IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 9, September 2019

New Fast Modified Non-Maximum Suppression Algorithm to Find Local Extrema in Grayscale Images

A. T. Nguyen[1], V. Yu. Tsviatkou[2]

PG Student, Dept. of Info-Communication Technologies, BSUIR, Minsk, Belarus[1]

Associate Professor, Dept. of Info-Communication Technologies, BSUIR, Minsk, Belarus[2]

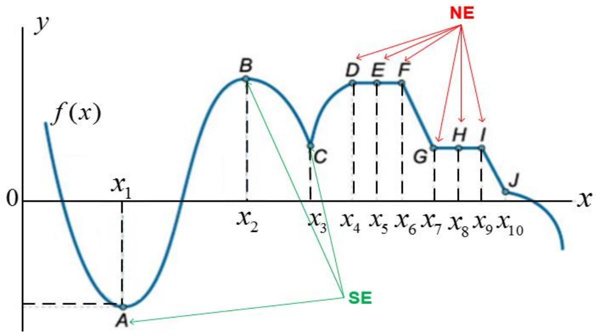
Abstract: The aim of the work is to develop an algorithm for extracting local extrema of images with low computational complexity and high accuracy. The known algorithms of the search for local extrema based on non-maximum (or non-minimum) suppression have low computational complexity, but only strict maxima or strict minima are detected without errors. The morphological algorithms give accurate results, in which the extreme areas are formed by strict and non-strict extrema. However, it has a high computational complexity, separate process of the search for maxima and minima (iterative processing). In this paper, a new modified non-maximum suppression algorithm to find all local extrema on grayscale images is proposed. The essence of the algorithm is to search for single-pixel local extrema and regions of uniform brightness, comparing the values of their boundary pixels with the values of the corresponding pixels of adjacent regions by following: the region is a local maximum (minimum) if the values of all its boundary pixels are larger (or smaller) to the values of all adjacent pixels. The proposed algorithm allows to detect all single-pixel local extrema and extreme areas in images. Besides, the proposed algorithm in comparison with the morphological algorithm requires low computational complexity and reduces the processing time and the use of RAM.

Keywords: Local Extrema; Non-Maximum Suppression; Image Segmentation; Region Growing

I. INTRODUCTION

Non-Maximum Suppression (NMS) is the task of finding all local maxima in an image. The term ‘non-maximum suppression’ first appeared in an edge detection context as a method to reduce thick edge responses to thin lines [1]. This type of directional NMS operates one dimensionally (1-D) perpendicular to the edge. Kitchen and Rosenfeld [2] extended the concept to isotropic NMS to locate two-dimensional (2-D) feature points from an image. The feature points are selected as local maxima of a ‘cornerness’ image over some neighborhood. This NMS approach to corner detection was subsequently adopted by many interest point detectors [3–5].

Image processing often requires the determination of initial elements, which can be local 2-D extrema (local maxima and local minima). To search for local extrema, some block algorithms [6–11] and morphological algorithms [12] are used.
In block algorithms, the search for extrema is computed within overlapping blocks, usually 3×3 or $(2n + 1) \times (2n + 1)$ pixels in size. Such algorithms have low computational complexity, but they extract local extrema by iterative processing (local strict maxima and local strict minima are extracted separately), skip non-strict extrema or extreme regions (for a homogeneous region consisting of several adjacent pixels with the same values, none of the pixels in this region is detected as a local extremum) (Fig. 1).



SE - strict extremum; NE – non-strict extremum

Fig. 1 Types of local extrema in one-dimensional representation

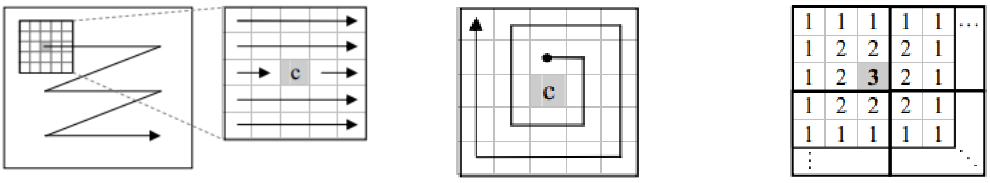
The morphological algorithm is used for the extraction of local maxima and minima in an image, respectively, by image reconstruction based on dilatation and erosion operations. It gives accurate results compared to block algorithms, highlighting both strict extrema and extreme areas (multi-pixel extrema). However, the morphological algorithm has high computational complexity, which is associated with separate processing of maxima and minima, as well as iterative processing of the neighborhoods of all pixels.

In this paper, the aim of the work is to develop an algorithm for extracting local extrema in grayscale images with low computational complexity, high accuracy and requires less than RAM.

II. PREVIOUS SOLUTIONS

A straightforward approach to NMS over a rectangular neighborhood is described in Fig. 2a. The input image pixels are visited in a raster scan order (from left to right, then from top to bottom). Each visited pixel is compared to other pixels in its neighborhood also in a raster scan order. The central pixel is a non-maximum if a larger or equal neighbor is found. The algorithm then skips to the next pixel in the scan line. The straightforward method is simple to implement but it can take a long time to process an image. On average, however, the straightforward method requires $O(n)$ comparisons per pixel.

It turns out that the complexity of a raster scan algorithm can be significantly reduced by visiting the neighbouring pixels in a different order. Forstner and Gulch [9] presented such an algorithm with a local spiral order as shown in Fig. 2b. By comparing with closer neighbours first, the central pixel is guaranteed to be a 3×3-neighborhood local maximum before it is tested against a larger neighbourhood. Because the number of 3×3 local maxima in an image is usually small ($\leq 25\%$ of the total number of pixels), the spiral order algorithm quickly finds any non-maximum and skips to the next pixel. The number of $(2n+1) \times (2n+1)$ local maxima also decreases rapidly as the neighborhood size increases. As a result, the computational complexity of this algorithm is roughly constant (≤ 5 comparisons per pixel to detect a 3×3 non-maximum) irrespective of the neighbourhood size.

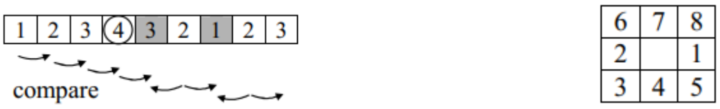


(a) Raster scan order [9] (b) Spiral order [10] (c) Block partitioning [9]

Fig. 2 Previous solutions for 2-D non-maximum suppression (5x5 neighborhood)

Recently, Neubeck and Van Gool [9] presented an efficient NMS algorithm that requires 2.39 comparisons per pixel on average & 4 comparisons per pixel in the worst-case. They observed that the maximum pixel of a $(2n+1) \times (2n+1)$ neighborhood is also the maximum of any $(n+1) \times (n+1)$ window that encloses the pixel. The input image is partitioned into non-overlapping blocks of size $(n+1) \times (n+1)$, and the local maximum of each block is detected (Fig. 2c illustrates this for $n = 2$). The block maximum is then tested against its $(2n+1) \times (2n+1)$ neighborhood minus the enclosing $(n+1) \times (n+1)$ block. Using only one comparison per pixel, the block partitioning step reduces the number of local maximum candidates by a factor of $(n+1)^2$. As a result, the Neubeck method is quite efficient for large neighbourhood sizes.

