Content-Based Superpixel Segmentation and Matching Using Its Region Feature Descriptors

Jianmei ZHANG†, Pengyu WANG†, Feiyang GONG†, Hongqing ZHU†(a), Nonmembers, and Ning CHEN†, Member

SUMMARY Finding the correspondence between two images of the same object or scene is an active research field in computer vision. This paper develops a rapid and effective Content-based Superpixel Image matching and Stitching (CSIS) scheme, which utilizes the content of superpixel through multi-features fusion technique. Unlike popular keypoint-based matching method, our approach proposes a superpixel internal feature-based scheme to implement image matching. In the beginning, we make use of a novel superpixel generation algorithm based on content-based feature representation, named Content-based Superpixel Segmentation (CSS) algorithm. Superpixels are generated in terms of a new distance metric using color, spatial, and gradient feature information. It is developed to balance the compactness and the boundary adherence of resulted superpixels. Then, we calculate the entropy of each superpixel for separating some superpixels with significant characteristics. Next, for each selected superpixel, its multi-features descriptor is generated by extracting and fusing local features of the selected superpixel itself. Finally, we compare the matching features of candidate superpixels and their own neighborhoods to estimate the correspondence between two images. We evaluated superpixel matching and image stitching on complex and deformable surfaces using our superpixel region descriptors, and the results show that new method is effective in matching accuracy and execution speed.

key words: image matching, superpixel region descriptor, multi-features fusion, superpixel, superpixel segmentation

1. Introduction

Image matching is an active research topic in computer vision and has been widely used in security monitoring, scene recognition, constructing panoramas, and medical imaging, etc. Generally, there has been mainly three categories of image matching, based on geometric transform, gray correlation, and invariant features.

In recent years, scholars have done a lot of research on image matching and stitching by exploring the validity of invariant features [1]. The advantage of using invariant features is that multi-features fusion can be achieved [2]. This process includes detecting feature points from each image, searching images according to the detected feature points, and matching the images via them. Finally, user takes these matching points to seamlessly blend images. Therefore, two important procedures, feature extraction and seamless blend, ensure the quality of final results. Common feature extraction algorithm is scale invariant feature transform (SIFT) [3]. SIFT descriptor is invariant to image rotation, scaling, and illumination. As an accelerated version of SIFT, speeded up robust feature algorithm (SURF) [4] utilized integral image and box filter to simplify the second-order partial derivative of Gaussian function. Inspired by SRUF, Rublee et al. [5] developed the oriented fast and rotated BRIEF (ORB) descriptor for speed-up feature extraction. Leutenegger et al. [6] introduced a binary descriptor invariant to scale and rotation, named BRISK. It exhibits better results with respect to distinctiveness. Both BRISK and ORB features are much faster computed than SURF and SIFT. As is well-known, if color content is ignored, many objects may be mismatched. Therefore, Huang et al. [7] reported AR-SURF, which is an adaptive registration method for extracting color features based on SURF descriptor. Similarly, Li et al. [8] combined color invariant and ORB descriptor. Their method was more stable with respect to variations in photometric imaging conditions. Recently, Li et al. [9] introduced dual-feature warping based stitching method to preserve geometrical information of scenes. This method strongly relies on the accuracy of feature matching, and stitching susceptible to noise. Because feature-based matching is accomplished by feature descriptors, these descriptors are required to resist the changes caused by noise in images. In order to avoid mismatch caused by noise, Zhao et al. [10] formulated the image matching task as a Markov Random Field (MRF) energy functional optimization problem. Their scheme leaded to a fast and accurate matching, and resisted the influence of noise as well. In [11], Zhang et al. formulated the problem of dense matching as a Bayes decision task via MRF. This method took both texture feature similarity and spatial consistency into considered to eliminate the effect of noise. The above methods are based on invariant feature matching. It is the main direction of the image matching recently. As the feature point scheme shows a good robustness and reliability on the illumination changes, rotation and other geometric changes.

Compare with the pixel grid pattern, superpixel offers consistent information of support areas. This allows similar scene to retain superpixels that are similar in appearance. Recently, superpixel segmentation algorithms have been research intensively [12]–[14] etc. Each method has its own characteristics and may be suitable for specific applications. The most famous and widely used methods are SLIC [15] and LSC [16]. The former adheres to the strong
gradient boundaries due to the simple Euclidean distance metric. In LSC, each pixel is mapped to a weighted point in a ten dimensional feature space, then K-means clustering in this space is applied for partition. Machairas et al. [17] proposed waterpixels as a strategy for generating superpixels which relies on the marker controlled watershed transformation. Xu et al. [18] introduced a geodesic distance-based superpixel segmentation model, of which the basic idea is to measure the distance among pixels along minimum spanning tree. Giradu et al. [19] presented a robust superpixels using color and contour features along linear path between the pixel and the corresponding superpixel barycenter.

Recently, some attention has been paid to match two images in superpixel manner. For example, a low-dimensional superpixel descriptor for video correspondence estimation was presented by Du et al. [20]. They extracted shape, texture, and color features from superpixel. Xie et al. [21] reported a superpixel correspondence approach based on the graph matching technique. This method has the ability to get the consistent intermediate-level semantic information in a pair of images. Dong et al. [22] developed a hierarchical superpixel-to-pixel image matching algorithm. They estimated superpixel pairing between two images to drive the matching in pixel level. To our knowledge, few works consider image matching via features inside superpixels in coupled images. There are two main reasons: (i) due to the obvious geometric distortion existing in a pair of images, even for the same scene and target, their superpixels have completely different sizes and shapes; (ii) low-level features extracted from the corresponding superpixels may be very different due to geometric deformations.

