Weakly-Supervised Sparse Coding with Geometric Prior for Interactive Texture Segmentation

Yuhui Quan, Huan Teng, Tao Liu and Yan Huang*

***Abstract*—Texture segmentation is about dividing a texture-dominant image into multiple homogeneous texture regions. The existing unsupervised approaches for texture segmentation are annotation-free but often yield unsatisfactory results. In contrast, supervised approaches such as deep learning may have better performance but require a large amount of annotated data. In this paper, we propose a user-interactive approach to win the trade-off between unsupervised approaches and supervised deep approaches. Our approach requires the user to mark one pixel in each texture region, whose label is directly propagated to its neighbor region. Such labeled data are of very small amount and even partially erroneous. To effectively exploit such weakly-labeled data, we construct a weakly-supervised sparse coding model that jointly conducts feature learning and segmentation. In addition, the geometric constraints are developed for the model to exploit the geometric prior on the local connectivity of region boundaries. The experiments on two benchmark datasets have validated the effectiveness of the proposed approach.**

***Index Terms*—Texture segmentation, sparse coding, weakly-supervised learning, geometric constraints**

I. Introduction

Texture regions of various types are prevalent in the images from daily life and real applications. Objects and backgrounds in natural scenes often consist of different textures [1]. An image about materials may encompass various texture regions. MRI (Magnetic Resonance Imaging) and CT (Computed Tomography), are about tissue cells modeled by mixtures of texture regions [2–4]. SAR (Synthetic Aperture Radar) images contain mountains, rivers and lands, which exhibit texture regions of different forms [5]. Separating such texture regions not only enables separate treatments to different image regions and objects, but also provides perceptual attributes and mid-level cues for visual recognition and understanding.

A closely-related topic in image processing is called texture segmentation [6, 7], whose task is about dividing a texture-dominant image into multiple disjoint homogeneous texture regions. Texture segmentation has its practical values in many fields, including the analysis on material, medical, biological and chemical images [3, 8–10], natural scene understanding [11], automatic navigation [12], remote sensing [5], etc.

Most traditional methods model texture segmentation as the patch-level clustering which involves two stages: *(i)* represent image patches by feature vectors (*e.g.* local spectral histogram [6, 13] and local regularity spectrum [14]); and *(ii)* conduct global segmentation via clustering in the feature space (*e.g.* k-means [6], graph cut [15] and mean-shift [16]).

Regarding the patch representation, traditional methods use hand-crafted features which are not adaptive to data. There are some approaches learning local features from input images for improvement. Khalilzadeh *et al.* [9] used sparse coding to extract patch features from MRI images. Rahmani *et al.* [5] used dictionary learning to obtain discriminative features for SAR images. Yuan *et al.* [7] applied low-rank factorization on the input to obtain the representative features of natural images. Regarding the feature clustering, most existing methods may ignore the spatial regularity of region boundaries. Thus, a few approaches [13, 17] use the Mumford-Shah models for enforcing global geometric properties on boundaries.

The performance of the two-stage approaches is extremely dependent on the effectiveness of the local features. For improvement, some approaches jointly conduct local representation and global clustering. Wang *et al.* [18] proposed a variational model that combines dictionary learning of image patches and Mumford-Shah active contours. Kiechle *et al.* [19] proposed to learn the filters from data such that the filtering responses tend to generate piece-wise constant label maps. All aforementioned approaches are unsupervised, whose results are often unsatisfactory, especially with unknown number of regions. Besides, since texture is a scale-related visual concept, users in different tasks may have different understandings on the texture regions being partitioned, even on the same images. Unsupervised methods are difficult to adapt to different cases.

Recently, a few supervised methods [3, 4, 20, 21] based on deep learning have shown improvement over the unsupervised one. Nevertheless, the deep learning needs an abundant of annotated data for supervision, the collection of which is expensive for texture segmentation. In particular, multi-time annotations are needed for an image, as subjective ambiguity probably exists on the boundaries of texture regions. In addition, the texture types may vary a lot over different segmentation tasks, making the transfer of annotated data infeasible. Furthermore, professional knowledge and specific devices are required in many fields [3]. To alleviate the requirement on annotations, Vincent *et al.* [20] used simple clustering and shallow segmentation to obtain the rough labels for supervision. Chen *et al.* [3] transferred the learned features from natural images for gland segmentation. Huang *et al.* [21] assumed all texture types are known and proposed an effective polygon-based training data generation scheme.

Unsupervised methods and supervised deep methods have their merits and weaknesses. To win their trade-off, we pro-

This research is supported by National Natural Science Foundation of China (61602184, 61872151, 61902130, 61672241, U1611461), Natural Science Foundation of Guangdong Province (2017A030313376, 2016A030308013), Fundamental Research Funds for the Central Universities (x2js-D2181690, x2js-D2190650), and China Postdoctoral Science Foundation (2019M652896).

