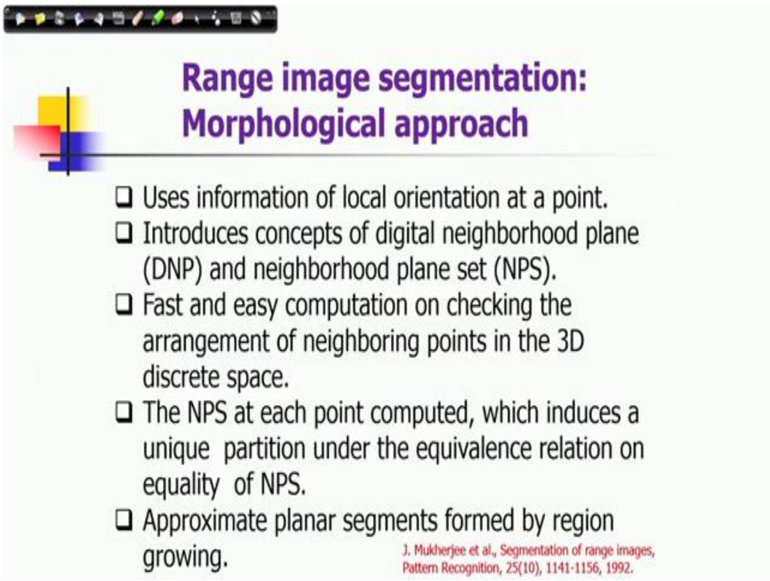
Computer Vision
Prof. Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 45
Range Image Processing – Part

We continue our discussion on segmentation of Range Images. In the previous lecture, we have seen how a greedy approach could perform the segmentation of planar faces in range images.

(Refer Slide Time: 00:20)

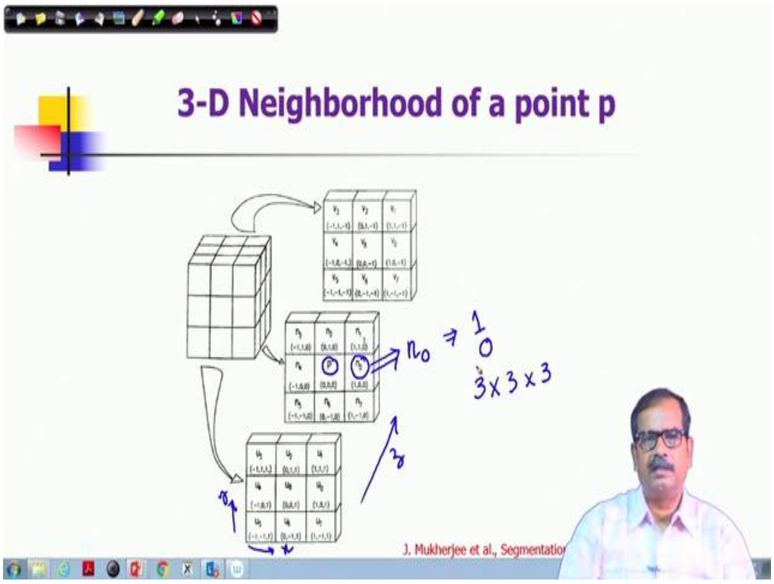


Today we will discuss another approach, this is a morphological analysis based approach where it uses information of local orientation at a point and we will find that this approach could be effectively used also to extract the planar segments in range images.

So, in this approach a concept of a digital neighborhood plane and neighborhood plane set those concepts are introduced. I will explain these concepts in this lecture. And the advantage of working with this approach is that, it is very fast and very easy computation on performing checks of local neighborhood arrangements and then taking a decision on the local orientation. And after that you can aggregate pixels of similar orientations and form the segments.

So, essentially what it does that it computes a set of neighborhood planes. which again I will be discussing and then this itself acts like a feature, this set itself acts like a feature and it induces an unique partition under the equivalence relation of equality of NPS and you can get the segment. And this planar segments are of course, approximate and they can be formed by vision growing, as I mentioned by considering the equality of the local features defined by neighborhood plane set or NPS.

(Refer Slide Time: 02:17)



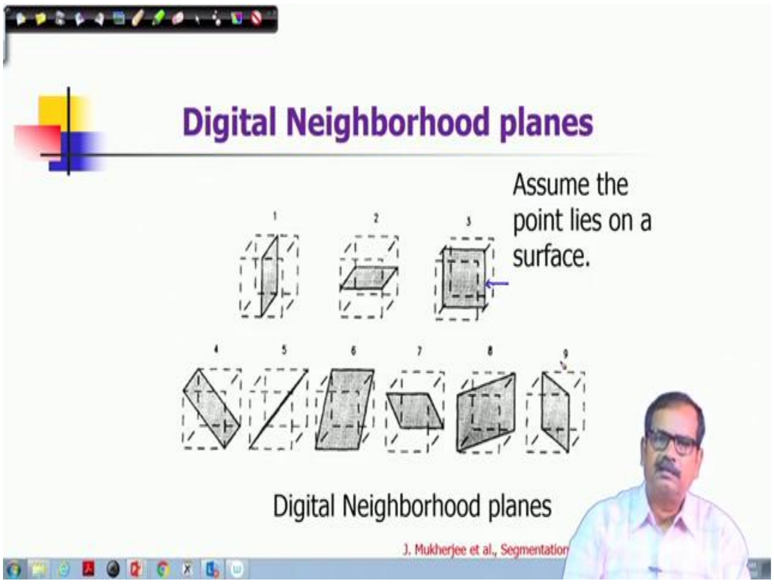
So, first let me explain how 3 dimensional neighborhood of a point is described. As you know that in a discrete space a 3 dimensional neighborhood can be described by extending the notion of 2 dimension itself. In discrete with phase space, so we will have a $3\times3\times3$ that is a, rectangular tessellations of 3 dimensional space and those points are shown here. So, in this case this is the point which is the central point and whose neighborhood are described. So, this is a point P and you can see that there are 3 cross sections if I move along the directions of x, y and z directions in our conventional notations.

So, we can consider say this is y, this is x, and this is z, this is a direction and with respect to this point this is a front plane, and this is the middle section and this is a back plane. And the variables are also identified by considering the positions of neighboring pixels. So, if I assume this is a origin, then with respect to that; for example, this voxel, so in 3 dimension we call the elements in the discrete grid as voxel and say this variable has been named as n_0 and the corresponding positions we can see this is a right neighbor of the point p in the middle section.

