

# Summit Navigator: A Novel Approach for Local Maxima Extraction

Tran Hiep Dinh, Manh Duong Phung and Quang Phuc Ha

**Abstract**—This paper presents a novel method, called the Summit Navigator, to effectively extract local maxima of an image histogram for multi-object segmentation of images. After smoothing with a moving average filter, the obtained histogram is analyzed, based on the data density and distribution to find the best observing location. An observability index for each initial peak is proposed to evaluate if it can be considered as dominant by using the calculated observing location. Recursive algorithms are then developed for peak searching and merging to remove any false detection of peaks that are located on one side of each mode. Experimental results demonstrated the advantages of the proposed approach in terms of accuracy and consistency in different reputable datasets.

**Index Terms**—Multi-thresholding, histogram analysis, histogram segmentation, image segmentation, Summit Navigator.

## I. INTRODUCTION

Machine vision offers an excellent tool for critical real-world inspection in industrial applications [1], for example, for detecting surface defects of film capacitors [2] or vision-based surface inspection with drones [3]. In an automated vision-based system, image processing and, in particular, image segmentation remain an essential task [4], for which automatic extraction of local maxima is an important process not only to extract the foreground for post processing but also to separate objects from the background for identification and classification. Techniques used for extraction of local maxima vary depending on the data representation employed, but in general can be categorized into groups of histogram analysis, clustering and entropy based methods [5], [6]. Among them, the histogram shape-based thresholding is widely adopted due to its simplicity, effectiveness and computational efficiency [7]. Approaches in this direction are typically based on grey-level histograms

to identify thresholds that separate an image into segmented regions. The thresholds can be either single or multiple values corresponding to a bilevel or multilevel thresholding method.

In bilevel thresholding, a global threshold of histogram characterized by two different modes is used to binarize an image into the foreground and background [8], [9], [10], [11]. The pioneer approach by Otsu [8], albeit simple and consistent, appears to be unattractive in terms of computational cost and accuracy, particularly in dealing with a multilevel thresholding problem. More recently, local features and Gaussian mixture modeling are used to extract an interested object from its image background [11]. The method however is not sensitive and threshold values have to be tuned manually.

Multilevel thresholding is used to deal with segmentation of more than two prominent modes, corresponding to some particular objects of an image. A challenge in multi-modal histogram thresholding remains the automated implementation of the image processing algorithm because natural images usually contain much unknown information, such as the number of classes or the characteristics of the histogram modes. To overcome this issue, many authors developed *ad hoc* approaches by incorporating heuristic input factors [12], [13], [14]. A Gaussian kernel convolution is employed in [12] to resolve the multiple threshold issue, but it is not self-adjusted as specific input from the user is required. Heaviside functions have recently been proposed in [13] for approximating the gray level histogram to detect local extreme. The method was recognized for satisfactory quality and low computational demands but a fixed number of phases of level sets is still required. Adaptive image thresholding is introduced in [15] using a minimax optimization technique for a weighting factor whereby the choice of energy functions is however not straightforward.

For multilevel thresholding, a hybrid approach is proposed in [16] to compute an image energy function before applying a Genetic Algorithm fitness function to overcome limitations. Nevertheless, the algorithm may be unstable, as user interactions are needed to limit the algorithm's sensitivity to irregularities in the his-

Tran Hiep Dinh and Quang Phuc Ha are with School of Electrical and Data Engineering, University of Technology Sydney (UTS), 15 Broadway, Ultimo NSW 2007, Australia. (email: Tran.Hiep.Dinh@uts.edu.au, Quang.Ha@uts.edu.au).

Manh Duong Phung is with University of Engineering and Technology, Vietnam National University (VNU), 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam. (email: duongpm@vnu.edu.vn).

togram. To deal with the complicated problem, the use of artificial intelligence with soft computing techniques [17], [18], [19], [20] has been an attractive option. These methods are, however, based on some predefined control parameters and may become inefficient with a large problem size. Other authors suggested a feasible process of finding and removing peaks [21], [22] to partition an image histogram into different modes. For example, in [21], peak sharpness and the horizontal difference between two peaks are evaluated to determine significant values of the histogram. These methods have advantages in low computational cost and simplicity of implementation but tend to be unreliable when applying to data with noise or radical variations.

Other non parametric approaches [22], [23], [24] have been proposed to detect peaks [22], [23] or segment the image histogram into different modes [24]. While being able to detect threshold values without information about the histogram distribution or the number of objects and endorsed as efficient against some segmentation tasks, these techniques are also ineffective when dealing with a noisy data [25].

In this paper, a novel algorithm is developed to automatically detect true peaks from gray-scale histograms of images and to locate these peaks without the requirements of manual inputs, as preliminarily indicated in [26]. Here, inspired by the strategic planning of mountain explorers, we formulate two location-based parameters, namely the offset distance and observability index, for the image segmentation process. These parameters are used to search for all possible dominant peaks from the best among observing locations. Notably, this non-heuristic approach does not require any *a priori* knowledge of the number of modes or distance between modes in processing. Results from various experiments are discussed to show the validity of the proposed approach.

## II. SUMMIT NAVIGATOR APPROACH FOR MULTIPLE PEAK DETECTION

It is observed that without navigating equipment, a mountain explorer has to perceptually determine routes by finding a nearest and highest location for gaining the maximum range of view. From that point, along with the same distance and height condition, the explorer can identify the next appropriate destination to go forward with the journey. By repeating this process, this mountain explorer can eventually reach the planned checkpoints to accomplish the exploration. Inspired by that tactic, the Summit Navigator is developed in this paper to deal with the peak detection issue in multilevel histograms. The algorithm comprises three main steps: (i) preprocessing with initial peak detection, (ii) searching for dominant

peaks, and (iii) merging dominant peaks found on the same side of a mode. The flowchart of the proposed algorithm is illustrated in Fig. 1.

### A. Preprocessing

In the preprocessing stage, the input histogram is first smoothed by a moving average filter to remove high frequency noise and retain peaks for further detection. For computational efficiency and effectiveness on raw data, the kernel width of the filter is chosen to be 03, the minimum interval that can take into account the previous, current and next intensity level. As the result, a pixel intensity  $h_i$  at level  $i$  after applying the moving average filtering is given by:

$$h_i = \frac{h_{i-1}^I + h_i^I + h_{i+1}^I}{3}, \quad (1)$$

where  $h_i^I$  is the intensity at level  $i$  of the input histogram. Those smoothed intensities are then compared with their two nearest neighbors in order to determine initial peaks. The selected ones should meet the following condition:

$$\begin{cases} h_i > h_{i-1} \\ h_i > h_{i+1}. \end{cases} \quad (2)$$

Let  $\mathbf{S}$  be the set of locations of initial peaks. Since local maxima are among initial peaks, it becomes a draft plan for the travel strategy to be developed. Our next stage is to determine true peaks from  $\mathbf{S}$ , which is the core of the Summit Navigator algorithm.

### B. Peak searching

To start a new search, the explorer needs to find a good observing location. This should be a position with the possibility to observe several peaks in order to detect the most dominant one. The goal of peak searching is thus to find the best observing location of initial peaks so that dominant peaks can be extracted based on this search.

Considering an input histogram of an image as the mount under exploration, let  $P$  be the number of initial peaks and  $s_k$ ,  $0 \leq k \leq P$ , the  $k$ -th element of vector  $\mathbf{S}$ . Given initial peaks satisfying  $h_{s_k} > h_{s_{k-1}}$ , the observing location,  $L_k$ , as illustrated in Fig. 2, is the point on the intensity axis fulfilling the following condition:

$$L_k \leq s_k - \Delta L_k, \quad (3)$$

where  $\Delta L_k$  is the offset distance from the current peak to the observing location. This offset distance is calculated based on the location and frequency of two consecutive peaks  $k$  and  $k-1$  as follows:

$$\Delta L_k = \frac{h_{s_k}(s_k - s_{k-1})}{\epsilon + |h_{s_k} - h_{s_{k-1}}|}, \quad (4)$$

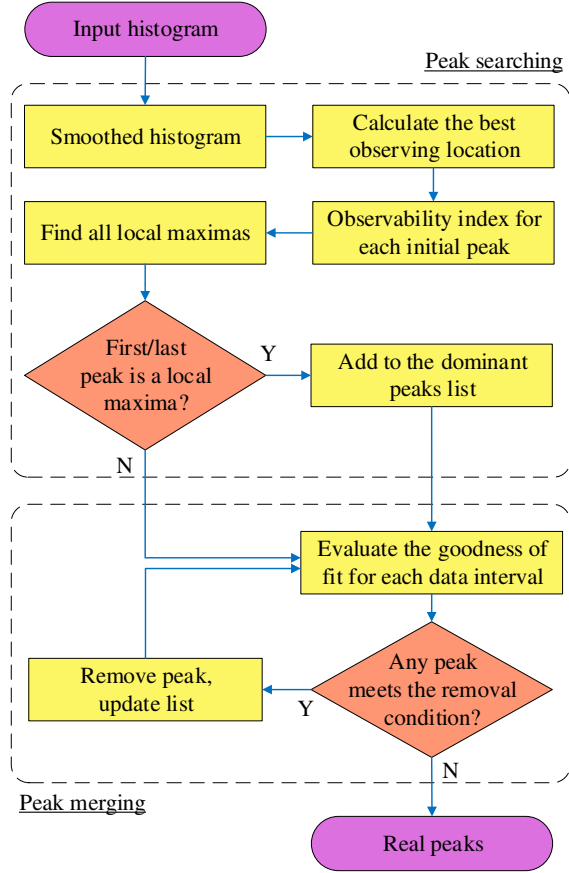


Fig. 1: Flow chart of the Summit Navigator algorithm.

where  $\epsilon$  is an arbitrarily small quantity chosen to avoid zero division. Let us now define the *observability index* as:

$$C_k = h_{s_k}/L_k. \quad (5)$$

This index represents the view angle of each peak from the observing location, whereby a larger value implies a higher probability of being a dominant peak. As an illustration for the searching mechanism, Fig. 2 shows a scenario where  $s_k$  and  $s_{k+3}$  are dominant peaks to be detected against the initial ones. In this scenario, the observing location is selected at  $L^*$  and the observability indices are calculated accordingly. The peak at  $s_k$  is firstly considered as a dominant peak as  $C_{s_k} > C_{s_{k+1}}$ . The following dominant peak is then determined at  $s_{k+3}$  because of its ability to block the sight from the observing location to the next peak, i.e.,  $C_{s_{k+3}} > C_{s_{k+4}}$ , while being not hidden from the view with respect to its previous peak, i.e.  $C_{s_{k+3}} > C_{s_{k+2}}$ .

