Interactive Texture Segmentation using Random Forests and Total Variation

Jakob Santner
santner@icg.tugraz.at
Markus Unger
unger@icg.tugraz.at
Thomas Pock
pock@icg.tugraz.at
Christian Leistner
leistner@icg.tugraz.at
Amir Saffari
saffari@icg.tugraz.at
Horst Bischof
bischof@icg.tugraz.at

Institute for Computer Graphics and Vision
Graz University of Technology
Graz, Austria

Abstract

Common methods for interactive texture segmentation rely on probability maps based on low dimensional features such as e.g. intensity or color, that are usually modeled using basic learning algorithms such as histograms or Gaussian Mixture Models. The use of low level features allows for fast generation of these hypotheses but limits applicability to a small class of images. We address this problem by learning complex descriptors with Random Forests and exploiting their inherent parallelism in a GPU implementation. The segmentation itself is based on a convex energy functional that uses weighted Total Variation regularization and a point-wise data term allowing for continuous foreground/background membership hypotheses. Its globally optimal solution is obtained by a fast primal-dual algorithm providing a reasonable convergence criterion. As a result, we present a versatile interactive texture segmentation framework. We show experiments with natural, artificial and medical data and demonstrate superior results compared to two recent approaches.

1 Introduction

Interactive image segmentation is the task of semi-automatically separating a foreground object $\mathcal{F}$ from image background $\mathcal{B}$. Usually, two steps are required to perform this task: (i) generation of a hypothesis describing the likelihood that a certain pixel belongs to $\mathcal{F}$, and (ii) regularization to prevent overfitting of the hypothesis. Results can be corrected by providing additional user input, leading to an iterative process. This process is illustrated in Figure 1.

Edge information can provide valuable information during regularization. Therefore, a wide range of segmentation algorithms are based on the Geodesic Active Contour (GAC) energy introduced by Caselles *et al.* [9]. In a continuous formulation, the GAC energy can be

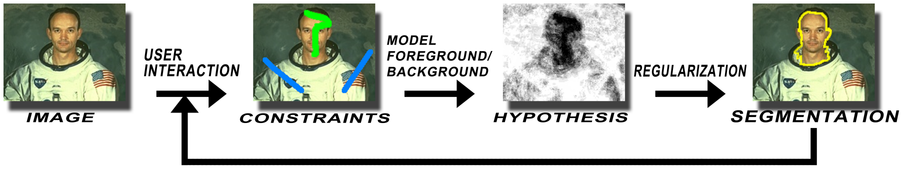


Figure 1: Interactive segmentation: Regions based on user constraints are modeled, the resulting hypothesis is used by the segmentation algorithm to obtain a discrete labeling. The user can iteratively correct the segmentation by providing additional constraints.

realized with the weighted Total Variation (TV) [8]. Image segmentation using the weighted TV was done e.g. in [8, 16, 25]. In the `TVSeg` framework, Unger *et al.* [25] used simple color histograms to create pixelwise three-state hypotheses combined with a variational minimization problem. Also discrete methods such as graph cuts [6] can be used to solve the GAC energy. Rother *et al.* [19] combined graph cuts with Gaussian Mixture Models (GMMs) learning color distributions in their `GrabCut` tool. Additionally, graph-based approaches for image segmentation were presented in [2, 10, 14, 24, 27]. Other work in the field of interactive segmentation comprises Friedland *et al.* [13], who clustered color signatures in the CIELAB space. They postprocessed their results by suppressing small regions and smoothing boundaries. Their tool SIOX (Simple Interactive Object Extraction) is part of the software packages GIMP and Inkscape. Bai and Sapiro [3] modeled $\mathcal{F}$ and $\mathcal{B}$ using kernel density estimation on LUV color features and used the resulting probability map to compute geodesic distances.

A good interactive segmentation framework provides accurate segmentation results with as little user interaction as possible in a reasonable amount of time. `TVSeg` and `GrabCut` are fast and easy to use, but both use solely color information to create their hypotheses (See figure 2). Han *et al.* [15] adapted Rother's [19] approach using a multi-scale nonlinear structure tensor (MNST) in addition to color. They showed promising results, although still being restricted to a single specific descriptor. Xiang et al. [28] used Topographic Independent Component Analysis to model filters for the extraction of texture features. They learned these features using Spectral Graph Partitioning and regularized these hypotheses using spatial median filtering.