And similarly we have, so if I consider only a 2 dimensional cross section then this is defining a 8 neighbor configurations; but when it is a 3 dimension it is a 26 neighboring configurations and the nomenclature of the variables are also shown here. So, middle plane we have seen all the variables are named using the notations n and their subscript from 0 to 7. Similarly in the back plane, the variables are denoted as v and with subscript 0 to 7. Their variable means, if there is a point then the value of this variable would be 1 or 2 and if it is empty then the value is 0.

So, this is a representation of a point set in a discretized space you can say. And as I mentioned range image is nothing but a set of points in 3 dimension, they should lie on surface points; but in a 3 dimension you can describe them also as a 0.6. So, with respect to any point in the range image, you can have a 3 dimensional neighborhood in the corresponding space.

(Refer Slide Time: 05:30)



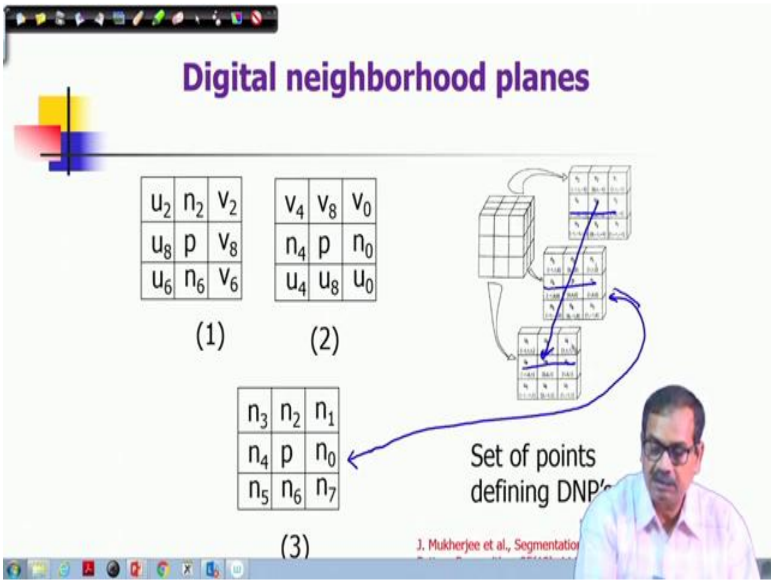
Now, let me define what is meant by digital neighborhood planes. Now, we assume that the point lies on a surface, then there could be neighboring points also should be lying on some of the neighboring planes. They are defined in these directions, in these configurations; mostly if it is a planar phase in the discrete orientations you are expecting those points should lie in one of those planes.

It is once again it is an extension of the corresponding directions, discretized directions of 2 dimensional space; where you have a 0 to 7 or 1 to 8 directions and used in chain codes in 2 dimensional images when we describe the sequence of points in a contour. But in 3 dimension you have surface points here, and there we have to look at the corresponding 3 dimensional configuration of neighboring points.

So, in this description, we have shown that what kind of configurations neighboring planes can have. So, if I have all the points in the middle plane, then this is one kind of configurations. There are certain indexes by which these planes are referred to here, for example, this is the index here is 3. So, there are nine such configurations, and all of them actually formed by as you can see that if I consider the this cubic face; then they are formed either by the principal planes in the parallel to the faces of the cubes or their diagonal planes which is connecting the corresponding diagonal edges and showing this planes.

So, the numbers here, you can see starting from 1 to 9. So, there are 9 such neighborhood plane which has been defined in this configuration. And in each neighborhood plane there would be once again 9 points including the point p which is a central point. Let me show you those points.

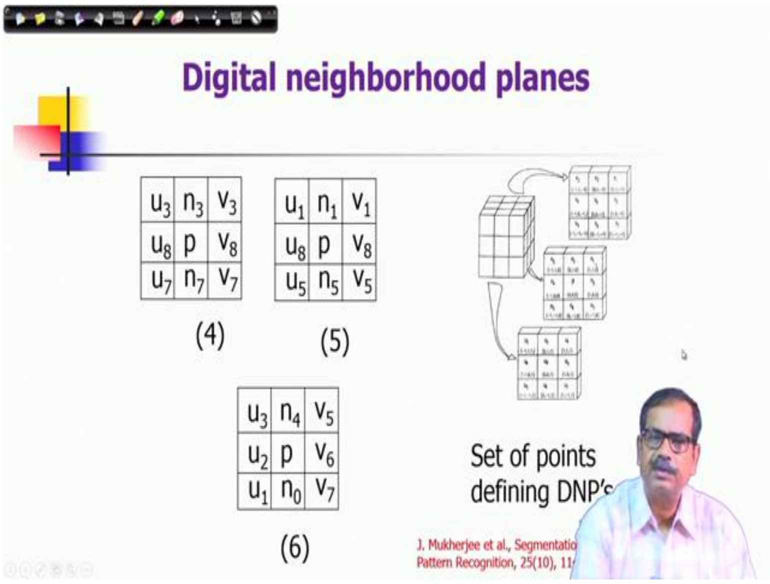
(Refer Slide Time: 07:50)



So, these are the set of points which have been know shown here, and here the corresponding planes, their indexes are shown by their indexes or identities are shown by the corresponding number 1, 2, 3 these are the principal planes. And if you check with the variables that we defined earlier for the digital map, for the neighborhood, $3 \times 3 \times 3$ neighborhood those variables are listed in these planes, those are corresponding variables.

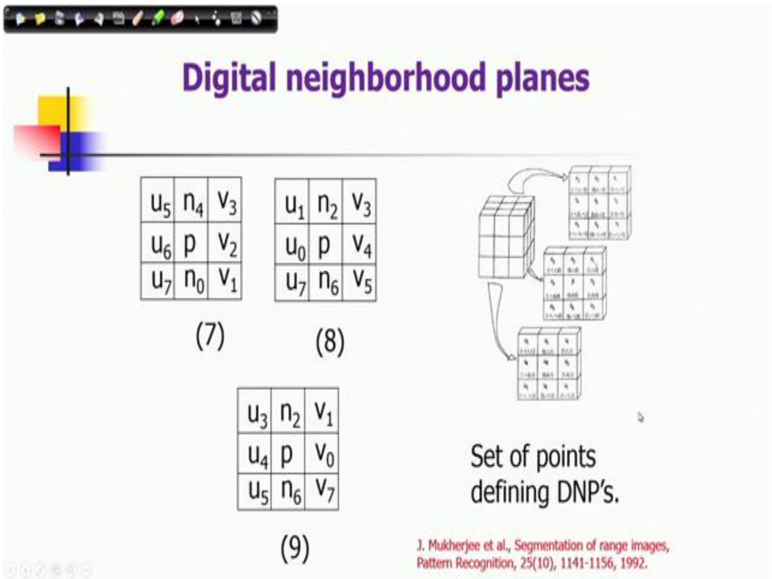
So, this was a middle plane, so the variable name and you can find out all the variables in this plane, they correspond to the corresponding variables of this third plane. And similarly if I consider all the columns of these cross sections along these directions that will give me the, it is I hope it is second plane this is v 2 v 8 v 6 and now this is the first plane. So, this is the first plane directions and if I use this, this is the second plane directions. So, in this way you can always form this point set by observing the neighborhood variables.