Different from the popular grid scheme, here we develop an effective and rapid image matching method using superpixel-to-superpixel scheme, called Content-based Superpixel Image matching and Stitching (CSIS). The goal of our system is to match the images by considering the similarity of the superpixel contents associated with visual features such as color, texture and gradient. The proposed method takes advantages of superpixel because extracting features from regions are more convenient than extracting features from pixels. In our CSIS, we first introduce a scheme for applying a new distance metric to superpixel generation which is suitable for superpixel matching purpose. The proposed Content-based Superpixel Segmentation (CSS) method explores the contents of the image including color and spatial features, and divides the pixels into superpixels along the directions of fast gradient change. Then, input image pair is partitioned into many visually compact and homogeneous superpixels using our superpixel segmentation algorithm. Next, for reliable matches, we calculate the entropy of each superpixel to pick out those with distinctive characteristics. Given an input target image and a reference image, for superpixels selected from the previous step, we compute the low-level features of these superpixels. Only features from the selected superpixels are merged for image match purpose. In this way, the proposed CSIS can explore the region-based similarity rather than similarity based on keypoints in coupled images. Selecting reliable matched superpixel pairs, this study adopts a matching from rough to fine by means of comparing the features of candidate superpixels and their own neighborhoods to estimate the correspondence between two images. We evaluated the proposed CSIS on published datasets, and the results demonstrate that our framework is faster and more effective than some advanced methods.

The contributions of this study include the following:

- We develop a superpixel segmentation algorithm CSS based on a new distance metric. The method divides the pixel into superpixel along the directions of fast gradient change, and guarantees superpixel with better boundary adherence and regularity.
- The paper explores the contents of superpixels, and proposes a superpixel internal feature-based scheme CSIS to implement superpixel-to-superpixel matching.
- While for matching or stitching purpose, superpixel is utilized to replace the pixel grid which will help to decrease computational cost.

The remainder of this paper is organized as follows. Section 2 introduces the datasets. Section 3 describes a novel superpixel segmentation algorithm in detail. The proposed image matching algorithm based on superpixel region descriptors is introduced in the same section. The experimental results and analyses are reported in Sect. 4. Section 5 summarizes the whole paper.

2. Materials

This paper evaluates our algorithm with three datasets shown in Fig. 1.

**Dataset I:** Images in this dataset contained flower image pair (625×625 pixels) and river image pair (781×781 pixels). These test images are selected for the purpose of explaining our algorithm in detail. They are available on web page1.

**Dataset II:** This paper evaluates the performance of superpixel segmentation approaches on Dataset II. It is a subset of Berkeley segmentation dataset (BSDS500) [23] with a resolution of 481×321, or 321×481 pixels. BSDS500 contains 500 natural images along with ground truth provided by different individuals.

**Dataset III:** This article conducts superpixel matching and image stitching on two groups of images used by Zaragoza et al. [24]. Each image of car image pair is approximately 3264×2448 pixels, and for mountain image pair, each image is approximately 506×506 pixels.

3. Methodology

The flowchart of the proposed framework is shown in Fig. 2. This framework consists of five parts: (1) content-based superpixel segmentation; (2) key superpixels detection; (3) superpixel descriptors generation; (4) superpixel-to-superpixel

---

1https://www.mianfeiwendang.com/doc/b1abc77494beddeead82638
matching; (5) image stitching. We start by proposed superpixel segmentation algorithm.

### 3.1 Content-Based Superpixel Segmentation

Superpixel aims to over-segment an image into small compact areas with homogenous appearance. Superpixel extraction actually can be considered as a problem of pixel labeling. A good superpixel generation scheme should provide users with relatively accurate, regular, compact and low complexity superpixels. Recently, many existing superpixel algorithms focus on color and spatial features to guarantee superpixels with uniform perception and compact shape. For some areas with poor color contrast, it is difficult to ensure that the superpixels are well aligned to image boundaries. To be sure, the more effective of superpixel extraction is, the better the superpixel region descriptors is. Therefore, in this study, a content-based superpixel generation algorithm is proposed for superpixel extraction. Our superpixel algorithm considers gradient feature, and combines it with color and spatial features to improve the ability of superpixel to adhere the image boundaries. The proposed CSS consists of three stages, parameter initialization, label assignment, and superpixel segmentation and refinement.

**Initialization:** Let \( i \) represent the pixel index of image \( I \) of \( h \) in height and \( w \) in width. To make the segmented superpixels consistent more with human vision, the proposed approach performs on the CLELAB color space \((l_i, a_i, b_i)\). Here, let six dimensional vector \( z_i = (l_i, a_i, b_i, x_i, y_i, g_i)^T \) describe pixel \( i \), where \((x_i, y_i)\) denotes the position of pixel \( i \), \( i = (1, 2, \ldots, N) \), and \( g_i \) represents the gradient of pixel \( i \). Supposed that \( n_x \) and \( n_y \) are nodes with horizontal and vertical directions, the number of superpixel \( K \) can be expressed as
two vectors. A new kernel distance for calculating the similarity between the category to which each pixel belongs. This process is tical and horizontal directions. The application of gradient gi, where vx and vy represent the length and width of superpixels respectively. Set vx = vy to produce roughly equally sized superpixels. Our method begins by placing the initial superpixel seeds over the image plane. Here, the initial location of each superpixel seed is the center of the regular hexagon. The method will implement segmentation on hexagonal grid, since they are more isotropic than squares.

