Active contours: Application to plant recognition

Loreta ŞUTA[1], Fabien BESSY[2], Cornelia VEJA[1], Mircea-Florin VAIDA[1]
[1]*Technical University of Cluj-Napoca, 400114, Cluj Napoca, Romania*
[2]*Université François Rabelais, Tours, France*
{Loreta.Suta, Cornelia.Veja, Mircea.Vaida}@com.utcluj.ro, {Fabien.Bessy}@etu.univ-tours.fr

***Abstract*— The problem we address in this paper is object segmentation applied to plant recognition. The image can contain one or more plants on a natural background. More precisely, we aim to segment flowers. This approach poses several challenges, such as texture, multiple colors that form one object, natural background, non-homogeneous regions, etc. We propose a segmentation method built upon the Chan-Vese model,** [1]**. This model works well when there is a clear difference between the background and the objects. Our approach presents ongoing work towards flower segmentation and recognition. In this paper we present the main outcome of our research on segmentation problem: curve evolution from Chan-Vese model with a recent and faster approach,** [2]**, and its optimization at implementation level. At recognition level, we studied five features that may be extracted during segmentation. Experiments have been carried out over the Oxford Flowers dataset,** [3]**.**

Keywords- active contours; image segmentation; object recognition; natural images.

I. INTRODUCTION

Object recognition - in computer vision, can be viewed as the task of extracting one object from a given image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different viewpoints, in many different sizes / scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems in general.

In computer vision, segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, it is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The problem we address in this paper is object segmentation applied to plant recognition. The image can contain one or more plants on a natural background. Anyway, it is preferred that the these lie entirely within the image. More precisely, we aim to segment flowers. This approach poses several challenges, such as texture, multiple colors that form one object, natural background, non-homogeneous regions, etc.

We propose a segmentation method built upon the Chan-Vese model, [1]. This model works well when there is a clear difference between the background and the objects which should have the same color. We want to adapt the model to multimodal images without the need of a multiphase approach. The computational time is also an issue. We also aim to design a fast segmentation method that may suit mobile implementations.

Future directions of our research concern the combination of plant recognition with Semantic Social Software, on the edge of computer vision and human interaction.

The paper is organized as follows: section 2 describes recent developments in the segmentation domain. Section 3 describes our active contour based approach for the segmentation of natural images. Experimental results are shown in section 4 followed by future work in section 5 and conclusion.

II. RELATED WORK

Recent segmentation techniques focus on background/foreground separation, [4], geometrical models, [5], superpixel segmentation, [6], [7], [8] or active contour models, [2] and [1].The object, in this case, is the aggregation of several components. For example, the components of a plant can be the flower, the leaves and/or the stem. According to the level of detail we would like to study, segmentation methods offer multiple detail levels, such as multi-scale analysis.

Rother et al. proposed in [4] an efficient background/foreground segmentation that combines hard segmentation obtained by graph-cuts with border matting. Results show a good performance on natural images, being independent of object type, size and position. The disadvantage is the correct choice of the initialization, which is an important stage for reliable segmentation results.

In [6] the authors propose a co-segmentation method to classify flower species. The algorithm combines a previously initialized GrabCut method, [4], used for a pixel-level segmentation and discriminative learning in the superpixel descriptor space. In [7] mean-shift is employed to preserve discontinuity filtering and segmentation. Its convergence on lattices is proven, therefore presenting an unsupervised segmentation technique. The major drawback is the correct choice of the segmentation parameters which are highly dependent of the objects we need to extract. While one set is working well for greater object size, it fails to segment objects of rather smaller dimensions or presenting color variations that will be separated into different sub-objects. These are common issues encountered in plant images. [8] describes a graph-based segmentation method that relies on three initial parameters: the blurring strength, a threshold for the energy functional and the minimum area of each produced segment.

In [5] present a segmentation method adapted to extract flowers from images. An initial separation of background/foreground is computed and optimized using a MRF cost function and graph-cuts. Then, a generic shape model is fitted onto the previous result to detect petals. Convergence is obtained after several re-iterations of the previous steps, hence a heavy computational time. This drawback prevents it from being used in applications that require in real-time processing or with limited resources (e. g. smartphones).