(Refer Slide Time: 09:22)



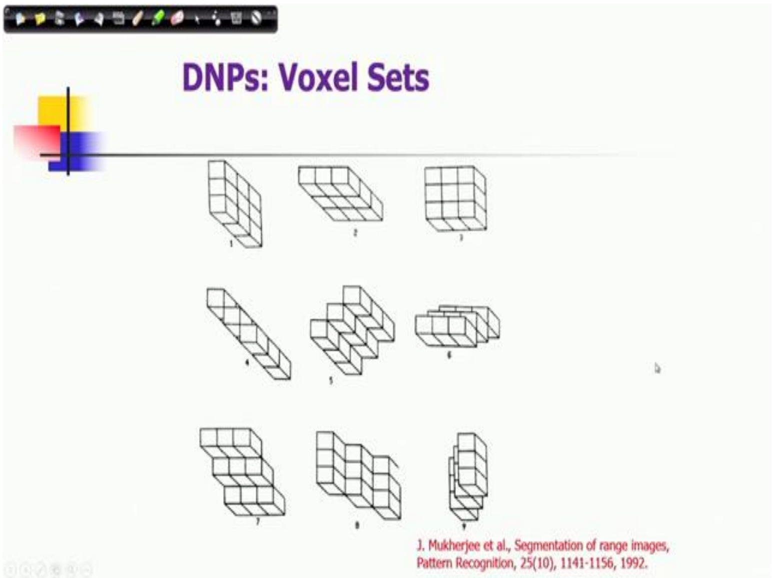
So, in the same way we can define all other diagonal planes also, I am not pointing out the corresponding configurations you can do it yourselves by observing the name of the variables and corresponding configurations of those diagonal planes.

(Refer Slide Time: 09:40)



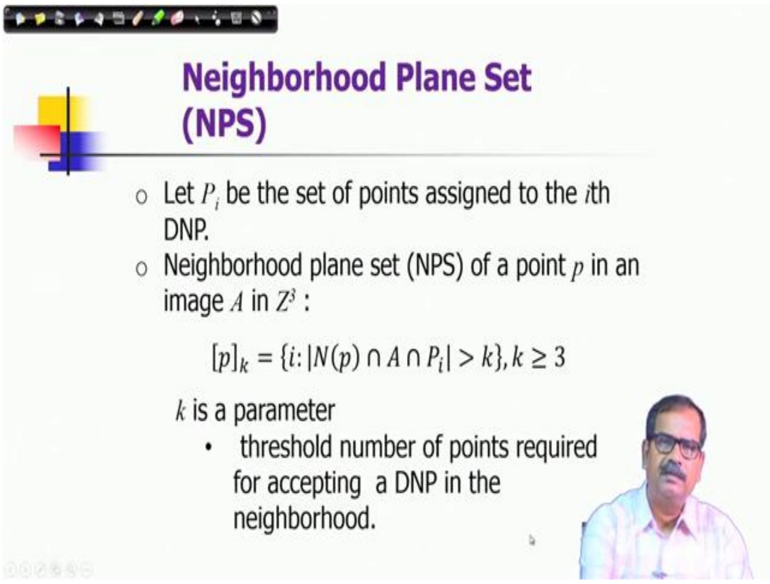
And this is the plane of 7th, 8th, 9th those planes are also described here.

(Refer Slide Time: 09:46)



So, just to have an idea how those planes looks in a discretized grid with voxels when those all the points are there in the plane. So, you can see the corresponding shapes of those planes in the digital grid, those are shown here with their numbers.

(Refer Slide Time: 10:05)



So, now let me define the neighborhood plane set or in acronym NPS. So, it is considered P_i, is a ith DNP; that means, the set of points assigned to the ith DNP. Now, we consider the P_i could be an element of neighborhood plane set of a 3 dimensional point P; if it is neighbors contains sufficient number of points in the or lying on that P_i itself.

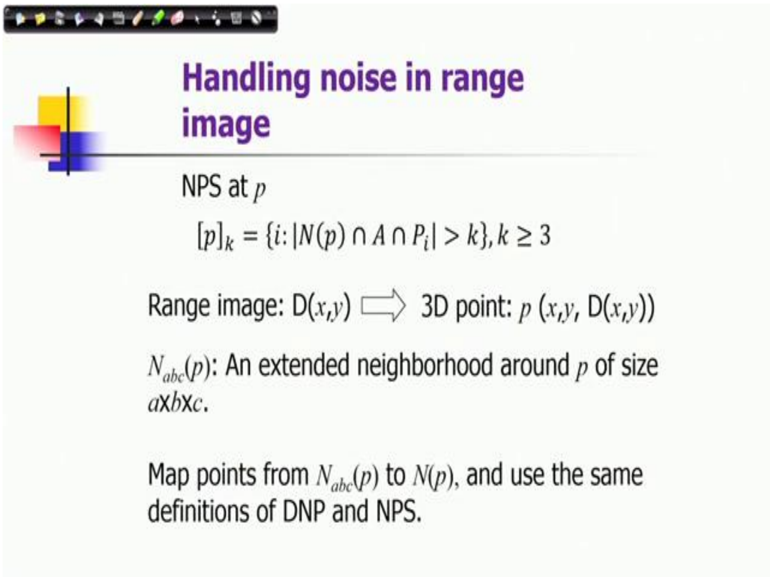
So, that that is the inclusion of P_i. So, in this way you can check for other planes also and set of all such planes which satisfy this property that the neighboring points of that point P, they lie sufficiently on those planes. So, mathematically we can define neighborhood planes in this way, it is the set of plane i.

So, i denotes the i-th data neighborhood plane, such that the neighborhood points of P which is denoted as a set $N(p)$ and the corresponding set of points in planes as defined that intersection, and A is just object point. So, only the points which are one and that is defining the object; when it is just making it more precise that we are considering neighboring points where there is a volume, I mean which are not empty which has only object points.

$$[p]_k = \{i : | N(p) \cap A \cap P_i | > k\}, k \geq 3$$

So, those points which are lying on a particular plane with a sufficient number that number is k. And usually we take the threshold k is greater than equals 3 two; that means, which means at least four points should lie including the point P. And at least yeah, at least three points should lie including the point P; but we can you can use any other threshold maybe 4 or 5 is sufficient in mostly used in our cases. So, k is a parameter as you can see and it is a threshold number of points required for accepting a digital neighborhood plane in the neighborhood.

