K-convexity Shape Priors for Segmentation

Hossam Isack[1(✉)], Lena Gorelick[1], Karin Ng[2], Olga Veksler[1], and Yuri Boykov[1]

[1] University of Waterloo, Waterloo, Canada
hisack@uwaterloo.ca
[2] University of Western Ontario, London, Canada
kng263@uwo.ca

Abstract. This work extends popular star-convexity and other more general forms of convexity priors. We represent an object as a union of "convex" overlappable subsets. Since an arbitrary shape can always be divided into convex parts, our regularization model restricts the number of such parts. Previous k-part shape priors are limited to disjoint parts. For example, one approach segments an object via optimizing its k coverage by disjoint convex parts, which we show is highly sensitive to local minima. In contrast, our shape model allows the convex parts to overlap, which both relaxes and simplifies the coverage problem, e.g. fewer parts are needed to represent any object. As shown in the paper, for many forms of convexity our regularization model is significantly more descriptive for any given k. Our shape prior is useful in practice, e.g. biomedical applications, and its optimization is robust to local minima.

1 Introduction

Regularization is common in computer vision problems/applications such as photo or video editing, biomedical image analysis, weakly-supervised training of semantic CNN segmentation, etc. Typical regularization techniques often correspond to imposing various priors, e.g. smoothness [1–3], shape [4–8], hierarchical [9–11], volumetric [12], or other priors. This work proposes a particularly simple, yet sufficiently discriminant and efficient model of a general shape prior based on the geometric concept of convexity. While our main ideas could be expressed in either discrete or continuous settings, for simplicity we focus on the former and propose a combinatorial optimization technique based on graph-cuts [2,6].

Convexity is a powerful regularization concept for segmentation [4,8,13,14]. However, in practical applications it is rare that objects are strictly convex. Our premise is that an object of interest can be represented as a union of a small number of convex parts. We propose a form of multi-convexity shape prior, namely k-convexity, to regularize the problem of segmenting such objects. Our definition of k-convexity is a generalization of k-stars in computational geometry literature [15], but it differs from how k-convexity is used in [16]. Our general k-convexity approach can be based on different forms of convexity, e.g. star [4], geodesic-star [13], hedgehog [14], or regular convexity [8]. In segmentation, the concept of k-convexity was first discussed by [13] in the context of stars [4], but citing **NP**-hardness they focused on an easier-to-optimize multi-star prior with a predefined region for each star, see k-regions versus k-convexity in Table 1.

Predefined regions for object parts [13] could be avoided by segmenting these parts as independent objects with appropriate convexity priors a la [14]. This is a viable alternative to k-convexity, see k-disjoint in Table 1, but we found that representing an object via disjoint convex parts leads to local minima. Moreover, compared to overlapping convex parts in k-convexity, a larger number of disjoint convex parts may be needed to represent the same shape, see Table 1.

Similar to [4,6], our shape prior methodology is presented within graph-cut optimization framework, but other optimization techniques are possible. Besides multi-part object modeling, our approach easily adapts to segmenting independent overlapping objects, e.g. cells, addressed earlier by other priors in active contours [7,17], level-sets [18] and graph cuts [19].

(a) regular   (b) star [4]   (c) geodesic-star [13]   (d) hedgehog [14]
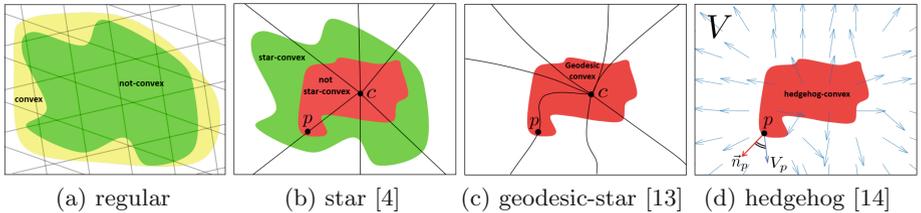
Fig. 1. *Different types of convexity*: (a) regular convexity; yellow shape is convex and green is not. (b) star-convexity; green shape is star-convex w.r.t. center c while red is not. (c) shows geodesic paths to c for which the red shape is geodesic-star-convex. (d) shows vector field V for which the red shape is hedgehog-convex. Lines, rays and vector fields in (a–d) are used to define convexity constraints, see text for details.

Types of Convexity: there is more than one way to define convexity. A shape S is considered to be convex in the regular sense if it forms a convex set, i.e. if $p, q \in S$ then line pq also lies in S. In practice regular convexity is usually approximated by enforcing convexity constraints only along a predefined number of orientations [8] as shown in Fig. 1(a). Furthermore, even when segmenting a single convex object the resulting function is non-submodular [8], i.e. **NP**-Hard. Alternatively easier-to-optimize types of convexity are used in practice, e.g. star-convexity [4].

A shape S is considered star-convex w.r.t. a center c if for any pixel $p \in S$ the line cp lies in S, see Fig. 1(b). Star-convexity was first used as a shape prior in [4]. Later on [13] proposed geodesic-star-convexity, which imposes the same constraints as star-convexity but along a geodesic path between c and p, see Fig. 1(c). The paths are computed using image color information and the distance between c and p. Both [4,13] encode their shape priors as local pairwise pixels constraints which requires ray or path tracing.

