Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-83

Okay, and these are the tasks in the mid-level vision that we had actually envisioned. So optical flow rate that I talked about already, image segmentation, we want to talk about tracking right when it comes to, when it comes to the high-level vision at that point of time I will talk about tracking. Okay, and image retrieval right we are not doing. Okay, I think you have got enough background to be able to read it on your own. So this image segmentation right, so what it means is you want to be able to identify groups of, groups of points right in a sense. So when, when you say something is a cluster right, a cluster simply means that there is something which is, which is sort of common right, you know, among those points.

That is when you say, that is when you say right that there is a group, that there is a cluster and so on. So basically segmentation right you can also look upon it as some kind of a clustering task right where you have to say right which of these, which of these things right would you know would kind of fit into one basket, which of these things can be put into another basket and so on. And you know it can happen at various levels you know it can happen at the intensity level it can happen at the color level, it can happen at some you know in some other feature space you know you may want to take that image convert into some other feature space and maybe there you want to do, want to kind of do the clustering. So in what dimension right you do it, in what feature space you do it right there is, there is nothing standard about that okay.

So when you go from one application to another right different applications require right different ways of doing things. For example right, in this case, in this case right to say that you know I will go by, I will go by intensity and segment right it does not even make sense or because, because if you look at, look at this particular this one tiger and you would want to kind of segment a tiger as a whole right that is because there is a texture on it which is kind of, which is kind of right uniform right and that texture is what, is what singles it out whereas the background is like you know another kind of a texture and therefore it really segment and then maybe the, maybe the ground and then something at the back I do not know whether that is water or something. So, so that way right you ideally want kind of four segments or something and not like, they are not like every black thing out there is one, one segment you know every orange thing there is one segment right. So, again at what level you want to think about the segmentation problem right and how you want to pose it a lot depends upon what kind of feature space right when you know you want to get a go into and actually do it and what feature space you decide depends upon what, what is your own goal right, what goal you have in your mind and so, so the idea is to kind of assign labels right finally the problem is what you want to be able to assign labels. For example right in this case you have got like four labels right you want to be able to tell where, what class each pixel right is from in the sense that you know whether it is grass, whether it is water, whether it is tiger, whether it is a ground right that is what you want to do.

So, basically it is a task of assigning labels or for example it could also be just a, just a foreground background kind of thing where you do not have multiple labels you have only two labels and you want to do a background foreground. This is often done for example you would have seen that you know people sort of change the background against which they are sitting right. So, for example I might just want to make this a classy background when I am sitting. So, if you want to do that then you have to segment me out, you have to segment the background out then you have to impose another background that is called alpha matting. So, you have to mat another background on me and then it should look realistic in fact Google meet and all they do this right.

So, that is why you have to, they have to segment you out right because they have to know that that is you and then you know everything else is the background and change the background and then as you move right they have to of course keep track of where all you are going right but that is what is done okay. So, again right I mean that is also a segmentation task because you have to know you have to know who is the person right. So, that is one that is the foreground and then what is the background is another is another segment right and you want to be able to fuse them. An application right like this you know where let us say right you want to remove this boy from that background right and then maybe you know if you do alpha matting you can put something right you know to the back of him but then right that is not an easy problem I mean especially when there is hair and all right I mean you can easily show it off that you are that you are putting some background you know which is not the original because the hair will carry something from the background. So, alpha matting is actually you know a different problem in itself where you have to you know we have to look at transparency and all that.

I am just saying right one of the first steps that you need that is to be able to segment the person and then maybe right you can even think of a better classification right in the sense that if you want to build a network maybe the network will get less confused you just show the object of interest rather than show a lot of things around it right. So, here for example if you just you know etch out the that flower right then maybe it is easier for some other thing to sort of tell what it is because when there is a background it could actually it could be it could be a little right it could it could probably think that there is something else going on it is a so there is that kind of confusion right which can occur which you can probably reduce right that is what we mean by a better classification okay. And then there is a lot of theory right there is called you know gestalt kind of you know theory which actually you know which is actually which is which is a psychophysical theory actually makes a lot of sense in the sense that right when you whenever right when do you kind of group things together right. But unfortunately right even though this is very rich and all but then it is very hard to translate these into algorithms the one thing that uses gestalt is this what is this what is called a tensor voting right tensor voting is one mathematical framework that uses gestalt which has been reasonably successful right. So I would not I would not say I would not totally agree with the last statement whether very difficult to translate because tensor voting actually does that in a you know in a nice way but there are not kind of you know too many algorithms like that that actually use this principle of proximity, similarity, continuity whatever right and all of that okay.

