Mersin Photogrammetry Journal

https://dergipark.org.tr/en/pub/mephoj

e-ISSN 2687-654X

Connected pixels-based image smoothing filter

Erkan Besdok *[1], **Pinar Civicioglu**[2]

[1]Erciyes University, Department of Geomatics Engineering, Türkiye
[2]Erciyes University, Department of Aircraft Electrics and Electronics, Türkiye

Keywords
Connected Pixels
Image Smoothing
Bezier Search Optimization Algorithm

Research Article
DOI: 10.53093/mephoj.1279877

Received:09.04.2023
Revised: 05.05.2023
Accepted:08.05.2023
Published:27.05.2023

Abstract

Digital image processing heavily relies on the connectivity of pixels, as it is a vital component for accurate object identification and analysis within an image. Grouping together pixels with similar features such as colour and intensity, allows for the formation of meaningful patterns or objects, which is essential for object recognition and segmentation. This approach is particularly valuable in photogrammetric imaging, video surveillance, deep learning as it facilitates the isolation of regions of interest and object tracking. Image smoothing is also a crucial aspect in enhancing visual quality by reducing noise and enhancing details, especially in applications such as aerial mapping, medical imaging, video compression, image resizing and computer vision. The absence of connected pixels and image smoothing would make image processing tasks more challenging and less reliable, making them fundamental to digital image processing and critical to various applications in diverse fields. This paper introduces a novel image smoothing filter called Connected Pixels Based Image Smoothing Filter (CPF), which is based on gray connected pixels. The success of the CPF was compared to that of the Non-Local Means Filter (NLMF) in terms of Structural Similarity Index (SSIM) for the same Mean Squared Error (MSE). The experimental results showed that CPF has a better ability to preserve image details compared to NLMF.

1. Introduction

Pixel connectivity refers to the relationship between neighboring pixels in a digital image. It is a fundamental concept in image processing and computer vision, as it plays a crucial role in many image analysis tasks such as edge detection, segmentation, and object recognition [1-5]. In general, pixel connectivity refers to the notion of how pixels are connected or related to each other. There are different ways to define pixel connectivity, depending on the context and the specific requirements of the image processing task at hand [6-9]. One of the most common definitions of pixel connectivity is based on the notion of adjacency, which refers to whether two pixels are neighbors or not. In a 2D image, two pixels are considered adjacent if they are located next to each other horizontally, vertically, or diagonally [10-12]. In other words, two pixels are adjacent if they share a common edge or corner. Pixel connectivity is often quantified using a connectivity matrix or a connectivity graph, which represents the relationships between adjacent pixels in the image. A connectivity matrix is a binary matrix that encodes the adjacency relationships between pixels, where a value of 1 indicates that two pixels are adjacent, and a value of 0 indicates that they are not. A connectivity graph, on the other hand, is a graph-theoretic representation of the connectivity matrix, where each pixel is represented as a vertex, and the adjacency relationships between pixels are represented as edges. The definition of pixel connectivity can also be extended to 3D images, where it refers to the relationships between neighboring voxels (3D pixels). In a 3D image, two voxels are considered adjacent if they are located next to each other in any of the three spatial dimensions (x, y, z). Pixel connectivity is closely related to other concepts in image processing, such as image topology and morphological operations. In image topology, the connectivity of a set of pixels is defined as the number of connected components in the set. Morphological operations, such as dilation and erosion,

* **Corresponding Author**

*(ebesdok@erciyes.edu.tr) ORCID ID 0000-0001-9309-375X
(civici@erciyes.edu.tr) ORCID ID 0000-0003-1850-8489

Cite this article

Beşdok, E., & Civicioğlu, P. (2023). Connected pixels-based image smoothing filter. Mersin Photogrammetry Journal, 5(1), 24-31

rely on the notion of pixel connectivity to manipulate the shapes and boundaries of objects in an image. In conclusion, pixel connectivity is a fundamental concept in image processing and computer vision that refers to the relationships between neighboring pixels in a digital image. It plays a crucial role in many image analysis tasks, and it is quantified using connectivity matrices or graphs [13-16]. The definition of pixel connectivity can be extended to 3D images, and it is closely related to other concepts in image processing such as image topology and morphological operations. Algorithms developed to find connected pixels can be slow for several reasons. Large image sizes require significant processing time, as the algorithm must check each pixel and its neighbors for connectivity. Complex algorithms also take longer to execute, while outdated hardware may not handle the algorithm efficiently. The data structure used to store the image and pixels can impact the algorithm's speed if not optimized, and inefficient implementation can slow processing time. Optimization of these factors may result in faster processing times for algorithms designed to find connected pixels [17-22].

