IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 4, 2020, accepted February 7, 2020, date of publication February 11, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973286

Content-Based Superpixel Matching Using Spatially Constrained Student's-t Mixture Model and Scale-Invariant Key-Superpixels

PENGYU WANG, HONGQING ZHU, (Member, IEEE), AND XIAOFENG LING, (Member, IEEE)
School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Corresponding author: Hongqing Zhu (hqzhu@ecust.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61872143, and in part by the Natural Science Foundation of Shanghai under Grant 19ZR1413400.

ABSTRACT This paper addresses an image matching methodology designed for correspondence problem in computer vision. Firstly, a novel superpixel segmentation model driven by spatially constrained Student's-*t* mixture model (SMM) is proposed. The tails of Student's *t*-distribution are heavier than that of traditional Gaussian distribution, therefore, SMM is more insensitive to outliers and noise. In this model, a spatially constraint term based on Markov random field (MRF) is designed, so that good boundary adherence and intensity homogeneity would be achieved. Next, by constructing an adaptive superpixel Gaussian filter and a superpixel salient detector, this paper establishes an innovative key-superpixel detection method by building a superpixel scale-space pyramid. Different from conventional keypoint based detection, two images could then be matched directly in a superpixel-to-superpixel manner. During the matching process, a combinatorial feature descriptor that merges color, shape, gradient and texture features is set up to distinguish each considered key-superpixel. One main advantage of this approach is that implementation time would be largely reduced by less matching demand for key-superpixels and few corresponding local features. Some experiments on datasets at the end would demonstrate a relatively better performance of our model.

INDEX TERMS Superpixel segmentation, spatially constrained Student's-*t* mixture model, key-superpixel detection, superpixel descriptor, superpixel matching.

NOMENCLATURE

i	Pixel index.
I	Input image.
H	Input image height.
W	Input image width.
N	Number of pixels.
V_x	Initial superpixel length.
V_y	Initial superpixel width.
j	Superpixel index.
K	Number of superpixels.
z_i	Observation value.
Z	Observation set.
Ω_i	Superpixel label of pixel.
f	Density function.
P	Joint conditional density function.
π_{ij}	Prior probability.
Π	Prior probability set.
S	Multivariate Student's *t*-distribution.
Θ_j	Model parameters set.
μ_j	Mean value.
Σ_j	Covariance.
v_j	Freedom degree of Student's *t*-distribution.
ζ	Mahalanobis squared distance.
Γ	Gamma function.
ψ	Digamma function.
Ξ	Smoothing prior.
M_{ij}	Smoothing factor.
L	Log-likelihood function.
J	Objective function.
∂	Neighborhood set.
m	Neighborhood index.
δ	Seed index of superpixel.
E	Image entropy.

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

r_{ij}	Posterior probability.
w_{ij}	Expected weight.
t	Number of iteration.
SG_j	Superpixel-based Gaussian kernel.
NG_j	Normalized superpixel-based Gaussian kernel.
SI	Superpixel intensity image.
SL	Superpixel image.
u_j	Pixel set of superpixel.
U_j	Pixel number of superpixel.
o_n	Intensity of pixel in superpixel.
SF	Superpixel Gaussian filtering image.
th	Threshold of saliency superpixel detector.
OS	Saliency superpixel set.
KS	Key-superpixel set.
$\bar{Z}_{pq}$	Zernike moment.
p	Order of Zernike polynomials.
q	Repetition of Zernike polynomials.
s	Superpixel standard distance.
F	Manhattan distance.
$\bar{O}$	Key-superpixel descriptor.

I. INTRODUCTION

Image matching has been widely used recent years due to their potential applications in surveillance systems [1], visual tracking [2], object recognition [3], etc. Nowadays, a majority of available algorithms adopt keypoint based matching scheme to stitching images [4], [5]. But some of these methods, including ORB [6] and SURF [7], may encounter difficulties when images are recorded under imperfect circumstances, including poor light, camera defocus, affine transforms, scene motion, etc. Besides, keypoint detection is usually realized by building a scale-space pyramid, such as SIFT [8]. It needs long matching time and high dimension feature vector. Therefore, pixel-based processing strategies may limit their use in real-time applications.

The essence of superpixel segmentation is partitioning an image into a number of connected and unified pixel groups with perceptual significance. Adopting superpixel images can minimize computational cost in subsequent feature extracting processes [9]–[11]. However, real-world image pairs with perfect alignment could hardly be found so that any slightly difference would lead to a completely different superpixel description.

This study is consisted of two main steps, image segmentation and matching based on superpixels. In the first part, the superpixel segmentation algorithm is driven by Student's-t mixture model (SMM), incorporated with a spatially constraint term Markov distribution, so that the superpixels obtained would achieve better boundary adherence and regularity. The main advantage of the Student's t-distribution is that it is heavily tailed than Gaussian distribution. Therefore, SMM provides a more powerful and flexible approach for probabilistic data clustering compared with the classical Gaussian mixture model (GMM). Besides, a fast and robust parameter estimation scheme would be realized by replacing covariance matrix with a fixed diagonal matrix associated with image global entropy as well. Next, a key-superpixel based image matching framework could be constructed by a series of processes. (i) Superpixel Gaussian kernel would be adopted to set up an adaptive superpixel filter; (ii) in terms of superpixel neighborhood of different levels, a saliency superpixel detector regarding based on color, shape, gradient & texture features would be developed; (iii) by collaborating the superpixel filter and feature detector, a multiscale superpixel pyramid would be designed for key-superpixel detecting. Finally, by extracting the centroid of each key-superpixels found, the two images could be matched.

The rest of this paper is organized as follows: Section II gives a brief view of some existing works on superpixel segmentation and feature matching. Section III presents the proposed superpixel segmentation model. Section IV proposes a model about multiscale key-superpixel detection and description. Section V discusses the experimental results and Section VI summarizes this paper.

II. RELATED WORKS

In this section, a review of some relevant algorithms on superpixel segmentation and keypoint matching is provided.

A. SUPERPIXEL SEGMENTATION

Many models have been reported for superpixel segmentation, and could be classified into three categories: graph-based, gradient-based, and clustering-based algorithms.

Graph-based approaches [12]–[15] took account of image brightness, contour and texture, and produced superpixels with relatively good visual compactness. A typical example of these methods is NC superpixel [12]. After this, Liu *et al.* [13] introduced entropy rate superpixel (ERS), which based on the entropy rate of random walk and an equilibrium term. It helps to create superpixels with high segmentation precision. Another commonly used graph-based model is lazy random walk (LRW) [14]. LRW mainly considers the texture cues of image. Recently, Zhou *et al.* [15] reported a bilateral geodesic distance superpixel (BGDS) generation strategy. Spatial and color distance difference between nodes in graph is combined, and the seed-dependent gradient formulation made this method achieve generally higher speed and better results.

Generally, gradient-based superpixel segmentation [16]–[18] can provide a good adherence to object boundaries. As one of the most popular and widely used gradient-based method, Turbopixels [16] adopted level set technology to build superpixels with linear computational complexity. In addition, watershed-based scheme has also been considered. For example, Machairas *et al.* [17] introduced marker-controlled watershed transformation for superpixel segmentation. They used spatially regularized gradient to achieve approximately identical sub-regions. Recently, Zhang *et al.* [18] proposed an efficient distance-based superpixel algorithm that could satisfied the boundary adherence and intensity homogeneity.

Clustering-based algorithms [19]–[25] set up superpixels using linear iterative strategy that had lower computational complexity compared to graph-based model. Achanta *et al.* [19] first applied k-means clustering method (SLIC) to generate superpixels. SLIC can control the balance between color and spatial similarity well with short running time. The intrinsic manifold SLIC [20] is an improved version of classical SLIC that can better handle superpixels with small size and high intensity in content-dense regions. Another popular superpixel approach called VCells [21], which was more advanced compared to SLIC on eliminating problems such as boundary crossing, collision, etc, was built based on the edge-weighted centroidal Voronoi tessellations. Besides, linear spectral clustering (LSC) [22] is another effective superpixel algorithm, which designs a kernel function to approximate similarity metric in high-dimensional feature space.

In a recent study [23], Xiao *et al.* introduced a content-adaptive superpixel (CAS) model. It adopted clustering-based discriminability measure to evaluate the importance of color, contour, texture, and spatial features, and achieved relatively moderate segmentation precision and regularity. More recently, the statistical-based superpixel approach driven by Gaussian mixture model scheme (GMMS) [24] was addressed to have the highest boundary recall, and relatively high running speed. Similar to GMMS, ultra-fast superpixel extraction method (USEAQ) [25] divides pixels into different groups in a one-pass mode through employing maximum posteriori probability.

B. KEYPOINT MATCHING

SIFT descriptor [8] is a general recognized method that detects keypoints by difference of Gaussian (DOG) pyramid. However, it is difficult to access real-time application due to the amount of computation needed for dominant gradient direction. As a similar but an accelerated version of SIFT, SURF [7] builds a keypoint descriptor by calculating the Haar-wavelet feature in a circular neighborhood. Alternatively, Calonder *et al.* [26] proposed BRIEF model to realize local feature description. BRIEF builds a 256-bit binary descriptor by selecting and comparing 256 pairs of random pixels. This approach is very fast, but has no rotation and scale invariance. Later, Rublee *et al.* [6] presented an ORB using a scale-space pyramid and corner measure. ORB solves the shortcoming that BRIEF is sensitive to rotation. In addition, Duval-poo *et al.* [27] proposed a scale invariant descriptor which relied on multiscale signal analysis framework. Many subsequent efforts focused on improving keypoint matching accuracy. In [28], Alcantarilla *et al.* structured a nonlinear scale-space detector by using a second-order partial differential equation to enhance the robustness against noise and photometric. Another leading approach that accelerate keypoint detection in nonlinear scale-space is named as A-KAZE [29]. Recently, some attention has been paid to match two images in superpixel manner. For example, a low dimensional superpixel descriptor (LDSD) for video correspondence estimation was presented by Du *et al.* [30]. They extracted shape, texture, and color features from superpixel. More recently, Yang *et al.* [31] introduced a novel superpixel region binary descriptor (SRBD) as a multilevel semantic feature for robust template matching. A rotation-invariant SRBD can be obtained by coding the orientation difference vector to one binary vector.