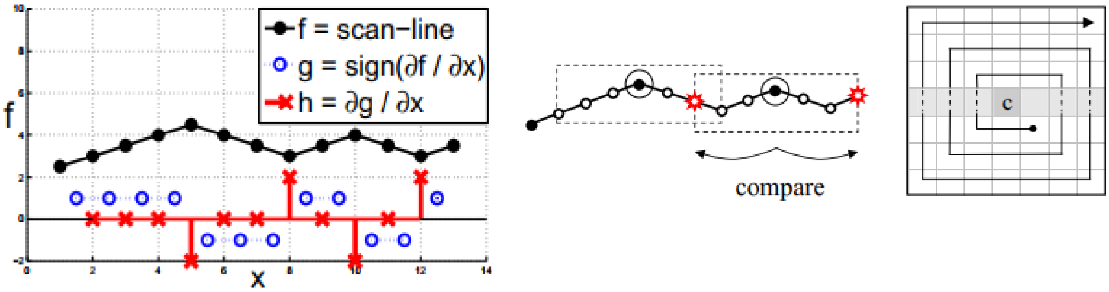
In paper [11], the algorithms in [9] and [10] are extended to reduce the number of comparisons to fewer than two comparisons per pixel. The algorithm first searches for 1-D local maxima along the scan line. Each scan-line maximum is then compared against its neighbors in adjacent rows. A rolling buffer of two binary masks is kept for a current and a next scan line. As a new central pixel is processed, its future neighbors are masked out if they are smaller than the central pixel. Masked pixels will be skipped when it is their turns for processing. This 1-D non maximum suppression algorithm therefore requires one comparison per pixel (Fig. 3).



(a) 1D non-maximum suppression (b) 3x3 scan order

Fig. 3 3-neighborhood non-maximum suppression and 3×3-neighborhood scan order

The scan-line algorithm for 3×3–neighborhoods before can be extended to handle $(2n+1) \times (2n+1)$ –neighborhoods for $n > 1$. First, $(2n+1)$– neighborhood maxima on the current scan-line are located. These 1-D maxima serve as candidates for the 2-D maxima. Each candidate is then compared against its $(2n+1) \times (2n+1)$–neighborhood in a spiral order similar to that of Forstner's method [10]. Note that the neighbors on the same scan-line have already been compared and can therefore be skipped. This results in a maximum of $(2n) \times (2n+1)$ neighbors to be compared per candidate. In practice, the average number of comparisons per candidate is much smaller thanks to the spiral traverse order (Fig. 4c).



(a) 1D peak and trough detection (b) 1D non-max suppression (c) spiral traverse

Fig. 4 Scan-line algorithm for $(2n+1) \times (2n+1)$ non-maximum suppression $(n = 3)$

From the above algorithms, we can describe the expressions to search for local maximum and local minimum by following:

$$e_{MAX}(y,x)\begin{cases}1, & \forall y'\forall x'(f(y,x) > f(y+y',x+x'),\\0, & \forall y'\forall x'(f(y,x) \leq f(y+y',x+x').\end{cases} \quad (1)$$

$$e_{MIN}(y,x)\begin{cases}-1, & \forall y'\forall x'(f(y,x) < f(y+y',x+x'),\\0, & \forall y'\forall x'(f(y,x) \geq f(y+y',x+x').\end{cases} \quad (2)$$

where $y = \overline{0, Y-1}, x = \overline{0, X-1}, (y' \in \{-1,0,1\}) \wedge (x' \in \{-1,0,1\}) \backslash ((y' = 0) \wedge (x' = 0))$. As a result of combining the matrices E_{MAX} and E_{MIN}, a matrix of local extrema E is obtained, the values of the elements of which are calculated using the expression

$$e(y,x) = e_{MAX}(y,x) + e_{MIN}(y,x) \quad (3)$$

Regional extrema are detected better by morphological reconstruction by dilation and erosion [11, 12], in which a *regional minimum M* of an image *f* at elevation *t* is a connected component of pixels with the value *t* whose external boundary pixels have a value strictly greater than *t*. *M* is a *regional minimum* at level *t* ⟺ *M* is connected and

$$\begin{cases}\forall p \in M, f(p) = t,\\ \forall q \in \delta^{(1)}(M)\backslash M, f(q) > t.\end{cases} \quad (4)$$

Similarly, a *regional maximum M* of an image *f* at elevation *t* is a connected component of pixels with the value *t* whose external boundary pixels have a value strictly less than *t*. *M* is a *regional maximum* at level *t* ⟺ *M* is connected and

$$\begin{cases}\forall p \in M, f(p) = t,\\ \forall q \in \delta^{(1)}(M)\backslash M, f(q) < t.\end{cases} \quad (5)$$

The regional extrema of an image are defined as the union of its regional minima and maxima. According to Eq. 4, the set of all maxima of an image *f* at level *t* corresponds to the connected components of the cross-section of *f* at level *t* that are not connected to any component of the cross-section of *f* at level *t*+1. They are therefore not reconstructed by the morphological reconstruction by dilation $CS_t(f)$ of from $CS_{t+1}(f)$. Denoting by the regional maxima of an image at level *t*, we can write:

$$RMAX_t(f) = RMAX(f) \cap T_t(f) = CS_t(f) \backslash R^{\delta}_{CS_t(f)}[CS_{t+1}(f)]. \quad (6)$$

The set of all maxima is defined by considering the union of the maxima obtained at each level *t*:

$$RMAX_t(f) = \cup_t \{CS_t(f) \backslash R^{\delta}_{CS_t(f)}[CS_{t+1}(f)]\}. \quad (7)$$