To extend the concept wholly to other initial peaks, we first calculate a set of observing locations for each initial peak at  $s_k$  as follows:

$$\mathbf{L}_k = \mathbf{x}_k - (\mathbf{y}_k \circ \Delta \mathbf{x}_k) \oslash \Delta \mathbf{y}_k, \quad (6)$$

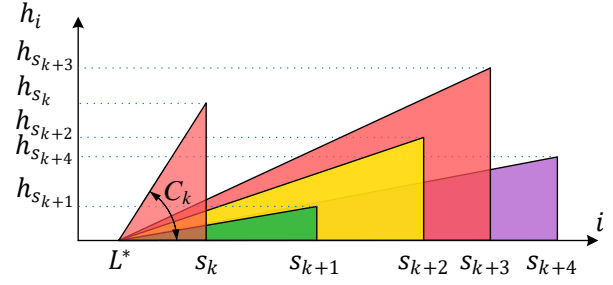


Fig. 2: Illustration of the peak searching mechanism.

where

$$\mathbf{x}_k = [s_k \ s_k \ \dots \ s_k], \quad (7)$$

$$\mathbf{y}_k = [h_{s_k} \ h_{s_k} \ \dots \ h_{s_k}], \quad (8)$$

$$\Delta \mathbf{x}_k = \mathbf{x}_k - [s_1 \ s_2 \ \dots \ s_{k-1}], \quad (9)$$

$$\Delta \mathbf{y}_k = \mathbf{y}_k - [h_{s_1} \ h_{s_2} \ \dots \ h_{s_{k-1}}], \quad (10)$$

in which  $\circ$  and  $\oslash$  are respectively the element-wise multiplication and division operators, with any zero element of  $\Delta \mathbf{y}_k$  being replaced by  $\epsilon$ . The best observing location,  $L^*$ , satisfying (3) for any peak  $s_k$  is then given by:

$$L^* = \min_k \{ \min_m \{ L_{k,m} \} \}, \quad (11)$$

where  $L_{k,m}$  is the  $m$ th element of  $\mathbf{L}_k$ . The observability indices for all initial peaks are then computed as:

$$\mathbf{C} = \mathbf{Y}_P \oslash (\mathbf{X}_P - \mathbf{L}^*), \quad (12)$$

where:

$$\mathbf{Y}_P = [h_{s_1} \ h_{s_2} \ \dots \ h_{s_P}], \quad (13)$$

$$\mathbf{X}_P = [s_1 \ s_2 \ \dots \ s_P], \quad (14)$$

$$\mathbf{L}^* = [L^* \ L^* \ \dots \ L^*]. \quad (15)$$

Finally, an initial peak  $k$  is determined as a dominant peak subject to the following condition:

$$C_k = \max\{C_{j+1}, C_{j+2}, \dots, C_{k+1}\}, \quad (16)$$

where  $j \geq 0$  and  $C_j$  is a previous dominant peak. The pseudo code for the peak searching algorithm is presented in Algorithm 1.

---

**Algorithm 1** Peak searching
 

---

```

1: for  $i \leftarrow 2, P$  do
2:    $\mathbf{L} \leftarrow \mathbf{x} - (\mathbf{y} \circ \Delta \mathbf{x}) \circ \Delta \mathbf{y}$ 
3:    $L_{tmp}(i-1) \leftarrow \min(\mathbf{L})$ 
4: end for
5:  $L^* \leftarrow \min(L_{tmp})$ 
6:  $\mathbf{C} \leftarrow \mathbf{Y}_P \circ (\mathbf{X}_P - \mathbf{L}^*)$ 
7: Find all local maxima of  $\mathbf{C}$ , save to  $\mathbf{v}$ 
8: if  $C(1) > C(2)$  then
9:    $\mathbf{v} \leftarrow [1 \ \mathbf{v}]$ 
10: end if
11: if  $C(end) > C(end-1)$  then
12:    $\mathbf{v} \leftarrow [\mathbf{v} \ \text{length}(\mathbf{C})]$ 
13: end if

```

---

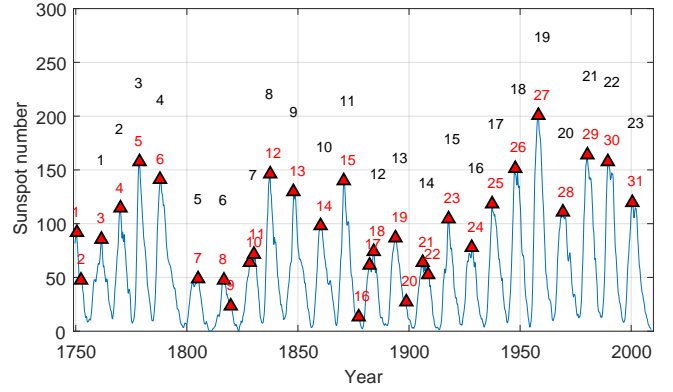
### C. Peak merging

While the peak searching algorithm can remove most non-dominant peaks, a false peak detection may still exist because of noise at lower frequencies. For instance, Fig. 3(a) shows the result of peak searching for the sunspot number dataset provided by the US National Oceanic and Atmospheric Administration (NOAA) [27]. Thirty-one peaks (red numbers) have been detected instead of 23 ground-truth ones due to incorrect reception of peaks 1, 2, 9, 10, 16, 17, 20, 22. A peak merging process is thus proposed to deal with this problem. By using our approach, a histogram can be segmented into a set of unimodal  $h_i$  on the interval  $[a_i, b_i]$  if there exists  $c_i \in [a_i, b_i]$  such that  $h_i$  follows the increasing hypothesis on  $[a_i, c_i]$  and the decreasing hypothesis on  $[c_i, b_i]$  [24].

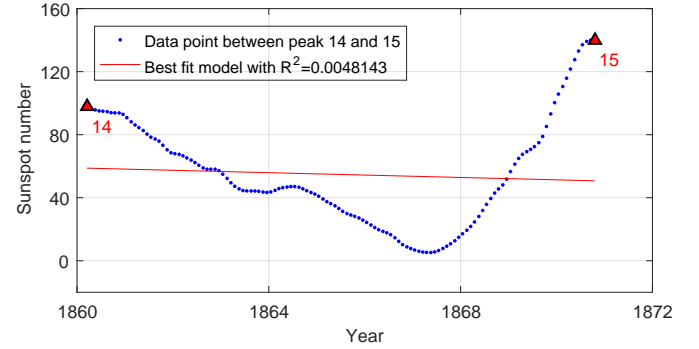
Let  $\mathbf{v}$  be the set of locations of dominant peaks and  $v_k$  be the  $k$ th element of  $\mathbf{v}$ . For each interval  $[v_k, v_{k+1}]$ , in which  $h_i$  has an average  $\bar{h}$ , a best fit line  $y_k = \alpha x_k + \beta$  is computed by using the least square regression. The coefficient of determination  $R_k^2$  of each model is then employed to estimate whether these peaks are lying on a same side of a unimodal mode. This coefficient is determined as:

$$R_k^2 = 1 - \frac{\sum_{i=v_k}^{v_{k+1}} (h_i - y_k)^2}{\sum_{i=v_k}^{v_{k+1}} (h_i - \bar{h})^2}. \quad (17)$$

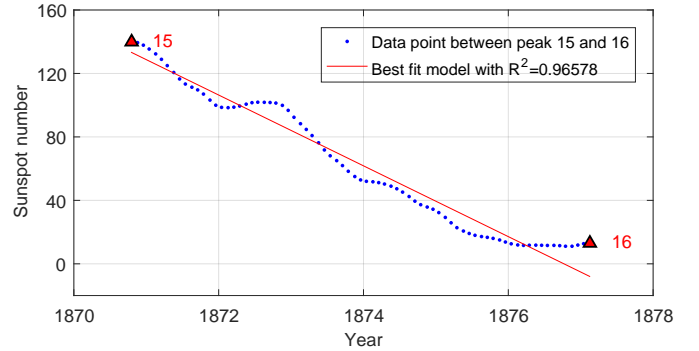
Two consecutive peaks  $k$  and  $k+1$  are then considered as lying on an increasing or decreasing side of a unimodal mode if the data in the interval  $[v_k, v_{k+1}]$  are well predicted by a linear model. An illustration is shown in Figs. 3(b) and (c). In Fig. 3(b), an R-squared of 0.48% is obtained for the linear model between peaks 14 and 15 as these peaks belong to two different modes.