The segmentation quality directly depends on the representation for $\mathcal{F}$ and $\mathcal{B}$. A huge amount of features exist that were shown to be well suited to model complex appearance or shape, but most of them are high-dimensional, thus, computationally intensive. In order to model strong hypotheses based on different high-level features, we need a very efficient learning algorithm capable of handling arbitrary input data. Random Forests (RFs) introduced by Breiman [7] are very fast to compute while yielding state-of-the-art performance in machine learning and vision problems. Their inherent parallel structure dedicates them to be implemented on the GPU [23]. Several publications showed the applicability of RFs to different image features (e.g. [4]: HOGs, SIFTs , [22]: Textons, colour, filterbanks, HOGs, [23]: Filterbanks, generalized Haar-like features, rectangle sums, pixel differences). Additionally, RFs perform feature selection inherently which can be exploited to increase performance significantly by using multiple features. The segmentation approaches mentioned above use basic, low-level features to model $\mathcal{F}$ and $\mathcal{B}$ with simple machine learning algorithms. We propose the use of arbitrary features learned with Random Forests.

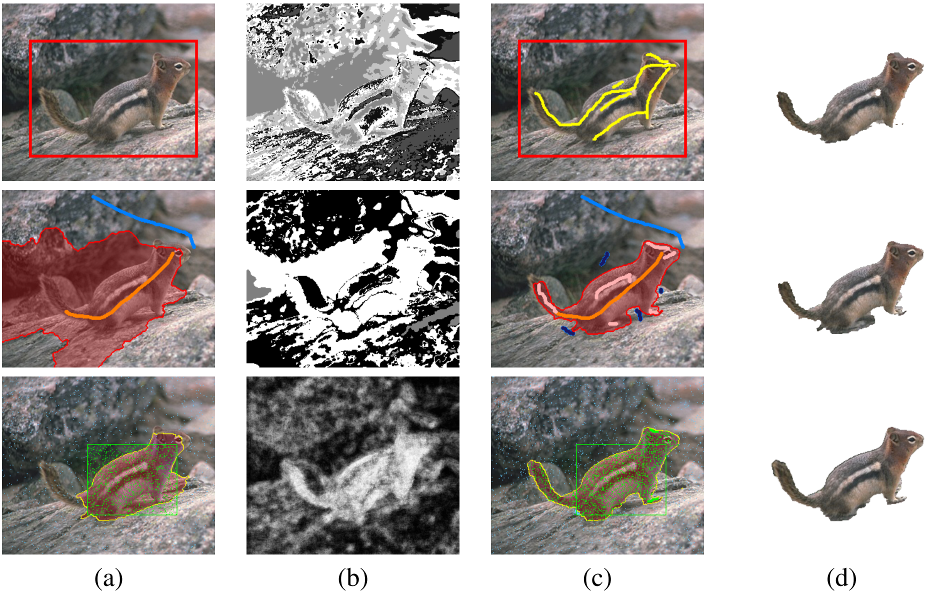


Figure 2: Using solely color information causes weak hypotheses where color is not descriptive. The first row shows `GrabCut`, the second `TVSeg` and the third row shows our approach. The algorithms are initialized interactively by giving a rough cue to the foreground object (a). Probabilistic models are trained and evaluated (b), that lead, combined with additional local constraints (c), to the final segmentation result (d). Our approach uses iterative refinement of the foreground samples in step (c). The hypotheses in column (b) were scaled such that dark areas correspond to $\mathcal{B}$ and light areas to $\mathcal{F}$

In the experiments, we compare our algorithm to the continuous approach `TVSeg` [25] and the graph based framework `GrabCut` [19]. As already shown in [25], continuous approaches have the advantage of low memory consumption, no discretisation errors and high parallelization potential. By implementing them on the GPU, fast minimization algorithms obtain speeds comparable to discrete methods. We therefore decided to use a variational approach that combines weighted TV with a more flexible data term and a faster minimization procedure than `TVSeg`. Our tool comprises the extraction of features, Random Forest classification and convex energy minimization, all implemented on a GPU. Using graphics processors speeds up the whole segmentation process to a few seconds and allows for convenient user interaction.

