

Patch-Cuts: A Graph-Based Image Segmentation Method Using Patch Features and Spatial Relations

Gerd Brunner^{1,2}
gbrunner@bcm.edu

Deepak R. Chittajallu¹
drchittajallu@uh.edu

Uday Kurkure¹
ukurkure@uh.edu

Ioannis A. Kakadiaris¹
ioannisk@uh.edu

¹ Computational Biomedicine Lab, Dept. of Computer Science, University of Houston, Houston, TX, USA

² Section of Atherosclerosis and Vascular Medicine, Department of Medicine, Baylor College of Medicine, Houston, TX, USA

Abstract

In this paper, we present a graph-based image segmentation method (patch-cuts) that incorporates features and spatial relations obtained from image patches. In the first step, patch-cuts extracts a set of patches that can assume arbitrary shape and size. Patches are determined by a combination of intensity quantization and morphological operations and render the proposed method robust against noise. Upon patch extraction, a set of intensity, texture and shape features are computed for each patch. These features are integrated and minimized simultaneously in a tunable energy function. Patch-cuts explores the benefit of information theory-based measures such as the Kullback-Leibler and the Jensen-Shannon divergence in its energy terms. In our experiments, we applied patch-cuts to general images as well as to non-contrast Computed Tomography heart scans.

1 Introduction

Image segmentation is a long-standing problem in computer vision where energy minimization methods have been intensively researched. In particular, graph-based partitioning methods have proven successful in various applications [5, 6, 10]. Among the most important advantages of graph-based segmentation methods are their ability to effectively handle topology changes, and the fact that their optimization typically results in a global minimum [2, 3]. More recently, Vu *et al.* [11] proposed an energy term based on a shape distance known from level set formulations. The shape prior approach was generalized to handle multiple, possibly overlapping objects. Slabaugh *et al.* [8] incorporated an elliptical shape prior into the graph-cut framework. Xu *et al.* [12] introduced a graph-cuts based segmentation method motivated by the active contours approach to object segmentation. The method is iterative, starts with an initial contour, and uses the graph-cut segmentation framework proposed by Boykov *et al.* [4] to find the globally optimal contour in a belt-shaped neighborhood around the contour. Kim *et al.* [1] were first to use the mutual information in

an energy minimization algorithm for determining visual correspondences. Their proposed method uses the maximization of mutual information, a powerful technique for computing visual correspondences of two images that does not require an *a priori* model of the relationship between scene intensities in different views. Recently, graph-cuts have also been integrated and combined in more elaborated learning-based schemes. Learning, boosting and statistical learning models in combination with graph-cuts are not the subject of this paper [4, 19], though patch-cuts might be integrated with some of these methods.

Without doubt, graph-based methods have advanced our understanding of image segmentation. However, current graph-cuts methods do exhibit certain limitations. Both the basic graph-cuts formulations, as well as many extensions, base their n-link energy terms solely on absolute pixel intensities or intensity gradients. In the case of shape priors, the energy term usually contains additive shape energies which require templates [20] or assume circular or ellipsoidal regions [11, 18]. These priors greatly improve performance in the case of object classes with similar shapes or in the presence of templates or statistical shape models. However, the smoothness energy term in most graph-cuts methods is based on pixel intensities only. It is known that pixel intensities can be locally erroneous due to noise and other image acquisition problems. Thus, in these cases, noise can adversely affect the performance of graph-based segmentation methods. In addition, typical graph-cuts energy terms contain little flexibility with respect to intensity, shape or texture information.

In this paper, we propose *patch-cuts*, a graph-based segmentation method that: i) incorporates image patches in graph cuts; ii) introduces a tunable energy function that consists of intensity, shape, texture, and spatial terms; and iii) integrates a robust class of information theory energy terms. The advantages of patch-cuts are: i) the use of patch descriptors makes the proposed method robust against noise; ii) the tunable energy function is versatile; and iii) using the Kullback-Leibler [24] and Jensen-Shannon [5] divergence could possibly result in better segmentations. Patch-cuts combines the strength of patches, information theory measures and a tunable energy equation in a single framework. Texture, intensity, shape, and spatial information are simultaneously minimized resulting in a clear, adjustable and robust problem formulation.

2 Patch-Cuts

Patch-cuts¹ incorporates aggregated pixel information, in the form of patches and their properties, into the formalism of graph-based segmentation. The individual steps of patch-cuts are outlined in Algo. *Patch-Cuts*.