All the authors are with School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China. Asterisk denotes the corresponding author (huangkaiyan@scut.edu.cn).

pose a user-interactive and learning-based approach under a configuration appealing to practice, which requires minimal effort on preparing labeled data with minimal user interaction involved. Note than many user-interactive methods have been proposed for natural image segmentation or segmenting textured objects from the background (*e.g.* [22, 23]), however, they are either inapplicable to texture segmentation due to the different characteristics between texture images and natural images, or with limited performance.

The proposed method is composed of a user-interaction scheme and a weakly-supervised sparse coding model. The interaction scheme uses a simple and friendly way to obtain a small amount of possibly partially erroneous labeled data from the input image, which are used for weak supervision. The sparse coding model utilizes the weakly-labeled regions to jointly conduct local feature learning via dictionary learning and global segmentation via linear prediction, with some geometric constraints defined by fixed points of opening/closing operators for imposing local connectivity prior on segmented region boundaries. The experimental results on two datasets demonstrated the effectiveness of the proposed method.

II. Proposed Method

A. User Interaction

Our user-interaction scheme is illustrated in Fig. 1. A user only needs to mark a pixel inside each texture region, which can be done by one click. Then we expand each marked point to a small square region which is of size $S \times S$ and centered at the marked point, and use the expanded regions as labeled data. Such labeled regions are small and even possibly erroneous as they may cover other texture regions besides the original ones. In other words, the supervision information provided by the labeled regions is likely incomplete and inaccurate which is rather weak. Using such weakly-labeled data for learning is task of the so-called weakly-supervised learning. Then, a weakly-supervised learning model is required for our task.

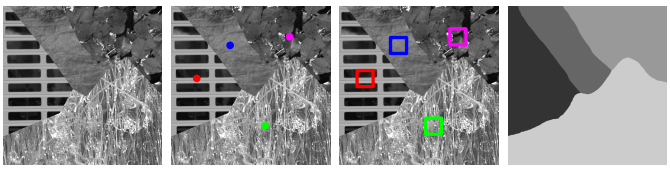

Fig. 1. Interface for user interaction. From left to right are input image, points marked by user, square regions expanded from marked points, and segmentation result of our method using the expanded labeled regions.

B. Sparse Coding Model

We establish a weakly-supervised learning model to make use of the labeled regions obtained from the user's interaction. Given an $M_1 \times M_2$ input image with K regions labeled, let $\mathbb{A}$ denote the indices of labeled pixels and $\bar{\mathbb{A}}$ denote those of unlabeled ones. Let $M = M_1 M_2$. Define $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_M] \in \mathbb{R}^{P \times M}$ where $\boldsymbol{y}_m$ denotes the feature vector extracted on the mth pixel. In implementation, we use the local LoG/Gabor spectrum feature of [7]. Let $\boldsymbol{L} = [\boldsymbol{l}_1, \cdots, \boldsymbol{l}_M] \in \{0,1\}^{K \times M}$ denote the label matrix with $\boldsymbol{l}_m$ as the one-hot label of $\boldsymbol{y}_m$ (*i.e.*, $\boldsymbol{l}_m(k) = 1$ if the m-th pixel belongs to the k-th region). We denote such an operation by $\mathcal{R} : \mathbb{R}^{1 \times M} \to \mathbb{R}^{M_1 \times M_2}$ which is to reshape the label vector to a label map and its inversion by $\mathcal{R}^{-1} : \mathbb{R}^{M_1 \times M_2} \to \mathbb{R}^{1 \times M}$ which is to vectorize a 2D label map. Given $\boldsymbol{L}_1 = \boldsymbol{L}(\mathbb{A})$, our goal is to predict $\boldsymbol{L}_2 = \boldsymbol{L}(\bar{\mathbb{A}})$.

We first relax the one-hot coding of $\boldsymbol{L}_2$ and predict $\boldsymbol{L}_2$ by

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{C},\boldsymbol{W},\boldsymbol{L}_2} \ & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_{\mathrm{F}}^2 + \beta_1 \|\boldsymbol{L}_1 - \boldsymbol{W}\boldsymbol{C}_1\|_{\mathrm{F}}^2 + \\ & \beta_2 \|\boldsymbol{L}_2 - \boldsymbol{W}\boldsymbol{C}_2\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{W}\|_{\mathrm{F}}^2 + \alpha \|\boldsymbol{L}_2\|_0, \\ \text{s.t.} \ & \mathcal{B}(\boldsymbol{L}_{[i]}) \in \mathbb{O}, \ \|\boldsymbol{c}_j\|_0 \le T, \ \|\boldsymbol{d}_k\|_2 = 1, \ \forall i,j,k, \end{aligned} \tag{1}$$

