Figure-ground segmentation by transferring window masks

Daniel Kuettel
Computer Vision Laboratory
ETH Zurich
dkuettel@vision.ee.ethz.ch

Vittorio Ferrari
School of Informatics
University of Edinburgh
vferrari@staffmail.ed.ac.uk

Abstract

We present a novel technique for figure-ground segmentation, where the goal is to separate all foreground objects in a test image from the background. We decompose the test image and all images in a supervised training set into overlapping windows likely to cover foreground objects. The key idea is to transfer segmentation masks from training windows that are visually similar to windows in the test image. These transferred masks are then used to derive the unary potentials of a binary, pairwise energy function defined over the pixels of the test image, which is minimized with standard graph-cuts. This results in a fully automatic segmentation scheme, as opposed to interactive techniques based on similar energy functions. Using windows as support regions for transfer efficiently exploits the training data, as the test image does not need to be globally similar to a training image for the method to work. This enables to compose novel scenes using local parts of training images. Our approach obtains very competitive results on three datasets (PASCAL VOC 2010 segmentation challenge, Weizmann horses, Graz-02).

1. Introduction

Figure-ground segmentation is a fundamental operation in computer vision.

The task is to produce a binary segmentation of the image, separating foreground objects from their background [35]. A good figure-ground segmentation is a valuable input for many higher-level tasks. For example, object recognition techniques [4, 40] benefit from segmentation as they can compute shape descriptors. Human pose estimation techniques are often based on human silhouettes [20].

Figure-ground segmentation has recently been addressed successfully by interactive segmentation works [6, 26, 28, 35]. Typically these works require the user to manually guide the algorithm by giving an indication of the location of objects in the image (often in the form of a rectangle around each object [6, 26, 35]). The segmentation process is often casted as minimization of a binary, pairwise energy function where variables are pixels. The unary potentials estimate the likelihood for a pixel to be foregound [6, 9, 35, 43], according to appearance models, which are typically *derived from the user input*.

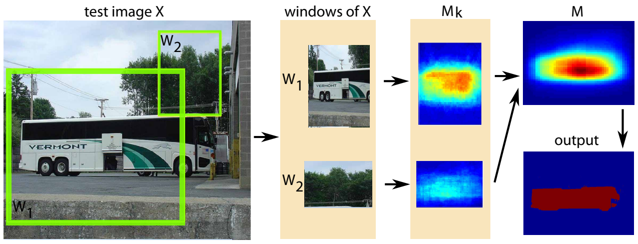

Figure 1. An example showing important parts of our pipeline. Windows (W_1 and W_2) are extracted from the test image X. Segmentation is transferred for windows using nearest neighbors, resulting in soft masks (M_1 and M_2). The final segmentation is based on an energy minimization that takes cues from the merged mask M. See sec. 2 for details.

In this paper we propose a novel approach for figure-ground segmentation which is based on minimizing an energy function of the same form as interactive segmentation works. However, our approach is fully automatic and requires no user input. The key idea is to *transfer segmentation masks* from a supervised training set to the test image. The transferred segmentation masks are then used to derive the unary potentials of the energy function of the test image. Importantly, the transfer process is not based on the global similarity between the test image and the training images (as done in [34] or in other areas such as inpainting [18] or image tagging [17]). Instead we first extract candidate windows likely to contain foreground objects [2] and then transfer masks from training windows that are visually similar to windows in the test image. The intuition is that visually similar windows often have similar segmentation masks. As these windows exhibit less variability than whole images and are often centered on objects, they form much better support regions for segmentation transfer. Afterwards, the energy minimization stage combines the local evidence of all windows into a coherent global segmentation of the whole test image.

In an extensive experimental evaluation on the PASCAL VOC 2010 segmentation challenge [13], the Weizmann Horses dataset [7], and the Graz-02 dataset [31] we show that (i) our segmentation transfer scheme outperforms GrabCut [35] initialized from a fixed box in the image center; (ii) transferring based on windows is better than based on the whole image; (iii) each component of the model contributes considerably to segmentation performance; (iv) we achieve results competitive with state-of-the-art approaches [1, 5, 11, 16, 28, 31] and in particular we outperfom the very recent technique of [34].