As deep learning technologies are developing, some new approaches have been introduced into the fields of keypoint detection and matching by means of convolutional neural networks [32]–[34]. For example, A research by Ono *et al.* [32] reported a deep neural network LF-Net that predicted keypoints. Even though learning-based approaches appear to see an improvement compared to traditional methods, training data is still very crucial. Hence, deep networks aligned features by minimizing distance function across the domains were used in some approaches [35], [36]. Besides, large number of manual labels on keypoints required also limits the expanding of deep networks in real-time operation.

III. SUPERPIXEL SEGMENTATION USING STUDENT'S-T MIXTURE MODEL

The implementation of our framework consists of three steps: (i) superpixel segmentation using spatially constrained Student's-t mixture model; (ii) multiscale key-superpixel detection and description; (iii) superpixel matching and stitching. Fig. 1 displays a block diagram of the whole framework.

A. SUPERPIXEL SEGMENTATION MODEL

Let i represents the pixel index of an input image I with height H and width W, where $i = (1, 2, \ldots, N)$, and $N = H \times W$ is the number of pixels. Our model adopts a five-dimensional vector $z_i = (l_i, a_i, b_i, x_i, y_i)^T$ to represent each pixel, where x_i and y_i denote the pixel coordinates in the image plane. (l_i, a_i, b_i) is its color components in CLELAB color space. For a specified number of superpixels K, length V_x and width V_y of the initial superpixel are defined as

$$V_x = V_y = \sqrt{W \times H/K}. \tag{1}$$

To label N pixels into K superpixels, each superpixel is associated with a Student's t-distribution.

Our scheme assumes that each observation z_i is independent of the label $\Omega_i \in [1, K]$. The density function of an observation $z_i = (l_i, a_i, b_i, x_i, y_i)^T$ is described by

$$f(z_i|\Pi, \Theta) = \sum_{j=1}^{K} \pi_{ij} S(z_i|\Theta_j), \tag{2}$$

where $\Pi = \{\pi_{ij}\}$, $i = (1, 2, \ldots, N)$ is the prior probability, which satisfies the following constraints

$$0 \leq \pi_{ij} \leq 1 \quad \text{and} \sum_{j=1}^{K} \pi_{ij} = 1, \tag{3}$$

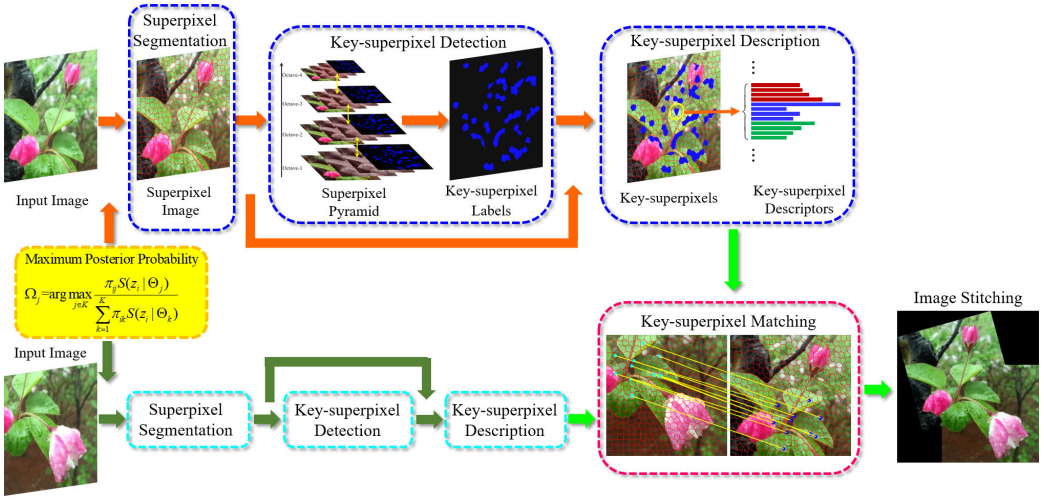

FIGURE 1. Framework of the proposed method.

and $S(z_i|\Theta_j)$ denotes the multivariate Student's t-distribution with mean μ_j, covariance Σ_j, and freedom degree v_j.

$$S(z_i|\Theta_j) = \frac{\Gamma\left((v_j+D)/2\right)|\Sigma_j|^{-\frac{1}{2}}}{(\pi v_j)^{\frac{D}{2}}\Gamma\left(\frac{v_j}{2}\right)\left[1+v_j^{-1}\zeta(z_i|\mu_j,\Sigma_j)\right]^{((v_j+D)/2)}}, \tag{4}$$

where $\Theta_j = \{\mu_j, v_j, \Sigma_j\}$, D is the observation variable dimension (here $D = 5$), $|\Sigma_j|$ is the determinant of Σ_j, $\zeta(z_i|\mu_j,\Sigma_j) = (z_i-\mu_j)^T\Sigma_j^{-1}(z_i-\mu_j)$ is the Mahalanobis squared distance, and $\Gamma(x) = \int_0^{+\infty} y^{x-1}\exp(-y)dy$ is the Gamma function. Thus, for observation set $Z = (z_1, z_2, \ldots, z_N)$, the joint conditional density of observation data can be modeled by

$$P(Z|\Pi,\Theta) = \prod_{i=1}^{N} f(z_i|\Pi,\Theta) = \prod_{i=1}^{N}\sum_{j=1}^{K}\pi_{ij}S(z_i|\Theta_j). \tag{5}$$

Superpixel segmentation is sensitive to illumination change, noise, and wiggly boundaries. Therefore, in this study, we introduce MRF distribution [37] to consider the spatial correlation between the pixels.

$$P(\Pi) = A^{-1}\exp\left\{-\frac{1}{B}\Xi(\Pi)\right\}, \tag{6}$$

where A and B are normalizing parameters and $\Xi(\cdot)$ is the smoothing prior. According to Bayes' rules, the posterior probability density function (PDF) can be defined as

$$P(\Pi,\Theta|Z) \propto P(X|\Pi,\Theta)\cdot P(\Pi). \tag{7}$$

Then, the log-likelihood function of (7) could be written as

$$\begin{aligned} L(\Pi,\Theta|Z) &= \log\left(P(\Pi,\Theta|Z)\right) \\ &= \sum_{i=1}^{N}\log\{\sum_{j=1}^{K}\pi_{ij}S(z_i|\Theta_j)\} - \log A - \frac{1}{B}\Xi(\Pi). \end{aligned} \tag{8}$$

There are various selections for smoothing prior, our model selects smoothing prior $\Xi(\Pi)$ used by Nguyen *et al.* [37].

$$\Xi(\Pi) = \sum_{i=1}^{N}\sum_{j=1}^{K} M_{ij}\log\pi_{ij}, \tag{9}$$

where M_{ij} is the smoothing factor. It is defined as a superpixel that is a weighted sum of neighborhood posterior probability r_{mj} and prior probability π_{mj}.

$$M_{ij} = \exp\left[\frac{1}{N_i}\sum_{m\in\partial_i}(r_{mj}+\pi_{mj}+\frac{1}{K})\right], \tag{10}$$

where ∂_i represents the set of neighborhoods around the observation z_i, and N_i stands for the number of pixels in the window. We regard the initial prior probability $1/K$ as a regularization term. For a square window of size 5×5, $N_i = 25$. From (10), it is quite obvious that the weighted sum of neighborhood posterior probability can effectively improve the robustness of model against noise. From (6) and (9), the final MRF distribution function is in the form

$$P(\Pi) = A^{-1}\exp\left\{-\frac{1}{B}\sum_{i=1}^{N}\sum_{j=1}^{K}M_{ij}\log\pi_{ij}\right\}. \tag{11}$$

Based on above equations, the complete log-likelihood function (8) of proposed model is rewritten as

$$\begin{aligned} L(\Pi,\Theta|Z) = &\sum_{i=1}^{N}\log\{\sum_{j=1}^{K}\pi_{ij}S(z_i|\Theta_j)\} - \log A \\ &- \frac{1}{B}\sum_{i=1}^{N}\sum_{j=1}^{K}M_{ij}\log\pi_{ij}. \end{aligned} \tag{12}$$

Finally, according to the Jason's inequality [38], the hidden variable, i.e. posterior probability r_{ij} is introduced in following objective function. Then, maximizing the log-likelihood

function in (12) would lead to the maximizing of the following expression.

$$J(\Pi, \Theta|Z) = \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left\{\log \pi_{ij} + \log S(z_i|\Theta_j)\right\} - \log A - \frac{1}{B}\sum_{i=1}^{N}\sum_{j=1}^{K} M_{ij}\log \pi_{ij}. \quad (13)$$

So far, the superpixel segmentation model has been finished.

B. PARAMETER INITIALIZATION

Parameter initialization has some influence on model performance. This paper records the initial prior probability as

$$\pi_{ij} = 1/K, \quad i = (1, 2, \ldots, N),\ j = (1, 2, \ldots, K), \quad (14)$$

To check the seed index of the j-th superpixel in observation set Z conveniently, we define the seed index of the j-th superpixel as

$$\delta = V_x(j \mod \frac{W}{V_x}) + jV_xV_y + \frac{1}{2}(WV_y + V_x). \quad (15)$$

Then, the color component $\mu_{j,c}$ of mean μ_j are calculated as

$$\mu_{j,c}(l_j, a_j, b_j) = \frac{1}{N_\delta}\sum_{m\in\partial_\delta} z_{m,c}(l_m, a_m, b_m), \quad (16)$$

where ∂_δ represents the neighborhood set of the δ-th observation. N_δ stands for the number of neighboring pixels around the seed. And a square window of size 7×7 is used in our method. The spatial component $\mu_{j,s}$ of mean μ_j is

$$\mu_{j,s}(x_j, y_j) = z_{\delta,s}(x_\delta, y_\delta). \quad (17)$$

This definition would be able to incorporate the consideration of neighborhood pixel intensity value and local spatial information. Therefore, our superpixel model could withstand the influence of noise.