where $\beta_1, \beta_2, \lambda, \alpha > 0$ are four weights, $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_Q] \in \mathbb{R}^{P \times Q}$ denotes a dictionary with Q atoms, $\boldsymbol{W} \in \mathbb{R}^{K \times Q}$ is a linear classifier, $\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_M] \in \mathbb{R}^{Q \times M}$ denotes the sparse coding matrix with $\boldsymbol{C}_1 = \boldsymbol{C}(\mathbb{A})$ and $\boldsymbol{C}_2 = \boldsymbol{C}(\bar{\mathbb{A}})$, $\mathcal{B}$ denotes the mapping that sets all non-zero entries to 1, and $\mathbb{O} \subset \{0,1\}^{1 \times M}$ is the feasible set which imposes geometrical on segmentation boundaries. After obtaining $\boldsymbol{L}_2$, we transform it to one-hot coding by

$$\boldsymbol{L}_2^*(i,j) = \begin{cases} 1 & \text{if } \boldsymbol{L}_2(i,j) \ge \boldsymbol{L}_2(\bar{i},j) \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

There are five terms in the model of (1). The first term is the reconstruction term for feature learning, where the sparse codes under the learned dictionary are employed as the local features for segmentation. The second term is for weakly-supervised learning which uses the labeled regions to train a classifier and refine the sparse codes, such that the learned classifier correctly classifies the refined sparse codes associated with known labels. The third term is about pixel classification for segmentation, which applies the trained classifier to the unlabeled pixels. The fourth term is a classic regularization on the classifier. The last term is to enforce the sparsity on the predicted labels. Note that we do not impose one-hot constraint on $\boldsymbol{L}_2$ to avoid the difficulty in optimization.

Our model unifies sparse coding and segmentation effectively. Sparse coding extracts the underlying patterns of texture regions for segmentation, and the segmentation result guides better sparse coding. Such a loop is closed by iteratively refining the local features and region boundaries. Compared to reconstructive sparse models (*e.g.* [24]), ours jointly learns representations in patch space and infers the region boundaries in image space. Compared with supervised sparse models (*e.g.* [25, 26]) for classification, ours additionally considers the geometric property of spatial distribution of labels.

C. Geometric Constraints

The supervision from our expanded labeled regions may be rather weak. A direct use of model (1) without any constraint on the segmented results may be unstable; see *e.g.* Fig. 2. To increase the stability during learning, we define the feasible set $\mathbb{O}$ such that it encodes the local connectivity of region boundaries which is often seen in images. We characterize such local connectivity by the fixed point of the so-called *closing* operator and *opening* operator in image morphology [27].

Let $\otimes$ denote the 2D discrete convolution and $\boldsymbol{H}$ denote a structure element matrix (*e.g.* $\boldsymbol{H} = [1,1,1]^\top [1,1,1]$). For a

binary image $\boldsymbol{B}$, the closing operator $\mathcal{C}_{\boldsymbol{H}}$ and opening operator $\mathcal{O}_{\boldsymbol{H}}$ are defined by $\mathcal{C}_{\boldsymbol{H}}(\boldsymbol{B}) = (\boldsymbol{B} \oplus \boldsymbol{H}) \ominus \boldsymbol{H}$, $\mathcal{O}_{\boldsymbol{H}}(\boldsymbol{B}) = (\boldsymbol{B} \ominus \boldsymbol{H}) \oplus \boldsymbol{H}$. where $\oplus, \ominus$ are the *dilation* and *erosion* operators defined as $\boldsymbol{B} \oplus \boldsymbol{H} = \mathcal{B}(\boldsymbol{B} \otimes \boldsymbol{H})$, $\boldsymbol{B} \ominus \boldsymbol{H} = 1 - \mathcal{B}((1 - \boldsymbol{B}) \oplus \boldsymbol{H}^\top)$. When applied to an estimated binary label map, $\mathcal{C}_{\boldsymbol{H}}$ can remove some tiny holes from the map, while $\mathcal{O}_{\boldsymbol{H}}$ can remove some tiny isolated regions. The composition of these two operators, denoted by $\mathcal{M}_{\boldsymbol{H}}(\cdot) := \mathcal{O}_{\boldsymbol{H}}(\mathcal{C}_{\boldsymbol{H}}(\cdot))$, can improve the local connectivity of a label map.

For a binary image $\boldsymbol{B}$, after applying the composite operator $\mathcal{M}_{\boldsymbol{H}}(\cdot)$, the output $\mathcal{M}_{\boldsymbol{H}}(\boldsymbol{B})$ is a fixed point of $\mathcal{M}_{\boldsymbol{H}}(\cdot)$, *i.e.* $\mathcal{M}_{\boldsymbol{H}}(\mathcal{M}_{\boldsymbol{H}}(\boldsymbol{B})) = \mathcal{M}_{\boldsymbol{H}}(\boldsymbol{B})$. Therefore, for a binary label map, its local connectivity regularization can be expressed as the fixed point of $\mathcal{M}_{\boldsymbol{H}}$. Such a property motivated us to define the feasible set $\mathbb{O}$ of geometric constraints as follows:

$$\mathbb{O} = \{\boldsymbol{b} \in \{0,1\}^{1\times M} : \mathcal{R}(\boldsymbol{b}) = \mathcal{M}_{\boldsymbol{H}}(\mathcal{R}(\boldsymbol{b}))\}. \tag{3}$$

We illustrate the effectiveness of our geometrical constraints in Fig. 2. It can be seen that the constraints imposed by $\mathbb{O}$ have the effects of eliminating the tiny mis-segmented regions and 'denoising' the label map with improvement on the results.

