Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-74

Now, bundle why do we call it as a bundle right that comes from this. So, for example, it imagine that imagine that I have a camera here, then I have a camera here maybe right I have something like that right whatever or else let me just draw it a little better. So, maybe right I go this way and then I go this way ok. Now, I have a scene ok, some points are there ok, this is one point, this is one point, there are several such points right in the scene whatever ok. Now, if you if you actually think about it, so here are my here are the say centers of my camera right which is what is the pose that I have got from my whatever whatever method I apply, I have some initial estimate, but those estimates I know right are not are not really exact right I know that probably there is an error and all of that. But if you really think about it right what will happen is see for example, I mean if you if you kind of look at this point right I mean you know it is going to come and probably hit here, the other point will come like that, I mean third point right will come like that and all this right I am assuming that that you have an image plane right which can actually over which this will fall and then maybe and then maybe right beyond a point you may not even be able to see these points.

For example, this camera may not see this, this and that it may not see ok, but then the next guy right perhaps right will see this ok and then it can see this, then maybe right it will also see this whatever right and then maybe maybe right it would not see this, maybe it will see this, but then not the other two points and so on right, so you can understand right. So, what is happening is so you have a you have your 3D scene right and there are these, so all these are your scene points right and and you have these rays coming out of out of the scene points and then each one is going and hitting the image plane of whatever camera is let us say right within its view. Now, if you if you if you look at the image plane of any any camera right, now you see that you see that a bundle of rays are coming and impinging on it right depending upon the upon the say number of scene points right which you have and of course, you know you have a whole image right it is not simply these points from where from where the rays are coming there is everything out there, but but then we know that we have only the correspondences for for some of for these points that we have shown there only for them we have the correspondences. And and what we are saying is kind of and then and then right this is the pose, so we have like C1, C2, C3, C4 we may just want to fix the world coordinate system with respect to C1 and then think about everything else is like some R, T right that we need to find out and we have some we have some initial estimates right for for the for the poses and we have some initial estimates for the for the x, y, z also.

But we are saying that right that by itself may not be may not be exact because of the feature correspondence errors and also, so ideally if you look at it what should you be doing right if you look at this bundle of rays that is coming in, so bundle adjustment means that we want to adjust these rays what does it mean adjust these rays right in the sense that these these 3D points themselves may have to move around a little bit it need not stay where you have given as initial estimate even 3D points may have to move around around around the place right where they are they may have to move around a little bit and at the same time what should also happen is perhaps the camera poses rate should also probably move move a little bit because they may not be exact right. So the centres of the camera could also be moving around a little bit you may have to move them around such that when you actually project right the reprojection error as it is called I mean I have not used that word till now but reprojection error is something like I have I have a point correspondence for that for that for that particular C3D point let us say let us say for this 3D point right I know I know that I know that right that my that my correspondence is here but then when this point moves during my optimization right when this point moves and then and then when I join it with the optical centre right it may hit somewhere close by right it does not have to hit there because I do not know whether that itself is correct. So when it when it in this kind of optimization when this ray comes and hits now the reprojection error you want that to be as small as possible right because you can only go by what you have and you have these you have this is a correspondence with you. So if you look at look at the correspondence this is right now to the way to see the see a correspondence is fix a scene point and that scene point right you kind of pass it through the one of the optical centres you get you get one point pass it through the other optical centre you will get another point on the image plane right which is which is a projection that you are doing and if you have point correspondences you know where these should come right I mean you know you know the correspondence is with you you are kind of doing this bundle adjustment and you are trying to see how how should I get this reprojection error to be as small as possible right that is the that is the idea I mean is that clear. Of course you know it is a it is a kind of a it is more a kind of you know statistical kind of framework which I will which I will tell now but the point is this is the this is the idea right everything else is a math and all is okay right we can we can kind of see figure that out but but our want is to sink in when you say bundle adjustment right this is exactly what you mean okay.

So I am not going to write all this okay because I think I have already said okay so so the idea is this right so what what you want to do is so you want to account for noise okay account for noise in the in the feature correspondences correspondences and some of them some of them could be say outright wrong I mean it is not just just a small amount of noise some of them could be could be outright wrong in the sense that you are saying that this point is that and then that could be actually wrong right could be outright wrong outright wrong and you see until now right what we have seen we have we have only used what are called what are called linear approaches we have only used linear approaches right I mean we have always been going to say SVD and write a manner of solving everything use linear approaches but but then if you want to account for such things right where you have to account for noise and right I mean noise in the feature correspondences because finally the only information that you have are these feature correspondences right let us again let us be clear about it there is no other information that we have whatever you derived is also based upon that and whatever you are going to derive is will also be based on that right nothing else we have. So all that we have we know is right from that from that from that point right 3D point if I come right plot a plot array which actually it intersects the center wherever it intersects that is where that is where my actual sort of rate of point should be whether whether whether my sort of feature correspondence occurs there or not we do not know that is what that is what this is trying to do. So we have used only right linear approaches and now it is time to sort of account for noise and the moment you account for noise right what does it mean so it means that means that if you kind of think about your x ~ at which we have which we have written as P x ~ right plus some let us say n right because this noise right could actually come okay. So what we are saying is right there could be a noise and the simplest thing to do is to assume that right this is actually AWGN the other thing right that we have not also taken into account in the previous thing is that what if what if let us say if one point right you cannot you cannot see in let us say in some other frame right I mean you know when you are doing this right it is quite it is quite possible like I showed in that particular figure right that that some points may not even be seen by certain camera. So what is called you know a visibility map right a visibility map you have to have right so that so that you know like which ones when should you take that cost into account and when you should not I mean you do not even see that correspondence there is no point trying to trying to write compute the error right.

