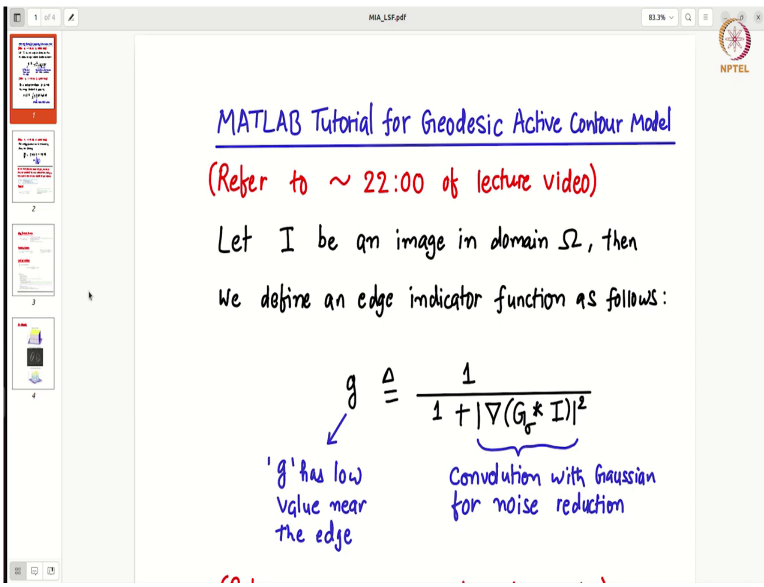
Medical Image Analysis
Professor Ganapathy Krishnamurthi
Biomedical Engineering Design
Indian Institute of Technology, Madras

Lecture 35

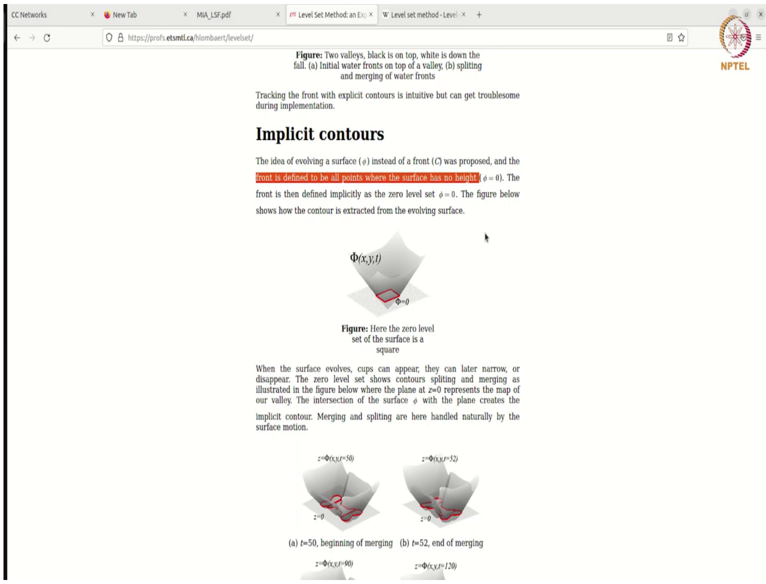
Level Set Method

(Refer Slide Time: 00:15)



Welcome to the second tutorial of this Active Control Model. Here, we will cover Geodesic Active Control Model, which is based on level set formulation. It is different from what we have just covered, the Snake model. So, how is it different? So, in the Snake model, if we start with the initial snake, initial, an initial curve, for example it is C, the curve was C, it was parameterized by S, and then we had an explicit evolution equation of C, like del C by del t is equal to something. But in level set formulation, we do not have an, we do not have a such explicit equation for C. Instead, we have something else.