Table 1. *Different types of multi-convexity based on* (1): *the additional constraints corresponding to each type of multi-convexity are shown in the second row. The last two rows show examples for* k = 2. *Shape* S *is shown in solid red, while internal boundaries between parts* $\{S_i\}$ *are shown in dotted red. The additional constraints limit* k-*regions and* k-*disjoint shape representational power, e.g. they require more than two parts to describe the top example of* k-*convexity.*

	k-**regions**	k-**disjoint**	k-**convexity** (ours)
additional constraints	$R_i \cap R_j = \phi$ $\forall i \neq j \in [1, \dots k]$ $S_i \in R_i \ \ \forall i \in [1, \dots k]$	$S_i \cap S_j = \phi$ $\forall i \neq j \in [1, \dots k]$	None
example 1			
example 2			

Recently [14] proposed hedgehog-convexity that gives the user more control over the shape space and, unlike [4,13], does not require ray or path tracing. Instead, hedgehog-convexity requires some vector field V to constrain the shape normals $\{\vec{n}_p \,|\, p \in \partial S\}$ to be within a certain tolerance θ, i.e. $\angle \vec{n}_p V_p \leq \theta$, see Fig. 1(d). Hedgehog-convexity is more general than star and geodesic-star convexity [4,13]. For example, it reduces to star-convexity for radial vector field V centered at c and $\theta = \frac{\pi}{2}$. Furthermore, if $\theta = 0$ in the aforementioned case then shape S must be a circle centered at c, as in the examples shown in Fig. 2.

As mentioned earlier, in practice objects of interest are rarely convex but any arbitrary object can always be divided into convex parts. Multi-convexity is a regularization model where an object of interest is assumed to be the union of convex parts. Different forms of multi-convexity where introduced in [13,14] in the context of geodesic-star and hedgehog convexity, respectively.

Multi-Convexity: without lost of generality we will review multi-convexity in the context of regular convexity but our arguments are general and apply to all types of convexity covered earlier. In multi-convexity the final segmentation S is the union of k parts

$$S = \bigcup_{i=1}^{k} S_i \tag{1}$$

where each part S_i is convex. Note that multi-convexity does not guarantee connectivity of S, i.e. S could contain more than one connected component. This could be addressed by adding a connectivity prior but such priors are **NP**-hard [20]. Shape connectivity is beyond the scope of this paper.

Previous forms of multi-convexity enforce additional constraints that either simplify optimization [13] or inherently appear in a different context, e.g. segmentation of independent objects [14]. We observe that such constraints unnecessarily limit the descriptiveness of the multi-convexity shape prior in the context of multi-part object segmentation.

In [13] the image domain is split into k disjoint predefined regions, e.g. Voronoi cells of the star-centers in the context of star-convexity. In addition, each S_i is constrained to be convex and restricted to be within its corresponding region. In the case of star, geodesic-star or hedgehog convexity tying each part to a predefined region results in a submodular energy, i.e. could be solved optimally in polynomial time. We refer to this approach by k-**regions** [13], see Table 1.

Unlike [13], [14] does not tie an object part to a predefined region. However, [14] enforces mutual exclusion between parts. Mutual exclusion was a reasonable assumption in [14] as the authors introduced multi-convexity in the context of multi-object segmentation not multi-part. Nonetheless, [14] could be applied to segmenting multi-part objects but in practice it is very sensitive to local minima. We refer to [14] multi-convexity approach by k-**disjoint**, see Table 1.

Our main contribution is a novel multi-convexity shape prior, k-**convexity**. Unlike [13,14], our approach does not impose any additional constraints on parts besides convexity. Table 1 juxtaposes k-convexity with the previous multi-convexity approaches. Figure 2 demonstrates k-regions and k-disjoint practical drawbacks. Although k-regions could be solved optimally, it is clear that its shape representational power is limited, see Fig. 2(b). While k-disjoint removes the restriction of parts to predefined regions, it is sensitive to initialization and prone to local minima, see Fig. 2(c). Our k-convexity overcomes these drawbacks by relaxing the solution space, i.e. allowing the parts to overlap, see Fig. 2(d).

Our k-convexity prior can also be motivated by shape reconstruction via *medial axis transform* (MAT) [21], which is the union of overlapping skeleton-centered circles with given radii. As discussed earlier, circle can be seen as a particularly tight form of convexity shape prior. Thus, segmentation with our k-convexity shape prior could be seen as a relaxation of MAT shape reconstruction: instead of the union of circles we compute the union of convex parts, we do not assume fixed radii or scales, and we use partial skeletons, e.g. user-scribbles, instead of full skeletons. Note that segmentation with k-convexity shape prior estimates the scale of each part based on image data (e.g. object color model), while MAT reconstruction assumes known circle radii. These differences are illustrated in Fig. 3.

(a) user seeds   (b) k-regions [13]   (c) k-disjoint [14]   (d) k-convexity (ours)
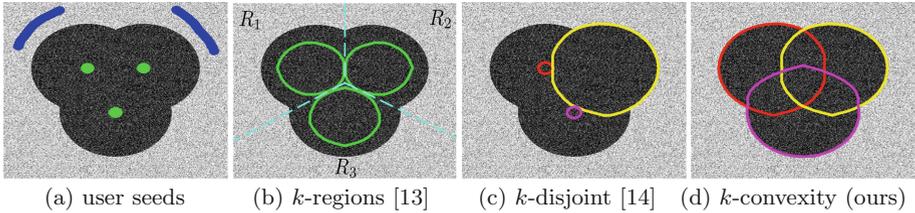
Fig. 2. *Limitations of multi-convexity approaches*: to emphasize them, this synthetic example uses a tight form of convexity shape prior (hedgehogs with $\theta \approx 0$ and radial vector field) enforcing near-circularity for each part. In fact, circularity priors are useful in practice, e.g. cell segmentation [7,17], see Fig. 7. In (b) regions shown in dotted cyan.