Label assignment: The goal of this stage is to update the category to which each pixel belongs. This process is accomplished by distance measurement. Here, we propose a new kernel distance for calculating the similarity between two vectors.

Let z_{i,c} = (l_i, a_i, b_i), z_{i,s} = (x_i, y_i), and z_{i,g} = g_i denote the color, spatial, and gradient components of the i-th pixel, respectively. Similarly, μ_{j,c} = (l_j, a_j, b_j), μ_{j,s} = (x_j, y_j), and μ_{j,g} = g_j are color, spatial, and gradient components of the mean (seed) with respect to the j-th label. Given a pixel vector z_i = (l_i, a_i, b_i, x_i, y_i, g_i)^T and the mean vector μ_j = (l_j, a_j, b_j, x_j, y_j, g_j)^T, the difference of color feature vector z_{i,c} and μ_{j,c} is defined as

\[ K_l(i, j) = \exp\left(\frac{|l_i - l_j|^2}{\sigma R_l^2}\right) \cdot K_a(i, j) = \exp\left(\frac{|a_i - a_j|^2}{\sigma R_a^2}\right), \]

\[ K_b(i, j) = \exp\left(\frac{|b_i - b_j|^2}{\sigma R_b^2}\right), \]  

(2)

where \( R_l = l_{\max} - l_{\min} + 1, \) \( R_a = a_{\max} - a_{\min} + 1, \) and \( R_b = b_{\max} - b_{\min} + 1. \) The \( l_{\max} - l_{\min} \) denotes the maximum value of the color component minus its minimum value. The color feature distance is computed in the CIELAB color space, which is defined as follows

\[ d_c(i, j) = \sqrt{K_l(i, j) + K_a(i, j) + K_b(i, j)}. \]

(3)

The difference of spatial feature vector is calculated using

\[ d_s(i, j) = \sqrt{\exp\left(\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{(\sigma R_s)^2}\right)} \]  

with \( R_s = \sqrt{w \times h / K}. \)

(4)

The kernel distance in gradient feature space is written as follows

\[ d_g(i, j) = \sqrt{\exp\left(\frac{(g_i - g_j)^2}{\sigma R_g^2}\right)} \]  

with \( R_g = g_{\max} - g_{\min} + 1, \)

(5)

\[ V_g \] and its magnitude is calculated by

\[ g_i = ||V_g|| = \sqrt{g_{i,v}^2 + g_{i,h}^2}, \]

(6)

where \( g_{i,v} \) and \( g_{i,h} \) represent changes in intensity in both vertical and horizontal directions. The application of gradient feature makes the model divide pixels into superpixels according to the direction of fast gradient change. Combining the specific feature distances mentioned above, the final distance \( D \) between two feature vectors \( z_i = (l_i, a_i, b_i, x_i, y_i, g_i)^T \) and \( \mu_j = (l_j, a_j, b_j, x_j, y_j, g_j)^T \) is as follows

\[ D(i, j) = \omega_c d_c(i, j) + \omega_s d_s(i, j) + \omega_g d_g(i, j) \]  

(7)

where \( \omega_c, \omega_s, \) and \( \omega_g \) are weight coefficients which adjust the proportion of the color, spatial, and gradient distance terms.

Superpixel segmentation and refinement: In this stage, we compute the distance metric between pixel \( i \) and its surrounding superpixel seeds, and label this pixel into the seed that has the smallest distance metric. Then, pixels with the same label form a superpixel. We calculate the location of the new seed via the average pixel feature vectors of all image pixels in the same cluster. The labeling strategy is executed iteratively until generated superpixel converges.

A good superpixel technology has to be capable of adhering well to image boundaries, follows the contour of real object in the scene as well. To have our approach explicitly force the connection along boundaries, some considerations should factor into our superpixel extraction method:

(i) Superpixel, which size is smaller than \( v_x \) pixels, should be merged into other adjacent superpixels in terms of color feature;

(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 to smooth boundary.

3.2 Key Superpixel Detection

Many superpixels might fail to match correctly while the features inside them are not obvious. Discarding these superpixels significantly helps to improve matching precision and to reduce the computation cost. We do this by local entropy. Local entropy is a statistical measure of the uncertainty associated with a random variable that provides a natural way of finding disorder contained in an image. Therefore, entropy can measure the image intensity distribution and texture feature. By computing the local entropy of each superpixel, some superpixels that do not reach the threshold values will be removed. Those superpixels that failed to meet the threshold requirements are indicated to have less prominent internal features and rich details, including color, texture and scene. Therefore, by setting the threshold value, the superpixels with rich details would be selected to participate in the following description and matching, which reduces execution time for our method.

Let \( h(h \in H) \) represents the pixel index of any one of the superpixel. This paper defines the local entropy in a superpixel domain as follows.

\[ E = -\sum_{h=1}^{H} p_h \log^p_h, \]  

(8)

where \( p_h \) is the probability of the \( h \)-th pixel \( f_h \) in a given
superpixel $s (s \in S)$. It is defined by

$$p_h = \frac{f_h}{\sum_{h=1}^{h=1} f_h}. \quad (9)$$

In this research, input images are RGB (red, green, and blue) color space. Thus, for the $s$-th superpixel, the below mentioned equation has been used for computing the local entropy.

$$E_s = E_s^R + E_s^G + E_s^B. \quad (10)$$

Finally, part of the superpixels with smaller entropy value ($E_s < \text{threshold}$) are abandoned, since they are uniformity on intensity, texture, and color. Figure 3 gives some examples about superpixels that are still remained after this procedure. As shown in Fig. 3, superpixels remain after local entropy contain rich contents, indicated by blue stars.