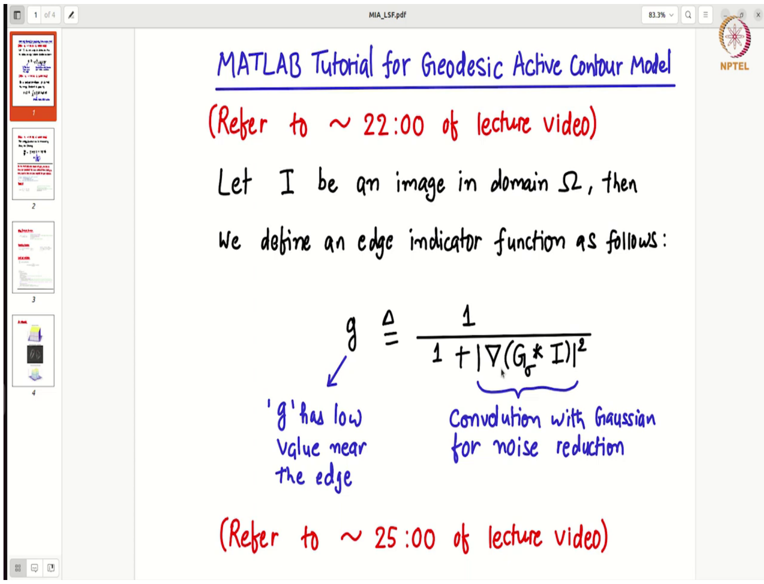
(Refer Slide Time: 01:12)

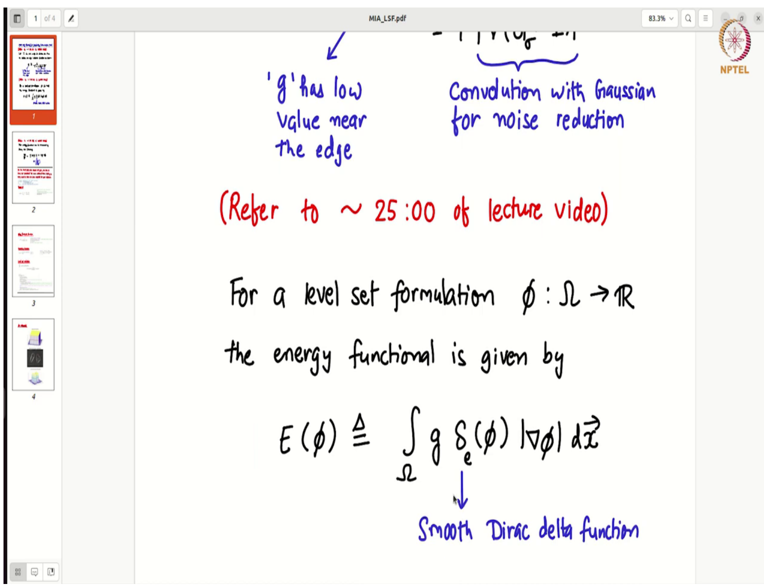


So, what is this? So, you can refer to this page. So, the idea of… so basically these are implicit contours, so this is phi. So, what is this phi? So, instead of evolving C we evolve a surface which is represented by phi. And this, and with the help of phi, we can calculate, we can estimate the location of C in an implicit way. How? It is, as you can read here, the idea of evolving a surface instead of front, C, was proposed. And the front is defined to be all points.

So, the front is defined to be all points where the surface has no height. So, phi, you can also think of as a height function, that we are going to evolve. So, this is the main difference between the 2. For more details please refer to the lecture videos, my explanation is not very good but you can refer to a lecture video. I have added the time stamps here.

(Refer Slide Time: 02:06)





NPTEL

(Refer to ~ 28:00 of lecture video)

This energy functional can be minimized by solving the following

$$\frac{\partial \phi}{\partial t} = g\,|\nabla \phi|\,\kappa + \nabla g \cdot \nabla \phi$$

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right)$$

In the MATLAB code shared with you, the idea is same but functional has some additional terms. Nothing to worry about it. Here are some snapshots for your reference.