The process of identifying and labeling subsets of connected components based on a given heuristic is referred to as connected-component labeling, also known as connected-component analysis, region labeling, blob extraction, region extraction, or blob discovery. This algorithmic application is based on graph theory and aims to assign unique labels to each subset of connected components. It is important to note that connected-component labeling should not be mistaken for segmentation.

There are several algorithms [19, 23-31] that can be used to find connected gray pixels in an image. Here are a few:

1. Flood fill algorithm: This is a simple algorithm that starts from a given seed pixel and fills all adjacent pixels of the same gray value. It continues until all connected pixels are filled.
2. Depth-first search (DFS): This is a graph traversal algorithm that can be used to find all connected gray pixels in an image. It starts at a given seed pixel and explores as far as possible along each branch before backtracking.
3. Breadth-first search (BFS): This is another graph traversal algorithm that can be used to find connected gray pixels. It explores all the neighboring pixels at the current level before moving on to the next level.
4. Connected component labelling (CCL): This is a more complex algorithm that assigns a unique label to each connected component of gray pixels in an image. It can be used to identify and separate multiple connected regions with different gray values [28, 32-35].

In this paper, we introduce the Connected Pixels Based Image Smoothing Filter (CPF), which was developed using gray connected pixels. CPF can produce a smoothed image while preserving the detail information in the original image relatively well. It is a simple and effective image smoothing filter that can be easily adapted to different applications due to its simple structure.

The rest of this paper is organized as follows: In Section 2, Connected Pixels Based Image Smoothing Filter (CPF) is mentioned. In Section 3, Experiments are presented. In Section 4, Results and Conclusions are given.

2. Connected pixels-based image smoothing filter (CPF)

This section introduces the Connected Pixels Based Image Smoothing Filter (CPF). CPF is a non-recursive, non-iterative, non-linear image-smoothing filter. CPF uses the median value of the gray-connected pixels to the central pixel of the sliding window to generate the smoothed value of the central pixel. CPF has two parameters: '*win*' and '*T*', which denote the size of the sliding window and the threshold value, respectively. The result obtained is not affected by the order in which neighbors are selected, as long as they meet the threshold requirement. The algorithmic structure of the CPF is shown in Figure 1.

```
Input: img, win, T
Output: q
w0 = (win − 1)/2
img := padarray(img)
[M, N, Dim] ← size(img)
for i=1:Dim do
    BandImg = img(:, :, i)
    for j=w0+1:M-w0 do
        for k=w0+1:N-w0 do
            Temp = BandImg(j − w0 : j + w0, k − w0 : k + w0)
            centerpixel = Temp(w0 + 1, w0 + 1)
            BinaryImg = |Temp − centerpixel)| ≤ T
            Generate the pixel labels for BinaryImg by using Eq.s 1-2
            Get the pixels, u, that have same label with center pixel of Temp
            q(j, k, i) := median(u)
        end
    end
end
q = q(w0 + 1 : end − w0, w0 + 1 : end − w0, :)
```

Figure 1. Pseudo-Code of the Connected Pixels Based Image Smoothing Filter (CPF)

Connected component labeling is a crucial process in computer vision, particularly for object recognition. It entails identifying regions in a binary image where pixels are connected. To perform this operation on a 2D image stored in a 2D array, we scan each pixel one by one and assign a label based on the labels of its neighbors or a new label if all its neighbors are background pixels. Suppose a pixel with a value of 0 in array I represents a background pixel, while a pixel with a value of 1 represents an object pixel. To store the labels, we use an array L of the same size as I. In our implementation, we use a single array to hold both I and L. The goal is to fill the array L with integer labels so that object pixels neighboring each other have the same label. Although we use integer labels for simplicity, other label types can also be used. In this paper, we utilized the mask topology presented in Figure 2 of Rosenfeld for an 8-pixel neighborhood based connected pixel labelling [36]. The pixel in the scan mask is represented by letters a, b, c, d, and e, and we use the same letters instead of their (i, j) coordinates.

