

Optical Engineering

[SPIDigitalLibrary.org/oe](https://spiedigitallibrary.org/oe)

Randomized SUSAN edge detector

Zhi-Guo Qu
Ping Wang
Ying-Hui Gao
Peng Wang



Randomized SUSAN edge detector

Zhi-Guo Qu, Ping Wang, Ying-Hui Gao, and Peng Wang

National University of Defense Technology,
ATR Key Laboratory, Changsha,
Hunan 410073, China
E-mail: green20001@sina.com

Abstract. A speed up technique for the SUSAN edge detector based on random sampling is proposed. Instead of sliding the mask pixel by pixel on an image as the SUSAN edge detector does, the proposed scheme places the mask randomly on pixels to find edges in the image; we hereby name it randomized SUSAN edge detector (R-SUSAN). Specifically, the R-SUSAN edge detector adopts three approaches in the framework of random sampling to accelerate a SUSAN edge detector: procedure integration of response computation and nonmaxima suppression, reduction of unnecessary processing for obvious nonedge pixels, and early termination. Experimental results demonstrate the effectiveness of the proposed method. © 2011 Society of Photo-Optical Instrumentation Engineers (SPIE). [DOI: 10.1117/1.3647520]

Subject terms: edge detection; SUSAN edge detector; random sampling.

Paper 110577LRR received May 27, 2011; revised manuscript received Aug. 30, 2011; accepted for publication Sep. 19, 2011; published online Nov. 3, 2011.

1 Introduction

The SUSAN edge detector is a popular edge detection technique for its good performance in localization of edges, connectivity at junctions, and noise suppression effect.¹ Many authors have presented various improved algorithms of SUSAN edge detectors^{2,3} from different perspectives. To the best of the authors' knowledge, however, no work on the acceleration of a SUSAN edge detector has been presented in the literature. Although acceleration algorithms for bilateral filter^{4,5} may be applied to the SUSAN edge detector analogously because of their similarity, the speed-up is only obvious for large kernel size which is not the case for the SUSAN edge detector. In this letter, we propose an acceleration algorithm for the SUSAN edge detector based on random sampling, an idea which has been successfully applied in several image processing algorithms.^{6,7} Instead of sliding the mask pixel by pixel on an image as the SUSAN edge detector does, the proposed scheme places the mask randomly on pixels to find edges in the image; we hereby name it the randomized SUSAN edge detector (R-SUSAN).

2 SUSAN Edge Detector

The SUSAN edge detector is implemented by sliding a circular mask with radius of 3.4 pixels pixel by pixel through an image. For each pixel in the raster scan order, an intensity threshold t defines the range of intensities for pixels around it in the circular mask which form the so-called SUSAN area:

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0), \quad (1)$$

$$c(\vec{r}, \vec{r}_0) = e^{-((I(\vec{r}) - I(\vec{r}_0))/t)^6}, \quad (2)$$

where \vec{r}_0 and \vec{r} are the positions of the nucleus and the surrounding pixel compared to it; $I(\vec{r}_0)$ and $I(\vec{r})$ are the respective intensities of these pixels. The initial edge response is given by:

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & \text{if } n(\vec{r}_0) < g \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where the geometrical threshold g is set to $0.75 N_{\max}$ for the SUSAN edge detector and $N_{\max} = 37$ pixels is the number of pixels of the mask. For convenience of discussion, we name the response computation for each pixel given by Eqs. (1)–(3) initial response calculation (IRC), which includes N_{\max} additions, $N_{\max} - 1$ exponential functions, and one comparison operation. In the above implementation of the SUSAN edge detector, redundant computation exists which makes it somewhat inefficient and affects its speed. Next, we will analyze the origin of redundancy and introduce the R-SUSAN edge detector which can decrease the redundancies to accelerate the SUSAN edge detector.

3 Randomized SUSAN Edge Detector

The R-SUSAN edge detector is implemented by placing the circular mask randomly on pixels of an image and taking different actions according to its position to improve efficiency.

3.1 Random Placing of the Mask

Randomly placing the circular mask on a pixel $\vec{r}_0 = (i, j)$, the SUSAN area $n(i, j)$ is first computed using Eqs. (1) and (2). Based on $n(i, j)$, three different cases corresponding to where the mask is placed are considered: edge region, homogeneous region, and near-edge region, as shown in Fig. 1. Accordingly, different acting rules are taken as follows.