(a)



(b)



(c)

Fig. 3: Illustration for the peak merging algorithm: (a) detected peaks after the searching phase, (b) best fit model for the interval between peak 14 and 15 and (c) best fit model for the interval between peak 15 and 16.

Conversely in Fig. 3(c), 96.6% in R-squared for the data points between peaks 15 and 16 is predicted by the linear model as these peaks belong to a decreasing side of mode. Consequently, a peak at  $s_k$ ,  $k \in \mathbf{v}$ , can be determined as false under the following conditions:

- $R_k^2$  is smaller than a pre-defined threshold, and
- its height,  $h_{s_k}$ , is smaller than either the height of the previous peak,  $h_{s_{k-1}}$ , or the next peak,  $h_{s_{k+1}}$ .

Algorithm 2 presents the pseudo code for the proposed peak merging process.

**Algorithm 2** Peak merging

---

```

1:  $cont \leftarrow 1$ 
2: while  $cont > 0$  do
    $j \leftarrow 0$ 
4:   del  $\leftarrow [ ]$ 
   Calculate R-squared for all intervals in  $\mathbf{v}$ 
6:   for  $i \leftarrow 2, (\text{length}(v) - 1)$  do
     if peak  $i$  is non-dominant then
8:        $j \leftarrow j + 1$ 
       del( $j$ )  $\leftarrow i$ 
10:    end if
   end for
12:   $cont \leftarrow \text{length}(\text{del})$ 
   if  $cont > 0$  then
14:     $\mathbf{v}(\text{del}) \leftarrow [ ]$  ▷ Remove insignificant peaks
   end if
16: end while

```

---

## III. RESULTS AND DISCUSSION

To evaluate performance of the proposed algorithms in retrieving information of local maxima for peak detection, histogram analysis, image segmentation and background removal, various experiments on sunspot data [27] [28], synthetic data, Berkeley dataset [29] and MSRA-B dataset [30] have been conducted and evaluated by using the F-measure as a compromise between recall and precision. Let  $tp$ ,  $tn$  be the correctly reported positive and negative results,  $fp$  and  $fn$  the falsely reported positive and negative results, the F-measure for binary classification is calculated as:

$$F = \frac{2 \times p \times r}{p + r}, \quad (18)$$

where  $p$  and  $r$  denote the precision and recall measure defined respectively as:

$$p = \frac{tp}{tp + fp}, \quad (19)$$

$$r = \frac{tp}{tp + fn}. \quad (20)$$

For multi-class classification, the macro-averaging technique [31] is employed to compute the evaluation metrics  $F_M$ ,  $p_M$ , and  $r_M$  as below:

$$p_M = \frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fp_i}}{l}, \quad (21)$$

$$r_M = \frac{\sum_{i=1}^l \frac{tp_i}{tp_i + fn_i}}{l}, \quad (22)$$

TABLE I: Detection results on NOAA sunspot number dataset

Method	$F$	$p$	$r$
SN	<b>0.9787</b>	0.9583	1
FTC	<b>0.9787</b>	0.9583	1
HTFCM	0.5227	0.3538	1
MATLAB	0.9388	0.8846	1

$$F_M = \frac{2 \times p_M \times r_M}{p_M + r_M}, \quad (23)$$

where  $tp_i$ ,  $tn_i$ ,  $fp_i$ ,  $fn_i$  are the assessment defined for each individual class  $C_i$  ( $i = 1, \dots, l$ ), correspondingly. For performance evaluation, the Histogram Thresholding Fuzzy C-means Hybrid (HTFCM) [22], Fine to Coarse (FTC) [24], local fuzzy thresholding (FTH) [20], Slope Difference Distribution (SDD) [9], [25], and the Superpixel-based Fast Fuzzy C-Means clustering (SF-FCM) [32] methods as well as the standard MATLAB function *findpeaks* are used to compare with our Summit Navigator (SN) method. For experiments on the background removal, two binary classification methods, the Otsu [8] and the Iterative Triclass Thresholding technique (ITTH) [33], are additionally employed for comparison.

## A. Experiment with sunspot number dataset

Two official datasets of sunspot numbers are first used in this experiment. The first one is compiled by NOAA [27]. The second one is obtained from the World Data Center - Sunspot Index and Long-term Solar Observations (SILSO) [28]. Each of them consists of more than 3000 data points.

1) *Results with the NOAA dataset:* The NOAA dataset is compiled with the sunspot numbers from 1749 to 2009 along with the dates when the maximum sunspot cycles are recorded. Those dates are employed as the ground truth to evaluate the performance of the above algorithms. Figure 4 shows the peaks detected together with their ground truth. It can be seen that although all 23 true peaks are extracted by all four methods, the false positives are different among them. Table I shows the quantitative results. Both our method and FTC have only one false positive prior to cycle 1 and thus score the highest F-measure and Precision. HTFCM incorrectly detected 42 peaks resulting in the worst F-measure and Precision which are 0.5227 and 0.3538, respectively. The MATLAB function *findpeaks* delivered a rather good result with 3 false positives resulting in the F-measure and Precision of 0.9388 and 0.8846,

TABLE II: Detection results on SILSO sunspot number dataset

Method	$F$	$p$	$r$
SN	<b>0.9796</b>	0.9600	1
FTC	<b>0.9796</b>	0.9600	1
HTFCM	0.4948	0.3288	1
MATLAB	0.9600	0.9231	1

respectively. The MATLAB function however requires meticulously manual tuning of its parameters. Without that, its F-measure and Precision drop to 0.2 and 0.1111 with 207 incorrect peaks.

2) *Results with the SILSO dataset:* The SILSO dataset consists of sunspot numbers from 1749 to 2017 with 24 maximum cycles recorded. Figure 5 shows the detected and ground-truth peaks. Table II presents the quantitative results. As all 24 true peaks are detected by the four methods, their Recall values are all 1. Our method and FTC however scored the highest F-measure and Precision with only one false positive recorded compared to 2 and 49 as in MATLAB function and HTFCM. Again, the rather good results of MATLAB function were obtained by limiting the minimal peak-to-peak distance to 7 years. Nevertheless, the experiments with sunspot numbers have proved that our algorithm outperformed the MATLAB function and HTFCM while having the same accuracy as FTC on large datasets. The results also indicate that HTFCM may not maintain its efficiency for peak detection and segmentation tasks as the data length and number of modes increase.

### B. Experiment with synthetic data

To further evaluate the robustness of the proposed method, synthetic images composed of three objects subject to increasing Gaussian noise have been generated. The images have different coefficients of signal to noise ratio (SNR) defined as:

$$SNR = 10 \log \left[ \frac{\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} I^2(x, y)}{\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (I(x, y) - I_n(x, y))^2} \right], \quad (24)$$

where  $I(x, y)$  and  $I_n(x, y)$  are the intensity at  $(x, y)$  of the original and noisy images and  $N_x$  and  $N_y$  are their width and height, were then used for segmentation. Figure 6(a) shows the original image while Fig. 6(c) depicts its histogram in which three objects 1, 2, and 3 and the background 4 are represented as four isolated modes.

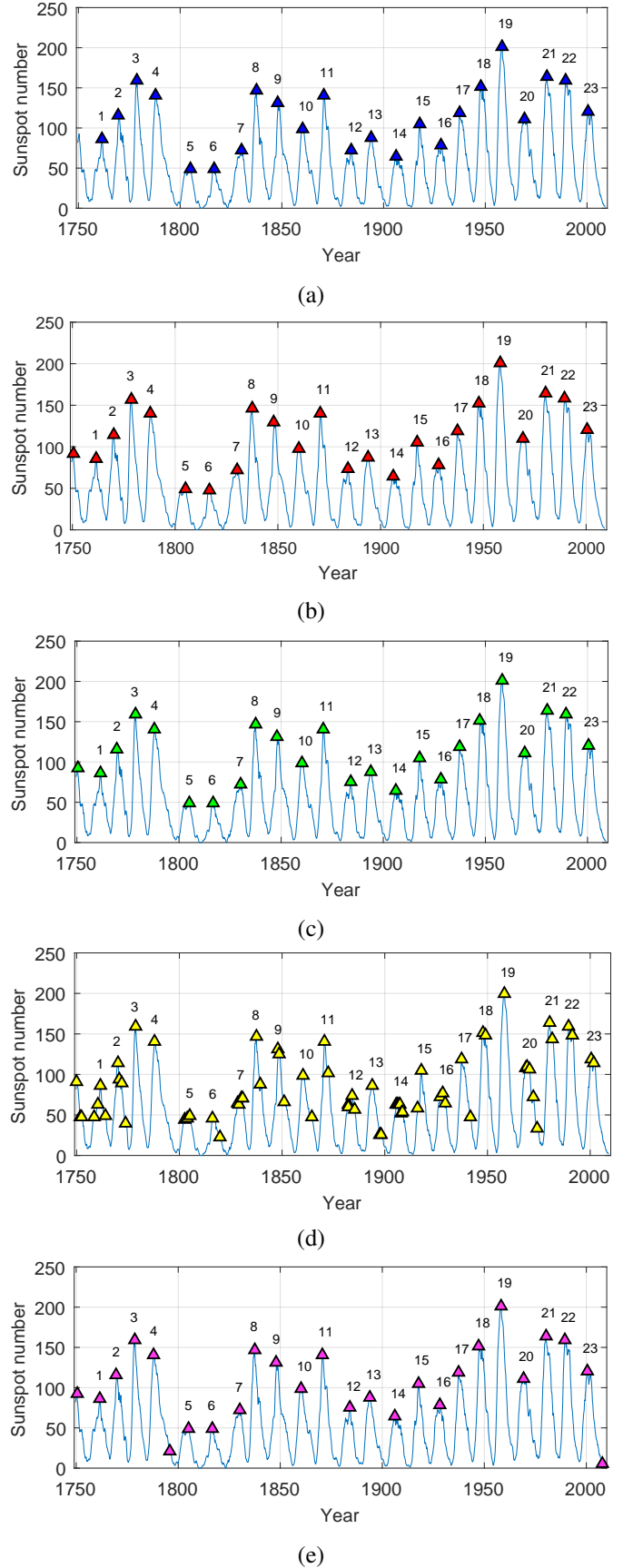


Fig. 4: Experiment with NOAA Sunspot cycle data: (a) Ground-truth values, peaks detected by (b) SN, (c) FTC, (d) HTFCM and (e) MATLAB.