(a) MAT shape reconstruction   (b) k-convexity segmentation
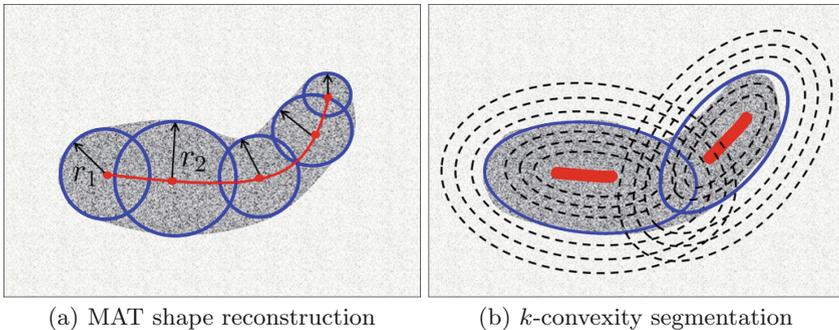
Fig. 3. Illustrates shape reconstruction from skeleton/partial-skeleton (red). The reconstructed shape is the union of the blue parts. (a) reconstruction using a skeleton and radial function. (b) reconstruction using partial skeleton, color cues, and k-convexity with k = 2 and hedgehog-convexity [14] where $\theta \approx 0$. Note, for hedgehog-convexity using the gradient of the scribble's distance map as V and $\theta \approx 0$, limits the set of allowed shapes to the level-sets (black dashed contours) of the distance map.

Our list of contributions are summarized below:

- a novel multi-convexity shape prior for multi-part object or overlapping objects segmentation, namely k-convexity.
- a graph-cuts optimization framework for k-convexity based on [2,6].
- experimental results comparing our k-convexity shape prior to existing multi-convexity approaches [13,14]. We also show k-convexity results for different types of convexity.
- for completeness, a proof that our general formulation of k-convexity is **NP**-Hard.

The paper is organized as follows. In Sect. 2 we formulate k-convexity as multi-labeling energy that permits labels to overlap. We show how to optimize k-convexity in Sect. 3. We compare and validate our approach in the context of biomedical segmentation in Sect. 4, and apply k-convexity to different types of convexity. Finally, Sect. 5 concludes and discusses future work.

2   Energy

Let Ω be the set of all image pixels, and $\mathcal{L} = \{1, \ldots, k\}$ be the set of indices of k overlappable foreground parts, i.e. labels. Also, let $\mathbf{f} = \{f_p \mid \forall p \in \Omega\}$ be a labelling of Ω where f_p is a pixel labelling such that $f_p = \{f_p^i \in \{0, 1\} \mid \forall i \in \mathcal{L}\}$. A pixel p belongs to label i if $f_p^i = 1$ and 0 otherwise. Furthermore, a pixel is considered a background pixel if it is not assigned to any foreground label. For notational simplicity in identifying background pixels we will use an indicator function $\phi(f_p)$

$$\phi(f_p) = \begin{cases} 1 & \text{if } f_p^i = 0 \ \ \forall i \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Our k-convexity multi-part segmentation energy is

$$E(\mathbf{f}) = \overbrace{\sum_{p \in \Omega} D_p(\phi(f_p))}^{\text{data}} + \lambda \overbrace{\sum_{p,q \in \mathcal{N}} V(f_p, f_q)}^{\text{smoothness}} + \overbrace{\sum_{i \in \mathcal{L}} C_i(\mathbf{f}, \theta)}^{\text{convexity}}, \tag{3}$$

where λ is a normalization constant, $\mathcal{N}$ is the pixels' neighborhood system, and the energy terms are described in more details below.

In our *data* term, $D_p(\phi(f_p))$ measures how well a pixel fits the background (Bg) or foreground (Fg) color model depending on in its current labeling. One of the most commonly used data terms is the negative log likelihood

$$D_p(\phi(f_p)) = \begin{cases} -\ln \Pr(I_p \mid Bg) & \phi(f_p) = 1 \\ -\ln \Pr(I_p \mid Fg) & \phi(f_p) = 0 \end{cases} \tag{4}$$

where I_p is the image intensity at pixel p. Since we are segmenting a single object as a multiple convex parts, we assume that the foreground parts have the same color model. Nonetheless, the color models of foreground parts could different if needed, similar to [6].

The *smoothness* term is a regularizer that discourages labeling discontinuities between neighboring pixels $p, q \in \mathcal{N}$. A discontinuity occurs whenever a pixel is assigned to background while its neighbor is assigned to at least one foreground[1]. The simplest form of pairwise discontinuity is,

$$V(f_p, f_q) = w_{pq}[\phi(f_p) \neq \phi(f_q)] \tag{5}$$

where $[\,]$ is the Iverson bracket and w_{pq} is a non-increasing function of I_p and I_q. Note that our energy only penalizes the outside boundary of the union of the foreground parts.

---

[1]   Discontinuities between foreground labels could be penalized, as in cell segmentation.

The *convexity* term is used to forbid (or penalize) solutions with non-convex parts. In (3) $C_i(\mathbf{f}, \theta)$ encodes the convexity prior of label i, while θ is a prior specific parameter(s). It is possible to enforce any of the following convexity priors; star [4], geodesic-star [13], hedgehog [14], or regular [8] convexity. For instance, to enforce hedgehog convexity [14] we define C_i as follows

$$C_i(\mathbf{f}, \theta) = w_\infty \sum_{(p,q) \in \mathcal{E}_i(\theta)} [f_p^i = 1, f_q^i = 0], \tag{6}$$

where w_∞ is a very large constant, θ is the shape tightness parameter, and $\mathcal{E}_i(\theta)$ is, as defined in [14] eq (3), the set of pairwise directed edges used to approximate the hedgehog shape prior for label i given θ. From now on we will adhere to hedgehog-convexity as a show case.

