Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-42

So, let us start with the feature detectors. So, why do you think you need features? By the way, right, what would be your understanding of a feature? Yeah, which not just one point, maybe a collection of, a collection of let us say interest points or things like that, right. So you can think of these as being interest points or something like that, right. So which kind of allows you to, allows you to, allows you to, you know, identify certain things in one image and then maybe you can actually match it across another image, right. And this feature detector thing, right, often arises, so, okay, so I think let me start by kind of talking about types of features. So example could be corners, well, in fact, we could start with lines and edges, right, that is something that we saw earlier.

So you can think of lines, corners, blobs, it could be so many things that you can think of as features and so on, I mean, so, etc. And when we talk about features like these, right, these are, these are, these are kind of, these are not image specific, okay, these are kind of generic in nature, by which I mean that, for example, if you look at how they match fingerprint and all that, those are very, very specific kind of features that are, that are kind of unique to, unique to, unique to, you know, fingerprints and so on, that is not like that, right. Even facial features and all sometimes when you do face recognition, there are certain things, certain things you actually find out, I mean, that helps you recognize people and so on, but does not like that. So when we are talking about feature detectors, we are talking about something which is much more generic, right, so which should be something that is uniform, you know, whichever image I pick, I should be able to tell where are the lines, where are the corners and so on.

Okay, so what is the need for these? Why do we need them? The need for features, right, so I can list a bunch of things, okay, that, that, you know, where it is useful. So need for features, you could use it for, let us say, mosaicing is one of the most common things that it is used for. It basically means that, you know, you have, you have an image and then you have actually, you have, let us say, right, multiple images of a scene and you want to get a stitch them all together in order to create a panorama, right. So, so then, I mean, you need to be able to find out what is the, say, relative geometric alignment between these images so that, you know, they can all be aligned properly, otherwise you cannot just put them one next to another, okay, that would not look good. So you have to align them and therefore to align them, right, what you need are, what you need are actually features that you can track across.

So you should be able to tell that this feature has gone here and the same feature has perhaps appeared here and that way you can kind of keep track of where the feature is. That kind of helps you and if you have enough number of them, then you can actually do this kind of geometric alignment, okay. And this mosaicing does not even require depth information, that is what is nice about it, okay. It does not require you to know the depth of the scene. Without that, right, you can actually do it purely based upon, based upon what is called homography, which I think we will talk about in very, you know, very briefly when we do this one, when we do, what do you call, this one geometry.

Then there is something called stereo, right, or what is called multiview stereo. Stereo typically means two view, multiview typically means multiple views. And here, right, this might be more from a point of understanding about what is this depth of a point, right. Here like I said, there is no notion of really a depth coming in, you do not really care about that. You just want to be able to align images, whereas in stereo, right, like I said before, you may have a scene point somewhere and then, right, you have a camera, right, which is whose center is here and then you have another camera, right, whose center is here and both of them are actually watching this and you have a plane here, the image plane here, okay, and this point, right, lies here and when seen through this view, seen through this viewpoint, right.

So you see like that and if you just see it from one view, which let us say view number one and then you have got another which is like view number two, when you see from view number one, so this is a 3D point, right, some P point, so let us say it falls here on the image plane and when you are watching it from elsewhere, right, then maybe it falls here, right, according to this ray and you have to triangulate to tell where this point is. So from one view, right, you can of course extend this ray behind but you do not know where to stop and it goes all the way, correct, I mean it just goes all the way, so you do not know where to stop, whereas if you could do triangulation that means you have another view and from which, right, but then what would you need then if I had to triangulate, if I had to reach this point, right, what would I need here? Yeah, you need to know a correspondence, right, you need to know that this point is here, then only this line can be drawn, right, otherwise you cannot draw it arbitrarily, so you need to know that this point has actually occurred here and therefore, right, you can actually extend this ray and then see where they meet, I mean in an ideal situation they would meet but given noise and all that, they may not actually always intersect but they will be very close and then based on some least squares since you can sort of tell where that point is, right. So again in such cases you are actually doing a correspondence matching, what is called the feature matching problem and in all of this, right, there are actually 2 things which you can see, one is the fact that you know there is something called a feature, I mean you need a detector, right, that means you need to be able to flag features in an image that means somebody gives you an image, you need to tell what are all the, you know, interesting features out there, then you need to be able to match them, right, so suppose I have 2, 3 images, I find features here, I find features here, I find features here but just that is not enough, I need to be able to do one more thing which is to be able to match them because unless I match them, right, none of this is very useful, I mean if I simply get feature points in every image what do I do with them, right, except for knowing that you know that this image has a lot of activity but then beyond that I cannot make much use of those images, so what you need is a descriptor, right. So in that sense, right, you have a detector and then you have what is called, you know, a descriptor, a descriptor actually defines the feature so that you are able to match, okay, because it should have something you know that, that borrows from whatever is around it and then it should be able to tell that okay this is the way my, this is the way I am right at that point and if I look the same in another point, right, then it is me, right, the other image and that is how you have to actually and of course you know you can make use of certain constraints in order to be able to not, you know, look all over the image, right, if possible for example in stereo and all, right. There are ways to actually restrict the search space and so on that we will talk about when we do the, when we do actually geometry but in general, right, let us just say that we are able to match the features and then if you can do the matching through a description, through you know a descriptor feature, this one descriptor, then you can actually you know do a lot of things, right and other, some other things that you can do is for example if people in fact match, you know, different viewpoints, of course you know this is all assumed and you know this is actually subsumed in that but this I am putting because you know this is being used for example there was a time when you know people before deep learning, advent of deep learning, there was a lot of work that used to show that given an image taken in the day and then given the same image taken in the night and from a totally different viewpoint that you could still match all those features and establish that it is the same scene.