Method	Features	Hypothesis	Regularization
Unger *et al.* [25]	Color	Histograms	**TV**
Rother *et al.* [19]	Color	GMM	Graph Cut
Han *et al.* [15]	Color, MNST	GMM	Graph Cut
Our approach	**arbitrary**	**RF**	**TV**

Table 1: Comparison of methods, bold faced letters indicate GPU implementations.

The remainder of the paper is organized as follows: In Section 2, we explain RFs and the features we used for image description. Section 3 describes the segmentation model which is based on a convex variational formulation using weighted Total Variation and a pointwise data term. In Section 4, we compare our method to `TVSeg` and `GrabCut` and show results on natural, artificial and medical images. We finally conclude in Section 5. In this paper, we use images and groundtruth data of the Berkeley Segmentation Dataset [12, 17].

2 Random Forests

Random Forests introduced by Breiman [7] have become the method of choice for many computer vision applications. RFs combine Breiman's bagging [6] with randomized decision trees proposed by Amit *et al.* [1]. They are *fast*, inherently *parallel*, multiclass capable and robust against label noise while showing classification performance rates competitive to Boosting and SVMs (e.g. [4, 7]).

A Random Forest is a set of N binary decision trees $\{f_1,\cdots,f_N\}$ trained with an initial dataset $\mathcal{X}_\mathcal{T} \subseteq \mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{|\mathcal{X}_\mathcal{T}|},y_{|\mathcal{X}_\mathcal{T}|})\}$, $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^M$ and $y_i \in \mathcal{Y} = \{1,\cdots,K\}$, where M denotes the dimension of feature $\mathbf{x}_i$ and K denotes the number of classes. Each of the decision trees receives its dedicated training set by randomly subsampling with replacement of the entire forest training set $\mathcal{X}_\mathcal{T}$.

There are two types of nodes in a binary decision tree: (i) Split nodes represent a binary decision function propagating a sample to either the node's left or right child. (ii) Leaf nodes represent probabilities p_k for all classes in that particular node. During training a forest, each tree selects appropriate decision functions and assigns leaf node probabilities. For the evaluation, each sample is propagated through each tree resulting in a probability $p_n(k|\mathbf{x})$, for the n^{th} tree. These probabilities can be combined to a forest's joint probability

$$p(k|\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} p_n(k|\mathbf{x}). \tag{1}$$

To find a suitable binary decision function for a split node, a set of random functions is generated and evaluated on every sample in that node. The best hypothesis is selected according to a splitting criterion such as e.g. Gini impurity $\sum_{k=1}^{K} p_k(1-p_k)$ or Information Gain $-\sum_{k=1}^{K} p_k \log(p_k)$. Usually, the decision function only takes a small part of the inputs' dimensions into account, which amounts to inherent feature selection.

Split Node Decision Function Breiman suggested using linear combinations of randomly selected features in split nodes, which Bosch *et al.* [4] successfully used with high-level descriptors as HOG features or SIFT keys: A split node receives several input vectors $\mathbf{x}$ with different labels y. We first choose multiple hypotheses of weights w and threshold values θ randomly, then evaluate

$$\mathbf{x}^T\mathbf{w} \leq \theta \tag{2}$$

for all hypotheses and all input vectors and finally pick the best one according to the resulting Gini index. Most of the weights in $\mathbf{w}$ are zero, only a randomly selected subset ranges randomly between $[-1,1]$. The trees are usually grown to a certain depth without pruning.

Online Random Forests Usually RFs are trained in batch-mode, *i.e.*, the entire training data is available at once. However, in interactive segmentation, one does not want to retrain an entire classifier after each additional user input. Thus, online learning would be desirable.

Therefore, we additionally adopted the recently proposed on-line version of RFs [21]. Please note, however, that we skip the details here due to space limitations.

Using Graphics Processors RFs are inherently parallel: During training, each node at a certain tree level can be trained independently. During evaluation, every tree's probability for every sample $p_n(k|\mathbf{x})$ can be evaluated separately, even the combination $p(k|\mathbf{x})$ is independent for each sample. Sharp [23] was the first to implement RFs on a GPU using shaders with HLSL. We use a similar approach implemented using NVidia's CUDA.