(Refer Slide Time: 12:16)



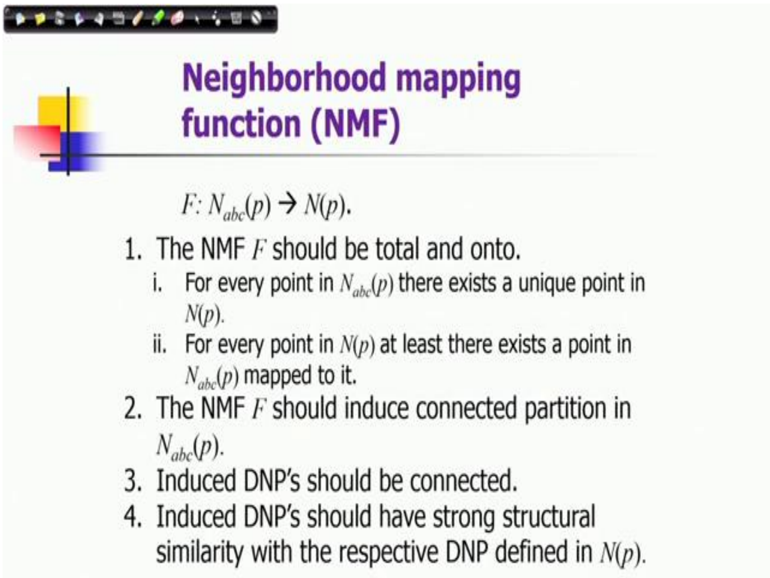
Now, the question is that, this definition what I know given, what I discussed this definition is meant for 3 dimensional grid and where you assume that is an idealized grid, there is no noise and then perfectly you can associate neighborhood planes with the point P. But the problem is that, now if we have noise, so even some of the points which are not lying exactly in your in the $3\times3\times3$ neighborhood or 26 neighborhood have a point P; still those points should have been should be considered because those are deviated due to noise, and they could be a possible candidate of forming DMP.

So, that is the case when we are trying to handle range image, because range image is an image which has taken from the real life scenario and there is a expected that there would be noise in those imaging systems. So, in a range image D(x,y) we can define a 3 dimensional point as (x, y, D(x, y)). And around its neighborhood, then to handle this kind of tolerance of deviations of points which may lie on a neighborhood plane. What we considered; we consider an extended neighborhood around p, and this extended neighborhood for example, a size $a \times b \times c$. So, minimally there should be 3. So, this all of them should be greater than 3. So, that is how the extended neighborhood, and for maintaining symmetry usually they are all odd numbers.

So, what we can do here, that is to this is a trick what has been used in this technique; that this tolerance has been accounted for by the fact that we map a set of points in this extended neighborhood to a point of the neighborhood $3 \times 3 \times 3$ neighborhood to those variables. So, a particular variable would be true in $3 \times 3 \times 3$ neighborhood, if one of the points in that set which has been mapped to it, is true.

So, that is why, that is how we can make a simple adaptation of this concept of DNP in this case. And then all the definitions of digital neighborhood plane, and neighborhood plane set they remain the same; only this mapping will take care of those tolerances, giving tolerances to those deviated points or to those noisy points.

(Refer Slide Time: 15:04)



So, before giving an example let me discuss that what are the properties should be maintained when we define such an mapping functions. There could be various possibilities for mapping a set of points to a neighborhood point of N(p); but we should consider those mappings which maintain certain consistency, and which should be helpful in solving our problem.

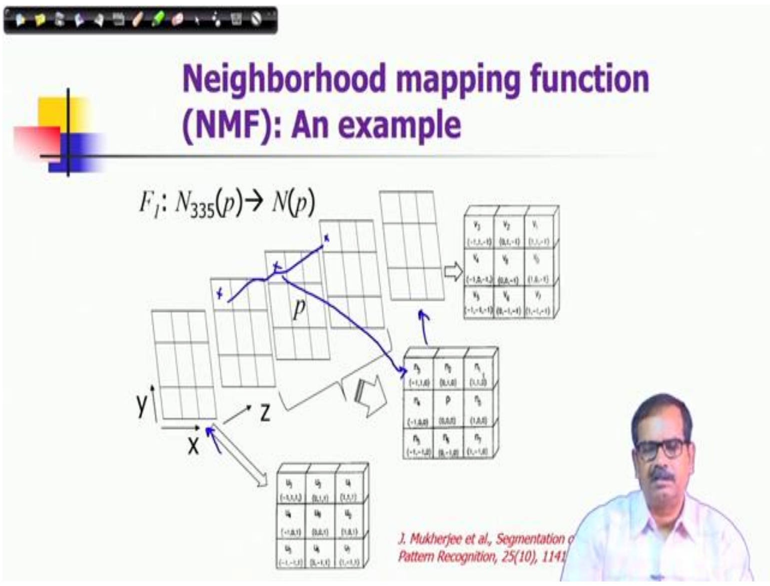
So, we are considering a mapping functions in this case we are naming it as a neighborhood mapping functions, that is a mapping from extended neighborhood of $a \times b \times c$ to a neighborhood of $3 \times 3 \times 3$. So, first thing that the function should be total and onto; which means that, for every point in the extended neighborhood there exists a unique point in N(p).

So, for every point you have to have a definition, you have to have a mapping that is the total property; and for every point the in the $3 \times 3 \times 3$ neighborhood there exist at least one point in the extended neighborhood which has been mapped to it. So, that is a property of onto function.

And then this function should be should induce connected partitions in $N_{abc}(p)$; that means, when we are making these mappings, so the points of the extended neighborhood which are mapped to the same point in the $3 \times 3 \times 3$ neighborhood they should be connected that is the connected partition in that case.

And again as a result the digital neighborhood planes what it will look in the extended neighborhood of the point p, they should be also connected. So, the induced digital neighborhood planes also should be connected. And moreover for the good quality of segmentations it should have strong structural similarity, with the respective DNPs defined in N(p).

(Refer Slide Time: 17:19)



So, one such possible neighborhood mapping function is described here. So, in this case you can see that we are extending the neighborhood by $3 \times 3 \times 5$; which means, there is an extension along the z directions in both the front and backward know directions and there is an additional 3×3 plane in front of or in the in behind the point p.

