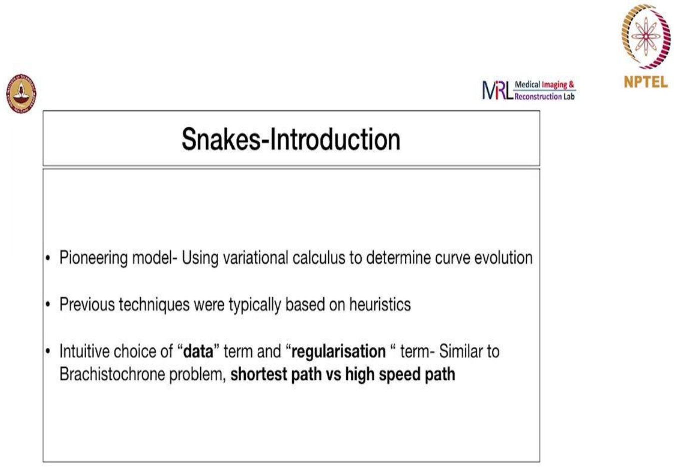
Medical Image Analysis
Professor Ganapathy Krishnamurthi
Department of Biomedical Engineering Design
Indian Institute of Technology Madras
Snakes - Active Contour Models

Hello and welcome back. So, in this video, we will look at the so-called active contour models or snakes. So, one of the, systematic kind of PDE based models to come out of calculus of variations, and one of the pioneering methods in that class of methods to pioneering work.

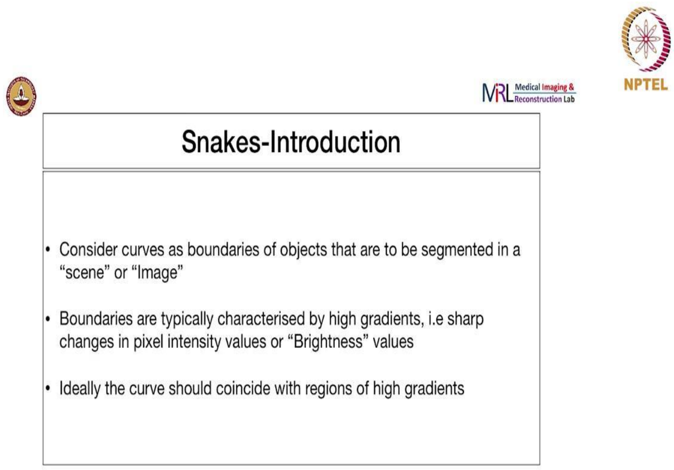
(Refer Slide Time: 0:37)



So, like I said before, it is a pioneering model using calculus variation calculus to determine curve evolution. So, why is this so important? Because, previous techniques or techniques that came before this for all typically based on heuristics, like a bunch of rules that you put together and hope that it worked. But in this one, this is of course used when I say this is used for segmentation. It actually the rules actually came out of variational calculus, the, by putting together a correct choice of, data terms, loss function and regularization terms, in the functional, we arrived at a curve propagation equation, without having a explicitly specify rules.

So, here, when you say data term and regularization term, how do you to think back about the Brachistochrone problem where we looked at, whether we go fast, by dropping vertically down a bit, so that time is reduced, or we reduce time by going that satisfy. So, that was a tension there, here, there is a tension between data and regularization terms, we will look at this in more detail in the next few slides.

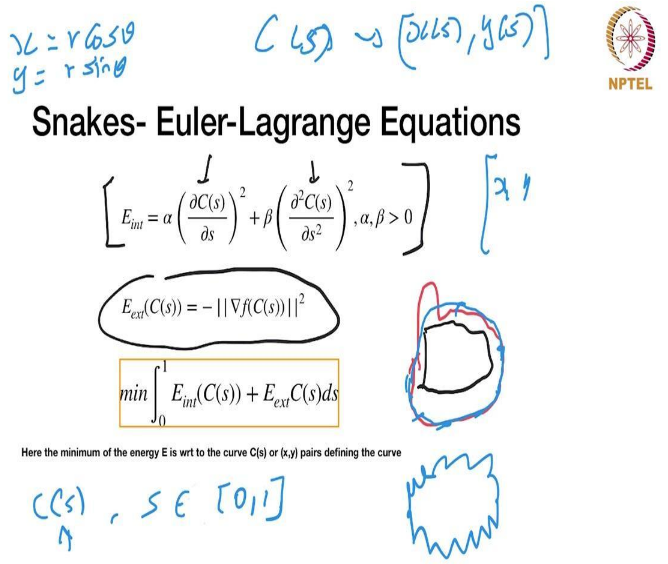
(Refer Slide Time: 1:49)



So, why are they called Snakes? So, if you consider any object, that is to be segmented, in this case, an organ to be segmented. In a scene or an image in the medical image, the boundaries are basically the curves, the boundaries of objects, or the curves. And that is the one that, so we seek to segment objects by propagating closed curves, so that they can go and latch on to the boundaries of objects. Now, what are the other characteristic of boundary's object? Typically, in a medical imaging setting, the boundaries are characterized by high gradients, the sharp changes in pixel intensity values or brightness values.