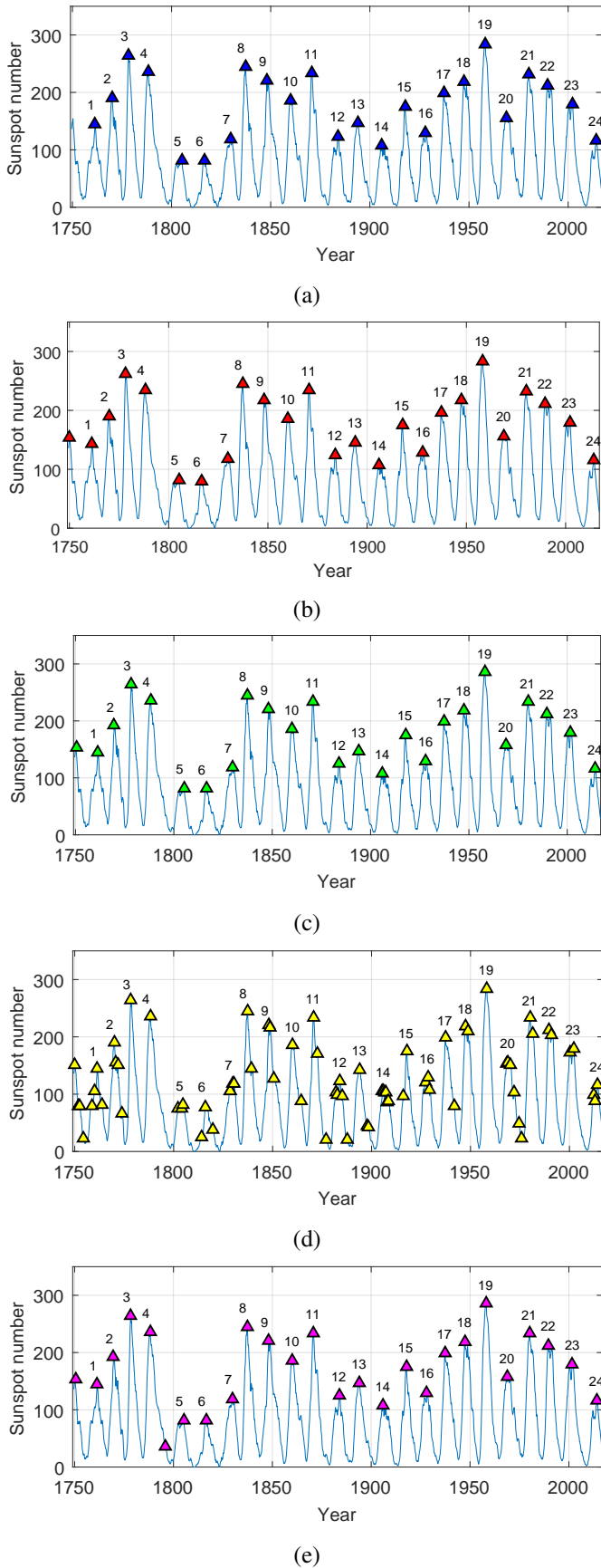


Fig. 5: Experiment with SILSO Sunspot cycle data: (a) Ground-truth values, peaks detected by (b) SN, (c) FTC, (d) HTFCM and (e) MATLAB.

The manually segmented image is shown in colors in Fig. 6(d), which is used as the ground truth to evaluate the segmentation performance of the participated methods.

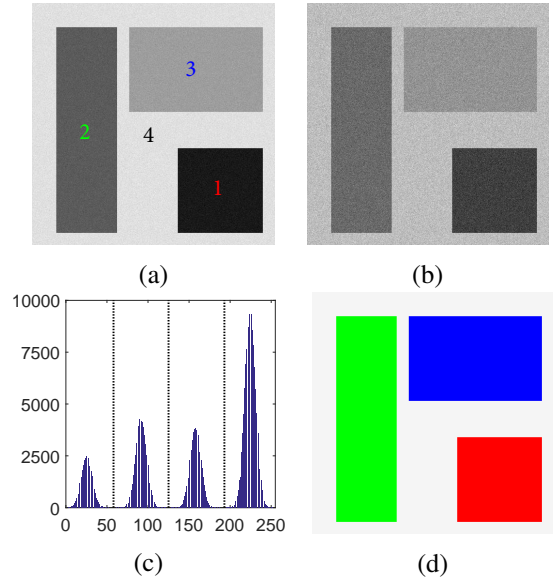


Fig. 6: Reference data for the synthetic test: (a) Original image, (b) noisy image at 10 dB, (c) histogram of the original image and (d) ground-truth image.

In the case of increasing Gaussian noise, the width of each mode increases leading to the overlapping between modes and thus decreasing the number of modes found. Figure 6 (b) shows the noisy image with SNR = 10 dB. The segmentation results of our method together with FTC, HTFCM and MATLAB are presented in Fig. 7(a)-(h). Table III presents the quantitative results. It can be seen in Fig. 7(b) that our method accurately detects three modes while FTC introduces an under segmentation with only two modes found therein. HTFCM and MATLAB on the other hand demonstrate an over segmentation with six modes detected as illustrated in Fig. 7(c) and (d) respectively. As a result, FTC is unable to separate objects 1 and 2 and objects 3 and 4 as depicted in Fig. 7(f) whereas HTFCM and MATLAB can separate four objects but also adding extra ones as observed in Fig. 7 (g) and (h). The over segmentation of MATLAB function is caused by its implementation which tends to overly detect peaks in histograms if no user-specific condition is given. In contrast, the problem of FTC is its sensitivity to noise, as indicated in [25].

Figure 8 presents the plot of F-measure, Precision and Recall obtained from another experiment in which noise level varies. It can be seen that all the four methods could segment well the images of SNR = 15 dB onwards. At lower SNR, our method however introduces better

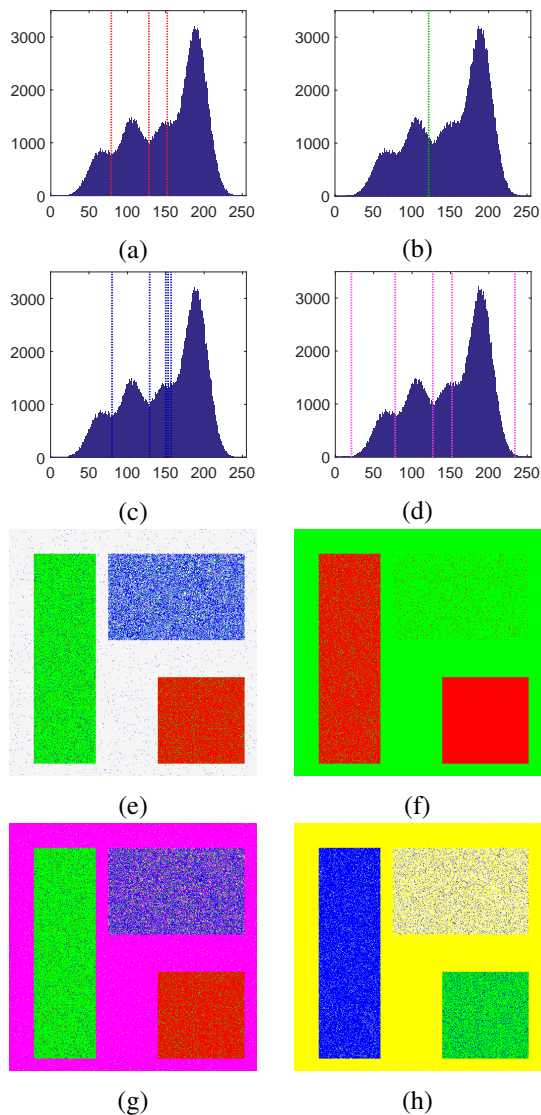


Fig. 7: Segmentation results on the noisy image with SNR = 10 dB:

Segmented histograms with (a) SN, (b) FTC, (c) HTFCM and (d) MATLAB, visualized results of (e) SN, (f) FTC, (g) HTFCM and (h) MATLAB.