3.1.1 Case 1

If $n(i, j) < g$, then the nucleus is in the edge region, see Fig. 1(a). In this case, we slide the mask along the edge until it reaches the ends. For each point in the scanning order, neighborhood pixels in the direction perpendicular to the edge (pixels in the dashed rectangle) are checked to get their IRC, and the one with the highest IRC is chosen as the candidate edge point while others are suppressed. After this edge is detected, the mask is then placed randomly on another unprocessed pixel.

3.1.2 Case 2

If $n(i, j) = N_{\max}$, then the nucleus is situated in homogeneous region, see Fig. 1(b). In this case, neither the nucleus nor the surrounding pixels in the circular mask are edge pixels. Therefore, the IRC for the surrounding pixels in the mask are redundant and can be skipped. However, for avoidance of a saw-tooth effect on the edges, we only disable the pixels in the dashed square instead of the whole mask and thus save IRC for them, as shown in Fig. 1(b). Since no edge information is provided in this case, the mask is then placed on another unprocessed pixel randomly.

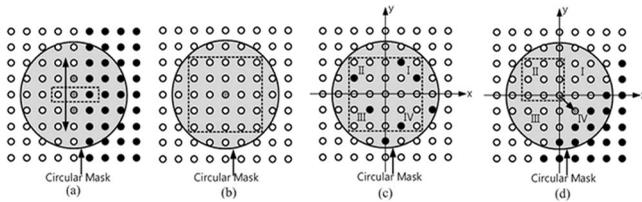


Fig. 1 Different positions the mask is placed. (a) Edge region; (b) homogeneous region; (c) homogeneous region corrupted by noise, and (d) near-edge region.

3.1.3 Case 3

If $g \leq n(i, j) < N_{max}$, then the nucleus may lie in a homogeneous region corrupted by noise [Fig. 1(c)] or near-edge region [Fig. 1(d)]. To differentiate the two situations from each other, we partition the circular mask into four quadrants and compute their USANs, and then find the maximum value n_{max} and the minimum value n_{min} among the four USANs. The nucleus lies in the near-edge region if the following condition is satisfied:

$$n_{max} - n_{min} \geq \frac{N_{max} - n(i, j)}{2} \tag{4}$$

Otherwise, we think the nucleus lies in the homogeneous region corrupted by noise and processes it as in Case 2. For the situation that the nucleus is situated in the near-edge region, we disable the pixels in the dashed square of the quadrant whose SUSAN is equal to n_{max} to save IRC for these pixels, and then slide the mask by one pixel toward the quadrant whose SUSAN equals n_{min} where it is nearer to the edge [Fig. 1(d)].

Generally, the process of random sampling should be repeated until all the pixels in the image are processed (including the disabled pixels hereafter). However, since edges usually cover up a small portion of an image and due to the strategy we adopt to find edges, we can terminate the process earlier without accessing all the pixels in the image. In Sec. 3.2, we develop an early termination condition for the algorithm to further save computation cost.

3.2 Early Termination Condition

The total number of pixels processed, q , will be sufficient if the probability of sampling all edges present in the image at least once is high. We can write down this probability if we have some idea of the number and length of the edges in the

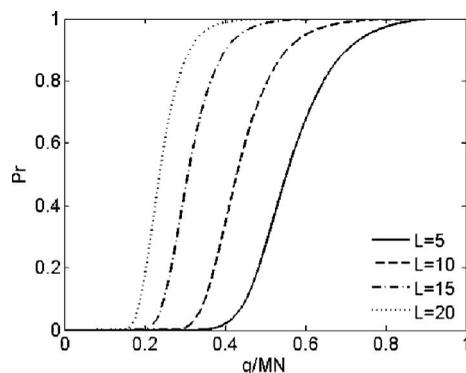


Fig. 2 Probability of detecting all edges after q pixels are processed.

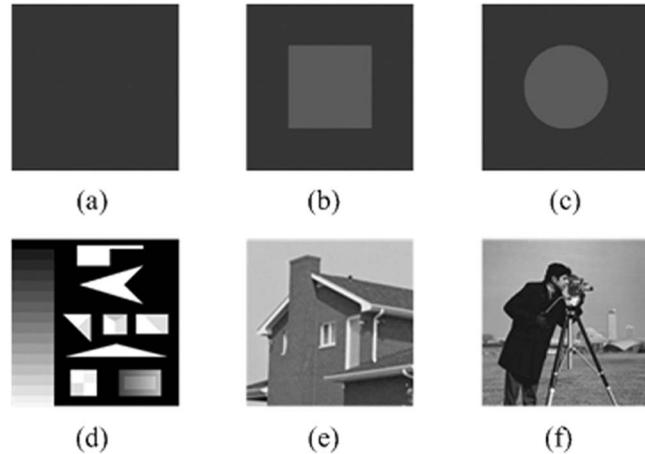


Fig. 3 Test Images: (a) Image No. 1; (b) Image No. 2; (c) Image No. 3; (d) Image No. 4; (e) Image No. 5; and (f) Image No. 6.