So, ideally, the curve that we are propagating that is we are trying to latch on to the boundary, we should also coincide with regions of high gradients. So, that is one of the important criteria to that is considered in this method.

(Refer Slide Time: 2:52)



So, we have, when we are looking at the various, terms that go into the functional. So, first we discussed that, the gradient is an important term. So, the gradient will tell you were sharp transitions in pixel density occurs, and we surmise that, wherever there are sub transition pixel that is this means that there is a change in the regions, that is the edge of the object that we are looking for.

So, we actually do not know the object, but our hypothesis is that, the edges of those objects correspond regions of high intensity, and we want our curves to go and rest, or latch onto regions of high density. So, that is one term. So, that is the term that you see here. Negative gradients square.

So, that means that, you have, you are looking for regions of, why it is negative? So, where the gradient is high, which means that it is a very high positive value, so you put in a negative sign there, so that it becomes a very small number. So, you minimize, so it becomes a very small number. So, you are trying to minimize our functional. So, one of the terms in the integrand is this red becomes a very small number when the red is very high.

The other term that we want very small, are they so that is, so here, this is where the tension is because we want them to latch on to the gradients, but we do not want them to be smooth, or they would not be smooth curves.

So, this is your regularization term, which makes your curve smooth. Of course, there are various explanations for this. But I will just give you a couple here. So, what this term does, the delta C or delta is I will tell you what these variables mean? This is our first derivative square this square there to make sure it is always, dealing with positive numbers, the square. So, the delta C or delta s make sure that you do not have very sharp discontinuities in your curve. And what this also ensure is that you have a very tight curve or at your, basically it have as small as possible.

So, it does not have sharp discontinuities and it is generally very tight here we say short as possible, the length of the curve is shortened, is the link shortening term? That is the first derivative, this second term, this just to make sure that it goes not too wiggly. So, do you do not want some. So, there if you have an object, let us say this object that you are trying to segment and your curve, you do not want some curves going like this. This is what we saw, you do not want a lengthy curve.

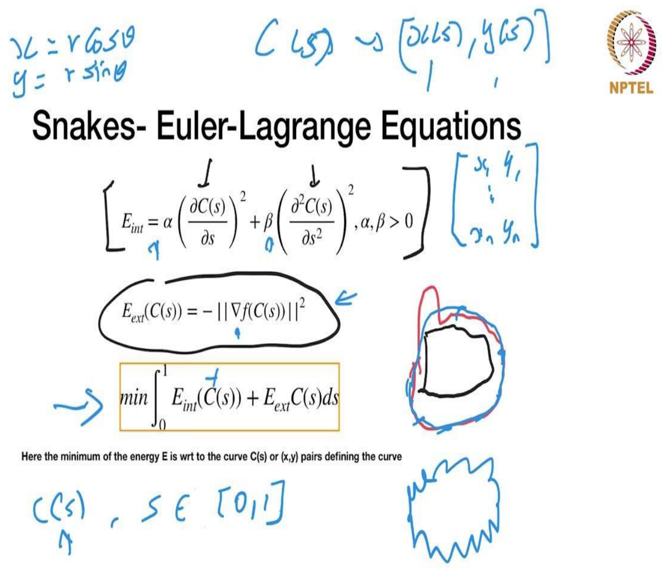
So, this is a very long curve. And this term makes sure that the sharp discontinuous do not occur. Rather, you have a smooth curve, fairly short curve. The second term, what it does is make sure that no, you can also have curves like this. So, you are abrupt changes in the first derivative. So, it has is what this happens is make sure that your curve is not super wiggly, it is more smooth. So, what are these C's the curve, obviously, just a notation, what is $c(S)$ represent? This is just some parameterization. So, S is a parameter that goes from 0 to 1. I am sorry, I am just writing it in a very poor way. Let me just rub this off.