TABLE III: Segmentation results on the noisy image with SNR = 10 dB

Method	$F$	$p$	$r$
SN	<b>0.8315</b>	0.8075	0.8570
FTC	NaN	0.2840	NaN
HTFCM	NaN	NaN	0.4438
MATLAB	NaN	NaN	0.2343

performance with  $F_1 > 0.8$  at 10 dB. This accuracy level is reached by using FTC and HTFCM at 11 dB, and MATLAB at 13 dB. The average accuracy of all these

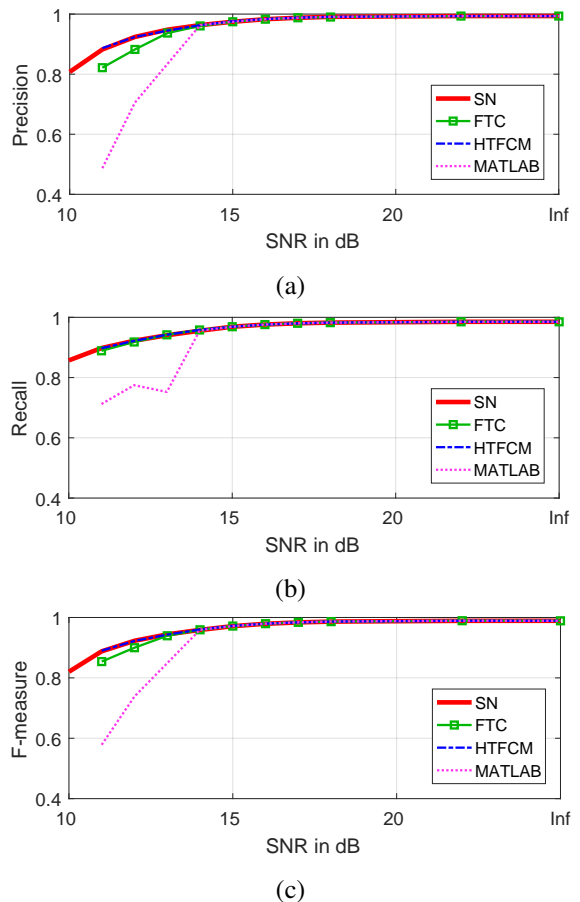


Fig. 8: Performance comparison in the presence of Gaussian noise: (a) Precision, (b) Recall and (c) F-measure.

methods is presented in Table IV, which confirms the effectiveness of our method compared to FTC, HTFCM and MATLAB.

TABLE IV: Average segmentation results on synthetic data

Method	$F$	$p$	$r$
SN	<b>0.9668</b>	0.9688	0.9636
FTC	0.9608	0.9591	0.9628
HTFCM	0.9665	0.9692	0.9638
MATLAB	0.9231	0.9149	0.9201

### C. Experiment with Berkeley dataset

To verify the effectiveness of the proposed algorithm with real life images, we evaluated its performance on the Berkeley dataset for two tasks, histogram analysis and image segmentation.



TABLE V: Comparison between SN and FTC in terms of peak detection on Berkeley images

Image	Ground-truth thresholds	SN				FTC			
		$F$	$p$	$r$	$U$	$F$	$p$	$r$	$U$
Animal 1	[85, 239]	<b>0.9121</b>	0.9984	0.8396	<b>0.9888</b>	0.8471	0.7492	0.9744	0.9846
Animal 2	[76]	0.9946	0.9996	0.9896	<b>0.9966</b>	0.9946	0.9996	0.9896	0.9966
Animal 3	[35, 166]	<b>0.9897</b>	0.9801	0.9994	<b>0.978</b>	0.9668	0.9374	0.9981	0.9763
Animal 4	[132]	<b>0.9815</b>	0.9678	0.9956	<b>0.9877</b>	0.9323	0.8852	0.9848	0.9808
Human 1	[27, 83, 158, 231]	<b>0.9102</b>	0.8729	0.9508	<b>0.9915</b>	0.7892	0.7307	0.8578	0.9878
Human 2	[109 228]	<b>0.9919</b>	0.9867	0.9972	<b>0.9758</b>	0.9578	0.9239	0.9942	0.9743
Nature 1	[54 180]	<b>0.9942</b>	1	0.9886	<b>0.9989</b>	0.9738	0.9528	0.9959	0.9986
Nature 2	[73 138 238]	0.9822	0.9803	0.984	<b>0.982</b>	NaN	0.2502	NaN	0.9779
Nature 3	[175]	<b>0.9964</b>	0.998	0.9948	0.9623	0.9803	0.9887	0.9719	<b>0.9658</b>
Nature 4	[142]	<b>0.9918</b>	0.9947	0.9890	<b>0.9819</b>	0.9842	0.9783	0.9901	0.9812

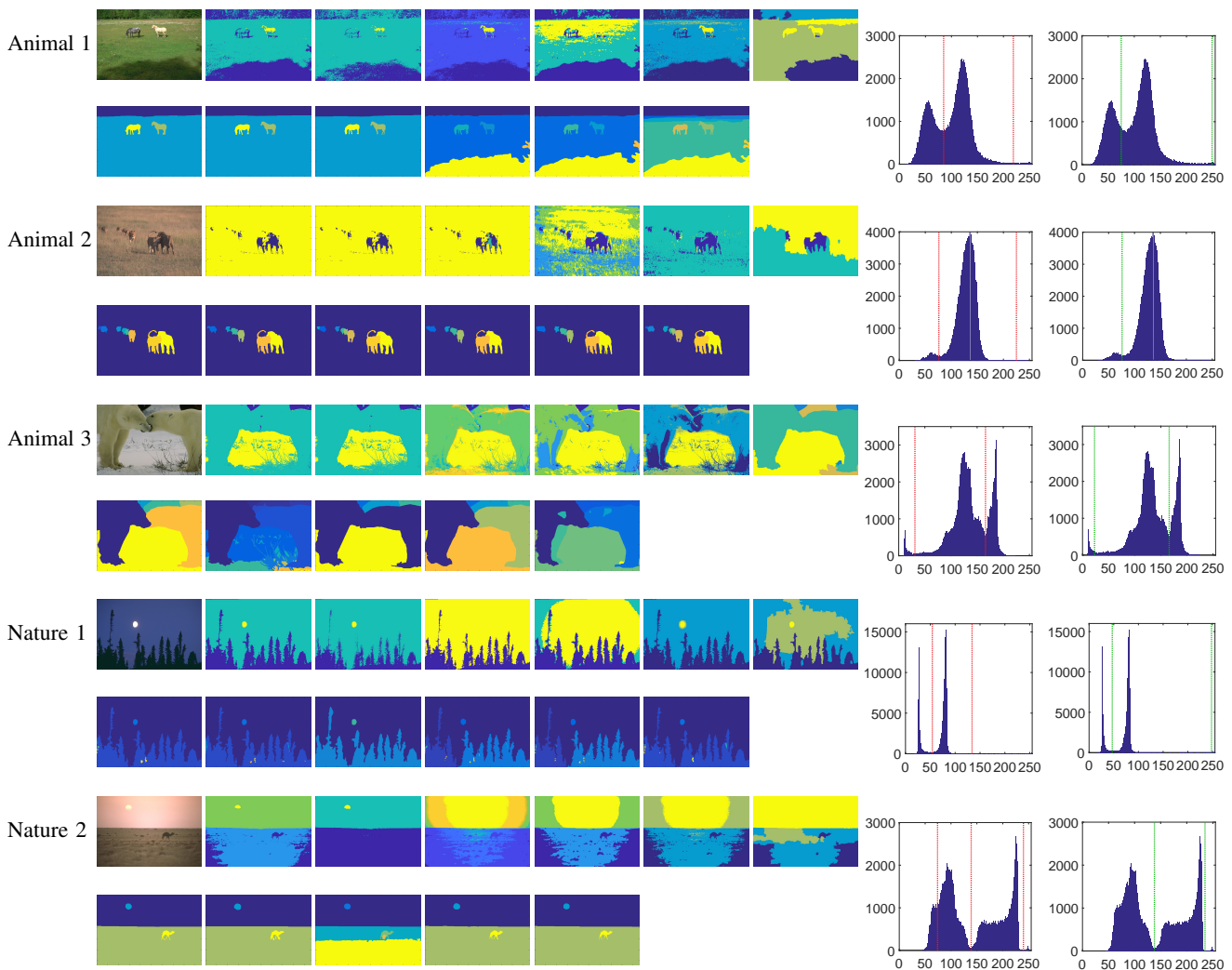


Fig. 9: Image segmentation results. First and second columns: image name and original image. Third to eighth columns: segmentation respectively by SN and FTC, HTFCM, FTH SDD and SFFCM. Last two columns: histogram thresholding by SN and FTC. Second row of each image: ground-truth segmentation of the Berkeley dataset.