A. Overview of Shi-Karl algorithm

Shi and Karl propose in [2] an algorithm that uses a discrete approach for the approximation of level-set based curve evolution (implicit active contours). The algorithm execution is very fast because it needs not to solve partial differential equations (PDE) in the image domain. Moreover, there is no periodical initialization of the level-set function using the signed distance function to the active contour and each calculation is done on the active contour. This method preserves the advantages of level-set methods, such as the automatic handling of topological changes. Considerable speedups (×100) have been demonstrated as compared to PDE-based narrow band level-set implementations. The optimization algorithm uses curve evolution having a double representation.

An implicit representation of the curve is used, noted Φ, as a simplified level-set function defined on a grid, usually of the same size with the image. This level-set function can take only four values that approximate the signed distance to the contour. It is the discrete approach of the method. The relation below presents the level-set function, Φ:

$$\Phi(x) = \begin{cases} 3, \text{if } x \text{ is an exterior point} \\ 1, \text{if } x \text{ is in the outside boundary} \\ -1, \text{if } x \text{ is in the inside boundary} \\ -3, \text{if } x \text{ is an interior point} \end{cases} \quad (1)$$

So, the points where $\Phi = 1$ or $\Phi = -1$ are connex in a 8-connex neighborhood.

An explicit representation of the curve is composed of two lists of points of the boundary. The list L_{out} contains all the points belonging to the outside boundary, i.e., all x with $\Phi(x) = 1$. The list L_{in} contains all points that belong to the inside boundary, i.e., all x with $\Phi(x) = -1$. This double representation of the curve is essential as Φ contains the information about the neighborhood of each point. The lists allow direct localization of the active contour without the need to scan the level-set function. All calculations are done on the active contour domain and use only integers. Using the presented implementation, the complexity of the algorithm is given by the size of the lists.

The algorithm is composed of two different cycles of evolution that are alternated permanently. Cycle 1 concerns the data dependent evolution or external evolution, whereas, cycle 2 concerns an internal evolution that performs a smoothness regularization of the active contour. During cycle 1, a speed F_d is computed and the curve evolves a number N_a times. During cycle 2, a speed F_{int} is computed and the curve evolves a number N_s times. N_s must be very low compared to N_a to allow a good convergence of the algorithm and prevent oscillations. One iteration in cycle 1 is algorithmically the same as one iteration in cycle 2. It differs uniquely in the computation of the speed function.

One iteration or one step of curve evolution in each cycle is performed using 4 procedures completed in the following order:

- switch_in(), performed during a scan through L_{out}
- a procedure to eliminate the redundant points of L_{in}
- switch_out() performed during a scan through L_{in}
- a procedure to eliminate the redundant points of L_{out}.

We note x the current position of a point in the list and y, its neighbor in a 8-connex neighborhood.

The switch_in() procedure performs a local outward movement of the curve and is done if the speed of a point x of L_{out} is strictly positive.

$$\begin{aligned} & if\ \Phi(y) = 3 \implies \Phi(y) \leftarrow 1 \ \wedge\ append\ y\ to\ L_{out} \\ & \Phi(x) \leftarrow -1 \\ & append\ x\ to\ L_{in} \\ & delete\ x\ from\ L_{out} \end{aligned} \quad (2)$$

Symmetrically, the switch_out() procedure performs a local inward movement of the curve and is done if the speed of a point x of L_{in} is strictly negative.

$$\begin{aligned} & if\ \Phi(y) = -3 \implies \Phi(y) \leftarrow -1 \ \wedge\ append\ y\ to\ L_{in} \\ & \Phi(x) \leftarrow 1 \\ & append\ x\ to\ L_{out} \\ & delete\ x\ from\ L_{in} \end{aligned} \quad (3)$$

Moreover, a procedure to eliminate redundant points in L_{in} is performed if at least one point is switched to preserve the connectivity of the lists.

$$if\ \exists\ y | \Phi(y) \geq 0 \implies \Phi(x) \leftarrow -3 \ \wedge\ delete\ x\ from\ L_{in} \quad (4)$$

Symmetrically, a procedure to eliminate redundant point of L_{out} is completed after the switch_out() procedure if at least one point is switched to preserve the connectivity of the lists.

$$if\ \exists\ y | \Phi(y) \leq 0 \implies \Phi(x) \leftarrow 3 \ \wedge\ delete\ x\ from\ L_{out} \quad (5)$$

After each iteration in the cycle 1, a stopping condition is checked based on the changes performed in lists:

$$signum(F_d) = \begin{cases} -1, \forall x \in L_{out} \\ +1, \forall x \in L_{in} \end{cases} \quad (6)$$

If there are no switched points, cycle 1 is stopped and cycle 2 is performed. The speed F_{int} in the cycle 2 is calculated with a Gaussian kernel applied on function $signum(\Phi)$ for the current points of the lists.