Fig. 2. Effectiveness of the geometrical constrains. From left to right are input image, segmentation map without $\mathcal{B}(\boldsymbol{L}_{[i]}) \in \mathbb{O}$, segmentation result using $\mathbb{O}$, and ground truth segmentation map respectively.

D. Numerical Solver

We use an alternating iterative scheme to solve (1). First, we initialize the dictionary $\boldsymbol{D}^{(0)}$ and sparse code $\boldsymbol{C}^{(0)}$ by K-SVD [24]. The classifier $\boldsymbol{W}^{(0)}$ is initialized by least-square fitting on the labeled data. Afterwards, for $t = 0, 1, \cdots, T$, we alternatively update the unknowns $\boldsymbol{L}_2$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{W}$ as follows:

1) Label map update: At the beginning of the $(t+1)$th iteration, we update the label matrix $\boldsymbol{L}_2^{(t+1)}$ by solving

$$\min_{\boldsymbol{L}_2} \|\boldsymbol{L}_2 - \boldsymbol{W}^{(t)}\boldsymbol{C}_2^{(t)}\|_{\mathrm{F}}^2 + \frac{\alpha}{\beta_2}\|\boldsymbol{L}_2\|_0, \text{s.t. } \mathcal{B}(\boldsymbol{L}_{[i]}) \in \mathbb{O}, \forall i \tag{4}$$

which has the closed-form solution given by hard thresholding:

$$\boldsymbol{L}_2^{(t+\frac{1}{2})}(i,j) = \begin{cases} 0 & |\boldsymbol{G}^{(t)}(i,j)| \le \sqrt{\frac{\alpha}{\beta_2}} \\ \boldsymbol{G}^{(t)}(i,j) & \text{otherwise} \end{cases}. \tag{5}$$

where $\boldsymbol{G}^{(t)} = \boldsymbol{W}^{(t)}\boldsymbol{C}_2^{(t)}$. Afterwards, we have $\boldsymbol{L}^{(t+\frac{1}{2})}$ such that $\boldsymbol{L}^{(t+\frac{1}{2})}(\mathbb{A}) = \boldsymbol{L}_1$ and $\boldsymbol{L}^{(t+\frac{1}{2})}(\bar{\mathbb{A}}) = \boldsymbol{L}_2^{(t+\frac{1}{2})}$, which is projected onto the feasible set $\mathbb{O}$ by the following scheme. Let $\boldsymbol{S}_{[i]}^0 = \mathcal{B}(\boldsymbol{L}^{(t+\frac{1}{2})}), \boldsymbol{S}_{[i]}^1 = \mathcal{R}^{-1}(\mathcal{M}_{\boldsymbol{H}}(\mathcal{R}(\boldsymbol{S}_{[i]}^0)))$ indicate the support of $\boldsymbol{L}^{(t+\frac{1}{2})}$ before/after applying the morphology operator to $\boldsymbol{L}^{(t+\frac{1}{2})}$ respectively. Then $\boldsymbol{L}^{(t+1)}$ is calculated by

$$\boldsymbol{L}_2^{(t+1)}(i,j) = \begin{cases} \boldsymbol{G}^{(t)}(i,j) & \boldsymbol{S}^0(i,j) = 0 \ \& \ \boldsymbol{S}^1(i,j) = 1 \\ 0 & \boldsymbol{S}^0(i,j) = 1 \ \& \ \boldsymbol{S}^1(i,j) = 0 \\ \boldsymbol{L}_2^{(t+\frac{1}{2})}(i,j) & \text{otherwise} \end{cases}. \tag{6}$$

2) Sparse approximation: After $\boldsymbol{L}_2^{(t+1)}$ has been predicted, the sparse coding matrix is updated by solving

$$\min_{\boldsymbol{C}_i} \|\bar{\boldsymbol{Y}}_i - \bar{\boldsymbol{D}}_i\boldsymbol{C}_i\|_F^2 \ \text{ s.t. } \|\boldsymbol{c}_j\|_0 \le T, \forall j, \tag{7}$$

for $i = 1, 2$, where $\bar{\boldsymbol{Y}}_i = \left[\boldsymbol{Y}_i; \sqrt{\beta_i}\boldsymbol{L}_i\right]$, $\bar{\boldsymbol{D}}_i = \left[\boldsymbol{D}; \sqrt{\beta_i}\boldsymbol{W}\right]$. We solve the problem by orthogonal matching pursuit [28].

