

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 segmen-

tation.

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 embedable 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 \mathbf{x} of intensity $I(\mathbf{x})$ being inside the segmented region (R_{in}) resp. outside the segmented region (R_{out}).

$$\begin{aligned} P_{in}(I(\mathbf{x}), \mathbf{x}) &= P(I(x) \wedge x \in R_{in}) \\ P_{out}(I(\mathbf{x}), \mathbf{x}) &= P(I(x) \wedge x \in R_{out}) \end{aligned} \quad (1)$$

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:

$$\begin{aligned} P(I(x) \wedge x \in R_{in}) &= P(I(x)|x \in R_{in}) \cdot P(x \in R_{in}) \\ P(I(x) \wedge x \in R_{out}) &= P(I(x)|x \in R_{out}) \cdot P(x \in R_{out}) \end{aligned} \quad (2)$$

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

$$E_{original} = \int_{S_{in}} -\log \left(\frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) d\mathbf{x} \quad (3)$$

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).

$$E_{region} = \int_{S_{in}} -\log \left(\frac{P_{in}(I(\mathbf{x}), \mathbf{x})}{P_{out}(I(\mathbf{x}), \mathbf{x})} \right) d\mathbf{x} \quad (4)$$

$$E_{region} = \int_{S_{in}} -\log \left(\frac{P(I(x) \wedge x \in R_{in})}{P(I(x) \wedge x \in R_{out})} \right) d\mathbf{x} \quad (5)$$

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.

$$\begin{aligned} E_{region2} &= - \int_{S_{in}} \log \left(\frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) + \\ &+ \log \left(\frac{P(x \in R_{in})}{P(x \in R_{out})} \right) d\mathbf{x} \end{aligned} \quad (6)$$

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 α into this equation and by convex combination of these two models we can influence behavior of segmentation process.

$$\begin{aligned} E_{region3} &= - \int_{S_{in}} \alpha \cdot \log \left(\frac{P(I(x)|x \in R_{in})}{P(I(x)|x \in R_{out})} \right) + \\ &+ (1 - \alpha) \cdot \log \left(\frac{P(x \in R_{in})}{P(x \in R_{out})} \right) d\mathbf{x} \end{aligned} \quad (7)$$

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.

$$E_{shape2D} = - \oint_C \mathbf{u} \cdot \mathbf{v}^+ dr \quad (8)$$

$$E_{shape3D} = - \iint_S \mathbf{u} \cdot \mathbf{v}^+ dS \quad (9)$$

Where \mathbf{u} is trained vector field and \mathbf{v}^+ 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.

$$E_{shape2D} = - \int_S \nabla \cdot \mathbf{u}(x, y) dx dy \quad (10)$$

$$E_{shape3D} = - \int_S \nabla \cdot \mathbf{u}(x, y, z) dx dy dz \quad (11)$$

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.

$\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3, \dots$ are vector fields of dimension n from training set.

$$E(\mathbf{u}) = \int_{\Omega} - \sum_i \mathbf{v}^i \cdot \mathbf{u} + \lambda \sum_i \sum_j \left(\frac{\partial u_i}{\partial x_j} \right)^2 + \kappa \|\mathbf{u} - \sum_i \mathbf{v}^i\|^2 \quad (12)$$

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 λ and κ 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 $\mathbf{w} = \sum_i \mathbf{v}^i$.

$$E(\mathbf{u}) = \int_{\Omega} -\mathbf{w} \cdot \mathbf{u} + \lambda \sum_i \sum_j \left(\frac{\partial u_i}{\partial x_j} \right)^2 + \kappa \|\mathbf{u} - \mathbf{w}\|^2 \quad (13)$$

$\mathbf{u} = [u_1, u_2, \dots, u_n]$ where n is field dimension. We solve this problem using set of *Euler-Lagrange equations*. For i in $1, 2, \dots, n$, where n is field dimension.

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

$$2\kappa u_i - (2\kappa + 1)w_i - 2\lambda \Delta u_i = 0 \quad (14)$$

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

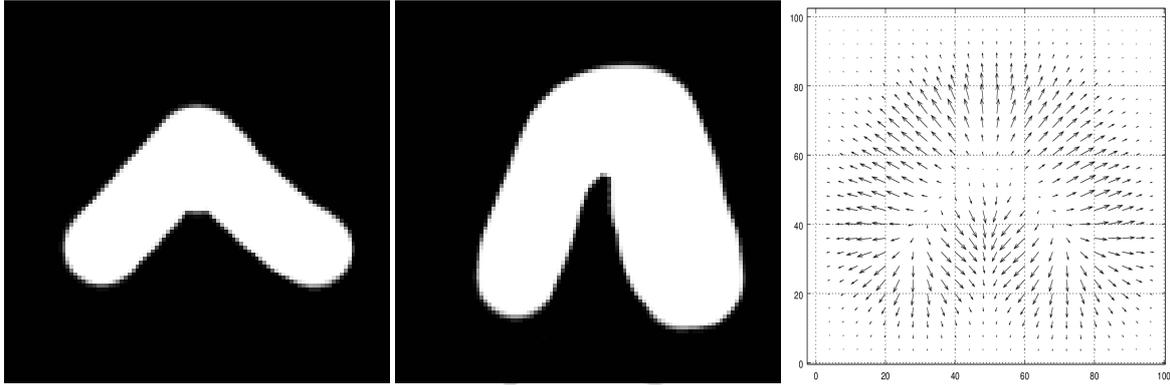


Figure 1: Two binary masks from training set and computed vector field.

