

Delving into the whorl of flower segmentation

Maria-Elena Nilsback and Andrew Zisserman

Visual Geometry Group

Department of Engineering Science, University of Oxford

Abstract

We describe an algorithm for automatically segmenting flowers in colour photographs. This is a challenging problem because of the sheer variety of flower classes, the intra-class variability, the variation within a particular flower, and the variability of imaging conditions – lighting, pose, foreshortening etc.

The method couples two models – a colour model for foreground and background, and a generic shape model for the petal structure. This shape model is tolerant to viewpoint changes and petal deformations, and applicable across many different flower classes. The segmentations are produced using a MRF cost function optimized using graph cuts.

The algorithm is tested on 13 flower classes and more than 750 examples. Performance is assessed against ground truth segmentations.

1 Introduction

There is an interesting research theme in computer vision of using a general class model to initialize a segmentation, and then improving the segmentation using image specific features. The LOCUS approach of Winn and Jovic [15] is a good example, where a shape matte (e.g. for a car or horse) is used to propose a foreground/background segmentation, colour distributions are then gathered in each region, and the final segmentation achieved using a colour based binary MRF optimized with graph cuts. The ObjCut algorithm of Kumar *et al.* [5] similarly proposes a position and configuration for a class instances (e.g. a cow or horse) using a pictorial structure with boundary shapes and texture features, and then, again, incorporates colour measured from the proposed regions to carry out a MRF segmentation with graph cuts. Similar ideas are present in several other recent class based segmentation methods [2, 7, 8, 13]. Even though these methods are applied to a variety of classes, e.g. cars, cows, faces, horses, for the most part a different model is used for each class, and a different model is used for each view of the class (e.g. cars rear, cars side).

In this paper we introduce two variations on this theme: first, we reverse the order in which the features are used – we start with colour to propose a foreground/background segmentation and use this to initialize image specific shape measurements; second, we use a generic shape model which is applicable across a number of classes and viewpoints.

Our target application is segmenting flowers from photographs as part of the process of automated flower classification [11]. In the manner of ObjCut [5] our goal is to automatically segment out the flower given only that the image is known to contain a flower, but no other information on the class or pose. Figure 1 illustrates the challenge of the segmentation task, and shows the fitted generic flower shape model for several flower classes.

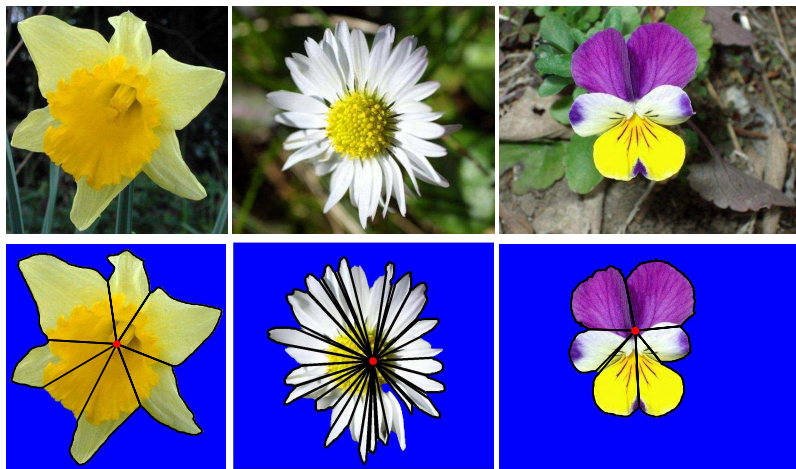


Figure 1: Top row: typical flower photographs, note the variety of imaging conditions. The classes are (left to right) daffodil, daisy and pansy. Bottom row: the flower shape model fitted automatically (note the lines to the flower “centre”) and the resulting segmentation. The same shape model is used in all cases despite differing numbers of whorls, the large variation in the number of petals (daffodil & pansy vs daisy), and variation in the shape of the petals.