During the initialization, the covariance is set to a 5 by 5 diagonal matrix. Since the direct optimization of covariance matrix for a large number of superpixels is very expensive, we provide an approach which each element of the covariance matrix would be fixed to a constant by adopting entropy. The reasons why entropy might be effective are shown below: (i) entropy is a statistical measure of the uncertainty associated with random variable that provides a natural way of finding disorder contained in an image; (ii) if an image has rich detail information, its global entropy is relatively large. In this case, if the intensity of each pixel is regarded as an observation, the variance of each uncertain observation variable (pixel value) tends to become greater also. Therefore, entropy can characterize information as well by describing the uncertainty of data. In our method, each element in the variance matrix, represented by entropy, is defined as [39]

$$E = -\sum_{g=0}^{G-1} \lambda_g \log \lambda_g, \quad (18)$$

where λ_g is the probability of the grayscale g in given image, which can be obtained by gray histogram. And G represents the number of grayscale. For example, a 8-bit gray image allows $G = 256$ grayscales (from 0 to 255). In this way, the covariance matrix is defined as follows.

$$\Sigma_j = \alpha \cdot E \cdot \mathbf{I} = \alpha \cdot \begin{bmatrix} E & 0 & 0 & 0 & 0 \\ 0 & E & 0 & 0 & 0 \\ 0 & 0 & E & 0 & 0 \\ 0 & 0 & 0 & E & 0 \\ 0 & 0 & 0 & 0 & E \end{bmatrix}, \quad j = (1, 2, \ldots, K), \quad (19)$$

where $\mathbf{I}$ is identity matrix, and α is a regularization parameter. Then, we rewrite the Mahalanobis squared distance as

$$\zeta(z_i|\mu_j, \Sigma_j) = |z_i - \mu_j|^2 \Big/ \alpha E. \quad (20)$$

C. SUPERPIXEL PARAMETER ESTIMATION

Given the Student's t-distribution (4), the objective function in (13) can be rewritten in the form

$$\begin{aligned} J(\Pi, \Theta|Z) = &\sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left\{\log \pi_{ij} - \frac{1}{2}\log|\Sigma_j| \right. \\ &\left. + \log\Gamma\left(\frac{v_j + D}{2}\right) - \log\Gamma\left(\frac{v_j}{2}\right)\right\} \\ &- \sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij}\left\{\frac{D}{2}\log(\pi v_j) \right. \\ &\left. + \frac{v_j + D}{2}\log\left[1 + v_j^{-1}\zeta(z_i|\mu_j, \Sigma_j)\right]\right\} \\ &- \sum_{i=1}^{N}\sum_{j=1}^{K} M_{ij}\log \pi_{ij}, \end{aligned} \quad (21)$$

where parameters A and B are set to 1 for simplicity. The unknown parameter $\Theta_j = \{\mu_j, v_j, \Sigma_j\}$ can be estimated by maximizing the log-likelihood function (21). Thus, the posterior probability can be calculated as

$$r_{ij}^{(t+1)} = \frac{\pi_{ij}^{(t)} S(z_i|\Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} S(z_i|\Theta_k^{(t)})}, \quad (22)$$

We maximize $J(\Pi, \Theta|Z)$ over the mean μ_j for obtaining the following updating function

$$\frac{\partial J}{\partial \mu_j} = \sum_{i=1}^{N} r_{ij}\left[-\Sigma_j^{-1}\frac{\alpha E(v_j + D)(\mu_j - z_i)}{\alpha E v_j + |z_i - \mu_j|^2}\right]. \quad (23)$$

In the expression above, let

$$w_{ij}^{(t+1)} = \frac{\alpha E(v_j^{(t)} + D)}{\alpha E v_j^{(t)} + \left|z_i - \mu_j^{(t)}\right|^2}. \quad (24)$$

By calculating $\partial J(\Pi, \Theta|Z)/\partial \mu_j = 0$, the estimation of μ_j can be obtained at the $(t+1)$ step.

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} r_{ij}^{(t)} w_{ij}^{(t)} z_i}{\sum_{i=1}^{N} r_{ij}^{(t)} w_{ij}^{(t)}}. \quad (25)$$

Setting the derivative of the objective function $J(\Pi, \Theta|Z)$ with respect to π_{ij} at the $(t+1)$ iteration step, and using the Lagrange multiplier η_i for each pixel, we have

$$\frac{\partial}{\partial \pi_{ij}}\left[J-\sum_{i=1}^{N}\eta_i\left(\sum_{j=1}^{K}\pi_{ij}-1\right)\right]=\frac{r_{ij}}{\pi_{ij}}-\frac{M_{ij}}{\pi_{ij}}-\eta_i. \quad (26)$$

From (26), the estimates π_{ij} can be computed as follows:

$$\pi_{ij}^{(t+1)}=\frac{r_{ij}^{(t)}-M_{ij}^{(t)}}{\sum_{k=1}^{K}\left(r_{ik}^{(t)}-M_{ik}^{(t)}\right)}. \quad (27)$$

The fixed covariance matrix (19) that help reduce computing cost would be applied to all Student's t-distributions.

Next, to optimize the degrees of freedom v_j, we need to set the derivative of function $J(\Pi, \Theta|Z)$ with respect to v_j at the $(t+1)$ iteration step using

$$\psi\left(\frac{v_j^{(t)}+D}{2}\right)-\psi\left(\frac{v_j^{(t+1)}}{2}\right)-\log\left(\frac{v_j^{(t)}+D}{2}\right)+\log\left(\frac{v_j^{(t+1)}}{2}\right)+1+\frac{\sum_{i=1}^{N}r_{ij}^{(t)}\left(\log w_{ij}^{(t)}-w_{ij}^{(t)}\right)}{\sum_{i=1}^{N}r_{ij}^{(t)}}=0, \quad (28)$$

where $\psi(x)=\partial\left(\ln\Gamma(x)\right)/\partial x$ is the digamma function. The solution of (28) does not exist in a closed form. A closed form approximation of this equation has been devised heuristically by Shoham [40].

After the parameter estimation is finished, the label of each pixel could be calculated using

$$\Omega_i=\underset{j\in K}{\operatorname{argmax}}\frac{\pi_{ij}S(z_i|\Theta_j)}{\sum_{k=1}^{K}\pi_{ik}S(z_i|\Theta_k)}. \quad (29)$$

To have our model strengthen the connection between boundaries, the post-processing [19] should factor into the generated superpixels: (i) superpixel, which size is smaller than V_x pixels, should be merged into other adjacent superpixels in terms of color information; (ii) impose a morphological closing operation on each superpixel, and subtract the original superpixel from its result. The obtained pixels are reallocated to the nearest superpixels for smoother boundary. After post-processing, some small superpixels are removed, and superpixel becomes more regular. The steps of the proposed superpixel segmentation model can be summarized as Algorithm 1.

Algorithm 1 Spatially Constrained Student's-t Mixture Model Superpixel Segmentation

Input: Image I, superpixels number K.
Output: The superpixel label $\Omega_i \in [1, K]$ of each pixel.
1: Initialize the color and spatial components of μ_j using (16) and (17), π_{ij} using (14), and entropy-based Σ_j using (19), respectively.
2: For $t = 1$ to iterations **do**
3: **Update** r_{ij}, w_{ij}, μ_j, π_{ij}, and v_j using (22), (24), (25), (27) and (28).
4: End for
5: Compute the superpixel label for each pixel using (29), and generate K superpixels.
6: Superpixel refinement for pleasant visual effects.

IV. MULTISCALE KEY-SUPERPIXEL DETECTION AND DESCRIPTION

The detection of key-superpixel would be implemented by three steps: (i) superpixel Gaussian filter; (ii) saliency superpixel detector; (iii) scale-space saliency superpixel detection.

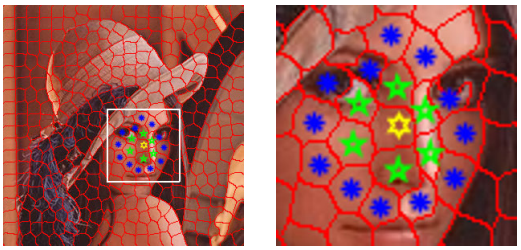

FIGURE 2. Superpixel and its neighborhood, where green pentagon represents first-level neighborhood and blur asterisk stands for second-level.

A. KEY-SUPERPIXEL DETECTION

1) DESIGNING SUPERPIXEL GAUSSIAN FILTER

Superpixels are often used to replace pixel-grid to promote speed. In this study, we design a superpixel Gaussian filter. Unlike the pixel-grid scheme, as shown in Fig. 2, the number, shape and size of each superpixel's neighborhood are uncertain. In order to describe our algorithm, we specify that the superpixels directly adjacent to central superpixel (see the place indicated by yellow hexagon) are the first-level neighborhood (marked in a green pentagon). Those who directly adjacent to its first-level neighborhood are the second-level neighborhood (drawn in blur asterisk).

Next, this paper defines a 2-dimensional superpixel-based Gaussian kernel function as follows:

$$SG_j(X_m, Y_m)=\frac{1}{2\pi\sigma^2}\exp\left[\frac{(X_m-X_j)^2+(Y_m-Y_j)^2}{2\sigma^2 s^2}\right], \quad (30)$$

where $s=\sqrt{W\cdot H/K}$ is the superpixel standard distance, and σ^2 is the standard deviation, which is similar to traditional Gaussian distribution. m refers to the index of neighborhood superpixel, $m \in \partial_j$. The distribution is assumed to have zero mean. For the j-th superpixel, ∂_j is its neighborhood superpixel set including itself. (X_j, Y_j) is the centroid of j-th superpixel. By normalizing kernel function (30), we have

$$NG_j(X_m, Y_m)=\frac{SG_j(X_m, Y_m)}{\sum_{k\in\partial_j}SG_j(X_k, Y_k)}. \quad (31)$$

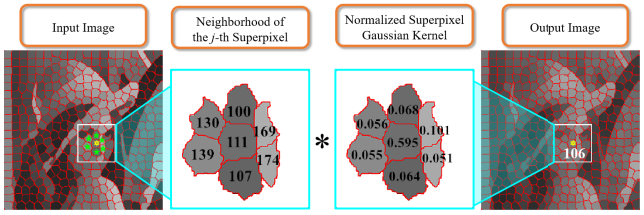

FIGURE 3. The process of superpixel Gaussian filtering.