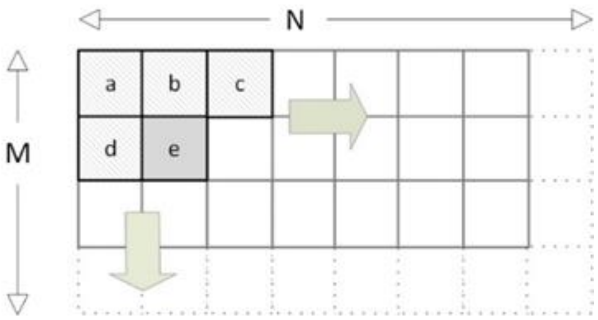


Figure 2. The mask topology of Rosenfeld for 8 pixels neighborhood.

Figure 3 provides an illustrative example of 8-connected component labeling using the Rosenfeld algorithm.

L[e] represents the label of the current pixel being scanned, and I[b] represents the pixel value of the neighbor directly above e in the vertical direction. We start with an integer variable l initialized to 1. During the first scan, we provisionally assign a label to e as follows: L[e] is assigned 0 if I[e] = 0. If a, b, c, and d in the scan mask are all background pixels, we assign it a new label l, and we increment l by 1. Otherwise, we assign it the minimum of the provisional labels already assigned to the scan mask. This process ensures that all pixels in the same object are assigned the same provisional label. In subsequent scans, we change the labels of object pixels to the minimum labels of their neighboring object pixels. Specifically, if L[e] is greater than the minimum of L[b], L[c], and L[d], we update L[e] to the minimum value. We repeat this process until there are no more changes to the labels. In conclusion, connected component labeling is a fundamental operation in computer vision that involves identifying regions in a binary image where pixels are connected. By scanning each pixel and assigning labels based on the labels of its neighbors, we can accurately identify these regions. This technique can be applied to a wide range of applications, including object recognition and image segmentation.

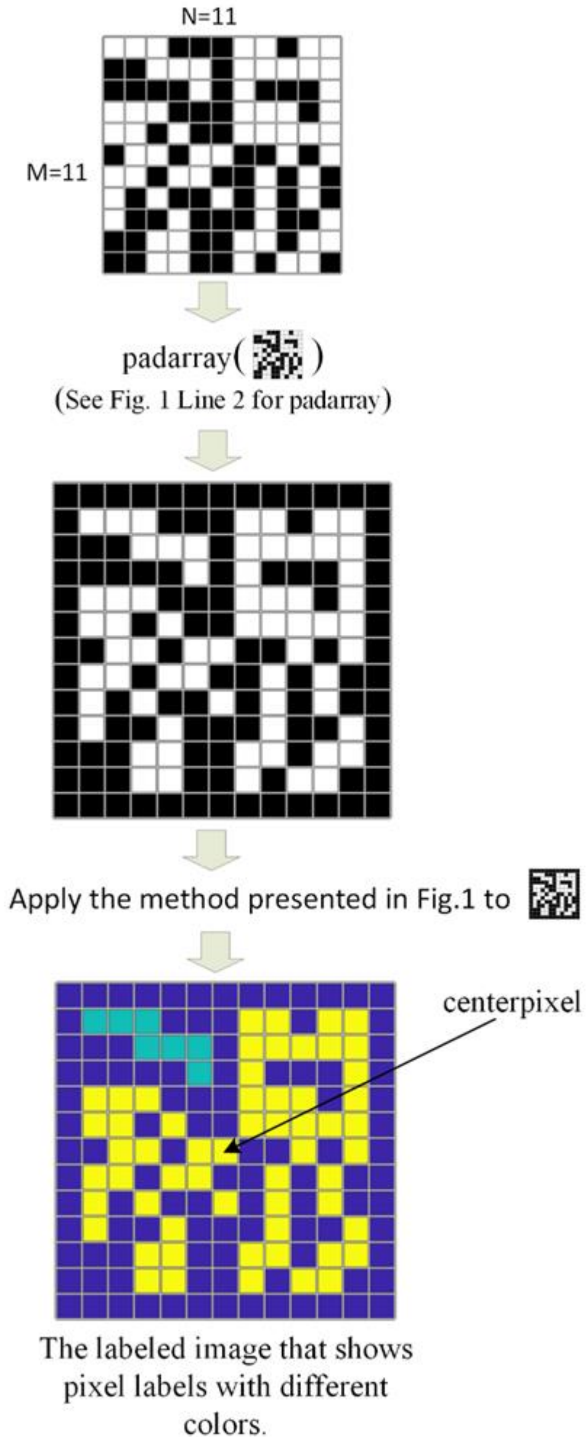


Figure 3. An example of 8-connected component labeling using the Rosenfeld algorithm