The method is evaluated on the Oxford Flower Dataset available at [1]. The dataset has 17 flower classes (e.g. buttercup, daffodil, iris, pansy), with photographs exhibiting typical (large) variations in viewpoint, scale, illumination and background. Segmenting such photographs is challenging due to both the variety of colours and the variety of shapes. If we knew that we were looking for a daffodil or a bluebell, we could build one foreground and one background colour model for each of these classes – though this would still give problems with the sky being segmented as foreground in a bluebell photograph for example. But here we do not wish even to specify the flower class in advance. Shape also poses a challenge because of the many different types and whorls of petals. Even on the same flower there are local deformations of the petal shape.

Previous work on segmenting flowers using colour by Das *et al.* [4] has exploited domain knowledge on the precise colour of flowers. For example, a flower is rarely brown or green (this is not valid for the sunflower class in the Oxford Flower Dataset). In addition [4] learn an image specific background colour model from the periphery of the image. Again, this method is not applicable in our case since the flowers often reach the image periphery and so the background model would be corrupted with foreground colours. Saitoh *et al.* [12] have also proposed a method for extracting flowers regions. It is based on “Intelligent Scissors” [10], which find the path between two points that minimizes a cost function dependant on image gradients. The method works under the assumption that the flower is in focus and in the centre of the photograph and that the background is out of focus. Under this assumption the cost between any two points on the flower is smaller than the cost between a point in the background and a point in the foreground. By fixing the midpoint of the image as part of the flower this can be used as a starting point for finding the flower region. This method requires no prior colour information. Unfortunately these assumptions do not apply in our case.

This paper is organized as follows. In Section 2 we describe our segmentation algorithm, and introduce the flower shape model. Section 3 specifies the experimental procedure, and Section 4 assesses the quality of the flower segmentations against ground truth.

2 The Segmentation Algorithm

2.1 Overview

We first obtain an initial flower segmentation using *general* (non-class specific) foreground and background colour distributions. These distributions are learnt by labelling pixels in a few sample images of each class in the dataset as foreground (i.e. part of the flower), or background (i.e. part of the greenery), and then averaging the distributions across all classes. Given these general foreground and background distributions, a binary segmentation is obtained using the contrast dependent prior MRF cost function of [3], optimized with graph cuts. This is the method used in [11]. This segmentation may not be perfect, but is often sufficient to extract at least part of the external boundary of the flower.

The generic flower shape model is then fitted to this initial segmentation in order to detect petals. The model selects petals which have a loose geometric consistency using an affine invariant Hough like procedure. The image regions for the petals deemed to be consistent are used to obtain a new *image specific* colour foreground model. This image specific foreground model replaces the general foreground model (the background model is unchanged), and the MRF segmentation is repeated. In cases where the initial segmentation was not perfect, the use of the image specific foreground often harvests more of the flower. The steps of shape model fitting and image specific foreground learning can then be iterated until convergence.

The algorithm is illustrated in figure 2. We first describe these stages in more detail and then give implementation details.

2.2 Segmentation

The segmentation method is an implementation of the binary MRF with a contrast dependent prior proposed by [3]. Each pixel has two possible states corresponding to foreground or background. The probability of each state depends on the likelihood that the pixel's colour corresponds to the foreground or background colour distribution. The cost function that is minimized has a unary term for each pixel (corresponding to the log likelihood of its state) and a pairwise prior which adds a cost if there is a difference in the state between a pixel and its neighbour, *but* this cost is diminished if there is a strong gradient between the pixels (as measured in the images). The inclusion of data-dependent pairwise terms for pixels in a clique gives a substantial improvement in segmentation quality, and the resulting MRF can still be minimized using graph cuts as described in [3].

2.3 Generic flower shape model

The flowering parts of a flower can be either petals, tepals or sepals. For simplicity we will refer to these as petals.

