Low Level Statistical Models for Initialization of Interactive 2D/3D Segmentation Algorithms

Jan Kolomazník, Jan Horáček and Josef Pelikán

Department of Software and Computer Science Education, Faculty of Mathematics and Physics, Charles University in Prague, Prague, Czech Republic

Keywords: Statistical Models, Interactive Segmentation, Active Contours, Graph-cuts.

Abstract: In this paper we present two models which are suitable for interactive segmentation algorithms to decrease amount of user work. Models are used during initialization step and do not increase complexity of segmentation algorithms. Model describe spatial distribution of image values and classification as either foreground or background. Second part of the model is vector field which constrains direction of boundary normals. We show how to use these models in parametric snakes/surfaces framework and minimal graph-cut based segmentation.

1 INTRODUCTION

Image segmentation is one of the most essential problems in the field of computer vision. Although this topic has been extensively studied, common segmentation algorithms often serve as a preprocessing method of other algorithms. Automatic segmentation can hardly obtain satisfied results without high level knowledge of interest object (Heimann and Meinzer, 2009; Cremers et al., 2007; Leventon et al., 2002; Yue and Tagare, 2009).

In medical imaging is often the situation complicated by the fact that we often need to segment organ affected by certain pathologies (tumors, deformations, scar tissue), which are hard to model due to their unpredictability.

Also, methods suitable for automatic segmentation tend to be time consuming especially when dealing with low quality input (low resolution, noise, scanning artifacts, etc.).

Besides automatic and semi-automatic methods variety of fast interactive methods (Zhao and Xie, 2013) were published. User provides information about segmented object in form of constrains, seeds or similar mechanism. By evaluation of the segmentation output user can improve the result by updating input interactively. Session ends when object is segmented with desired precision.

Main disadvantage of the interactive segmentation algorithm is the amount of work user must do to get satisfying results, particularly for 3D volume segmentation.

We tried to address these issues by designing low level statistical models, which can provide most of the required input information and user can focus on the most problematic (blurred, damaged) parts of the segmented object.

We call these models low level because they do not describe complex properties and relations to surrounding objects like other sophisticated but slow methods (Tsai et al., 2003), (Okada et al., 2008).

We chose presented models with these properties in mind:

- Easily embeddable into various segmentation algorithms.
- Low computational complexity.
- Applicable during preprocessing step.

We ended up using two models. One describing spatial distribution of intensity values and the other directions of possible contour normals.

These models were embedded and tested in two segmentation frameworks. As energy minimization problem in parametric snakes/surfaces (Jacob et al., 2001) and in minimal graph-cut based segmentation (Boykov and Jolly, 2001; Yi and Moon, 2012; Kolomaznik et al., 2012).
2 INTENSITY DISTRIBUTION

2.1 Formulation

We model spatial intensity distribution of segmented object (foreground) and its surroundings (background). Model provides two probabilities for pixel/voxel on position \( x \) of intensity \( I(x) \) being inside the segmented region \( (R_{in}) \) resp. outside the segmented region \( (R_{out}) \).

\[
P_{in}(I(x), x) = P(I(x) \land x \in R_{in})
\]

\[
P_{out}(I(x), x) = P(I(x) \land x \in R_{out})
\]  

(1)

In special case when foreground/background intensities are independent of position (homogeneous object on homogeneous background) we can use basic properties of conditional probability and get these equations:

\[
P(I(x) \land x \in R_{in}) = P(I(x)) \cdot P(x \in R_{in})
\]

\[
P(I(x) \land x \in R_{out}) = P(I(x)) \cdot P(x \in R_{out})
\]  

(2)

In (Jacob et al., 2001) authors used energy term \( E_{original} \)

\[
E_{original} = \int_{S_{in}} -\log \left( \frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) dx
\]  

(3)

It relates probability of point being inside segmented region \( (R_{in}) \) having some intensity value and probability of being outside \( (R_{out}) \) with actual intensity value. This energy term reaches its minimum when regions \( S_{in} \) (intermediate segmentation result) and \( R_{in} \) are the same. It works well if we are trying to select object of homogeneous intensity on homogeneous background. Segmentation driven by this energy often fail if there is an area with similar intensity as the segmented object, because there are no spatial constrains (if not introduced in another way).