1.1. Related Work

Object segmentation. Single-class, fully supervised segmentation techniques aim at separating instances of an object class from their background (e.g. horses, faces, cars [5, 7, 21]). They are supervised in that the training set shows images of other instances of the class along with their binary segmentations. There are also multi-class fully supervised segmentation methods, which assign a label from a predefined set of classes to each pixel (e.g. cow, bird, car [25, 38]). Our work is related to both strands, as we aim at a binary segmentation separating all object classes from all kinds of background. However, we do not distinguish between different object classes, our techniques is class-agnostic. It does not learn explicit models for each class, but instead directly transfers segmentation masks of individual training instances based purely on visual similarity.

Interactive segmentation [6, 37] has been thoroughly researched since the very popular GrabCut work [35]. Most of these approaches minimize a binary pairwise energy function whose unary potentials are determined by appearance models estimated based on user input on the test image. Our approach builds on their energy formulation, but is fully automatic. The user input is replaced by our novel segmentation transfer mechanism.

More distantly related are works on weakly supervised segmentation, where the training images are annotated by the labels of the classes they contain, but pixel-level labels are not given [3, 42, 45].

Candidate object windows. As spatial support for our segmentation transfer operations, we use a set of candidate windows likely to contain objects, detected by the technique of [2]. However, there are also some other methods to obtain such candidates [11, 12, 41] and we believe they could form a valid alternative.

Annotation transfer by nearest neighbors. The core operation of our method is to transfer segmentation masks from *windows* in training images to visually similar windows in the test image. This is related to previous works that transfer annotations between images based on their *global* similarity, [17, 18, 29, 34, 36] as done in inpainting [18], image tagging [17], and scene parsing [29]. Malisiewicz et al. [30] proposes to employ per-exemplar SVMs to find neighbors for transfer, instead of simply measuring appearance similarity. Their idea could be incorporated in our pipeline as well. The previous work most related to ours is [34], which also transfers segmentation masks, but between images rather than windows. As we argue in sec. 4, (object) windows offer better spatial support for segmentation transfer.

2. Overview

We give here an overview (fig. 2) of our novel approach to figure-ground segmentation. The task is to label each pixel in a test image X as either foreground or background. The training data consists of images annotated with foreground-background masks. We first detect windows $\{W_k\}$ likely to contain foreground objects on all training images as well as on the test image using [2]. For each W_k in X, we then *transfer segmentation masks* from the training windows with the most similar appearance (nearest neighbors in appearance space, fig. 3 'window neighbors'). The key intuition is that visually similar windows often have similar segmentation masks. The last stage integrates the information transferred via individual candidate windows. Since the candidate windows are overlapping, they can now combine their local evidence into a coherent global segmentation of X. For this we apply a single global graph-cut segmentation to the entire image, with unary potentials tailored to each individual pixel, as derived from the candidate windows containing it (fig. 3 'M'). In this fashion, while the segmentation transfer operation is based on individual candidate windows, the final segmentation integrates information from all of them (fig. 3 'output').

In sec. 3, we introduce the segmentation energy model we use for the test image X. Sec. 4 explains how we find similar training windows to windows in X. Sec. 5 details how we transfer the segmentation masks from these appearance neighbors to form the unary potentials of the segmentation energy model for the test image. We conclude with an experimental evaluation in sec. 6.

3. Segmentation model

We cast the task of segmenting a test image X as a pixel labelling problem. Each pixel $x_i \in X = \{x_1, x_2, \ldots, x_N\}$ should be labelled as either foreground $c_i = 1$ or background $c_i = 0$. Hence, a labelling $C = \{c_1, c_2, \ldots, c_N\}$ represents a segmentation of X. An energy function is defined over pixels and their labels, and the optimal labeling is found by minimizing the energy over all possible labelings. This general approach is very popular in the segmentation