### 3.3 Superpixel Descriptors

Unlike those traditional pixel-based feature extraction schemes, the highlight of our approach is feature description at interest superpixels. Here, three local features of each candidate superpixel, gradient feature, texture feature, and color feature are aggregated in a feature vector. The resulting feature vectors are applied to superpixel-to-superpixel matching.

**Linear binary pattern (LBP):** Texture is a prominent feature of an image that can be discerned in a form of small repeated patterns. LBP operator was designed for extraction texture feature found by Ojala et al. [25]. LBP operator is tolerance against illumination changes. Current study labels each pixel of a superpixel through thresholding the 3-by-3 neighborhood of this pixel with the center pixel value. If the value of those surrounded is bigger than the central one, set the result to 1, otherwise set to 0. After performing LBP operation for every superpixel, the 256 dimensional feature histogram, encoded as a binary number, could be used to describe the texture inside the superpixel.

**Histogram of oriented gradients (HOG):** HOG [26]

is an efficient descriptor used to extract the direction for an object. It measures the directional intensity changes of a superpixel. HOG hence provides discriminative information for matching these two superpixels. HOG feature is less sensitive to illumination variations. For a superpixel, the first step is detecting the gradient values of each pixel via 1-D centered derivative in horizontal and vertical directions. Then, the direction for every pixel is binned into uniformly quantized direction channels spreading from 0° to 360°. Each interval of direction is 20°. Finally, HOG feature is extracted via computing the histogram of gradients. With this method, the HOG feature is obtained with 18-dimensionality.

**Colour histogram (ColourHist):** Color is an important feature of color image, neglecting it may lead to poor illumination robustness and mismatching. ColourHist [27], developed based on human visual perception mechanism, is widely used in computer vision. In the process of color feature extraction, every input superpixel is firstly converted from RGB (red, green, blue) color space to HSV (hue, saturation, and value). The main reason is that hue and saturation components of HSV space are insensitive to the illumination variations. Our method employs a color histogram that is relatively invariant to the direction and rotation of superpixels. For every superpixel, color feature is extracted through considering the color similarity information by spreading each pixel’s total membership value to all the histogram bins. In this way, we compute a 64-bin histogram of each HSV channel and list them together, leading to a 192-dimensional ColourHist.

After that, 256-dimensional texture feature, 18-dimensional orientation feature, and 192-dimensional color feature are normalized between 0 and 1, and 466 dimensional feature vectors are aggregated. It is regard as each superpixel descriptor. Detail of the feature merging process can be illustrated in Fig. 4 (a).

### 3.4 Superpixel Matching

To search for the superpixel pairs between each pair of images, we should explore the similarity among the candidate superpixels. Here, every candidate superpixel is represented by $466 \times 1$ feature vector including LBP, HOG, and ColourHist features. It is worth noting that not all superpixels can be matched, and it is possible for one superpixel to have multiple matched superpixels in another image. This paper utilizes merging scheme to achieve high precision matching. The whole matching process includes two steps: initial matching and neighborhood matching.

**Initial matching:** This step aims at performing superpixel matching by finding the most similar of all superpixels in another image. Here, we apply the maximum correlation coefficient ($cc$) between two superpixels in an image pair to perform superpixel initial matching.
ZHANG et al.: CONTENT-BASED SUPERPIXEL SEGMENTATION AND MATCHING USING ITS REGION FEATURE DESCRIPTORS

Fig. 4 Descriptors, (a) descriptors of single superpixel; (b) merging descriptors of adjacent superpixels.

Fig. 5 Superpixel matching, up: initial matching; down: neighborhood matching.

\[ cc = \frac{\sum_{d=1}^{D} (X_d - \bar{X})(Y_d - \bar{Y})}{\sqrt{\sum_{d=1}^{D} (X_d - \bar{X})^2} \sqrt{\sum_{d=1}^{D} (Y_d - \bar{Y})^2}}, \quad (11) \]

where \( D \) represents the dimension of the feature vector. \( X \) and \( Y \) are feature vectors of source superpixel and target superpixel, and \( \bar{X} \) and \( \bar{Y} \) are mean values with respect to feature vectors written by

\[ \bar{X} = \frac{1}{D} \sum_{d=1}^{D} X_d, \quad \bar{Y} = \frac{1}{D} \sum_{d=1}^{D} Y_d. \quad (12) \]

Here, \( cc \) indicates the similarity level of the matched superpixel pair. The matching process of superpixel is similar to that of keypoints. If two superpixels have higher similarity, their \( cc \) is more close to 1. The number of candidate superpixels in reference image is set to be less than 5 in our method. More specifically, for a superpixel in input target image selected from the above step, we tried to find no more than five superpixels in the reference image that might best match this superpixel (see Fig. 5). The aim of initial matching is to reduce the number of candidate superpixels of reference image. In this way, the time consumption of the next step is cut down when superpixel neighborhood matching is performed.