Algorithm *Patch-Cuts*

1. Extract Patches
2. Compute Intensity, Texture and Shape Descriptors
3. Construct a Graph Based on Patches
4. Minimize the Tunable Energy Function

2.1 Patch Extraction

We define a patch as a contiguous image region of at least three pixels, even though patches are typically much larger. The underlying motivation of patches is that *homogeneous* regions provide a very natural

¹We chose the term *patch* since in the literature it was used for both rectilinear areas and arbitrarily shaped areas [24][11].

approach to segmentation of objects. We motivate our approach from a physiological point of view, as human perception of objects and scenes, is to a large extent, based on particular spatial configurations and changes in intensity [13]. The extraction of patches is performed in several steps. First, we smooth the input image. This is accomplished by convolving the raw 2-D image $\mathbf{I}(x, y) \in \mathbb{R}^2$ with a Gaussian. Next we perform a quantization of the intensity space of the smoothed input image. Let $\mathbf{S}(x, y)$, with $x \in (1, 2, \dots, N)$, $y \in (1, 2, \dots, M)$ and $\mathbf{S}(x, y) \in [0, 255]$ be the gray scale channel of the smoothed input image. Further, let h_I be the distribution of all gray levels of $\mathbf{S}(x, y)$. Then we map the gray values within the range of $[0, 255]$ to the range $[0, \mathcal{N}^g]$, where \mathcal{N}^g is the number of intensity quantization levels. Figure 1 depicts sample images and their corresponding quantized (patch) images.

Once the quantized image $\mathbf{Q}(x, y)$ is obtained, we proceed with the extraction of arbitrarily shaped non-overlapping patches. Specifically, we split $\mathbf{Q}(x, y)$ into K non-overlapping patches $\mathcal{P} = \{p_k \mid k = 1, 2, \dots, K\}$. The number of patches is determined by the number of regions R for each individual gray scale \mathcal{R}^g in the range $[0, \mathcal{Q}^g]$. More specifically, the number of patches \mathcal{P} is $\sum_{g=1}^{\mathcal{Q}^g} \sum_{r=1}^R R_r^g$, where r and g represent the indices over all regions for every gray scale, respectively. The number of patches may vary with image size and the number of quantized gray scales \mathcal{Q}^g .

In order to smooth patch boundaries and to remove single pixels within patches, we employ morphological operations. Patch-cuts performs a morphological closing operation for each patch. The Gaussian smoothing and especially the morphological closing operation (which is a common method for noise removal [14]), makes the patch extraction step robust against the presence of noise. For typical images, as presented in this paper, patch-cuts extracts between 1,000 and 12,000 patches per image. The exact number of patches for every image used in this paper can be found in Section 3. In the next section, we outline the information extracted from every patch p_k .

2.2 Patch-Based Information Extraction

This section describes the procedure to extract intensity, texture and shape information from every patch p_k . The intensity and texture information is extracted from pixel set ψ of the k^{th} patch p_k in the input image space $\mathbf{S}(x, y)$. Specifically, we create an intensity histogram H_k^I for every patch p_k . H^I is used in the patch intensity term $U_I(\mathcal{L}_k, \mathcal{L}_o)$ (Section 2.6). Note that we do not use individual pixel values of the patches, but aggregate descriptors in the form of a histogram. By doing so, we render patch-cuts robust against noise.

In the subsequent step, texture features are computed for each patch p_k . Specifically, we compute for each patch p_k a set of Haralick [9] gray level co-occurrence matrix (GLCM) features. Let $\Omega^j(l, m, \lambda, \phi)$ be the co-occurrence matrix with pixel pairs $\{l, m\}$. Specifically, we derive four GLCM features: 1) correlation (Υ); 2) energy (Γ); 3) contrast (Δ); and 4) homogeneity (Ξ). In our experiments, we set $\lambda = 1$ and $\phi = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. More details on parameter setting are listed in Section 3. The GLCM features are incorporated in a patch-based texture feature vector of the form $H^T = \{\Upsilon, \Gamma, \Delta, \Xi\}$. If λ consists of more than one element, then the individual texture vectors H_k^T are simply concatenated. Note that in Section 2.6 we refer to the k^{th} patch texture vector simply as H_k^T .

In addition to intensity and texture we add basic shape descriptors to the set of patch descriptors. This is accomplished by computing a set of simple shape features for each patch p_k . Let b_x and b_y be the Euclidean lengths of the bounding box containing patch p_k , and A denote the pixel area of patch k . Specifically, we extract nine features for each patch p_k : 1) solidity (area/convex area); 2) eccentricity; 3) area; 4) perimeter; 5) minor and 6) major axes lengths; 7) compactness ($C = \frac{A}{b_x b_y}$); 8) form factor ($F = \frac{4(b_x + b_y)^2}{b_x b_y}$); and 9) roundness ($\frac{4\pi A}{\text{perimeter}^2}$). The nine individual features are concatenated into one patch shape vector H_k^S for every patch p_k . We use simple shape descriptors as they can be efficiently computed in real time. In the next section, we outline the graph construction based on patches including the details of the energy function.