So, S belongs to 0, 1. So, it varies S very smoothly from 0 to 1. So, it is a parameter which generates the curve. So, parameter is curve by S, $c(S)$ basically, it just means you can think of it as generating $x(S)$, $c(S)$ is nothing but a pair of points $x(S)$ and $y(S)$. So, x and y are basically points on the curve. So, you are think of curve it is in a plane, x and y are points on the curve. And you can generate the curve by varying as smooth as there are many parameterizations available, we would not go into that.

So, for instance, you can think of S maybe, $cos(\theta)$ and $sin(\theta)$, and for instance, x can be, let us say, $rcos(\theta)$, and y can be $rsin(\theta)$, this is one parameters you can put out r to be 1, so you get only $cos(\theta)$ $sin(\theta)$. And you can actually vary θ, so you can of course normalize it. So, that data varies from 0 to 1, and then you multiply accordingly. So, those things can be done. So, that is S, it is just one parameterization like that. So, lots of parameterization available, do not have to worry about that. Now, we will cross that bridge when you get there.

So, $c(S)$ is nothing but it is a vector of these points, x and y. It is a collection of x and y, which make up the curve. So, that is what we always want to say, C, we are only looking at this bunch of projects come away.

(Refer Slide Time: 7:52)



Once again, I will just rub this out and it is better. So, you have this points x comma y so you can generate them. So, you can x_1, y_1 up to x_n, y_n if you discretize it if you choose to discredit otherwise, it is a continuous parameterization. So, S smoothly varies from 0 to n so that you can generate all the x and y is possible. So, infinite number of x and y is possible, which make up the curve. So, there are these two terms, one is, like I said, the negative of the gradient squared, the gradient of F here, F refers to the image, here, we can use I, but typically, it is bad choice in this kind of work, F is refers to the image.

C of s here, it basically we are evaluating the gradient only at the curve. So, at all the points that make up the curve, we are evaluating F of C of S. So, the gradient so for the curve to be optimal. So, if you want to the curve to latch on to the object boundary, we stipulate that at the points wherever the curve is, you want a very high gradient which means that if you take the negative or the green, which means a very large, very large negative value, which means a very small number of them. So, the minimum that you are trying to minimize that.

So, large gradients and we want the curve to be smooth, so which means that is why we are adding the first and second derivatives, this is called regularization terms. So, these are the two competing forces, forces that were talked about, like the shortest path, high speed path, we are looking at, it has to latch onto strong gradients. At the same time, it should not be a too big a curve or too long a curve and also not be too wiggly curve. So, that is the tension here. And of course, we are trying to minimize this following functional, E internal is this energy, which corresponds to the regularization term.

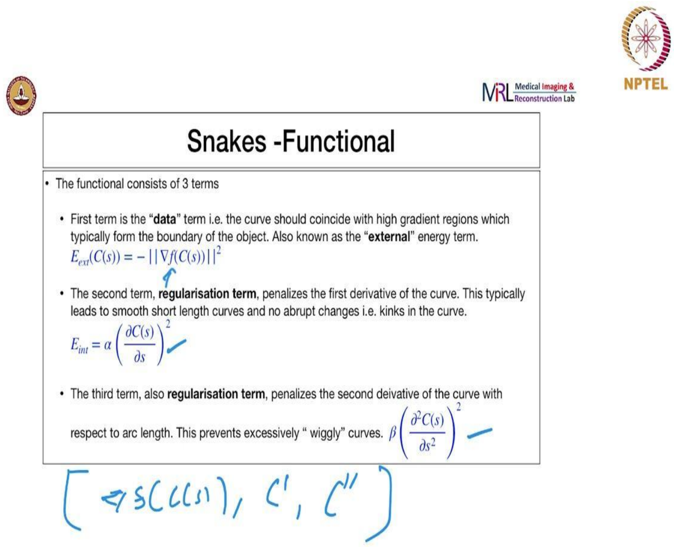
And this I call the data term because f is the image. That is the data that is given to you. And you are trying to estimate the gradient from the data. So, that is the data. So, I am here in the paper, they have referred to as an external term. E internal energy and the external energy, I like to call it regularization and the data term.

So, there is just a interplay between these two terms, typically, you will have these parameters alpha and beta. Now, alpha and beta can depend on S, also, or they can be independent of S either way is fine. And they are the terms that, gives you the ability to control how much importance you give to each of those terms.