Features All examples shown in this paper use pixel descriptions consisting of colour information, image patches (5x5 pixels, for each color channel) and basic HOGs [11] (3x3 cells of size 6x6, 180°, 9 orientation bins). We composed our final feature from several descriptors extracted from Gaussian smoothed images with different values of σ.

3 The Segmentation Model

In the previous Section, we described the use of Random Forests to model arbitrary image features in order to gain point-wise continuous foreground/background membership hypotheses. In this Section, we show how these hypotheses can be regularized based on a convex energy functional using weighted Total Variation.

Minimization Problem We propose to use the following convex minimization problem

$$\min_u \left\{ E_p = \int_\Omega g\,|\nabla u|\,dx + \lambda \int_\Omega u f dx \right\} , \tag{3}$$

with Ω the image domain, $f : \Omega \to \mathbb{R}$ the segmentation cue, and $u : \Omega \to \{0,1\}$ the binary labeling into foreground and background. As already shown in [8] we can make use of convex relaxation by letting u vary continuously in the interval $[0,1]$. Thus the energy in (3) becomes convex and we can find a globally optimal solution. By selecting an arbitrary levelset of u, the binary segmentation is obtained. We make use of the same edge detection function as `TVSeg` ($g(x) = \exp\left(-\alpha|x|^\beta\right)$, for some reasonable values of α and β) and also apply ROF denoising [20] on the input image to improve edge information. Of course for highly textured images, edge information becomes less important.

The product term (pointwise data term) has several advantages compared to the L^1 data term used in `TVSeg`. We are not restricted to binary values for the segmentation cue f. Instead f can be a continuous function and can be directly derived as a linear function of some probability measurement. Different cases can be distiguished: By setting $f(x) = 0$ the data term vanishes, and only the GAC energy is minimized. If $f(x) < 0$ the algorithm will try to make this pixel foreground. The smaller the value, the stronger the foreground constraint. Equally $f(x) > 0$ is interpreted as a background constraint. The positive parameter λ controls the amount of regularization that is applied. Setting $f(x) \to -\infty$ or $f(x) \to \infty$ is respectively interpreted as a hard foreground or hard background constraint. Compared to `TVSeg`, the simple product in the data term makes minimization easier and faster. Furthermore no approximation is needed, and we obtain the true minimum of (3).

Solution Recently primal-dual algorithms showed high performance gains for variational models [29, 30]. We therefore adapt these primal-dual approach to solve the minimization problem defined in (3). Using principles of duality, we arrive at the following primal-dual energy formulation:

$$\min_u \left\{ \sup_{||p||\leq g} \left\{ E_{p,d} = -\int_\Omega u\nabla\cdot p dx + \lambda \int_\Omega u f dx \right\} \right\} , \tag{4}$$

where $p : \Omega \rightarrow \mathbb{R}^d$ is the dual variable. By deriving the Euler-Lagrange equations for u and p, and by applying gradient descent and gradient ascent respectively, we arrive at the following iterative algorithm:

$$u^{n+1} = \Pi_{[0,1]} \left\{ u^n + \tau_p \left(\nabla \cdot p^n - \lambda f \right) \right\} , \tag{5}$$

$$p^{n+1} = \Pi_g \left\{ p^n + \tau_d \nabla u^{n+1} \right\} . \tag{6}$$

The projection $\Pi_{[0,1]}(u)$ in the primal update (5) can be simply realized by clamping u between 0 and 1. The reprojection in the dual update (6) can be realized as an Euclidian reprojection on a disc with radius g:

$$\Pi_g(p(x)) = \frac{p(x)}{\max\left\{1, \frac{||p(x)||}{g(x)}\right\}} . \tag{7}$$

Empirically, the scheme is stable, as long as $\tau_p \tau_d \leq \frac{1}{2}$, but we do not have a proof for that.