Expressing the assignment of a provisional label for "e" during the initial scan can be described by using Equations 1-2.

$$L[e] \leftarrow \begin{cases} 0 & I[e] = 0 \\ l, (l \leftarrow l + 1) & \forall i \in (a, b, c, d),\ \ I[i] = 0 \\ \underbrace{min}_{i \in (a,b,c,d) | I(i)=1} (L(i)) & otherwise \end{cases} \tag{1}$$

Where

$$L[e] \leftarrow \underbrace{min}_{i \in (a,b,c,d)\,|\,I[i]=1} \left(L(i)\right)\ ,\quad If\ \ I[e] = 1\ , \\ and\ \ \exists i \in (a, b, c, d)\ such\ that\ \ I[i] = 1 \tag{2}$$

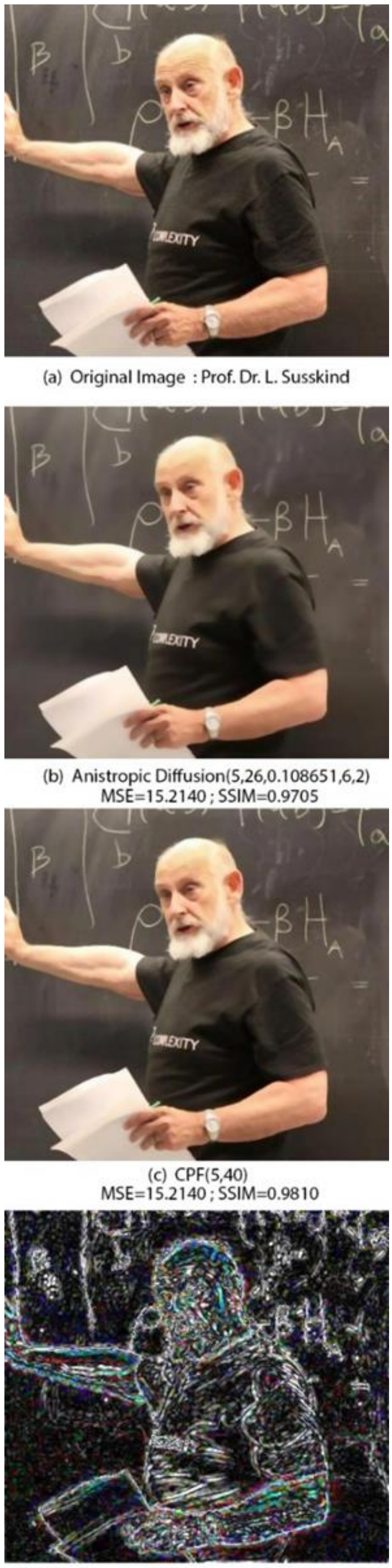
3. Experiments

In this section, we test the success of the Connected Pixels Image Smoother Filter (CPF) in smoothing images using Test Images. The Test Images used are 8-bit RGB with dimensions of 512x512 pixels. We compare the image smoothing capability of CPF to that of the Non-Local Mean Filter [37, 38] for an unbiased review. The smoothing parameter of NLMF is optimized using BSD [39] so that the MSE values calculated between the original image and the smoothed images calculated by CPF and NLMF are the same. Bernstein-Search Differential Evolution (BSD) is a nature-inspired optimization algorithm that is widely used for solving complex problems in various fields such as engineering, finance, and artificial intelligence. BSD combines the concepts of Bernstein polynomials and Differential Evolution to improve the search process for finding the optimal solution. BSD uses a population-based approach where each individual represents a candidate solution, and the optimization process involves generating new candidate solutions through mutation and crossover operations. BSD has shown significant improvements in convergence speed and solution quality compared to traditional optimization algorithms. Overall, BSD is a powerful optimization technique that can be used to solve a wide range of problems efficiently.

The Anisotropic Diffusion filter [40] is an image processing technique used to enhance images by reducing noise while preserving edges. It works by diffusing the image while applying less diffusion to edges and more diffusion to flat regions. This helps to remove noise without blurring edges, resulting in a sharper and clearer image. The degree of diffusion is controlled by a parameter called the diffusion coefficient, which can be adjusted to achieve the desired level of smoothing. Anisotropic diffusion is commonly used in computer vision, medical imaging, and other applications where image clarity is important. The image smoothing performance of CPF and Anistropic Diffusion is tested using the Susskind Test Image and the results are shown in Figure 4.