**Neighborhood matching:** This step focuses on neighborhood matching of query superpixel. After obtaining candidate superpixels in the initial matching step, the neighborhood of query superpixel has been employed to further improve the accuracy in matching processes. As well known, the essence of superpixel segmentation is partitioning an image into a number of connected and unified pixel groups with perceptual significance, and the pixels in a given superpixel will have similar color or grayscale. Thus, for a key superpixel in the current image, many superpixels with the similar color and scene can be matched in the target image. Therefore, matching the neighborhood of superpixel can effectively eliminate the matching error caused by single superpixel matching, improve the matching accuracy and reduce the matching time. This is because of the neighborhood superpixels of a pair of superpixels are highly similar. Based on these considerations, this paper incorporates the adjacent superpixels into the matching algorithm. Figure 5 shows an example for finding the best neighborhood in the reference image that matches the query superpixel’s neighborhood. Superpixel neighborhood descriptors are calculated through merging the features of all adjacent superpixels as follows:

\[ X_d = \frac{n}{\tau \sum_{d=1}^{D}} X_{\tau}^{(\tau)}, \quad (13) \]

where \( n \) is number of superpixel neighborhoods, and \( \tau \) represents the index of neighborhood superpixels. As shown in the above Eq. (13), superpixel neighborhood descriptor is a vector with the size of \( D \times 1 \), and the value of each element is between 0 and 1. Figure 4 (b) shows the detailed representation of neighborhood descriptors calculation for a query superpixel. Similarly, the similarity measurement of superpixels’ neighborhood pair between the target image and the reference image is implemented using (11) again. The maximum \( cc \) indicates higher similarity for a pair of matched neighborhood. This means that their respective superpixels are also the best matching. This research discusses the case that the centroid is enclosed in the superpixel. Depend upon the count of candidate superpixel pairs, we calculate the centroid of each superpixel, and match them. These centroids are named keypoints in our model. Exemplary matches for searched and target image pair are displayed in Fig. 6. It can be seen that the proposed method worked a highly precise matching.

Finally, create a new image big enough to hold the panorama and composite the two images into it. Panoramic image stitching [28] is implemented using matched centroids. The results generated by our CSIS are provided in
Fig. 6 Matching results using proposed framework.

Fig. 7 Stitching results using proposed method.

Fig. 7. We found from this figure, the proposed approach achieves a better stitching performance.

4. Experimental Results and Discussion

4.1 Evaluation Metrics and Experiment Environment

The evaluation of experimental results is conducted via standard metrics, including Boundary Recall (BR), Undersegmentation Error (UE), Precision and Recall. They have been widely used in most recent superpixel literatures.

BR measures the percentage of superpixel boundaries coinciding with ground truth boundaries, written in following form:

\[
BR = \frac{qp}{bp},
\]

where \(qp\) is the number of boundary pixels in segmentation results that meet the condition of at least one pixel in the 3 \(\times\) 3 neighborhood should be the boundary pixel of ground truth. \(bp\) is the total number of the boundary pixels of segmentation results. A high BR rate means that real boundaries are rarely missed.

UE calculates the proportion of over-segmentation superpixels, while UE value is close to zero, superpixels approaches to the ground truth. UE is defined as

\[
UE = (-1) + \frac{1}{N} \sum_{|s_j \cap s_g| > \omega |s_j|} |s_j|,
\]

where \(s_j\) and \(s_g\) are the pixel sets of superpixel \(j\) and ground truth \(g\), respectively. Parameter \(\omega\) is set to 0.05. The lower the UE, the fewer superpixels across multiple objects.

In addition to above metrics for the segmentation accuracy, we also evaluate the accuracy of keypoints matching by Precision and Recall. The number of correct matches out of total matches is represented by Precision.

\[
Precision = \frac{\text{# number of correct matches}}{\text{# number of matches}}.
\]

The Recall is computed as a ratio where the number of corrected matches divided by total number of correspondences (possible correct matches), defined as

\[
Recall = \frac{\text{#correct matches}}{\text{#correspondences}}.
\]

The equipment used is Intel (R) Core (TM) i7-7700k@4.20GHz CPU, 32G memory. The experimental environment is Windows 10 and all algorithms are carried out in C++ and Matlab R2018a.

4.2 Experiment for Superpixel Segmentation

To demonstrate the superiority of the CSS, we compare it with five advanced superpixel segmentation methods, including SLIC\(^\dagger\) [15], LSC\(^\dagger\dagger\) [16], ERS\(^\dagger\dagger\dagger\) [29], LRW\(^\dagger\dagger\dagger\dagger\) [30], and Waterpixel\(^\dagger\dagger\dagger\dagger\dagger\) [17] on Dataset II. The results of these approaches are obtained via online source codes. In qualitative analysis, we consider the implementations of Waterpixel since it is a gradient-based approach, the quality of watershed is dependant on the borders contrast. In CSS model, each weight coefficient, described in (7), is set to 1/3. In order to compare fairly with other algorithms, each algorithm extracts the same number of superpixels. All the test images are partitioned into roughly 400 superpixels. We first give the visual comparison results of these algorithms shown in Fig. 8. An area of interest is amplified to illustrate further details as shown in Fig. 9. ERS presents clear segmentation details, the overall visual effect is the worst, because of the rough boundary. LSC generates uniform and regular shape in size, although its boundary adhesion is not very good. We can see that Waterpixels can in principle tend to be well attached to target boundaries. If we look at the contours of targets missed by Waterpixel, we observe that is due to the weakness of the gradient (see ivory), as illustrated in Fig. 9. SLIC boundary adhesion is poorer than LSC. Superpixel boundaries by LRW adhere to the object boundaries very well. Our algorithm can well describe the region with strange gradient boundaries.