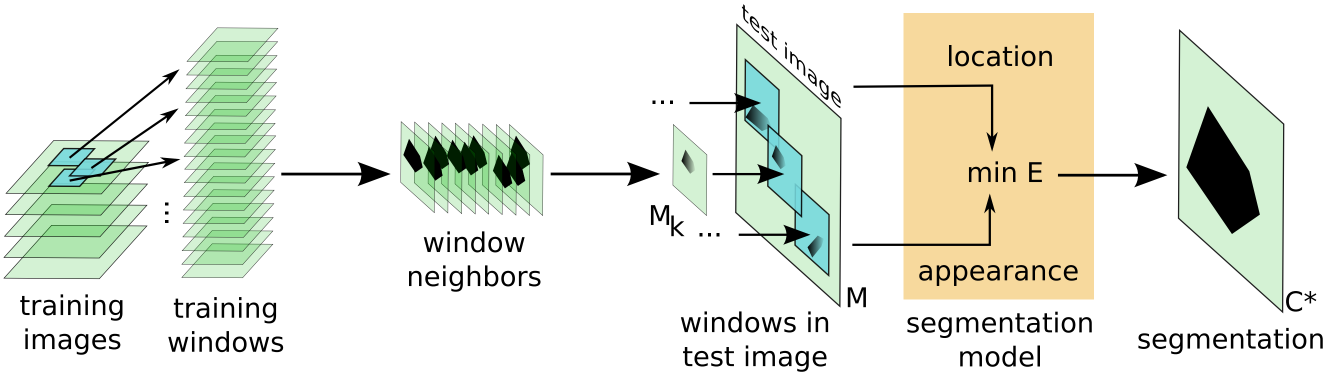

Figure 2. Overview of our approach. Windows from the training images with similar appearance to windows in the test image transfer their segmentation masks. These are then used to derive location and appearance unary potentials in the segmentation model of the test image. The final segmentation is obtained by minimizing the energy via graph-cuts.

community [7, 9, 19, 35, 39, 43]. We adopt the following class of energy functions

$$\begin{array}{rcl} E(C) & = & U(C) + V(C) \\ U(C) & = & \sum_i u_i(r_i) \\ V(C) & = & \sum_{ij\in\mathcal{E}} v_{ij}(c_i, c_j) \end{array} \qquad (1)$$

where i indexes over the pixels in the image; u_i and v_{ij} are unary and pairwise potentials; and $\mathcal{E}$ is the set of edges connecting pixels in a 8-neighborhood grid. The segmentation of X is defined as the optimal labeling

$$C^* = \arg\min_C E(C) \qquad (2)$$

The global optimum for this class of energy functions can be found efficiently using graph-cuts [9, 10, 23] as long as the pairwise potentials are submodular (as in our case).

For the pairwise potential, we define

$$v_{ij}(c_i, c_j) = \gamma d^{-1}(i,j)[c_i \neq c_j]e^{-\beta|x_i - x_j|^2} \qquad (3)$$

As in [6, 9, 35, 38, 43] this potential encourages smoothness by penalizing neighboring pixels taking different labels. The penalty depends on the color contrast between the pixels, being smaller in regions of high contrast (image edges).

The unary potentials is the novel element in this paper, as it carries the segmentation transfer information

$$u_i(c_i) = -\log A(x_i|c_i) - \log L_i(c_i) \qquad (4)$$

This potential evaluates how likely a pixel x_i is to take label c_i, according to an appearance model A and a location model L. The appearance model A computes the probablity of pixel x_i being foreground ($c_i = 1$), or background ($c_i = 0$). The appearance model parameters need to be estimated beforehand in some fashion. This estimation is crucial to good perfomance. In interactive segmentation works [6, 35, 44] this is achieved by *manually* drawing, *e.g.*, a rectangle around the object of interest, and then estimating A from the pixels inside vs outside the rectangle. In other works, an initial guess for the segmentation of the object is manually designed and hardwired into the system (e.g. for persons in [15]). In this paper instead, we propose a generic and robust approach that estimates appearance models by *transferring segmentations* from training images (sec. 5.2).

The location model L_i computes the probablility of pixel x_i to be foreground based on its location in the image. We derive a location model tailored to each individual pixel x_i by combining the segmentations transferred through all candidate windows containing it (sec. 5.1).

In summary, A and L are the novel elements in our approach. They carry information from our segmentation transfer scheme into the segmentation model of the test image.

4. Finding similar windows

In this section we explain how we retrieve training window similar to windows in the test image. The main intuition of our work is that similar windows often have similar similar segmentation masks. Therefore, we rely on nearest neighbors in the appearance space to transfer segmentation masks.

Image-level neighbors. Our idea is related to previous works that transfer annotations between images based on their *global* similarity [17, 18, 29, 36]. For our purpose however, the quality of these image-level neighbors is not good enough. For many test images, the most visually similar training images have very different figure-ground segmentations. See fig. 3 and 4 for an illustration. This happens because there is too much variability at the level of the whole image. In fig 3, the nearest neighbor training images show road scenes but the foreground objects do not match the motorbikes in the test image. In fig 4, the nearest neighbor training images do contain a sheep-on-grass. For the