2.3 Patch-Based Graph Representation

This section describes the construction of a graph based on patches using the methodology of graph-cuts. We build a weighted undirected graph $\mathcal{G} = (N, \mathcal{E})$. The graph is composed of a set of N nodes connected through a set of weighted edges \mathcal{E} . The weights \mathcal{W} are non-negative and represent the cost $\mathcal{W} : \mathcal{E} \rightarrow \mathcal{R}^+$. The main difference from the usual graph construction is that we connect patches instead of pixels. As we are working with binary segmentation problems, we add two terminal nodes (t-nodes), and one source and one sink node (s, t , respectively) to complete the graph topology. In this paper, we discuss binary segmentation problems only, however, there can be several foreground objects in one image provided they belong to the same class (last image in Fig. 1). The t-links determine to which label a patch can be assigned, and the n-links connect pairs of neighbor patches. The goal is to assign a label $\mathcal{L}_k \in \{0, 1\}$ to every patch in a given image. Specifically, $U_L(\mathcal{L}_k)$ determines the energy to assign label \mathcal{L}_k to the k^{th} patch. The energy function to be minimized consists of $U_L(\mathcal{L}_k)$ and the sum of the four subsequent patch-based terms $U_P(\mathcal{L}_k, \mathcal{L}_o)$ (Eq. 3).

$$E(\mathcal{L}) = \sum_{k=1}^K U_L(\mathcal{L}_k) + \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_P(\mathcal{L}_k, \mathcal{L}_o). \quad (1)$$

The second term $U_P(\mathcal{L}_k, \mathcal{L}_o)$ computes the cost of assigning the labels $\mathcal{L}_k, \mathcal{L}_o$ to the neighboring patches k and o . Further, \mathcal{N}_k is the neighborhood system for patch k (i.e., it refers to the set of patches adjacent to patch k). The energy function of patch-cuts consists of four additive terms (Eq. 3).

The actual segmentation is obtained by partitioning the graph [8, 10]. Specifically, an s/t cut is obtained by completely separating all terminal nodes from each other: $D(C) = \{(s/t) \in \mathcal{E} \mid s \in U, t \notin U\}$. Here $s \in S$ is the node subset containing all sources and $t \in T$ is the node subset containing all sinks, with $S, T \in N$. Hence, the cut consists of a subset of edges $C \subset \mathcal{E}$. The total cost of the segmentation is obtained by a simple summation over all weights \mathcal{W} as $|C| = \sum_{c \in C} w_c$, with w_c representing the c^{th} edge weight of C and $w_c \in \mathcal{W}$. Next, we discuss information theory measures that are incorporated into patch-cuts.

2.4 Energy Terms Based on Information Theory Divergence

Mutual information is a well-known information theory measure that is used to determine the dependence between random variables. The mutual information fulfills two important properties. First, it is always non-negative, and second, it is zero if and only if the given variables are statistically independent. These two properties led to numerous applications where the mutual information was used as a similarity measure between variables and distributions. As outlined in the introduction, Kim *et al.* [10] were the first to use the mutual information in a minimization algorithm. Information measures such as mutual information do fulfill the submodularity criterion which is of importance for graph-cuts [13]. However, in general, mutual information is very difficult to estimate. In patch-cuts we incorporate the Kullback-Leibler and Jensen-Shannon divergence [9].

The Kullback-Leibler divergence is a natural distance measure between a *true* probability distribution and an arbitrary probability distribution. Note that the Kullback-Leibler measure is not symmetric and therefore is only a divergence and not a metric. Specifically, the Kullback-Leibler divergence is defined as

$$\mathcal{K}(U, V) = \sum_x U(x) \log_2 \frac{U(x)}{V(x)} = \sum_x U(x) \log_2 U(x) - \sum_x U(x) \log_2 V(x).$$

Alternatively, the Jensen-Shannon divergence is written as

$$\mathcal{J}(U, V) = Z(\pi_1 U(x) + \pi_2 V(x)) - (\pi_1 Z(U(x)) + \pi_2 Z(V(x))),$$

with π_1 and π_2 as weights of the distributions, satisfying $\pi_1 + \pi_2 = 1$, and

$$Z(U) = -\sum_x U(x) \log_2 U(x); \quad Z(V) = -\sum_x V(x) \log_2 V(x)$$

denote the Shannon entropies of the probability distributions of U and V . In contrast to the Kullback-Leibler divergence, the Jensen-Shannon divergence is symmetric, always well defined, and bounded. In patch-cuts we are working with distributions extracted from patches of varying size. Hence, in patch-cuts, the probability distributions U and V of Eqns. 2 and 2 are replaced with the respective patch descriptors outlined in Section 2.2. The Kullback-Leibler and Jensen-Shannon divergence are used in the energy function (Eq. 3).