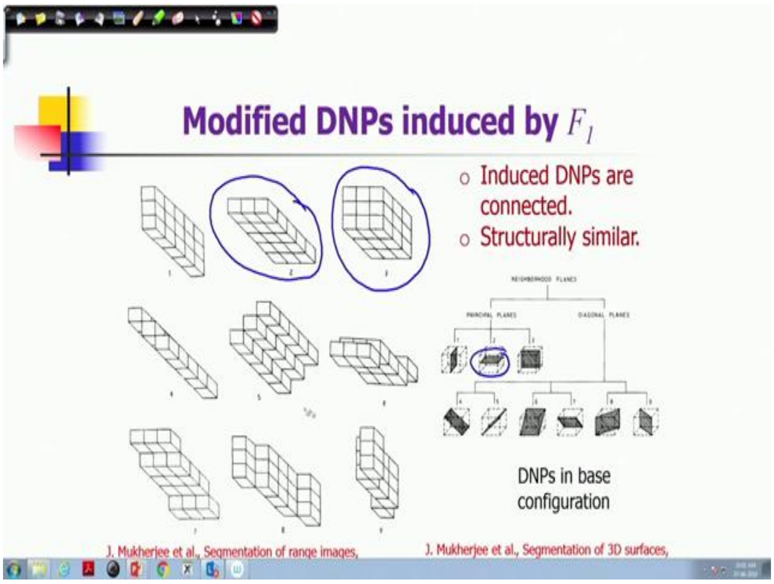
So, you can see here, see this is the additional plane which has been additional 3×3 cross sections which has been added to the neighborhood definition of p. So, now if this is the extended neighborhood, then let us see how the mapping is carried out to maintain the properties which I mentioned.

So, first thing as you can see that we have mapped all these points to the central cross section of the $3 \times 3 \times 3$ neighborhood; which means, if I consider say these points, all these points together they are mapped to n_3, together means no they are all of them are mapped to n_3. So, if any one of them is true, then the variable n_3 becomes true. So, it is all logic by which this variable is related to with those points.

So, in this way the no extension has been done, and in the same way the other know very other neighboring variables of points they are also mapped. So, the cross section which is behind by two unit, it is mapped to the actually all the variables of backplane of the original neighborhood definitions of $3 \times 3 \times 3$. And cross sections which is in front of two units, again it is mapped to the all the variables, corresponding variables are mapped to the variables in the front cross section of $3 \times 3 \times 3$ unit.

So, in this way as you can see we can extend the neighborhood definitions we can map them to the variables. And finally, we are working with only say 26 variables with p and with these kind of definitions. So, then the rest of the definitions of digital neighborhood plane, and neighborhood plane set it remains same.

(Refer Slide Time: 19:54)

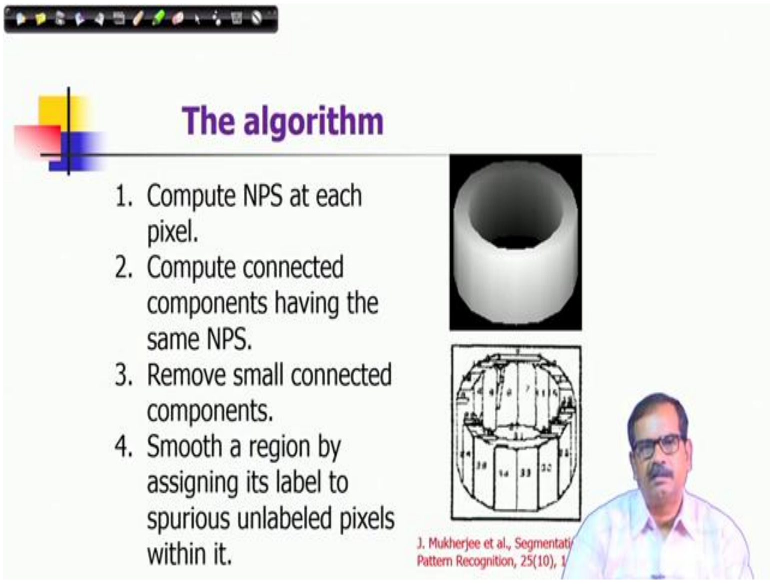


So, just to look at the results of the digital neighborhood planes due to these operations or the configurations what you can have. So, these are the digital neighborhood planes which have been induced by this function F. You may note that the central plane or the plane 3 which I have shown earlier, actually it is accepting any points in this volume.

So, all these points in that volume it corresponds to plane 3 because of that tolerate, because we have given this tolerance. And in some of these planes also this more number of points are there, and there is a particular rule by which of course, you have to check the planarity of this test of these set of points.

So, first thing induced DNPs are connected. So, you have seen that all the digital neighborhood planes here they are connected, and they are structurally similar. Let me show you the once again the configurations of the neighborhood planes, and you can see that the corresponding shapes are similar; say this is plane 2 and this is plane 2 which has been defined, and in this way we have to understand we have we can find out that similarity of shapes.

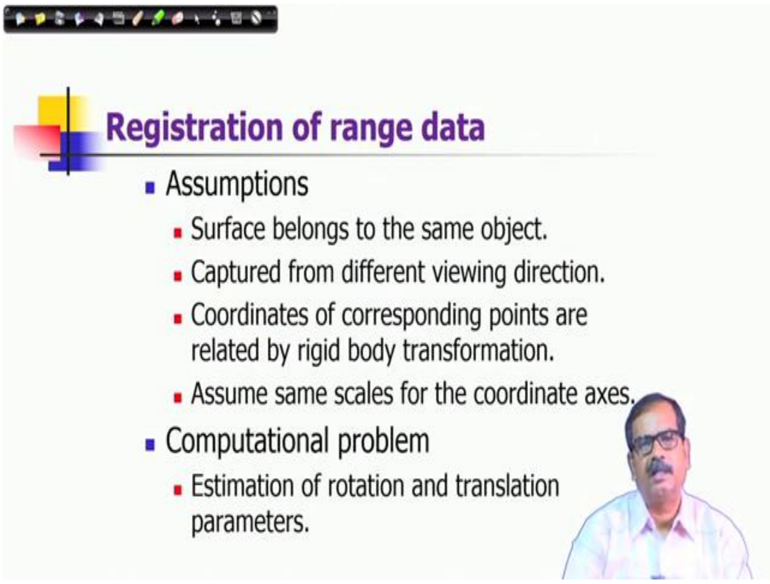
(Refer Slide Time: 21:24)



So, it satisfies the properties what we wanted to have in this neighborhood mapping function. And then by applying the definitions of neighborhood plane set we can from the segment. So, the algorithm goes like this, that you can compute the neighborhood plane set at each pixel, then you compute connected components having the same neighborhood plane set; and remove small connected components from them, then smooth a region by assigning its level to spurious unlabeled pixels within it.