3) Dictionary learning: With $\boldsymbol{C}^{(t+1)}$ calculated, the dictionary is refined by solving

$$\min_{\boldsymbol{D}} \|\boldsymbol{Y}^{(t)} - \boldsymbol{D}\boldsymbol{C}^{(t)}\|_{\mathrm{F}}^2, \ \text{ s.t. } \|\boldsymbol{d}_q\|_2 = 1, \forall q. \tag{8}$$

Based on the proximal method [29], we update the dictionary atom by atom as follows: for $q = 1, \ldots, Q$,

$$\begin{cases} \boldsymbol{s}_q^{(t)} = \boldsymbol{d}_q^{(t)} - \frac{1}{\mu_q^t}\nabla_{\boldsymbol{d}_q}\mathcal{F}(\boldsymbol{C}^{(t)}, \widetilde{\boldsymbol{D}}_q^{(t)}; \boldsymbol{Y}) \\ \boldsymbol{d}_q^{(t+1)} = \boldsymbol{s}_q^{(t)}/\|\boldsymbol{s}_q^{(t)}\|_2 \end{cases}, \tag{9}$$

where μ_q^t is the step size, $\mathcal{F}(\boldsymbol{C}, \boldsymbol{D}; \boldsymbol{Y}) = \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_{\mathrm{F}}^2$, and $\widetilde{\boldsymbol{D}}_q^{(t)} = [\boldsymbol{d}_1^{(t+1)}, \cdots, \boldsymbol{d}_{q-1}^{(t+1)}, \boldsymbol{d}_q^{(t)}, \boldsymbol{d}_{q+1}^{(t)}, \cdots, \boldsymbol{d}_Q^{(t)}]$.

4) Classifier training: Once the dictionary is refined, the classifier is updated by solving

$$\min_{\boldsymbol{W}} \frac{\beta_1}{\lambda}\|\boldsymbol{L}_1 - \boldsymbol{W}\boldsymbol{C}_1^{(t)}\|_{\mathrm{F}}^2 + \frac{\beta_2}{\lambda}\|\boldsymbol{L}_2^{(t)} - \boldsymbol{W}\boldsymbol{C}_2^{(t)}\|_{\mathrm{F}}^2 + \|\boldsymbol{W}\|_{\mathrm{F}}^2, \tag{10}$$

whose explicit solution is given by $\boldsymbol{W}^{(t+1)} = (\beta_1\boldsymbol{L}_1\boldsymbol{C}_1^{(t)\top} + \beta_2\boldsymbol{L}_2\boldsymbol{C}_2^{(t)\top})(\beta_1\boldsymbol{C}_1^{(t)}\boldsymbol{C}_1^{(t)\top} + \beta_2\boldsymbol{C}_2^{(t)}\boldsymbol{C}_2^{(t)\top} + \lambda\boldsymbol{I})^{-1}$, which is calculated by conjugate gradient, since $(\beta_1\boldsymbol{C}_1^{(t)}\boldsymbol{C}_1^{(t)\top} + \beta_2\boldsymbol{C}_2^{(t)}\boldsymbol{C}_2^{(t)\top} + \lambda\boldsymbol{I})$ is positive definite.

III. Experiments

We evaluated our method on the Prague dataset [30] and the Histological dataset [31]. To simulate users' annotations, we locate the center of each region based on the ground-truth label map. To explore the influence of the expanded region size S, we tested our method using $S = 5, 9$ respectively. On both datasets, we set the number of dictionary atoms to 20 times as that of texture regions. The 21 metrics used in [19] are employed for evaluation. For interest, we also tested the extreme case using $S = 1$ labeled pixel each region.

A. Evaluation on Prague Dataset

The Prague dataset [30] contains 80 texture images synthesized using the regions randomly selected from 114 texture images of 10 categories. The number of texture regions varies from 3 to 12. We set the parameters $\beta_1 = 2.5, \beta_2 = 0.05, \lambda = 0.01$, $\alpha = 0.002$, and $T = 7$. Our method is compared to several unsupervised methods: RS [11], VMS [13], DL-SRC [32], ORTSEG [10], FSEG [7], and MLLIF [19]. Among them, FSEG is the most related to ours which uses the same features as ours, and MLLIF is the latest unsupervised method. For fair comparison with FSEG, we also use its post-processing scheme. We also modified FSEG [7], denoted by FSEG*, such that it utilizes the same annotation information as ours. In addition, we use FCNT-MK [20], a deep method for comparison.

The results are listed in Table I. With only one pixel labeled (*i.e.* $S = 1$), our method already outperforms the other

TABLE I
RESULTS ON PRAGUE DATASET. UP/DOWN ARROWS IMPLY LARGER/SMALLER VALUES ARE PREFERRED. BEST RESULTS ARE IN BOLD.

