American Journal of Signal and Image Processing
1(1):25-32, June 2019.

Efficient single image dehazing by modifying the dark channel prior

Sebastian Salazar

Received: 9 November 2018 **Accepted:** 8 May 2019 **Published:** 3 June 2019

Abstract

Outdoor images can be degraded due to the particles in the air that absorb and scatter light. The produced degradation generates contrast attenuation, blurring, and distortion in pixels, resulting in low visibility. These limit the efficiency of computer vision systems such as target tracking, surveillance, and pattern recognition. In this paper, we propose a fast and effective method, through modification in the computation of the dark channel which significantly reduces the artifacts generated in the restored images presented when using the ordinary dark channel. According to our experimental results, our method produces better results than some state-of-the-art methods in both efficiency and restoration quality. The processing time in tests shows that the method is adequate for images with high-resolution and real-time video processing.

Keywords

Channel Prior , Dehazing, Image enhancement, Single image dehazing

1. Introduction

The presence of environmental disturbances such as haze and smog gives outdoor images and videos undesirable characteristics that affect the ability of computer vision systems to detect patterns and perform an efficient feature selection and classification. These characteristics are caused by the decrease in contrast and color modification originated by the presence of suspended particles in the air. Hence, the task of removing the haze, fog, and smog (dehazing), without compromising the image information, takes on special relevance. Therefore, to improve the performance of systems such as surveillance [1], traffic [2], self-driving vehicles [3] is essential to develop new and better dehazing methods. This problem has been studied extensively in the literature with two main approaches: methods that use multiple images [4] and methods that use just a single image [1].

Within the single-image approach, some results can be mentioned relevant results, such as the obtained by Tan et al. [5], Fattal [6], and Tarel et al. [7] where the main problem of these proposed methods is the time processing required and that the proposed methods are not based on solid physics concepts. The most studied method in the literature is presented by He et al. [8] where the dark channel prior (DCP) is introduced. The DCP is a simple but effective approach in most cases, although it produces artifacts around regions where the intensity changes abruptly. Usually, in order to eliminate the artifacts, a refinement stage is necessary, which has an impact on time processing [1, 9]. To get around this problem, He et al. [8] uses a soft-matting process, Gibson et al. [10] proposed a DCP method based on the median operator. Zhu et al. [11] introduced a linear color attenuation prior, and Ren et al. [12] used a deep multiscale neural network.

This paper presents a fast novel method in which a modified dark channel is introduced, improving the quality of the depth estimations of the image elements and reducing significantly the artifacts generated when the traditional dark channel is used. The modification of the proposed dark channel, unlike most state-of-the-art methods, makes a refinement stage unnecessary; this has a positive impact on the simplicity and speed of the dehazing process. Experimental results demonstrate the effectiveness of the proposed method, and when compared with three state-of-the-art methods, the proposed method achieves a higher restoration quality and requires significantly less time. The paper is organized as follows. In Section 2, the image degradation model and the dark channel prior used is discussed. The proposed method is presented in Section 3. In Section 4, experimental results and analysis are shown. The conclusions are described in Section 5.

2.Literature survey

In this section, we briefly describe the related methods that apply to our proposed approach.

2.1 Superpixel segmentation

One type of image segmentation method is called the superpixel segmentation method. It groups the pixels of an image into perceptually meaningful atomic regions that can be used to replace the rigid structure of the pixel grid. A simple linear iterative clustering (SLIC)-based superpixel algorithm is proposed by Achanta et al. [29]. It uses a *k*-means clustering approach to efficiently generate superpixels, and it can adhere to the boundaries very well. The only parameter (*k*) in the SLIC algorithm is to assign the desired number of approximately equally sized superpixels. The algorithm is briefly described in the following paragraph. Details of the procedures have been reported in [29].

