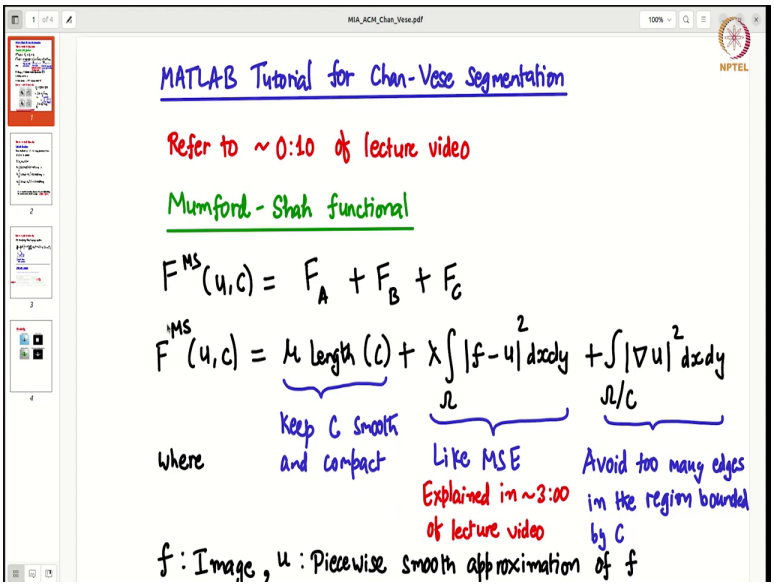
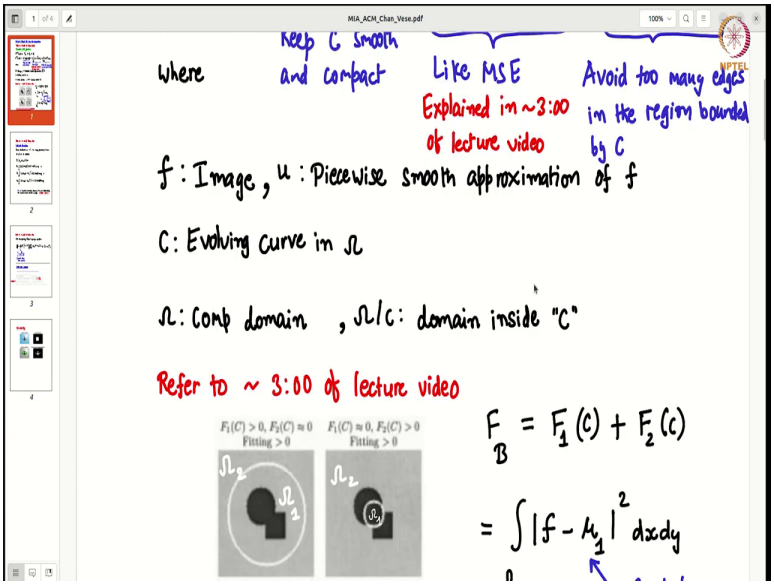
Medical Image Analysis
Professor Ganapathy Krishnamurthi
Biomedical Engineering Design
Indian Institute of Technology, Madras