I mean to a reasonable extent you could say that this perhaps is the same scene that we actually, right, you know, took in the morning. So again you need to have enough number of matching there, that is when you can, you can say with some confidence that because photometrically you cannot match them because this day that is night or different times of the day, right, so you cannot simply, you know, you cannot subtract or something, right, you cannot do that but what you can do is you can, you can match enough number of, if there is enough number of matches that happen then you can say, right. So this is like day night, then object recognition, okay, it can even be applied for let us say recognizing an object which basically means that, you know, if you wanted to, if you wanted to do some kind of an identification, right, of, this in particular object then maybe you can actually, you know, get what is called, you know, what would be, what would be like, you know, like sort of, you know, holistic sort of a description of that object, right, based upon the features that you have and then based upon that you can, you can match them. There is something called HOG, Histogram of the Grade, of Oriented Gradients. So there are various types of, you know, these features, so, I mean each one has something, you know, that is, that is sort of interesting and the motion estimation.

So this is like, this is again something that is useful for example and if you want to, if you want to track, right, if you want to track a point across frame, right, you have let us say a video. Suppose let us say, right, you have, you know, a cricket ball, somebody is, you know, bowling and you have a, you have a video and then suppose you wanted to track that ball, right, so you can actually make use of, again you need features to be able to track that ball, right. So you can use something like, you know, a temporal consistency and so on because you know that, you know, between frames there is not going to be much of a movement. That means the ball cannot just suddenly go somewhere else, right. So there is automatically a constraint.

So you can have constraints like, you know, it cannot have moved too much, it cannot have changed its intensity too much within a window and you can actually formulate problems around that and be able to come up with features that allow you to track them across frames, right. The main difference between that and the other ones that I showed, the earlier ones is that, the earlier ones, right, they do not care about any kind of a temporal consistency. It is like today I take an image, tomorrow I take an image, there is no consistency in terms of, you know, the time or anything, right, just that two, I have two images and I have to match them. Whereas motion, when you say, right, that is like a video, it is like a continuous activity, right, which you take in the, you know, within a minute or something. And then within that there is a lot of constraint, right.

I mean, like I keep saying, right, it is actually, you know, it is kind of, I mean, if you really sit back and think about it, right, a video, you know, has so much information and there is so much consistency there that we do not seem to, seem to even be aware of it, right. I mean, you know, you try to play around with the video, insert something that is wrong, do something to it, it will just immediately, immediately show up. I mean, it is very difficult to hide something. I mean, it is very difficult to, you know, tamper with it and do, of course, people keep tampering but I think that is, that all comes out because, because there is inherently, right, things are so nicely tied up that you cannot, you cannot, you know, alter it so easily. Whereas static scenes, you know, are relatively easy to play around with.

The moment you put a video, right, so, so there are features that make use of constraints like that where, where you sort of impose, you know, impose constraints of the kind that

I, that I can just now mention. That means, you know, the, the motion would be slow for cross frames and then the illumination would not have changed so much and so on. Okay, so the properties of features, right, so I will write down a few of them, okay, what is called repeatability, okay, which, which actually, which actually means that repeatability means that if I, if I do it on one image, next time I, I, I write attempted, I mean, it may be a slightly sort of a different, you may take an slightly different condition but I should be able to get the, get the same feature there, right. So that repeatability is important. Then this one, a distinctiveness, right, I think somebody pointed out.