This SLIC algorithm is adopted in CIELAB color space. The SLIC algorithm adapts a *k*-means clustering approach to efficiently generate the superpixels, and it adheres to the boundaries very well. First, the clustering procedure begins with an initialization step where the *k* initial cluster centers, where (l, a, b) are the three color components of a pixel, and (x, y) are its two spatial coordinates, are sampled on a regular grid (called a superpixel), spaced S pixels apart. The S interval is N/k−−−√N/k, in which *N*represents the number of pixels for an image. In order to avoid centering a superpixel on an edge or on a noisy pixel, the centers are moved to seed locations corresponding to the lowest gradient position in a $n \times n$ neighb$C_i = [l_i, a_i, b_i, x_i, y_i]^T$, $i = 1, 2, \cdots, k$,orhood. As is known to us, the edge or noisy pixel is often positioned on a pixel point that has the largest gradient variation. Therefore, selecting the lowest gradient pixel point to position the center for a superpixel can efficiently reduce the chance of seeding a superpixel with an edge or a noisy pixel.
Additionally, in order to speed up the SLIC algorithm, the search area is reduced to the size of $2S \times 2S$ around the superpixel center, in contrast to the traditional *K*-means clustering method. Then, by computing the distance between the center point and other pixel points within the cluster, an update step adjusts the cluster centers to be the mean vector of all the pixels belonging to the cluster, once each pixel has been associated to the nearest cluster center. The residual error is computed by means of the L_2 norm between the new cluster center locations and previous cluster center locations. Finally, the assignment and updated steps can be repeated iteratively until the error converges. As [29] discussed, after iterating ten times, most images can achieve the convergence. Figure 1 shows an example of SLIC segmentation for a superpixel that is roughly the size of 300 pixels.

In our work, since all points lie in a plane, the Helmert transform becomes transformations from one rectangular coordinate system to another rectangular system. These transformations include rotation, scaling, and translations for all points. The transformation equations can be formed in matrix notation using mathematical operations [30].
[XpYp]=[A−BBA][xpyp]+[txty],[XpYp]=[AB−BA][xpyp]+[txty],
(1)

where (x_p, y_p) coordinates are transformed into (X_p, Y_p) coordinates by the addition of translations t_x and t_y. A and B are the transformation parameters. This transformation is called the Helmert transformation [30], also known as similarity transformation. Helmert transformations have a lower degree of freedom, therefore they have lower computational complexity available to transform the coordinates of points in one point (x, y) into coordinates in another point (X, Y). As shown in Eq. (1), only four parameters are needed to compute the coordinate transformations, such as rotation, scaling, and translations. In addition, a well-known transformation known as the affine transformation usually uses map coordinate transformations. However, affine transformations require six parameters to achieve transformations. The advantages of the Helmert transformation include not only resistance to rotation, scaling, and translations, but also reduced computational complexity. For instance, given the coordinates of two pairs, we can obtain four parameters of Helmet transformation by Eq. (1). Hence, in our experiments, we adopt the Helmert transformation instead of affine transformation to acquire the coordinates after transformation.

3. Proposed Method

In this study, we propose keypoint-based image forensics based on the Helmert transformation and SLIC algorithm. The main procedures include keypoint extraction and matching, clustering and group merging, and forgery region localization and refining. Figure 2 illustrates the flowchart of the proposed system. Details of procedures are described in the following subsections.

3.1 Keypoint extraction and matching

Based on the SIFT algorithm [22], we can obtain all candidates of keypoints and the corresponding descriptors for an image. Using these candidates, we will search for the best matching pairs to perform additional grouping.

First, each keypoint within all candidates will compute the Euclidean distance between other keypoints via corresponding descriptors, and will also perform the matching operation. The nearest neighbor distance ratio (NNDR) [31], which is the ratio of the smallest distance to the second-smallest distance, is used to perform the matching. This ratio is depicted as
D(A,B)D(A,C)≤TNNDR,D(A,B)D(A,C)≤TNNDR,

where *D* is the Euclidean distance between the descriptors of two keypoints, keypoints *A* and *B* are the nearest neighbors, and keypoint *A* and keypoint *C* are the second-nearest neighbor. T_{NNDR} is a constant value. If Eq. (2) is satisfied, keypoints *A* and *B* are regarded as a matching pair. Generally, keypoint *A* is the source point and keypoint *B* is the target point. Our approach uses the Euclidean distance between descriptors to estimate the similarities.

After computing the distances for all keypoints, we can obtain all matching pairs in an image. In order to avoid incorrect matching pairs, if the distance between matched pairs is less than T_{NNDR}, they will be ignored and deleted.