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 Φ 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.

$$\begin{aligned} \int_S F(x, y) dx dy &= \oint_C \left(\int_{-\infty}^y F(x, \tau) d\tau \right) dx \quad (15) \\ &= - \oint_C \left(\int_{-\infty}^x F(\tau, y) d\tau \right) dy \quad (16) \end{aligned}$$

$$\begin{aligned} \int_V F(x, y, z) dx dy dz &= \iint_S \mathbf{G}_x \cdot d\mathbf{S} = \\ &= \iint_S \mathbf{G}_y \cdot d\mathbf{S} = \iint_S \mathbf{G}_z \cdot d\mathbf{S} \quad (17) \end{aligned}$$

$$\mathbf{G}_x = \left(\int_{-\infty}^x F(\tau, y, z) d\tau, 0, 0 \right) \quad (18)$$

$$\mathbf{G}_y = \left(0, \int_{-\infty}^y F(x, \tau, z) d\tau, 0 \right) \quad (19)$$

$$\mathbf{G}_z = \left(0, 0, \int_{-\infty}^z F(x, y, \tau) d\tau \right) \quad (20)$$

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.

$$\iint_S \mathbf{G}_x \cdot d\mathbf{S} = \int_u \int_v \mathbf{G}_x(\Phi) \cdot (\Phi_u \times \Phi_v) du dv \quad (21)$$

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

$$\frac{\partial}{\partial c_{i,x}} \int_u \int_v \mathbf{G}_x(\Phi) \cdot (\Phi_u \times \Phi_v) du dv \quad (22)$$

$$\begin{aligned}
 &= \int_u \int_v \frac{\partial}{\partial c_{i,x}} [\mathbf{G}_x(\Phi) \cdot (\Phi_u \times \Phi_v)] dudv \\
 &= \int_u \int_v \frac{\partial}{\partial x} [\mathbf{G}_x(\Phi) \cdot (\Phi_u \times \Phi_v)] \frac{\partial x}{\partial c_{i,x}} dudv \\
 &= \int_u \int_v \left[\frac{\partial \mathbf{G}_x(\Phi)}{\partial x} \cdot (\Phi_u \times \Phi_v) + \underbrace{\mathbf{G}_x(\Phi) \cdot \frac{\partial (\Phi_u \times \Phi_v)}{\partial x}}_{=0} \right] \frac{\partial x}{\partial c_{i,x}} dudv \\
 &= \int_u \int_v (F(\Phi), 0, 0) \cdot (\Phi_u \times \Phi_v) \frac{\partial x}{\partial c_{i,x}} dudv
 \end{aligned}$$

$\frac{\partial x(u,v)}{\partial c_{i,x}}$ will be often in simple form. For splines we get basis functions.

So we obtain equation for each partial derivative in respect to curve/surface control point, which is easily computable.

4.3 Minimal Graph-cut

Another method we used the models for is segmentation based on computing minimal graph cut (Boykov and Jolly, 2001; Yi and Moon, 2012; Kolomaznik et al., 2012). Each image element (pixel, voxel, subregion) is represented as *vertex* in the constructed graph. We also add another two special vertices (*source* and *sink*)

Information about neighbors is integrated in form of the edges (*n-links*) with weight based on image values or image gradient magnitude. All vertices in the graph are connected to the source and sink by *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.

$$M(\mathbf{x}) = \begin{cases} F & P_{in}(I(\mathbf{x}), \mathbf{x}) > t_F \\ B & P_{out}(I(\mathbf{x}), \mathbf{x}) > t_B \\ 0 & \text{Rest} \end{cases} \quad (23)$$

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.

$$\mathbf{G}_{new}(\mathbf{x}) = \mathbf{G}(\mathbf{x}) \left(1 + \alpha (\max(0, \hat{\mathbf{G}}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}))) \right) \quad (24)$$

Where \mathbf{G} is original image gradient, $\hat{\mathbf{G}}$ normalized image gradient, \mathbf{u} our shape model and α 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