We decided to extend this model by spatial information and use 3-dimensional (2D segmentation) or 4-dimensional (3D segmentation) probability distribution function instead of 1-dimensional from the original term. So we propose formulation (4).

\[
E_{region} = \int_{S_{in}} -\log \left( \frac{P_{in}(I(x), x)}{P_{out}(I(x), x)} \right) dx
\]  

(4)

\[
E_{region} = \int_{S_{in}} -\log \left( \frac{P(I(x) \land x \in R_{in})}{P(I(x) \land x \in R_{out})} \right) dx
\]  

(5)

In cases when we can assume independence of intensity and position (homogeneous regions), we can use equations 2 Using this and the fact that logarithm of a product is sum of logarithms we get simplified version of the energy term.

\[
E_{region2} = -\int_{S_{in}} \log \left( \frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) dx + \log \left( \frac{P(x \in R_{in})}{P(x \in R_{out})} \right) dx
\]

(6)

When intensity and position are independent \( E_{region2} \) equals \( E_{region} \). First term is same as \( E_{original} \) and the second term is based only on spatial distribution. Second term was used for example in (Jacob et al., 2001) as constrain energy, defined by user or trained from samples. We can introduce another parameter \( \alpha \) into this equation and by convex combination of these two models we can influence behavior of segmentation process.

\[
E_{region3} = -\int_{S_{in}} \alpha \cdot \log \left( \frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) dx + (1-\alpha) \cdot \log \left( \frac{P(x \in R_{in})}{P(x \in R_{out})} \right) dx
\]  

(7)

This simplified model works well only for homogeneous regions on homogeneous background due to intensity-location independence assumption.

2.2 Training

First step is proper alignment of the images from the training set (section 4.1). For each aligned image we need binary mask representing segmented object.

Intensity of each image element is recorded to one of the histograms available for its spatial coordinates. One histogram is for \( P_{in} \) probability and the second is for the \( P_{out} \) probability. Decision about which should be used is made by the binary mask query.

These histograms tend to be sparse due to the low number of images from training set in comparison to the number of possible values in the intensity range. We use Parzen window to get smooth density estimation. Also, the resolution of the model is lower than resolution of the input images to ensure spatial smoothness of the trained model.

3 SHAPE MODEL

Previous model works quite well but still it has some flaws. With increasing variability of data in training set there grows larger area around segmented region, where \( P_{in} \) and \( P_{out} \) are almost the same. Boundary
tends to fluctuate or takes the shortest path (depending on internal and constrain energy) in these regions. So we need some other energy, which forces boundary to have proper shape.

### 3.1 Formulation

We use a vector field which tries to model behavior of the boundary normals. The shape energy is based on dot product of boundary normal and vector in our field on the same position. We use curve integral in 2D and surface integral in 3D over object’s boundary.

\[
E_{\text{shape}}^{2D} = -\int_C \mathbf{u} \cdot \mathbf{n}^+ \, dr
\]

\[
E_{\text{shape}}^{3D} = -\int_S \mathbf{u} \cdot \mathbf{n}^+ \, dS
\]

Where \( \mathbf{u} \) is trained vector field and \( \mathbf{n}^+ \) is vector field of curve/surface unit normals oriented outwards from segmented region.

As will be shown in section 4 it is useful to express shape energy as region integral (surface, volume) instead of integral over curve/surface. For that we apply Divergence theorem (Green’s theorem in 2D, Gauss–Ostrogradsky theorem in 3D) on previous equations.

\[
E_{\text{shape}}^{2D} = -\int_S \nabla \cdot \mathbf{u}(x,y) \, dx \, dy
\]

\[
E_{\text{shape}}^{3D} = -\int_S \nabla \cdot \mathbf{u}(x,y,z) \, dx \, dy \, dz
\]

### 3.2 Training

We need either binary masks, or contour representation of segmented objects for each aligned image from training set. We compute vector field containing region normals for each dataset either by computing normals from boundary representation or by computing normalized gradient of region binary mask. These vector fields are too sparse and rough (those computed as binary mask gradient). So we need to smooth them and increase the area of influence.