3.2 Clustering and group merging

Our clustering strategy includes clustering and group combining. We improve the clustering method proposed in [16] to perform the coarse clustering process. A clustering yields two match groups: source and destination. They are considered as correspondent regions inside the image and are good cloning. In [16], the clustering strategies only used spatial distance and correspondence angle between matched pairs to perform the clustering. However, when the forgery region is too large, it could result in the matching pairs belonging to the same group that are assigned to the different groups, as shown in Fig. 3. That is, a group may be segmented into many subgroups. In Fig. 3, the red subgroups could not be merged together into a group, and the blue subgroups could not be merged together either.

Hence, in order to solve this problem, we improve the clustering strategies proposed by [16] to achieve the coarse clustering. The modified clustering schemes are described by the following. Given any two matching pairs belonging to corresponding subgroups (source and target subgroups), they are considered as correspondent regions in an image and are tampering candidates.

- Spatial adjacency: consider that we have a match pair between keypoints *A* and *B*belonging to group *G*. Keypoint *A* might belong to the G_{source} subgroup, and keypoint *B* might belong to the G_{target} subgroup, or vice versa. For a subgroup to admit a paired keypoint as a new member, the spatial distance between the keypoint and its nearest keypoint in such a subgroup needs to be smaller than a predefined threshold, T_c. Moreover, it is necessary to analyze both matched keypoints, since they have to be in the same group, but in different subgroups.
- The angle consistency: the angle in the range of [0°, 360°] with a 15° step is used to determine the angle consistence. It can obtain 24 range partitions. As described above, a new

keypoint A candidate to be included into G_{source} will be included in G_{source}, only if the angle of the line that connects the candidate point A and its matching point B stays in the same range of the other points in G_{source}.

After performing coarse clustering, we will further merge these clusters based on the Helmert transformation and spatial adjacent relationship between clusters. Therefore, the transformation can efficiently merge some clusters with a high correlation into a compact cluster. A Helmert transformation is used to describe the relationships between two different coordinate systems without distortion. In 2D space, the Helmert transformation is defined as Eq. (1). We use the Helmet transformation to analyze the geometric relationships between matching pairs. Assuming that the number of keypoints in a cluster is greater than one, we will compute the Helmert parameters of the cluster (source and target subgroup); otherwise, this cluster will be discarded. For instance, given any two matching pairs, by assuming that (X_p, Y_p) are target coordinates and (x_p, y_p) are source coordinates, the transformation can easily compute and obtain four Helmert parameters by Eq. (1).

Assuming that there is a keypoint from another group, C', within the search range we specified, this keypoint will be checked whether it belongs to source or target subgroup. It is because we do not constrain which keypoint stays in source or target subgroup for a matching pair in the previous matching process. During the matching process, the same region may be clustered into different groups, and the matching pairs may stay in the subgroup opposite to the other, as shown in Fig. 4a. Assuming that this keypoint belongs to the target subgroup in group C', we will exchange all members in the target subgroup with those of the source subgroup in group C', as shown in Fig. 4b. Afterwards, we transform all members in the source subgroup for group C' to new members in target subgroup by means of the Helmert parameters derived from group C. Then, we will compute the difference in the spatial coordinates between target keypoints in group C and new target keypoints in group C'. When this difference is smaller than a threshold, T_h, two groups are merged and then Helmert parameters derived from group C are updated. Based on our experimental test, we assigned the threshold value, $T_h = 10n$, where n denotes the number of keypoints in the group.

Fig. 4 An example for clustering profile. **a** The matching pairs stay in the opposite subgroup corresponding to group C and group C′ in the same region. **b** Clustering objective

Next, we use a rectangular search range, which is defined as (x_{max}, y_{max}, x_{min}, y_{min}) belonging to the lower right and upper left coordinates of keypoints in source subgroup, to perform group merging. The target subgroup also creates a rectangular search range. If there is no keypoint presented in the rectangular search range, this rectangular range will expand the search range to find other clusters until one of the terminal conditions is satisfied. The terminal conditions are defined as follows.

1. The number of the extension (N_e) has reached a value of five, and there is no cluster that can be combined. Here, the range of each extension (R_e) is multiplied the rectangle searching region by a factor of 1.25.