2 of 4 MIA_LSF.pdf 83.3%

NPTEL

$$\frac{\partial \phi}{\partial t} = g\,|\nabla \phi|\,\kappa + \nabla g \cdot \nabla \phi$$

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right)$$

In the MATLAB code shared with you, the idea is same but functional has some additional terms. Nothing to worry about it. Here are some snapshots for your reference.

```
% Author: Chunming Li, all rights reserved
% E-mail: lchunming@gmail.com
%         li_chunming@hotmail.com
% URL:  http://www.imagecomputing.org/~cmli/
```

Initial ϕ

$$\phi_0(\mathbf{x}) = \begin{cases} -c_0, & \text{if } \mathbf{x} \in R_0 \\ \ldots \end{cases}$$

```
% initialize LSF as binary step function
c0=2;
initialLSF=c0*ones(size(Img));
% generate the initial region R0 as a rectangle
initialLSF(10:55, 10:75)=-c0;
```

NPTEL

Edge Indicator function

$$g \triangleq \frac{1}{1+|\nabla G_\sigma * I|^2}$$

```
G=fspecial('gaussian',15,sigma);
Img_smooth=conv2(Img,G,'same');  % smooth image by Gaussiin convolution
[Ix,Iy]=gradient(Img_smooth);
f=Ix.^2+Iy.^2;
g=1./(1+f);  % edge indicator function.
```

Smoothing function

$$\delta_\varepsilon(x) = \begin{cases} \frac{1}{2\varepsilon}\left[1+\cos\left(\frac{\pi x}{\varepsilon}\right)\right], & |x| \le \varepsilon \\ 0, & |x| > \varepsilon \end{cases}$$

```
function f = Dirac(x, sigma)
f=(1/2/sigma)*(1+cos(pi*x/sigma));
b = (x<=sigma) & (x>=-sigma);
f = f.*b;
```

Level set evolution

$$\frac{\partial \phi}{\partial t} = \mu \mathrm{div}\left(d_p(|\nabla \phi|)\nabla \phi\right)$$

3 of 4 MIA_LSF.pdf 83.3%

NPTEL

Level set evolution

$$\frac{\partial \phi}{\partial t} = \mu \mathrm{div}\left(d_p(|\nabla \phi|)\nabla \phi\right) + \lambda \delta_\varepsilon(\phi)\mathrm{div}\left(g\frac{\nabla \phi}{|\nabla \phi|}\right) + \alpha g \delta_\varepsilon(\phi)$$

```
phi=phi_0;
[vx, vy]=gradient(g);
for k=1:iter
    phi=NeumannBoundCond(phi);
    [phi_x,phi_y]=gradient(phi);
    s=sqrt(phi_x.^2 + phi_y.^2);
    smallNumber=1e-10;
    Nx=phi_x./(s+smallNumber); % add a small positive number to avoid division by zero
    Ny=phi_y./(s+smallNumber);
    curvature=div(Nx,Ny);
    if strcmp(potentialFunction,'single-well')
        distRegTerm = 4*del2(phi)-curvature;  % compute distance regularization term in equation (13) with the single-well 
    elseif strcmp(potentialFunction,'double-well');
        distRegTerm=distReg_p2(phi);  % compute the distance regularization term in eqaution (13) with the double-well pote
    else
        disp('Error: Wrong choice of potential function. Please input the string "single-well" or "double-well" in the drls
    end
    diracPhi=Dirac(phi,epsilon);
    areaTerm=diracPhi.*g; % balloon/pressure force
    edgeTerm=diracPhi.*(vx.*Nx+vy.*Ny) + diracPhi.*g.*curvature;
    phi=phi + timestep*(mu*distRegTerm + lambda*edgeTerm + alfa*areaTerm);
end
```

So, although there is little difference in what we are going to evolve, whether it is a contour or the height function, the main theme of all these models is this. You start with a functional, and functional contains some energy terms, with that you calculate some forces also. And then use all this Euler-Lagrange equation and find some partial differential equation that you solve numerically. And then you find the… and then you do the segmentation.

So in this method, an edge indicator function is first defined because it is used in the energy functional. So here, the edge indicator function is

$$\frac{1}{1+|\nabla(G_{\sigma}*I)|^2}$$

So, you would have, you have already seen this thing earlier. Given an image, we can, we use this Gaussian blurring to widen the well, widen the potential energy well.

So, that is one thing. And second, and then, this indicator function as you can see, when gradient is high, gradient is high, its value becomes low. So, it becomes less. So, this is the purpose of it. So, this G is used in the energy functional here. Then we have some direct delta function also that we have to smooth because we want our function to be differentiable at all times. So we use some smooth dirac delta function.