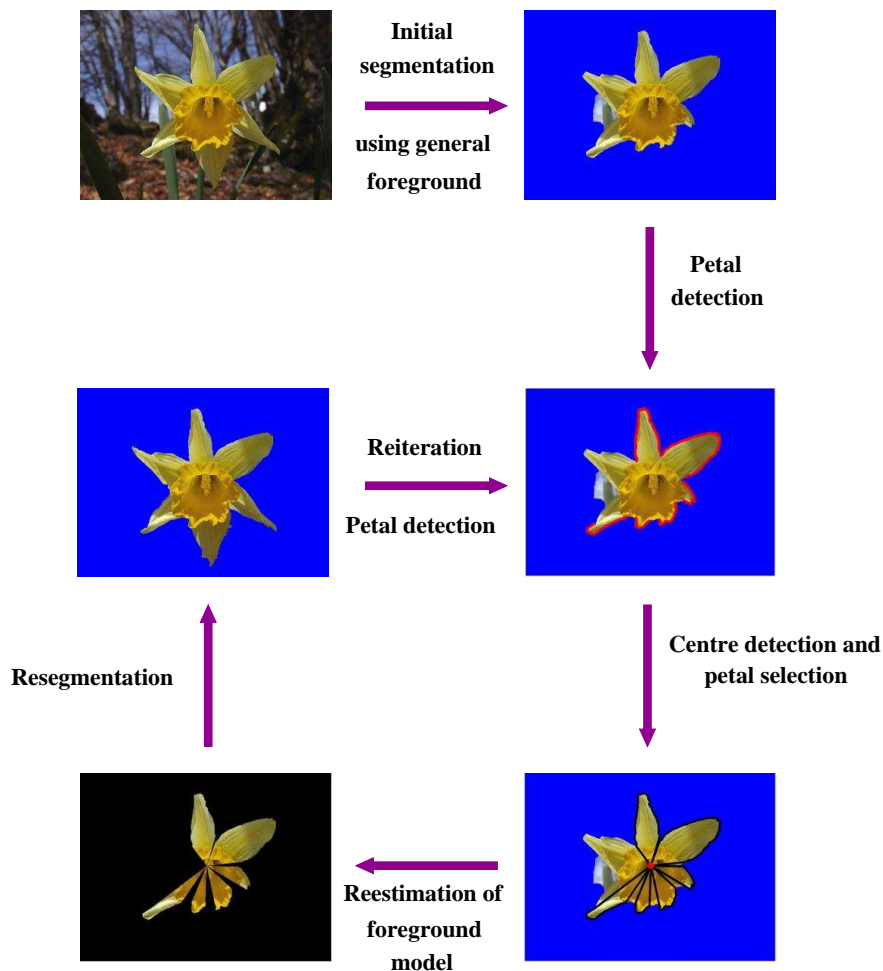


Figure 2: An overview of the segmentation algorithm.

The geometric representation of a petal and flower is illustrated in figure 3. In a frontal view of a “perfect” flower (think of a daffodil) the lines joining the extreme point to the mid-point of each petal coincide at the flower centre. However, under reasonable viewpoints, and provided the deformation of the flower is not excessive, this construction is still a good approximation, and the agreement of each petal with a common flower centre can be used to verify or remove putative petals. The model represents a loose geometric configuration of petals. The particular shape of the petals and their number is not specified, nor is the symmetry of their arrangement. The only requirement is that the image regions deemed to be petals should agree on the flower “centre”.

The construction of the extreme point is based on the canonical frame representation of Lamdan *et al.* [6]. Note, this construction (and indeed the entire flower shape model) is affine invariant. So, for example, it is unaffected if the image is rotated, scaled, or its aspect ratio changed. In particular this degree of invariance makes the construction

tolerant to out of plane rotations – as the flower head turns from the camera.

Since the model does not make strong requirements on the number or arrangement of petals, it is applicable to flowers with a small number of petals with rotational symmetry (e.g. daffodils, windflowers), to flowers with very many petals with rotational symmetry (e.g. sunflowers, dandelions), and to flowers with a small number of petals without rotational symmetry (e.g. iris, pansies). Somewhat surprisingly, the same construction may be used for side views of flowers as illustrated by the fritillaries and tiger lilies in figure 5.