2.5 Probability Distributions of the Object and Background Regions

The intensity distribution of the object and background regions are obtained by seed points placed interactively by a user. The intensity distribution of patch k is given by H_k^I introduced in Section 2.2. Let \mathcal{F} and \mathcal{B} be patch subsets marked by the user for the respective foreground and background regions. Then we estimate the intensity probability distribution of the foreground region \mathcal{F} and of the background region \mathcal{B} as $p(H^I|\mathcal{F})$ and $p(H^I|\mathcal{B})$, respectively. Subsequently, we use the negative log-likelihood of the intensity probability distribution as data term $U_L(\mathcal{L}_k)$

$$U_L(\mathcal{F}) = -\ln p(H_k^I|\mathcal{F}); U_L(\mathcal{B}) = -\ln p(H_k^I|\mathcal{B}). \quad (2)$$

Next, we present the details of the four energy terms $U_P(\mathcal{L}_k, \mathcal{L}_o)$ (Eq. 3).

2.6 Tunable Energy Function

Patch-cuts takes advantage of extracting region properties which single pixels cannot provide, such as intensity, texture, shape and spatial features. Incorporating region properties in the energy function positions patch-cuts as a highly adjustable graph-cut segmentation algorithm. The four energy terms incur costs based on independent characteristics of a region. The individual terms can be tuned by multiplicative weight factors. Hence, the energy terms can be adjusted to specific scenarios, such as when texture is of more importance. In this case, we would modify the energy function such that all weight factors are equal to zero except for the texture energy $U_T(\mathcal{L}_k, \mathcal{L}_o)$. In detail, we propose a tunable patch-based energy function $U_P(\mathcal{L}_k, \mathcal{L}_o)$ which incorporates intensity, texture, shape, and spatial distance information. We incorporate descriptors (Section 2.2) obtained from all patches in the energy function. The complete energy function $E(\mathbf{L})$ is defined as

$$E(\mathbf{L}) = \sum_{k=1}^K U_L(\mathcal{L}_k) + \left(w_I \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_I(\mathcal{L}_k, \mathcal{L}_o) + w_T \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_T(\mathcal{L}_k, \mathcal{L}_o) \right. \\ \left. + w_S \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_S(\mathcal{L}_k, \mathcal{L}_o) \right) * \left(1 - w_D \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_D(\mathcal{L}_k, \mathcal{L}_o) \right), \quad (3)$$

where $U_L(\mathcal{L}_k)$ is the data term (Section 2.5), $U_I(\mathcal{L}_k, \mathcal{L}_o)$ captures the intensity information, $U_S(\mathcal{L}_k, \mathcal{L}_o)$ denotes the shape energy, $U_T(\mathcal{L}_k, \mathcal{L}_o)$ represents the texture energy and $U_D(\mathcal{L}_k, \mathcal{L}_o)$ refers to the spatial energy. w_I , w_T , w_S and w_D denote the weights for the intensity, texture, shape and distance energy terms, respectively. Each weight is bounded by $[0, 1]$. In the following, we spell out all individual terms of Eq. 3. The intensity energy term $U_I(\mathcal{L}_k, \mathcal{L}_o)$ is given by

$$U_I(\mathcal{L}_k, \mathcal{L}_o) = \exp\left(-\frac{\mathcal{K}(H_k^I, H_o^I)^2}{2}\right), \quad (4)$$

where $\mathcal{K}(H_k^I, H_o^I)$ is the Kullback-Leibler divergence (Eq. 2) between the intensity histograms H^I of patches k and o , respectively. Next, the texture energy term $U_T(\mathcal{L}_k, \mathcal{L}_o)$ is introduced as

$$U_T(\mathcal{L}_k, \mathcal{L}_o) = \exp\left(-\frac{1}{1 + \mathcal{J}(H_k^T, H_o^T)}\right), \quad (5)$$

with $\mathcal{J}(H_k^T, H_o^T)$ denoting the Jensen-Shannon divergence (Eq. 2) between the texture descriptor H^T of patches k and o , respectively.