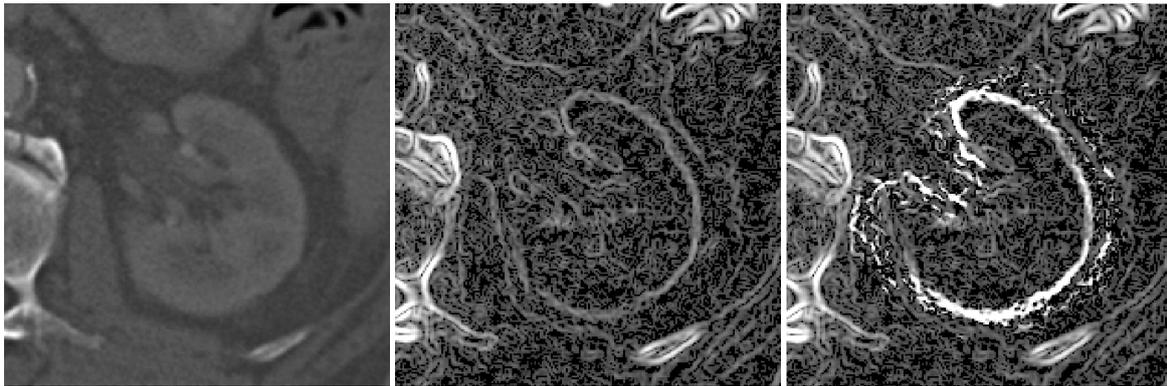


Figure 2: a) Human kidney image - part of abdominal CT scan b) Gradient magnitude c) Modulated gradient magnitude.

Table 1: Left kidney (post-contrast): *Correct* – part of organ volume correctly classified; *E_{out}* – false positive; *E_{in}* – false negative; *E_{snakes}* error of snake based segmentation (false positive and negative)

patientID	<i>Correct</i>	<i>E_{out}</i>	<i>E_{in}</i>	<i>E_{snakes}</i>
1	88.9%	8.6%	11.0%	7.1%
2	70.0%	6.0%	30.0%	26.3%
3	96.6%	13.0%	3.3%	6.2%
4	91.5%	4.5%	8.5%	1.6%
5	91.9%	3.5%	8.0%	1.9%
6	65.2%	9.5%	34.8%	4.2%
7	84.0%	2.4%	15.9%	10.3%
8	89.1%	2.2%	10.9%	5.4%

Table 2: Left kidney (without contrast agent): *Correct* – part of organ volume correctly classified; *E_{out}* – false positive; *E_{in}* – false negative; *E_{snakes}* error of snake based segmentation (false positive and negative)

patientID	<i>Correct</i>	<i>E_{out}</i>	<i>E_{in}</i>	<i>E_{snakes}</i>
9	84.5%	6.4%	15.4%	2.2%
10	91.5%	29.3%	8.5%	20.6%
11	89.5%	3.6%	10.5%	1.0%
12	92.8%	11.1%	7.1%	10.4%
13	93.4%	17.4%	6.6%	17.2%
14	71.7%	2.3%	28.3%	9.4%
15	91.4%	2.8%	8.6%	0.2%
16	61.8%	3.7%	38.2%	2.6%

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

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1.

Cremers, D., Rousson, M., and Deriche, R. (2007). A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72:215.

Heimann, T. and Meinzer, H.-P. (2009). Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis*, 13(4):543 – 563.

Jacob, M., Blu, T., and Unser, M. (2001). A unifying approach and interface for spline-based snakes. In *Proc. SPIE Med. Imaging*, l. 4322, pages 340–347.

Jacob, M., Blu, T., and Unser, M. (2004). Efficient Energies and Algorithms for Parametric Snakes. *IEEE Transactions on Image Processing*, 13(9):1231–1244.

Kolomaznik, J., Horacek, J., Krajicek, V., and Pelikan, J. (2012). Implementing interactive 3d segmentation on cuda using graph-cuts and watershed transformation. *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*.

Leventon, M., Grimson, W., and Faugeras, O. (2002). Statistical shape influence in geodesic active contours. In *Biomedical Imaging, 2002. 5th IEEE EMBS International Summer School on*, page 8 pp.

Okada, T., Yokota, K., Hori, M., Nakamoto, M., Nakamura, H., and Sato, Y. (2008). Construction of hierarchical multi-organ statistical atlases and their application to multi-organ segmentation from ct images. In *Proceedings of the 11th international conference on Medical Image Computing and Computer-Assisted Intervention - Part I, MICCAI '08*, pages 502–509, Berlin, Heidelberg. Springer-Verlag.

Paragios, N., Mellina-Gottardo, O., and Ramesh, V. (2001). Gradient vector flow fast geodesic active contours. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 67 –73 vol.1.

- Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W. E., and Willsky, A. (2003). A shape-based approach to the segmentation of medical imagery using level sets. *IEEE transactions on medical imaging*, 22(2):137–154.
- Xu, C. and Prince, J. L. (1997). Gradient vector flow: A new external force for snakes. In *IEEE Proc. Conf. On*, pages 66–71.
- Yao, J. and Summers, R. (2009). Statistical location model for abdominal organ localization. In Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., and Taylor, C., editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2009*, volume 5762 of *Lecture Notes in Computer Science*, pages 9–17. Springer Berlin, Heidelberg.
- Yi, F. and Moon, I. (2012). Image segmentation: A survey of graph-cut methods. In *International Conference on Systems and Informatics*.
- Yue, Y. and Tagare, H. (2009). Learning to segment using machine-learned penalized logistic models. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 0:58–65.
- Zhao, F. and Xie, X. (2013). An overview of interactive medical image segmentation. *Annals of the BMVA*, 2013(7):1–22.