Thus, we could obtain each superpixel value, by calculated

$$SI(j) = \frac{1}{U_j} \sum\nolimits_{n \in u_j} o_n, \tag{32}$$

where u_j and U_j denote the pixel set and the pixel number of the j-th superpixel, o_n represents the intensity of the n-th pixel in the j-th superpixel. Applying our superpixel Gaussian filter to the superpixel intensity image by updating the value of each superpixel with the following expressing (33), and this image at the moment could be regarded as a superpixel Gaussian filtering image.

$$SF(j) = \sum_{m \in \partial_j} SI(m) \cdot NG_j(X_m, Y_m). \tag{33}$$

Fig. 3 shows the block diagram of the proposed superpixel Gaussian filter with a zero mean and a standard deviation of 0.5. In this diagram, first-level neighborhood is used.

2) SALIENCY SUPERPIXEL DETECTOR

In this study, we define an effective saliency superpixel detector engaging the consideration of neighbouring superpixels. This detector could help detect two types of salient superpixels. By applying detector to the j-th candidate superpixel, first-level neighbourhood superpixels would help identify corner-like saliency superpixels, and second-level neighborhood superpixels could be used to find the superpixels with a significant difference on concentrations with the surroundings. Superpixel that satisfies either one of the following neighbourhood conditions ((34) or (35)) would be regarded as a saliency superpixel.

We label the j-th candidate superpixel as a saliency superpixel if $\text{Int}(0.5 \cdot N_j^{(1)} + 0.5)$ superpixels in its first-level neighborhood are conformed to

$$|SI(m^{(1)}) - SI(j)| \geq th^{(1)}, \quad m^{(1)} \in \partial_j^{(1)}. \tag{34}$$

Then, this candidate superpixel is defined as a corner-like prominent superpixel. If at least $\text{Int}(0.8 \cdot N_j^{(2)} + 0.5)$ superpixels exist in its second-level neighborhood which satisfy

$$\left[(SI(m^{(2)}) - SI(j) > 0) \vee (SI(m^{(2)}) - SI(j) < 0)\right] \wedge \left[\sum\nolimits_{m^{(2)} \in \partial_j^{(2)}} |SI(m^{(2)}) - SI(j)| \geq th^{(2)}\right]. \tag{35}$$

The intensity of this candidate superpixel would be significantly different from those around it. In (34) and (35), $th^{(1)}$ (or $th^{(2)}$) stands for the threshold. $\text{Int}(\cdot)$ is the integer-valued function. $SI(j)$ denotes the value of candidate superpixel, $N_j^{(1)}$ (or $N_j^{(2)}$) is the number of pixels in the first-level neighborhood $\partial_j^{(1)}$ (or second-level neighborhood $\partial_j^{(2)}$). $SI(m^{(1)})$ (or $SI(m^{(2)})$) represents the value of the $m^{(1)}$-th (or $m^{(2)}$-th) superpixel in the first-level (or second-level) neighborhood.

3) SCALE-SPACE SALIENCY SUPERPIXEL DETECTION

Our approach is inspired by the work of pixel-level keypoint detection [41]. In order to achieve scale invariance, which is crucial for saliency superpixel, a scale-space pyramid using superpixel Gaussian filter and saliency superpixel detector is constructed. Our scale-space pyramid layers consist of four octaves, and each octave is characterized by superpixel Gaussian filtering with standard deviation σ shown stacked in Fig. 4. After defining the superpixel scale-space pyramid, we adopt superpixel detector to obtain the saliency superpixel sets of each octave and record them as $OS^{(1)}$, $OS^{(2)}$, $OS^{(3)}$, and $OS^{(4)}$, respectively. The rule of detecting key-superpixel is designed to be an idea that key-superpixel needs to be detected as a saliency superpixel in all octaves. Thus, we gather the intersection of these sets as the key-superpixel set KS.

$$KS = OS^{(1)} \cap OS^{(2)} \cap OS^{(3)} \cap OS^{(4)}. \tag{36}$$

We summarize the scale-space pyramid in Algorithm 2.

Algorithm 2 Superpixel Scale-Space Pyramid (See Fig.4)

Input: Original image I.
Output: Key-superpixel set KS
1: For $t = 1$ to 4 **do**
2: **If** $t = 1$ **do**
3: **Compute** superpixel image $SL^{(t)}$ using **Algorithm 1**
4: **Else**
5: **Compute** $SL^{(t)}$ by sub-sampling $SL^{(t-1)}$
6: **End if**
7: **Compute** superpixel intensity image $SI^{(t)}$ of $SL^{(t)}$ using (32)
8: **Compute** superpixel Gaussian filtering image $SF^{(t)}$ of $SI^{(t)}$ using (33)
9: **Compute** saliency superpixel sets $OS^{(t)}$ of $SF^{(t)}$ using (34) and (35)
10: End for
11: Compute key-superpixel set KS using (36)

B. KEY-SUPERPIXEL DESCRIPTORS

In order to match images successfully, the key-superpixel in target image that has high similarity with the key-superpixel obtained in current image needed to be found. This paper designs a superpixel descriptor to describe the local features of each key-superpixel, so that we can select similar superpixel pairs between the current image and the reference image. The low-level visual features of a superpixel, such as color, shape, gradient and texture are directly related to the

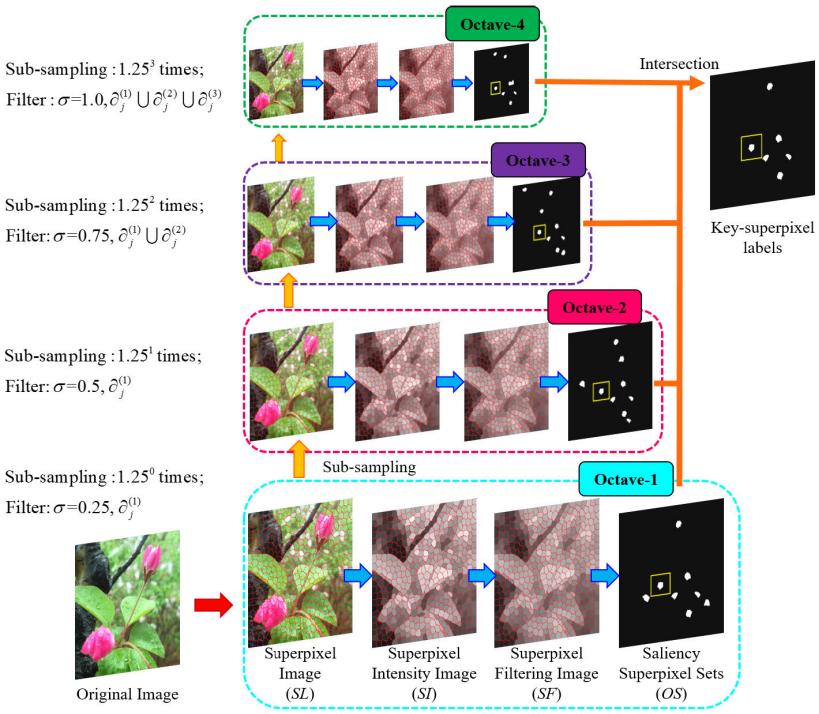

FIGURE 4. Block diagram of superpixel scale-space pyramid.

superpixel content. These features are encoded into a high-dimensional feature vector.

1) SUPERPIXEL COLOR DESCRIPTOR (HSVHist)

Color information is the most elementary local feature of images [30]. Considering that intensity information in RGB (red, green, blue) color space is susceptible to illumination, we use the three channels of HSV (hue, saturation, value) color space as the color feature. This can be achieved by computing the 16 bins histogram of H, S, and V channels, respectively. In this way, the color feature of 48-dimensional vector could be obtained.

2) SUPERPIXEL SHAPE DESCRIPTOR (SZM)

The matching of superpixels independent of their position and orientation is important in our model. The second part of our superpixel feature descriptor is based on Zernike moment invariants, which would not be affect by rotation transform. Zernike moment is orthogonal inside the unit circle [42]. The 2-D continuous Zernike moment of an image intensity function $I(\rho, \theta)$ is defined as follows

$$\bar{Z}_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 I(\rho, \theta) V_{pq}^*(\rho, \theta) \rho \, d\rho d\theta, \quad (37)$$

where Zernike polynomials of order p with repetition q is defined as

$$V_{pq}^*(\rho, \theta) = R_{pq}(\rho) \exp(-\mathrm{i}q\theta). \quad (38)$$

The real-value Zernike radial polynomials is defined by

$$R_{pq}(\rho) = \sum_{h=0}^{(p-|q|)/2} \frac{(-1)^h (p-h)! \rho^{p-2h}}{h!((p+|q|)/2-h)!((p-|q|)/2-h)!}, \quad (39)$$

where $p \geq |q| \geq 0$, and $(p-|q|) \bmod 2 = 0$.

Due to the irregular geometry of superpixels, preprocessing is needed before calculating Zernike moment of superpixel. Fig. 5 shows a diagram of computing Zernike moment for superpixel. Taking the centroid as the center of a square, the side of it is $1.2s$. Then, for each square $\widehat{I}(x, y)$ containing superpixel (see Fig. 5), the image coordinate transformation to the interior of the unit circle is given by

$$\begin{aligned} \rho &= \sqrt{(d_1 x + d_2)^2 + (d_1 y + d_2)^2}, \\ \theta &= \tan^{-1}\left(\frac{d_1 y + d_2}{d_1 x + d_2}\right), \end{aligned} \quad (40)$$

with

$$d_1 = \sqrt{2}/(\widehat{N} - 1) \quad \text{and } d_2 = -1/\sqrt{2}, \quad (41)$$

where $\widehat{N}$ denotes the number of pixel of $\widehat{I}(x, y)$. The magnitude of Zernike moment is regarded as a rotation invariant feature of the underlying superpixel. Thus, our rotation invariant descriptor is constructed as an 8-dimensinal vector listed below

$$\left[|\bar{Z}_{00}|, |\bar{Z}_{11}|, |\bar{Z}_{20}|, |\bar{Z}_{22}|, |\bar{Z}_{31}|, |\bar{Z}_{33}|, |\bar{Z}_{40}|, |\bar{Z}_{42}|\right]. \quad (42)$$

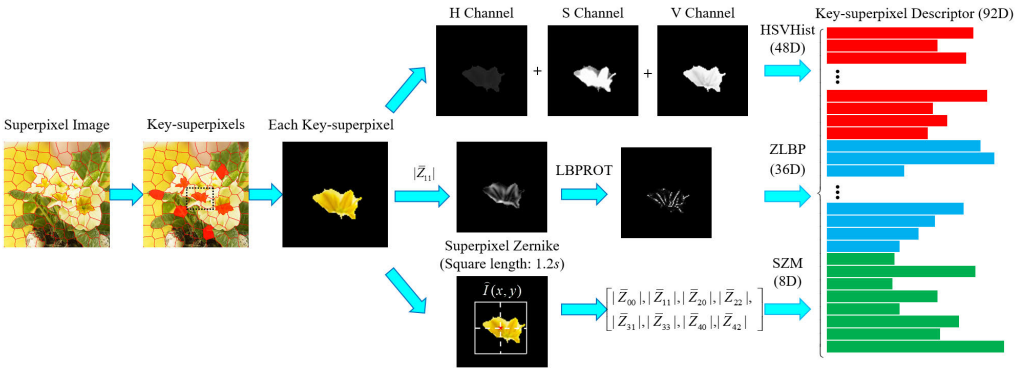

FIGURE 5. Superpixel feature descriptor.

