Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-72

So, multi view means now right until now I talked about right two views, now you know you can have right n number of views and you can ask how do I how do I do my scene reconstruction, the whole idea being that you should be able to go around and take right as many views as you can, you can actually cover the object nicely and then you can do it for your buildings or whatever right outside. Now, the idea is that right multi view you know there are typically two approaches, one is what is called sequential or what is called you see incremental, incremental sort of when I say SFM and another is actually a parallel, sequential and parallel or what is called batch, batch means you kind of operate on on a certain number of them simultaneously right that is batch. So, sequential is also called actually you know incremental, incremental you see SFM. So, sequential is kind of seldom used, you know sequential actually means that right just like just like we did a triangulation right last time okay I have actually two views right and then and then let us say let us say let us assume that it is a calibrated case or whatever then then with two views right if I if I do a triangulation then I can go like so 1 to 2 I have done. So, for example, when I go go from C 2 to 3 right again I mean all of this will assume that I have feature correspondences right based upon which I can estimate my essential matrix and all that then from there I get my P and from there I go right I can actually go back and triangulate. The only problem that that happens is that your translation right because it involves involves a scale right it needs to be fixed.

So, what what people typically do is from 1 and 2 right once you have estimated some 3D points when you go from 2 to 3 look for at least one common point between the two right that that is visible both in 1 and 2 as well as in actually rate 2 and 3 which is which is common okay then then you can just look at what you reconstructed for that point in between using 1 and 2 and and basically and you know scale the translation of 2 and 3 right by that factor okay just to make sure that make sure that that your that your see translations do not right I mean there has to be there has to be what you call you know a systematism when you go from 1 to 2, 2 to 3, 3 to 4 right. So, so one way to sort of one way to sort of make sure that make sure that your translations are are in order right what you do is you pick one common point or more if you wish you can average right if you want more at least one right which is common in both the views use that use the already

computed 3D coordinate of that to know as to what by what factor the translation ought to be scaled okay that is that is that is typically how the incremental is have work. So, you will add one view at a time, but then right that does not seem to make too much sense because it is there I mean it is not like rate you know it cannot be done, but then if you do that right better than that is to actually look at what is called what is called batch processing. So, batch processing will mean that will mean that I have got lot of views I cannot use all of them simultaneously because you know it can it is so it can it is possible that there are certain points which you cannot you cannot even see in the other views right you would have gone too far away.

So, what you can see let us say in the one first ten frames right may be in the eleventh frame you do not even see anything common between within one and eleven right. So, what is done is you do what is called what is called a batch processing. So, so you take a bunch of images where at least right you can assume that assume that at all points are basically seen by you know in all the frames. It is an assumption right that they make it does not have to be strictly obeyed, but that is a that is a kind of an assumption that is made and the and the and the most popular method right for that is for that is what is called a factorization method. This is due to two people Thomasi-Karnaday factorization that is what it is called there is actually two forms of this there is a perspective model a projection model which is the most most general right that we should be using, but then that is that is that is very complicated in the sense that it will take you know two to three classes to just do that.

So, instead of that just to give you a view for just to give you an idea right how this works right I am going to use what is called what is called as a orthographic model. Like I said right there are actually different camera models we have not talked about all that. So, there is something called you know a perspective model which is the most general that we have been using then depending upon certain intrinsic that you can that you can that you know about the camera right you can actually reduce it to something like you can make it a fine then from a fine that you can get something like you know scalar orthographic then you see orthographic and so on. There are various models and we are not going to kind of look at all of that, but just that this one it uses an orthographic model. So, this is orthographic model right it is not you know there is hardly any real camera that follows an orthographic model.

What what does orthographic model mean that your P matrix right without without the extrinsics will look like this 1 0 0 0 0 1 0 0 0 0 0 1. What has changed between the earlier and this? Whenever I was writing a perspective model right I had a k on the left and then something right on the right and then then I would have my extrinsic right. So, this P what has changed between last earlier and now? Can't see third row third row had had usually

one would be there right incorporating in co involving z right now this would not involve z. So, what this actually means is the the the assumption is that the camera center is sufficiently far away from the image plane. So, that you can make the assumption that the rays that are actually falling from the object onto the image plane are actually orthogonal to the image plane.

So, it is like you are getting a bunch of parallel parallel rays. So, it is like saying the camera center right is very far away right that is when the orthographic model is valid right. Now, it is not true that your regular camera is follow this, but then there is something called a telecentric lens which actually uses this orthographic model which actually works on this principle and except that the camera is kind of right is built that way and it is used in industrial inspection because in industrial inspection right if you if you move an object behind or front right when they do this let us say you have a conveyor built in which objects are going and you want to recognize an object. If that object comes forward or backward right with respect to the camera if the scale changes of the object right then you have to train machine learning and all right to accommodate for scale and all that. Whereas orthographic means wherever you are right I do not I do not get a sense for size at all it will all look the same because it is like right it is like.