B. Overview of Chan-Vese model

The active contour model proposed in [1] is based on the following properties: no stopping edge-function, no use of image gradient and a stopping term based on the Mumford-Shah segmentation technique. The resulting advantage is object detection with discontinuous boundaries and/or smooth edges.

The algorithm represents the minimization of an energy-based segmentation defining the fitting term:

$$\begin{aligned} F(c_1, c_2, C) = \mu \cdot Length(C) + v \cdot Area\big(inside(C)\big) \\ + \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 dxdy \\ + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 dxdy \end{aligned} \quad (7)$$

where $u_0(x,y)$ is the intensity of the pixels (x,y), C is a variable curve, $\mu \geq 0, v \geq 0$ and $\lambda_1, \lambda_2 > 0$ are fixed integer parameters, the constants c_1, c_2 are the averages of $u_0(x,y)$ depending on the curve C.

The object boundary is the minimizer of the fitting term.

We focused on the Chan-Vese model because of its multiple advantages. The algorithm is capable to segment an image if it is corrupted by several distortions, such as blur, gaussian noise, etc. Usually, blur is one of the most unwanted distortion for segmentation methods, as it impedes to obtain reliable results. Instead of trying to include other processing steps (such as preprocessing, noise detection, blur assessment, etc.), we can implement a segmentation method that is robust to all the aforementioned image degradations. The drawback remains in the high computational costs. However, curve evolution proposed by Shi and Karl, [2], gives a very simple and fast solution.

III. OUR APPROACH

The model introduced by Chan-Vese in 2001 works well when there is a clear difference between the background and the objects which should have the same color. It offers the possibility to segment objects that are not well defined by their gradient. Our goal is to achieve fast color image segmentation. We consider the active contour based energy functional where we aim to optimize curve evolution with an implicit level-set representation.

A. Image Segmentation

Contrary to [1], a field speed is used by the algorithm instead of the energy functional. A zero-crossing search derives of the minimization problem. We propose the following demonstration that adaptes the algorithm of Shi and Karl, [2] with the Chan-Vese model, [1].

Let us consider the implicit representation of the curve in a continuous domain. In this case the equation (7) can be rewritten as follows, where the terms $Length(C)$ and $Area(C)$ lose their significance:

$$\begin{aligned} F(c_1, c_2, C) = \lambda_1 \int_{\Phi(x,y)=-\infty}^{\Phi(x,y)=0} |u_0(x,y) - c_1|^2 dxdy \\ + \lambda_2 \int_{\Phi(x,y)=0}^{\Phi(x,y)=+\infty} |u_0(x,y) - c_2|^2 dxdy \end{aligned} \quad (8)$$

The derivate of (8) can be written as:

$$\frac{dF(c_1, c_2, C)}{dxdy} = \lambda_1 \frac{d \int_{\Phi(x,y)=-\infty}^{\Phi(x,y)=0} |u_0(x,y) - c_1|^2 dxdy}{dxdy} + \lambda_2 \frac{d \int_{\Phi(x,y)=0}^{\Phi(x,y)=+\infty} |u_0(x,y) - c_2|^2 dxdy}{dxdy} \quad (9)$$

Solving equation (9), we obtain the speed function for each point, given by (10):

$$F_d(x,y) = \lambda_2 |u_0(x,y) - c_2|^2 - \lambda_1 |u_0(x,y) - c_1|^2 \quad (10)$$

where F_d is the speed function, $\lambda_1, \lambda_2 > 0$ are fixed integer parameters, c_1, c_2 are the means and $u_0(x,y)$ is the intensity of the pixels (x,y) in the YUV color space. As we work on natural images, the chrominance is more important than the luminance. As a result, we give more wheight to the UV components.