3) SUPERPIXEL GRADIENT AND TEXTURE DESCRIPTOR (ZLBP)

An important step for our matching model is to describe each key-superpixel by extracting gradient feature. However, gradient is sensitive to noise. To overcome this problem, the one order Zernike moment descriptor has been taken into account to this feature vector, because $|\bar{Z}_{11}|$ contains oriented gradient feature and is rotation invariant to graphic transform. Therefore, for each pixel inside superpixel, we calculate its Zernike moment $|\bar{Z}_{11}|$ in a small window of 7×7. Thus, the resulted image reflects the gradient information of the image (see Fig. 5).

Texture feature is another important feature closely related to human perception since users recognize objects through regular patterns of the spatial arrangement of pixel. To describe the texture feature, rotation invariant local binary pattern (LBPROT) [43] would be applied. For each superpixel, the histogram of LBPROT is calculated using method in [43], and thus resulting in a 36-dimensional vector. Therefore, the proposed feature vector integrates the oriented gradient with texture features, so that extracted features are rotationally invariant to texture feature.

Finally, by combining the 48-dimensional color descriptor, 8-dimensional shape descriptor and 36-dimensional gradient & texture descriptor, we establish a 92-dimensional local feature descriptor of each key-superpixel. Fig. 5 shows the framework of our superpixel descriptor.

C. KEY SUPERPIXEL MATCHING AND IMAGE STITCHING

Since one key-superpixel could only be matched with one in target image, key-superpixel pairs need to be collected from the possible selections. We could calculate Manhattan distance to obtain real key-superpixel pairs with the following formula.

$$F(\bar{O}_1, \bar{O}_2) = \sum_{c=1}^{C} \left|\bar{O}_1(c) - \bar{O}_2(c)\right|, \tag{43}$$

where $\bar{O}_1$ and $\bar{O}_2$ are two different key-superpixel descriptors, and the feature dimension $C = 92$.

Next, the centroid of each matched key-superpixel would be extracted and used to implement corresponding image stitching using APAP algorithm [44].

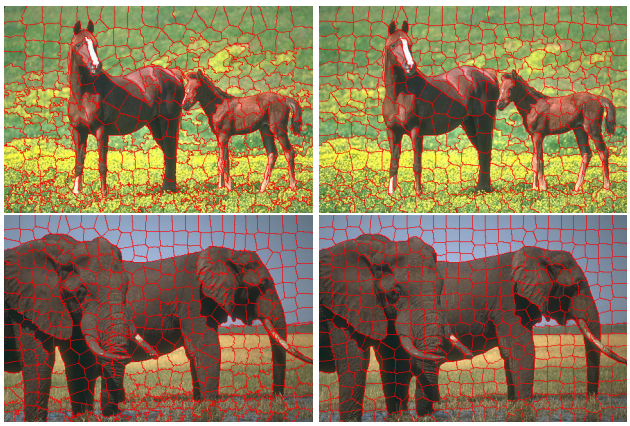

FIGURE 6. Some examples of superpixel segmentation using our method, the first column without constrained term; the second column with MRF-based spatially constrained term.

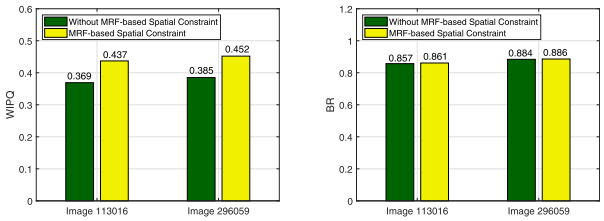

FIGURE 7. Quantitative evaluation of superpixel segmentation on two natural images (113016 and 296059) from BSDS500 dataset.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section will comprehensively evaluate the proposed method. All algorithms are carried out on the Intel(R) Core(TM) i5-9400f, 4.1GHz, and 6GB NVIDIA GTX 1660 Ti with MATLAB 2018a in Window 10 system. To evaluate the effectiveness of the proposed method, several state-of-the art algorithms are considered for comparison, including TP [16], ERS [13], SLIC [19], LRW [14], LSC [22], GMMS [24], SIFT [8], SURF [7], ORB [6], A-KAZE [29], and LDSD [30] respectively.

A. DATASETS DESCRIPTION AND EVALUATION CRITERIA

All test images are collected from three public available datasets: Berkeley segmentation dataset (BSDS500) [45], Oxford dataset [29], and Iguazu dataset [28]. The Berkeley

dataset consists of 500 natural images with the resolution of 481 × 321, or 321 × 481. While the Oxford dataset contains blur, light, zoom, rotation, JPEG compression. Each category contains six test images. The Iguazu dataset is composed of six images with size of 900 × 675 pixels, each of them is contaminated by Gaussian noise. As image index number increases, the level of noise varies will increase as well.

The performance of superpixel segmentation is quantitatively evaluated by four criteria, boundary recall (BR) [22], under-segmentation error (UE) [22], achievable segmentation accuracy (ASA) [24] and weighted isoperimetric quotient (WIPQ) [24]. BR testifies the percentage of superpixels boundaries coinciding with ground truth boundaries

$$\mathrm{BR} = \mathrm{SP}/\mathrm{GP}, \tag{44}$$

where SP is the number of boundary pixels in segmentation results which meet the condition that at least one pixel in the 3 × 3 neighborhood should be the boundary pixel of ground truth. GP stands for total boundary pixel numbers of the segmentation results. High BR represents the number of real boundaries that are rarely missed. UE computes the proportion of over-segmentation superpixels, while UE value comes close to zero, superpixels would approaches to the ground truth. UE is defined by

$$\mathrm{UE} = -1 + \frac{1}{N}\sum\nolimits_{|u_j \cap u_\gamma| > \omega |u_j|} |u_j|, \tag{45}$$

where u_j and u_γ are the pixel sets of the j-th superpixel and ground truth, respectively. Parameter ω is set to 0.05 for well-established [24]. The lower the UE, the fewer superpixels across multiple objects. Similar to UE, ASA measures the extent accuracy of superpixel segmentation could achieve when each of them is assigned a ground truth label that covers the biggest portion. A higher ASA indicates better segmentation accuracy.

$$\mathrm{ASA} = \frac{1}{N}\sum\nolimits_{j=1}^{K} \max\left\{|u_j \cap u_\gamma|\right\}. \tag{46}$$

In addition to above metrics for the segmentation accuracy, we also evaluate the regularity by WIPQ

$$\mathrm{WIPQ} = \frac{1}{N}\sum\nolimits_{j=1}^{K} \frac{4\pi |u_j|^2}{\mathrm{BP}_j^2}, \tag{47}$$

where BP_j is the number of the boundary pixels of j-th superpixel.

The following four metrics are used to measure the performance of superpixel matching approaches: *Accuracy*, Matching score (MS), *Recall*, and *Precision*. The higher values they have, the better performance descriptor would achieve.

Accuracy can also be called *repeatability*, which measures the ratio of correct matching between the detected keypoints in two images of the same scene

$$Accuracy = \frac{CK}{MK} \times 100\%, \tag{48}$$

where CK is the number of corresponding keypoints and MK is the minimum number of detected keypoints in both images.

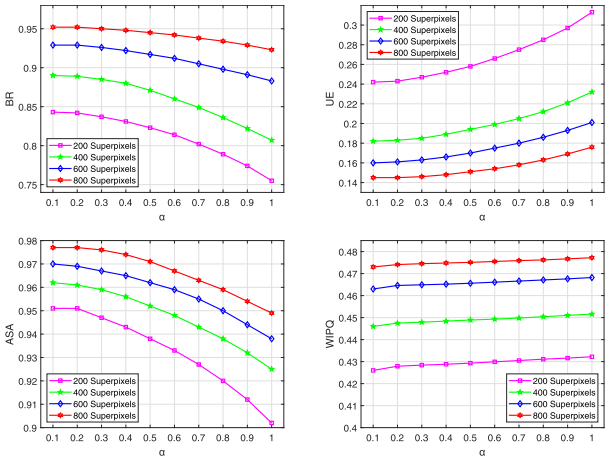

FIGURE 8. Performance of our method with difference parameters α.

MS functions as an accuracy assessment and is defined as follows

$$\mathrm{MS} = \frac{\#correct\ matches}{\#features}. \tag{49}$$

This expression describes the number of initial features that would result in correct matches. *Recall* is computed as a ratio where the number of corrected matches divided by total number of correspondences (possible correct matches), defined as

$$Recall = \frac{\#correct\ matches}{\#correspondences}. \tag{50}$$

The number of correct matches out of total matches is represented by *Precision*.

$$Precision = \frac{\#correct\ matches}{\#correct\ matches + \#false\ matches}. \tag{51}$$

B. PRIOR ANALYSIS

The first experiment would discuss the effect of the MRF-based spatially constrained term on superpixel segmentation. Two performances are engaged, one with constrained term, and the other without. Fig. 6 shows the segmentation results to two nature images (113016 and 296059) in the BSDS500 dataset. It could be seen that by incorporating constrained term into Student's-t mixture model, the effect of noise could be reduced due to the filtering characteristics of MRF-based spatially constrained term. Also, this method tends to be less sensitive to noise, and superpixels generated show better boundary adherence and regularity. We also qualitatively evaluate the superpixel segmentation results of these images. As illustrated in Fig. 7. The proposed approach with constrained term gives better segmentation for complex scene according to the BR and WIPQ.