And finally, we have this equation. So we are not going to the details of how we got here and even what these terms mean, but you can refer to this time stamp of lecture video to understand what this equation means. So, in the MATLAB code that we will share with you, the idea is same but the functional that has been used has some additional terms also. So, you need not to worry about it.

Even though the code is little bit advanced, but you will find that the ideas that we have seen in the lecture videos are sitting there. This is the main purpose of this tutorial. So that Snake one was a more detailed tutorial but here, we are just trying to show you that whatever you have learnt in this lecture video, it is sitting there even in the advanced codes.

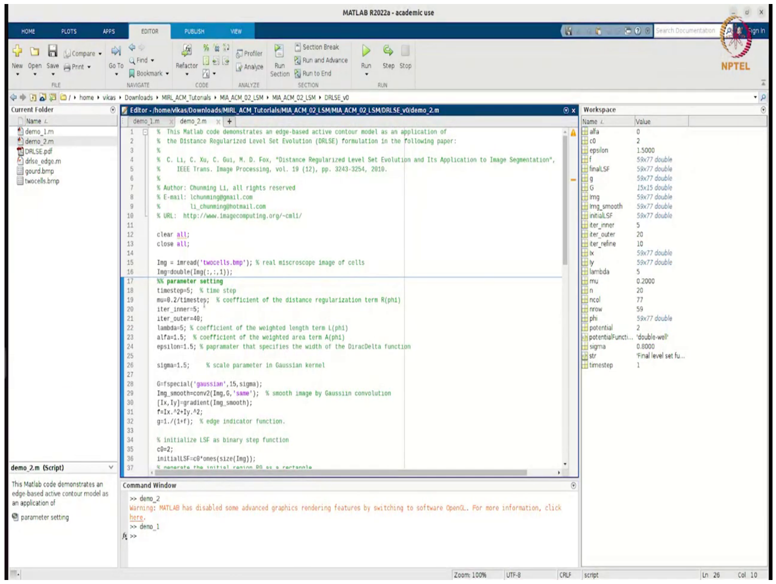
So, for example, in this code, you will see we start with some initial phi, for example this, a binary step function. Then we have the same indicator function that we were, that you, that sir taught in the class, $\frac{1}{1+|\nabla(G_\sigma * I)|^2}$. So this is how you implement it in MATLAB. So I am just attaching some snapshots of that code.

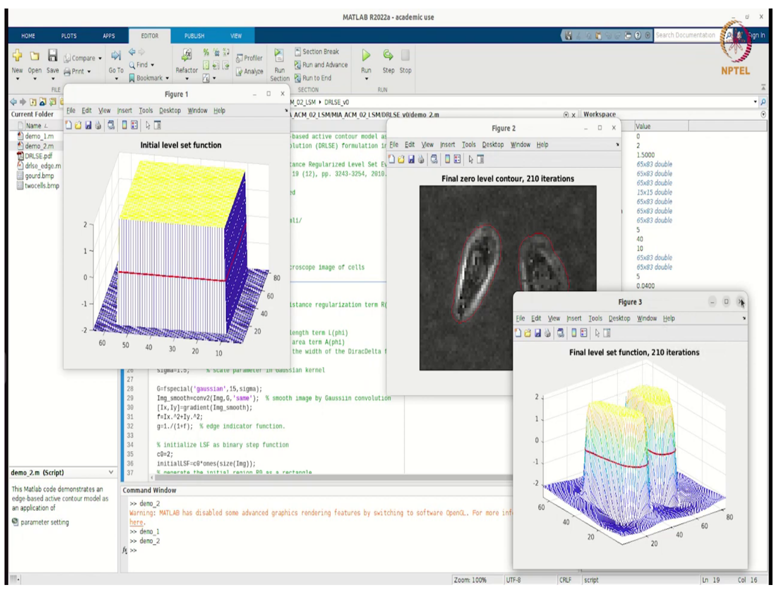
So here you have initial phi then you have edge indicator function, then this is smoothing function also. So, here, they have used some other function. You can try, you can also try the one that sir taught in the class, it is up to you. So, but the idea is smoothing function is used there.