So okay this is just put there just to just to say that when you segment right what kind of factors you take into account as a human being right when you segment them and right you do not like sharpness you do not like sharp transitions you like things to right within a group you would not want things to be homogeneous all those things to look similar you might even assume a continuity when it is not there right your brain sort of you know interpolates extrapolates things that that is what the segment for example right if I had a circle but then if I just gave you some dots right your brain will automatically think that okay there ought to have been a circle there right. So that is the kind of filling in we do right in order to do some segmentation and so on. So this is a gestalt theory but of course you know I know that is I mean some algorithms use it but not let us say everything okay that is now let us come to let us come to know this one a clustering right I mean how do you how do you kind of do this. So so right there is one thing that is called a k-means okay k-means method okay there is something called the k-means, k-means is this one a clustering method okay I will start with this and then and then I will kind of say go on to go on to more sophisticated versions of this. So this is like unsupervised right so what it means is that I mean you have you have a you have a set of examples let us say X1, X2 up to Xn right and these means actually mean mean centroids okay this is not like our English mean right this is that statistical mean so what you are saying is there are there are actually k centroids right it is like you know you can think of think of each one having having a group of data points around it and there is a kind of centroid and you have got to see k centroids and and and here is your set of examples and you want to know right which one which one ought to be sitting where should X1 be here or there should X2 be here or there or there and where should basically read each each each of these points be.

Now if if you write I mean you know if somebody already told you that let us say write I mean you know I mean in the sense that if you knew the group assignment and the whole thing is easy but then this called unsupervised because we just have these you see data points we do not we do not know from where they came right we do not know how they came okay we just know that right we have this X1 to Xn and we want to kind of do some kind of a grouping a clustering right and and that is why we call it unsupervised because because we do not have we do not have any kind any form of a guidance right and and the way right this works is okay. Now right one of the one of the things is that I mean you know there are certain shortcomings of this one of the one of the first very first shortcoming is that you need to know k in the sense that that knowledge you have to know for example you have to know that right this is probably coming from four clusters five clusters for example right in that image that I showed you red tiger land and all you have to sort of say that well I have four groups there right so so some such thing you should have in your head okay. Of course you know there are there are see versions of k-means that can there is something called ISO data and all which does handle it to some extent but then the but then a basic k-means right will not okay so what I am talking about is simply a basic k-means not the not the not the higher level extensions of that and so on okay. Now there is even something called hierarchical sort of you know clustering through k-means and all we will not go into that we will just look at look at the look at the classical case okay. Now what it does is actually right simple right in the sense that the kind of the the key let me let me see that if there is something that I okay unsupervised and so on okay and and the way it works is that works is right or see for example I mean you know if you if you take x1 right then then what it will do is you know in order to make make in order to assign right x1 to some sort of you know a cluster then it is like this red-ammoned if you take if you take xi right and if you want to say that it is no it belongs to a cluster number l right where where a rate of you know l is l is right anywhere anywhere in this is you want to k then or or let us say right let us just call call this right k star right let us say that k star is the optimal then you see k star is something like this k star will be like argmax over over all of all of k xi minus let us say mu k square I mean if I am if I am looking at a scalar case if it is a vector case it is a norm okay.