So it should not be, it should not be something, you know, which, you know, which, which, I mean, it should stand out, right, but distinctiveness is more in terms of, so it should be, right, easy to match. What we mean is by this, what we mean is it should be easy to match. Then locality, okay, so normally, right, we expect the features to be kind of local, okay. We do not like PCA or something which is more or less a global kind of a feature, right. PCA can also give you features and all, that is more like a kind of a global feature but most of the time, why do you think that a global feature is not a good idea? What would it, what, what kind of problems would that run into? Sort of, yes, maybe, yeah, that I agree, what are the, what are the problems that you have? Mainly, right, the problems that you have is in terms of what are called occlusions.

See, for example, if you are seeing a spanner, let us say, right, you want to see a spanner but something is, you know, sitting on top of it, a part of it, right, you do not even see. Then what happens if you are looking at a global feature, right, and you do not know whether it is occluded or not, right. So when you try to pick up a feature from there, right, it will not simply match at all because, because most, part of it is, right, you cannot even see. So therefore global features are, are not actually a good idea. So what is, what is, what, what would be good is if it is in the spanner, right, if you can pick up a lot of local features and then you say that a major fraction matches.

So I have an original spanner which I know has these many features and then I have this sort of occluded spanner for which also I can get a bunch of features and I know that, and I, and I know, and see, so I will, when I match, there will only be a certain fraction but there will be still a certain fraction that will match. And maybe I can say that with some sort of reasonable probability, right, I might say that because so much fraction is matching, I might want to believe that, you know, probably it is a spanner, right. Because if I take a global thing there is no way to tell whether it is that object at all, right, because it simply will not match, okay, locality, right. So what that means is robust occlusions clutter and so on. Quantity, right, quantity means there should be, right, enough of them.

I mean, you know, it should not be like, you know, like in an image if I, if I apply these features I get only 3 of them or something, right. I should be able to get enough number of them. They should be repeatable, they should stand out distinctive, they should be locally expressive and quantity, right. So you should have, you know, enough of them, right. You should have enough of them so that we are able to, we are able to, you know, do some matching and so on.

Because, you know, many applications require, you know, it is not enough if you have just one feature or something, right. You will need a good number of them and normally there will be noise, right, when you try to match there is always going to be some noise in terms of, you know, what you are trying to match. And any amount of robustness that you want to bring to that noise will depend upon how many features you have, right. So you can, then, you know, that will sort of take care of some of the outliers and so on.

So you need this. Then accuracy. So by accuracy what we mean is, okay, it should be a precise localization, okay. That means, that means the feature, right, you should be able to tell where it is exactly, right. So when you say locality, right, that simply means that, you know, that is something which you can actually derive out of a local area. But that does not talk anything about how precise you are, you are in terms of telling where that particular feature is, right.

So the accuracy in terms of localization and most of the times it is all about where it is, okay, is the image. The intensity is and other things are being used to actually detect, you know, have a description of that feature. But in addition to that it is also important to know where it is in that image, okay, because that is what we will kind of say decide a lot of things, the, I mean, geometric information, right. So precise localization.

Then efficiency. Then efficiency which is like, which is like, you know, it should be, I mean, to the extent possible it should be something, you know, which you can do in a serial time. That means, you know, given an image, it should not take a long time, efficiency means computationally efficient, okay. So you should have, you should be able to get algorithms that can do it in a relatively small amount of time, right. So all these are, I mean, you can add this, add to this list, but then these are also the most important things that you would expect of a feature. Now, let me just show you a few examples just to give you, give an idea about, you know, what this might look like.

So for example, right, I mean, so for example, when you kind of say, talk about object recognition, right, you could have a database of objects. For example, you could have a phone, you could have a shoe, you could have this, you know, toy, soft toy and then, you know, you can have different features coming out of that and then you can store them and then, right, you can actually match them and then, yeah, right, this is what I mean. So when you want to do recognition under occlusion, right, so the fact that, the fact that, right, the phone is hidden by the shoe should not hamper the fact that, should not hamper your ability to be able to tell that there is a phone and, you know, behind the shoe. So such occlusions are very common. Then you see a location recognition, right.

So what that is, what this means is that, okay, so if you look at the top image and then if you look at the bottom image, they are not exactly taken from the same viewpoint, right, as you can see, because this is more of the left, right, as compared to the one on the top, right. So it is kind of a different viewpoint. And then all these things that you are seeing out there should stand out in terms of being features that you can actually pick up and match, you know, across, right, across images and that will give, that will, so location recognition means like I said, right, be it day or be it night, be it this viewpoint or that viewpoint, you should still be able to tell that I am probably looking at the same scene. Then this you can use for, you know, what is called, you know, I mean you can also use such features which you will see later to even tell where the camera is and so on. Then you can do image matching, this is somewhat similar to what we saw earlier.