The computation of means c_1 and c_2 is calculated at the beginning of each iterations in cycle 1. Naively, it needs to scan through the image for each iteration. The resulting complexity, in this case, is in $O(image\ size)$. We propose a method to reduce the complexity of means computing.

Let us define c_1 and c_2 the means of intensities of a given pixel x with respect to $\Phi(x) > 0$ and $\Phi(x) < 0$ and not just $\Phi(x) = 3$ and $\Phi(x) = -3$.

Among the four procedures, uniquely switch_in() and switch_out() change the sign of Φ for the current point but not for the neighboring points. In consequence, we define two new procedures before switch_in() and switch_out() preformed in the both cycles that will update a number of points in each region, N_{out} and N_{in}, and a number of sums of intensities in each region, S_{out} and S_{in}. Using this optimization, the complexity will become $O(size\ Lin + size\ Lout)$, except for the first iteration where an initialization is needed the complexity is always in $O(image\ size)$.

Before performing the switch_in() procedure, we perform the following steps:

$$\begin{gathered} N_{out} \leftarrow N_{out} - 1 \\ N_{in} \leftarrow N_{in} + 1 \\ \\ S_{out} \leftarrow S_{out} - u_0(x,y) \\ S_{in} \leftarrow S_{in} + u_0(x,y) \end{gathered} \quad (11)$$

Before performing the switch_out() procedure, we perform the following steps:

$$\begin{gathered} N_{in} \leftarrow N_{in} - 1 \\ N_{out} \leftarrow N_{out} + 1 \\ \\ S_{in} \leftarrow S_{in} - u_0(x,y) \\ S_{out} \leftarrow S_{out} + u_0(x,y) \end{gathered} \quad (12)$$

Figure 1 shows the initial state of the active contour before the current iteration. Note the initial state of the two lists and of the level-set function.

Figure 2 shows the evolution of the curve with the presented approach for one iteration in cycle 1. In the final step, the curve is always preserving its property of 8 connectivity. The curve is being evolved towards the object. Note the changes in the two lists. The list which is outside the object evolves towards the object while the outside list stays outside the center of the object.

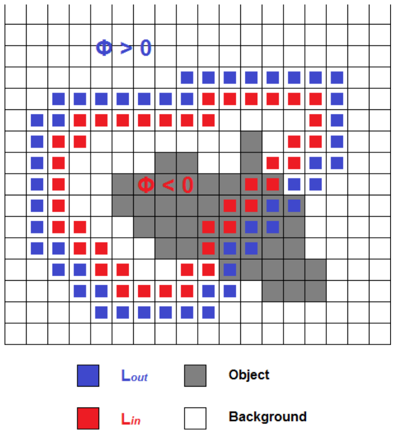

Figure 1: State of the active contour before a current iteration.

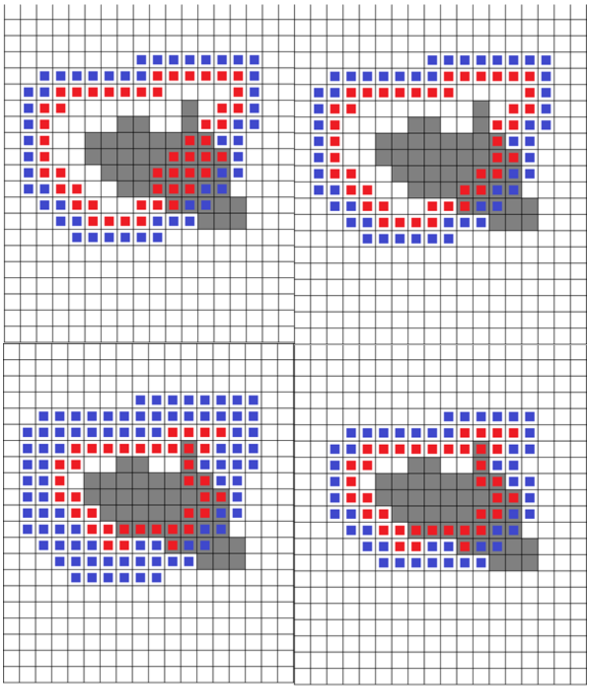

Figure 2: Curve evolution for one iteration in cycle 1 for each of the four procedures. First row from left to right : procedure switch_in and procedure to eliminate redundant points of L_{in}. Second row from left to right : procedure switch_out and procedure to eliminate redundant points of L_{out}.