2. The rectangle search region (S_r) is greater than 0.125 times of size of a host image.

Repeat the above steps until no clusters can be combined. Finally, we remove the invalid clusters that involve less than five keypoints.

Then, we apply a Gaussian filter to the correlation map in order to reduce the noisy pixels, and a binary correlation map is given by means of a threshold (T_b). If the ZNCC value for point (x, y) is greater than a threshold, this point (x, y) is assigned as true; otherwise, this point is assigned as false. Next, we will perform connected-component labeling on this binary map. This threshold, T_b, is set to 0.55, which is a value obtained through experimentation.

If the largest region involved in connected-component labeling touches the border of the binary map, it means that the range of this region is bigger than the range of the binary map, as shown in Fig. 7a. The top and right sections of this region touch the borders. Therefore, this region will be expanded in a rectangular interval along the touched border. The steps described above are repeated until the largest region does not touch the border, as shown in Fig. 7b. Based on an empirical value obtained in our experiments, the expanded range (E_r) is multiplied the width or height of this sub-image by a factor of 1.25 depending on the direction of touching border. All points in image *I* are finished, the content of the binary correlation map is filled to the ZNCC binary map corresponding to the location. For instance, Fig. 8a shows the ZNCC binary map. Next, we combine the SLIC superpixel segmentation described in Section 2.1 to achieve the forgery region localization.

The host image is segmented into many sub-regions by the SLIC algorithm. In the SLIC algorithm, the smaller the size of a superpixel (*S*), the greater the number of superpixels present. Moreover, very few true edges are missed. In contrast to increasing size, the number of superpixels is reduced, and many true edges will be missed. Therefore, in our approach, the size of a superpixel (*S*) is assigned to 300 pixels by experiments. For each sub-region, we will count the number of pixels that are considered true in the ZNCC binary map. If this number (N_d) is greater than a threshold in the relative sub-region, all the pixels in this sub-region are labeled as a detection map that serves as a part of forgery regions, as shown in green color areas of Fig. 8d. Afterwards, we label the connected components as the detection map, and delete the regions that have an area less than 0.1%. Finally, each of the remaining regions will use the convex-hull morphologic method to connect together in the binary detection map. Figure 8 illustrates the profile of the detection map. After performing our proposed method, we can efficiently detect and localize the forgery regions more precisely.

4. Results and Analysis

To verify the performance of the proposed image forensics, the experimental results are compared to Amerini et al. [13], Silva et al. [16], Pun et al. [18], and Li et al. [19] to perform the forgeries, including copying and translations, scaling, rotation, and compression.

4.1 Experimental setup and datasets

Table 1 illustrates the parameters presented in the experiments. According to our experiments, we systematically vary the related thresholds within 50% to 200% and observe performance changes; afterwards, they are given, and some thresholds are derived from the literature [16, 31]. However, the assignment of these parameter values can be modified by the user based on the data. The experiments were implemented in Microsoft Visual Studio C#, on an Intel® core i5–4570@ 3.2 GHz computer with 4 GB of RAM running a Windows 7 64 bits platform.
Every image in every dataset has its own binary ground truth displaying the original and duplicated regions in white color. And the tampered region within the datasets is of a single region copied one time and stayed in the same image.

4.2 Performance evaluation

For performance evaluation, we used the precision, recall, F_1 [8, 18], and the false positive rate (FPR) [16] to demonstrate our proposed method. These evaluation criteria are expressed as:
Precision: represents the probability that the detected regions are truly the forgery regions, as expressed in (4).

precision=|TP||Ωretrieved|precision=|TP||Ωretrieved|
where $|\Omega_{retrieved}|$ denotes the number of the detected forgery pixels by our proposed method from the datasets, |TP| (true positive) represents the number of correctly detected forged pixels labeled as forged regions in the ground truth.
Recall: represents the probability that the forgery regions are detected, as expressed in (5).

recall=|TP||Ωrelevant|.recall=|TP||Ωrelevant|.
where $|\Omega_{relevant}|$ represents the ground truth forgery regions of the datasets.
F_1: this score combines both the precision and recall into a signal value. It is calculated by (6).