The shape energy term $U_S(\mathcal{L}_k, \mathcal{L}_o)$ is of the form

$$U_S(\mathcal{L}_k, \mathcal{L}_o) = \exp\left(-\frac{\mathcal{J}(\mathcal{H}_k^S, \mathcal{H}_o^S)^2}{2}\right), \quad (6)$$

where $\mathcal{J}(\mathcal{H}_k^S, \mathcal{H}_o^S)$ is the Jensen-Shannon divergence between the shape descriptor H^S of patches k and o , respectively.

Arbitrarily sized and shaped patches influence the energy minimization in the graph-cuts framework. Distance priors in a traditional pixel-based graph result in penalty patterns that are identical across pixel cliques, as the distance between pixels does not change. However, in the case of patches, the distance between the geometric centers of patches can differ significantly. This observation holds true within one clique of patches and also across different patch cliques. Patch-cuts utilizes this finding as it incorporates spatial information $U_D(\mathcal{L}_k, \mathcal{L}_o)$ in the energy function in the form of Euclidean distances between the geometric centers of patches. Specifically, the energy function is designed such that the distance term $U_D(\mathcal{L}_k, \mathcal{L}_o)$ is multiplicative with the other energy terms. The $\left(1 - w_D \sum_{k=1}^K \sum_{o \in \mathcal{N}_k} U_D(\mathcal{L}_k, \mathcal{L}_o)\right)$ factor ensures flexibility for the case of $w_D = 0$. Specifically, the distance energy term is defined as

$$U_D(\mathcal{L}_k, \mathcal{L}_o) = \frac{1}{1 - \exp\left(-\frac{|C_{pk} - C_{po}|^2}{2}\right)}, \quad (7)$$

where C_{pk} , C_{po} are the centroid coordinates of the k^{th} and o^{th} patch. The use of intensity-based centroids would not affect the energy term, as the patches are designed to be homogeneous. The energy function is minimized directly with graph-cuts using the max-flow algorithm introduced by Boykov *et al.* [9]. In the next section, we present experimental results that illustrate the flexibility of the presented energy formulation.

3 Experiments

We have conducted all experiments on a 2.67 GHz PC equipped with an Intel Core Duo processor and 4GB RAM. The implementation of patch-cuts was done in MATLAB and C++. The graph optimization problem was solved using the maxflow algorithm [9]. The user interface is based on code from Gupta and Ramnath [9]. The runtime scales with image size and the number of patches. Typical processing times including patch extraction and segmentation are on the order of a few seconds. However, the code is not optimized for speed, so the given runtime could be improved. We compared the performance of patch-cuts using an implementation of state-of-the-art algorithms [8, 10, 11].

In our experiments, we determined the number of intensity quantization levels \mathcal{N}^g for the patch extraction as follows. Let \mathcal{N}_{80}^g be the number of gray scale values (i.e., the number of bins in h_I) necessary to account for 80% of the pixel counts in $\mathbf{S}(x, y)$. Based on our experiments, we found that $\mathcal{N}^g = 0.15 \mathcal{N}_{80}^g$ is a good approximation for the number of quantization levels. This empirical determination of \mathcal{N}^g gives more or fewer quantization levels depending on the image content. For instance, an image with large homogeneous areas results in very few gray scales, whereas an image representing complex scenes is quantized into a larger number of gray scales. We opted for an empirical approach as it delivers the best results in our experiments. Typically, we expect the number of patches to be on the order of hundreds to thousands. Moreover, H^T was computed with $\lambda = 1$ and $\phi = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. The results in Figs. 1 and 4 have been obtained using an intensity histogram H^I of 8 bins as it results in a robust and compact representation. Figure 1 shows results of patch-cuts for different images. The weights w_I , w_T , w_S , and w_D in Eq. 3 were set equal to one. The images of the

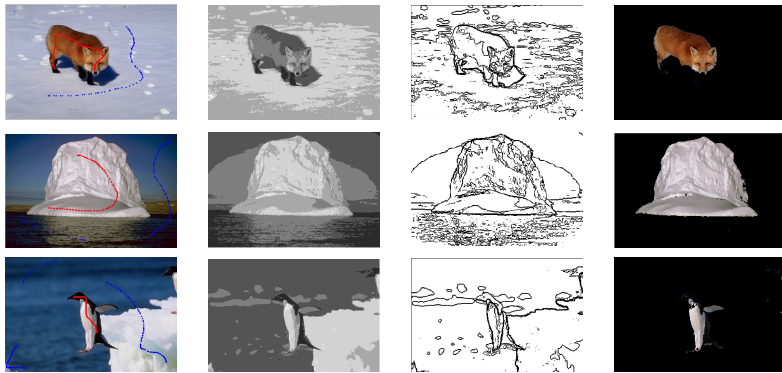


Figure 1: Sample patch-cuts segmentation results. First column: images with user-supplied fore- and background seed points; second column: images illustrating the patches; third column: binary images showing the patch boundaries; and fourth column: the segmentation results using patch-cuts with all weights equal to one (Eq. 3). The number of patches, K is 1,709, 2,929, and 1,220 for the first, second, and third image, respectively. Best if viewed in color.