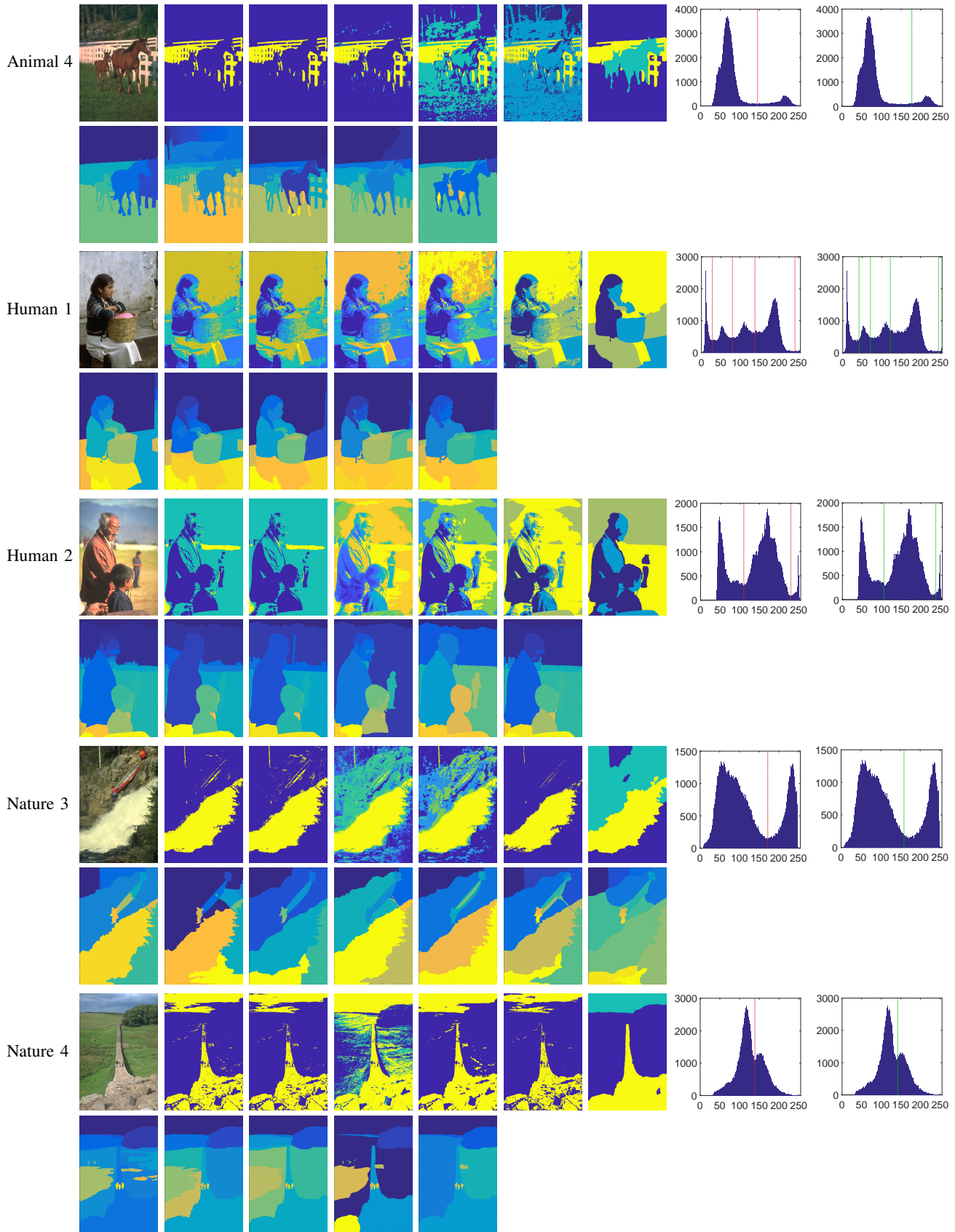


Fig. 10: Image segmentation results. First and second columns: image name and original image. Third to eighth columns: segmentation respectively by SN and FTC, HTFCM, FTH, SDD and SFFCM. Last two columns: histogram thresholding by SN and FTC. Second row of each image: ground-truth segmentations of the Berkeley dataset.

1) *Histogram analysis*: In this experiment, we selected from the Berkeley dataset a variety of benchmark images wherein their histograms can be distinctly distributed into different isolated modes. The selected images are first manually segmented so that the result can be employed as the ground truth for comparison. To ensure the accuracy of the manual segmentation, the local maxima thresholds are determined based on the peak distance and valley obviousness as described in [21]. We additionally employ the uniformity measure,  $U_I$ , for the sake of objectivity in evaluation as this metric is not dependent on the ground-truth information [34]. To compute  $U_I$ , the test image is first segmented by the thresholds determined as the minimum values between two adjacent dominant peaks. Let  $h_j(x, y)$  be the intensity value of each pixel  $(x, y)$  belonging to the segmented region  $R_j$ , and  $A_j$  be the number of pixels in that area. The mean  $\mu_j$  and variance  $\sigma_j^2$  for each segmented region are then defined as:

$$\mu_j = \sum_{(x,y) \in R_j} h_j(x, y) / A_j, \quad (25)$$

$$\sigma_j^2 = \sum_{(x,y) \in R_j} (h_j(x, y) - \mu_j)^2 / A_j. \quad (26)$$

The maximum variance  $\sigma_{max}^2$  is then determined based on the maximum,  $h_{max}(x, y)$ , and minimum,  $h_{min}(x, y)$ , intensities of the whole image:

$$\sigma_{max}^2 = (h_{max}(x, y) - h_{min}(x, y))^2 / 2. \quad (27)$$

Let  $N$  be the total number of pixels of image  $I$  and  $\mu_{jI}$  is the contribution of region  $R_j$  to the measure, which is given by:

$$\mu_{jI} = A_j \sigma_j^2 / N \sigma_{max}^2. \quad (28)$$

The region uniformity measure of an image is then obtained by:

$$U_I = 1 - \sum_{R_j \in I} \mu_{jI}. \quad (29)$$

A value of uniformity measure close to 1 represents a well-segmented image and vice versa. As the HTFCM and MATLAB always provide an over segmentation in all of the tested images, these two methods are omitted in this experiment. Figures 9 and 10 show the original images and their segmented results obtained by our method and FTC. It can be seen that both methods are quite identical in the number of modes selected. However, the locations of the thresholds estimated by each method are slightly different. This is explained in that each mode segmented by FTC is selected based on the unimodal hypothesis and hence a mode will be

bounded by an interval  $[a, b]$  if it follows the increasing hypothesis on  $[a, c]$  and the decreasing hypothesis on  $[c, b]$ , where  $c \in [a, b]$  is the peak location of that mode. This approach could lead to a wrong segmentation if two modes are either overlapped or located too far from each other. This can be seen in the quantitative results presented in Table V where our algorithm provides not only a better F-measure value in all test images but also a higher uniformity measure regardless of the manual segmentation.

2) *Image segmentation*: In image segmentation, when more regions are detected, the variance  $\sigma_j^2$  of each segmented region becomes smaller and so does the contribution of region  $R_j$  to the measure. Hence, the segmentation with more regions often lead to the uniformity closer to 1. For a better measure of the goodness in this case, we employ  $F'$  and  $Q$  [35] to penalize the over-segmentation for a large number of very small and non-homogeneous regions. Those indices are defined as follows:

$$F'(I) = \frac{\sqrt{\sum_{A=1}^M [\rho(A)]^{1+1/A}} \times \sum_{j=1}^R \frac{e_j^2}{\sqrt{A_j}}}{10000 \times N}, \quad (30)$$

$$Q(I) = \frac{\sqrt{R} \times \sum_{j=1}^R \left[ \frac{e_j^2}{1 + \log A_j} + \left( \frac{\rho(A_j)}{A_j} \right) \right]}{10000 \times N}, \quad (31)$$

where the normalization factor here is 10000 for a total of  $N$  pixels of the test image,  $R$  is the number of detected regions,  $e_i$  is the average squared color error of the  $i$ th region,  $\rho(A_j)$  is the number of regions that has an area of exactly  $A_j$ , and  $M$  denotes the largest region in the segmented image. A lower value of  $F'$  and  $Q$  represents a well segmentation and vice versa.

The segmentation results of SN, FTC, HTFCM, FTH, SDD and SFFCM are shown in Fig. 9 and 10 together with the ground-truth segmentations from the Berkeley dataset. It is noted that although the segmentation results of all methods are close to the ground truth, those of HTFCM and FTH are more noisy due to their high number of detected regions. Figure 11 presents the quantitative results. It can be seen that SN and FTC outperform HTFCM, FTH, SDD and SFFCM with the smallest  $F'$  and  $Q$ . Moreover, SN provides a better region uniformity than FTC for the same number of detected regions.

In another comparison, the performance of all algorithms is evaluated on the whole Berkeley dataset. Out of 500 images, only segmentation results on 408 images are considered by ignoring the images for which one of

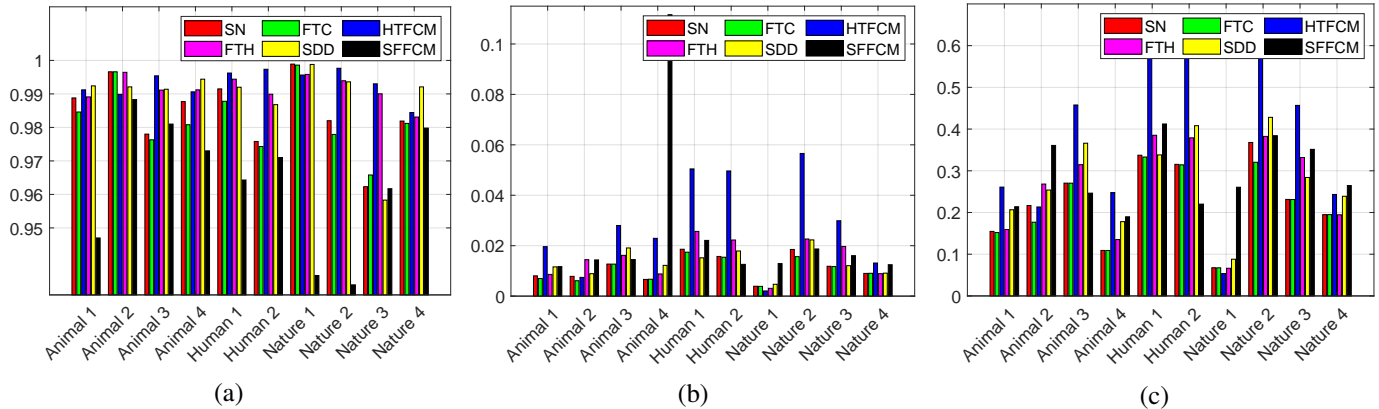


Fig. 11: Quantitative results of SN, FTC, HTFCM, FTH, SDD and SFFCM on test images from the Berkeley dataset: (a)  $U$ , (b)  $F'$  and (c)  $Q$ .

the above methods fails to return a segmentation. The average values of  $U$ ,  $F'$  and  $Q$  are presented in Table VI. Therein, SN shows the best homogeneity, but is only better than FTC and SFFCM in terms of region uniformity. This is consistent with the previous comparison and implies that the uniformity and homogeneity can be diverged in image segmentation.