DOI 10.17148/IJARCCE.2019.8901

Since $CS_{t+1}(f) = CS_t(f-1)$ and $RMAX_t(f) \cap RMAX_{t1}(f) = \emptyset$ for $t \neq t1$ all, the set difference in Eq. 7 can be replaces with an algebraic difference and the union by a summation. The summation is then distributed and the threshold superposition principle gives:

$$RMAX(f) = f - R_f^{\delta}(f-1) \tag{8}$$

If the image data type does not support negative values, the following equivalent definition must be considered:

$$RMAX(f) = f + 1 - R_f^{\delta}(f) \tag{9}$$

Similarly, the regional minima of an image f at level t are denoted by $RMIN_t(f)$: $RMIN_t(f) = RMIN(f) \cap T_t(f) = R^{\varepsilon}_{CS_{t+1}(f)}[CS_t(f) \setminus CS_{t+1}(f)]$. The set of all regional minima is denoted by RMIN and defined by threshold superposition:

$$RMIN(f) = R_f^{\varepsilon}(f+1) - f \tag{10}$$

From the expressions 1 and 2 the main disadvantages of block algorithms for isolating local extrema follow:
1. Redundancy of processing. The independent formation of matrices by expression 1 also leads to redundancy of processing, since the matrix pixels are re-processed when the matrix is formed to find the local minimum.
2. Skipping non-strict extrema. From the expression 1 it follows that if the matrix has a local maximum homogeneous region consisting of several adjacent pixels with the same values (non-strict maxima), not one of the pixels in this region is detected as a local maximum. The same is true for local minima by expression 2. The number and area of such regions grow during quantization, filtering, and restoration of images after lossy compression. In such cases, skipping non-strict extremes leads to incomplete image segmentation, errors in detection, localization, and parameterization of objects.
3. The need for additional processing of the resulting matrix to assign label numbers to local extrema.

From the expressions 4 to 10, the main disadvantages of morphological algorithms for isolating local extrema follow, leading to high computational complexity:
1. Separate processing of local maxima and local minima.
2. Iterative processing of neighbourhoods of all image pixels.
3. The need for additional segmentation for assigning label numbers to extreme regions.

III. NEW FAST MODIFIED NON-MAXIMUM SUPPRESSION ALGORITHM

To eliminate the above disadvantages a new fast modified non-maximum suppression (MNMS – Modified Non-Maximum Suppression) and its mathematical model to extract all local extrema in grayscale images with low computational complexity and high accuracy are proposed (expressions 11, 12, Fig. 6). The essence of the algorithm is to search for single-pixel local extrema and areas of uniform brightness, comparing the values of their boundary pixels with the values of the corresponding pixels of adjacent areas by following: the region is a local maximum (minimum) if the values of all its boundary pixels are greater (less) or equal to the values of all adjacent pixels. Along with single-pixel extrema, the algorithm takes into account homogeneous regions (of two or more pixels), which are local maxima or minima with respect to adjacent regions due to image segmentation and analysis of brightness changes at the boundaries of the regions.

The values of the elements $e(y,x)$ of the matrix E of local extrema indicate that the corresponding pixels of the image belong to the $n_S - th$ maximum ($e(y,x) = n_S$), minimum ($e(y,x) = -n_S$), or non-extremum ($e(y,x) = 0$) of region $R(n_S)$. The proposed algorithm allows high accuracy in comparison with block algorithms [6-11] and less computational complexity in comparison with morphological search algorithm [12] by: combined search for maxima and minima by assigning elements $e(y,x)$ both positive and negative values; taking into account non-strict extrema by estimating the neighborhood of a pixel $f(y,x)$ using non-strict inequalities; the elimination of search errors of local extreme regions due to the assessment of the neighborhoods of all pixels of each homogeneous region $R(n_S)$; the absence of the need for segmentation of the matrix E of local extrema due to the assignment of numbers to single-pixel extrema and extreme regions; a single processing of the neighborhoods of all pixels due to the cultivation of homogeneous areas $R(n_S)$.

$$\forall y \forall x \exists n_S (f(y,x) \in R(n_S)) \tag{11}$$

$$e(y,x)=\begin{cases} n_s \text{ if } \left(f(y,x)\in R(n_s)\right)\wedge\Big(\left(\forall\Delta y_1\forall\Delta x_1\left(f(y,x)>f(y+\Delta y_1,x+\Delta x_1)\right)\right)\vee \\ \qquad\vee\left(\forall y'\forall x'\forall\Delta y_2\forall\Delta x_2\left(\left(f(y',x')\in R(n_s)\right)\Rightarrow\left(f(y',x')\geq f(y'+\Delta y_2,x'+\Delta x_2)\right)\right)\right)\Big), \\ 0 \text{ if } \left(f(y,x)\in R(n_s)\right)\wedge \\ \qquad\wedge\exists y'\exists x'\exists\Delta y_1\exists\Delta x_1\left(\left(f(y',x')\in R(n_s)\right)\Rightarrow\left(f(y',x')<f(y'+\Delta y_1,x'+\Delta x_1)\right)\right)\wedge \\ \qquad\wedge\exists y''\exists x''\exists\Delta y_2\exists\Delta x_2\left(\left(f(y'',x')\in R(n_s)\right)\Rightarrow\left(f(y'',x')>f(y''+\Delta y_2,x'+\Delta x_2)\right)\right), \\ -n_s \text{ if}\left(f(y,x)\in R(n_s)\right)\wedge\Big(\left(\forall\Delta y_1\forall\Delta x_1\left(f(y,x)<f(y+\Delta y_1,x+\Delta x_1)\right)\right)\vee \\ \qquad\vee\left(\forall y'\forall x'\forall\Delta y_2\forall\Delta x_2\left(\left(f(y',x')\in R(n_s)\right)\Rightarrow\left(f(y',x')\leq f(y'+\Delta y_2,x'+\Delta x_2)\right)\right)\right)\Big) \end{cases} \tag{12}$$

where $y=\overline{0,Y-1}$, $x=\overline{0,X-1}$, $(\Delta y_1\in\{-1,0,1\})\wedge(\Delta x_1\in\{-1,0,1\})\backslash((\Delta y_1=0)\wedge(\Delta x_1=0))$, $(\Delta y_2\in\{-1,0,1\})\wedge(\Delta x_2\in\{-1,0,1\})\backslash((\Delta y_2=0)\wedge(\Delta x_2=0))$, $n_S\in[0,N_S]$ – the number of homogeneous brightness extreme region $R(n_S)$, N_S – the number of extreme regions. The block diagram of the proposed algorithm is shown in Fig. 6.

