Fuzzy Landform Classification with Azimuth-based Fourier Series

Gábor Nagy

Óbuda University, Alba Regia Faculty, Institute of Geoinformatics, Pirosalma utca 1-3, H-8000 Székesfehérvár, Hungary, nagy.gabor@amk.uni-obuda.hu

Abstract: The elevation differences between a point of the terrain surface and the points of a circle whose centre is this point may be the base of a Fourier series, where the azimuth (from the examined point to a point of the circle) is the variable of the function. The coefficients of this azimuth-based Fourier series are applicable for the analysis of the elevation models, for example classifying the points of the surface. The described analyses are implemented easy in varied GIS software because the coefficients are calculated by convolution filters. This paper describes equations for calculating fuzzy values of the landform classes from the coefficients of the azimuth-based Fourier series.

Keywords: landform classification; terrain analysis; geomorphology; fuzzy

1 Introduction

The parts of the terrain surface can be classified to different landforms types. The peaks and pits are the local maximum and minimum points of the surface. The ridges and valleys are the lines where two hillside (slope) join. The saddle is a special point of the terrain surface where the lines of valleys and ridges join. The plain areas are the part of the ground surface where the elevation differences are not greater than a threshold.

There are other classifications, which use more classes. For example, Márkus [26] uses three class for the slope areas: even, concave and convex slopes. MacMillan [24] describes various landform classification with other classes and in some cases sub-classes. Guilbert [14] writes about landform ontologies.

Several methods have been published for making geomorphology analysis and classifying the landform types. This article describes a new method, which is capable to classify the landform elements or make other geomorphological analysis.

2 Some Well-known Methods for Landform Classification

This section contains the short descriptions of some well-known landform classification methods. Bishop [2] presents some methods for geomorphological mapping, and Evans [9] writes about the landform mapping. Güler [15] describes a vector based morphometric tool.

Yokoyama [51] describes a method, which calculates the zenith angle of the terrain surface in different directions (in azimuths which are multiple of 45 degrees) and various distance. The classification is based in these parameters. Jacek [19] presents a similar method, which works in a DEM with the neighbour node's elevations.

Stepinski [40] presents a method of geomorphons. A point's geomorphon contains the signs of the elevation differences in 8 different directions.

Prima [33] uses supervised classification from DEM-derived parameters to separate different landform elements in a Japanese test area. The classification uses not only the DEM data, for example Saadat [37] presents a classification from DEM and satellite imagery data. The classification may be used with a segmentation, as Drăguţ and Eisank described [8].

There are various values derived by the elevation models and related to the landform. For example, Weiss [53] introduced the TPI, the Topographic Position Index. This value is the difference between the elevation of a point and the average elevation of this points neighbourhood (with a defined radius). The TPI values with different radiuses may be usable for the landform classification.

3 The Suggested Method

3.1 The Principles of Azimuth-based Fourier Series

Similar to more well-known methods, the suggested procedure studies the elevation differences between the studied point and other places around this point in a circle with a defined radius (denoted *r*). In this case, the elevation difference is a function of the azimuth (the azimuth from the studied point to a point of the circle, denoted δ):

$$\Delta H(\delta) = h(x_0 + r \cdot \sin\delta, y_0 + r \cdot \cos\delta) - h(x_0, y_0) \quad (1)$$

where the $\boldsymbol{h : R^2 \rightarrow R}$ function means the digital elevation model which can produce the elevation in a horizontal position determined by x and y coordinates. The r (radius of the circle) is a constant parameter in these equations.

A function ($\boldsymbol{f : R \rightarrow R}$) may be approximated by a s_N function:

$$f(x) \sim s_N(x) = \frac{a_0}{2} + \sum_{i=1}^{N} (a_i \cos(ix) + b_i \sin(ix)) \tag{2}$$

where a_i and b_i ($i \leq N$) are called Fourier coefficients. If $N = \infty$, the function is called Fourier series. ([11, 12]) In the practice, we use finite N values for the calculations. The Fourier coefficients of the $\boldsymbol{f}$ function can be calculated by the following equations:

$$\begin{aligned} a_i &= \tfrac{1}{\pi} \textstyle\int_{-\pi}^{\pi} f(x) \cos(ix)\, dx \\ b_i &= \tfrac{1}{\pi} \textstyle\int_{-\pi}^{\pi} f(x) \sin(ix)\, dx \end{aligned} \tag{3}$$