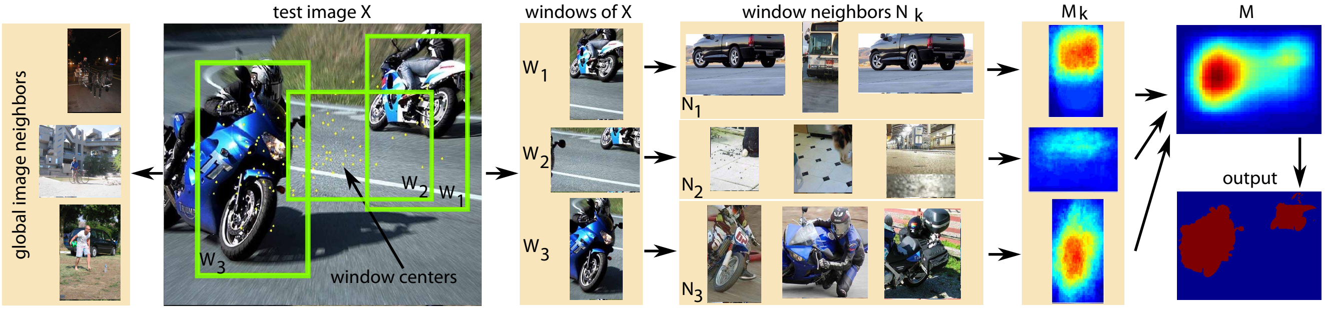

Figure 3. An example of the full pipeline. Three windows (W_1, W_2, and W_3) are highlighted in the test image X (out of 100, the centers of the others are displayed as dots). To the right of X we show the three extracted windows along with their nearest neighbors $\mathbf{N}_i$ in the training set. For W_1, the neighbors are a good match for segmentation transfer, even though the window does not cover the motorbike perfectly. This results in an accurate transfer mask M_1. W_2 on the other hand is not on an object. Its neighbors correctly transfer an almost void mask M_2. W_3 is tightly centered on an object and gets very good neighbors. On the rightmost column, the M_k from all windows are integrated into a soft mask M for the whole image, which is used to derive the unary potentials of the segmentation model and ultimately the final segmentation output (below). On the leftmost column we show the nearest neighbors of X at the level of the whole image. They are loosely related to X, being street scenes, but do not offer similar segmentation masks.

idea of segmentation transfer to work, we need much better neighbors, with truly similar pixelwise segmentations.

Window-level neighbors. To overcome the limitations of image-level nearest neighbors, we work instead at the *window level*. First we detect windows likely to contain an object using the 'objectness' technique of [2]. It tends to return more windows covering an object with a well-defined boundary in space, such as cows and cars, rather than amorphous background elements, such as grass and sky. In our experiments, sampling only 100 windows per image already covers most foreground objects (fig. 3 and 4). We extract these windows for all training images and for the test image. Because many such windows are centered on a foreground object, they *exhibit far less variability than whole images*. This leads to retrieving much better neighbors, whose segmentation masks match better the test image, and therefore are more suitable for transfer. Fig. 3 shows two motorbikes in the test image. The nearest neighbor windows accurately depict similar vehicles in similar poses, resulting in well matching foreground masks. Fig. 4 shows two sheeps in the test image. The nearest neighbors of the windows contain well matching foreground masks.

Given a new test image, we compare each test window W_k to all training windows W_l. The set $\mathcal{N}_k$ containing the segmentation masks of the top 100 training windows most similar to W_k is passed on to the next processing stage.

5. Segmentation transfer

In this section we explain how we use the segmentation masks of the training windows retrieved in the previous section to derive the unary potentials of the segmentation energy for the test image (eq (4)).

5.1. Location model L_i

We want the location model L_i to convey a sense of the likely segmentation of a pixel based only on its location within the image, so that $L_i(c_i = 1)$ gives the probability of pixel at location i to be foreground. We construct the L_i for each pixel i from the segmentation masks transferred via all windows containing it.

Soft masks for windows. Following sec. 4, for each test window W_k we have a set $\mathcal{N}_k$ containing the segmentation masks of neighbors from the training set. We now compute a soft segmentation mask M_k for each W_k as the pixelwise mean of the masks in $\mathcal{N}_k$. For this, all masks in $\mathcal{N}_k$ are resized to the size of W_k in both their width and height dimensions. In this aligned space, a pixel value in M_k corresponds to the probability for it to be foreground in $\mathcal{N}_k$ (fig. 3, M_k).