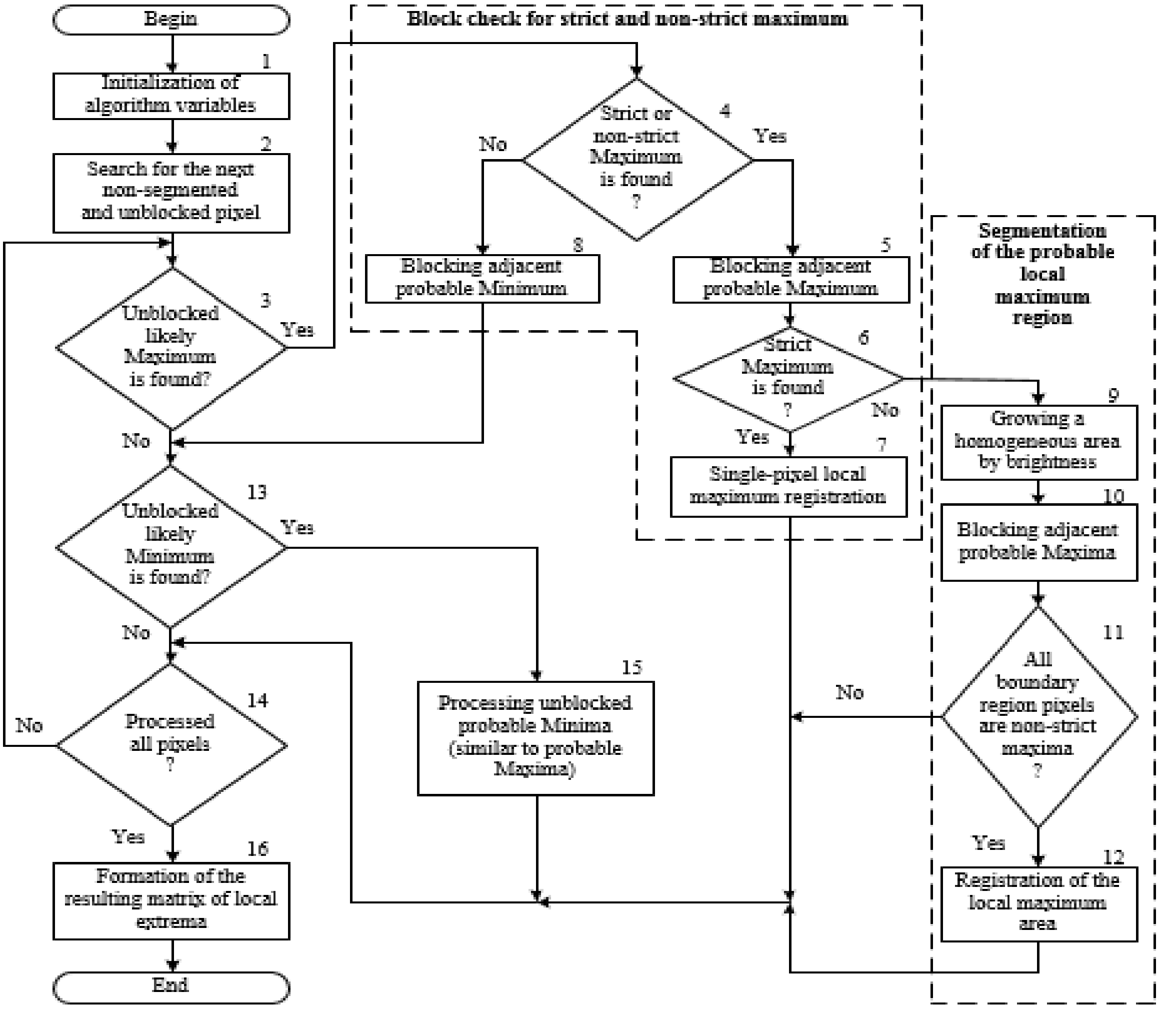

Begin
1
Initialization of algorithm variables
2
Search for the next non-segmented and unblocked pixel
3
Unblocked likely Maximum is found?
Yes
No
Block check for strict and non-strict maximum
4
Strict or non-strict Maximum is found ?
No
Yes
8
Blocking adjacent probable Minimum
5
Blocking adjacent probable Maximum
6
Strict Maximum is found ?
No
Yes
7
Single-pixel local maximum registration
Segmentation of the probable local maximum region
9
Growing a homogeneous area by brightness
10
Blocking adjacent probable Maxima
11
All boundary region pixels are non-strict maxima ?
No
Yes
12
Registration of the local maximum area
13
Unblocked likely Minimum is found?
Yes
No
15
Processing unblocked probable Minima (similar to probable Maxima)
14
Processed all pixels ?
No
Yes
16
Formation of the resulting matrix of local extrema
End

Fig. 6 Block diagram of the proposed algorithm

As shown in Fig. 6, the algorithm begins with an initialization block (block 1). Then, in the loop, the next non-segmented and unblocked pixel is searched (block 2). If an unblocked probable maximum is found (block 3), then it is block-checked for a strict and non-strict maximum (blocks 4–8). If the current pixel is not a strict or non-strict maximum (block 4), then the corresponding adjacent probable minimum is blocked (if the current pixel is less than the

DOI 10.17148/IJARCCE.2019.8901

adjacent one, then the adjacent pixel cannot be a minimum) (block 8) and the processing proceeds to the unblocked probable minimum (block 13). If the current pixel is a strict or non-strict maximum (block 4), then all adjacent probable maxima are blocked (block 5) and the current pixel is checked for a strict maximum (block 6).

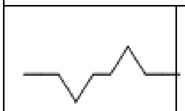
If the condition is met (block 6), then the current pixel is registered as a single-pixel local maximum (block 7) and the processing of the next pixel is completed (block 14). If the condition is not met, then the segmentation of the probable local maximum region is carried out (blocks 9–12). To do this, a region of uniform brightness in brightness is grown as a result of the gradual addition of neighboring pixels with equal values to the current pixel (block 9). All pixels adjacent to the boundary pixels of the formed region, which are non-strict maxima, are blocked as probable maxima (block 10). If at least one boundary pixel of the region is not a non-strict maximum (block 11), then the entire region is not a local maximum. In this case, the transition to processing the next pixel is performed (block 14). If all the boundary pixels of the region are non-strict maxima (block 11), then the selected homogeneous region is registered as a local maximum (block 12) and the process proceeds to the processing of the next pixel (block 14). Unblocked probable minima are processed similarly (block 15). If all the pixels are not processed (block 14), then the transition to the search for the next non-segmented and unblocked pixel (block 2) is performed. If all the pixels are processed (block 14), then the resulting matrix of local extrema is formed (block 16).

IV. RESULTS

For a visual assessment of the accuracy of the algorithms, Table I shows eight matrices with a different number of local extrema, the brightness distribution, and the number of extreme regions for each matrix. From table 1 it follows that the block algorithm Scanline3x3 (SL) [11] does not detect local extrema formed by several identical elements (rows 2, 5, 7, 8 of Table II). The proposed MNMS and morphological algorithm (M) extract all local extrema without errors consisting of one or more elements.