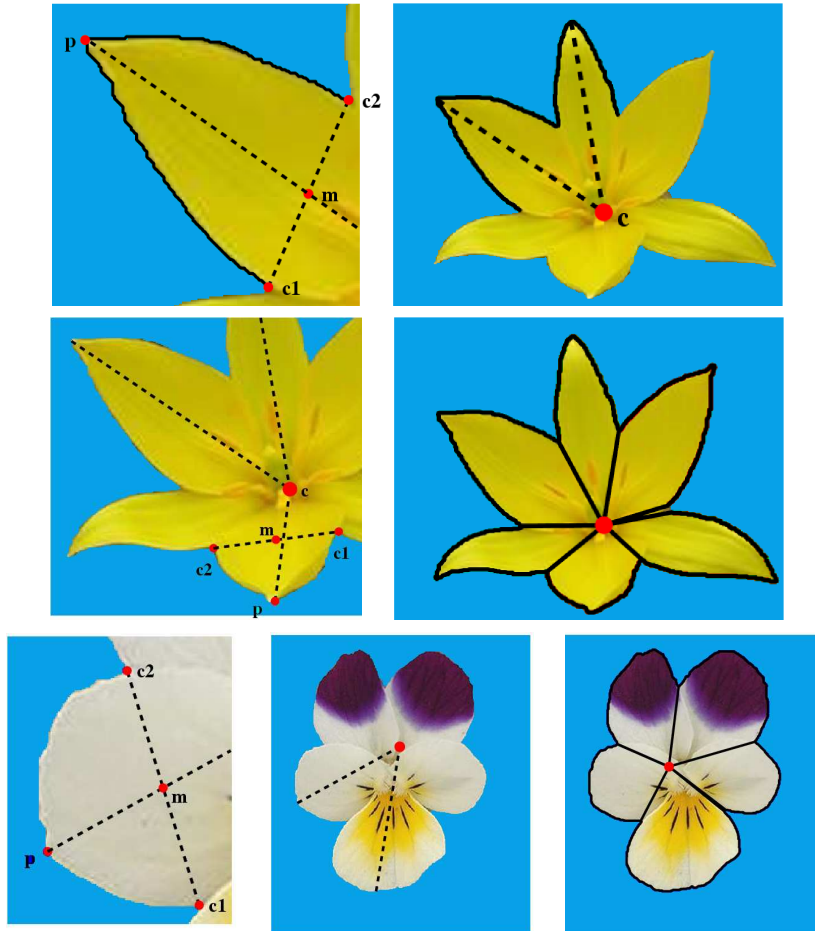


Figure 3: **The geometric flower model.** Top row: (left) A petal is defined by the boundary curve between the corners c_1 and c_2 . The extreme point p is the point on the curve furthest from the line $\langle c_1, c_2 \rangle$, and with tangent parallel to the line. (right) a hypothesis for a centre obtained by intersecting two petal mid lines $\langle p, m \rangle$. Middle row: (left) The consistency of a putative centre c is measured by the deviation of the line $\langle c, p \rangle$ from the mid-point m , and (right) the petal regions consistent with the centre with greatest consistency. Bottom row: A second example of fitting the model, here for a pansy, left to right: single petal model, putative centre, final centre and verified petals consistent with the centre.

2.4 Implementation details

Colour distribution representation and likelihood. The colour distribution is represented in RGB-space using a histogram with $N = 14^3$ bins, i.e. 14 bins per dimension. The log likelihood of a pixel under the normalized histogram $hist(r, g, b)$ is then proportional to $-\log(hist(r, g, b))$. Empty bins are dealt with by adding $0.5/N$ to all bins (including non-empty ones) and then renormalizing the histogram. This avoids the problem of computing the log of empty bins, and is equivalent to a uniform prior over the colour space [9].

Petal detection. The boundary of the segmented foreground region is traced by examining an 8-neighbourhood around each foreground pixel. Once all the boundary pixels are found, corners are detected using a “worm” that slides around the boundary and declares a corner at a boundary point where the distance to the straight line between its head and tail exceeds a threshold, and is a local maximum. Convex corners (wrt to the background region) are the potential petal corners.

The top row in figure 3 shows how petals are detected. Given two neighbouring corner points, \mathbf{c}_1 and \mathbf{c}_2 , the line $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ is determined. Next we search for a point, \mathbf{p} , on the boundary between \mathbf{c}_1 and \mathbf{c}_2 , which has the same orientation as the line $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$. If no points are found or if the ratio between the distance along the boundary and the length of $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ is smaller than a threshold, t , the petal between the two corners is removed. The line $\langle \mathbf{p}, \mathbf{m} \rangle$ joining \mathbf{p} , and the middle point \mathbf{m} of the segment $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ indicates where the centre of the flower might lie.