So so what one does is the following right and since since you are assuming noise to be AWGN right so the so the so the thing to kind of look at it is look at look at look at what is called what is called right a map estimate what is called what is called a maximum a posteriori. So for example right I mean what this means is that if you have some kind of a prior knowledge about let us say x now I am going to write this in a sort of a simple form suppose suppose I say that you know x, r and then t right these are the only unknowns that I have by x right what I mean is I mean the seen points xj the set of seen points xj okay j equal to whatever what am I using here let me just be kind of consistent so j is equal to 1 to n okay that means I have I have that many seen points each is a each is a vector by the way okay. So this x is actually a collection of seen points which I have to find out r is of course you know rotation but again right r will have like you know so r will again be a set so it is like it is like ri right i is equal to how many views you have what am I using here m okay 1 to m and similarly t right will actually which consists of all these poses right ti i equal to 1 to m right all these are all these are unknowns okay and the idea is that right when you when you when you say that when you say that maximum a posteriori right what you mean is after having observed what have you observed the I mean feature correspondence right nothing else okay so after having observed so you want to say that right can I kind of maximize right p you know x, r, t after having seen right x ~ I mean there could be a prior prior prior probability which we do not even know what it is right I mean how they should occur together and all we do not even know. So therefore typically right they simply used as a uniform just assume to be uniform because you do not have any any any specific knowledge about it so simply assume it to be uniform and therefore right what what you want to do want to do is maximize the maximize the kind of posterior probability which is like having seen x ∼ right I mean you know I want to I want to kind of maximize the posterior but then this we know is is actually proportional to proportional to p of x ~ right given whatever x, r, t by by x, r, t right mean I mean the entire group of poses the entire set of 3D points right does not one pose or one camera pose right I mean the whole set right p of x ~ right and into whatever right p of x, r, t right and since right this we know we do not have any knowledge about this or we can simply use it to be uniform okay then then then as you know right map and MLE become the same now. So now we are looking at something like a like a like a maximum likelihood estimation right so what what does it mean it means the right find that that x, r, t combination which maximizes the likelihood of having seen x ~ right when x ~ is what you observe and therefore it maximum likelihood is what find the unknown whatever θ or the parameters that you have such that it maximizes the likelihood of having seen the observations that I have seen right so in this case x ~ so x ~ by x ~ again mean the entire set of feature correspondence not one feature correspondence right I am just using a notation like this but I hope the the the information right is clear that that we are looking at the entire set of feature correspondence right and and and and right if you have a model like this where let us say x ~ is actually related to x, r, t is in this form right this this this p will contain contain your r and t x ~ has your x right then then what does it mean so so this is equivalent to equivalent to maximizing let us say right with respect of course you know x, r, t okay you want to maximize p if so what this means is n is equal to x ~ - p x ~ given whatever x, r, t right because we we are we are assuming an additive Gaussian model for the noise for the for the noise right and and and of course you know once you give x, r, t then then n is known right I mean you know there is there is no no no no uncertainty here once I tell x, r, t therefore this would be simply because you have assumed the noise to be Gaussian right so so you can even so you can actually you can actually throw this out that you do not have a dependence on that anymore.