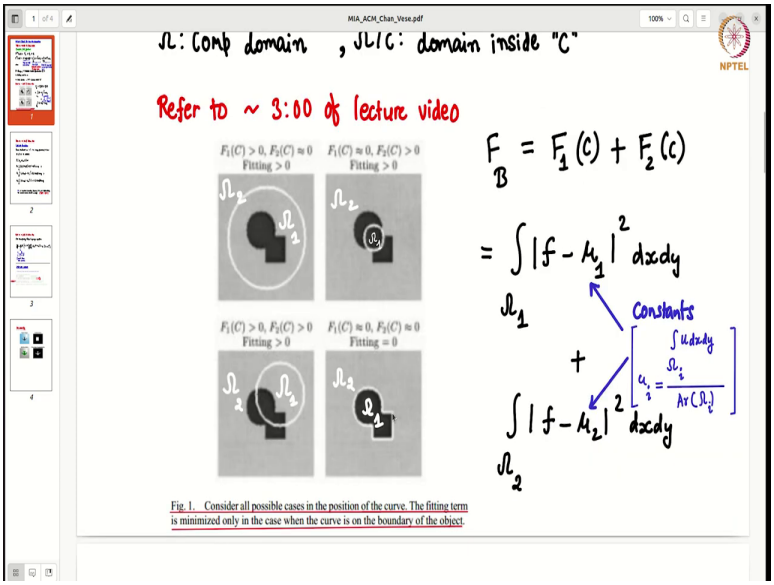
Lecture 36

Chan Vese Segmentation

(Refer Slide Time: 00:14)







In this tutorial, we will see the MATLAB implementation of Chan-Vese Segmentation Model. So, this model is different from a Snake and the Geodesic Active Contour Model because of its, because of its functional. So, the overall theme, both in the explicit method or the Snake kind of method as well as the level set method that we used for Geodesic Active Contour is this.

You start with some functional, then Euler–Lagrange equation, then we find a PDE and then we try to solve this PDE with some numerical method. So, the functional used for Can-Vese segmentation is different from the other 2. So, that is why we have, we are treating it separately. So, the functional is called Mumford-Shah functional.

Its expression is shown on your screen. It has three terms, F_A, F_B, F_C. F_A, and here some notations are there. C is same as the evolving curve, that we are using all the time. Here, small F represents image, u represent its piecewise smooth approximation, Ω represents the complete computational domain, and Ω/C represents the domain within C. So, with this notation, you can understand the functional here.

The first term penalizes for large length for a, large length of the contour. So, its objective is to keep C smooth and compact. F_C, the third term, it penalizes for the presence of gradient, gradients within, within the region bounded by the C. So, basically it, its main objective is to avoid too many edges in the region bounded by the C, by the curves.

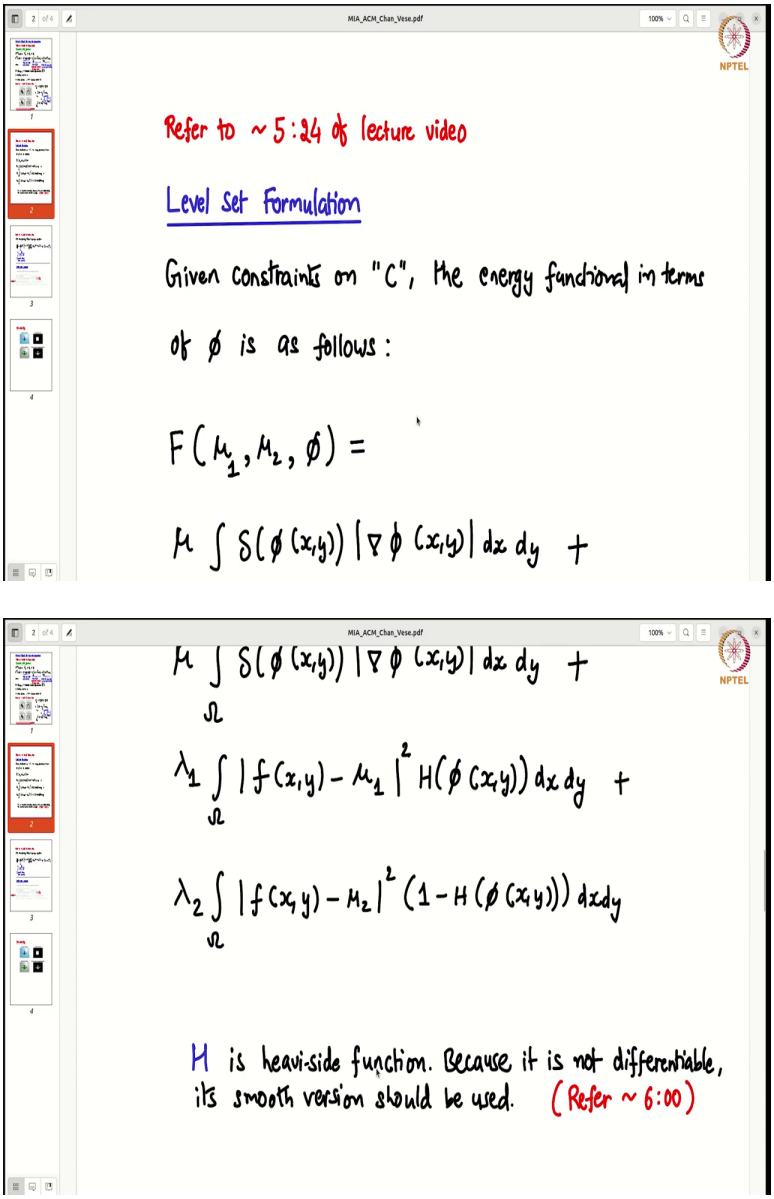
The second term, F_B term is the main term, but it looks like a mean square error. Mean square error between what, the original image and its piecewise smooth approximation. So, sir has explained this in this timestamp, almost 3 minutes in the lecture video, you can refer to the video. I am taking a simple example to just explain what he taught there. So, for example if you have an image like this, so, we have an object here, this black, this, which is shown in this black color and we want to segment it from its surrounding, which is in white color.

