, that now you consider a two normalized histograms because no as I assume that their mass should be same. So, the safest way to start with is that normalize both the histograms. So, their some of areas should be equal to 1 that is the total mass they have and represent say ith bin of histogram P's small pi and ith bin of histogram Q's small q and now you consider the ranges then there are n number of bins and the mij denotes the mass transform from ith bin of p to jth bin of q and dij distance is expresses a distance between ith and jth bin there is a distance between locations.

So, the earth movers distance is the minimum normalized work required to transforming required for transforming p into q. So, which means you are trying to compute this one.

So, this is an optimization problem that you have to minimize this particular quantity as you can see this is the cost of transfer from ith bin to jth bin of a mass mij and this is a total mass that has been transformed that is the normalization part of the work.

$$EMD(P, Q) = min_{M=\{m_{ij}\}} \left(\frac{\sum_{i,j} m_{ij} d_{ij}}{\sum_{i,j} m_{ij}} \right), m_{ij} \geq 0$$

So, what is the set of transfers that would gives you the distribution Q from P and what is out of those possible sets which one will give you the minimum cost.

So, this is what is your computational problem and since now you are dealing with mass there are certain constraints you need to consider one other constraint is that, every mass should be positive or 0. So, if there is a transfer it has to be a positive transfer then the other one are your transfer of mass from the ith bin of p should should not exceed the content what it has in that bin; that means, that is a capacity that bin has to transfer masses from that to any jth bin of Q. So, so that is expressed mathematically in this form if I accumulate all masses from ith bin, it should not exceed the original mass of that bin in the histogram p.

Similarly then what you are transferring to the jth bin is that should not again exceed what is the content in that jth bin because that is what would like to transform pth P histogram to Q histogram. So, you have to ensure that that that it should go always less than equals it should not exceed that. And in overall the transform or of all these masses should be less than the total mass of this histograms one of this histograms. So, minimum of these two histograms. So, it is actually considering this framework does not really require that two histograms should have same mass, but in our context we considered that they should have same mass.

So, in this case it should be always less than equal to 1 because that is what we have assumed that total mass is 1 in the normalized histogram. So, this is a computational problem and I am not going to discuss this computational solution of this problem which is not within the scope of this particular course, but if you are interested, you can go through this particular paper where a typical solution has been presented and efficient solution this paper has been published in IEEE transactions on pattern analysis and machine intelligence in 2007 and you can find out an algorithm for that. So, with this let me stop this lecture at this point and we will continue this discussion of matching of feature descriptors as well as the other topic of model fitting in this series of lectures.

Thank you very much.

Keywords: Feature matching, nearest neighbor, earth mover's distance, matching criteria