Metric	RS	VMS	DL SRC	ORT SEG	FSEG	FSEG*	MLL IF	FCNT -MK	Ours (S=1)	Ours (S=5)	Ours (S=9)
↑ CS	46.02	72.27	77.46	30.64	69.02	75.97	77.73	79.34	80.74	81.82	**84.18**
↓ OS	13.96	18.33	28.40	10.07	17.30	**3.38**	15.92	13.67	14.05	13.55	10.95
↓ US	30.01	9.41	**0**	6.99	11.85	5.53	6.31	6.25	0.44	0.49	**0**
↓ME	12.01	4.19	7.13	50.81	6.28	11.82	3.93	**3.80**	7.71	7.54	6.90
↓NE	11.77	3.92	7.39	49.74	5.66	11.49	3.92	**3.80**	8.70	7.80	7.04
↓O	35.11	7.25	8.58	30.14	10.79	9.12	7.68	**6.47**	7.08	7.28	6.84
↓C	29.91	**6.44**	29.48	28.76	13.75	9.34	24.24	22.88	9.56	9.27	7.09
↑CA	58.75	81.13	83.41	55.56	77.50	80.26	82.80	84.17	85.68	85.73	**87.05**
↑CO	68.89	85.96	87.36	67.44	84.11	88.09	86.89	87.97	90.06	90.52	**91.58**
↑CC	69.30	91.24	**95.16**	71.58	86.89	88.19	93.65	94.15	94.64	94.25	94.85
↓I	31.11	14.04	12.64	32.56	15.89	11.91	13.11	12.03	9.94	9.48	**8.42**
↓II	8.63	1.59	1.19	5.97	2.60	2.47	1.50	1.42	**1.17**	1.41	1.28
↑EA	65.87	87.08	89.70	66.51	83.99	87.4	88.03	88.97	91.52	91.57	**92.53**
↑MS	55.52	81.84	84.74	51.17	78.25	82.46	83.93	85.23	86.93	87.31	**88.63**
↓RM	10.96	4.45	2.42	7.31	4.51	2.99	3.27	3.12	2.32	2.24	**1.89**
↑CI	67.35	87.81	90.44	67.95	84.71	87.76	89.03	89.91	91.92	91.97	**92.86**
↓GCE	11.23	8.33	9.56	29.93	10.82	15.08	7.40	**6.46**	10.34	10.60	10.21
↓LCE	7.70	5.61	7.17	21.93	7.51	11.71	5.62	**4.75**	7.98	8.01	7.71
↓dD	18.52	9.06	9.08	25.21	–	10.34	8.57	–	8.05	7.84	**7.21**
↓dM	23.67	5.88	5.40	15.47	–	6.59	5.30	–	4.75	4.90	**4.40**
↓dVI	**13.31**	14.54	15.18	14.66	–	14.32	14.88	–	14.71	14.61	14.52

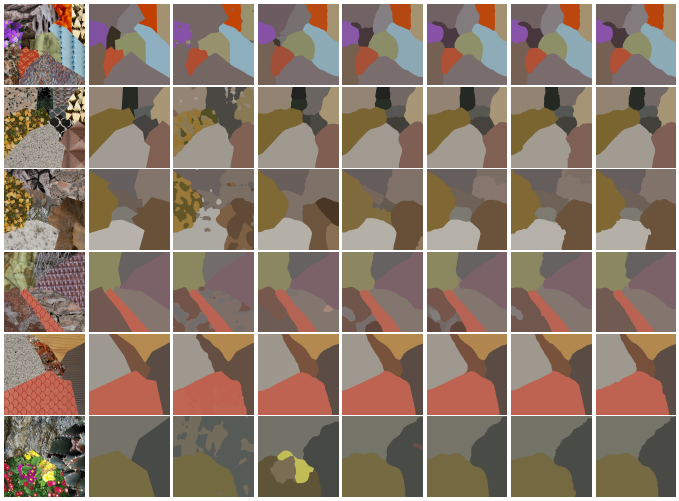

Fig. 3. Segmentation results on Prague dataset. Columns from left to right are input images, ground truth label maps, results of ORTSEG, results of FSEG, results of FSEG, results of FSEG*, and results of our method with $S = 1, 5, 9$ respectively.

compared methods on 11 metrics. As the marked region size increases, our results are improved. Our method using $S = 5$ has better results than using $S = 1$ in terms of 14 metrics. When $S = 9$, our method achieved the best results on 11 metrics, while other compared methods performed the best on no more than 5 metrics. See Fig. 3 for some visual results.

Sometimes the users' annotations are not guaranteed to fall onto the center of each region and may be placed around the boundaries. Then, the expanded marked points may go into other regions, making the labels $\boldsymbol{L}_1$ erroneous. To study the performance of our method in such cases, we conduct the robustness analysis experiment as follows. First, we divide each region into nine rings regions $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_9$ (from near to far). Secondly, we randomly pick up a point of the ith ring in each region as users' marking and run our method. By varying i we examine the influence of the annotation locations. The results are shown in Fig. 4. With the distances between the annotated points and region centers increased, our method performed worse. The performance change is small when the labeled points do not deviate the region center much.