Centre voting. For each pair of petals we find the intersection between the corresponding lines $\langle \mathbf{p}, \mathbf{m} \rangle$. This point of intersection, \mathbf{c} , is a putative centre. Additional petals are added with a cost corresponding to deviation from \mathbf{m} of the intersection, \mathbf{i} , of the lines $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$ and $\langle \mathbf{p}, \mathbf{c} \rangle$ (the cost is measured on each petal). The deviation is normalized by the half-length of the segment $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle$. If the deviation is greater than unity, i.e. the intersection \mathbf{i} lies outside the segment, then the petal is not added to the putative centre. The centre with highest consistency, in terms of number of petals and lowest cost, is chosen to be the centre of the flower. The petals consistent with this centre, together with the triangular wedge of the centre region, are used to refine the foreground model. If no centre is found we lower the threshold, t , in order to detect more petals.

Segmentation convergence. The detection of petals and refinement of the foreground model is repeated iteratively. Once there is no change in the segmented region between two consecutive iterations the algorithm has converged. If the algorithm does not converge, we choose the initial segmentation.

Parameter selection The size of the worm and the threshold, t , are learnt from a subset of 10 images per category by optimizing the overlap score.

3 Experimental procedure

For the experiments we exclude images in the Oxford Flower Dataset that contain fields of flowers or very small flowers, that are too subsampled, or where there is no clear foreground region. Following these criteria, four of the classes (snowdrops, lily of the valley’s, cowslips and bluebells) have insufficient images and are removed. This leaves 13 flower classes with a total of 753 images.

All of the remaining images were ground truth labelled into foreground and background regions using a trimap. In the trimap a small proportion of pixels and certain regions are left unlabelled – principally the pixels very close to the flower boundary, and regions that are difficult to classify, e.g. a small out of focus flower in the background. Not labelling pixels close to the boundary greatly simplifies the annotation procedure as precisely selecting pixels near a boundary is painfully slow. Due to the contrast dependent prior the graph cuts algorithm generally aligns its boundaries with image edges, so we are more interested in assessing mistakes elsewhere. Figure 4 shows an example trimap labelling for an image. The red area is labelled foreground, the green area background, and the black area is unlabelled. The trimaps are available at [1].

The performance is measured by computing an overlap score, P , between the ground truth segmentation and the segmentation obtained by the algorithm:

$$P = \frac{\text{true foreground} \cap \text{segmented foreground}}{\text{true foreground} \cup \text{segmented foreground}} \quad (1)$$

In evaluating this score only the labelled pixels are considered (i.e. the unlabelled pixels have no effect). The score is 1 for a perfect segmentation, and less than 1 otherwise.



Figure 4: An image and its corresponding trimap ground truth labelling. The black region is unlabelled, and is not used in assessing performance. The red region is foreground, and the green region background.

4 Results

Figure 5 shows examples of the coupled algorithm improving the segmentation for different classes. It can be seen that it both manages to retrieve missing flower regions, and remove background regions that initially were misclassified. Typically only a few iterations are required – in 68% of the cases there is very little change after the first iteration. After five iterations 97% of the segmentations have converged.

For a quantitative comparison we plot the proportion of images that have an overlap score greater than P , against P . This graph is chosen to resemble an ROC curve, in that perfect performance corresponds to a curve approaching the top left corner. In figure 6 we compare performance using only the general colour distributions without using the