3   Optimization

In Appendix A we prove that (3) is **NP**-hard. To find an approximate solution we follow in the foot steps of [2,6]. They maintain a current labeling $\hat{\mathbf{f}}$ and iteratively try to decrease the energy by switching from $\hat{\mathbf{f}}$ to a near by labeling. Similar to [6], at each iteration in our approach, Algorithm 1, a label $\alpha \in \mathcal{L}$ is randomly chosen and its support region is allowed to simultaneously expand and contract without affecting other foreground parts' support regions. We refer to the aforementioned move by Expansion-Contraction Move (EC-Move), and it is a binary submodular move, see Fig. 4. The algorithm stops when it cannot find an α-EC-Move that decreases the energy anymore.

Algorithm 1: ALPHA-EXPANSION-CONTRACTION

1 $\hat{\mathbf{f}} :=$ initial labeling
2 **repeat**
3   **for each** $\alpha \in \mathcal{L}$
4          $\mathbf{f}^\alpha := \arg\min_{\mathbf{f}} E(\mathbf{f})$ where $\mathbf{f}$ is an α-expansion-contration of $\hat{\mathbf{f}}$
5   **if** $E(\mathbf{f}^\alpha) < E(\hat{\mathbf{f}})$
6              $\hat{\mathbf{f}} := \mathbf{f}^\alpha$
7 **until** converged

3.1   Expansion-Contraction Move (EC-Move)

An α-EC-Move allows α to gain or lose pixel support, which is a binary move. We only apply EC-Moves to foreground labels, because an EC-Move on the background label is a non-submodular multi-label move, since a background pixel has more than one foreground label to choose from when contracting. However, it is possible to only allow the background to expand as in [14].

(a) current labeling   (b) EC-Move binary choices   (c) feasible EC-Move
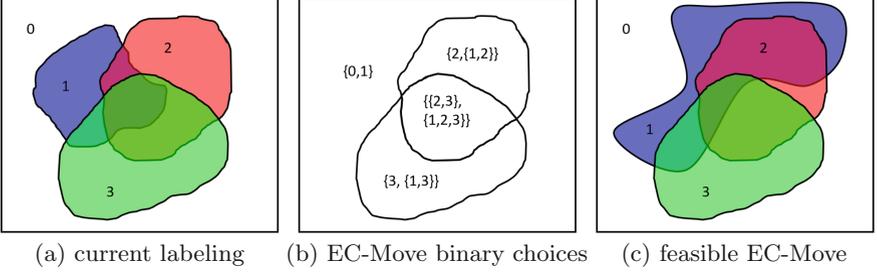
Fig. 4. *EC-Move illustration*: (a) shows a current labeling for a 3-part object. (b) shows the binary choices for each pixel when applying EC-Move for $\alpha = 1$. As you can see in (c) any pixel is allowed to gain or lose α, while other foreground parts remain intact. During an α-EC-Move only the convexity prior of α is taken into account.

Given current labeling $\hat{\mathbf{f}}$ an EC-Move on $\alpha \in \mathcal{L}$ can be formulated as a binary energy as follows:

$$E^\alpha(\mathbf{x}) = \sum_{p \in \Omega} D_p^\alpha(x_p) + \lambda \sum_{p,q \in \mathcal{N}} w_{pq} V^\alpha(x_p, x_q) + C^\alpha(\mathbf{x}, \theta), \tag{7}$$

where $\mathbf{x} = \{x_p \in \{0,1\} \mid \forall p \in \Omega\}$ such that $x_p = 1$ means that p adds α to its current set of labels $\hat{f}_p$ while $x_p = 0$ means removing α, and functions D^α, V^α and C^α are discussed below.

The *data* term in (7) is defined as

$$D_p^\alpha(x_p) = \begin{cases} D_p(0) & x_p = 1 \\ D_p(\phi(\hat{f}_p)) & x_p = 0, \end{cases} \tag{8}$$

the *smoothness* term is defined as

$$V^\alpha(x_p, x_q) = \begin{cases} [\phi(\hat{f}_p) \neq \phi(\hat{f}_q)] & x_p = 0, x_q = 0 \\ [\phi(\hat{f}_p) \neq 0] & x_p = 0, x_q = 1 \\ [\phi(\hat{f}_q) \neq 0] & x_p = 1, x_q = 0 \\ 0 & x_p = 1, x_q = 1, \end{cases} \tag{9}$$

and the *convexity* term is defined as

$$C^\alpha(\mathbf{x}, \theta) = w_\infty \sum_{(p,q) \in \mathcal{E}_\alpha(\theta)} [x_p = 1, x_q = 0]. \tag{10}$$

Submodularity: as shown in [22], any first-order binary function could be exactly optimized if its pairwise terms are submodular. A binary function h of two variables is submodular if $h(0,0) + h(1,1) \leq h(1,0) + h(0,1)$. Our energy (7) is submodular as it could be written as the sum of submodular pairwise binary energies over all possible pairs of p and q. We prove that V^α is a submodular by showing that

$$V^\alpha(0,0) + V^\alpha(1,1) \leq V^\alpha(0,1) + V^\alpha(1,0) \qquad (11)$$

$$[\phi(\hat{f}_p) \neq \phi(\hat{f}_q)] + 0 \leq [\phi(\hat{f}_p) \neq 0] + [\phi(\hat{f}_q) \neq 0] \qquad (12)$$

holds for any $\phi(\hat{f}_p)$ and $\phi(\hat{f}_q)$

if $\phi(\hat{f}_p) = 0$, $\phi(\hat{f}_q) = 0$ **then** (12) reduces to $0 + 0 \leq 0 + 0$

if $\phi(\hat{f}_p) = 0$, $\phi(\hat{f}_q) = 1$ **then** (12) reduces to $1 + 0 \leq 0 + 1$

if $\phi(\hat{f}_p) = 1$, $\phi(\hat{f}_q) = 0$ **then** (12) reduces to $1 + 0 \leq 1 + 0$

if $\phi(\hat{f}_p) = 1$, $\phi(\hat{f}_q) = 1$ **then** (12) reduces to $0 + 0 \leq 1 + 1$.