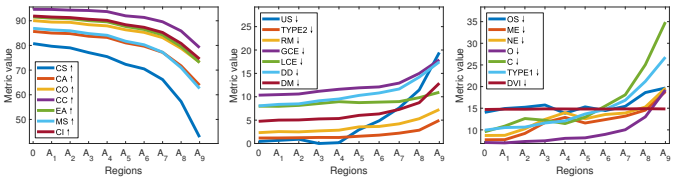

Fig. 4. Robust analysis for annotated points inside different ring regions.

B. Evaluation on Histology Dataset

The Histology dataset [31] consists of multiple real histological images of baboons of over 20 tissues types, with segmentation maps given by professional pathologists as the ground truths. Following [10], 36 sub-images containing two tissues are selected for the test. We set $\beta_1, \beta_2, \lambda, \alpha, T = 2.5, 0.01, 0.001, 0.002, 2$, and use DLSRC, ORTSEG, FSEG, FSEG* for comparison.

TABLE II
RESULTS ON HISTOLOGY DATASET. UP/DOWN ARROWS IMPLY LARGER/SMALLER VALUES ARE PREFERRED. BEST RESULTS ARE IN BOLD.

Method	↑ CS	↓OS	↓US	↓ME	↓NE	↓O	↓C	↑CA	↑CO	↑CC	↓I.
DLSRC	82.98	3.34	2.82	9.69	11.49	7.64	6.96	86.80	91.44	94.96	8.56
ORTSEG	66.50	6.59	3.98	23.13	23.60	13.83	11.44	80.56	87.37	92.25	12.63
FSEG	39.76	44.96	2.03	10.89	11.06	26.13	18.45	69.78	73.03	**95.30**	26.97
FSEG*	82.06	**1.78**	10.25	7.67	8.25	13.71	7.42	84.90	90.70	92.02	9.30
Ours(S=1)	80.69	6.68	**1.00**	6.47	7.62	8.39	7.34	85.26	90.11	94.77	9.89
Ours(S=5)	78.99	4.49	1.92	10.64	12.72	7.69	7.91	86.61	91.21	95.06	8.78
Ours(S=9)	**86.90**	3.29	1.94	**5.95**	**7.02**	**6.70**	**6.21**	**87.69**	**92.07**	95.28	**7.93**

Method	↓II.	↑EA	↑MS	↓RM	↑CI	↓GCE	↓LCE	↓dD	↓dM	↓dVI
DLSRC	5.08	92.60	87.17	6.55	92.89	10.42	6.77	7.54	12.88	5.88
ORTSEG	8.34	88.27	81.06	9.65	89.02	13.77	8.80	11.00	18.22	5.98
FSEG	5.91	80.72	68.46	17.53	82.39	9.81	7.61	16.30	23.60	7.37
FSEG*	11.40	90.66	86.05	7.42	91.00	**9.64**	**6.12**	7.65	14.00	**5.61**
Ours(S=1)	5.32	91.57	85.17	7.60	91.99	10.63	7.02	8.25	14.21	5.96
Ours(S=5)	**4.75**	92.46	87.15	6.55	92.79	10.27	6.62	7.58	12.95	5.93
Ours(S=9)	5.07	**93.15**	**88.11**	**5.82**	**93.40**	9.73	6.48	**6.90**	**11.99**	5.85

See Table II for the results. Among all compared methods except ours, FSEG* performed the best. In comparison, our method with $S = 1$ performed almost as good as FSEG*. With the marked region size increased, our method shows improvement. When $S = 5$, our method already outperformed FSEG*. When $S = 9$, our method achieved the best values on 14 metrics, while there are only 4 best metrics for FSEG*. Such results have demonstrated the effectiveness of our method.

IV. CONCLUSION

In this paper, we proposed a user-interactive approach to texture segmentation. The approach is built upon a weakly-supervised sparse coding model, which jointly conducts local feature learning and global segmentation, with geometric constraints imposing local connectivity on segmentation boundaries. The experiments show that the proposed method performs better than existing unsupervised methods while requiring minimal effort on annotation.

REFERENCES

[1] F. Zhang, X. Ye, and W. Liu, "Image decomposition and texture segmentation via sparse representation," *IEEE Signal Process. Letters*, vol. 15, pp. 641–644, 2008.

[2] Y. Kong, Y. Deng, and Q. Dai, "Discriminative clustering and feature selection for brain mri segmentation," *IEEE Signal Process. Letters*, vol. 22, no. 5, pp. 573–577, 2014.

[3] H. Chen, X. Qi, L. Yu, and P.-A. Heng, "Dcan: deep contour-aware networks for accurate gland segmentation," in *Proc. Conf. Comput. Vision Pattern Recog.* IEEE, 2016, pp. 2487–2496.

[4] M. M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. G. Mougiakakou, "Semantic segmentation of pathological lung tissue with dilated fully convolutional networks," *IEEE J. Biomed. Health Inform.*, 2019.