Superpixel looks for a method to decompose an image into regular regions that adhere to object boundaries. To assess the adhesion of superpixel to object boundaries, it is

\(^\dagger\)http://ivrl.epfl.ch/research/superpixels
\(^\dagger\dagger\)http://jschenthu.weebly.com/projects.html
\(^\dagger\dagger\dagger\)https://github.com/mingyuliutw/ers
\(^\dagger\dagger\dagger\dagger\)https://github.com/shenjianbing/lrw14
\(^\dagger\dagger\dagger\dagger\dagger\)http://cmm.ensmp.fr/~machairas/waterpixels
necessary to analyze some criteria for quantitative comparison. The study evaluates the performance of all algorithms according to BR, UE, and the evaluation is based on the average metrics of five color test images used in Fig. 8. Besides, to get a comprehensive and objective evaluation, the experiment exams our CSS on 200 images in the test partition of the BSDS500 dataset. We compute the average BR, UE of each model. Figure 10 displays the quantitative evaluation of all approaches use these two metrics. For a given number of superpixels, Fig. 10 reveals that the LSC segmentation is a very stability and presents better boundary adherence than SLIC. Waterpixel and LSC have better boundary recall performance, and the results are similar with the number of superpixels in segmentation. ERS has better boundary adherence, but it sacrifices regularity and perceptual satisfaction. Moreover, as seen in Fig. 10, the performance of SLIC is better than LRW at a low superpixel density. Waterpixel algorithm adopted a spatially regularized gradient to achieve a better balance between superpixel regularity and boundary adherence of target. For high superpixel density, our CSS offers the highest boundary recall. It is found that compared to the other algorithms, CSS created regular and compact superpixels with better boundary adherence because the proposed CSS has gradient feature represent so that image boundaries can be detected more accurately.

In order to evaluate whether the proposed method could
produce competitive results dealing with objects with weak and irregular boundaries, the next experiment will be applied to test our algorithm on a low contrast image from BSDS500 as shown in Fig. 11. It could be seen that clouds in the sky have weak boundaries, as well as the inside of the pyramid. Therefore, gradient-based models seem to be sensitive to weak boundaries in general. ERS has the worst performance in terms of superpixel compactness compared with other methods. Compared with ERS, LSC and SLIC obtain better contour fitness though LSC yields several region leakages in some low contrast areas. Our superpixel method achieves a better performance of both BR and UE measurements than LSE and SLIC do.

During initial matching, the number, shape and size of each key superpixel’s neighborhood are uncertain so that a key superpixel and a certain threshold may have more than one candidate superpixel in the target image blends to be match. The number of candidate superpixels that are matched is related to the threshold values selected by our algorithm, as well as the shapes and features of superpixels. Therefore, our method can set the maximum number of candidate superpixels in the target image participating in the matching task subjectively. Here, we first conduct an experiment to discuss the effect of the candidate superpixel numbers on matching accuracy. Table 2 lists the matching accuracies of our method under different candidate superpixel numbers. It is clear from the table that the most suitable empirical value of the candidate superpixels would be around 5, because the less value of it will lead to low matching accuracy. However, too many candidate superpixels participate in the subsequent neighborhood matching, which will increase the complexity of the algorithm and delay the matching time.

To illustrate the effectiveness of the proposed scheme, the experiment compares our method to other four well-known approaches, such as SIFT[3], SURF[4], and 800 superpixels in Table 1. Obviously, LRW has better ability to adhere to the object boundaries, which would improve the segmentation performance compared to others. As a graph-based approach, LRW could not only extract the weak boundary of object but also preserve the object topology structure. Benefiting from its graph model, it can detect the density change from the boundary pixel as well. By contrast, gradient-based methods, such as Waterpixel and our method could not accurately capture the smooth object boundaries due to the limitations of the models. Because gradient-based superpixel segmentation models measure pixels’ similarity based on distance metric could not efficiently describe relationship among pixels while these pixels are not obviously different from their neighboring pixels. Therefore, gradient-based models seem to be sensitive to weak boundaries in general. ERS has the worst performance in terms of superpixel compactness compared with other methods. Compared with ERS, LSC and SLIC obtain better contour fitness though LSC yields several region leakages in some low contrast areas. Our superpixel method achieves a better performance of both BR and UE measurements than LSE and SLIC do.

![Fig. 10](image1) Comparison with other algorithms, up: Dataset III, down: BSDS500 test set.

![Fig. 11](image2) Visual comparison between our method and other five well-known approaches on an image from BSDS500 ($K = 800$), from the first row to last is SLIC, ERS, LRW, LSC, Waterpixel, and our method, respectively.

| Superpixels 400 | 600 | 800 |
|----------------|-----|-----|
| Metrics       | BR  | UE  | BR  | UE  | BR  | UE  |
| SLIC          | 0.8796 | 0.2269 | 0.9053 | 0.1906 | 0.9234 | 0.1834 |
| ERS           | 0.8781 | 0.2401 | 0.8902 | 0.2014 | 0.9104 | 0.1950 |
| LRW           | 0.9132 | 0.2005 | 0.9267 | 0.1866 | 0.9462 | 0.1653 |
| LSC           | 0.8953 | 0.2135 | 0.9178 | 0.1897 | 0.9273 | 0.1783 |
| Waterpixel    | 0.8841 | 0.2206 | 0.9152 | 0.1934 | 0.9219 | 0.1667 |
| Ours          | 0.9057 | **0.1963** | 0.9206 | 0.1853 | 0.9312 | 0.1725 |
Table 2  Matching accuracy of our method under different candidate superpixel numbers.