So for example, even though the viewpoints are so very far away, right, quite far away between the two, but then you still should be able to, for example, like this one, look at the shape it takes there in the other image, but you still want to be able to match them, you want features that could, look at this, this is the Amnasa Mars rover image and if we as a human, right, if I ask you where are the points that you can match, does not look like an easy job at all, right, actually looks like these are two completely different scenes, but actually that is not the case. Apparently right, this bunch of points and this bunch of points are actually the same, okay, it is not, it is not, visually when you see the first time it looks like you are looking at, you know, a different terrain altogether. So as you can see, right, so you can see that there are, there are so many situations, you know, where you need, where you need features of this kind, okay, and which help you, will help you, you know, do these kinds of tasks. Now there is something, right, that so there is some invariance, okay, that we actually expect out of features, there is something called invariance and there is something called a covariance, not the covariance that we talk about, you know, in statistics, okay, these are covariant and invariant, right, so you know, so I think these are the two things that we use. So what we, what we actually believe, see changes in an image can occur due to actually two primary things, what could be those in an image, right, if you compare two images something is not the same, what could be those two things that actually makes the appearance of an image, you know, to be this one, a different, okay.

So one is, one is what is called photometric information, right, so photometric, so photometric information and what is the other, other could be geometry, right, because the viewpoint, right, you have the, can have the same lighting but then I take the image from another viewpoint, so that again makes it look, you know, it does not, those images do not look identical. So you have got what is called, what you called as, what you can call as geometric. So you see you have variations primarily, right, primarily, I mean, you know, you can take, you know, you can kind of think of changes occurring in an image because of photometry which is illumination or geometry which is typically based upon the viewpoint, okay. Now you want a feature, okay, so when you say that, you know, I want to be able to build features, so what you would like is you would want your features to be, let me say dash with respect to photometric changes, invariant, right, because so that if there is a slight change in intensity or if there is a sudden illumination, extra illumination, still want to be able to say that it is the same feature, right, invariant. And features to be with respect to, okay, now I think you can actually guess what might be the other answer because there are only 2 of them.

What would that mean if I write covariant, what would that mean? Yeah, they both should change the same way. For example, write a corner, right, if you had a corner somewhere in an image and suppose I did a rotation of that image then the corner should also rotate. If I translate, the corner should also translate, right. And means it is like saying that something like this, right, if I had phi on some f, okay, and if I had f, right, that is my feature, right, f is my let us say image, what you are saying is if I do phi of r of f, that means r, I mean, you know, by which I mean some kind of a geometric transformation or you can call it g of f, then I want this to be the same as r of phi of f, right, so that is a covariant property, okay, that is what we are asking and it makes sense, no? I mean if you scale an image, right, then again, I mean, depending upon the scales, see finally, right, see when you want to, see this is different from doing the matching. See when you want to match, okay, you want a descriptor that should probably be, right, invariant to all of this, a descriptor, a descriptor is not the same as a feature.

A feature is something that you want to track, right, I mean if you actually rotate the image then, then it should have also, it should have also gone the same way, so that, so that I know that a rotation happened, otherwise otherwise how would I know that actually rotation happened, right. So this covariant feature has to do with the fact that, you know, if I, if I kind of get that feature and if I have rotated that, that, that image itself, then rotated whatever you did, some geometric, you know, it could be affine, whatever it is that you do, then that same transformation should then get applied on the feature, so that, that is what will tell me information about what has happened between the 2 images, right. But, but then the thing is when I, when I match those 2 images that, that matching when I do, right, at that time the sensitivity to illumination and sensitivity to geometry and all should not be there because the underlying description should be independent of that, I mean otherwise if that becomes sensitive to all of them then I cannot ever match, right, so that description part we will come to later. This is okay Rohit, I think you are, you are, you look a little okay with that, okay. Which one? A quantity means enough number of them, that means, right, given an image, I mean if you say that I can only, I can only get you one feature, right, I have a feature, this one, I have an algorithm to find a feature, but that feature is so sparse that I can get only say 2 of them in an, what do I do with them? I should have enough number of them, right, they should be distinctive, there should be enough number of them because most of the times it is not enough if you just have one correspondence and all, you actually need multiple number of them, that is what I mean by enough.

So, enough number of them, enough of them means enough number of them, there should be enough number of them, okay. Now, there are, there are various, there are, there are several types of FDs, you know, detector detectors and the first one that I thought I will talk about is actually a coronal detector.