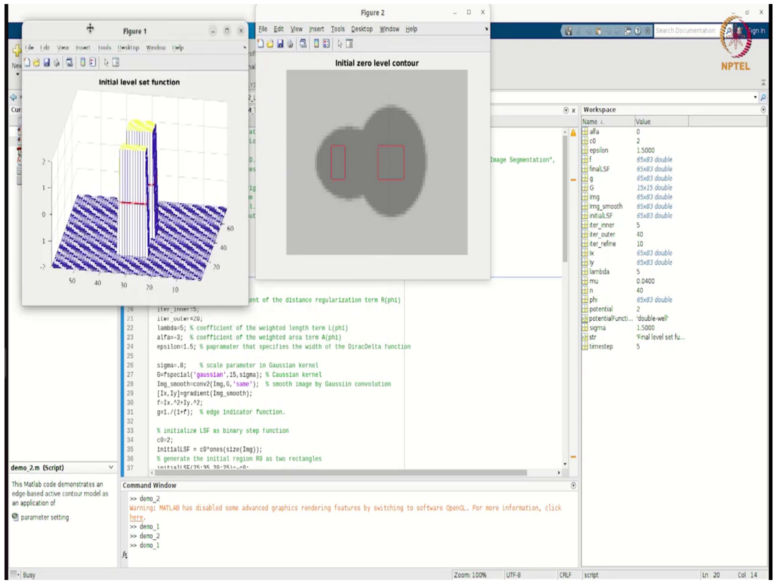
And this level set evolution equation you see, here you have some additional terms also. And their weights are mu, lambda and alpha. So, if you want to like shut down some of the terms, you can just set alpha equal to 0 or mu equal to 0 and maybe, so, you understood. So, then you can like, and this is the evolution, it is the solution numerical solution for this.

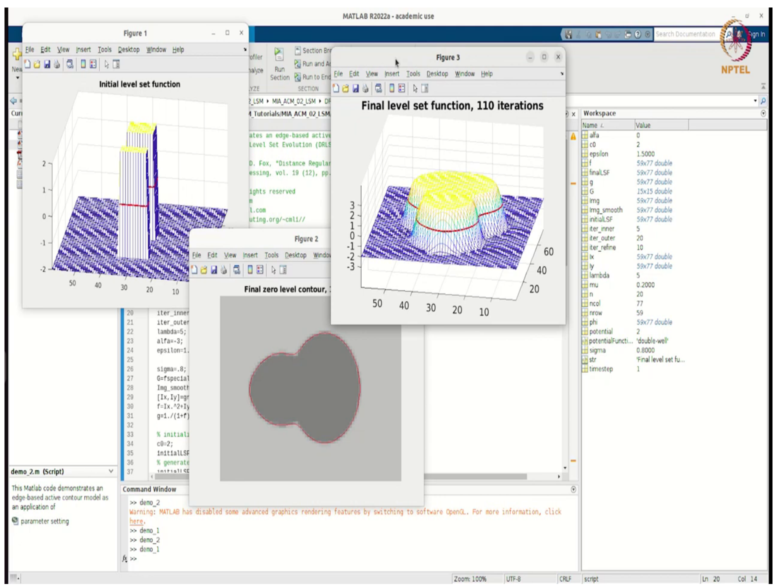
So the idea is, whatever you have been taught is already present in this code also. You can play with it, you can add your own terms, remove some of them, and understand the code more. So, for example, I will show some demo here. So, hopefully, you would have seen this.

(Refer Slide Time: 06:38)









So, you start with some initial, initial function, level set function. In this case, it is binary step. And after some time, it evolves. So after some 200 iterations, this binary function is evolved to this shape. And now, we are able to capture these 2 also. These 2 objects are very nicely segmented by our level set formulation. So, that is first demo.

You can, there is another demo that is available. So initially you have a level set function of this type and after some time it evolves and it captures the boundary of the object. And this is the final shape of the level set function. So the main idea is, you can see, we are evolving our level set function, and therefore, we are able to see the evolution of contour as well, in an implicit way.

And the main thing is whatever thing, whatever ideas that sir has conveyed during the lecture video, that are present in this code as well, and some additional ideas also. So, you can take a look at this code, play with it, and understand it more. So, that is all for this tutorial. Thanks for your time.