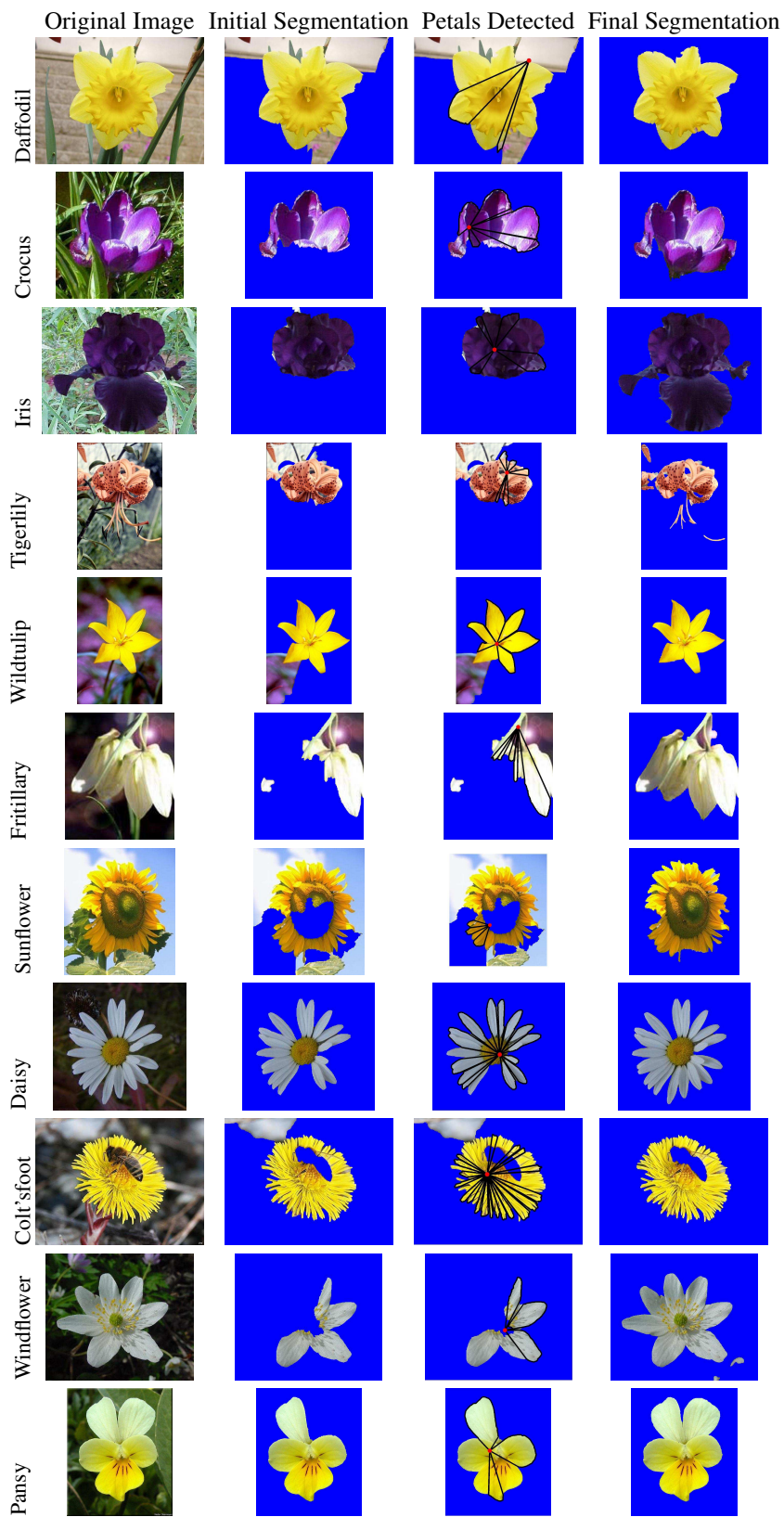


Figure 5: Example results for a number of classes. In each case iteratively updating the foreground colour model leads to a full segmentation, despite only a partial segmentation at the start. There are examples where missed foreground is retrieved (crocus, iris) and where erroneous background is removed (tiger lily, wild tulip).

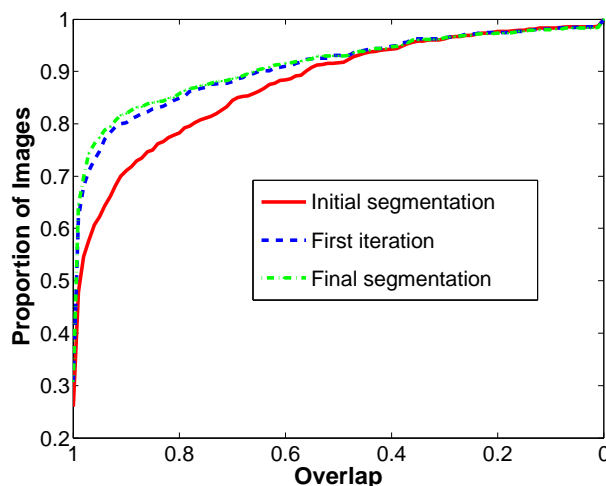


Figure 6: Comparison of segmentation performance with and without using the iterative algorithm. The x axis shows the overlap score P , and the y axis the proportion of images that have an overlap score greater than or equal to P . The solid red line shows the results of the initial segmentation, the dashed blue line the results of the first iteration, and the green line is the result after convergence. Note the clear improvement brought by the iterative algorithm over the initial segmentation.

shape model (essentially the method of [11]), to the performance using the coupled segmentation algorithm (iterating colour and shape fitting). It can be seen that learning an image-specific foreground distribution significantly improves the segmentation: for example the initial segmentation has an overlap of 95% for 62% of the images and this is increased to 77% (with 75% of the images reaching this point after a single iteration).