Primal and dual updates are iterated until convergence. In contrast to the projected gradient algorithm used by `TVSeg`, the primal-dual formulation gives us a meaningful convergence criterion by observing the primal-dual gap [30]. As the optimization problem is a saddle-point problem, the primal energy E_p presents an upper boundary of the true minimizer (3), and the dual energy E_d presents a lower boundary. Therefore we have to find the purely dual formulation of energy. For a fixed value of p, it is easy to see that the minimization problem form (4) can be solved by setting

$$u(x) = \begin{cases} 1 & \text{if} \quad -\nabla \cdot p(x) + \lambda f(x) < 0 \\ 0 & \text{else} \end{cases} \tag{8}$$

We can therefore rewrite the segmentation energy in the following purely dual formulation

$$\sup_{||p||\leq g} \left\{ E_d = \int_\Omega \min\left\{-\nabla \cdot p + \lambda f, 0\right\} dx \right\} , \tag{9}$$

The normalized primal-dual gap is now defined as

$$G(u,p) = \frac{E_p(u) - E_d(p)}{E_p(u)} . \tag{10}$$

The algorithm is terminated if the normalized primal-dual gap $G(u,p)$ falls below a certain threshold.

4 Experiments

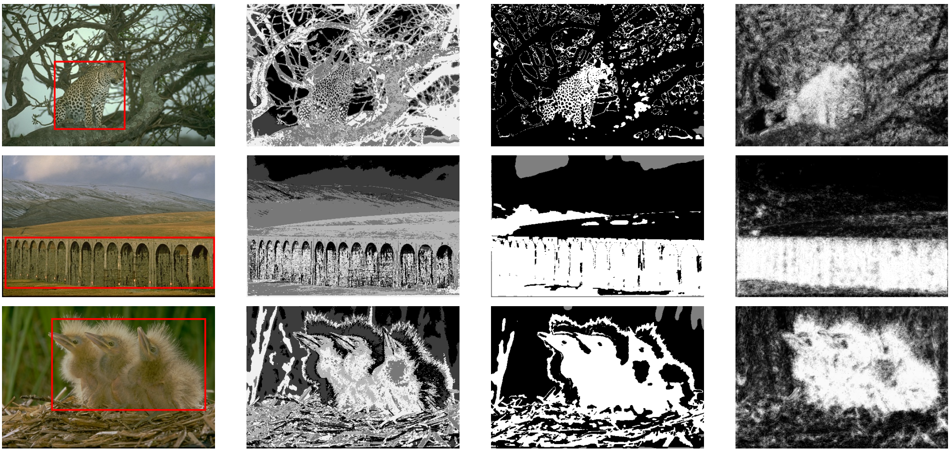
All experiments were conducted on a desktop PC featuring an Intel Q9450 (2.67 GHz) CPU and a GeForce GTX280 graphics adapter.

Framework, Performance First of all, features are computed for every pixel in the image. The extraction of HOGs, 5x5 patches and color features on a 500x350 image takes $\sim$35ms on our GPU. The user begins by labeling parts of $\mathcal{F}$ and $\mathcal{B}$: He can

- add foreground/background pixels to the RF training set with brush strokes or
- draw a rectangle over the object to segment. Randomly sampled points inside and outside the rectangle are used as training set.

Then the labeled samples are learned using a GPU-based RF implementation. Training a RF with 25 trees of depth 10 using 2000 samples takes $\sim$1s, the evaluation of this forest for every single pixel of a 500x350 image takes $\sim$140ms. The segmentation algorithm converges in less than 500ms for all images presented in this paper.

Hypotheses The quality of the final segmentation strongly depends on the discriminative power of the hypothesis. Experiments comparing the hypotheses of `GrabCut`, `TVSeg` and the proposed method show the expected superiority of our approach (see Figure 3).

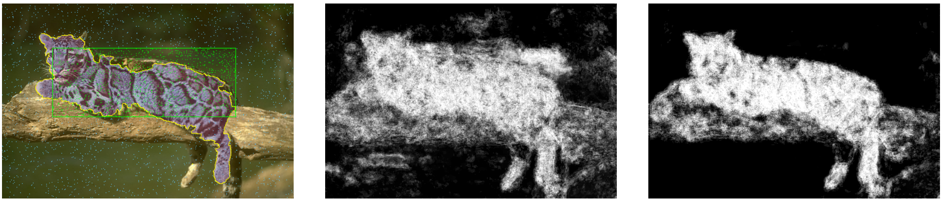


Initialisation | `GrabCut` | `TVSeg` | Our approach

Figure 3: Hypotheses on natural images using different methods: `GrabCut` and `TVSeg` fail to produce descriptive global hypotheses, i.e., there are always several parts of the image background having a good foreground probability. Our approach yields strong global hypotheses for all three images.