C. PARAMETER SETTINGS

Our approach has one primary parameter α in (19). This parameter is related to the selection of covariance matrix

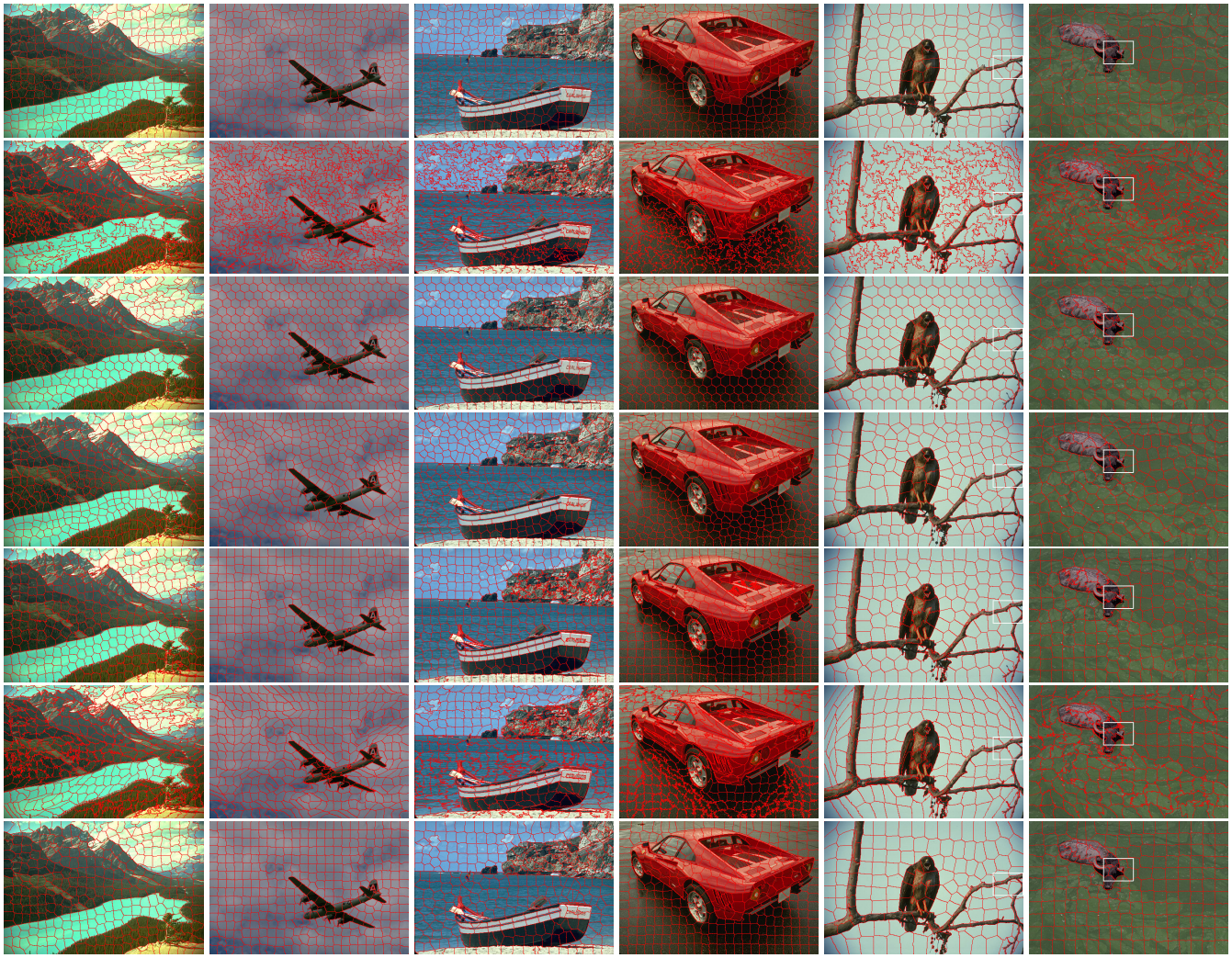

FIGURE 9. Superpixel segmentation results, from top to down is TP, ERS, SLIC, LRW, LSC, GMMS, and Ours, respectively.

which helps our superpixel segmentation implement more effectively. This subsection conduces an experiment to discuss the setting of parameter α. We set the parameter α from 0.1 to 1 continuously and obtain the corresponding quantitatively evaluation results. Fig. 8 shows the values of BR, UE, ASA, and WIPQ versus varying values of α. It is clear from the figure that the most suitable empirical value of α would be around 0.2, because the less value of α will lead to a less steep probability distribution slope that affects the discrimination of pixel.

D. SUPERPIXEL SEGMENTATION

In this subsection, we compare SMMS to several popular algorithms including TP,[1] ERS,[2] SLIC,[3] LRW,[4] LSC,[5] and GMMS.[6] For these algorithms, the implementations are based on publicly available codes from their respective websites. Fig. 9 provides the segmentation results of each approach, where 400 superpixels are extracted in four images and 200 in the other two. As shown in Fig. 9, ERS shows delicate segmentation details. But because of the rough boundary, the overall visual effect seems to be the worst among seven. LRW tends to be slightly better than TP. LSC outperforms GMMS in terms of regularity. Synthetically, LSC and GMMS show considerably competitive visual effect. Our approach achieves similar regularity with the current advanced LSC. An area of interest is amplified as displayed in Fig. 10. The comparison in this figure demonstrates that the accuracies of LRW and TP are moderately poor. The segmentation by TP tends to have indiscernible boundaries between different areas. Fig. 11 provides our experimental results of different superpixels, and each image is segmented approximately into 800\400.

[1] http://www.cs.toronto.edu/ babalex/research.html
[2] https://github.com/mingyuliutw/ers
[3] http://ivrl.epfl.ch/research/superpixels
[4] https://github.com/shenjianbing/lrw14
[5] http://jschenthu.weebly.com/projects.html
[6] https://github.com/ahban/GMMSP

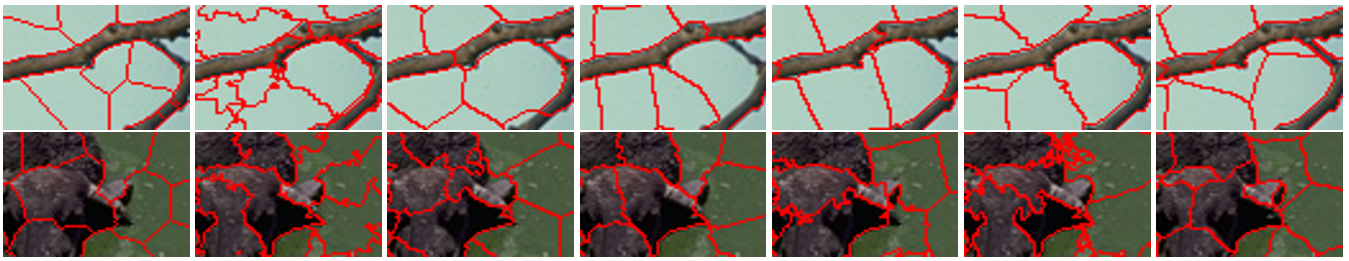

FIGURE 10. Visual comparison of detailed parts (200 superpixels), from left to right is TP, ERS, SLIC, LRW, LSC, GMMS, and Ours, respectively.

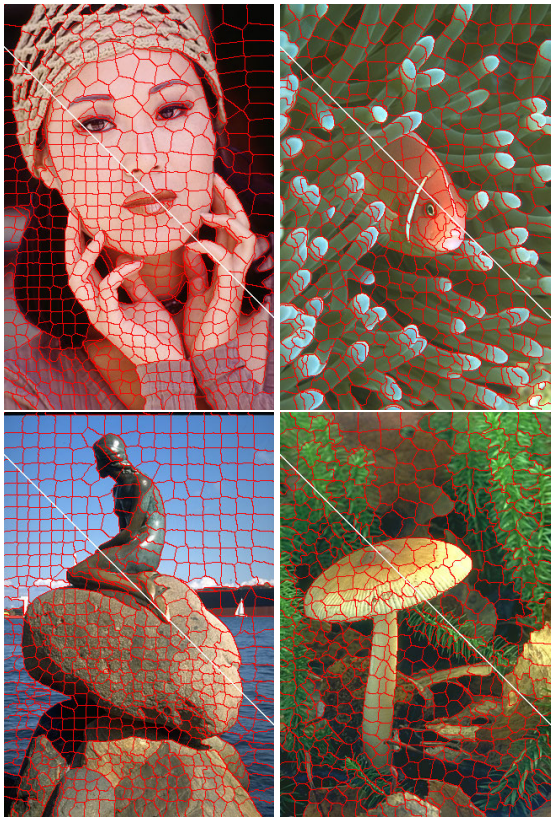

FIGURE 11. Images are segmented approximately into 800\400 superpixels using our method.

The quality obtained by the proposed method is subjectively assessed and compared to other algorithms. Fig. 12 illustrates the comparative performance between our algorithm and other approaches on test images of BSDS500. All the results given are derived from averaging the results. The numbers of superpixels are set to 200, 400, 600, and 800, respectively. From this figure, we can arrive at the following conclusions: (i) both GMMS and LSC obtain good compactness and accuracy since BR is higher than 0.93; (ii) superpixel generated by LRW and TP have more regular shapes, and the WIPQ is much higher than any other methods at above 0.53; (iii) as shown in Fig. 12, the UE value in our method is the lowest among all methods, this means that a better compactness of superpixel segmentation can be achieved.

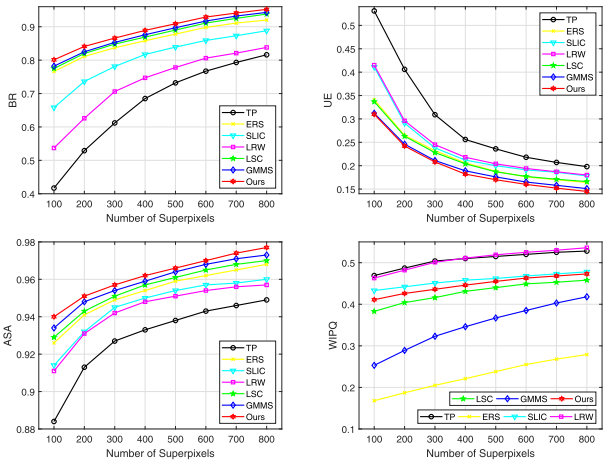

FIGURE 12. Comparison of superpixel segmentation performances.