So, by making the alpha and beta are small or 0, you can make their influence smaller or larger. So, that is the idea. So, that is the entire loss function. So, that you put them into the loss functional, you are trying to minimize the loss function with respect to what you are trying to estimate $c(S)$ like we had for the curve, in the Brachistochrone problem. Here, we are trying to estimate the segmenting curve C of S, which will form the boundary of object. So, once you have the boundary of object, we can find the interior also.

So, we are trying to estimate$c(S)$, or alternatively$c(S)$ is nothing but collection of these points x comma y, which make up the curve where estimator and decimate a bunch of x comma y, which form the boundary of the object by minimizing this function. So, and of course, remember, S, is the parameterization of the curve, S, varies continuously from 0 to 1. And as S varies continues from 0 to 1, you can generate x and y as you will.

(Refer Slide Time: 11:33)



So likewise, just to, get a better understanding of this snakes functional, consisting of three terms, like I said, the first term, the data term, the curve should coincides with high gradient regions, which typically form the boundary of the object. So, in medical imaging, also, in lots of situations, the boundary does form, the boundary does have high gradients, it is sometimes the paper they refer to as the external energy term.

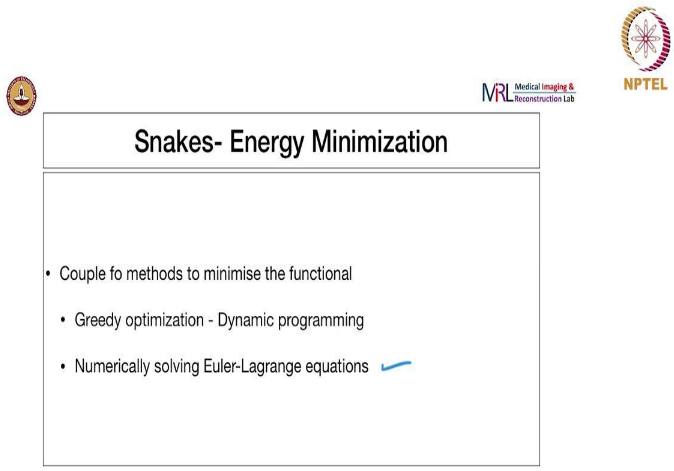
The second term is regularization term penalizes the first derivative. So, it means this typically leads to a smooth short length curve, there is no without kinks, suddenly there will be a good making. So, it is more short length curves. So, this is a curve short length curve term like, so it will just collapse the curve into a point, there is nothing stopping it.

The third term is also the regularization term is again penalizes the second derivative of the curve with respect to arc length. That is a arc length, but, is it S basically some parameterization. This prevents excessively wiggly curves. That is what back, sorry to that.

So, there are two terms, the first derivative term, second derivative term, so you have some function, if you will see that there is an energy term called energy term, there will be this f, gradient of f depends on $c(S)$, so it is a function of that. You also have the, it is, the integrand, inside the functional is has the C prime term, and the C double prime term. Here, the C is implicit in F.

So, these three terms are there. And so, then you have the form of the equation, to use to (()) (13:21). So, I had used the f of C S. Ideally, think of just do not use f, maybe you just use the z, this f is basically because I have used this f inside the functional when you are trying to derive the Euler-Lagrange equations. So then, that can be a bit confusing. So, feel free to interrupt if you some other symbol. So, that, you do not get confused. So, I am not.

(Refer Slide Time: 13:47)



So, now, sorry about this spelling mistakes, couple of methods, not couple fo methods. So, we are going to be only looking at the second one. This here, a numerically solving the Euler-Lagrange equations. So, we will derive Euler-Lagrange equations. Of course, you get it, I would not go through every step because all the steps are just, calculating derivatives. So, I asked you to do that. Once you know what the Euler-Lagrange equations are, you know what f is? And, there is again, like you saw here, that the curve propagation equations for this bear with me. Usually problem.

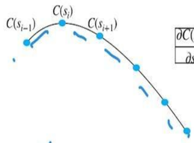
(Refer Slide Time: 14:25)

Snakes-Euler-Lagrange Equations

$$\boxed{-\alpha \frac{1}{\partial s}\left(\frac{\partial C(s)}{\partial s}\right) + \beta \frac{1}{\partial s^2}\left(\frac{\partial^2 C(s)}{\partial s^2}\right) + \left(\nabla E_{ext}\left(C(s)\right)\right) = 0}$$