Soft mask for the test image. We now integrate the M_k for all windows into a single soft segmentation mask M for the test image X. For each window we place its soft mask M_k at the image location defined by W_k. The soft mask M of the test image is the pixelwise mean of these placed masks. A pixel value in M is the probability for it to be foreground, according to all transferred segmentations (fig. 3, M). Therefore, we define the location model as

$$\begin{aligned} L_i(c_i = 1) &= M(i) \\ L_i(c_i = 0) &= 1 - M(i) \end{aligned} \tag{5}$$

Integration effects. As discussed in sec. 4, the quality of the neighbors improves when retrieving them based on windows rather than the whole image. Furthermore, here we integrate the soft foreground masks of the individual

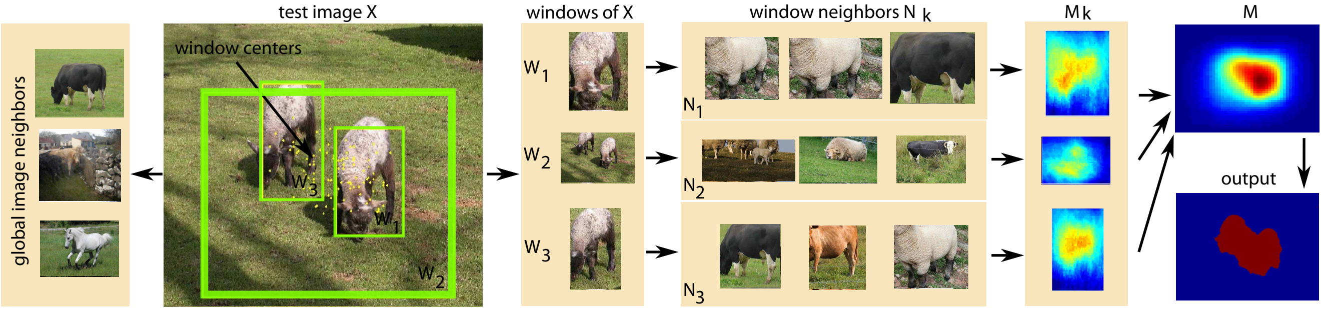

Figure 4. A second example of the full pipeline. See caption of fig. 3.

windows, which leads to even more robust results. The key observation is that we extract many *overlapping* windows (100 per image). A certain window might not have good neighbors in the training set, leading to an inaccurate or even completely incorrect mask transfer M_k. However, other windows overlapping with it will probably have good neighbors. The integration step above will diminish the effect of the inaccurate M_k. A second interesting effect happens for windows not covering a foreground object, e.g. a patch of road/grass/sky (fig. 3, W_2). This does not pose a problem to our approach, as the training images are decomposed in the same type of windows [2] (sec. 4). Therefore, a background window will probably have appearance neighbors on backgrounds in the training images, which results in correctly transferring a 'void' segmentation mask (fig. 3, M_2). In a summary, our approach is fully symmetric over foreground and background windows.

5.2. Appearance model A

The second part of the unary potential in eq. (4) is the appearance model A. The form of our appearance model is similar to the one in [35]. It consists of two gaussian mixture models (GMM), one for the foreground A_1, one for the background A_0

$$\begin{aligned} A(x_i|c_i = 1) &= A_1(x_i) \\ A(x_i|c_i = 0) &= A_0(x_i) \end{aligned} \qquad (6)$$

Each GMM has 5 components. Each component is a full-covariance gaussian over the RGB color space. A pixel $x_i \in X$ is represented as a vector with 3 components.

Parameter estimation. The crucial issue is how to set the parameters of A. Obviosuly, they should be adapted to the test image X, to capture the appearance of its particular foreground objects and background materials. In interactive segmentation [6, 35, 44] the user is asked to draw a rectangle around the foreground object. The appearance model can then be estimated from the pixels inside vs outside it. In this work instead we estimate the appearance models automatically, based on the segmentation transfer idea. We start from the soft segmentation mask M of X, constructed in sec. 5.1. Then we threshold M at a value t_a which we learn so to optimize the score on the training data. Finally, A_0 is estimated using all pixels i with $M(i) < t_a$ and A_0 using all pixels with $M(i) >= t_a$.

Effect of appearance model. As discussed in sec. 5.1, the location model gives a rough estimation of the foreground location. However, the location model acts only on individual pixels, so starting from imperfect segmentation transfers, optimizing eq. (2) will not improve the segmentation substantially. Instead, the appearance model is estimated from information over larger image region, and can transfer it across distant parts of the image. In the top example of fig. 5, the appearance model can learn that white is definitely a foreground color in this image, and then alter the unary potential of other pixels toward foreground, even if the transferred segmentation mask M suggests it should be background.