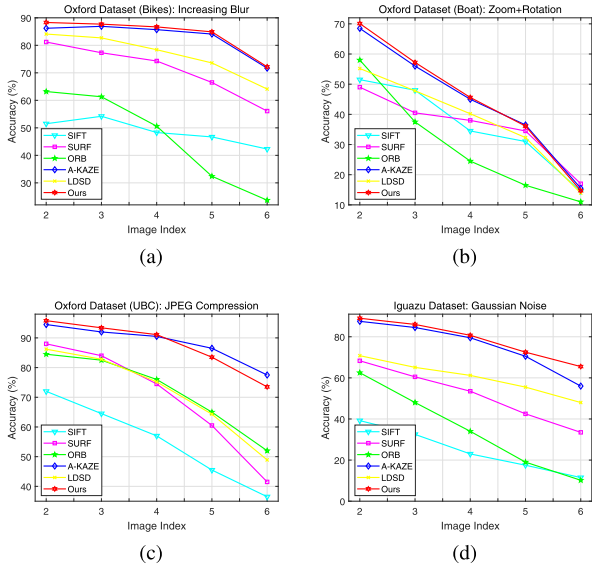

FIGURE 13. Superpixel detector response to various attacks.

E. KEY-SUPERPIXEL DETECTOR RESPONSE FOR VARIOUS ATTACKS

In this experiment, we conduct the experiments to assess the robustness of our superpixel detector for a variety of attacks, including blur, zoom-rotation, JPEG compression and Gaussian noise. Test images are selected from public available Oxford [29] and Iguazu [28] datasets. We compare our

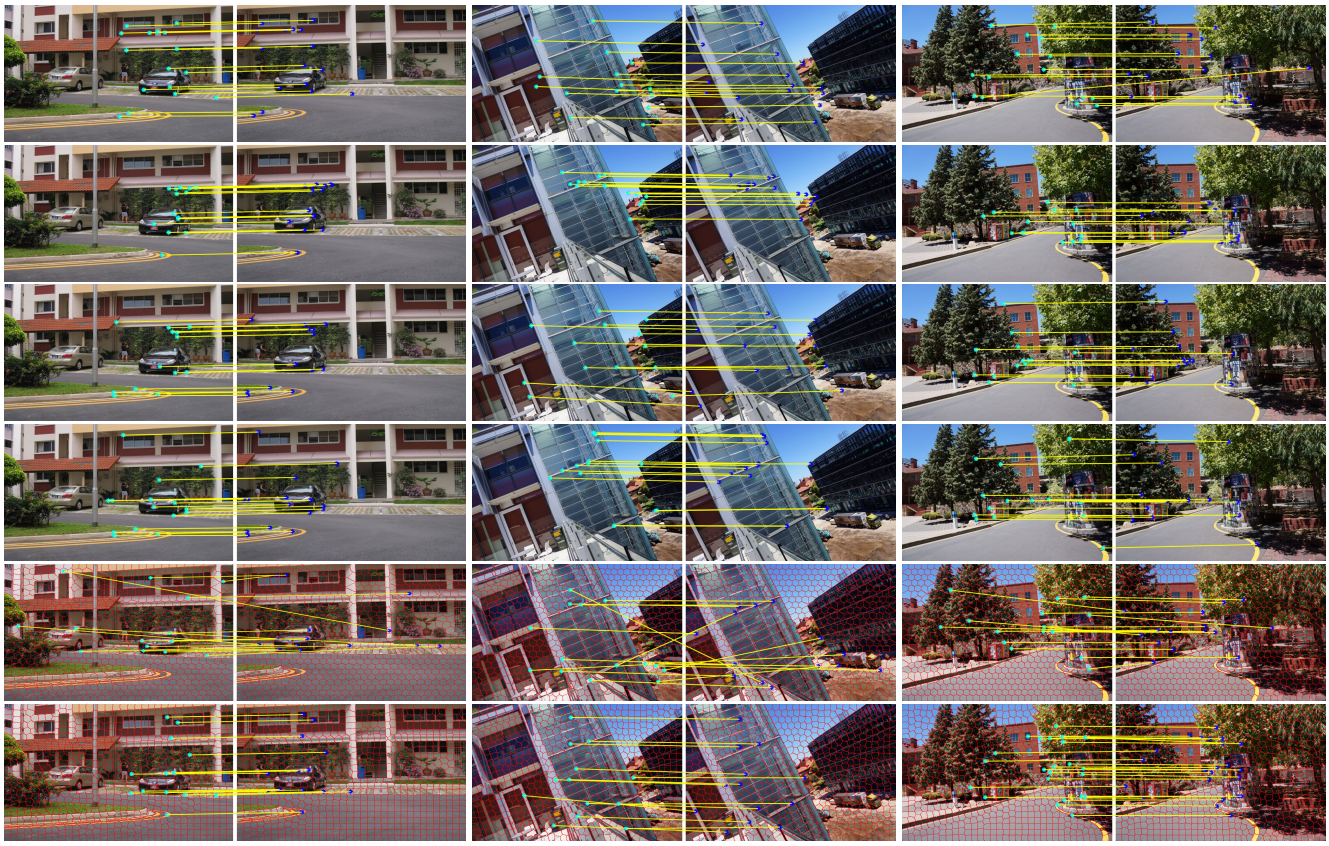

FIGURE 14. Visual comparison of matched images, from top to down is SIFT, SURF, ORB, A-KAZE, LDSD and Ours, respectively.

algorithm with SIFT [8], SURF [7], ORB [6], A-KAZE [29] and LDSD [30], respectively. The purpose of this experiment is to demonstrate that our method can be applied to different superpixel images and conserves good characteristics under various attacks. The gather of these attacks and their detection accuracies is depicted in Fig. 13. Generally, since blur tends to cause uncomfortable viewing experience, such an attack may bring down the accuracy of a detector. To test the robustness of our approach to blur attacks, the first experiment is carried out on six test images on increasing blur subset of Oxford dataset. As shown in Fig. 13 (a), our method has a relatively good result, second only to A-KAZE with a tiny difference. Besides, image rotation is another common form of geometric attacks. The proposed experiment considers that images are simultaneously attacked by rotation and scale. The proposed experiment calculates detection accuracy on each attacked images. As shown in Fig. 13 (b), we found that key-superpixels are well detected by A-KAZE and our method. In JPEG test, it could be observed that while JPEG compression ratio increases, our detector achieves moderately lower accuracy. This phenomenon shows that the proposed superpixel detector is slightly sensitive to JPEG compression. To evaluate the robustness of our detector with regard to Gaussian noise, we compare the accuracies of the six images. Fig. 13 (d) displays that our proposed detector outperforms other approaches on all six test images of Iguazu dataset.

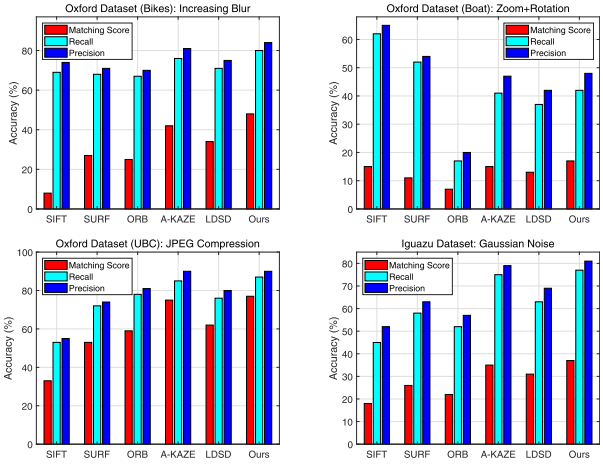

FIGURE 15. Matching results of various approaches under all attacks.

F. SUPERPIXEL MATCHING UNDER VARIOUS ATTACKS

In order to assess the performance of our proposed superpixel descriptor, we compare the image matching results of the other five well-known algorithms with our method on test images in [44]. In this test, we specified the superpixel number K as 800, and parameter $\alpha = 0.2$. In Fig. 14, compared with these detectors, the proposed approach could achieve relatively higher quality even when there are less feature