$C(s), 0 < s < 1$ Deformable Contour holding pairs of coordinates x,y

The derivatives of the curves are approximated by finite differences and the curve is discretised using finite set of points



$$\frac{\partial C(s_i)}{\partial s} = \frac{C(s_i) - C(s_{i-1})}{h}$$

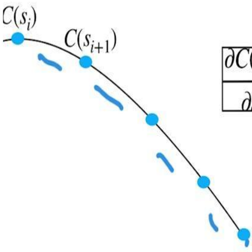
$$\frac{\partial^2 C(s_i)}{\partial s^2} = \frac{C(s_{i+1}) - 2C(s_i) + r(s_{i-1})}{h^2}$$

$$h = ||C(s_i) - C(s_{i-1})||$$

$$-\alpha \frac{1}{\partial s}\left(\frac{\partial C(s)}{\partial s}\right) + \beta \frac{1}{\partial s^2}\left(\frac{\partial^2 C(s)}{\partial s^2}\right) +$$

eformable Contour holding pairs of coordinates x,y

The derivatives of
finite differences a
finite set of points



$$\frac{\partial C(s_i)}{\partial s} = \frac{C(s_i) - C(s_{i-1})}{h}$$

$$\frac{\partial C(s)}{\partial s} + \beta \frac{1}{\partial s^2}\left(\frac{\partial^2 C(s)}{\partial s^2}\right) + \left(\nabla E_{ext}\left(C(s)\right)\right) =$$

ling pairs of coordinates x,y

The derivatives of the curves are approximat
finite differences and the curve is discretised
finite set of points

$$\frac{\partial C(s_i)}{\partial s} = \frac{C(s_i) - C(s_{i-1})}{h}$$

$$\frac{\partial^2 C(s_i)}{\partial s^2} = \frac{C(s_{i+1}) - 2C(s_i) + C(s_{i-1})}{h^2}$$

$$h = ||C(s_i) - C(s_{i-1})||$$

So, if we actually apply. So, remember, just to be clear, I think I am using this f too much. So, I am going to just switch notation a bit, because if you look at the, I am sorry, I have to go back and forth, but you have to try to do the derivation. So, when you actually formulate the Euler Lagrange equation, this is what you will get.

So, now, how are you going to solve this problem? So again, gradient of E external C of S, I am could have just using this term as it is gradient with respect to x and y, actually. That is what you have. So, and these terms will only with respect to S and delta S. And the second derivative with respect to delta S in fact second and fourth derivative, you know delta S and delta S square.

Ideally, this alpha should be inside. Because if it is a function of S, you can take that out, but assuming it is constant in this case, so what is it? Now, here to understand this, so the way to solve this, because you are solving there are no analytical solution. So, to solve it using numerical approximations, so I urge you to go and actually derive this, it is not you just have to plug in the correct function expressions in the Euler Lagrange equations that we saw in the previous video. And then you can get to this very quickly, it is not too hard to do.

So, how do we discretize this? So, the discretization scheme this what I have shown is, if this is the curve, the blue points are the discretization. So, S of I minus 1 there, you take, let us say capital N points were defined as capital N points. And in fact, you have to choose it, so that they are equal sense of a part typical, so it is ample. This length and this length are the same, even though in my picture, they are not on the scale, ideally, you choose those points N points on the curve for a district variation. So, that they are equally spaced. And of course, the first and last points coincide.

That is your boundary condition, the first and the last points coincide. And, the way you calculate the derivatives with respect to S any parameterization, is that you actually, they use your discretization itself. So, for instance, the first derivative with respect to if I can zoom it a bit, along with this a bit, it might get tilted a bit. So, the first derivative is calculated by, a finite difference approximation. So, the delta C or delta S at Si is nothing but $\partial C(s_i)/\partial s \; = \; (C(s_i) - C(s_{i-1}))/h$ here h is the distance between these two points that you have discretized.

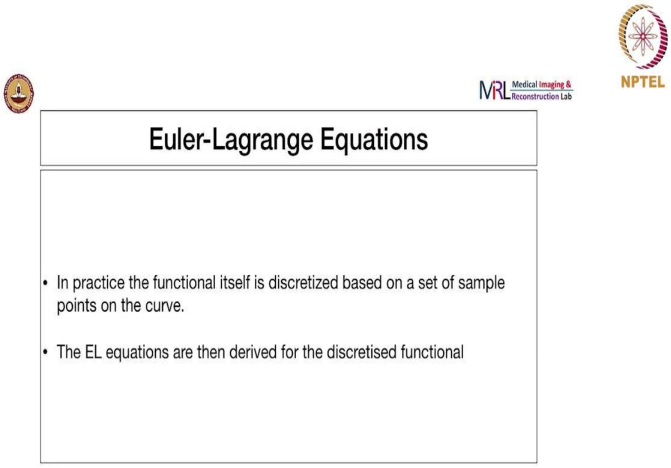
So, that is the first derivative. Similarly, you can, write down for a secondary derivative also. So, this again, if you are using numeric, if you do some numeric analysis, these are standard formulas. Similarly, for the second derivative of the curve at Si, we have this formula. Again, I have made a mistake here. It is not r it is C. Once again, I was referring to multiple textbooks and I was trying to use C because it seems to be more commonly used than r, somehow it slipped in there, sorry about that.