So, this white line is the curve C. So, within C, we have Ω_1, and outside it, we have Ω_2. So, we consider all the possible cases for the position of curve. So, and also we define our $F_B = F_1(c) + F_2(c)$ because here it is over all the Ω, not just inside. So, $F_1(c)$ is mean, calculate the mean square error within $\Omega_1$and $F_2(c)$ calculates mean square, mean square error within Ω_2.

Here, u in this function is the area integral, basically, average of image intensity in the region, so, this is μ_1, μ_2. So, here you can see, very intuitively, in the first figure, in Ω_2, all values have same, all the pixels have same value. So, the value is same as the average. That is why $F_2(c)$ is almost equal to 0. But here, in Ω_1, this will not be, this will not be the case because the average will be different from the individual pixels.

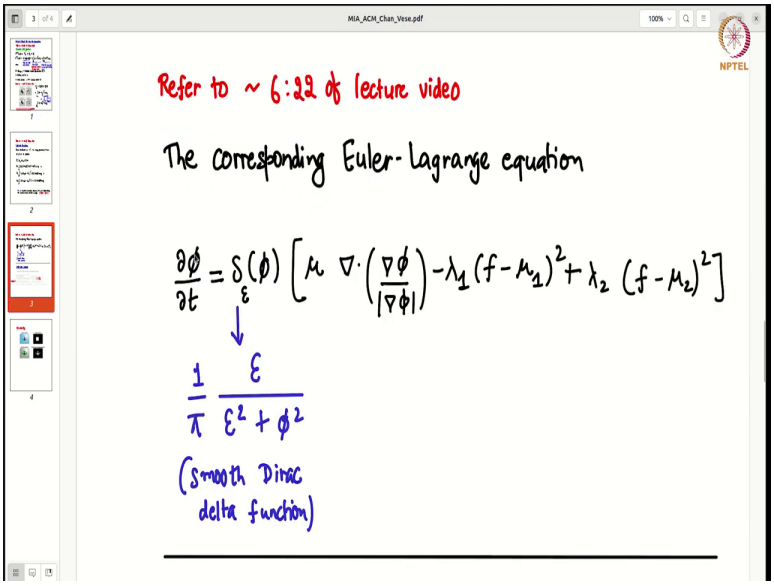
In the second case, Ω_1, Ω_1 1 is like, it contains only black pixels. So, that is why $F_1(c)$ is close to 0 but $F_2(c)$ will not be 0. So, you can see it is, the fitting term is minimized only in the case when the curve is on the boundary of the object. So, that is the purpose of this, this functional term. So, that is the main, main hallmark of this Chan-Vese segmentation model, this Mumford-Shah functional.