[5] M. Rahmani and G. Akbarizadeh, "Unsupervised feature learning based on sparse coding and spectral clustering for segmentation of synthetic aperture radar images," *IET Comput. Vision*, vol. 9, no. 5, pp. 629–638, 2015.

[6] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recog.*, vol. 24, no. 12, pp. 1167–1186, 1991.

[7] J. Yuan, D. Wang, and A. M. Cheriyadat, "Factorization-based texture segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3488–3497, 2015.

[8] S. Gorthi, M. B. Cuadra, P.-A. Tercier, A. S. Allal, and J.-P. Thiran, "Weighted shape-based averaging with neighborhood prior model for multiple atlas fusion-based medical image segmentation," *IEEE Signal Process. Letters*, vol. 20, no. 11, pp. 1034–1037, 2013.

[9] M. M. Khalilzadeh, E. Fatemizadeh, and H. Behnam, "Automatic segmentation of brain mri in high-dimensional local and non-local feature space based on sparse representation," *Magn. Resonance Imag.*, vol. 31, no. 5, pp. 733–741, 2013.

[10] M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovacevic, "Images as occlusions of textures: A framework for segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2033–2046, 2014.

[11] J. Yuan, D. Wang, and R. Li, "Image segmentation using local spectral histograms and linear regression," *Pattern Recog. Letters*, vol. 33, no. 5, pp. 615–622, 2012.

[12] M. R. Blas, M. Agrawal, A. Sundaresan, and K. Konolige, "Fast color/texture segmentation for outdoor robots," in *Proc. Int. Conf. on Intell. Robots and Syst.* IEEE, 2008, pp. 4078–4085.

[13] N. Mevenkamp and B. Berkels, "Variational multi-phase segmentation using high-dimensional local features," in *Conf. on Appl. Comput. Vision*. IEEE, 2016, pp. 1–9.

[14] N. Pustelnik, H. Wendt, P. Abry, and N. Dobigeon, "Combining local regularity estimation and total variation optimization for scale-free texture segmentation," *IEEE Trans. Computat. Imag.*, vol. 2, no. 4, pp. 468–479, 2016.

[15] J.-S. Kim and K.-S. Hong, "Color–texture segmentation using unsupervised graph cuts," *Pattern Recog.*, vol. 42, no. 5, pp. 735–750, 2009.

[16] S. Todorovic and N. Ahuja, "Texel-based texture segmentation," in *Proc. Int. Conf. on Comput. Vision*. IEEE, 2009, pp. 841–848.

[17] M. Storath, A. Weinmann, and M. Unser, "Unsupervised texture segmentation using monogenic curvelets and the potts model," in *Proc. Int. Conf. on Image Process.* IEEE, 2014, pp. 4348–4352.

[18] J. Wang and K. L. Chan, "Incorporating patch subspace model in mumford–shah type active contours," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4473–4485, 2013.

[19] M. Kiechle, M. Storath, A. Weinmann, and M. Kleinsteuber, "Model-based learning of local image features for unsupervised texture segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1994–2007, 2018.

[20] V. Andrearczyk and P. F. Whelan, "Texture segmentation with fully convolutional networks," *arXiv preprint arXiv:1703.05230*, 2017.

[21] Y. Huang, F. Zhou, and J. Gilles, "Empirical curvelet based fully convolutional network for supervised texture image segmentation," *Neurocomputing*, vol. 349, pp. 31–43, 2019.

[22] B. Ham, D. Min, and K. Sohn, "A generalized random walk with restart and its application in depth up-sampling and interactive segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2574–2588, 2013.

[23] A. V. Bhavsar, "An efficient weakly supervised approach for texture segmentation via graph cuts," *Journal of Intell. Syst.*, vol. 22, no. 3, pp. 253–267, 2013.

[24] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[25] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *Proc. Conf. Comput. Vision Pattern Recog.* IEEE, 2010, pp. 2691–2698.

[26] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 2013.

[27] H. Ji, Y. Luo, and Z. Shen, "Image recovery via geometrically structured approximation," *Appl. Computat. Harmonic Anal.*, vol. 41, no. 1, pp. 75–93, 2016.

[28] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst. Comput.* IEEE, 1993, pp. 40–44.

[29] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1356–1369, 2015.

[30] M. Haindl and S. Mikes, "Texture segmentation benchmark," in *Proc. Int. Conf. on Pattern Recog.* IEEE, 2008, pp. 1–4.

[31] J. A. Ozolek and C. A. Castro, "Teratomas derived from embryonic stem cells as models for embryonic development, disease, and tumorigenesis," in *Embryonic Stem Cells-Basic Biol. to Bioinf.* IntechOpen, 2011.

[32] S. Yang, Y. Lv, Y. Ren, L. Yang, and L. Jiao, "Unsupervised images segmentation via incremental dictionary learning based sparse representation," *Inform. Sciences*, vol. 269, pp. 48–59, 2014.