image. Consider K edges of the same length L distributed randomly in an image of size $M \times N$, accounting for 0.1% of the total number of pixels in the image, i.e., $KL = 0.1MN$. Let p_0 be the probability of placing the mask on an edge, then $p_0 = L/(MN)$. Given that we think an edge is detected once a point on it is sampled, the probability of having detected all edges after q pixels are processed is:

$$Pr = 1 - \sum_{i=1}^K (-1)^{i-1} \binom{K}{i} (1 - ip_0)^q, \tag{5}$$

where i stands for the i 'th edge. This probability can be plotted as a function of q and can then be used to gauge which value would constitute a reasonable threshold, as shown in Fig. 2. We can see that, the longer the edge, the earlier the probability reaches 1, which means that the lower the termination threshold can be set. The value $q_{TH} = 0.8MN$, which guarantees the probability is 1 for edges longer than 5 and

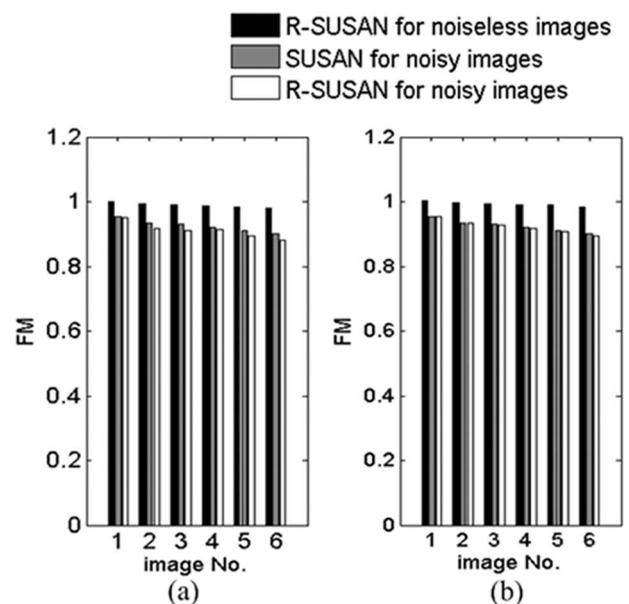


Fig. 4 Detection performance comparison using FM. (a) $q_{TH} = 0.8MN$. (b) $q_{TH} = MN$.

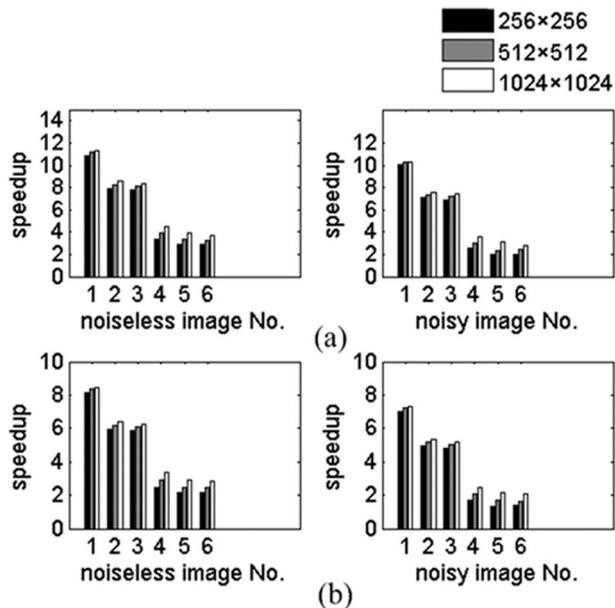


Fig. 5 Average speedup of R-SUSAN over SUSAN. (a) $q_{TH} = 0.8MN$. (b) $q_{TH} = MN$.

almost 0.9753 for edges with length of 5, would be a reasonable threshold. In fact, too short edges make little sense and will be removed eventually in the final edge map.