So, h is again the distance between successive points. Now, these are for the first and second derivative, what you do is you take this and you plug it back in this equation. So, now we are going to calculate delta over delta S of delta C or delta S, now you have delta C or delta S that formula here. That formula is there sorry, there is something I have done I need to take care of this. So, now, we have delta C or delta S the expression for this. Now, we have to do delta out. I have put one over delta S, but I should have put delta over delta S I am sorry, there seems to be again, a mistake here, we will correct that.

So, delta over delta S of here it is delta, delta S of delta C. So, now, all you do is not to do this, you instead of C of Si, you just put delta C over delta S, C of Si minus 1 you put delta C over delta S. Or you can directly use this formula better still directly use delta.

So, because alpha is not a functional base, you assume needs to be constant. We assume at constant we are directly plugged this in this equation for the second derivative again of the second derivative for every instead of C of Si plus 1, you have to substitute delta square C of delta S square at I plus 1 and delta square C of delta square I at S f I and delta square C by delta square at I minus 1. So again, I seem to change this. You put this in there, write down these expressions and you can simplify once again. I am not chosen to provide it we can look at it later time.

(Refer Slide Time: 19:50)



In practice, you can also discretize the functional itself so it is possible to do that straight away, the original expression, we had the functional expression, we can discretize that and then take the derivative then based on the discretized function.

(Refer Slide Time: 20:10)

Discretised Euler Lagrange Equations

$$\boxed{-\alpha\frac{1}{\partial s}\left(\frac{\partial C(s)}{\partial s}\right)+\beta\frac{1}{\partial s^2}\left(\frac{\partial^2 C(s)}{\partial s^2}\right)+\nabla E_{ext}\left(C(s)\right)=0}$$

$$\boxed{\frac{\partial^2 C(s_i)}{\partial s^2}=\frac{C(s_{i+1})-2C(s_i)+r(s_{i-1})}{h^2}} \qquad \boxed{\frac{\partial^2 C(s_i)}{\partial s^2}=\frac{C(s_{i+1})-2C(s_i)+r(s_{i-1})}{h^2}}$$

$$\left\{\begin{aligned} Ax+\frac{\partial E(C(s))}{\partial x}&=0\\ Ay+\frac{\partial E(C(s))}{\partial y}&=0\end{aligned}\right\}$$

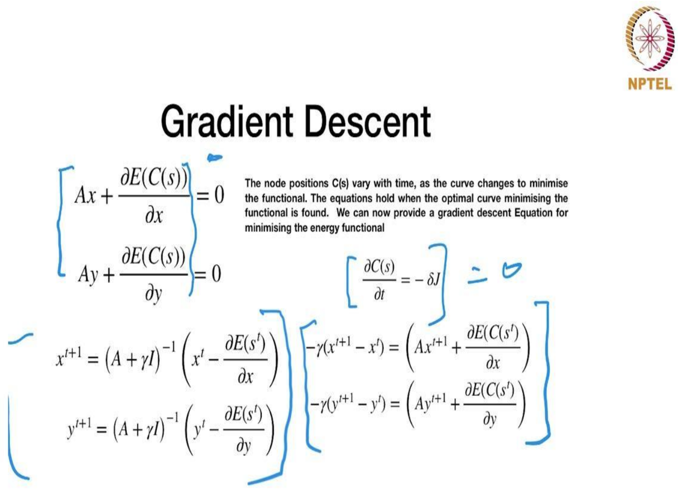
$$A=\begin{pmatrix} -2\alpha+6\beta & \alpha-4\beta & \beta & \cdots & 0\\ \alpha-4\beta & -2\alpha+6\beta & \alpha-4\beta & \cdots & \\ \beta & \alpha-4\beta & -2\alpha+6\beta & \cdots & \beta\\ \cdots & \cdots & \cdots & \cdots & \alpha-4\beta\\ 0 & 0 & 0 & \cdots & -2\alpha+6\beta \end{pmatrix}$$

So, if you do that, we end up with the following set of equations, you can actually derive them, it is very, not too hard to do, again, are you to do that. And if we find time, I will go through this in slightly more detail than what I am doing now.