6. Experiments

6.1. Datasets.

We present experiments on three datasets: the PASCAL VOC 2010 segmentation challenge [13], Weizmann horses [7], and Graz-02 [31, 33].

PASCAL VOC 2010 (segmentation challenge). This is one of the most challenging datasets for segmentation and it contains real-world consumer images from Flickr. Many images have multiple foreground objects appearing at a variety of scales and locations. The dataset is annotated with pixelwise segmentations of 20 different object classes. However, for our figure-ground segmentation task, we fuse them all into foreground. The public part of the dataset amounts to 1928 images, split evenly into training and validation parts. As suggested [11, 13], we use the training part as the training set, and test on the validation part.

Weizmann horses. This popular dataset contains 328 images of horses in various poses and backgrounds [5, 7, 8, 22, 24, 28]. It is easier than PASCAL VOC as most horses are rather large and centered in the image. However, it offers a complementary experimental setup, as there is only one object class. The dataset is annotated with ground-truth segmentation masks for all images. We follow the evaluation protocol of [5] and split the data into $2/3$ for training and $1/3$ for testing.

Graz-02. This dataset contains three subsets for the classes bike, car, and person. The objects appear at a variety of image locations and in various poses, which makes the dataset very challenging. The annotation is provided as a binary segmentation mask for each image. Each subset is annotated with one class only, but an image often contains more than one instance of the class. We follow the protocol in sec. 4.1 of [31], where for each class there are 150 images for training and 150 images for testing.

6.2. Implementation details.

In all experiments we compute GIST [32] *inside each window* to describe its appearance and we compare GIST descriptors with the L2 distance (sec. 4). Our segmentation transfer method (sec. 4 and 5) yields the two unary potentials A, L of the energy (1). For the pairwise potentials, we set $\gamma = 50$, as suggested in [6]. β is set to the mean squared distances of the pixel colors in the test image (see [35], eq. 5). We then apply the iterative optimization scheme of [35], which alternates between estimating the segmentation C using graph-cut (2), and updating the appearance models A based on C. As shown in [35], this improves performance over keeping the appearance models fixed to the initialization. During the iterations, the location potential L remains fixed.

6.3. Results.

PASCAL VOC 2010. We measure performance with the intersection-over-union (IoU) score [13]: $O(S, G) = \frac{S \cap G}{S \cup G}$, where S is the set of foreground pixels produced by the algorithm, and G is the ground-truth set of foreground pixels. This score penalizes both over- and under-segmentation and is scale invariant. It ranges from 0 (worst) to 1 (best). We report the average IoU over all test images (tab. 1).

As a baseline, we compare to plain grabcut [35] applied to a full test image. As initializiation area to estimate the appearance models, we use a box in the middle of the image, occupying 50% of its area (first row). This baseline performs modestly at 30% IoU, as the initialization is usually not aligned with the objects in the image.

As a second baseline we transfer the segmentation masks of the 5 most similar training images to the test image, according to the GIST [32] descriptor compute *globally* on the whole image. The 5 masks are averaged and thresholded to produce the final segmentation. We use a threshold of 0.5. This baseline reaches 27% IoU (second row). In contrast, our segmentation transfer based on objectness windows [2] already gets to 40% IoU (fifth row, obtained by directly thresholding the soft mask M produced in sec. 5.1, the threshold is set so as to maximize performance on the training set). This demonstrates our claim that segmentation transfer is substantially more effective when supported by windows than by the whole image. Moreover, we also tried switching off segmentation transfer while keeping the candidate windows. In this case we use a solid block of pure foreground as M_k. The resulting soft mask M is thresholded the same way as before. This performs at 35% IoU (fourth row), which demonstrates the benefits of transferring a segmentation mask from visually similar training windows, which is tailored to the local image content of the test image. It also shows that, while objectness windows are a valuable platform on which we build our approach, they do not solve the problem on their own.

In our complete framework, we use the transferred segmentation to derive the appearance and location unary potentials of the segmentation model of the test image (see sec. 5.1 and sec. 5.2). The segmentation model with the appearance potential A alone improves direct thresholding by 4% to an IoU of 44% (second-last row). Our full segmentation model, including also the location potential L, improves further by another 4%, reaching a final result of 48% IoU (last row). Compared to standard GrabCut with a default image-center initializiation (first row), this is a substantially better results, which validates our segmentation transfer technique as a useful way to automatically set the unary potentials of GrabCut.