For this purpose we already have mechanism in form of Gradient Vector Flow (Xu and Prince, 1997), (Paragios et al., 2001), which is used for image gradient enhancement in snake based segmentation algorithms. Parameters for diffusion depend on size of training set and properties of segmented regions.

Now we should have set of vector fields with increased support (nonzero area). We want to compute one final vector field with following properties:

- Vectors have same direction as segmented region boundary passing that point.
- Size of vector reflects certainty of the direction.
- Vector field should be smooth.

We try to achieve these properties by minimizing energy functional based on GVF.

\[
\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots \text{ are vector fields of dimension } n \text{ from training set}
\]

\[
E(\mathbf{u}) = \int_\Omega -\sum_i \mathbf{v}_i \cdot \mathbf{u} + \lambda \sum_i \sum_j \left( \frac{\partial u_i}{\partial x_j} \right)^2 + \kappa \|\mathbf{u} - \sum_i \mathbf{v}_i\|^2
\]

Our desired vector field \( \mathbf{u} \) should minimize this energy formulation, which consist from three parts. First term ensures minimal direction deviation from each training vector field. Second forces our field to be smooth by minimizing partial derivatives and the third term prevent divergence by trying to keep \( \mathbf{u} \) close to training vector field. Regularization parameters \( \lambda \) and \( \kappa \) tune the tradeoff between the first, second and third term.

Using basic properties of dot product we can simplify the equation by introducing sum of training vector fields \( \mathbf{w} = \sum_i \mathbf{v}_i \).

\[
E(\mathbf{u}) = \int_\Omega -\mathbf{w} \cdot \mathbf{u} + \lambda \sum_i \sum_j \left( \frac{\partial u_i}{\partial x_j} \right)^2 + \kappa \|\mathbf{u} - \mathbf{w}\|^2
\]

\[
\mathbf{u} = [u_1, u_2, \ldots, u_n] \text{ where } n \text{ is field dimension}
\]

We solve this problem using set of Euler-Lagrange equations. For \( i \) in \( 1, 2, \ldots, n \), where \( n \) is field dimension.

\[
-w_i + 2\kappa (u_i - w_i) - 2\lambda \sum_j \frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial x_j} = 0
\]

\[
2\kappa u_i - (2\kappa + 1) w_i - 2\lambda \Delta u_i = 0
\]

This can be used in steepest descent or similar optimization algorithm.

### 4 MODEL USAGE

#### 4.1 Alignment

Quality of the trained model and its usefulness depends on the proper alignment of the training images and alignment between model and the segmented object in the processed image.

Various approaches to this task exist (Yao and Summers, 2009), but most of them would again
increase computational complexity of the whole pipeline. Since we design method for interactive segmentation we decided to let user select important features from which we compute the aligning transformation.

For our test case (section 5) we let the user specify extreme points (poles) of segmented kidney in transversal slices of CT scan.

### 4.2 Parametric Contours/Surfaces

From wide range of segmentation methods based on deformable models (snakes, levelsets, etc.) we used parametric snakes/surfaces where contour is defined as spline curve/patch and its shape is controlled by set of control points.

In comparison to shape representation in form of a levelset the parametric boundary representation can be easily used only for segmentation of objects with simple topologies. It is quite problematic to introduce shape with holes.

Our motivation for usage of the parametric boundary representation is again speed. Whole framework is easily paralelizable. Also, curves and surfaces manipulated by set of control points is well known concept from other graphical software tools. These properties makes it good candidate for interactive segmentation algorithm.

If we want to use models from previous sections in parametric boundary segmentation it is useful to compute curve/surface integral instead of surface/volume integral (divergence theorem), which is more time consuming. (Jacob et al., 2001) used this step to bind all energy terms into one unified energy term. We use the exactly same approach.