4 Experimental Results

The images used for the performance evaluation of the proposed method were gray scale images of various sizes, such as 256×256 , 512×512 , and 1024×1024 , and were numbered from 1 to 6 according to their complexity, as shown in Fig. 3. The Gaussian noise was also added to test their performance in noisy environment. Parameters setup, noise level, and post-processing procedures were the same for both detectors: $t = 10$ for synthetic images and $t = 20$ for natural images. The early termination threshold for R-SUSAN was set to $q_{TH} = 0.8MN$ and $q_{TH} = MN$, corresponding to cases with and without early termination, respectively. The performance of R-SUSAN and SUSAN is compared in terms of figure of merit (FM):⁸

$$F = \frac{1}{\max\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}, \quad (6)$$

where I_I and I_A are the number of ideal and actual edge points, $d(i)$ is the pixel miss distance of the i 'th edge detected, and α is a scaling constant chosen to be $\alpha = 1/9$.

The detection performance comparison of the two detectors using FM for all six images are given in Fig. 4(a) for $q_{TH} = 0.8MN$ and Fig. 4(b) for $q_{TH} = MN$, where the detection result of the SUSAN edge detector for noiseless image is regarded as ideal edge map and detection results in other circumstances are considered as actual edge points. The two algorithms programmed in VC++ language were executed 50 times on a 2.5-GHz PC with 1024 Mbytes of memory and their average run time were measured. The average speedup of R-SUSAN over SUSAN detector is shown in Fig. 5(a) for $q_{TH} = 0.8MN$ and Fig. 5(b) for $q_{TH} = MN$, respectively.

From the results, we can see that the R-SUSAN edge detector accelerates the SUSAN edge detector effectively while achieving similar detection performance. The speedup is more obvious for simple images with a large portion of homogeneous regions (Image Nos. 1–3) than that for complex images with less homogeneous regions (Image Nos. 4–6). This is due to a different amount of IRC savings: for simple images with more homogeneous regions, more IRCs are saved by the pixel-disabling procedure during the random sampling process; while for complex images with fewer homogeneous regions, the amount of IRC savings is less. This also explains why the speedup is better in a noiseless environment than that in a noisy environment. As for the early termination threshold q_{TH} , comparing the FM in Figs. 4(a) and 4(b) and the speedup in Figs. 5(a) and 5(b), we can see that $q_{TH} = MN$ does not bring significant improvement in detection performance but leads to an obvious decrease in speedup. Therefore, it is unnecessary to access all pixels in the framework of random sampling and an early termination strategy can further save computational cost without spoiling its detection performance much.

5 Conclusions

A random sampling scheme for fast implementation of SUSAN edge detector named R-SUSAN was presented in this work. In the framework of placing the mask randomly, the R-SUSAN edge detector increases the computation efficiency of SUSAN edge detector via approaches such as procedure integration of IRC computation and nonmaxima suppression, reduction of unnecessary IRC for obvious nonedge pixels, and early termination strategy. It was shown experimentally that the proposed R-SUSAN edge detector successfully reduce the time requirement of the SUSAN edge detector while keeping its detection performance almost unspoiled.

Acknowledgments

The authors are grateful to the anonymous reviewers for their valuable comments.

References

1. S. Smith and J. Brady "SUSAN—a new approach to low-level image processing." *Int. J. Comput. Vis.* **23**(1), 45–78 (1997).
2. M. M. Perez and T. J. Dennis, "An adaptive implementation of the SUSAN method for image edge and feature detection." *IEEE International Conference on Image Processing (ICIP)*, Vol. 2, pp. 394–397 (1997).
3. E. Rafajłowicz, "SUSAN edge detector reinterpreted, simplified and modified." *International Workshop on Multidimensional (nD) Systems*, pp. 69–74, IEEE, Piscataway, NJ (2007).
4. W. Yu, F. Franchetti, J. C. Hoe, Y.-J. Chang, and T. Chen, "Fast bilateral filtering by adapting block size." *IEEE 17th International Conference on Image Processing (ICIP 2010)*, pp. 3281–3284 (2010).
5. M. M. Bronstein, "Lazy sliding window implementation of the bilateral filter on parallel architectures." *IEEE Trans. Image Process.* **20**(6), 1751–1756 (2011).
6. L. Xu and E. Oja, "Randomized Hough transform." *Encyclopedia of Artificial Intelligence 2009*, pp. 1343–1350, IGI Global (2009).
7. K.-L. Chung, Y.-H. Huang, J.-P. Wang, T.-C. Chang, and H.-Y. M. Liao, "Fast randomized algorithm for center-detection." *Pattern Recogn.* **43**, 2659–2665 (2010).
8. I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement thresholding edge detectors." *Proc. IEEE* **67**(5), 753–763 (1979).