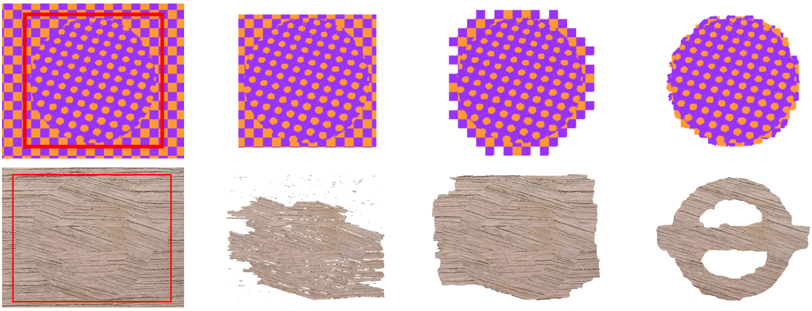
This figure mainly shows two things: (i) The incorporation of high-level features clearly strengthens the hypotheses and (ii) that RFs are less sensitive to label noise: Points labeled incorrectly are neglected to a certain amount. Additionally, the hypotheses can be improved by resampling training points iteratively using the previous segmentation as foreground constraint (See figure 4).

Artificial/Medical data Figure 5 compares results of our method to results of `GrabCut` and `TVSeg` on artificial data. This figure shows the failure of `GrabCut` and `TVSeg` on problems where the use of color alone cannot distinguish $\mathcal{F}$ from $\mathcal{B}$. Our approach is able to model the underlying textural structures and produces accurate segmentations.



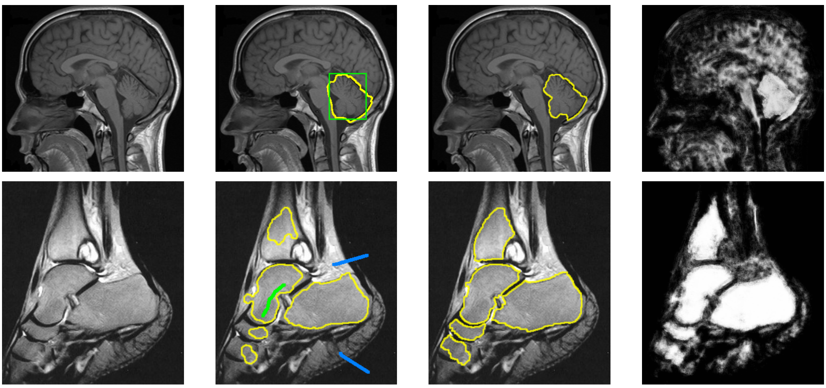
(a) (b) (c)

Figure 4: Random seed points are sampled from the initialization rectangle and learned, (a) yielding a dense probability map (b). Using the resulting binary segmentation to resample training points yields stronger hypotheses and thus better segmentation results (c).



Original `GrabCut` `TVSeg` **Our Approach**

Figure 5: Segmentation results on artificial data using different methods. Our approach and `GrabCut` are initialized with the rectangles depicted in the leftmost images, `TVSeg` is initialized with a few brushstrokes.



(a) (b) (c) (d)

Figure 6: Segmentation results using iterative refinement: The original image (a) is segmented using rectangular or brush stroke constraints (b). The intermediate segmentation is used to retrain the forest resulting in better hypothesis and segmentation (c,d).

Figure 6 shows segmentation results on medical data: We segment the cerebellum in an MRI recording of a human head and bone material in an MRI slice of an ankle. Here we use iterative refinement of the foreground labels using intermediate segmentation results.

Quantitative Evaluation on Natural Images In the following experiment, we compare the binary segmentations of `GrabCut`, `TVSeg` and our approach with hand-labelled segmentations of natural images taken from the Berkeley Segmentation Dataset [12, 17]. The hand-labelled segmentations offer several different labels per image. We merged labels to obtain a two-label foreground/background segmentation. This binary ground truth was compared to the segmentation results of the different algorithms on a per-pixel basis. The segmentations are shown in figure 7, the specific image/label numbers and the accuracy rates are stated in table 2.