Table I Types of Extrema on Test Matrices

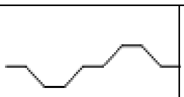
Type 1



(2 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	16	16	16	16	16	20	20	24	24	24	24	24	20	20	20	20
20	20	20	20	16	13	13	13	16	20	20	24	27	27	27	24	20	20	20	20
20	20	20	20	16	13	**10**	13	16	20	20	24	27	**30**	27	24	20	20	20	20
20	20	20	20	16	13	13	13	16	20	20	24	27	27	27	24	20	20	20	20
20	20	20	20	16	16	16	16	16	20	20	24	24	24	24	24	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

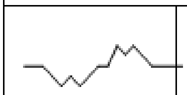
Type 2



(2 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	16	16	13	16	16	20	20	24	24	27	24	24	20	20	20	20
20	20	20	20	16	13	**10**	13	16	20	20	24	27	**30**	27	24	20	20	20	20
20	20	20	20	13	**10**	**10**	**10**	13	20	20	27	**30**	**30**	**30**	27	20	20	20	20
20	20	20	20	16	13	**10**	13	16	20	20	24	27	**30**	27	24	20	20	20	20
20	20	20	20	16	16	13	16	16	20	20	24	24	27	24	24	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

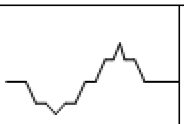
Type 3



(6 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	19	19	19	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	18	18	18	18	18	20	20	22	24	24	24	22	20	20	20	20
20	20	20	18	16	16	16	16	16	20	20	24	26	26	26	24	22	20	20	20
20	20	20	18	16	**13**	**18**	**13**	16	20	20	24	**30**	**22**	**30**	24	22	20	20	20
20	20	20	18	16	16	16	16	16	20	20	24	26	26	26	24	22	20	20	20
20	20	20	20	18	18	18	18	18	20	20	22	24	24	24	22	20	20	20	20
20	20	20	20	20	19	19	19	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

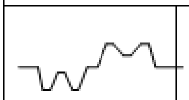
Type 4



(2 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	18	18	18	20	20	20	20	20	20	20	20	20	22	22	20	20	20
20	20	16	16	16	16	16	20	20	20	20	20	20	20	24	24	24	24	20	20
20	20	16	12	12	12	12	20	20	20	20	20	20	20	27	27	27	24	20	20
20	20	16	14	14	**10**	14	14	16	20	20	22	24	24	27	**30**	27	24	24	20
20	20	16	12	12	12	12	20	20	20	20	20	20	20	27	27	27	24	20	20
20	20	16	16	16	16	16	20	20	20	20	20	20	20	20	24	24	20	20	20
20	20	20	18	18	18	20	20	20	20	20	20	20	20	20	22	22	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

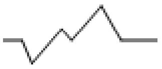
Type 5



(6 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	19	19	19	20	20	20	20	20	20	20	22	22	22	20	20	20	20	20	20	20
20	20	20	20	20	18	18	18	18	18	20	20	20	20	20	22	24	24	24	22	22	20	20	20	20	20
20	20	20	20	**10**	16	16	16	16	16	20	20	20	20	20	26	26	26	26	26	26	26	20	20	20	20
20	20	16	**10**	**10**	16	**18**	**18**	14	**10**	**10**	16	20	20	24	**30**	**30**	26	**22**	**22**	26	**30**	**30**	26	20	20
20	20	20	20	**10**	16	16	16	16	16	20	20	20	20	20	26	26	26	26	26	26	26	20	20	20	20
20	20	20	20	20	18	18	18	18	18	20	20	20	20	20	22	24	24	24	22	22	20	20	20	20	20
20	20	20	20	20	20	19	19	19	20	20	20	20	20	20	20	22	22	22	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

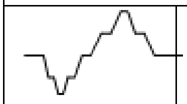
Type 6



(4 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	16	16	16	16	16	20	20	24	24	24	24	24	20	20	20	20
20	20	20	20	16	13	13	13	16	22	22	24	27	27	27	24	20	20	20	20
20	20	20	18	16	**10**	14	18	**26**	24	**18**	22	24	26	**30**	26	24	20	20	20
20	20	20	20	16	13	13	13	16	22	22	24	27	27	27	24	20	20	20	20
20	20	20	20	16	16	16	16	16	20	20	24	24	24	24	24	20	20	20	20
20	20	20	20	20	18	18	18	20	20	20	20	22	22	22	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

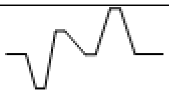
Type 7



(2 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	22	22	22	22	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	24	24	24	24	24	24	20	20	20
20	20	20	16	16	14	18	18	18	20	20	20	24	24	24	24	24	24	22	20	20
20	20	18	16	16	14	**10**	**10**	14	16	16	18	24	24	26	**30**	**30**	26	24	24	20
20	20	20	16	16	14	18	18	18	20	20	20	24	24	24	24	24	24	22	20	20
20	20	20	20	20	20	20	20	20	20	20	20	24	24	24	24	24	24	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	22	22	22	22	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