So, you will end up with this set of two equations. Again, the E is basically the image term, it is the gradient of the image evaluated at the curve, A is Penta diagonal banded matrix, which basically carries the weights of the finite difference terms. And, these are the two in a simplified or the final form for the Euler Lagrange equations.

Now, that we have this, what do we do? So, it is not done yet. So, this just gives you some way of, some relationship between or like I said, the tension between how we move the points, because A, is the regularization matrix, you can think of it that way. A is the matrix that regularizes your curve, delta E, delta x is gives you the gradient information. So, just like this tension between the two, so how would you propagate the curve, that is what we are going to look at. So, but it is not done there at.

(Refer Slide Time: 21:18)



But what we have to do is something called gradient descent. So, because what we have learned sight remember that at the optimal position, of the curve, once a curve has latched on to the boundary of the object, then the curve no longer moves. So, which means that this delta of C over delta t equal to 0. But this delta C over delta t equal to 0 is, again, just that is how, we have defined the equilibrium position. So, which means that we are tracking the time curve over time. So, if you think about it, if you want to evolve the curve over time, then we can actually do this so called gradient descent.

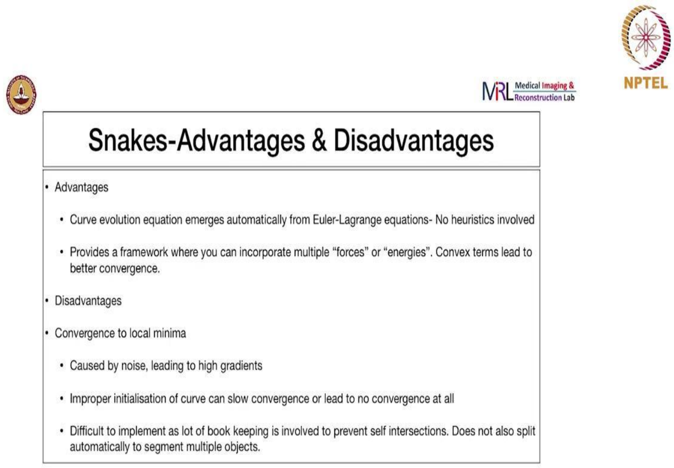
So, till we find the optimal position, we have to evolve the curve. How do we evolve the curve? We evolve the curve in the direction of increasing gradient, which is what the minus delta J tells you. That is how you propagate the curve. So, delta C, over delta S is minus delta J is just the gradient descent equation. So, you are propagating the curve in the direction of the opposite direction of increasing gradient, which is basically the gradient of the function. That is what we have, because we are trying to estimate the curve itself.

So, we have delta C, over delta t is minus delta J. And that is the delta J is what we estimated here. This is minus delta J, these terms, so then we can write this delta C over delta t as time stepping. So, this is again, you do a finite difference approximation. Like I said, C of S is nothing but pairs of vertices x comma y. So, then we can get this time stepping equation right here, which we can simplify to get the equation in this final form. So, A is a matrix that can be, there is again, a small confusion here as to why I put x of t plus 1 here, but I put t there?

So, this is called a semi-implicit formulation. Because, one way to look at it is, you propagate the curve, you move the curve based on a gradient equation at the current time step. And then you move it and once you move it and after you move it, then you regularize it, that is what we are doing. So, it is semi-implicit, it turns out, it has much more desirable numerical convergence properties. So, this is again, techniques borrowed out of numeric analysis. So, this is actually a very complicated method to actually understand and code.

So, you start with calculus of variations, you form the correct functional where there is a tension between how smooth your curve has to be regularized your curve has to be and, how badly you want it to latch on to the gradients in the image, the curve to latch on the gradients in the image. And then from there will be Euler-Lagrange equations come out, but they are non-trivial to solve. So, you solve them by using the semi-implicit numerical analysis technique. By and of course, you are discretize these things properly.

(Refer Slide Time: 24:22)



So, what are the advantages of this? So, the curve evolution equations emerge automatically from, you did not do anything other than intuitively decide this, you intuitively decided I want a smooth curve, and I want it to be attracted to gradients. That is the condition you imposed. And the rules for moving the curve around. The mathematics for moving the curve around came out automatically. Of course, it is very complicated, but it still does came up. And so, and think about this, actually, this actually provides a framework where you can incorporate multiple forces or other energies.