So it is simply p of n equal to x ~ - p x ~ which is simply e raised to whatever it you will have some normalization and then e raised to power - whatever norm of x ~ - - p x ~ square right and x ~ is actually 3 X 1 that we know right all that we know okay. So now now you see that maximizing this probability right I mean is equal to then one should just take the logarithm so that right things become easy and all of this boils down to actually minimizing I mean maximizing that probability right the posterior is equal to minimize so so what you want is an arg max right with respect to what is this x, r, t and in this case it becomes arg min right because of the fact that noise is Gaussian so it becomes arg min norm x ~ - p x ~ square right this is what we want and and this encompasses this as r and t in it this has all your all your you see x's in it x ~ right all the all the see here right 3D scene points are kind of sitting in this x ~ and then p has r and t and this is what you want to do right. So basically effectively boils down to boils down to a problem like this and this right typically we write this as a kind of a loss function okay this whole thing the the usual form in which we will write this as let us say x ~ or I can even write it as x okay because you do not need that this is ~ really right you are only interested in x, r, t again the entire group of poses and because of the fact that right I have assumed m number of views okay so i goes from 1 to m I have assumed n number of scene points so j goes from 1 to n and then there will be a variable okay this I will explain okay there will be a variable θ i j that will multiply this norm I am going to write this as x i j - a projection pi pi is really a projection a projection operator which is a function of x j r i t i norm square what does that mean so so right what is actually actually effectively means is now we are coming from see earlier right we are we were trying to match the feature correspondences right directly now we are coming in an indirect way so so right what we are saying is if you fix a view right so if you actually what to say right I mean if you actually fix okay j equal to 1 to n right so to kind of right think of okay x i j is okay let me just write this one is jth feature point okay so this is the x i j is the jth feature point in the ith view ith view means ith image okay in the ith view right so what you are saying it is like this right so you have so so jth feature point means that particular 3d the corresponding 3d point right whose image coordinate is is x i j that is what we mean okay so you have some some coordinate like say x j and and and then what then what you are saying is this has a correspondence sort of you know in the sense that we know that we know that x j maps to some okay in the ith view let us say in the first first view right this is some x 1 what is that I mean so if you have if you have if you call this as yeah ith view i is 1 right so I will call this as x 1 j then when I when actually when I okay now the way I have drawn it is not it is not very correct okay but I mean like why I should be drawing it like this so this is this is my this is my x 1 j and then let us say right I mean I have something like this right here is where my my other center is the camera center then this is my second view right so this is like x 2 j right so as my view changes and and and I know I know that I know that for example I know that this point right there is a correspondence which I got from shift or surf or something but when I when I when I could have do this bundle adjustment this x j can actually move around and when it moves around this point can shift but I have the observation which is my which is my x ~ right so so so this is my observed point which is coming from some surf or shift and this is a projection okay this guy is this guy this is like x i j dash or something if you want to call it and this is called the called the reprojection error right this term is actually the you know reprojection error is this clear is the same kind of thing right whatever we saw earlier except that right and and and all of this we need to do for every kind of 3D scene point. So if you fix a 3D scene point you can say that say that a correspondence that I that that I that I ought to get for this aX the images should match as closely as possible to a projection of this point onto let us say every camera center onto the image plane right and then and then I change my scene point I go to the next scene point I do the same thing for that again right I have I have a bunch of feature correspondences I will have I will have around a projection that I will do I will say that right make this as small as possible right that is all you are doing okay and then this this θ i j is actually you know a visibility visibility rate what this means is that if you if you if you do not have a correspondence okay then you are sort of assuming that probably that is not actually visible it would also be that your shift did not return it for some reason whatever it could be right but we simply assume that assume that right in this summation right now when we are doing from i equal to 1 to m so in a particular view if you do not have a correspondence then we will assume that is the θ i j should be just 0 there I mean we should not take this sum there into account because we do not have a correspondence okay so this θ i j is 0 is or let us say θ i j is 1 is 1 if if the if the jth point is actually visible in the in the i th view in the i th view jth point means that is the 3d 3d scene point if it is visible else it is 0 else it is 0 all that it means is in the summation right that that term you would not include you will sort of throw it away and then do the summation over the rest okay now there is a there is a there is a simplification of this right which you can do so for example right I mean so this entire operation is what this is nothing but I see p x ~ right that is what this is not this this is projection so if you write your p as you know involving everything your intrinsic everything right so if you write it I mean you will have like p 1 1 2 whatever p m 1 4 and then p 3 1 2 p 3 4 it is what you have as actually a 3 X 4 matrix right this is what your p so if you do if you do p into x ~ right that will give you that will give you a 3 X 1 vector right and suppose you call the you call the first scalar value right suppose you call this as what is that notation p i j okay so suppose you call this as p i j okay this is small p by the way okay this is q i j and this is lambda i j okay what does what this means is that p i j is but the first row or first element in fact I mean there is no there is no like a row there is only one value first row of p p x ~ this is the second row of p x ~ row means there is only one value right it is a vector 4 X 1 multiplying sorry 1 X 4 multiplying a 4 X just a value right whatever is that value that is p i j and therefore what you can do is you can replace this guy and you know I mean instead of having 3 elements in x i j you can actually you know write or reduce it to 2 and what you can do is you can write this l so the simplified form right will look like this x l r t is equal to again the summation over i and j and then θ i j will remain and then inside right you can write this as x i j now now right that is a vector now I have now brought it down to a scalar now okay and p i j by lambda i j square plus y i j - q i j this is all you know now right lambda i j square I mean that is your that is your third element no you have to scale the first and second by the third right to get the actual coordinate so the actual image coordinate is x i j y i j right that is the actual image coordinate and therefore right this can be further simplified into a form like that and θ i j all that is the same right and this is what this is what is now fed to actually this is this you can solve in different ways but then the most most most used algorithm is what you know is due to is actually a non-linear least squares method it is already there it is not built for this for this structure from motion at all it is called Levenberg-Marquois I think I have got this spelling correct Levenberg-Marquois method of non-linear least squares okay and this is actually you know a non-convex non-convex optimization okay so this is a non-linear least squares least squares method which is available in MATLAB and all of that right I mean there was a time when I used to actually teach this but it is not needed okay people simply use this now okay I mean even even Bundler and all just comes as a comes as a package now just throw your thing inside it will just run it for you okay so this so this Levenberg-Marquois is what you what you have to use to solve this.