B. Feature extraction for object recognition

For recognition we use the following region descriptors based on geometric properties. Several of the following features can be extracted during segmentation. These are used to train the decision tree C4.5 in order to classify the object. We opted for decision tree approach because of its advantages: fast training, fast recognition response and visible set of rules that can be used furthemore in other implementations.

a. Perimeter

$$P(s) = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (13)$$

b. Area

$$\sum_x \sum_y I(x, y)\Delta A \quad (14)$$

c. Compactness

$$C(s) = \frac{A(s)}{P(s)^2/4\pi} \quad (15)$$

d. Irregularity

$$I(s) = \frac{\pi \max((x_i-\bar{x})^2+(y_i-\bar{y})^2)}{A(s)} \quad (16)$$

e. Ratio of the maximum to the minimum of the radius:

$$IR(s) = \frac{\max(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2})}{\min(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2})} \quad (17)$$

where $I(x, y)$ is the binary representation of the segmented image, (x_i, y_i) is the coordinate of the i^{th} pixel in I and $(\bar{x}, \bar{y})$ represent the coordinate of the center of mass of the region.

IV. EXPERIMENTAL RESULTS

Tests have been performed over the Oxford Flowers dataset, [3]. We compared the proposed method to BiCoS, [6]. As the authors provide a ground-truth model, we computed two metrics, segmentation covering and the Hausdorff distance respectively.

We computed the accuracy of the segmentation based on the following formula:

$$Accuracy = \frac{t_p+t_n}{t_p+t_n+f_p+f_n} \quad (18)$$

Table 1 show the performance results obtained over 240 images from the dataset, [3]. The two methods have similar performance results but having different approaches. BiCoS is expected to perform better, as the method implements a supervised SVM over the database in order to achieve image segmentation.

Table 1: Summary table on performance results.

Method	Modified Hausdorff Distance	Segmentation Covering	Accuracy
BiCoS	19	0.93	0.90
Our approach	23	0.90	0.89

Using the caractheristics presented in the previous section, we performed the recognition on the analyzed dataset using the decision tree C4.5. The recognition rate in cross-validation is 0.75.

Figure 3: Results obtained with our approach (first row: original images; second row: segmentation results).

Table 2 presents the confusion matrix for the trained dataset. For the readability purposes, in the presentation of confusion matrix we have chosen to use for the flowers the common English names instead of Latin names.

Figure 2 presents our segmentation results on challenging images. Note that even if the objects are formed by several colors or they present texture information, segmentation can be performed accurately preserving contour and object details.

Table 2: Confusion matrix for 848 images organized in 16 classes from the tested dataset, [3].

Prediction / Class	daffodil	snowdrop	lily valley	bluebell	crocus	iris	tiger lily	wild tulip	fritillary	sunflower	daisy	colts foot	dandelion	buttercup	windflower	pansy
daffodil	55	1	0	0	1	2	1	1	1	0	2	1	0	3	3	0
snowdrop	0	32	1	0	1	3	2	0	4	0	2	1	0	1	2	1
lily valley	0	0	8	0	2	0	0	0	4	0	0	1	0	0	1	1
bluebell	0	0	0	18	1	1	1	0	1	1	1	0	0	0	3	1
crocus	2	1	0	1	29	1	0	3	0	2	1	0	0	4	3	3
iris	1	0	0	0	1	71	0	0	2	1	0	0	0	1	1	0
tiger lily	1	1	0	1	0	2	33	2	1	0	1	2	0	1	3	1
wild tulip	0	0	0	0	0	1	0	33	1	0	1	1	0	0	3	1
fritillary	0	2	0	0	0	1	0	1	54	3	0	0	1	0	2	1
sunflower	0	0	0	0	0	1	0	0	1	57	1	1	2	2	3	3
daisy	0	0	0	0	0	2	2	1	1	2	41	3	3	0	1	1
colts foot	1	1	1	0	0	1	0	1	0	2	1	29	0	0	3	2
dandelion	1	1	0	0	0	0	2	0	0	2	3	1	42	0	1	3
buttercup	1	0	0	0	0	2	0	0	1	0	0	0	0	45	2	3
windflower	2	0	0	1	1	5	1	0	0	0	1	1	0	2	44	5
pansy	2	1	1	0	0	1	1	0	2	0	0	1	0	2	0	45

Figure 4: Challenging flower classes for segmentation (first row – original images; second row – segmentation results).