We also compare to [11], which generates multiple segments intended to cover objects. It also ranks the returned segments, so that segments deemed more likely to cover objects are ranked higher. We use their publicly available code [1], which already contains a model trained for the VOC 2010 dataset. We apply it to the test set and compute the IoU scores when using the union of the top K ranked segments as foreground prediction. We evaluate the performance for all $K \in \{1, \ldots, 100\}$ and report the best result in tab. 1. Our method delivers considerably higher IoU, although the comparison is only indicative as the goal of [11] is not to produce a single figure-ground segmentation for an image, but a pool of plausible ones.

Finally, we also compare to the very recent, state-of-the-art approach of [34], as it reports results in exactly our setting. As the table shows, our method achieves 2% better IoU, confirming its high performance.

Weizmann Horses. We quantify performance as pixel-wise accuracy, as suggested in [5]. It measures the percentage of pixels classified correctly into foreground or background. In tab. 2 we compare to the recent, state-of-the-art work of [5]. Our pixewise accuracy is essentially the same,

[1] http://sminchisescu.ins.uni-bonn.de/code/cpmc/

model	IoU (%)
GrabCut 50% image center	30
Global image neighbor transfer	27
CPMC [11] (best K)	34
Rosenfeld [34]	46
windows (solid box)	35
windows (segmentation transfer)	40
segmentation model (appearance only)	44
segmentation model (appearance+location)	48

Table 1. Results on the PASCAL VOC 10 dataset. The last row corresponds to our full method. See main text for discussion.

model	pixelwise accuracy (%)
our method	94.7
Bertelli et al. [5]	94.6
Levin & Weiss [28]	95.5
Cosegmentation [22]	80.1

Table 2. Results on the Weizmann Horses dataset.

although we do not employ a separate sliding-window horse detector [14], making our method simpler and more unified. As a reference we also include the performance of [28]. While their score is slightly higher, their method relies on ground-truth bounding-boxes on the horses in the test images. Our method instead is fully automatic. Finally, we also report the score of an unsupervised segmentation work [22] based on cosegmentation (i.e. they do not require training images with segmented horses). These results place favorably our approach in the context of the state-of-the-art.

Graz-02. Following [31], we evaluate performance based on pixelwise precision/recall (PR). Since [31] outputs a soft segmentation mask, they compute precision/recall for varying thresholds and then report the equal error rate (EER). Instead, our method outputs a binary segmentation, so we directly compute its PR. From PR, we compute the F-measure as $F = 2pr/(p + r)$. In tab. 3 we compare to other recent works. While we outperform [1, 16, 31], the results of [27] are very impressive on this dataset.

7. Conclusion

We have presented a novel technique for figure-ground segmentation based on the idea of transferring segmentation masks from *windows* in the training images that are visually similar to windows in the test image. The transferred masks are used to derive the location and appearance unary potentials of a segmentation energy defined over the whole test image. The scheme is fully automatic, class-agnostic and dynamically adapts to the content of novel test images. The experiments demonstrate the high performance of our approach on challenging datasets.

model	car	people	bike	average
Marszalek & Schmid [31]	53.8	44.1	61.8	53.2
Fulkerson et al. [16]	54.7	51.4	66.4	57.5
Aldavert et al. [1]	62.9	58.6	71.9	64.5
Lempitsky et al. [27]	83.7	84.9	82.5	83.7
our method	74.8	66.4	63.2	68.1

Table 3. Results on the Graz-02 dataset. We report the F-measure (%) for each class and the average over classes.

References

[1] D. Aldavert, A. Ramisa, R. Lopez de Mantaras, and R. Toledo. Fast and robust object segmentation with integral linear classifiers. In *CVPR*, 2010.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[3] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, 2007.

[4] S. Belongie, J. Malik, and J. Puzicha. Matching with shape contexts. *IEEE Trans. on PAMI*, 24(4):509–522, 2002.

[5] L. Bertelli, T. Yu, D. Vu, and S. Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*, 2011.

[6] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, 2004.

[7] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *CVPR*, 2004.

[8] E. Borenstein and S. Ullman. Learning to segment. In *ECCV*, 2004.

[9] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.

[10] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2004.

[11] J. Carreira and C. Sminchisescu. Constrained parametric min cuts for automatic object segmentation. In *CVPR*, 2010.

[12] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[13] M. Everingham et al. The PASCAL Visual Object Classes Challenge 2010 Results, 2010.