F1=2·precision·recallprecision+recall.F1=2·precision·recallprecision+recall.
FPR: indicates the percentage of incorrectly located tampering regions. It is defined as

FPR=|FP||Ωnormal|,FPR=|FP||Ωnormal|,

where $|\Omega_{normal}|$ represents the number of pixels that do not belong to the tampering regions in the ground truth, |FP| (false positive) denotes the number of wrongly detected as tampering pixels by our proposed method.

Because the datasets have been tampered with in different ways, they are not consistent in our experiments, and therefore we compute the average values for these evaluation criteria in the dataset to verify the performance. As indicated above, the *precision* is the probability that a detected forgery is truly a forgery, and the *recall* is the probability that a forgery image is detected. Generally, a higher *precision* and a higher *recall* represent better performance.

5.Results

Regarding the different forgery images created by copying and translation, scaling, rotation, and compression, the experimental results are presented and discussed in the following section.

The forgery images are simply copied and moved operations, such as the CMH1 and D0 datasets. Tables 2 and 3 illustrate the detected results compared to our proposed method and the methods of Amerini et al., Silva et al., Pun et al., and Li et al..Figure 9 presents several detection results for simple copying.

Table 2

6.Conclusion

In this study, the major strategy of our proposed algorithm focuses on a single tampered region detection. And we have proposed keypoint-based image forensics for copy-move forgery images based on a Helmert transformation and SLIC superpixel segmentation. Compared to the sliding window approach, the keypoint-based technique can be applied at a lower computational cost because of the significantly reduced number of points required. In addition, we use the Helmert transformation to estimate the geometric relationships between matching pairs and to work the merging clusters. On the other hand, we use an SLIC algorithm to localize the tampering regions more precisely. Based on these strategies, we can keep much more important information to conduct image forensics.

As previously presented in the experiments, it is clear that the proposed method is highly robust against many kinds of forged images, such as geometric transformations (scaling, rotation) and JPEG compression. However, the current method is not robust against symmetric, recurring, and smooth patterns for tampering region. Progress in detecting symmetric, recurring, smooth forgery images, and tampering region copied multiple times will be a major focus in the future.

References

1. M. Al-Qershi Osamah, B.E. Khoo, Passive detection of copy-move forgery in digital images: state-of-the-art. Forensic Sci. Int. **23**(1), 284–295 (2013)View ArticleGoogle Scholar

2. J.W.G. Wang, J. Liu, A. Zhang, Y.W. Dai, Z.Q. Wang, Fast and robust forensics or image region-duplication forgery. Acta Automat., Sinica **35**(12), 1488–1495 (2009)View ArticleGoogle Scholar

3. S.J. Ryu, M.J. Lee, H.K. Lee, in *Information Hiding*. Detection of copy-rotate-move forgery using Zernike moments (Springer-Verlag, Berlin, 2010), pp. 51–65View ArticleGoogle Scholar

4. S.J. Ryu, M. Kirchner, M.J. Lee, H.K. Lee, Rotation invariant localization of duplicated image regions based on Zernike moments. IEEE Trans. Inf. Forensics Secur. **8**(8), 1355–1370 (2013)View ArticleGoogle Scholar

5. Y. Huang, W. Lu, W. Sun, D. Long, Improved DCT-based detection of copy-move forgery in images. Forensic Sci. Int. **206**(3), 178–184 (2011)View ArticleGoogle Scholar

6. X. Wang, X. Zhang, Z. Li, S. Wang, in *3rd Int. Conf. on Multimedia Information Networking and Security*. A DWT-DCT based passive forensics method for copy-move attacks (2011), pp. 304–308Google Scholar

7. S. Bravo-Solorio, A.K. Nandi, Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics. Signal Process. **91**(8), 1759–1770 (2011)MATHView ArticleGoogle Scholar

8. R. Davarazni, K. Yaghmaie, S. Mozaffari, Copy-move forgery detection using multiresolution location binary patterns. Forensic Sci. Int. **231**(1), 61–72 (2013)View ArticleGoogle Scholar

9. J.C. Lee, C.P. Chang, W.K. Chen, Detection of copy-move image forgery using histogram of orientated gradients. Inf. Sci. **321**, 250–262 (2015)View ArticleGoogle Scholar