CPF produced better results than Anistropic Diffusion even in edge regions.

The Non-Local Mean Filter (NLMF) is a popular image denoising technique used to reduce noise while preserving edges and details. Unlike traditional filters that operate on a pixel-by-pixel basis, NLMF considers the entire image when calculating the similarity between patches of pixels. By comparing similar patches, the filter can identify and remove noise while preserving image structure. The algorithm is computationally expensive, but several optimizations have been proposed to improve efficiency. NLMF has been successfully applied to a variety of image processing tasks, including video denoising, image deblurring, and super-resolution. NLMF has three parameters: 'win1', 'win2', and 'S'. 'win1' and 'win2' denote the match and search window sizes, respectively, while 'S' represents the smoothing parameter. Within this section, the notation NLMF (win1, win2, S) will be employed to denote the Non-Local Means Filter along with its corresponding parameters. Similarly, the notation CPF (win, T) will be utilized to represent the Constrained Parametric Filters along with its associated parameters. In the experiments carried out, Mean Squared Error (MSE) and Structural Similarity Index (SSIM) were used to measure the similarities of NLMF and CPF with the original image.



(a) Original Image : Prof. Dr. L. Susskind

(b) Anistropic Diffusion(5,26,0.108651,6,2)
MSE=15.2140 ; SSIM=0.9705

(c) CPF(5,40)
MSE=15.2140 ; SSIM=0.9810

(d) Pseudo-scaled difference Image for (b)-(c).

Figure 4. Comparison of CPF and Anistropic Diffusion for the same MSE values on Susskind Test Image (Please use zoom in to see details in the image)