So, what will be a small x in terms of the scene the capital X small y capital Y right there is no there is no there is no there is no sort of a you know a dependence on z at all. So, it basically means that either I further I keep it here I get the same image if I push it back I get the same image if I bring it back I get the same image that means then it totally becomes independent of z right that is why it is called an it is called an orthographic model and it is not like you know it is it is it is it is it is like I said right most camera models do not follow this law, but a telecentric lens and all does it and this is actually you know sometimes people use these camera models in order to just bring in in order to just to see right in order to make matters a little simple. So that things do not become unwieldy for example, the same thing right if I did using a perspective model right it will become very complicated. So, the idea is to just convey that when you want to do this kind of structure from motion even with just with an orthographic model this is a simplification, but even this will actually require some effort right as we will see ok. Now, the first thing right that that happens with respect to an orthographic model right is this.

So, if you have. So, what this means is that. So, now, right I am just going to use u v and 1 because just to stick to this common literature ok literature on this factorization method uses u v instead of small x small y. So, just to stick to standard notation for this method that I am using u v 1 therefore, you can kind of look at this as let us say 1 0 0 0 0 1 0 0 0 0 0 1 and then if you have an say extrinsics coming in. So, that is like R T 0 transpose 1 and then right you have you have to say right x tilde which is which is x y z 1 right this is

what you have.

Now, what you can do is right you can you can actually show that right this is equal to this is simple to show this you can be shown as R 1 transpose T 1 R 2 transpose R 2 transpose T 2 and then 0 transpose 1 into x y z 1 this is all simple I am not going to going to. So, what is R 1 transpose just the first row of R R 2 transpose is the second row of R and then because of because of this model right what is happening is R 3 transpose in fact has no role to play and similarly T 3 as you can see has no role to play because of this because of the model right which you are using which is an orthographic model and then sorry + yeah. So, I think this you can this you can simplify if I just pull out u and v right then this you can write as R 1 transpose R 2 transpose right x y z + this guy T 1 T 2 that is T 1 T 2 are the first two elements of T R 1 and R 2 are R 1 transpose and R 2 transpose are the kind of two rows of first two rows of R. Now, this how actually it will simplify this because we are using an orthographic model otherwise you will have you will have no right other things coming in. So, the idea is this right.

So, the idea is like this. So, if you take. So, right I am going to introduce you know a notation which says that u f q and v f q what this means is that you have let us say now suppose you have suppose you have q points in all see ideally right what do you want to do we want to do structure from motion right. So, ideally you have let us say q points q points and there are let us say let us say f frames f number of frames right. That means, you have like one to f frames and the point correspondences are available for actually q points that means, all q points are seen by seen by all the all the images in all the views that is why it is called batch right.

So, this is like batch processing right. So, if you taken. So, you have taken a batch of f frames the the assumption right being made is that batch of f frames the assumption that is being made is that all these point correspondences which you are talking about the q points they are all available there they can all be seen in all of them right. I I I could leave it you to think about well if I can if I can see it in just two of them right why should it then involve right f number of them what would be your answer. I mean you can ask know if I can see it in everything then even in two I can see well you know the idea is that it is going to be more see robust right because because it needs to satisfy in every one of those frame right you have to say that right it is not enough if if I can see it only in these two views it should be still right I should be able to see it in the third one I should be able to right see it in the fourth one and so on right.

And the whole idea is that is this is right like I said the last time they kind of see robustness that it can actually bring in with m number of views and also and also right when this is only one batch right and then of course, you can move on to the next batch then you can

you can move on to the next batch and then the idea is that this x y z that you want to reconstruct right you want to be able to right reconstruct that way. There are there is something called a bundle adjustment which I will do in the next class where let us say none of these needs to be valid for this method it requires this, but even this you can actually ask if I can see it in two views if I if I am putting a constraint that I can see it all all of them why cannot I just use the first two right, but the idea is that there is going to be robustness, but in reality you do not need any of this a bundling a bundler right as it is called does not need any of these assumptions ok, but this is for this method ok. So, batch of f frames and what this means is that right. So, you have. So, I am going to just introduce a introduce a notation right this is just going by that paper ok.

So, this is like j f transpose. So, do not think that i j are something crazy i f transpose just a first row of of r for that view ok this is for the f th view right f is a frame number right. So, so the small f is representing the frame number therefore, i f and r j f and all they represent r f the first row of r f the second row of r f and so on and then x y z and then here right I am going to write this as what do I write t f 1 and t f t f 2 ok this is again t t f right. So, this is a translation with respect to the f th view. Now, if you stack all the points right this is just this is just for let us say one point correspondence this is just for not even a point correspondence this is one point right that I have.