V. FUTURE WORKS

Future perspectives involve the combining of such computer vision approaches with Social Semantic Web paradigm. Social enabled platforms allows user to easily publish content. The convergence between social communication and semantically enriched information, the appearance of Mass Collaborative (MC) systems leads to virtualization even for organizations and communities.

Digital resources (assets, documents) could easy reside in many types of web systems: portals, Content Management Systems (CMS) or digital repositories [9]. Wiki systems, as Wikipedia, represent by excellence collaborative tools for shared content inside of a known community. Establishing wikis is an activity driven by the interest of a small of medium community of users with the purpose of eLearning, Virtual Research Environmets or documentation. Often, internal wiki digital repositories contain high quality images uploaded by specialists. Wikis, as broader environments in which shared digital repositories co-exist in and exchange information with contains structured and unstructured data. It represents part of the organization's valuable explicit knowledge. Only a little part of these assets have metadata attached. Sometimes, the minimal metadata set atached to an image benefits of semantic technologies on the form of meaningfull annotations. But even this emprouvement only partially address lack of information and structure [10].

Semantic properties provided by semantic wikis systems, are mostly related with wiki text. Certain wikis allows user to manually annotate part of images using on bespoke created tools. These tools allows user to rigidely annotate part of images.

Our vision is to create a tool that can combine benefits from both directions: semnatic annotation provided by semantic MediaWiki can be more accurate suggested by an image recognition system. Moreover, many unkown images burried in wikis'shared repositories can be automatically clasified, exposed and re-used when appropriate by external harvesting agents. Considering that many wikis are used by communities characterized by a low level of technological expertise, this kind of smart annotator tool could be easily used not only by specialists, but even by everyday user.

It would provide automatic analyses for investigators such as biologists analyzing the natural environment, ecologists promoting its preservation, students, etc.

VI. CONCLUSIONS

This paper addresses partially an old ambition of the computer vision: automatic image recognition. Specifically, we present ongoing work related with object segmentation applied to plant recognition. In this paper, we have presented several approaches, models, algorithms and related works in our area of interest.

We propose a combination of methods that adapts the algorithm of Shi and Karl, with the Chan-Vese model. At recognition level, we studied five features that may be extracted during segmentation. The experimental results on the Oxford database are having 89-90% of accuracy at the segmentation level and 76-80% at recognition level. Recognition may be improved by adding new features based on the contour and color properties that can better discriminate between flower categories. At segmentation level, further optimization includes to adapt the speed function based on particular flower properties.

These results make us confident in a possible implementation of the algorithm to mobile devices.

Future perspectives involve the inclusion of such computer vision approaches into a semantic MediaWiki. It would provide automatic analyses for investigators such as biologists analyzing the natural environment, ecologists promoting its preservation, students, etc.

ACKNOWLEDGEMENT

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

REFERENCES

[1] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266-277, 2001.

[2] Y. Shi and W. C. Karl, "A real-time algorithm for the approximation of level-set based curve evolution," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 645-656, 2008.

[3] Oxford Flowers Database. (2011) Segmentation data. [Online]. http://www.robots.ox.ac.uk/~vgg/data/bicos/

[4] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts," *ACM Transactions on Graphics* , vol. 23, no. 3, pp. 309-314, 2004.

[5] M. E. Nilsback and A. Zisserman, "Delving deeper into the whorl of flower segmentation," *Image and Vision Computing*, vol. 28, no. 6, pp. 1049-1062, 2010.

[6] Y. Chai, V. Lempitsky, and A. Zisserman, "BiCoS: A Bi-level Co-Segmentation Method for Image Classification," in *In Proc of ICCV*, 2011.

[7] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, 2004.

[9] M. Krötzsch, D. Vrandecic, M. Völkel, and H. Haller, "Semantic Wikipedia," *In Journal of Web Semantics*, no. 5/2007, pp. 251-261, 2007.

[10] C. Schindler, C. Veja, M. Rittberger, and D. Vrandecic, "How to Teach Digital Library Data to Swim into Research," in *In Proceedings of i-Semantics conference*, 2011, pp. 142-149.