One example of range image segmentation has been shown here; the upper one is a display of a range image, So, in this case the higher the brightness value nearer the pixels and the darker points are far behind. So, that is how this display has been made and the corresponding, the segmentation results are also shown here.

(Refer Slide Time: 22:30)

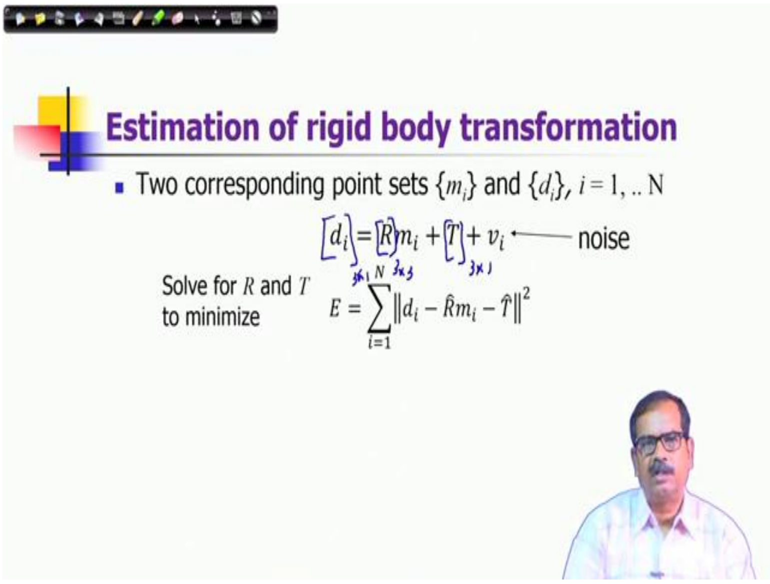


So, now let me discuss another kind of processing with range data and this is registration of range data. So, we know the registration problem that, if I give you two images and if they are related by certain transformation. So, we need to compute the transformation, so that you can obtain the other image by applying transformations, obtain an image from the other image by applying this transformation.

So, in the range data also we may consider that this data has been captured from different views. So, they are related by the corresponding coordinate transformations between these two views and that is a kind of rigid body transformation. So, the assumption said that, first thing that surface belongs to the same object; that means, you are viewing the same points.

And then captured from different viewing direction, coordinates of corresponding points are related by rigid body transformation and we assume in the same scales for the coordinate axis. So, all other things are similar, only thing is that in the coordinate transformation we have translation and rotation of the corresponding coordinate transformation. So, the computational problem is that estimation of those rotation and translation parameters.

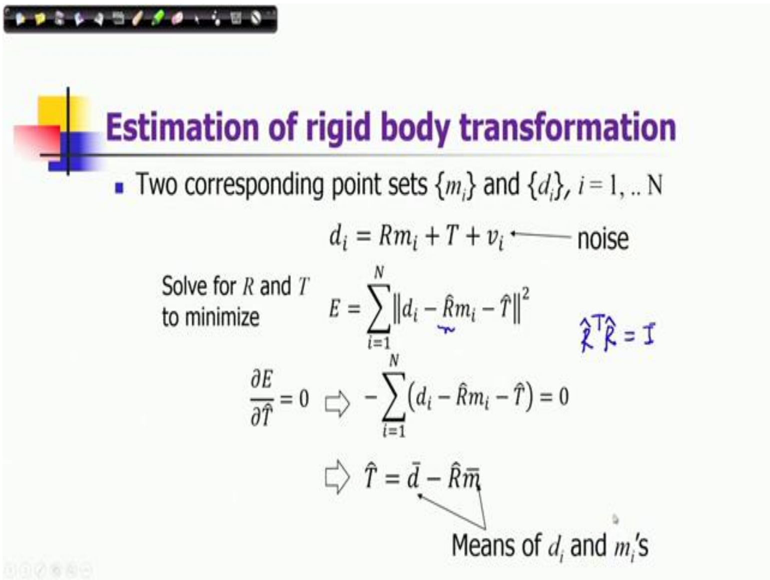
(Refer Slide Time: 24:00)



So, we are considering this problem of computing parameters of rigid body transformation. So, let us consider there are two corresponding point sets $\{m_i\}$ and $\{d_i\}$. So, $\{m_i\}$ corresponds to point sets of one view, and $\{d_i\}$ corresponds to point set of other view; and we would like to get a transformation from $\{m_i\}$ to $\{d_i\}$. And as I mentioned that, it is a rigid body transformation and related with rotation and translations, so we can express them in these relationships. So, all our 3 dimensional points, 3 dimensional vectors and they are expressed in non-homogeneous coordinate system. So, it is written as $d_i = Rm_i + T + v_i$, so just to explain.

So, if v_i is the noise here, and R is a 3×3 rotation matrix and T is a 3×1 translation matrix; and we know that points are described by 3×1 vectors d_i and m_i. So, this relationships are established. Now, our objective is to compute these parameters are R and T in presence of the noise. So, we would like to minimize the error of the model fit, model fitting and this is how the error of model fitting has been expressed here and this is the same sum of square error what we discussed earlier also as you can see. And the estimated rotation matrix and translation matrix they are all denoted here by theover it tilde in my slides. $E = \sum_{i=1}^{N} \left\| d_i - \hat{R}m_i - \hat{T} \right\|^2$

(Refer Slide Time: 26:06)

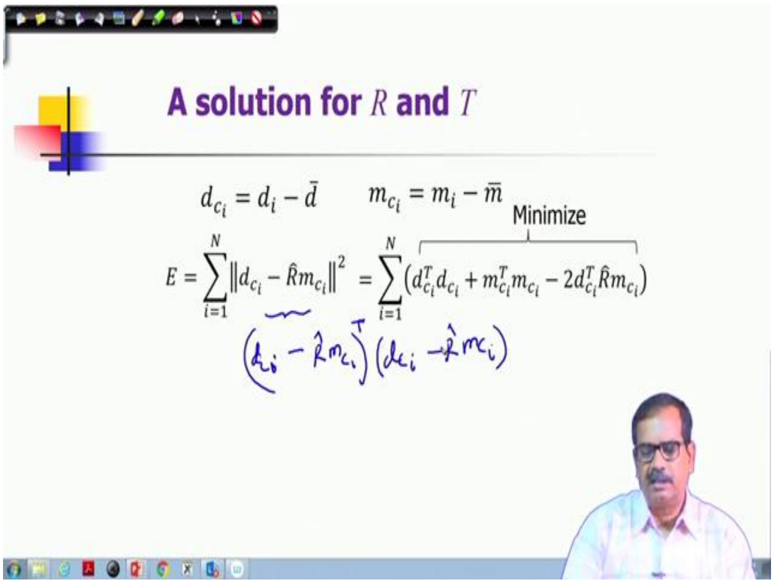


And one thing which it is, you should know that it is a constant optimization in the sense there is a property of rotation matrix and that property is that it is an ortho-normal matrix; which means that, this matrix should satisfy this condition, this $(\hat{R}^T \hat{R})$ should be equal to I or R^T should be equal to R^{-1} e. So, subject to that we have to solve this minimization. So, it is not a simple least squared error estimation method or simple minimization optimization method, it is a constraint optimization method that we need to consider here.