Type 8

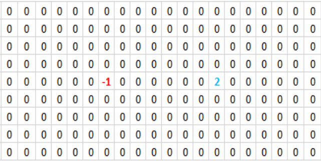
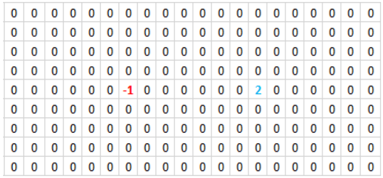
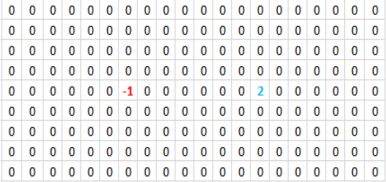
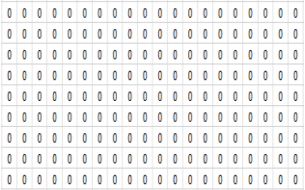
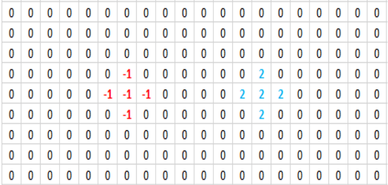
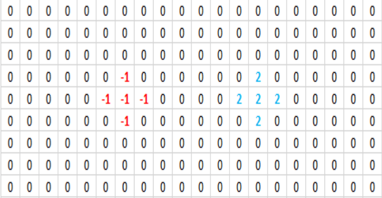
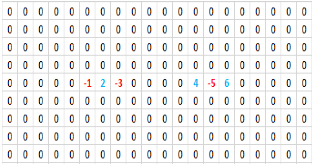
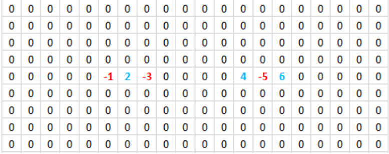
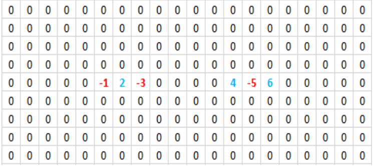
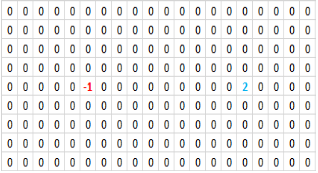
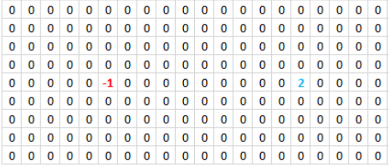
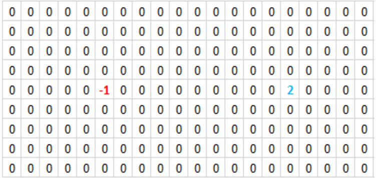
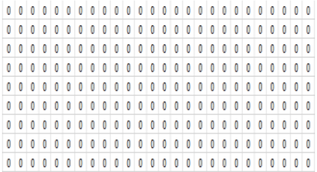
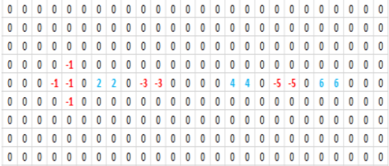
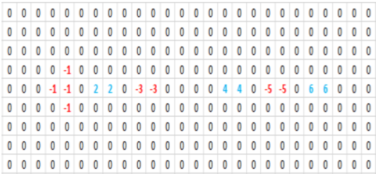
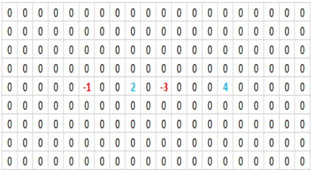
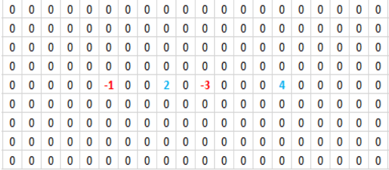
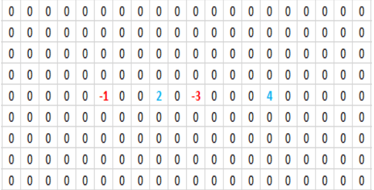
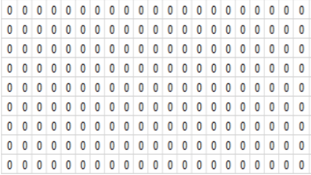
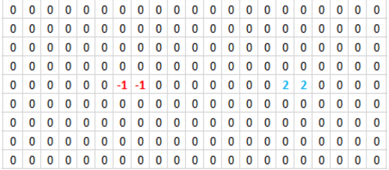
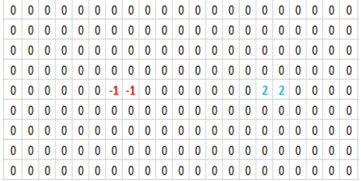
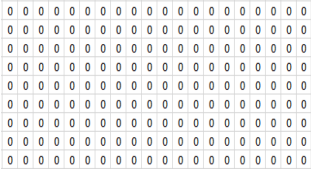
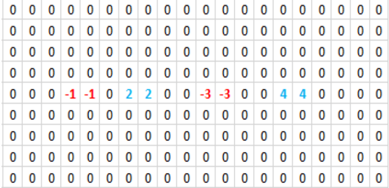
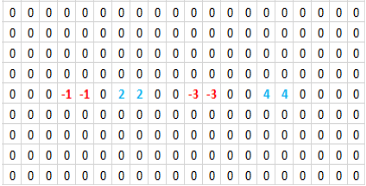


(4 extrema)

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	18	18	18	18	20	20	20	18	18	18	18	24	24	24	24	20	20	20
20	20	18	**10**	**10**	18	**24**	**24**	20	18	**14**	**14**	18	24	**30**	**30**	24	20	20	20
20	20	18	18	18	18	20	20	20	18	18	18	18	24	24	24	24	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

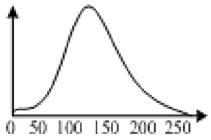
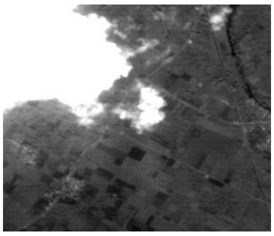
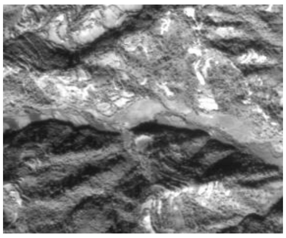
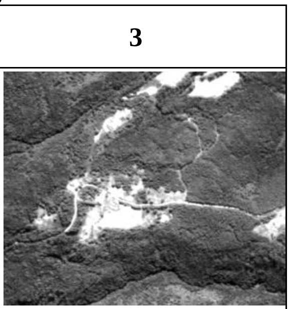
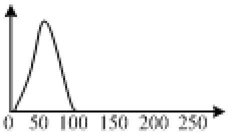
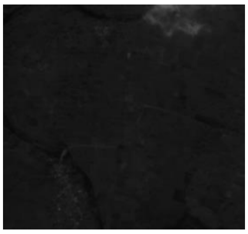
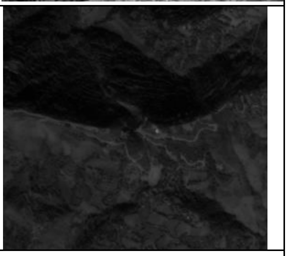
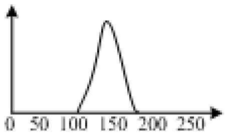
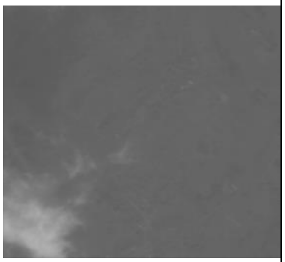
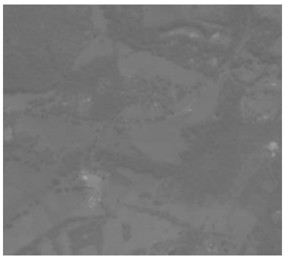
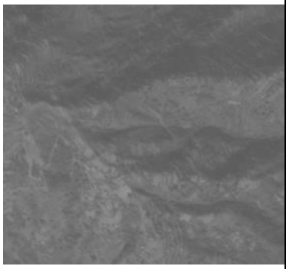
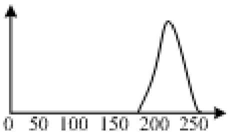
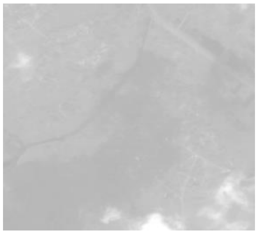
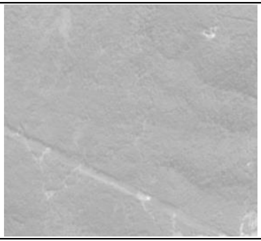
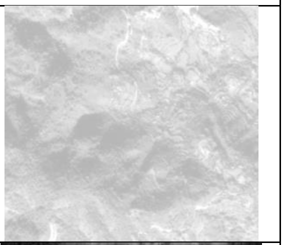
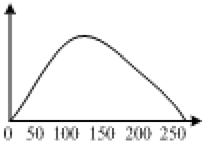
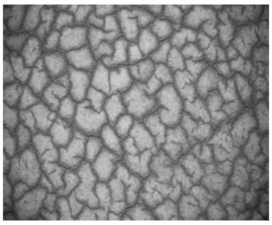
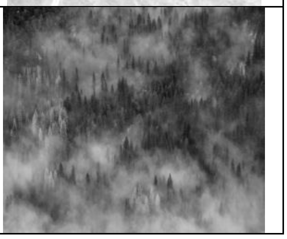
DOI 10.17148/IJARCCE.2019.8901