| Image Pair | Key Superpixels | Maximum Numbers of Candidate Superpixels |
|------------|-----------------|-----------------------------------------|
|            | 3   | 4   | 5   | 6   | 7 |
| car        | 200 | 0.7676 | 0.7765 | 0.7832 | 0.7929 |
|            | 300 | 0.8365 | 0.8432 | 0.8507 | 0.8486 |
|            | 400 | 0.8593 | 0.8612 | 0.8675 | 0.8701 |
| mountain   | 200 | 0.8486 | 0.8501 | 0.8539 | 0.8552 |
|            | 300 | 0.8473 | 0.8497 | 0.8527 | 0.8561 |
|            | 400 | 0.8421 | 0.8465 | 0.8505 | 0.8554 |

Fig. 12  Matching comparison between two image pairs using different approaches. From up to down: SIFT, SURF, ORB, AKAZE, SMCA, LDSD, and Ours, respectively.

Fig. 13  Comparison of accuracy for keypoints matching using different algorithms. From up to down: noise free and Gaussian noise with zero mean (300 keypoints), left: car image pair, right: mountain image pair.

Fig. 14  Comparison of accuracy for 300 keypoints matching using different algorithms. From up to down: noise free and Gaussian noise with zero mean, variance 0.05, left: car image pair, right: mountain image pair.

AKAZE [31], and ORB [5]. In addition, the comparisons are made among typical superpixel matching algorithms from recent years. One is low dimensional superpixel descriptor (LDSD) [20] for visual correspondence estimation, another is co-saliency detection using superpixel matching (SMCA) [32]. The tests also evaluate Dataset III under Gaussian noise environment. Due to the limited space, only a few examples can be included in this experiment. Figure 12 depicts visual matching results of these approaches respectively on two image pairs. As can be seen in Fig. 12, when the overlapping degree remains low and there is obvious deformation (car image pair), the number of feature points matched could listed from small to large is SMCA, LDSD, CSIS, ORB, SIFT, SURF, and AKAZE, respectively, so our method is the most moderate one.

For quantitative analysis, Precision and Recall are utilized to assess the performances. Figure 13 shows the relations between Precision values via numbers of keypoints of all matchings. We also added Gaussian white noise with zero mean to the test image pairs to evaluate how robust our method against the effect of noise, and reported matching precision of all methods shown in the same figure. Figure 14 illustrates the matching performances of all approaches by computing the Recall curves. We can observe from Figs. 13 and 14, the performance between our method and AKAZE...
Next, we executed our approach on two image pairs, and compared it with several popular stitching algorithms. Figure 15 show the alignment results. From this figure, one can see that the SIFT exhibits a visible seam. ORB and AKAZE approaches suffer from color discontinuities in the overlapping boundaries. The results of LDSD and SMCA are similar, and the overlapping regions of the stitching images are uniform in the luminance and colour. Because there are almost no ghosting and seams, the stitched images achieve good performance. But if one looks closely, some ghosts will still be found in some areas (indicated by white ellipse in Fig. 15). The proposed method reduces alignment errors without significant misalignments are found. The overlapping boundaries are well aligned due to the precise matching of these regions. Panoramic image stitching and better precision matching schemes are the two main reasons for CSIS to achieve seamless stitching.

4.3 Experiment for Matching Time-Consuming

To satisfy the needs of real-time applications, the computation complexity of algorithm should be limited. This paper makes a comparison for the matching time of every method, including detection and description. For our method, LDSD and SMCA, the time required for superpixel segmentation is not included. Figure 16 records the time required for each algorithm to match features extracted from test images of the dataset III. We consider the OpenCV implementations of SIFT, SURF and ORB since these implementations are highly optimized in terms of speed. The number of keypoints for SIFT, SURF, AKAZE and ORB can be set in their respective functions. In addition, we also show the timing results of the AKAZE algorithm, which was written by Alcantarilla et al in C++ at the publicly available website. The proposed method, SMCA and LDSD involved in the comparison are implemented based on C++ codes written by the authors and executed in the same test environment. As can be seen in this figure, SIFT requires longer execution time than SURF and AKAZE. The time consumption of SURF is acceptable for most real-time applications, whose computational time can be up to 3s (400 feature points). Meanwhile, it can be observed that the average runtime of the proposed method is faster than other methods expect ORB and AKAZE algorithms. It demonstrates a moderately improvement in processing speed. This is mainly because of the superpixel, adopted in feature matching, is utilized to replace the pixel grid which will help to improve the execution speed. The LDSD algorithm also uses the superpixel to describe the image features, which has a similar complexity to our algorithm. We can observe that the computational time of LDSD remains less than SURF, and even faster than methods such as SMCA. Besides, the proposed method overcomes classical image matching approaches' disadvantages, including box filters and additive operator.

algorithm is similar, and SIFT descriptor always returns slightly lower matching accuracy than ORB and SURF in noise case. The proposed CSIS is still accurate and efficient in both noise-free case and Gaussian noise case. The reason may be that superpixel neighborhood matching scheme adopted by CSIS can well resist the influence of noise in images.

---

\[http://www.robosafe.com/personal/pablo.alcantarilla/kaze.html\]
splitting, etc. Moreover, our method does not contain the construction of image pyramids commonly used in the classical image matching, such as SLIC et al. Therefore, the time consumption of our approach is moderately lower, especially there are a lot of feature points.