So, we apply in this strategy of a first removing the translation part. So, what we do? We have taken partial derivatives with respect to T and then we get these equations, if I take the derivatives we can find that translation does not have only minus 1 as the coefficients, and you can apply once again matrix algebra to reduce the derivatives into this form. So, from there we can get the estimate of translation in terms of the average of d_i is and average of m_i is given R.

So, still we cannot compute T, but we know the relation; if we can get the estimator of rotation matrix R then we can get translation matrix translation 3×1 translation parameters, because we can always compute average of d and average m. So, that is what it is means of d_i and m_i 's.

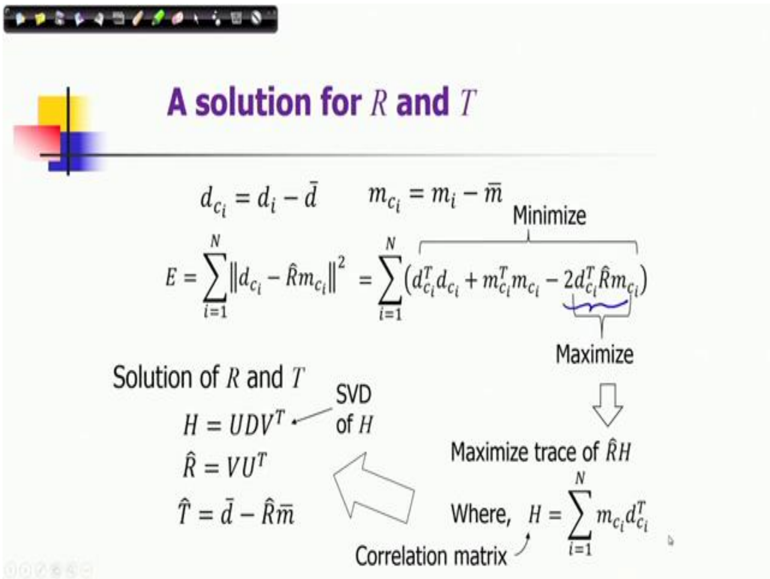
(Refer Slide Time: 27:34)



Let us proceed with that. So, let us make a coordinate transformation, so that we can, we try to remove the translation part from this expression. So, we perform this coordinate transformation and we can see that, with this coordinate transformation, the translation part could be removed because only rotation, with the relationships with rotation and the translation has been established in terms of beam in the estimation. So, the translation can be that parameters could be removed.

So, now the minimization problem becomes that, you have to compute R with that constraint of $R^T R$ should be equal to identity matrix, so that now this function gets minimized. Now, there is a particular type of solutions, so if I expand it is in this form, and you have to minimize this particular know part; this is once again it is the matrix algebra just to show you that this expression can be written as $(d_{c_i} - \hat{R}m_{c_i})^T (d_{c_i} - \hat{R}m_{c_i})$. So, if I perform the corresponding multiplications like simple and linear algebra like simple algebra, but it is all applicable with matrix algebra also because you can check with a dimensional matching and because of linear operations of matrix multiplications and additions. So, you will find that finally, it reduces to the expression what has been shown here.

(Refer Slide Time: 29:24)

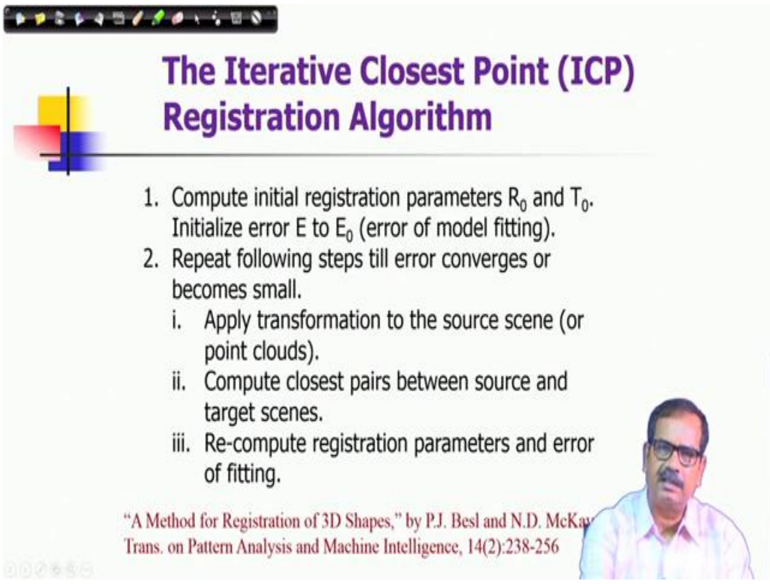


So, you can see here actually if you are going to minimize this thing which means, I need to maximize this particular part. So, what we are doing, we need to maximize this part($2d_{c_i}^T \hat{R} m_{c_i}$); and then the there is a solutions for that we have to maximize trace of $\hat{R}H$ I am not going to know discuss the how we have derived particular these relationships; but this is the solution, so when with the constraint optimization what I have mentioned here. So, the definition of H is that, H is given in this form $H = \sum_{i=1}^{N} m_{c_i} d_{c_i}^T$. So, it is basically covariance between the corresponding coordinates translated by their means.

So, we call it also a correlation matrix. And one of the solutions for maximizing this is, this is a solution that H can be decomposed using singular value decomposition $H = UDV^T$ and then R transpose R sorry $\hat{R} = VU^T$, you can check this is an orthonormal matrix.

So, this is a solutions and then T is obtained as this. But the fact is that, this will give you a particular solutions for a same point, but there would be error fit as we have seen, but we can perform iterative fitting of these points by performing we can refine these solutions, because there could be outliers in the corresponding points and we can remove those outliersby an iterative process.