Finally, the hedgehog-convexity constraint $h(x_p, x_q) = [x_p = 1, x_q = 0]$ is submodular

$$h(0,0) + h(1,1) \leq h(1,0) + h(0,1) \qquad (13)$$

$$0 + 0 \leq 1 + 0. \qquad (14)$$

Optimal EC-Move: the authors in [22] showed how to find the global optimal solution of a submodular energy such as (7) by computing the min-cut of a graph that encodes the submodular energy. It should be noted that not all convexity priors lead to a submodular EC-Move, e.g. [4,13,14] are submodular while [8] is non-submodular which renders the EC-Move **NP**-hard.

4   Experiments

In this section k refers to the number of object seeds provided by the user. We applied k-convexity to liver and overlapping cells segmentation. When applicable we compared our approach to other forms of multi-convexity, i.e. k-regions [13] and k-disjoint [14]. Furthermore, we tested k-convexity on submodular [14] and non-submoudlar [8] convexity priors. Unless stated otherwise parts' convexity prior is assume to be hedgehog-convexity for all multi-convexity approaches. Also, user-seeds were used to compute color models and convexity constraints. In all of our experiments, spatial discontinuity costs, i.e. w_{pq}, were non-negative weights computed using a non-increasing function of the difference in p and q intensities, similar to [23].

4.1   Liver Segmentation

As shown in Fig. 5 column 2, k-regions usually resulted in k disjoint regions. For tight θ, using k-regions often lead to conflicting shape constraints between those regions. In those cases, there was no single contour that would satisfy the conflicting constraints, thus the liver was over segmented as k independent contours. As shown in column 3, k-disjoint was prone to local minima and sensitive to the order in which the foreground parts expanded. Unlike k-regions, our approach was more likely to result in liver segmentation with one connected-component, because each part/label shape constraints were enforced independently. Furthermore, our approach was more robust towards local minima compared to k-disjoint because of its relaxed solution space that allow parts to overlap.

Table 2 shows the average F_1 score of 3D liver segmentation over 12 different subjects. It is clear that our approach is more robust towards the selected θ in comparison to k-regions and k-disjoint. Table 3 shows the average number of connected-components of the segmentation results. In contrast to k-regions and k-disjoint, our approach was more likely to result in a liver segmentation with one connected-component. Note that none of the three methods guarantee connectivity of the shape parts, unless the user provided a single seed. Figure 6 shows a sample of 3D liver segmentation for three different subjects.

| seeds ■ + | k-regions [13] | k-disjoint [14] | k-convexity (ours) |
| ground truth ■ | | | |
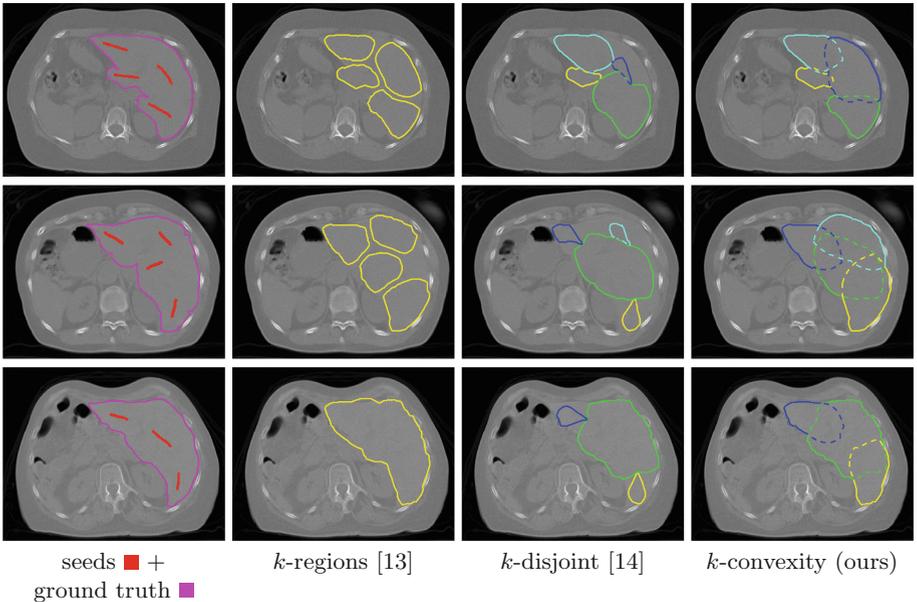
Fig. 5. Shows 2D liver segmentation of three different subjects. Different object parts are shown in different colors, their union is the liver segmentation (solid multi-colored contour). Also, 0 cost internal boundaries between parts are shown as dotted lines. k-regions was less likely to result in a single connected-component, especially for tight θ. k-disjoint was prone to local minima. Our approach out performed k-regions and k-disjoint. Results are shown for $\theta = 20°$, $\lambda = 1$ and $\mathcal{N}$ was the 8-neighborhood.