Now so so it is like this right so so what you are saying is if I had the centroids where this mu k are the means okay it is the kind of kth mean is mu k so I have like mu 1 mu 2 mu 3 whatever right up to mu k and I am going to compare xi with with each one of those means which which each you know with each of the centroid and and whichever it comes closest to I will say right I mean xi xi should go there okay. So so you can so you can sort of write inherently inherently imagine that probably right what you what you will kind of look at are you know spherical spherical groups right because I mean right something is something is equally away right I mean wherever it is right we would sort of you know we will kind of see welcome all of them right into that group. So these are some shortcomings right which you can already see for example right I mean no no if your if your data was right elliptical and so on right you could actually struggle right because because it is trying to trying to just look at look at the look at the Euclidean distance okay that is the basic k means okay and and that is why I read there are actually versions of this and the other thing right that you also see is that there is a hard assignment. With the hard assignment in the in the sense that you simply right allocated there in the sense that when xi xi comes close to some let us say mu k I will say that xi xi should go go right into you know into kth kth group but then it could also be that I mean you can have you can have groups of data points such that at the boundaries right data points seem to seem to I mean so if you should draw a circle it will look like well it could set in either right such things can happen but now in k means if you look at it is a hard assignment it will say only here right it is either either here or here or there right but then it cannot be like here as well as there right. So it is not it is not kind of you know a probabilistic thing I mean we cannot say that you know with so much certainty it is probably in that cluster but then with a with a kind of a lesser probability probably in the other cluster that is that is probably a better way to do things right but then k means right would not would not do it would not do any of that it is like a hard assignment okay.

So what you are doing is so when you are doing this right this is this is a hard assignment okay so this I will just write that down. So what does xi belong to k star? Oh xi belongs to k star which is actually a cluster kth k starth cluster I mean in the sense that k star will be will be let us say one of these one of these k's right at which xi comes closest to mu k that is that k star. So it will be it will be it will be one of these 1 to k clusters. Oh sorry argmin no no not max yeah argmin. So yeah so basically right k star is that is that is that whatever is that particular this one the question what did you say yeah argmin right.

So is that is so it is that mu k for which xi comes closest. So this is a hard assignment okay which is okay right which is which is way it works okay so here right I should have written x2 right up to some xn, xn is a data samples and the algorithm is actually fairly simple okay it is actually it is actually an iterative kind of you know a procedure which it has to be because you do not you do not know group assignment right so it is like you know a chicken and egg problem right. So it is like saying that I mean it is like saying that if I if I actually knew the group assignment I could compute the mean but then to get a compute the mean I need I need the assignment right I do not have either. So it is an see it is an you know iterative process. So algorithm is like this obtain k initial centers okay you have to start somewhere k initial centers or centroids right as you might want to call it randomly and again there is a host of things that have been that have been said about how to how to in fact choose the initial centers okay we will just we will just right now say randomly okay.

Obtain k initial centers randomly from the from the see this one a data set okay what this means is that so so right I mean so basically right I mean if you had well not probably like this so if you so if you had some points let us say if I can draw with you know a different color and so I have some points here and let us say right I have something here okay it need not be overlapping but then okay it could be somewhere here. So so right so out of these set of points this is randomly picked okay if I know k I will pick actually k initial centers I have to know k okay as you can already see that right it is going to be sensitive to how you choose the initial centers right I mean so suppose for example I choose all of them right within the first one then that is not going to be a going to be a good choice okay but there is a way around this let us say initially right which is which is the reason why you want to just kind of do it randomly so that the chances that you would end up you know picking all of them from the same sort of a cluster and so on it is going to be minimum. So randomly randomly okay obtain k from the from the data set then assign assign each data sample to the nearest centroid okay which is what this that is a k star we need to argue min in that one then to find the find the sum square error okay and and this you do for do for every sample right you walk through every x1 to xn and this is assigned then and then when you assign right whatever wherever it goes whichever whichever has the minimum sort of xi minus mu k square right that will be still an error right it need not be 0 okay so you can have sum up sum up all those errors I mean after you have done the optimum assignment right optimum in that step so I know that x1 is going to let us say fourth cluster so I find out x1 minus let us say mu 4 square right store it with me then x2 goes to let us say mu 2 after I check right whether it is close to mu 1 mu 2 mu 3 and so on suppose I realize that it goes closest to mu 2 then I will store x1 minus sorry x2 minus mu 2 square I will store that right so find the find the sum square error for all the samples for all the samples then 3 recompute so it is like saying right so it is like saying that you know if I started arbitrarily somewhere right let us say it is a right I do not know I took I took one centroid there maybe at one centroid somewhere here and maybe right another also here from here let us say right just by chance now after you after you do the assignment right now what will happen is so for example right so so around this green right you might you might actually get get a kind of the bunch of bunch of xi samples around it around the around the other centroid that you picked right you will have a will have a new set of samples now because you just started with some see arbitrary choice of the centroid now you start assigning right now that you have assigned samples each one will have some sort of neighbors around it right so so now you can actually compute the mean so so recompute now it is like saying that you have done done a group assignment right this algorithm has now done a group assignment it is saying that that this guy right now probably comes from there this guy probably comes from here and therefore you can actually recompute the the pattern centers or recompute this the centroids okay that means now now you have a fresh set of centers right you started with some arbitrary data points now you have data points within each group you have used them all to actually arrive at you know new means okay to be the so recompute there to be the centers of the to be the centers of the current clusters okay. Then four using the current centers again right reassign right current centers recompute the partitions or partitions group assignments whatever you want to call it recompute the partitions by again reassigning so now right you can again reassign okay now again again you need to go walk through from x1 to xn right now that the means have changed again see where it is going right it is not it is not it is not you know it is not essential that x1 if it was originally in mu2 it has now go to mu2 again it may it may go somewhere else right because your means have changed. You recompute the partitions by reassigning by reassigning what is that reassigning each each sample to the nearest centroid centroid okay then five if if the if the label assignment okay this is called this is called a label assignment right in fact if you if you have done a vector quantization right this is what this is like a form of vq okay if the if the label assignment remains unchanged or or if there is hardly any change right the label assignment is unchanged typically right the the kind of condition is it should be unchanged that means it is kind of it is going to see stabilized label assignment is unchanged stop else else go to go to see step three so no step two go to step two okay.