To give an idea of the type of error that occurs for various overlap errors, figure 7 shows typical segmentation deficiencies at levels of P in the range 0.92 – 1.0 for two different images.

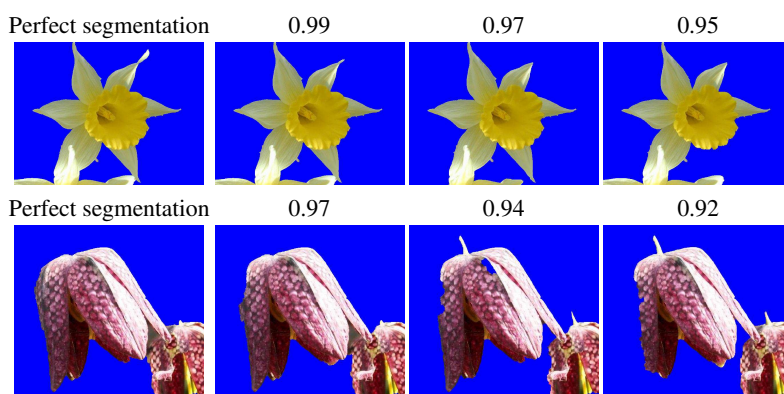


Figure 7: Segmentation deficiencies for different levels of overlap. The overlap score is given above the images. Top row: A bit of the top right petal disappears at 0.99, and at 0.95 an entire petal is lost. Bottom row: At 0.97 a bit of the left edge is lost, at 0.94 a bit of the stalk is added and a hole appears, and at 0.92 part of the left petal is missing.

5 Conclusion

We have demonstrated that flower segmentations can be significantly boosted by using an image specific colour distribution, and that this distribution can be learnt automatically by fitting a general flower shape model.

We have concentrated here on segmenting out the flower. A similar method (general colour model, generic shape model) could now be used for segmenting out the leaves and stalk – making strong use of green in the colour model. The current model is not suitable for fields of flowers. For such cases a model that groups repeated shapes would be more appropriate, such as [14].

The geometric model we have introduced here is a step towards extracting petal structure. The next step is flower classification based on petal shape and configuration.

Acknowledgements

This work was funded by EC Marie-Curie Training Network VISIONTRAIN.

References

- [1] <http://www.robots.ox.ac.uk/~vgg/data/flowers/>.
- [2] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *CVPRW*, volume 4, page 46, 2004.
- [3] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, volume 2, pages 105–112, 2001.
- [4] M. Das, R. Manmatha, and E. M. Riseman. Indexing flower patent images using domain knowledge. *IEEE Intelligent Systems*, 14(5):24–33, 1999.
- [5] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. CVPR*, 2005.
- [6] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *Proc. CVPR*, pages 335–344, 1988.
- [7] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *Proc. BMVC.*, volume 2, pages 264–271, 2003.
- [8] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. ECCV*, volume 4, pages 581–594, 2006.
- [9] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- [10] E. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proc. ACM SIGGRAPH*, pages 191–198, 1995.
- [11] M-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proc. CVPR*, volume 2, pages 1447–1454, 2006.
- [12] T. Saitoh, K. Aoki, and T. Kaneko. Automatic recognition of blooming flowers. In *Proc. ICPR*, volume 1, pages 27–30, 2004.
- [13] J. Shotton, J. Winn, C Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. ECCV*, pages 1–15, 2006.
- [14] T. Tuytelaars, A. Turina, and L. Van Gool. Noncombinatorial detection of regular repetitions under perspective skew. *IEEE PAMI*, 25(4):418–432, 2003.
- [15] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. ICCV*, pages 756–763, 2005.