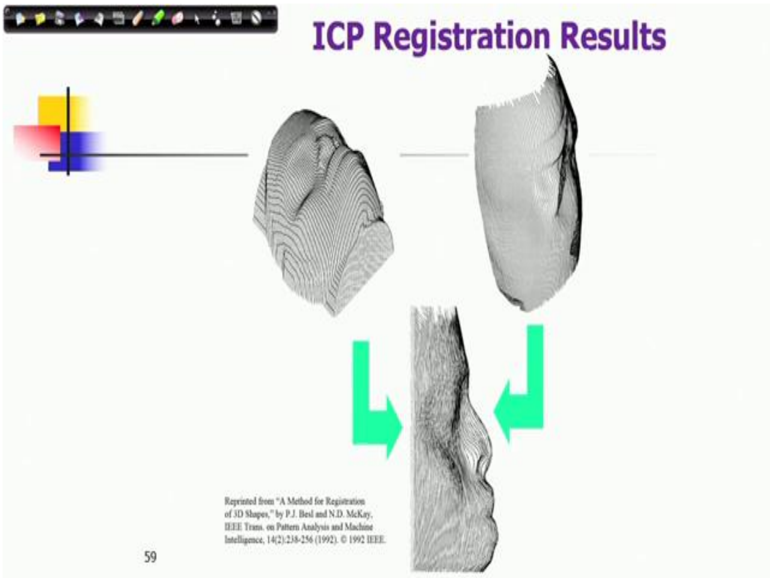
(Refer Slide Time: 31:03)



So, there is a technique which is called iterative closest point registration algorithm, in this technique it has considered to removal of those outliers in this way. So, what it does first? It computes initial registration parameters, like denoted as R_0 and T_0; and then we perform the steps iteratively in this way that, apply transformation to the sourcing or point clouds, and compute the closest pairs between source and target.

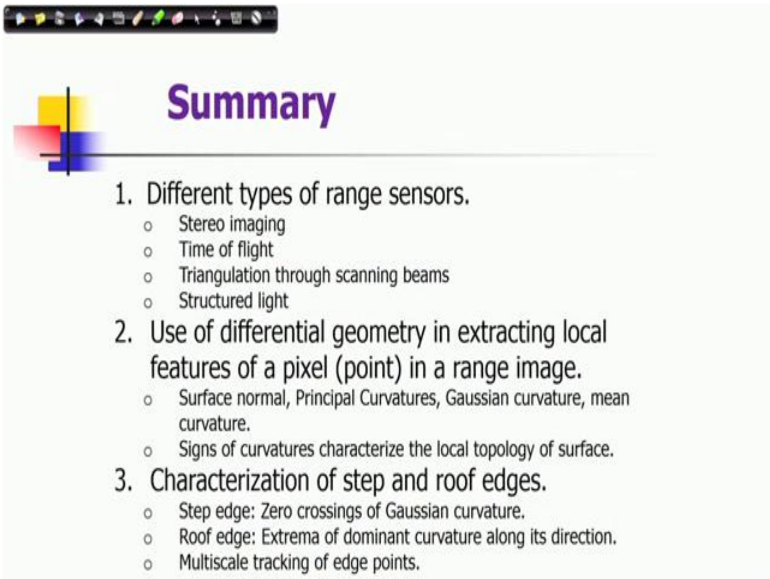
After applying transformations we can find out the closest pairs; that means, for every point in the source point what is it is nearest neighbor and every point in the trans-parity what is the nearest neighbor. If they correspond to each other then we take those points, and in that way we can get a new data set and use those data sets. So, we can again re-compute registration parameters by applying the same technique what I discussed; and go on doing these things till we get a good registration, till it converges.

(Refer Slide Time: 32:02)



So, this is one example of a registration result and this has been taken from a paper by Besl and McKay which has been published in 1992. So, this result I mention. So far these are the few techniques of processing of range images we have discussed.

(Refer Slide Time: 32:27)

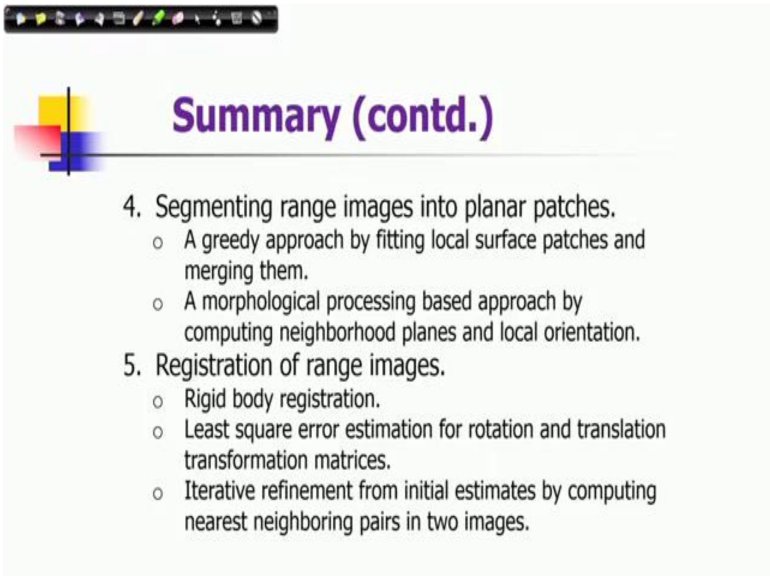


So, let me conclude this topic with the summary of different things what we discussed under this topic of range image analysis; one is that we have discussed about different types of range sensors. So, we considered stereo imaging.

Then time of flight based sensors, triangulation through scanning beams, then structured light. Then, we have considered use of differential geometry in extracting local features of a pixel or a point in a range image. These are the features which we could extract like, surface normal, principal curvature, Gaussian curvature, mean curvature; signs of curvature characterize the local topology of the surface.

We discussed also about characterization of step and roof edges. And step edge is could be detected by detecting zero crossings of Gaussian curvatures and roof edge could be detected by considering the Extrema of dominant curvature along its direction. And further if the multi scale tracking of edge points can refine the results.

(Refer Slide Time: 33:42)



We have also discussed about segmentation of range images into planar patches. We discussed a greedy algorithm by fitting local surface patches and merging them; and then a morphological processing based approach by computing neighborhood planes and local orientation.

And the last topic of this particular, last topic of this range image analysis it is registration of range images. So, we discussed how rigid body registration parameters could be computed by a technique, there you can use least square error estimation for rotation and translation transformation matrices, it is a constrained optimization problem. And then we can perform iterative refinement from initial estimates by computing nearest neighboring pairs in two images. So, these are the things we have covered in this particular topic.

Thank you very much for your listening.