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[15] V. Ferrari, M. Marin, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[16] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *ECCV*, 2008.

[17] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[18] J. Hays and A. Efros. Scene completion using millions of photographs. In *SIGGRAPH*, 2007.

[19] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. on PAMI*, 23(10):1075–1088, 2001.

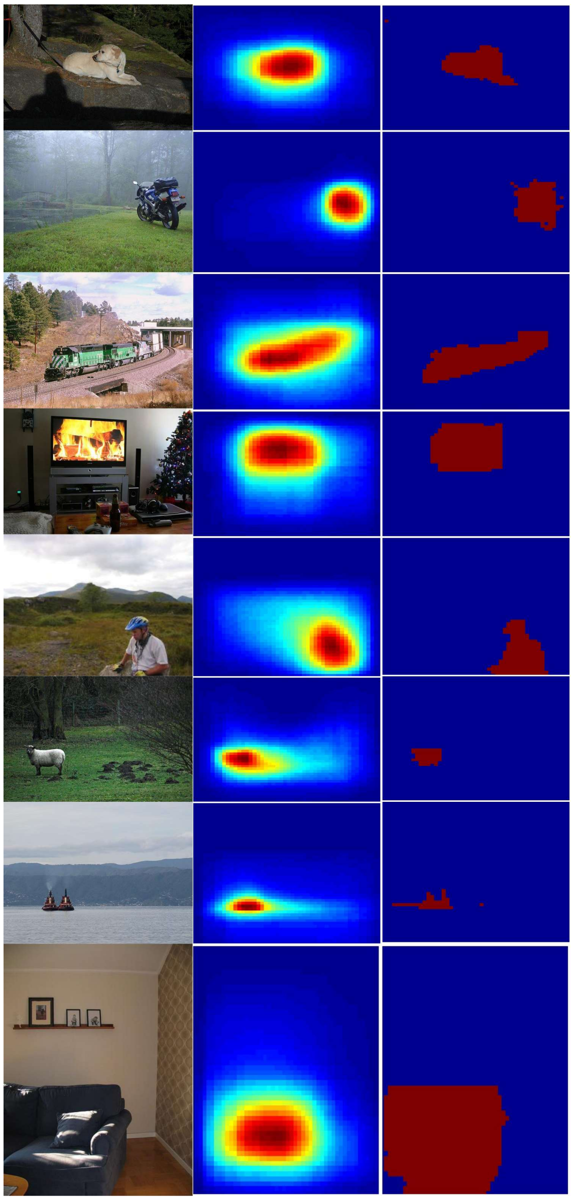

Figure 5. Example results from the PASCAL VOC 10 dataset. Left: input image. Middle: mask M (sec. 5). Right: final segmentation output by our method (sec. 3). Note how M localizes objects well, even when they are not centered in the image. The last example (sofa) contains objects in the top half (picture frames). The windows generated by [2] also cover these picture frames. However, since they are not annotated as foreground in the PASCAL VOC training data, they are not reflected in M.

[20] H. Jiang. Human pose estimation using consistent max-covering. In *ICCV*, 2009.

[21] N. Jojic, A. Perina, M. Cristani, V. Murino, and B. Frey. Stel component analysis: Modeling spatial correlations in image class structure. In *CVPR*, 2009.

[22] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image cosegmentation. In *CVPR*, 2010.

[23] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Trans. on PAMI*, 26(2):147–159, 2004.

[24] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005.

[25] L. Ladicky, C. Russel, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[26] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[27] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. In *NIPS*, 2011.

[28] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[29] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009.

[30] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[31] M. Marszalek and C. Schmid. Accurate object localization with shape masks. In *CVPR*, 2007.

[32] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[33] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. In *IEEE Trans. on PAMI*, pages 416–431, 2006.

[34] A. Rosenfeld and D. Weinshall. Extracting foreground masks towards object recognition. In *ICCV*, 2011.

[35] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[36] B. Russel, A. Torralba, C. Liu, and R. Fergus. Object recognition by scene alignment. In *NIPS*, 2007.

[37] T. Schoenemann and D. Cremers. Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In *ICCV*, 2007.

[38] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[39] A. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *ICCV*, 2007.

[40] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 63(2):113–140, 2005.

[41] K. Van de Sande, U. J.R.R., T. Gevers, and A. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[42] J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *CVPR*, 2007.

[43] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[44] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, 2005.

[45] J. Winn and N. Jojic. LOCUS: learning object classes with unsupervised segmentation. In *ICCV*, 2005.