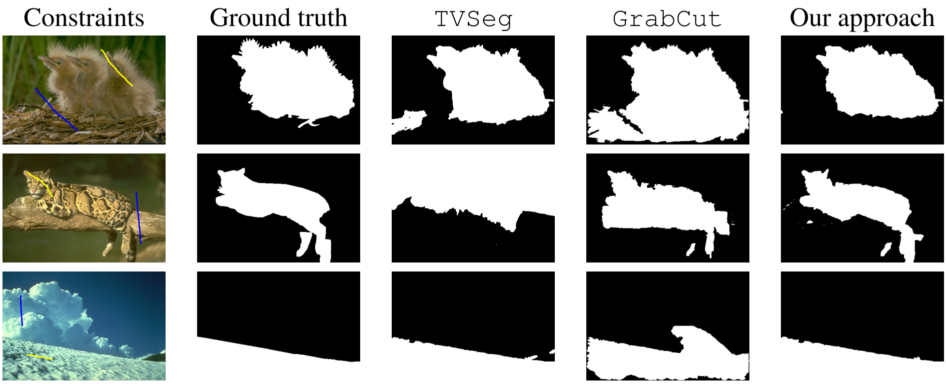
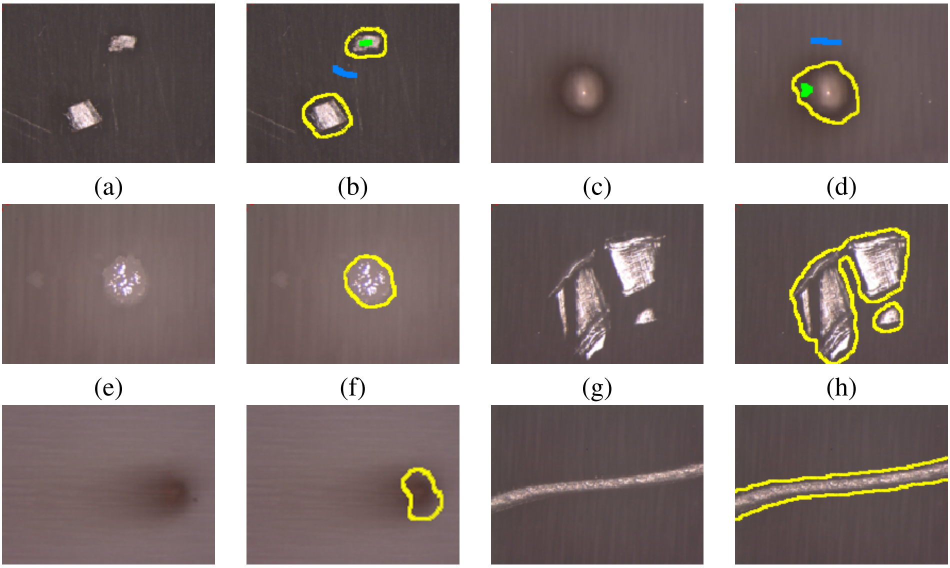


Figure 7: Quantitative evaluation of `TVSeg`, `GrabCut` and our approach: The ground truth is adapted from human labelled data of the Berkeley Segmentation Dataset. `TVSeg` and our approach were initialised only with constraints shown in the leftmost column, `GrabCut` is additionally constrained by a bounding rectangle. Accuracy rates are stated in table 2.

Image	Image Number	Label ID	`TVSeg`	`GrabCut`	**Our approach**
chicks	163085	1105	90.0%	73.8%	**94.6%**
leopard	160068	1103	59.9%	86.3%	**96.3%**
glacier	176039	1107	99.4%	89.4%	**99.6%**

Table 2: Quantitative evaluation on images from the Berkley Image Dataset. Accuracy rates for the evaluated approaches are given in percent of correct pixels compared to a hand-labelled ground truth.

Constraint Free Segmentation We showed that our approach is well-suited to learn difficult foreground-background models interactively. We now demonstrate the ability of these models to segment lots of images automatically: After successful interactive segmentation of an image we can apply the trained RF to similar images. Figure 8 shows an example: Given an industrial quality inspection task, where the appearance of parts and expected defects is known. The interactive approaches mentioned above produce too weak global hypotheses to make automatic segmentation possible. Training an accurate classifier using non-interactive approaches is tedious work to do. With our interactive framework, we can easily train a classifier once on representative data and use it to segment any upcoming image autonomously without any further user interaction.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 8: Our approach in an industrial quality inspection task: The classifier is interactively trained with a few images representing the expected appearance of good parts and defects (a-d). This classifier can be applied to future images without any further user interaction needed (e-l). Other interactive approaches fail here due to their weak global hypotheses.