TABLE VI: Average results of  $U$ ,  $F'$  and  $Q$  on the Berkeley dataset

Method	$U$	$F'$	$Q$
SN	0.9721	<b>0.0129</b>	<b>0.2320</b>
FTC	0.9718	0.0130	0.2331
HTFCM	<b>0.9944</b>	0.0395	0.4360
FTH	0.9886	0.0192	0.2885
SDD	0.9902	0.0149	0.2903
SFFCM	0.9566	0.0161	0.3079

#### D. Application in background removal

The practicability and superiority of our approach are now illustrated in the background removal scenario. Its performance will be compared with the FTC, HTFCM, FTH, SDD, SFFCM, Otsu and ITTH methods on images of the MSRA-B dataset [30]. The test images are chosen to have high contrast, a large number of background pixels, and different levels of background complexity. The ground truth for comparison are taken from [36]. As Otsu and ITTH are bi-level thresholding whereas the others are multiple thresholding, we use the micro-averaging measure  $F_\mu$  [31] for a fair evaluation. This measure treats bigger classes more favourably and can be calculated as follows:

$$F_\mu = \frac{2p_\mu r_\mu}{p_\mu + r_\mu}, \quad (32)$$

where

$$p_\mu = \frac{\sum_{i=1}^R tp_i}{\sum_{i=1}^R (tp_i + fp_i)}, \quad (33)$$

$$r_\mu = \frac{\sum_{i=1}^R tp_i}{\sum_{i=1}^R (tp_i + fn_i)}. \quad (34)$$

Figure 12 presents the background removal results. Although all methods are able to extract foreground objects from the background, our method is more accurate in determining background pixels. While FTC, HTFCM and FTH overly segment both the object and background due to a large number of modes/clusters detected, by comparing with two well-known binarizing algorithms, Otsu and ITTH, our SN on the opposite returns better results, even for the cases of more classes involved. Those results can be confirmed via  $F_\mu$  values as reported in Table VII.

#### E. Discussion

Through extensive experiments with both synthetic and real datasets, our method is confirmed to be more accurate and robust than other available methods. The rationale stems from our two-step searching and merging mechanism in which dominant peaks are detected with the best observing location and false positives are recursively filtered with the use of unimodal and linear fitting. In addition, our method can be adapted for different applications by tuning the threshold for the coefficient of determination. Although a fixed threshold  $T = 0.45$  has been used for all the tests taken in this paper, this value can be tuned for a desired segmentation based on the smoothness and the size of the input data.

In another note, Table VIII presents the average processing time of the proposed method together with

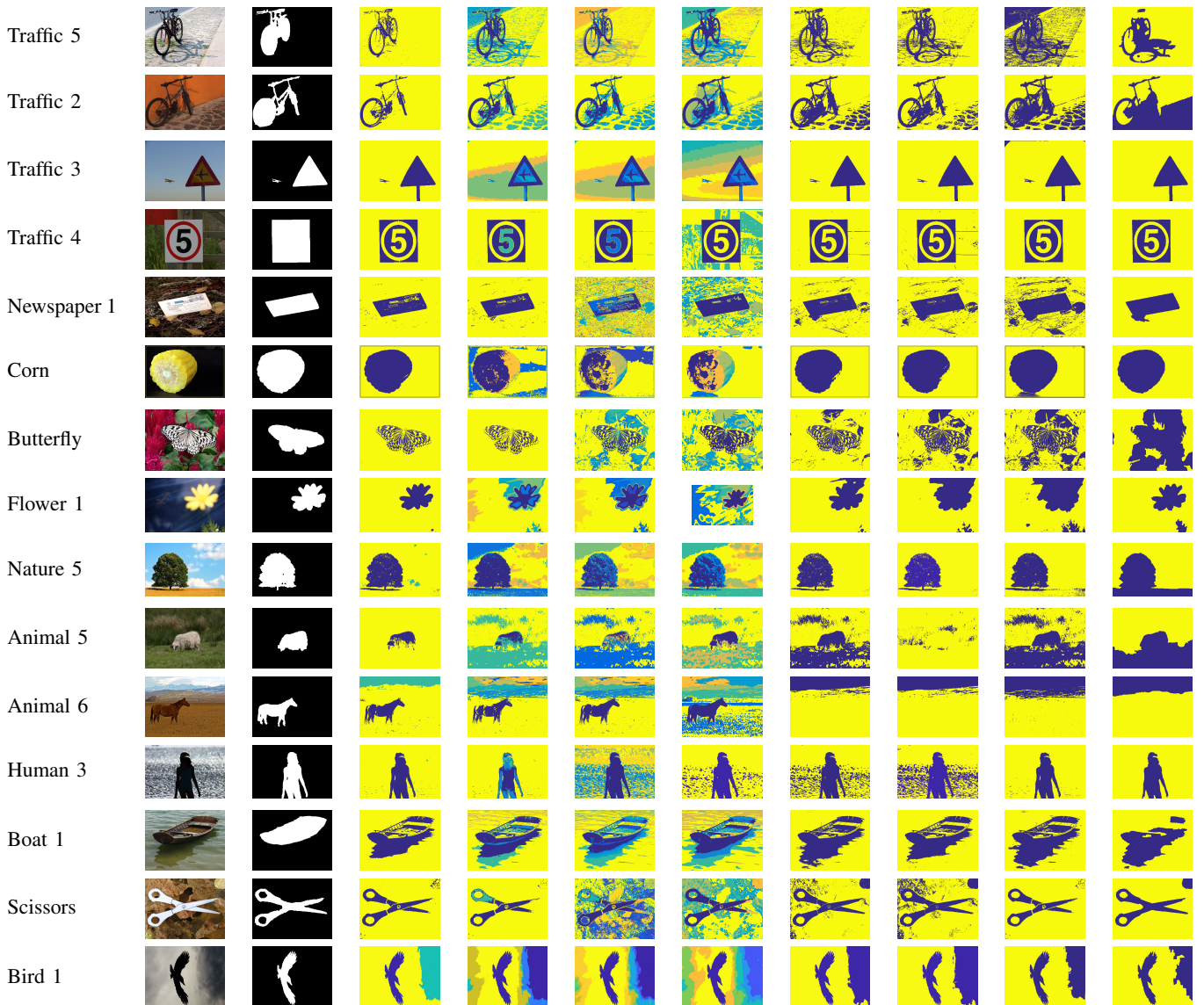


Fig. 12: Background removal results on MSRA-B dataset. First column: image name. Second to third columns: original and ground-truth image. Fourth to eleventh columns: image segmentation respectively by SN, FTC, HTFCM, FTH, Otsu, ITTH, SDD and SFFCM.

FTC, HTFCM, MATLAB, FTH, Otsu, ITTH, SDD and SFFCM on 8-bit images and sunspot data. Those experiments were executed by using MATLAB R2016a on an Intel(R) Core(TM) i5-5300U CPU @2.30 GHz with 64 bit Windows 7. It can be seen that MATLAB and HTFCM provide a short processing time due to their simplicity while Otsu remains the most computationally-effective for binarization tasks. For image segmentation, our method performs faster than FTC, FTH, SDD and SFFCM. However, for peak detection from a time series data, our method becomes less time-efficient as the size of the data increases. A parallelized instance of our algorithm may overcome this problem, which will be our future work.

#### IV. CONCLUSION

This paper has presented a novel method, the Summit Navigator, to automatically and accurately identify true peaks from multi-modal gray-scale histograms of images without *a priori* knowledge of the number of modes or distance between modes in processing. The proposed algorithms adopt the tactic in mountain exploration together with utilizing the density and intensity of the initial peaks to estimate the best observing location. Based on this location, the observability index can be computed to determine dominant peaks. To deal with the false positive class, a merging mechanism is also introduced whereby the concepts of unimodal in histograms and linear model fitting has been combined effectively. The

TABLE VII: Comparison between SN, FTC, HTFCM, FTH, Otsu, SDD and SFFCM on MSRA-B images

Image	SN	FTC	HTFCM	FTH	Otsu	ITTH	SDD	SFFCM
Traffic 1	<b>0.8951</b>	0.6564	0.5858	0.4860	0.8535	0.8367	0.6470	0.8089
Traffic 2	<b>0.8969</b>	0.7491	0.7218	0.5933	0.7861	0.8341	0.7535	0.5671
Traffic 3	0.9881	0.4575	0.4583	0.2401	0.9878	<b>0.9882</b>	0.9804	0.9846
Traffic 4	0.8763	0.8743	0.8711	0.5984	0.8796	0.8755	0.8759	<b>0.8886</b>
Newspaper 1	0.9817	0.9804	0.4296	0.6313	0.9404	0.9222	0.8348	<b>0.9834</b>
Corn	0.9673	0.6855	0.5771	0.7301	0.9729	0.9677	0.9523	<b>0.9950</b>
Butterfly	<b>0.8941</b>	0.8800	0.6641	0.5798	0.8550	0.7769	0.7113	0.5936
Flower 1	<b>0.9844</b>	0.7451	0.7198	0.4451	0.9482	0.8793	0.8262	0.9726
Nature 5	0.9656	0.5463	0.4437	0.4463	<b>0.9752</b>	<b>0.9752</b>	0.9650	0.8353
Animal 5	<b>0.9599</b>	0.5435	0.5369	0.5438	0.6307	0.8792	0.5756	0.6624
Animal 6	<b>0.8274</b>	0.6422	0.6435	0.4097	0.6652	0.6691	0.5716	0.6234
Human 3	0.9875	0.9054	0.3382	0.9208	0.8544	0.7840	0.9906	<b>0.9931</b>
Boat 1	<b>0.8040</b>	0.6265	0.5130	0.4762	0.7741	0.7988	0.7588	0.7349
Scissors	<b>0.9649</b>	0.9337	0.3247	0.4930	0.9119	0.8669	0.9642	0.9640
Bird 1	0.7937	0.4824	0.4816	0.3583	0.7617	0.7413	0.8257	<b>0.8470</b>