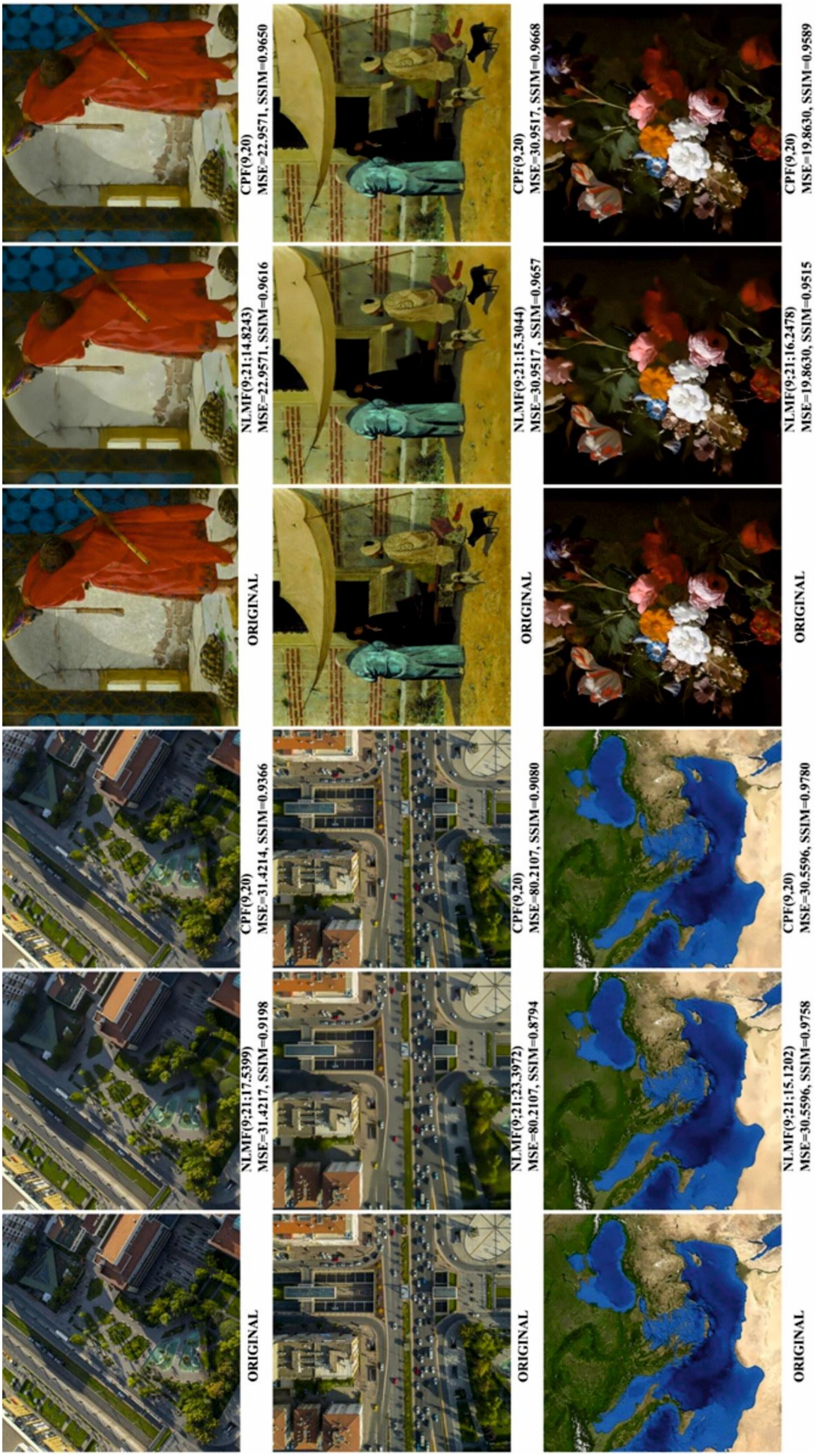


Figure 5. Comparison of CPF and NLMF for the same MSE values on Test Images (Please use zoom in to see details in the image)

MSE is a statistical measure that calculates the average of the squared differences between predicted and actual values. It is commonly used to evaluate the performance of regression models. The equation for MSE is given in Equation 3:

$$\mathrm{MSE} = \frac{1}{\mathrm{M} \cdot \mathrm{N}} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(A_{i,j} - B_{i,j}\right)^2 \qquad (3)$$

where "M" and "N" denote the 2D image size in pixels, and "A" and "B" denote the original and the smoothed images, respectively.

SSIM is a metric used to measure the similarity between two images. It compares the structural information of the images, including luminance, contrast, and structure, to determine the similarity. SSIM values range from -1 to 1, with 1 indicating a perfect match between the images. The formulation of the SSIM is given in Equation 4.

$$\mathrm{SSIM} = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_2\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \qquad (4)$$

μ_x: The pixel sample mean of *x*.
μ_y: the pixel sample mean of *y*.
σ_x^2: The variance of x.
σ_y^2: The variance of y.
σ_{xy}: The covariance of x and y.
$C_1=(k_1 \cdot L)^2$, $C_2=(k_2 \cdot L)^2$: two variables to stabilize the division with weak denominator;
L: the dynamic range of the pixel-values.
$k1 = 0.01k2 = 0.03$

SSIM has valuable properties, including indiscernible identity and symmetry. It's not a distance function because it doesn't fulfil triangle inequality or non-negativity. Yet, it can transform into a normalized root MSE measure that works as a distance function under certain conditions. The function's square is locally convex and quasiconvex, allowing for optimization. Please see [41] for the meanings of the symbols used in Equation 4. The experimental results for the test images using CPF and NLMF are shown in Figure 5. Upon examining Figure 5, we can conclude that while both methods yield the same MSE value, NLMF produces results that are over-smoothed, leading to partial loss of detail. In contrast, CPF produces images that are relatively smoother and of higher quality as measured by SSIM values, compared to NLMF.

4. Results and Conclusion

In digital image processing, the accurate identification and analysis of objects depend significantly on the connectivity of pixels. This concept plays a vital role in object recognition and segmentation in various domains, such as medical imaging and video surveillance. Image smoothing also plays a crucial role in reducing noise and enhancing details, thereby improving the visual quality in applications such as medical imaging, video processing, and computer vision. Without considering the connectivity of pixels and employing image smoothing techniques, image processing tasks would be challenging and less reliable.

In this paper, a novel image smoothing filter based on gray connected pixels, termed CPF, is proposed. The experimental results demonstrate that CPF outperforms the widely used non-local means filter (NLMF) in terms of detail preservation while having the same mean square error (MSE) values. Thus, CPF can be considered as a valuable tool for various image processing applications.

Some advantages and disadvantages of the CPF are given below:

1. Since the structure of the CPF is very simple, it can be easily adapted to different applications.
2. CPF is a non-recursive, non-iterative and non-linear image smoothing method.
3. Smoothed images produced with CPF contain relatively more details.
4. The speed of CDF, like NLMF, is dependent on the sliding window size.