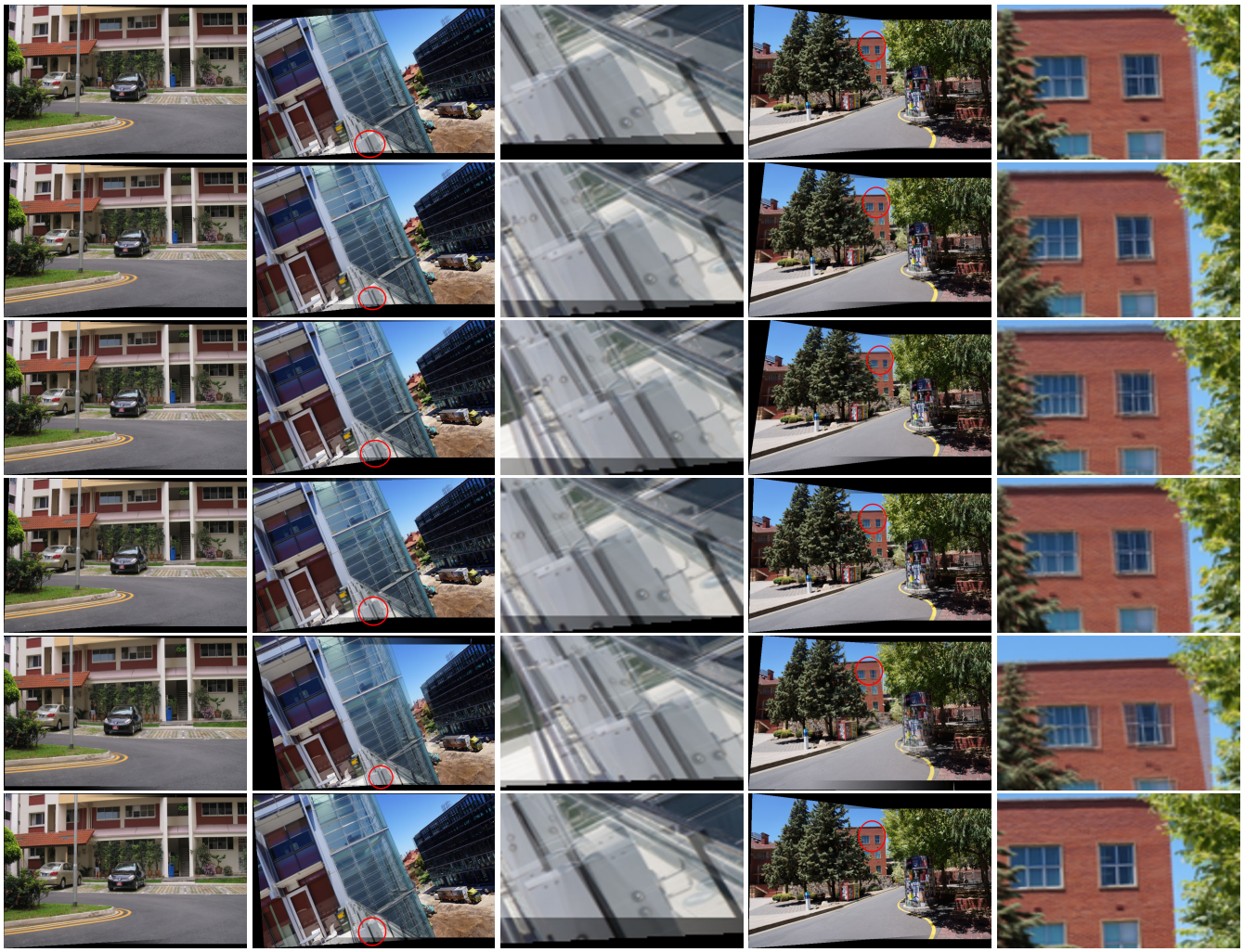

FIGURE 16. Visual comparison of stitched images, from top to down is SIFT, SURF, ORB, A-KAZE, LDSD and Ours, respectively.

points (centroid of key-superpixels). This would help save matching time.

For quantitative analysis, MS, *recall* and *precision* are utilized to assess the performance on Oxford and Iguazu datasets. The results for the selected algorithms can be observed in Fig. 15. In Zoom + Rotation test, SIFT tends to be better performed in *recall* and *precision*, while similar results with SURF and A-KAZE are obtained in Matching Score. For JPEG test, A-KAZE and our algorithm achieve relatively higher *precision* and *recall*. In Fig. 15, we also provide the matching accuracy results under scale and rotation transforms, our method achieves similar results as A-KAZE does. These results demonstrate that the superpixel pyramid provides scale invariance for detected key-superpixels, the change of image scale actually results in some losses of image information, and would affect the performance of our method to some extent. In Gaussian noise test, our method sacrifices Matching Score to improve the overall performance. The main advantage that have this method outperformed others in terms of *recall* and *precision* is that the proposed superpixel pyramid engaged in key-superpixel detecting considers not only the information of neighborhood superpixels, but also their intensities. One can observe from Fig. 15, the LDSD is still more accurate than ORB, SURF and SIFT in Gaussian noise case.

To further investigate, Fig. 16 shows the alignment results respectively on test images in [44]. In this test, in order to fairly compare the effectiveness of each feature descriptor, we apply the same stitching algorithm described in [44] to stitch full panoramas. It is clear that all algorithms can find accurate keypoints, whereas the stitched images generated by our algorithm has better and distinct local details in most of the image pairs (indicated by red ellipse in Fig. 16).

VI. CONCLUSION

In this paper, a novel approach of superpixel-based matching was provided for image stitching. The main conclusions of the study could be summarized as follows: (i) a spatially constrained Student's-t mixture model was used to drive superpixel segmentation, which was an innovative method

and should be considered as main contribution of this paper. We had demonstrated that our new superpixel segmentation model with MRF-based spatially constrained term was superior to other algorithms on regularity and accuracy; (ii) this work addressed superpixel Gaussian filter, superpixel detectors, and superpixel scale-space pyramid that no previous study has drawn attention to so far and many new directions for future research. Utilizing superpixel matching to replace individual keypoint matching, this is the key point to speed up our approach; (iii) fusing LBP and Zernike moment invariants would help obtain oriented gradient and texture features at once; (iv) the designed superpixel descriptor explored the content and shape of each generated key-superpixel to find the matched superpixels between image pairs. The experiments on Oxford and Iguazu datasets demonstrated the performance of this approach, and the relatively better results compared to other existing feature matching approaches were shown.

REFERENCES

[1] A. Radman and S. A. Suandi, "A superpixel-wise approach for face sketch synthesis," *IEEE Access*, vol. 7, pp. 108838–108849, 2019.
[2] Y. Li, H. Pan, X. Wu, Y. Liu, S. Pang, and L. Zhu, "Fast two-cycle level set tracking with interactive superpixel segmentation and its application in image retrieval," *IEEE Access*, vol. 7, pp. 159930–159942, 2019.
[3] W. Fang, T. Zhang, C. Zhao, D. B. Soomro, R. Taj, and H. Hu, "Background subtraction based on random superpixels under multiple scales for video analytics," *IEEE Access*, vol. 6, pp. 33376–33386, 2018.
[4] L. Liu, Q. Wang, W. Zhu, H. Mo, T. Wang, S. Yin, Y. Shi, and S. Wei, "A face alignment accelerator based on optimized coarse-to-fine shape searching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2467–2481, Aug. 2019.
[5] Y. Wang, C. Peng, and Y. Liu, "Mask-pose cascaded CNN for 2D hand pose estimation from single color image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3258–3268, Nov. 2019.
[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[9] X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren, "Automatic image segmentation with superpixels and image-level labels," *IEEE Access*, vol. 7, pp. 10999–11009, 2019.
[10] L. Cong, S. Ding, L. Wang, A. Zhang, and W. Jia, "Image segmentation algorithm based on superpixel clustering," *IET Image Process.*, vol. 12, no. 11, pp. 2030–2035, Nov. 2018.
[11] Y. Lei, X. Liu, J. Shi, C. Lei, and J. Wang, "Multiscale superpixel segmentation with deep features for change detection," *IEEE Access*, vol. 7, pp. 36600–36616, 2019.
[12] M. Ren, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
[13] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2097–2104.
[14] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.
[15] Y. Zhou, X. Pan, W. Wang, Y. Yin, and C. Zhang, "Superpixels by bilateral geodesic distance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2281–2293, Nov. 2017.
[16] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.
[17] V. Machairas, M. Faessel, D. Cardenas-Pena, T. Chabardes, T. Walter, and E. Decenciere, "Waterpixels," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3707–3716, Nov. 2015.
[18] Y. Zhang, X. Li, X. Gao, and C. Zhang, "A simple algorithm of superpixel segmentation with boundary constraint," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1502–1514, Jul. 2017.
[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[20] Y.-J. Liu, M. Yu, B.-J. Li, and Y. He, "Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 653–666, Mar. 2018.
[21] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012.
[22] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3317–3330, Jul. 2017.
[23] X. Xiao, Y. Zhou, and Y.-J. Gong, "Content-adaptive superpixel segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2883–2896, Jun. 2018.
[24] Z. Ban, J. Liu, and L. Cao, "Superpixel segmentation using Gaussian mixture model," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4105–4117, Aug. 2018.
[25] C.-R. Huang, W.-C. Wang, W.-A. Wang, S.-Y. Lin, and Y.-Y. Lin, "USEAQ: Ultra-fast superpixel extraction via adaptive sampling from quantized regions," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4916–4931, Oct. 2018.
[26] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.
[27] M. A. Duval-Poo, N. Noceti, F. Odone, and E. De Vito, "Scale invariant and noise robust interest points with shearlets," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2853–2867, Jun. 2017.
[28] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 214–227.
[29] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–11.
[30] S. Du and T. Ikenaga, "Low-dimensional superpixel descriptor and its application in visual correspondence estimation," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 19457–19472, Jul. 2019.
[31] H. Yang, C. Huang, F. Wang, K. Song, and Z. Yin, "Robust semantic template matching using a superpixel region binary descriptor," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3061–3074, Jun. 2019.
[32] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, Nov. 2018, pp. 1–13.
[33] D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 337–349.
[34] A. R. Widya, A. Torii, and M. Okutomi, "Structure from motion using dense CNN features with keypoint relocalization," *IPSJ Trans. Comput. Vis. Appl.*, vol. 10, no. 1, pp. 1–7, Oct. 2018.
[35] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4068–4076.
[36] Z. Ren and Y. J. Lee, "Cross-domain self-supervised multi-task feature learning using synthetic imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 762–771.
[37] T. M. Nguyen and Q. M. J. Wu, "Fast and robust spatially constrained Gaussian mixture model for image segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 621–635, Apr. 2013.
[38] M. Kuczma and A. Gilanyi, *An Introduction to Theory Functional Equations and Inequalities: Cauchy's Equation and Jensen's Inequality*. Cambridge, MA, USA: Birkhaüser, 2009.
[39] C. Li, D. Lin, B. Feng, J. Lu, and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.
[40] S. Shoham, "Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions," *Pattern Recognit.*, vol. 35, no. 5, pp. 1127–1142, May 2002.

[41] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.
[42] S. Aly and A. Sayed, "Human action recognition using bag of global and local Zernike moment features," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24923–24953, Sep. 2019.
[43] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[44] J. Zaragoza, T. Chin, Q. Tran, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1285–1298, Jul. 2014.
[45] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

PENGYU WANG received the B.S. degree in automation from Hebei University, Baoding, in 2015, and the M.S. degree in control engineering from Hebei University, in 2018. He is currently pursuing the Ph.D. degree with the East China University of Science and Technology, Shanghai, China. His research interests include image processing, deep learning, computer vision, and pattern recognition.

HONGQING ZHU (Member, IEEE) received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2000. From 2003 to 2005, she was a Postdoctoral Fellow at the Department of Biology and Medical Engineering, Southeast University, Nanjing, China. She is currently a Professor with the East China University of Science and Technology, Shanghai. Her current research interests include medical image processing, deep learning, computer vision, and pattern recognition. She is a member of the IEICE.

XIAOFENG LING (Member, IEEE) received the B.S. and Ph.D. degrees from Shanghai Jiao Tong University, China, in 2006 and 2012, respectively. From 2013 to 2015, he has served as the Director of research and development in a start-up company, and was committed to the research and development of 5G wireless communication technology. He is currently a Lecturer with the School of Information Science and Engineering, East China University of Science and Technology. His research interests include medical image processing, deep learning, computer vision, and signal processing.