TABLE VIII: Comparison of processing time in seconds

Method	SN	FTC	HTFCM	MATLAB	FTH	Otsu	ITTH	SDD	SFFCM
Sunspot data	1.4993	1.0505	0.1178	<b>0.0203</b>	N/A	N/A	N/A	N/A	N/A
8-bit image	0.1567	0.2952	0.0901	0.0172	3.7839	<b>0.0005</b>	0.0092	0.1655	0.7445

method has been successfully demonstrated on reputable datasets and indicated outperformance over the existing approaches, such as FTC, HTFCM, MATLAB, Otsu, FTH, ITTH, SDD and SFFCM in terms of accuracy and robustness. This can lead to promising applications in vision-based diagnostics, particularly in robotics and automation systems for surface inspection.

#### REFERENCES

- [1] E. Malamas, E. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image Vis. Comput.*, vol. 21, no. 2, pp. 171–188, 2003.
- [2] Y. Yang, Z.-J. Zha, M. Gao, and Z. He, "A robust vision inspection system for detecting surface defects of film capacitors," *Signal Process.*, vol. 124, pp. 54–62, 2016.
- [3] M. Phung, C. Quach, T. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection," *Autom. Construct.*, vol. 81, pp. 25–33, 2017.
- [4] N. M. Kwok, Q. P. Ha, and G. Fang, "Effect of color space on color image segmentation," in *Proc. 2nd Int. Congress Image Signal Process.*, Oct 2009, pp. 1–5.
- [5] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.
- [6] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognit.*, vol. 26, no. 9, pp. 1277 – 1294, 1993.
- [7] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Comput. Vis. Image Understand.*, vol. 110, no. 2, pp. 260–280, 2008.
- [8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 11, no. 285–296, pp. 23–27, 1975.
- [9] Z. Wang, "An efficient and robust method for automatically identifying the left ventricular boundary in cine magnetic resonance images," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 536–542, April 2016.
- [10] M. Jian and C. Jung, "Interactive image segmentation using adaptive constraint propagation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1301–1311, 2016.
- [11] N. Mitianoudis and N. Papamarkos, "Document image binarization using local features and gaussian mixture modeling," *Image Vis. Comput.*, vol. 38, pp. 33 – 51, 2015.
- [12] B. Yuan and M. Liu, "Power histogram for circle detection on images," *Pattern Recognit.*, vol. 48, no. 10, pp. 3268 – 3280, 2015.
- [13] A. Dirami, K. Hammouche, M. Diaf, and P. Siarry, "Fast multi-level thresholding for image segmentation through a multiphase level set method," *Signal Process.*, vol. 93, no. 1, pp. 139 – 153, 2013.
- [14] J. Zhao and L. Itti, "Classifying time series using local descriptors with hybrid sampling," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 623–637, 2016.
- [15] B. N. Saha and N. Ray, "Image thresholding by variational minimax optimization," *Pattern Recognit.*, vol. 42, pp. 843–856, 2009.
- [16] S. Patra, R. Gautam, and A. Singla, "A novel context sensitive multilevel thresholding for image segmentation," *Appl. Soft Comput.*, vol. 23, pp. 122 – 127, 2014.
- [17] N. M. Kwok, H. Shi, Q. Ha, G. Fang, S. Chen, and X. Jia, "Simultaneous image color correction and enhancement using particle swarm optimization," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2356–2371, 2013.
- [18] S. Sarkar, S. Das, and S. S. Chaudhuri, "Multi-level thresh-

olding with a decomposition-based multi-objective evolutionary algorithm for segmenting natural and medical images,” *Appl. Soft Comput.*, vol. 50, pp. 142 – 157, 2017.

- [19] F. Zhao, J. Fan, and H. Liu, “Optimal-selection-based suppressed fuzzy c-means clustering algorithm with self-tuning non local spatial information for image segmentation,” *Expert Syst. Applic.*, vol. 41, no. 9, pp. 4083 – 4093, 2014.
- [20] S. Aja-Fernandez, A. H. Curiale, and G. Vegas-Sánchez-Ferrero, “A local fuzzy thresholding methodology for multiregion image segmentation,” *Knowl. Based Syst.*, vol. 83, pp. 1 – 12, 2015.
- [21] H.-D. Cheng and Y. Sun, “A hierarchical approach to color image segmentation using homogeneity,” *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2071–2082, 2000.
- [22] K. S. Tan and N. A. M. Isa, “Color image segmentation using histogram thresholding fuzzy c-means hybrid approach,” *Pattern Recognit.*, vol. 44, no. 1, pp. 1 – 15, 2011.
- [23] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [24] J. Delon, A. Desolneux, J. L. Lisani, and A. B. Petro, “A nonparametric approach for histogram segmentation,” *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 253–261, 2007.
- [25] Z. Wang, J. Xiong, Y. Yang, and H. Li, “A flexible and robust threshold selection method,” *IEEE Trans. Circuits Syst. Video Technol.*
- [26] T. H. Dinh, Q. P. Ha, and H. M. La, “Computer vision-based method for concrete crack detection,” in *Proc. 14th Int. Conf. Control Automat. Robot. Vis. (ICARCV)*, Nov. 2016, pp. 1–6.
- [27] U. N. Oceanic and A. Administration. Solar-terrestrial physics (stp). [Online]. Available: <https://www.ngdc.noaa.gov/stp/space-weather/>
- [28] SILSO World Data Center, “The International Sunspot Number,” *International Sunspot Number Monthly Bulletin and online catalogue*.
- [29] 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,” in *Proc. IEEE Intl Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
- [30] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum, “Learning to detect a salient object,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb 2011.
- [31] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427 – 437, 2009.
- [32] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A. K. Nandi, “Superpixel-based fast fuzzy c-means clustering for color image segmentation,” *IEEE Trans. Fuzzy Syst.*, pp. 1–1, 2018.
- [33] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu, “A new iterative triclass thresholding technique in image segmentation,” *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1038–1046, 2014.
- [34] M. D. Levine and A. M. Nazif, “Dynamic measurement of computer generated image segmentations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 2, pp. 155–164, 1985.
- [35] M. Borsotti, P. Campadelli, and R. Schettini, “Quantitative evaluation of color image segmentation results,” *Pattern Recognit. Lett.*, vol. 19, no. 8, pp. 741 – 747, 1998.
- [36] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, “Salient object detection: A discriminative regional feature integration approach,” in *Proc. IEEE Conf. CVPR*, pp. 2083–2090.



**Tran Hiep Dinh** received the MSc degree in mechatronics from the Leibniz University Hannover, Germany in 2010. He is currently pursuing the PhD degree in engineering at the University of Technology Sydney, Australia.

His research interests include image processing, robotics, and machine learning.



**Manh Duong Phung** received the B.Sc. and Ph.D. degrees from Vietnam National University, Hanoi, Vietnam in 2005 and 2015, respectively.

He is currently an Assistant Professor with the VNU University of Engineering and Technology, Hanoi, Vietnam. He conducted research projects on Eye tracking with NTT Cyber Solution Laboratory, Japan (2007), Telehealth with Mechatronics and Automation Laboratory of National University of Singapore (2009), 3-D hand tracking with Samsung Vietnam Mobile R & D Centre (2013), Point Cloud Processing in Construction with SIAMA Pty Ltd, Australia (2015), and Robotics and Automation in Construction with Department of Defence, Australia (2017). He has authored and coauthored a number of research papers. His research interests include automation in construction, 3-D reconstruction, image processing, and mobile robot localization and mapping.

Dr. Phung was the recipient of several research awards including the 2016 Endeavour Research Fellowship of the Australian Government.



**Quang Phuc Ha** received the B.E. degree from Ho Chi Minh City University of Technology, Vietnam in Electrical Engineering in 1983 and the PhD degrees from Moscow Power Engineering Institute, Russia, in Complex Systems and Control in 1993, and the University of Tasmania, Australia, in Intelligent Systems in 1997.

He is currently an Associate Professor with the School of Electrical and Data Engineering of the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His research interests include automation, robotics and control systems with a focus on construction automation.

Dr. Ha has been on the Board of Directors of the International Association of Automation and Robotics in Construction since 2007, and a Senior Member of the IEEE since 2013. He was Conference Chair/Co-Chair of several international conferences on automation and intelligent systems. He was an Associate Editor of the IEEE Transactions on Automation Science and Engineering (2009-2013) and is currently on the editorial board of Automation in Construction and Mathematical Problems in Engineering. Dr. Ha was the recipient of a number of best paper awards from the IEEE, IAARC and Engineers Australia, including the Sir George Julius Medal in 2015.