first column show the seed points in red and blue for the foreground and background, respectively. The images of the second column show the extracted patches as described in Section 2.1. The images in the third column show the boundaries of the patches from the images to the left. Specifically, the third image in the second panel shows a large number of patches in the water area, which is due to intensity fluctuation and reflection effects. Patch-cuts extracted a total of 1,709, 2,929, and 1,220 patches from the images with the fox, iceberg, and penguin, respectively. The images of the fourth column depict the segmentation results using patch-cuts. Note that the image with the fox has also been used by Rother *et al.* [14]. The segmentation result obtained with patch-cuts is of similar quality compared to previous work ([14]), except that the result of patch-cuts does not include the shadow.

The third panel exhibits the performance of patch-cuts in the case of partial occlusion. The penguin to the right was correctly segmented using the distance term in the energy function. Note that we did not place any seeds on the second penguin. The fourth image from the left in Fig. 2 depicts the patch-cuts segmentation for the same penguin image (Fig. 1) by setting $w_D=0$ and $w_I=w_T=w_S=1$. This result illustrates the steerability of the energy function in patch-cuts. Setting the distance term $U_D(\mathcal{L}_k, \mathcal{L}_0)=0$ in Eqns. 3 and 7 removes the second penguin from the segmentation result. Note that, due to space considerations, we moved the segmentation result using the penguin from Fig. 1 to Fig. 2.

The first three images in Fig. 2 depict segmentation results using an implementation of GrabCut [14] and Lazy Snapping [16] by Gupta and Ramnath [8]. GrabCut and Lazy Snapping are parts of the Interactive Segmentation Tool-Box [8] which also features the iterative combination of GrabCut plus Lazy Snapping².

Lazy Snapping uses a watershed algorithm in its implementation which makes it very suitable to be compared with patch-cuts. We used the toolbox as is, without any modification. Figure 3 depicts more results using the Interactive Segmentation Tool-Box, using the same input image as in Fig. 1. This time we employed the interactive combination of GrabCut (dragging an initial bounding box) and Lazy Snapping for refining. Seed points are shown in red and blue for foreground and background, respectively. The images depict seed points for the first and second refinement and their respective segmentations. In this case, patch-cuts clearly performs better, as shown in Fig. 1. In addition, we

²The implementation of Lazy Snapping threw exceptions if too few seed points were provided. Therefore, we had to provide substantially more seed points compared to patch-cuts (for the iceberg image (Figs. 1 and 2) the Lazy Snapping implementation required at least 50% more seeds to execute).

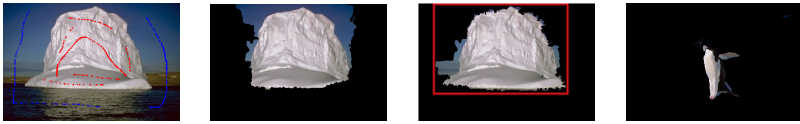


Figure 2: Comparison of patch-cuts with implementations of GrabCut [14] and Lazy Snapping [16] by Gupta and Ramnath [8] using an identical image from Fig. 1. Note that the implementation of Lazy Snapping did not execute with the same seed points used by patch-cuts in Fig. 1 – the implementation requires more seeds. The left-most image shows the new user-supplied seed points, and the second image displays the segmentation obtained with the implementation of Lazy Snapping. The third image shows the segmentation result using the implementation of GrabCut, where the red bounding-box represents the initial user selected region of interest. The last image shows the segmentation of the penguin (same seed points as in Fig. 1) using $w_I=w_T=w_S=1e$ and $w_D=0$. Best if viewed in color.