The $\Delta H(\delta)$ function may be approximated by a Fourier series because this is a periodic function. The Fourier coefficients can be calculated by the (3) and use $\Delta H(\delta)$ from the (1) instead of $f(x)$:

$$\begin{aligned} a_i = \tfrac{1}{\pi} \textstyle\int_{-\pi}^{\pi} &(h(x_0 + r\sin\delta, y_0 + r\cos\delta) \\ &- h(x_0, y_0)) \cos(i\delta) d\delta \\ b_i = \tfrac{1}{\pi} \textstyle\int_{-\pi}^{\pi} &(h(x_0 + r\sin\delta, y_0 + r\cos\delta) \\ &- h(x_0, y_0)) \sin(i\delta) d\delta \end{aligned} \tag{4}$$

In the practice, the integrals are changed to sums because we use numerical integral with M slices:

$$\begin{aligned} a_i = \tfrac{1}{M} \textstyle\sum_{j=0}^{M-1} &((h(x_0 + r\sin\tfrac{2\pi j}{M}, y_0 + r\cos\tfrac{2\pi j}{M}) \\ &- h(x_0, y_0)) \cos(i\tfrac{2\pi j}{M})) \\ b_i = \tfrac{1}{M} \textstyle\sum_{j=0}^{M-1} &((h(x_0 + r\sin\tfrac{2\pi j}{M}, y_0 + r\cos\tfrac{2\pi j}{M}) \\ &- h(x_0, y_0)) \sin(i\tfrac{2\pi j}{M})) \end{aligned} \tag{5}$$

3.2 Calculating Azimuth-based Fourier Series by Convolution Filters

When the terrain surface is modelled by a GRID model, the elevations of the surface are known in the nodes of a rectangular grid with τ distance. The *h(x,y)* function may use bilinear interpolation, and calculates the elevation as the linear combination of the neighbour grid points of the position determined by x and y coordinates. The interpolated elevation is a linear combination of the elevations of the four neighbour grid nodes.

This interpolation method can be used in (5), and the Fourier coefficients will be the linear combinations of the values (elevations) of the grid points. The parameters of these linear combinations may be ordered to matrices. These matrices can be applied as a convolution filter in a GIS software to calculate the Fourier coefficients in each grid points of a digital elevation model.

In the first step, the size of the convolution matrices must be determined. This is a function of the radius (r) which is given in the unit of the grid resolution (τ). The size of the matrices is $(2n + 1) \times (2n + 1)$, where $n = \lceil r \rceil$ (or $n = \lceil r/\tau \rceil$, if r is in a distance unit). The rows and columns of this matrix indexed $0 \ldots n \ldots 2n$ numbers, the *(n,n)* element is the centre of the matrix.

The elements of the convolution matrices are calculated by a numerical integral of the azimuth (δ) angle, see in the Figure 1. A point of the arc is calculated in each iteration:

$$\begin{aligned} x &= n + r \cdot \cos\delta \\ y &= n + r \cdot \sin\delta \end{aligned} \tag{6}$$

The elevation of the *(x;y)* point can be calculated from the neighbour grid points, denoted A, B, C and D. The grid coordinates (or the matrix indexes, y is the row, x is the column) of these points:

$$\begin{aligned} A_x &= \lfloor x \rfloor \\ A_y &= \lfloor y \rfloor + 1 \\ B_x &= \lfloor x \rfloor + 1 \\ B_y &= \lfloor y \rfloor + 1 \\ C_x &= \lfloor x \rfloor \\ C_y &= \lfloor y \rfloor \\ D_x &= \lfloor x \rfloor + 1 \\ D_y &= \lfloor y \rfloor \end{aligned} \tag{7}$$

Using the method of [28] the elevation is calculated from these grid point with weights, which are proportional to the opposite rectangular areas. In this case, we do not calculate the elevation, instead the products of the weights and the Fourier components are added to the elements of the matrices. In this addition may be used the Simpson's rule [41], if the odd iterations have double weight.