5. Conclusions

This paper developed a novel superpixel extraction method and an efficient superpixel-based image matching scheme, yielding accurate blending results. Our method has five points advanced than others. First, the study defined a new distance metric for generating compact superpixels which adhered to image boundaries well. Second, superpixel matching is used to replace individual keypoint, this is the key point to speed up our method. Third, the contents and shapes of the neighborhood superpixels are considered while generating feature descriptors. Fourth, for detecting superpixels of interest, the feature vector fuses three low-level features, texture, color, and orientation. Fifth, to speed up the search for superpixel pairs, the local entropy of each superpixel is computed to reduce the procedure in matching tasks. It also achieves accurate matching. Experiments were implemented with some natural images, and the results demonstrated that the features extracted by the proposed superpixel descriptor had the capability of describing the contents of an image, and reduced the execution time as well. The proposed descriptor is convenient for various applications, such as objection recognition, target tracking, FPGA based embedded systems, etc. There is a limitation for the proposed method. Specifically, we label the superpixels according to the gradient of pixels, therefore, some pixels on weak boundaries are not correctly classified.

Acknowledgments

The authors would like to thank the anonymous reviewers and the associate editor for their insightful comments that significantly improved the quality of this paper. This work was supported by the National Nature Science Foundation of China under Grant 61872143, 61771196.

References

[1] M. Kitagawa and I. Shimizu, “Memory saving feature descriptor using scale and rotation invariant patches around the feature points,” IEICE Trans. Inf. and Syst., vol.E102-D, no.5, pp.1106–1110, 2019.
[2] K. Krishnakumar and S.I. Gandhi, “Video stitching using interacting multiple model based feature tracking,” Multimed. Tools Appl., vol.78, no.2, pp.1375–1397, 2019.
[3] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.
[4] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, “SURF: Speeded-up robust features,” Comput. Vis. Image Und., vol.110, no.3, pp.346–359, 2008.
[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” Proc. IEEE Int. Conf. Comput. Vision (ICCV), pp.2564–2571, 2011.
[6] S. Leutenegger, M. Chili, and R.Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” Proc. IEEE Int. Conf. Comput. Vision (ICCV), pp.2548–2555, 2011.
[7] L. Huang, C. Chen, H. Shen, and B. He, “Adaptive registration algorithm of color images based on SURF,” Measurement, vol.66, pp.118–124, 2015.
[8] S. Li, R. Shi, and H. Ye, “An efficient approach of color image matching by combining color invariant and ORB feature,” Adv. Graph. Commun. Med. Technol., vol.417, pp.195–203, 2016.
[9] S. Li, L. Yuan, J. Sun, and L. Quan, “Dual-feature warping-based motion model estimation,” Proc. IEEE Int. Conf. Comput. Vision (ICCV), pp.4283–4291, 2015.
[10] L. Xu, B. Luo, M. Kong, B. Li, and Z. Pei, “Fast superpixel segmentation via boundary sampling and interpolation,” IEICE Trans. Inf. and Syst., vol.E102-D, no.4, pp.871–874, 2019.
[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. S¨usstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.11, pp.2274–2282, 2012.
[12] Z. Li and J. Chen, “Superpixel segmentation using linear spec-
tral clustering.” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.1556–1563, 2015.
[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, 2015.
[18] L. Xu, B. Luo, and Z. Pei, “Boundary-aware superpixel segmentation based on minimum spanning tree.” IEICE Trans. Inf. and Syst., vol.E101-D, no.6, pp.1715–1719, 2018.
[19] R. Giraud, V.-T. Ta, and N. Papadakis, “Robust superpixels using color and contour features along linear path.” Comput. Vis. Image Und., vol.170, pp.1–13, 2018.
[20] S. Du and T. Ikenaga, “Low-dimensional superpixel descriptor and its application in visual correspondence estimation.” Multimed. Tools Appl., vol.78, no.14, pp.19457–19472, 2019.
[21] Y. Xie, L. Xu, and Z. Wang, “Automated co-superpixel generation via graph matching.” Signal Image Video Process., vol.8, no.4, pp.753–763, 2014.
[22] X. Dong, J. Shen, and L. Shao, “Hierarchical superpixel-to-pixel dense matching.” IEEE Trans. Circuits Syst. Video Technol., vol.27, no.12, pp.2518–2526, 2017.
[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.” Proc. IEEE Int. Conf. Comput. Vision (ICCV), pp.416–423, 2001.
[24] J. Zaragoza, T.-J. Chin, Q.-H. 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, 2014.
[25] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns.” IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.7, pp.971–987, 2002.
[26] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection.” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.886–993, 2005.
[27] J. Han and K.-K. Ma, “Fuzzy color histogram and its use in color image retrieval.” IEEE Trans Image Process., vol.11, no.8, pp.944–952, 2002.
[28] M. Brown, R. Szeliski, and S. Winder, “Multi-image matching using multi-scale oriented patches.” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.510–517, 2005.
[29] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation.” Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.2097–2104, 2011.
[30] 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, 2014.
[31] P.F. Alcantarilla, J. Nuevo, and A. Bartoli, “Fast explicit diffusion for accelerated features in nonlinear scale spaces.” Proc. British Machine Vision Conf. (BMVC), pp.1–11, 2011.
[32] Z. Zhang, Z. Wu, Q. Jiang, L. Du, and L. Hu, “Co-saliency detection based on superpixel matching and cellular automata.” KSII Trans. Internet and Inf. Syst., vol.11, no.5, pp.2576–2589, 2017.