Figure 3: Segmentation results for the comparison of patch-cuts with implementations of GrabCut [14] and Lazy Snapping [16] by Gupta and Ramnath [8]. The results show two iterations using GrabCut and Lazy Snapping. Note that the input image is identical to the image in Fig. 1. The images from left to right show: 1) initial GrabCut segmentation with the user selected bounding-box; 2) added seed points for the first iteration using Lazy Snapping; 3) the respective segmentation result; 4) added seed points for a second refinement using Lazy Snapping, and 5) the respective segmentation result. Best if viewed in color.

have investigated the robustness of patch-cuts with respect to noise. Figure 4 shows different noise levels of 20%, 30% and 50% for the penguin image. Specifically, noise was introduced in the image by assigning a randomized color to a certain percentage (20%, 30%, 50%) of all image pixels. The segmentation results in the fourth column show the robustness of patch-cuts.

In order to demonstrate the versatility of patch-cuts we conducted additional experiments using medical images. Specifically, we worked on the problem of heart segmentation in non-contrast Computed Tomography (CT) data. The segmentation of the heart is of importance for the quantification of cardiac fat or of calcified arterial lesions and other cardiovascular disease related markers. The cardiac scans (DICOM format) presented in this paper have been obtained by EBCT (electron beam CT) imaging with a slice thickness of 3 mm and an axial pixel spacing of 0.508 - 0.586 mm. We extract the same descriptors as in the case of general images. All medical image segmentation results (Figs. 5 and 6) have been obtained using 12 bins for the intensity histogram H^I .

Figure 5 shows the seed points, the patch image, the patch boundaries, and segmentation results with different energy functions. Patch-cuts extracted 11,274 patches. Specifically, the first segmentation (fourth image from the left) was obtained with the weights w_I , w_T , w_S equal to one and w_D equal to zero in Eq. 3. The segmentation to the right was obtained with w_S equal zero and w_I , w_T , and w_D equal one. The shape descriptors of the patches do significantly affect the segmentation. The third segmentation result in the lower panel was computed with w_S and w_D equal zero, and w_I and w_T equal to one. Figure 6 shows another heart segmentation of a different patient. Here, all weights in Eq. 3 were set to one. Patch-cuts decomposed the scan into 11,333 patches. The fourth image shows the segmentation of the heart.

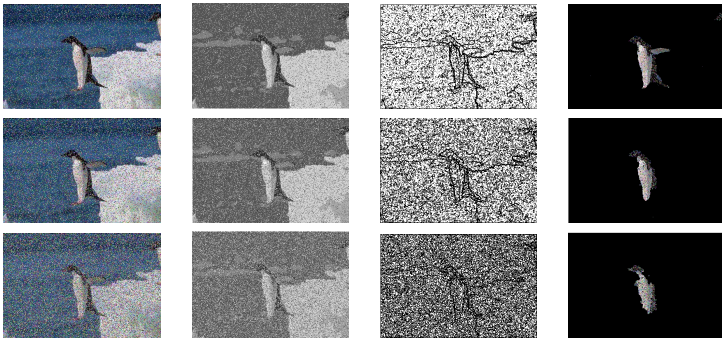


Figure 4: Robustness analysis of patch-cuts using the third image from Fig. 1 with the exact same seed points. The noise level is 20%, 30% and 50% (percentage of randomized image pixels) compared to the input image (Fig. 1). Each panel shows from the left: 1) the noisy input image (seeds are identical as in Fig. 1); 2) images illustrating the patches; 3) binary images showing the patch boundaries; and 4) the segmentation results using patch-cuts with all weights equal to one (Eq. 3). Best if viewed in color.

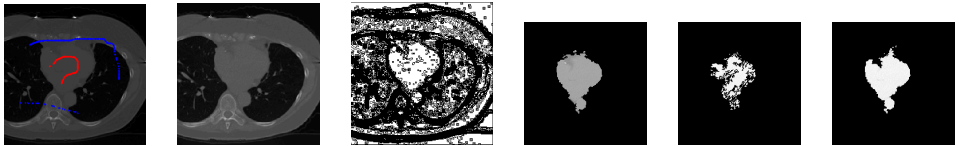


Figure 5: Patch-cuts segmentation results for an axial non-contrast cardiac CT scan, near the aortic root. From the left: 1) seed points; 2) the patch image; 3) the patch boundaries; 4) segmentation with $w_I=w_T=w_S=w_D=1$ (Eq. 3); 5) result with $w_S=0$ and $w_I=w_T=w_D=1$; and 6) result with $w_S=w_D=0$ and $w_I=w_T=1$. $K=11,274$. Best if viewed in color.