Table 2. Shows the average F_1 scores of 3D liver segmentation results for various smoothness λ and shape tightness θ. The three methods behave consistently over different values of λ. Unlike k-regions [13] and k-disjoint [14], our method is more robust towards θ. For $45° < \theta < 70°$ k-regions and our results were comparable. For $\theta \geq 70°$ all methods suffered from hedgehog discretization artifacts, i.e. under-constraining [24]

θ	$\lambda = 0.1$			$\lambda = 0.5$			$\lambda = 1$		
	k-reg.	k-dis.	ours	k-reg.	k-dis.	ours	k-reg.	k-dis.	ours
10°	0.07	0.39	**0.57**	0.06	0.38	**0.56**	0.06	0.29	**0.49**
20°	0.36	0.60	**0.79**	0.33	0.59	**0.79**	0.24	0.50	**0.73**
30°	0.82	0.63	**0.87**	0.83	0.61	**0.87**	0.83	0.61	**0.88**
45°	**0.84**	0.74	0.83	**0.86**	0.75	**0.86**	**0.87**	0.74	**0.87**

Table 3. Shows the average number of connected-components of 3D liver segmentation results. Ideally, the number of connected components should be 1. Our method was the most likely method to result in small number of connected-components, if not one.

θ	$\lambda = 0.1$			$\lambda = 0.5$			$\lambda = 1$		
	k-reg.	k-dis.	ours	k-reg.	k-dis.	ours	k-reg.	k-dis.	ours
10°	5.83	2.50	**1.92**	5.83	2.75	**2.00**	5.83	3.75	**2.67**
20°	4.58	1.25	**1.17**	4.58	1.42	**1.17**	5.00	2.00	**1.67**
30°	1.58	1.25	**1.00**	1.42	1.33	**1.00**	1.58	1.42	**1.08**
45°	1.08	**1.00**	**1.00**	1.08	**1.00**	**1.00**	1.08	**1.00**	1.08

4.2   Cells

Penalizing discontinuities between foreground parts is the main difference between segmenting overlapping objects, e.g. cells, and a multi-part object, e.g. liver. Unlike multi-part object segmentation, when segmenting cells we penalized the discontinuities between the foreground parts, i.e. cells, as follows

$$V(f_p, f_q) = w_{pq} \sum_{i \in \mathcal{L}} [f_p^i \neq f_q^i]. \tag{15}$$

Figure 7 shows various cell segmentation results. Figure 8 compares our approach to a specialized fluorescently stained cell nuclei segmentation approach [17]. In contrast to our approach, [17] used a more complex unary potential that took into consideration that overlapping regions are expected to be brighter than non-overlapping ones. This insight is specific to fluorescently stained nuclei.

4.3   Regular Convexity

Regular convexity is usually approximated by enforcing convexity constraints only along a predefined number of orientations [8] as illustrated in Fig. 1(a). In this section we will refer to regular convexity and its approximation [8] by regular convexity.
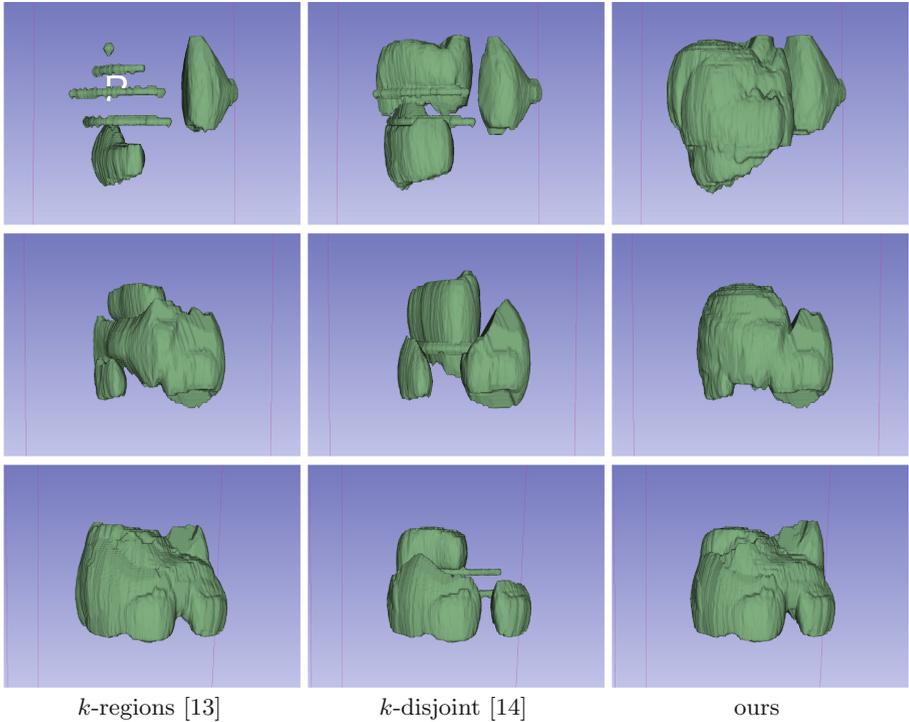
Fig. 6. Shows a sample of 3D liver segmentation results where each row corresponds to a different test subject. For most test cases, k-regions was sensitive to the selected θ while k-disjoint usually converged to a poor local minima. In contrast, our approach showed robustness towards the selected θ and k-disjoint local minima. These results were generated using $\theta = 20°$, $\lambda = 0.5$ and $\mathcal{N}$ was the 26-neighborhood system.

Enforcing regular convexity [8] renders energy (3) harder to optimize, since a single EC-move becomes non-submodular and therefore **NP**-hard. To address this problem we optimize each EC-Move with Trust Region based optimization proposed in [8], modifying it to account for the overlap between convex parts. We enforced convexity in an annealing fashion by gradually increasing the convexity term weight.