Table III Result of finding local extrema on test matrices in matlab

Segmented matrices of local extrema			
Type	**Scanline3x3 (SL)**	**Morphological algorithm (M)**	**Proposed algorithm (MNMS)**
1	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 2 -3 0 0 0 0 4 -5 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 2 -3 0 0 0 0 4 -5 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 2 -3 0 0 0 0 4 -5 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 2 2 0 -3 -3 0 0 0 0 4 4 0 -5 -5 0 6 6 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 2 2 0 -3 -3 0 0 0 0 4 4 0 -5 -5 0 6 6 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 2 0 -3 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 2 0 -3 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 2 0 -3 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 2 2 0 0 -3 -3 0 0 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 2 2 0 0 -3 -3 0 0 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The proposed MNMS algorithm is compared with the well-known block algorithm Scanline3x3 [11] and morphological algorithm [12] of searching for local extrema in terms of the number of extrema, processing speed, and RAM costs. The results were obtained with averaging over images divided into 15 types depending on the shape of the histograms of their brightness and brightness derivatives (Table III, IV, Fig.7, 8).

Table IIIII Grayscale Images of Size 512x512 for Testing

Types of brightness histogram		1	2	3
1				
2				
3				
4				
5				

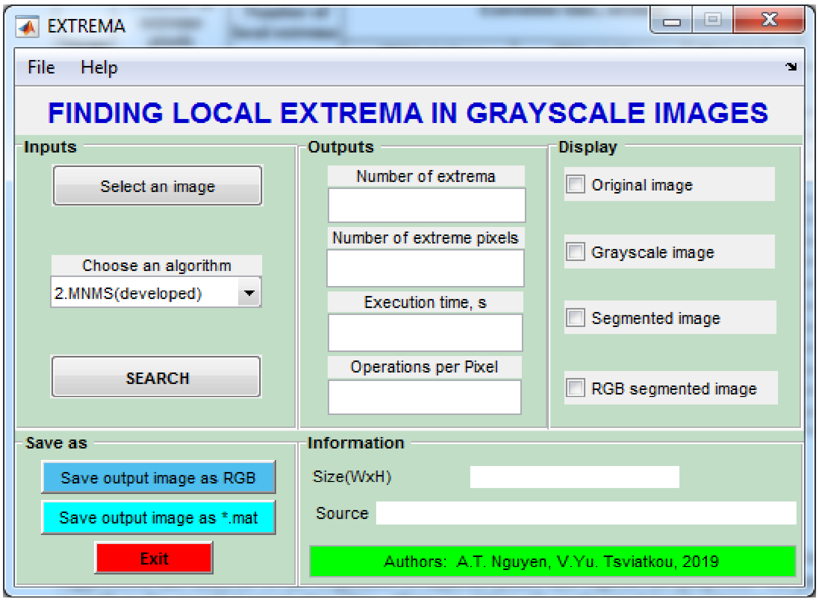
In Table IV it was experimentally established that the MNMS and morphological algorithms extract 1.4 times more the number of local extrema compared to the Scanline3x3 algorithm by taking into account non-strict extrema. The selection of extreme regions in addition to strict extrema leads to an increase in the computational complexity of the MNMS algorithm in comparison with the Scanline3x3 algorithm. When using computing platforms Intel Core i3 3.1GHz, 6GB of RAM, Windows7, Matlab2012 (IWM platform), Intel Core i3 3.1GHz, 6GB of RAM, Windows7, Open CV, C ++ (IWC platform), Raspberry Pi 3, ARM- A53, Linux, Open CV, C ++ (RLC platform) the MNMS algorithm in comparison with Scanline3x3 algorithm requires 1.6, 1.7, 1.9 times more processing time and 5.8 times more use of RAM, respectively. However, implementation of the MNMS algorithm on IWM and IWC platforms, as compared with the implementation of the morphological algorithm, requires 4.1, 3.1 times less processing time and 2.1 times less use of RAM, respectively.

Table IVV Result of finding local Extrema in Grayscale images (Table iii)

Image	Number of extreme pixels		Number of local extrema		Execution time, seconds								Number of Operations per Pixel		Number of missed extrema	
					IWM platform			IWC platform			RLC platform					
	SL	MNMS, M	SL	MNMS, M	SL	MNMS	M	SL	MNMS	M	SL	MNMS	SL	MNMS	SL	MNMS, M
1-1	16046	83053	16046	23304	0.087	0.161	1.987	0.028	0.052	0.147	0.089	0.241	16	47	7258	0
1-2	13091	24103	13091	16558	0.086	0.105	0.324	0.026	0.034	0.128	0.087	0.122	16	21	3467	0
1-3	14006	35713	14006	18575	0.085	0.109	0.403	0.027	0.037	0.129	0.087	0.135	16	23	4569	0
2-1	18344	44548	18344	27351	0.088	0.176	0.524	0.028	0.066	0.135	0.091	0.265	17	50	9007	0
2-2	9925	25433	9925	15127	0.087	0.147	0.534	0.026	0.053	0.128	0.086	0.198	16	38	5202	0
2-3	9260	20732	9260	13410	0.084	0.123	0.475	0.027	0.039	0.125	0.085	0.144	15	26	4150	0
3-1	16025	41380	16025	24137	0.088	0.183	0.782	0.028	0.067	0.132	0.089	0.263	17	50	8112	0
3-2	5612	24759	5612	10441	0.082	0.155	0.660	0.024	0.056	0.125	0.083	0.213	15	40	8429	0
3-3	8761	22494	8761	13486	0.082	0.124	0.390	0.026	0.042	0.127	0.085	0.152	15	28	4725	0
4-1	14461	36194	14461	20842	0.089	0.192	0.622	0.028	0.070	0.135	0.088	0.268	17	52	6381	0
4-2	11668	26065	11668	17059	0.084	0.122	0.432	0.026	0.038	0.130	0.086	0.137	16	25	5391	0
4-3	13102	25554	13102	17703	0.086	0.121	0.368	0.027	0.037	0.129	0.087	0.133	16	24	4601	0
5-1	28251	32573	28251	30547	0.090	0.099	0.242	0.029	0.031	0.134	0.093	0.108	17	18	2296	0
5-2	23473	27308	23473	25510	0.091	0.098	0.254	0.028	0.030	0.126	0.091	0.105	17	18	2037	0
5-3	13292	20049	13292	16269	0.087	0.104	0.275	0.027	0.033	0.122	0.087	0.116	16	20	2977	0