4 Conclusion

We present patch-cuts, a novel graph-based image segmentation method. Patch-cuts combines several concepts. First, the use of patches provides robustness against noise. Second, the use of patches allows for the incorporation of region-derived feature histograms in the energy function. To that end, patch-cuts introduces a tunable energy function that comprises intensity, shape, texture, and spatial terms. The novel energy function positions patch-cuts as a flexible graph-cuts framework. Third, patch-cuts introduces the Kullback-Leibler and the Jensen-Shannon divergences as a class of information theory measures in the energy function. The experimental results are encouraging and show that patch-cuts is robust with respect to noise and can be successfully applied to general images as well as to cardiac CT scans. The initial results of patch-cuts reveal that, depending on object appearance, certain weight settings in the tunable energy function perform preferably. A detailed analysis of exactly which weight settings perform *best* for specific object classes will be the subject of a forthcoming paper.

Acknowledgments:

This work was supported in part by the Biomedical Discovery Training Program of the W.M. Keck Center for Interdisciplinary Bioscience Training of the Gulf Coast Consortia (NIH Grant No. 1 T90 DA022885 and 1 R90 DA023418), and in part by NSF Grants IIS-0431144, CNS-0521527, and by NIH grant T32HL07812.

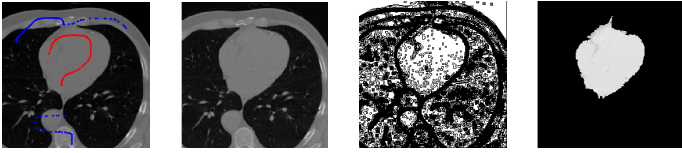


Figure 6: Patch-cuts segmentation results for an axial non-contrast cardiac CT scan, near the aortic root, for a second patient. From the left: seed points, the patch image, the patch boundaries, and the segmentation result with all weights in Eq. 3 equal to one. $K=11,333$. Best if viewed in color.

References

- [1] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? A unified approach to segment extraction. In *Proc. 10th European Conference on Computer Vision*, pages 30–44, Marseille, France, Oct. 12-18 2008.
- [2] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep. 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. 8th IEEE International Conference on Computer Vision*, volume 1, pages 105–112, Vancouver, Canada, Jul. 7-14 2001.
- [4] J Corso, A. Yuille, and Z. Tu. Graph-shifts: Natural image labeling by dynamic hierarchical computing. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.
- [5] T.M. Cover and J.A. Thomas. *Elements of information theory*. John Wiley & Sons, 1991.
- [6] G. Funka-Lea, Y. Boykov, C. Florin, M.P.A. Jolly, M.P. Jolly, R. Moreau-Gobard, R. Ramaraj, and D. Rinck. Automatic heart isolation for CT coronary visualization using graph-cuts. In *Proc. 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 614–617, Arlington, VA, Apr. 6-9 2006.
- [7] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.
- [8] M. Gupta and K. Ramnath. Interactive segmentation tool-box, <http://www.cs.cmu.edu/mohitg/segmentation.htm>, 2008.
- [9] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.
- [10] A. Tannenbaum, J. Malcolm, Y. Rathi. Graph cut segmentation with nonlinear shape priors. In *Proc. IEEE International Conference on Image Processing*, pages 365–368, San Antonio, TX, Sep. 16-19 2007.
- [11] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proc. 9th International Conference on Computer Vision*, page 1033, Nice, France, Oct. 13-16 2003.
- [12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

- [13] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proc. 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 324–333, Edinburgh, Scotland, Jul. 26-29 2005.
- [14] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.
- [15] J. Li and C. Kuo. Automatic target shape recognition via deformable wavelet templates. In *Proc. SPIE: Wavelet Application III*, volume 2756, pages 2–13, 1996.
- [16] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, Aug. 2004.
- [17] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, Aug. 2004.
- [18] G.G. Slabaugh and G. Unal. Graph cuts segmentation using an elliptical shape prior. In *Proc. International Conference on Image Processing*, pages 1222–1225, Genoa, Italy, Sep. 11-14 2005.
- [19] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *Proc. 10th European Conference on Computer Vision*, pages 582–595, Marseille, France, Oct. 12-18 2008.
- [20] Nhat Vu and B.S. Manjunath. Shape prior segmentation of multiple objects with graph cuts. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 24-26 2008.
- [21] Y. Wu, S. Zhu, and X. Liu. Equivalence of Julesz and Gibbs texture ensembles. In *Proc. International Conference on Computer Vision*, volume 2, pages 1025–1032, Corfu, Greece, Sep. 20-25 1999.
- [22] Ning Xu, Narendra Ahuja, and Ravi Bansal. Object segmentation using graph cuts based active contours. *Computer Vision and Image Understanding*, 107(3):210–224, 2007.
- [23] J. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, 2001.