10. L. Li, S. Li, H. Zhu, X. Wu, Detecting copy-move forgery under affine transforms for image forensics. Comput. Electr. Eng. **40**(6), 1951–1962 (2014)View ArticleGoogle Scholar

11. X. Pan, S. Lyu, Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. **5**(4), 875–867 (2010)View ArticleGoogle Scholar

12. I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, G. Serra, A sift-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (2011)View ArticleGoogle Scholar

13. I. Amerini, L. Ballan, R. Caldelli, A.D. Bimbo, D. Tongo, G. Serra, Copy-move forgery detection and localization by means of robust clustering with J-linkage. Signal Process. Image Commun. **28**(6), 659–669 (2013)View ArticleGoogle Scholar

14. B.L. Shivakumar, S.S. Baboo, Detection of region duplication forgery in digital images using SURF. Int. J. Comput. Sci. **8**(1), 199–205 (2011)Google Scholar

15. P. Mishra, N. Mishra, S. Sharma, R. Patel, Region duplication forgery detection technique based on SURF and HAC. Sci. World J. **2013**, 1–8 (2013)Google Scholar

16. E. Silva, T. Carvalho, A. Ferreira, A. Rocha, Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. J. Vis. Commun. Image Represent. **29**, 16–32 (2015)View ArticleGoogle Scholar

17. E. Ardizzone, A. Bruno, G. Mazzola, Copy-move forgery detection by matching triangles of keypoints. IEEE Trans. Inf. Forensics Secur. **10**(10), 2084–2094 (2015)View ArticleGoogle Scholar

18. C.M. Pun, X.C. Yuan, X.L. Bi, Image forgery detection using adaptive over-segmentation and feature points matching. IEEE Trans. Inf. Forensics Secur. **10**(8), 1705–1716 (2015)View ArticleGoogle Scholar

19. J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme. IEEE Trans. Inf. Forensics Secur. **10**(3), 507–518 (2015)View ArticleGoogle Scholar

20. D. Cozzolino, G. Poggi, L. Verdoliva, Efficient dense-field copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2284–2297 (2015)View ArticleGoogle Scholar

21. D. Chauhan, D. Kasat, S. Jain, V. Thakare, in *Int. Conf. On computational modelling and security (CMS2016)*. Survey of keypoints based copy-move forgery detection methods on image (2016), pp. 206–212Google Scholar

22. D.G. Lowe, Distinctive image features from scale-invariant keypoint. Int. J. Comput. Vis. **60**(2), 91–110 (2004)View ArticleGoogle Scholar

23. H. Bay, T. Tuytelaars, L. Van Gool, in *Computer Vision-ECCV2006*. SURF: speeded up robust features (Springer-Verlag, Berlin, 2006), pp. 404–417Google Scholar

24. N. Warif, A. Wahab, M. Idris, R. Ramli, R. Salleh, S. Shamshirband, K. Choo, Copy-move forgery detection: survey, challenges and future directions. J. Netw. Comput. Appl. **75**, 259–278 (2016)View ArticleGoogle Scholar

25. M. Zandi, A. Mahmoudi-Aznaveh, A. Talebopur, Iterative copy-move forgery detection based on a new interest point detector. IEEE Trans. Inf. Forensics Secur. **11**(11), 2499–2512 (2016)View ArticleGoogle Scholar

26. M. Ikhalyel, M. Hariadi, K.E. Pummama, A study of copy-move forgery detection based on segmentation. IJCSNS Int. J. Comp. Sci. Netw. Secur. **18**(7), 27–32 (2018)Google Scholar

27. J. Zheng, Y. Liu, J. Ren, T. Zhu, Y. Yan, H. Yabf, Fusion of block and keypoints based approaches for effective copy-move image forgery detection. Multidim. Syst. Sign. Process. **24**(4), 989–1005 (2016)MathSciNetView ArticleGoogle Scholar

28. V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)View ArticleGoogle Scholar

29. R. Achanta, A. Shaji, K. Simth, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)View ArticleGoogle Scholar

30. R.E. Deakin, Coordinate transformations in surveying and mapping. Geospat. Sci. (2004) *[Online]. available* http://www.mygeodesy.id.au/documents/COTRAN_1.pdf

31. K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)View ArticleGoogle Scholar