Author contributions

Erkan Beşdok: Data curation, Software, Validation. **Pınar Çiviciоğlu:** Conceptualization, Methodology, Software, Writing, Editing

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Perri, S., Spagnolo, F., & Corsonello, P. (2020). A parallel connected component labeling architecture for heterogeneous systems-on-chip. Electronics, 9(2), 292.
2. Chen, J., Nonaka, K., Sankoh, H., Watanabe, R., Sabirin, H., & Naito, S. (2018). Efficient parallel connected component labeling with a coarse-to-fine strategy. IEEE Access, 6, 55731-55740.
3. Narasimhan, R. S., Vengadarajan, A., & Ramakrishnan, K. R. (2017, March). Design of connected component analysis-based clustering of CFAR image in pulse Doppler radars. In 2017 IEEE Aerospace Conference (pp. 1-6). IEEE.
4. Civicioglu, P., & Alci, M. (2003, December). CCII based analog circuit for the edge detection of MRI images. In 2003 46th Midwest Symposium on Circuits and Systems (Vol. 1, pp. 341-344). IEEE.
5. Jeong, J. W., Lee, G. B., Lee, M. J., & Kim, J. G. (2016). A single-pass connected component labeler without label merging period. Journal of Signal Processing Systems, 84(2), 211-223.
6. Besdok, E. (2004). Impulsive noise suppression from images with a modified two-step iterative-median filter. Journal of Electronic Imaging, 13(4), 714-719.

7. Çivicioğlu, P., & Alçı, M. (2004). Edge detection of highly distorted images suffering from impulsive noise. AEU-International Journal of Electronics and Communications, 58(6), 413-419.
8. Wang, Y., & Bhattacharya, P. (1996). On parameter-dependent connected components of gray images. Pattern Recognition, 29(8), 1359-1368.
9. Beşdok, E., Çivicioğlu, P., & Alçı, M. (2004). Impulsive noise suppression from highly corrupted images by using resilient neural networks. In Artificial Intelligence and Soft Computing-ICAISC 2004: 7th International Conference, Zakopane, Poland, June 7-11, 2004. Proceedings 7 (pp. 670-675). Springer Berlin Heidelberg.
10. Wang, Y., & Bhattacharya, P. (1996, November). Gray connected components and image segmentation. In Applications of Digital Image Processing XIX (Vol. 2847, pp. 118-129). SPIE.
11. Donato, M., Hansen, K., Kalavakuru, P., Kirchgessner, M., Kuster, M., Porro, M., ... & Turcato, M. (2017). First functionality tests of a 64× 64 pixel DSSC sensor module connected to the complete ladder readout. Journal of Instrumentation, 12(03), C03025.
12. Tang, J. W., Shaikh-Husin, N., Sheikh, U. U., & Marsono, M. N. (2018). A linked list run-length-based single-pass connected component analysis for real-time embedded hardware. Journal of Real-Time Image Processing, 15, 197-215.
13. Ray, V., & Goyal, A. (2016, January). Image-based fuzzy c-means clustering and connected component labeling subsecond fast fully automatic complete cardiac cycle left ventricle segmentation in multi frame cardiac MRI images. In 2016 International Conference on Systems in Medicine and Biology (ICSMB) (pp. 36-40). IEEE.
14. Ito, Y., & Nakano, K. (2010). Low-latency connected component labeling using an FPGA. International Journal of Foundations of Computer Science, 21(03), 405-425.
15. Chang, W. Y., & Chiu, C. C. (2014, June). Directional Connected Components Algorithm Based on Gradient Information. In 2014 International Symposium on Computer, Consumer and Control (pp. 280-283). IEEE.
16. Beşdok, E., & Yüksel, M. E. (2005). Impulsive noise suppression from images with Jarque-Bera test based median filter. AEU-International Journal of Electronics and Communications, 59(2), 105-110.
17. Çivicioğlu, P., Alçı, M., & Beşdok, E. (2004). Impulsive noise suppression from images with the noise exclusive filter. EURASIP Journal on Advances in Signal Processing, 2004(16), 2434-2440
18. Yeong, L. S., Ang, L. M., & Seng, K. P. (2010, July). Efficient connected component labelling using multiple-bank memory storage. In 2010 3rd International Conference on Computer Science and Information Technology (Vol. 9, pp. 75-79). IEEE.
19. Gunen, M. A., Civicioglu, P., & Beşdok, E. (2016). Differential search algorithm-based edge detection. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 41(B7), 667-670
20. Bieniek, A., & Moga, A. (1998). A connected component approach to the watershed segmentation. Computational Imaging and Vision, 12, 215-222.
21. Sun, Y., Sun, C., & Wang, W. (2000, August). Color images segmentation using new definition of connected components. In WCC 2000-ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000 (Vol. 2, pp. 863-868). IEEE.
22. Ma, D., Liu, S., & Liao, Q. (2017). Run-based connected components labeling using double-row scan. In Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part III 9 (pp. 264-274). Springer International Publishing.
23. Asano, T. (2012). In-place algorithm for erasing a connected component in a binary image. Theory of Computing Systems, 50(1), 111-123.
24. Bekhtin, Y. S., Gurov, V. S., & Zavalishin, S. S. (2015, June). A run equivalence algorithm for parallel connected component labeling on CPU. In 2015 4th Mediterranean Conference on Embedded Computing (MECO) (pp. 276-279). IEEE.
25. Yunfeng, G., Feiyang, W., & Xiaotian, H. (2014, August). Connected components labeling algorithm based on run-length table searching. In 2014 9th International Conference on Computer Science & Education (pp. 700-704). IEEE.
26. Rasmusson, A., Sørensen, T. S., & Ziegler, G. (2013). Connected components labeling on the GPU with generalization to voronoi diagrams and signed distance fields. In Advances in Visual Computing: 9th International Symposium, ISVC 2013, Rethymnon, Crete, Greece, July 29-31, 2013. Proceedings, Part I 9 (pp. 206-215). Springer Berlin Heidelberg.
27. Il, H. J., Kim, H. K., & Oh, W. G. (2015, January). Fast text line detection by finding linear connected components on Canny edge image. In 2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV) (pp. 1-4). IEEE.
28. Kowalczyk, M., & Kryjak, T. (2021, September). A Connected Component Labelling algorithm for a multi-pixel per clock cycle video stream. In 2021 24th Euromicro Conference on Digital System Design (DSD) (pp. 43-50). IEEE.
29. Rouabeh, H., Abdelmoula, C., & Masmoudi, M. (2016, December). A new efficient connected component labeling algorithm and its VHDL circuit. In 2016 28th International Conference on Microelectronics (ICM) (pp. 105-108). IEEE.
30. Bailey, D. G. (2020, November). History and evolution of single pass connected component analysis. In 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ) (pp. 1-6). IEEE.
31. Wang, Y., & Bhattacharya, P. (1996, October). Image analysis and segmentation using gray connected components. In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929) (Vol. 1, pp. 444-449). IEEE.
32. Kang, S. M., Kim, J. H., Yuan, Z., Song, S. H., & Cho, J. D. (2014, June). A fast region expansion labeling of