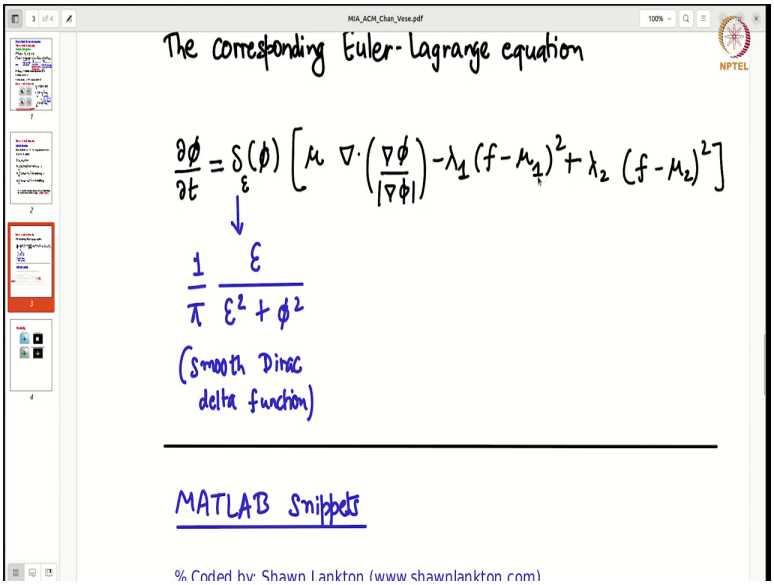
(Refer Slide Time: 05:27)

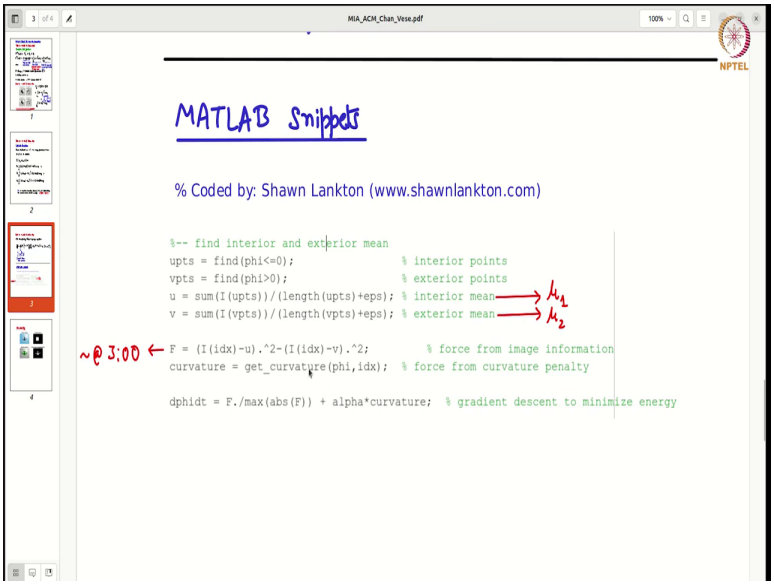


So, this is also a level set formulation method. Although the function is different, but still it is, it has a level set formulation. So, what happens in level set formulation is given constraints on C, we define an energy function in terms of phi. In case of Chan-Vese model, this functional F is this, with the representation which we have already seen. This F is actual image, mu 1 is what, the average and all. Here, capital H is the heavi-side function. And because it is not differentiable, it is a smooth function, a smooth version is used.

(Refer Slide Time: 06:06)