Now actually right this is actually a greedy algorithm in the sense that every time it is kind of looking at minimizing minimizing the cost right so so so in that sense it is very likely to get trapped in a in a in a local minima because it is it is totally a greedy algorithm right wants to wants always right keep that keep that error Euclidean error as small as possible therefore it is very likely that right when you actually end up computing these assignments you might have ended up in a local minima and that is the reason why why let us say in one run of k-means right you would not actually accept the final result what you would do is you know you again you will again start with another set of initial centers randomly picked again kind of do the assignment again run say another set of centers I mean it depends upon how much you can afford to do right given whatever time constraints you have and then after that what would you do which one which one would you pick I mean if I had let us say whatever if I did it let us say 20 times which one would I pick yeah which one whichever gives you that is why you are finding in time in step two the sum square error right. So the sum square error whenever for whatever iteration or whatever initialization you got the sum square error to be the minimum you will actually pick that pick that as your as your final final label assignment okay that is yeah for all the n samples right so that means whatever was there in the previous iteration every everybody is going back to the same clusters right x1 is going back to wherever it was x2 is going every one of them so nobody is getting shuffled right that is when that is when you stop right. Now this is what is actually k means and so this is the pros and cons right so the so the main con is that right you have to know k right which is which is like you know which is like a hard condition right a big condition yeah the other thing is it has problems in that you know in the sense that it will model spherical clusters right so it will kind of so because of this Euclidean nature of it right so it will always tend to be tend to sort of tend to sort of read group I think I have some examples right to show so maybe if we go and show that right it will become more clear. So if you see right here minus k means okay yeah so here so if you see right so if you have data like this and all right very very likely that with a with a few random trials right you will be able to you will you will be able to kind of write a group there it is totally unsupervised right as you can see when we do not know from where they came originally but then if you run a few few runs with let us say different initializations very very likely that if they are so very nicely spread out right then there is then there is no problem but of course you know there are still some problems here for example like this you are going to hard assign right this would be either here or there right and and you have you should be willing to accept that but then then the problem right might come for example for example if you had something like this right where let us say this is one cluster and then right right that is another cluster and if you try to use k means right it will try to it will try to kind of look for a spherical thing and then it will group like that which will be which will be entirely wrong right that is what but then that is what it will do because right that is the way it is and also the other problem right that it can have is with respect to outliers but these there are other ways to handle it I mean it is not so bad I mean there are there are ways to know right if something is an outlier and not really included inside a group therefore this I would say is not really such a big problem even though it is being shown here right but yeah but but something like this can happen right so when you have this this is a clusters coming in various forms and shapes right at the time okay k means will struggle okay therefore sensitive to outliers spherical clusters what will assume assuming and then sensitive to initial centers setting k and all that right but then it is simple and fast to compute okay.