Now, I actually have q number of such points right that is what I assume I have q number of points. So, then what I can do is I can write this as u f q sorry not not u f q I can write this as let us say u f 1 v f 1 u f 2 v f 2 whatever right I write all the way up to u f q v f q this is still I am still within a frame right I am still within the f th frame then what will be here this is still i f transpose see this is this size is what this is 2 cross q right and this is i f transpose a f transpose. 2 cross 3. This is 2 cross 3.

3 cross 3. That will be yes. So, this will be like x 1 y 1 z 1 x 2 y 2 z 2 sorry z 2 and then all the way up to x q y q z q right because each of these coordinates is coming from a different scene point right u I mean u f 1 is coming from x 1 y 1 u f 1 v 1 v f 1 is coming from x 1 y 1 z 1 u f 2 v f 2 is coming from x 2 y 2 z 2 and so on right. So, this will be like what 3 cross q right. So, you are.

So, you say right. So, dimensions are all ok right + + what will you have here there should be translation no. t 1 t 1 then then what is it like. So, so let us say suppose I write this is t f it is in fact t f all the way yeah right no because we are still in this f th frame right actually it has to be t f only right all the way it cannot be it cannot be t 1 1 t 2 1 and all this is for the f th frame right that is why I mean it just looked a little crazy, but yeah that is how it is it is just t f all the way because everything is just t f 1 t f 2 t f 1 t f 2 t f 1 t f 2 right for every point the translations are same no the camera pose is the same i f j f did not change

right they remain the same therefore, yeah right this is correct ok this will be like 2 cross 2 cross q 2 cross q. So, this is like t f repeated q times ok yeah it is correct this is there is nothing wrong. Now, now what you can do now if you if you bring in the the this for this for 1 frame right this for q points in 1 frame ok that we were able to write this.

Now, suppose I want to write the q points for all the frames right that means across all the views. So, if I stack them then what you will get is this right if I stack all of them is always something that. So, then actually we write it in a particular form right now I am going to write it precisely like that. So, u 1 1 u 1 2 all the way up to u 1 q and then what we do is we actually write it as u 2 1 I mean. So, it is like u 1 1 and u 1 2 will come after I mean you will just see the form ok just that it is not exactly written in that form you just kind of swap a little bit and then you have like u f 1.

So, that means the first coordinate right you write for all the f frames first then you write the say second coordinates ok. So, u f 2 and then u f q and similarly right you have like v 1 1. So, you will have like. So, ideally right you had like u 1 1 v 1 1 u 1 2 v 1 2 right they were put next to each other right instead of that we are just rearranging them. So, that we can write this as v 1 2 v 1 q and all the way up to whatever v f 1 v f 2 v f q right.

Now, this right now if you stack all this up right then what you can do is now you do not need this big right. So, what you can do is you can write this as I just leave it to you to kind of check this this should be very straight forward to check I 2 transpose I f transpose then J 1 transpose J 2 transpose all the way up to J f transpose. Now, now you have got rotations across the views now right you have got you have got all of them coming in I 1 J 1 I 2 J 2 all of them have to come in now into this guy is x 1 y 1 z 1 right this will remain the same yeah x q y q z q and then + here right the way we write it is this will be like t 1 1 all the way up to t 1 1 then you will have like t 2 1 all the way up to t 2 1 then yeah then it is like t f 1 t f 1 and then again this again this is second coordinate. So, it will be like t 1 2 to t 1 2 then all the way up to t f 2 to actually t f 2 it is not square this is t f 2. Now, if you see this size right what is the what is the size of this guy 2 f 2 f right 2 f cross q what is the size of this 2 f cross 3 correct 2 f cross 3 and this guy is 3 cross q right and this guy is 2 f cross cross q ok.

So, just see if the if the dimensions and all match yeah 2 f cross q right. Now, if you see this right this is interesting think of this what is this this is your image data right this is all your feature correspondence is across m number of whatever f number of views right you have all the. So, this is your image measurement what is this this involves the camera rotation for all the frames what is this this is your scene this is just your scene right x y z. So, this is your 3 d structure and this is your the camera translation right. So, what this means is now I hope you can see the structure from motion problem right what now you

have you have like on the there on the left you have got all the accumulated all the all the data that you have the correspondences that you have of course, you know.

So, what this means is that if you kind of walk along a column right if you if you kind of go down a column it will be it will be it will be the if you go along a column it will be the same point appearing in f number of views right one column is like is like the same correspondence the same point appearing in f number of frames when you go to the second column it is like the second point then across f number of frames right that is how you will interpret a column of this of this is measurement matrix right column is like fix a point go across to see where all where all you can find you know a correspondence then go to the next one that is the next feature point go across a correspondence right. So, now the idea is that given this we need to be able to solve this right and and the idea is that right if you can somehow split it now in the form that you can solve it and then you get your structure and then you get your motion that is that is what is basically right you know that is what is you see s f m structure from motion we have run out of time right. So, I will I will take that next class on Friday.