V. EXTENSIONS AND APPLICATIONS

The MNMS algorithm can be used to find distinctive feature points in an image. To improve the repeatability of a detected corner across multiple images, the corner is often selected as a local extremum whose corners is significantly higher (or lower) than the close-by second highest (or lowest) peak [4, 13].



EXTREMA

FINDING LOCAL EXTREMA IN GRAYSCALE IMAGES

Choose an algorithm: 2.MNMS(developed)

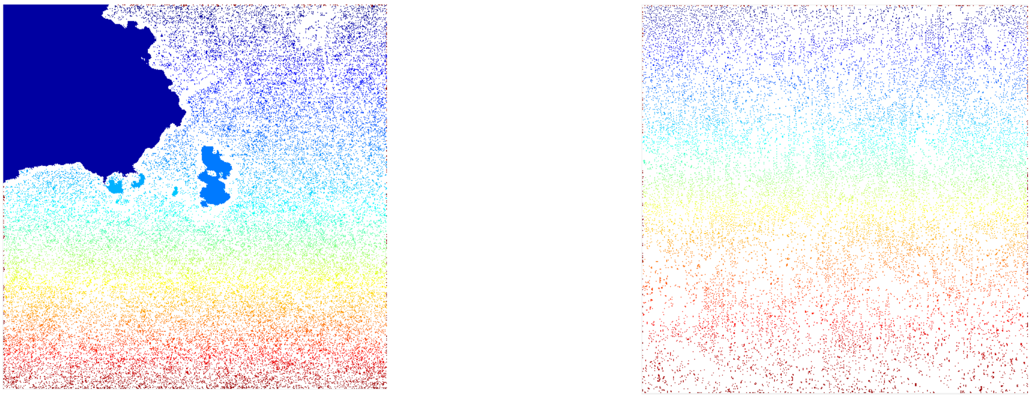
Authors: A.T. Nguyen, V.Yu. Tsviatkou, 2019

Fig. 7 Program interfaces of the proposed algorithm in Matlab

For some applications such as multi-view image matching, an evenly distributed set of interest points for matching is desirable. An oversupplied set of MNMS point features can be given to an adaptive non-extreme suppression process [14], which reduces cluttered corners to improve their spatial distribution.

Moreover, the MNMS algorithm can be used to video denoising by detecting highlight point in a video frame, aligning these points to estimate global shift, and average aligned video frames are used to improve Signal-to-Noise Ratio. In general, the MNMS algorithm enables some bonus applications: image segmentation, image recognition, image compression.

DOI 10.17148/IJARCCE.2019.8901



(a) RGB image of local extrema in image 1-1
Colour pixels – extrema;

(b) RGB image of local extrema in image 5-3
White pixels – background

Fig. 8 Example of getting RGB segmented images of local extrema

VI. CONCLUSION

A mathematical model and an algorithm for search of local extrema in grayscale images based on modified non-maximum suppression and analysis of the brightness of adjacent pixels and regions are proposed. The proposed algorithm MNMS, as well as the morphological algorithm, allows to detect all single-pixel local extrema and extreme areas consisting of pixels with the same values. Moreover, the proposed algorithm, in comparison with the morphological algorithm, requires less than 3-4 times depending on the computing platform with averaging over image types in time and 2 times in use of RAM.

REFERENCES

[1]. A. Rosenfeld and A. Kak, Digital Picture Processing. Academic Press, 1976. p. 457.

[2]. L. Kitchen and A. Rosenfelt, “Gray-level corner detection,” Pattern Recognition Letters 1 (2) (1982). pp. 92–102.

[3]. C. A. Harris and M. Stephens, “Combined corner and edge detector,” Proceeding of the Fourth Alvey Vision Conference, Manchester, 31 August-2 September 1988, pp.147–151.

[4]. D. Lowe, “Distinctive image features from scale-invariant keypoints,” IJCV 60 (2004), pp. 91–110.

[5]. K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” IJCV (60) 2004, pp. 63–86.

[6]. M. A. Van Herk, “Fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels,” Pattern Recognition Letters 13 (1992), pp. 517–521.

[7]. J. Gil and M. Werman, “Computing 2-D min, median, and max,” IEEE Trans. on PAMI 15 (1993), pp. 504–507.

[8]. M. Brown, R. Szeliski, and S. Winder, “Multi-image matching using multi-scale oriented patches,” Proceeding of Computer Vision and Pattern Recognition (CVPR’05) 1 (2005), pp. 510–517.

[9]. A. Neubeck and L. Van Gool, “Efficient non-maximum suppression,” Proceeding of ICPR 3, 2006, pp. 850–855.

[10]. W. A. Forstner and E. Gulch, “Fast operator for detection and precise locations of distinct points, corners, and centres of circular features,” Proceeding of Intercommission Conference on Fast Processing of Photogrammetric Data, 1987, pp. 281–305.

[11]. Tuan Q. Pham, “Non-maximum Suppression Using fewer than 2 Comparisons per Pixel,” Advanced Concepts for Intelligent Vision Systems (ACIVS) 12, Sydney, Australia, December 13-16 (2010), pp. 438–451.

[12]. P. Soille, Morphological Image Analysis: Principles and Applications. Springer, 2006.

[13]. C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” IJCV 37, 2000, pp. 151–172.

[14]. M. Brown, R. Szeliski and S. Winder, “Multi-image matching using multi-scale oriented patches,” Proceeding of Computer Vision and Pattern Recognition (CVPR’05) 1 (2005), pp. 510–517.

BIOGRAPHIES

Nguyen Anh Tuan – PG student of department of info communication technologies of Belerusian state university of informatics and radio electronics.

Tsviatkou Vikto Yurevich – Doctor of Engineering, associate professor, head of department of info communications of Belerusian state university of informatics and radio electronics.

DOI 10.17148/IJARCCE.2019.8901