connected components in binary image. In The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014) (pp. 1-2). IEEE.
33. Flatt, H., Blume, S., Hesselbarth, S., Schunemann, T., & Pirsch, P. (2008, July). A parallel hardware architecture for connected component labeling based on fast label merging. In 2008 International Conference on Application-Specific Systems, Architectures and Processors (pp. 144-149). IEEE.
34. Krämer, M., Afzal, M. Z., Bukhari, S. S., Shafait, F., & Breuel, T. M. (2012, November). Robust stereo correspondence for documents by matching connected components of text-lines with dynamic programming. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012) (pp. 734-737). IEEE.
35. Yapa, R. D., & Koichi, H. (2007, March). A connected component labeling algorithm for grayscale images and application of the algorithm on mammograms. In Proceedings of the 2007 ACM symposium on Applied computing (pp. 146-152).
36. Zhang, D., Ma, H., & Pan, L. (2019). A gamma-signal-regulated connected components labeling algorithm. Pattern Recognition, 91, 281-290.
37. Shim, J., Yoon, M., & Lee, Y. (2019). Feasibility of fast non local means filter in pediatric chest x-ray for increasing of pulmonary nodule detectability with 3D printed lung nodule phantom. Journal of Radiological Protection, 39(3), 872-889
38. Shim, J., Yoon, M., & Lee, Y. (2018). Feasibility of newly designed fast non local means (FNLM)-based noise reduction filter for X-ray imaging: A simulation study. Optik, 160, 124-130.
39. Civicioglu, P., & Besdok, E. (2019). Bernstain-search differential evolution algorithm for numerical function optimization. Expert Systems with Applications, 138, 112831.
40. Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence, 12(7), 629-639.
41. Civicioglu, P., & Besdok, E. (2022). Contrast stretching based pansharpening by using weighted differential evolution algorithm. Expert Systems with Applications, 208, 118144.