We assume that surface $\Phi$ is oriented, so normal vectors $(\Phi_u \times \Phi_v)/||(\Phi_u \times \Phi_v)||$ are oriented outwards. $F(x,y)$ and $F(x,y,z)$ are 2D/3D unified energy terms.

$$\int_S F(x,y)dx = \oint_F (\int_{-\infty}^{y} F(x,\tau)d\tau)dx$$

(15)

$$= -\oint_F (\int_{-\infty}^{y} F(\tau,y)d\tau)dy$$

(16)

$$\int_S F(x,y,z)dx dy dz = \iint_S G_x \cdot dS =$$

$$\iint_S G_y \cdot dS = \iint_S G_z \cdot dS$$

(17)

$$G_x = (\int_{-\infty}^{x} F(\tau,y,z)d\tau,0,0)$$

(18)

$$G_y = (0, \int_{-\infty}^{y} F(x,\tau,z)d\tau,0)$$

(19)

$$G_z = (0,0, \int_{-\infty}^{z} F(x,y,\tau)d\tau)$$

(20)

But for usage in optimization scheme we need partial derivatives with respect to control parameters (control points). We show only 3D version, 2D is special case and was presented in (Jacob et al., 2004).

We use definition of surface integral of second kind.

$$\iiint_S G_x \cdot dS = \int_u \int_v G_x(\Phi) \cdot (\Phi_u \times \Phi_v)dudv$$

(21)

And now we show how to compute partial derivative of our equation in respect to x-coordinate of i-th control point.

$$\frac{\partial}{\partial c_i} \int_u \int_v G_x(\Phi) \cdot (\Phi_u \times \Phi_v)dudv$$

(22)
690

foreground/background probability thresholds. t-markers automaticaly from the model.
input image using conditions 23. So we obtain initial tedious especially in 3D. So we apply thresholding to again.

If user is not satisfied with the result he can two sets. One is for foreground and second for back-

This information is then incorporated into the graph in form of t-links

User marks segmented object and background. This information is then incorporated into the graph in form of t-link weights.

By finding minimal graph cut we divide vertices in two sets. One is for foreground and second for background. If user is not satisfied with the result he can update markers (modify t-links) and run segmentation again.

Drawing foreground/background markers can be tedious especially in 3D. So we apply thresholding to input image using conditions 23. So we obtain initial markers automatically from the model. \( t_F \) and \( t_B \) are foreground/background probability thresholds.

\[
M(x) = \begin{cases} 
F & P_{gl}(I(x), x) > t_F \\
B & P_{gl}(I(x), x) > t_B \\
0 & \text{Rest} 
\end{cases} \tag{23}
\]

We use the second part of our model to modulate n-link weights. Boundaries of objects tend to follow strong edges/ridges in the input image. So it means that contour goes through parts of the image with big gradient magnitude and its normals at those parts have same direction as the gradient.

We have vector field which should represent normal directions of possible boundaries, so we can boost influence of parts of image with properly oriented gradient.

\[
G_{new}(x) = G(x) \left(1 + \alpha(\max(0, \hat{G}(x) \cdot u(x)))\right) \tag{24}
\]

Where \( G \) is original image gradient, \( \hat{G} \) normalized image gradient, \( u \) our shape model and \( \alpha \) parameter controlling strength of the effect.

To prevent distortion of vector field we don’t change direction of the vector, we only modulate its length. We do not shorten vectors. We only elongate those with proper direction. Example of gradient modulation in figure 2.

5 RESULTS

Presented models are aimed for initialization of interactive segmentation algorithms. So the quality the result also depends on user input. To rule out influence of the user input we implemented two tests, which work without user input.

First one was modified thresholding, where aligned model serves as classifier. We counted how many voxels were properly tagged, number of false positives and false negatives. We express these values relatively to object volume obtained from manual segmentation.

As a second method we used parametric snakes based on our model and term which attracts contour to areas with bigger gradient magnitude. Error was again expressed as a percentage of whole object volume.

In tables 1 and 2 are results of left kidney segmentation in CT images (with and without contrast agent – model trained for each case separately).

Measured errors show that our models work precise initialization in almost all cases – degenerated or damaged organs (patients 2 and 16) aren’t covered by models.

6 CONCLUSIONS

We introduce low level statistical models which successfully address initialization problem of interactive segmentation algorithms for class of compact objects.

Presented models describe only few certain properties of the segmented objects, but not their topology.
or shape variability. This leaves us with very rough initialization in case of complicated objects or objects with big shape variability. But even in this case we decrease amount of user labor needed for initialization of the full-fledged segmentation algorithm which follows.

### REFERENCES