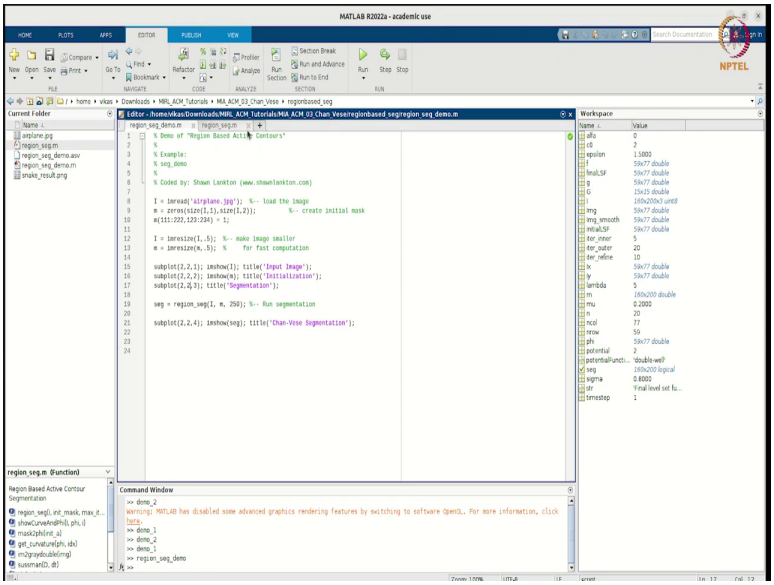
And when so, now we have a functional in terms of phi, not in terms of C, please note that. So, when we use our Euler–Lagrange equation, we find this, this PDE, del phi by del t is equal to this. Here, delta is the smoothened direct delta function. So, this is the PD that we solve. And we are not going to explain you the discretization and how, how we are going to do it. You have already seen how we calculate gradient and all, and how we take square. This, you can also do.

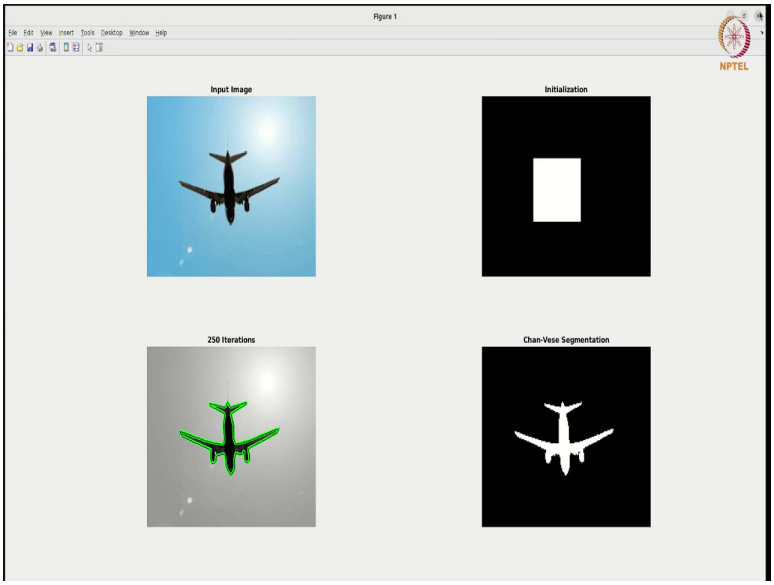
We are providing you the code directly and the code that we are giving to you contains some other terms also, I think, as usual. So, we just want to tell you, I just want to show you that whatever is taught to you in the class is sitting in the code as well. For example, in this snippet you can see, first is identifying Ω_1 and Ω_2. What is Ω_1, what is Ω_2, what is μ_1, μ_2.

So, here you can see, we dealt with phi. First, they found the points within Ω_1, these are the interior points. Then, the exterior points, points in Ω_2. Then we find the value of means. So, this is μ_1 this is μ_2. And then we calculate this force I minus u squared, I minus v squared. So, this is one penalty.

And then also, we have something for curvature that was not taught in the class that they have used. So, you can study the code and identify it, learn it and then we are solving some equation, d phi by d t.

(Refer Slide Time: 08:10)





So, an example implementation of Chan-Vese segmentation is here. So, you start with an input image of this aircraft, and then this is your initialization. After some 200 iterations, 250 iterations, it will completely segment the airplane. So, this is one demo. You can download other codes also, you can understand them, play with them, understand them.

Yeah so, that is all for Chan-Vese segmentation. We hope that all the 3 methods, you were able to understand the similarities and differences between the 3 methods with these implementations. And also because you have the codes for all the 3, you can play with it, you can implement your own terms.

So, you can like upgrade the code by adding more terms, or you can also simplify the code by putting some of the weights equal to zero. And we hope that with the code that will be provided to you and the inside, and the basic idea of the code you will be able to understand these active contour models better. So, thanks for your time, hope this tutorial helps.