Figure 9 shows a proof of concept example for k- convexity when using regular convexity [8]. Based on our experience, employing regular convexity prior usually resulted in final segmentation with minimal overlap between parts. However, allowing labels to overlap helped during the optimization intermediate steps.
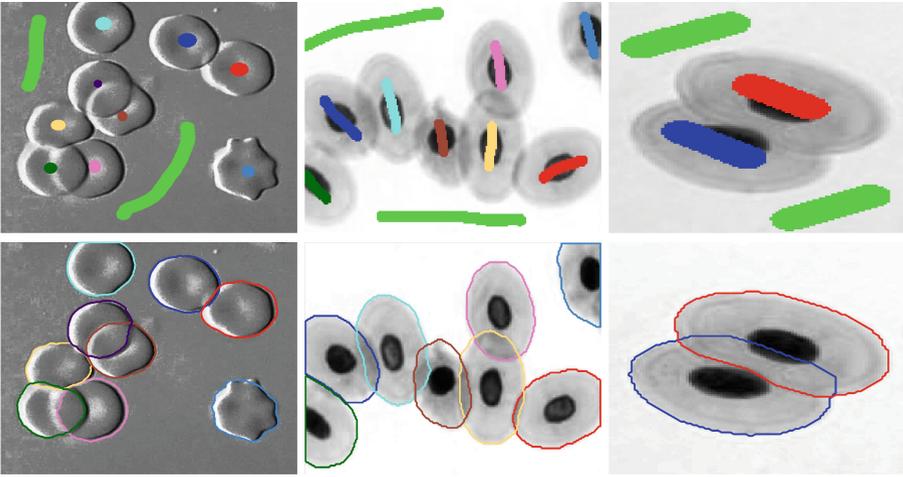
Fig. 7. (Top) shows user seeds, while (Bottom) shows our results. Left column is human red blood cells, while the others are frog blood cells. Results shown for $\theta = 5°$ and $\lambda = 1$.

user seeds   our approach   active-contours [17]
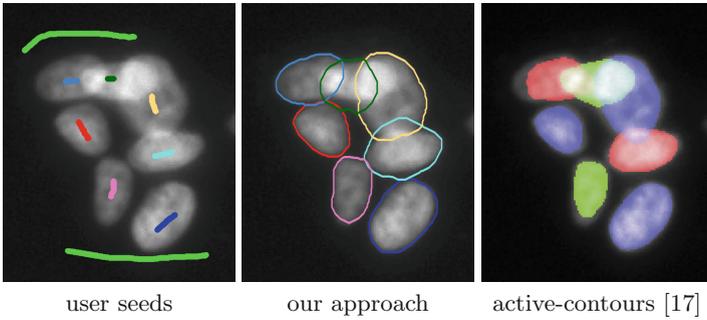
Fig. 8. Shows segmentation of fluorescently stained nuclei using k-convexity (with hedgehog-convexity) and [17]. Active-contours result copied from [17]. Note that [17] uses a different unary potential than ours, they assume overlapping regions are brighter.
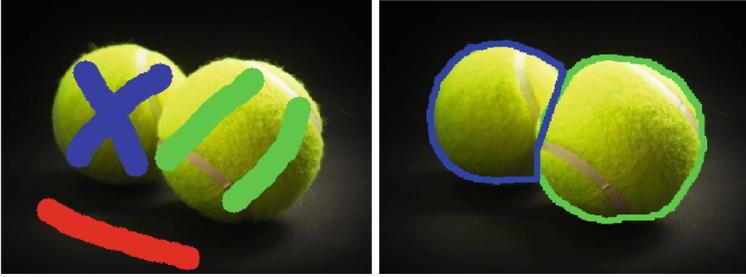
Fig. 9. (Left) shows user-seeds and (Right) shows our results when using regular (non-submodular) convexity [8]. Allowing the labels to overlap helped during the optimization intermediate steps by avoiding local minima. This result was generated using $\lambda = 0.01$ and w_∞ annealing schedule was $[0.002, 0.003, 0.005, 0.01, 0.1]$.

5   Conclusion

Our novel multi-convexity shape prior, i.e. k-convexity, regularizes an object segmentation under the assumption that it consists of k overlappable convex parts. We showed that k-convexity has higher shape representational power compared to existing multi-convexity priors [13,14]. In contrast to our approach, [13,14] use additional constraints that negatively impacts their shape representational power either to simplify the optimization problem [13] or to target a specific problem [14], i.e. multiple independent object segmentation. In addition, we empirically showed that k-convexity is more robust towards local minima and shape prior parameters compared to [13,14], respectively.

Our k-convexity approach is not tied to a specific type of convexity and could be used to enforce a multitude of convexity priors, e.g. star [4], geodesic-star [13], hedgehog [14], and regular convexity [8]. In addition, we illustrated the practicality of k-convexity when using hedgehog-convexity [14] in biomedical applications such as liver and overlapping cells segmentation.

Automating cell segmentation, i.e. dropping the user-seeds requirement, could be achieved by generating a large set of cell proposals, e.g. using Hough Transform for circles, and adding a sparsity prior [25] on top of k-convexity. By adding sparsity prior solutions that use fewer number of convex parts, i.e. cells, will become more favorable. The sparsity prior would not affect the EC-Move submodularity. However, in that case EC-Moves are expected to be prone to weak local minima. Thus, a more powerful move which would allow the removal of multiple parts simultaneously should be considered (a non-submodular move), we leave this as future work.

As discussed earlier multi-convexity priors do not impose parts connectivity. However, we empirically showed that k-convexity is more likely to result in a smaller number of connected components, if not one, compared to other multi-convexity approaches. In general, parts connectivity could be enforced by extending existing connectivity priors [26] to handle overlapping labels, but this will cause α-EC-Move to be non-submodular.

A   A NP-Hardness Proof

Optimization problem (3) is **NP**-hard. To prove this we will reduce a Set Cover problem instance to (3) in polynomial time. In Set Cover problem we are given a universe $U = \{u_1, u_2, \ldots u_n\}$, and a set of m subsets $S = \{S_i \subseteq U \mid \forall i \in [1, m]\}$. The Set Cover objective is to find the least number of subsets in S such that their union covers U. Given a Set Cover problem we can construct its corresponding k-convexity (3) instance as follows:

Label Set: $\mathcal{L} := \{i \mid \forall i \in [1, m]\}$ where label i corresponds to subset S_i.