5 Conclusion

In this paper, we have introduced an interactive segmentation framework using GPU-based Random Forests to learn arbitrary texture descriptors. We showed that our framework produces stronger hypotheses than recent approaches that model color distributions solely. A fast primal-dual algorithm was presented to minimize a convex energy functional that utilizes a point-wise dataterm allowing for the incorporation of continuous segmentation hypotheses. We showed superior segmentation results on natural, industrial, artificial as well as medical data. We implemented the feature extraction, RF training and evaluation as well as the variational minimization algorithm on the GPU, resulting in a framework fast enough for convenient user interaction.

Random Forests are inherently multiclass. Therefore, future work will focus on extending the framework from binary to multilabel segmentation. Our current segmentation scheme is only able to handle binary problems, however, Pock *et al.* [18] recently worked on TV-based multilabel minimization, which could be useful for our problem. Furthermore, we will work on additional features and extend the framework to segment volumes or videos in spatial-temporal representation [26].

Acknowledgements This work was supported by the Austria Research Promotion Agency within the project VMGPU (no. 813396).

References

[1] Yali Amit, Geman August, and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1996.

[2] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):106–118, 2006.

[3] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Proceedings 13th International Conference on Computer Vision*, 2007.

[4] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *Proceedings 13th International Conference on Computer Vision*, 2007.

[5] Yuri Y. Boykov and Marie P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings 8th International Conference on Computer Vision*, 2001.

[6] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[7] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[8] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Global minimizers of the active contour/snake model. In *Free Boundary Problems (FBP): Theory and Applications*, 2005.

[9] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[10] Antonio Criminisi, Toby Sharp, and Andrew Blake. GeoS: Geodesic image segmentation. In *Proceedings 10th European Conference on Computer Vision*, 2008.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[12] C. Fowlkes, D. Martin, and J. Malik. Local figure-ground cues are valid for natural images. *Journal of Vision*, 7(8):2, 1–9, 2007.

[13] G. Friedland, K. Jantz, and R. Rojas. Siox: simple interactive object extraction in still images. In *7th IEEE International Symposium on Multimedia*, 2005.

[14] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[15] S. Han, W. Tao, D.-S. Wang, X.-C. Tai, and X. Wu. Image segmentation based on Grabcut framework integrating multi-scale nonlinear structure tensor. UCLA CAM Report 09-14, 2009.

[16] S. Leung and S. Osher. Fast global minimization of the active contour model with TV-inpainting and two-phase denoising. In *3rd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2005.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings 8th International Conference on Computer Vision*, July 2001.

[18] Thomas Pock, Antonin Chambolle, Daniel Cremers, and Horst Bischof. A convex relaxation approach for computing minimal partitions. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[19] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3): 309–314, 2004.

[20] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[21] Amir Saffari, Christian Leistner, and Horst Bischof. Online random forests. Technical report, Institute for Computer Graphics and Vision, Graz University of Technology, 2009.

[22] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object class segmentation using random forests. In *Proceedings 19th British Machine Vision Conference*, 2008.

[23] Toby Sharp. Implementing decision trees and forests on a GPU. In *Proceedings 10th European Conference on Computer Vision*, 2008.

[24] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proceedings 13th International Conference on Computer Vision*, 2007.

[25] Markus Unger, Thomas Pock, Werner Trobin, Daniel Cremers, and Horst Bischof. TVSeg - interactive total variation based image segmentation. In *Proceedings 19th British Machine Vision Conference*, 2008.

[26] Markus Unger, Thomas Mauthner, Thomas Pock, and Horst Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted TV. In *7th Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009.

[27] V. Vineet and P.J. Narayanan. Cuda cuts: Fast graph cuts on the gpu. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008.

[28] Shiming Xiang, Feiping Nie, and Changshui Zhang. Texture image segmentation: An interactive framework based on adaptive features and transductive learning. In *Proceedings of the Asian Conference on Computer Vision*, 2006.

[29] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report 08-34, 2008.

[30] Mingqiang Zhu, Stephen J. Wright, and Tony F. Chan. Duality-based algorithms for total variation image restoration. UCLA CAM Report 08-33, 2008.