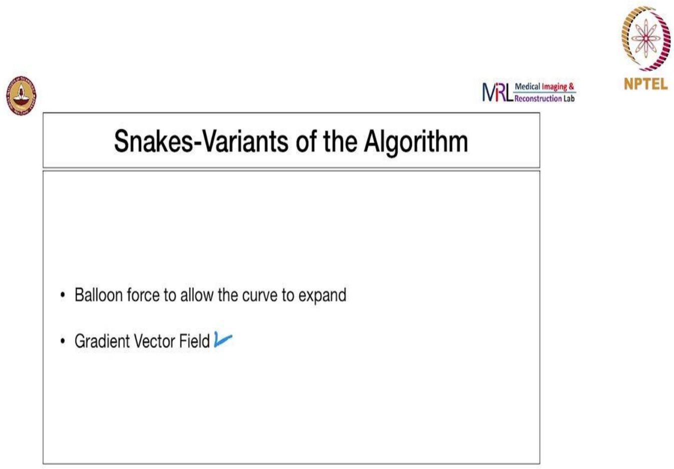
And of course, we use convex terms, which basically something that has only, a global minimum. But again, we would not get into that. So, you add terms which are amenable to, easy to differentiate and have no do not have too many local minima. So, you can like add more forces to make it move faster, et cetera, we will see a couple of them not in detail, as mentioned them, they call balloon forces, and there is something called a gradient vector field. What is the disadvantage, it converges to local minima?

If you take any image to think about it, a lot of images will have noise and noises are typically high gradients. So, which means that, curve can latch on to a noisy pixel or a group of noisy pixel, which might not correspond to the object of interest in the image. And improper initialization can cause slow convergence.

So, typically, this method is kind of sensitive initialization, you pretty much have to start very near to the boundary that you want to segment, you might say, that is no fun out a why you will want to do that is better than drawing by hand? If you have a large number of images to annotate, it is best that you start varying, you initialize it, and it automatically converges to a boundary.

Now, it is difficult to code because you will have to keep track of all the points, that is a double edged sword, because, you do not have to keep track of the, for an entire image, you are trying to segment the entire image, you have to keep track of only like, a finite set of points, you do not have to keep track of every pixel, keep track of every pixel. How we are keeping track of every each of these points? Because you are you are propagating them, it is hard when you do not want them to converge to one point, you do not want them intersecting all these are much more difficult programmatically to enforce.

(Refer Slide Time: 26:36)



So, there are variants of this, I will not go into the details, if I find time I will, there is something called balloon force, remember, the first derivative term makes the curve shrink, but sometimes, you want them to blow, to expand a bit based on, certain image statistics. So, that is possible that has been done, that is called a balloon force, which moves curve move in any direction, expand and compress. The gradient vector field also remember see sometimes, image is pretty smooth, contains maybe one or two objects, but away from the objects, there is no strong gradient.

So, if initially, slightly further away from the boundaries of the object, your curve will never converge, so it will just keep moving very slowly. Of course, if you continue to do time stepping for thousands of time steps, then maybe it converge step very slow and not practical. So, something called a gradient vector field, wherein you think of it as smoothing the gradients from the edges away from it, you have a very strong edge, but then you can maybe smooth it a bit. So, that the gradient values are kind of dispersed around the object a little bit farther than the boundary on either side.

So, that lets the constraints convert smoothly, again, this is accomplished this is non-trivial to accomplish, indeed, you have to solve one more, functionality, if you have to derive one more functional, and you actually formulate the Euler-Lagrange there, and then you end and you get a gradient vector field, which you use for propagating. So, this is kind of, two step, kind of think of it like a two step problem. But again, this is very popular, and people have used it very effectively.

Because for a long time, these methods were, there was only one which can be used without too many heuristics. Because, for every image, you cannot make heuristics every modality you cannot make up heuristics. This comes from very simple sound principles. So, thus, this concludes our overview of snakes. Like I said, some of the derivations I have skipped, showed you the general principle.

So, you can we know, plugging into the, Euler-Lagrange equations actually taking the derivatives, it is not too hard, if you are done plus two mathematically should be able to do. Only thing the only tricks there will be there is a discretization, et cetera, you have to actually work out it again, that is also not too hard. Some of these discretizes, like the only two formulas you need to know are the ones I showed you, and that you should be able to do.

So, we will end this class here. We look at, further development on this, because like I said, the disadvantage is keeping track of this snakes, it is hard problem. So, how do we get over that? That is where we run into level sets. So, we will talk about that in the next lecture.