Pixel Set: $\Omega := \{U \cup A\}$ where $A = \{a_i \mid \forall i \in [1, m]\}$. A is a set of auxiliary pixels/nodes. In this section we refer to pixels as nodes. For each set S_i, we introduce an auxiliary node a_i that will be used as an indicator of whether S_i is one of the selected sets to cover U or not.

Data Term: we define the data term as follows

$$
D_p(i) = \begin{cases} 0 & p \in U, \ p \in S_i & (16) \\ \infty & p \in U, \ p \notin S_i & (17) \\ 1 & p \in A, \ p = a_i & (18) \\ 0 & p \in A, \ p \neq a_i & (19) \end{cases}
$$

Equations (16) and (17) prohibit a node $u \notin S_i$ to gain label i. Equations (18) and (19) ensure that our energy increases by 1 if a_i is assigned to label i.

Neighbour System: in Set Cover there is no notion of neighborhood between the nodes, thus $\mathcal{N} := \phi$.

Shape Constraints: for a given set S_i we define the corresponding shape constraints as a set of pairwise edges $\mathcal{E}_i$ as follows:

$$ \mathcal{E}_i := \mathcal{C}_i \cup \mathcal{I}_i, $$

where the set of *connectedness* edges is $\mathcal{C}_i := \{(u, v) \mid \forall u, v \in S_i, u \neq v\}$ and the set of *indicator* edges is $\mathcal{I}_i := \{(u, a_i) \mid \forall u \in S_i, \ a_i \in A\}$. The edges in $\mathcal{C}_i$ ensure that if a node $u \in S_i$ gained label i then every other node in the set S_i will also gain label i. The edges in $\mathcal{I}_i$ ensures that if any node $u \in S_i$ gains label i then the corresponding auxiliary node a_i of S_i gains label i as well.

Objective: the reduced Set Cover problem (3) objective counts the number of selected subsets to cover U. Notice that if a node $u \in S_i$ decides to gain label i then a_i will also gain label i. And, since $D_{a_i}(i) = 1$ by definition then we can conclude that our energy counts the number of subsets used in the final solution.

References

1. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Patt. Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
3. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1133–1140. IEEE (2009)
4. Veksler, O.: Star shape prior for graph-cut image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5304, pp. 454–467. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88690-7_34
5. Boykov, Y., Kolmogorov, V., Cremers, D., Delong, A.: An integral solution to surface evolution PDEs via geo-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 409–422. Springer, Heidelberg (2006). https://doi.org/10.1007/11744078_32
6. Vu, N., Manjunath, B.: Shape prior segmentation of multiple objects with graph cuts. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008. pp 1–8. IEEE (2008)
7. Horváth, P., Jermyn, I., Zerubia, J., Kato, Z.: A higher-order active contour model for tree detection. In: 2006 18th International Conference on Pattern Recognition, ICPR 2006, vol. 2, pp. 130–133. IEEE (2006)
8. Gorelick, L., Veksler, O., Boykov, Y., Nieuwenhuis, C.: Convexity shape prior for binary segmentation. IEEE Trans. Patt. Anal. Mach. Intell. **39**(2), 258–271 (2017)
9. Isack, H., Veksler, O., Oguz, I., Sonka, M., Boykov, Y.: Efficient optimization for hierarchically-structured interacting segments (HINTS). In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
10. Delong, A., Boykov, Y.: Globally optimal segmentation of multi-region objects. In: International Conference on Computer Vision (ICCV) (2009)
11. Yin, Y., Zhang, X., Williams, R., Wu, X., Anderson, D.D., Sonka, M.: Logismos-layered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint. IEEE Trans. Med. Imaging **29**(12), 2023–2037 (2010)
12. Boykov, Y., Isack, H., Olsson, C., Ben Ayed, I.: Volumetric bias in segmentation and reconstruction: secrets and solutions. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
13. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: 2010 IEEE Conference on CVPR, pp. 3129–3136. IEEE (2010)
14. Isack, H., Veksler, O., Sonka, M., Boykov, Y.: Hedgehog shape priors for multi-object segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
15. Toranzos, F.A., Cunto, A.F.: Sets expressible as finite unions of starshaped sets. J. Geom. **79**(1–2), 190–195 (2004)
16. Aichholzer, O., Aurenhammer, F., Demaine, E.D., Hurtado, F., Ramos, P., Urrutia, J.: On k-convex polygons. Comput. Geom. **45**(3), 73–87 (2012)
17. Molnar, C., et al.: Accurate morphology preserving segmentation of overlapping cells based on active contours. Nat. Sci. Rep. **6**, 32412 (2016)
18. Qi, X., Xing, F., Foran, D.J., Yang, L.: Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. IEEE Trans. Biomed. Eng. **59**(3), 754–765 (2012)

19. Lee, H., Kim, J.: Segmentation of overlapping cervical cells in microscopic images with superpixel partitioning and cell-wise contour refinement. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 63–69 (2016)
20. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: 2008 IEEE conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
21. Siddiqi, K., Pizer, S.: Medial Representations: Mathematics, Algorithms and Applications, vol. 37. Springer, New York (2008)
22. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. IEEE Trans. Patt. Anal. Mach. Intell. **26**(2), 147–159 (2004)
23. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vis. (IJCV) **70**(2), 109–131 (2006)
24. Isack, H., Boykov, Y., Veksler, O.: A-expansion for multiple "hedgehog" shapes. Technical report (2016)
25. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minization with label costs. Int. J. Comput. Vis. (IJCV) **96**(1), 1–27 (2012)
26. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR) (2008)