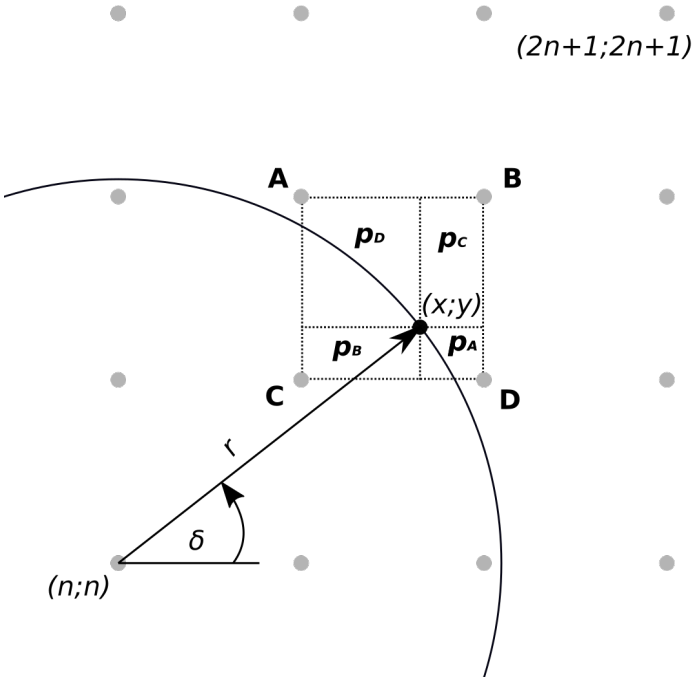

Figure 1
The calculations of the convolution filters in the coordinate system of the grid. This figure shows only the first quadrant (where the coordinates are between *n* and *2n*), but the method works similarly in the other quadrants.

The Python [44] module represented in Figure 2 calculates the convolution filters by the specified method.

The calccffilt function returns the calculated filters in a dictionary (hash), where the keys are the identifiers of the coefficients (for example 'A1') and the values are NumPy [49] arrays with the convolution filters. The input parameters of this program are the R (in unit of the grid, denoted τ), and optionally the default values of the azimuth resolution (*3600*) and the levels of the Fourier series (the default is *3*, which means *0*, *1* and *2*) can be modified.

```
import numpy as np

def calccffilt(r, m=3, azres=3600):
    """Create convolution filters for azimuth-based Fourier-series
       with r radius (in the unit of the grid), m level
       (i = 0, 1, ... m-1 Fourier coefficients) and azres
       resolution of the azimuth in the numerical integral.
       The return is a dictionary whose keys are strings with the
       identifier of the coefficient: 'A0', 'A1', 'B1', 'A2', 'B2', etc.
       The values of the dictionary are NumPy arrays with the
       two-dimensional discrete convolution filters."""
    n=int(np.ceil(r))
    filtarra=np.zeros((m, 2*n+1, 2*n+1))
    filtarrb=np.zeros((m, 2*n+1, 2*n+1))
    for azi in range(azres*2):
        azimut=np.pi*(azi/azres)
        simp=1+azi%2
        x=n+r*np.sin(azimut)
        y=n+r*np.cos(azimut)
        ptimp=( ( n,        n,        -1              ),
                ( int(x),   int(y),   (1-x%1)*(1-y%1) ),
                ( int(x),   int(y)+1, (1-x%1)*(y%1)   ),
                ( int(x)+1, int(y),   (x%1)*(1-y%1)   ),
                ( int(x)+1, int(y)+1, (x%1)*(y%1)     ) )
        for i in range(m):
            for pti in ptimp:
                filtarra[i, pti[0], pti[1]]+=pti[2]*np.cos(azimut*i)*simp
                filtarrb[i, pti[0], pti[1]]+=pti[2]*np.sin(azimut*i)*simp
    ret={}
    for i in range(m):
        ret[f'A{i}']=filtarra[i]/(azres*3)
        if i>0:
            ret[f'B{i}']=filtarrb[i]/(azres*3)
    return ret
```

Figure 2
The source code of the calccffilt function

Figure 3 shows a visualized example for the convolution filters. The provided Python function can calculate the filter with any other parameters. The size of the arrays of the returned filters depends on the radius.

Karasaridis [21] and Simoncelli [39] uses similar filters for image processing. Vörös [48] uses Fourier analysis to study the shape of a selected landform element (cones).

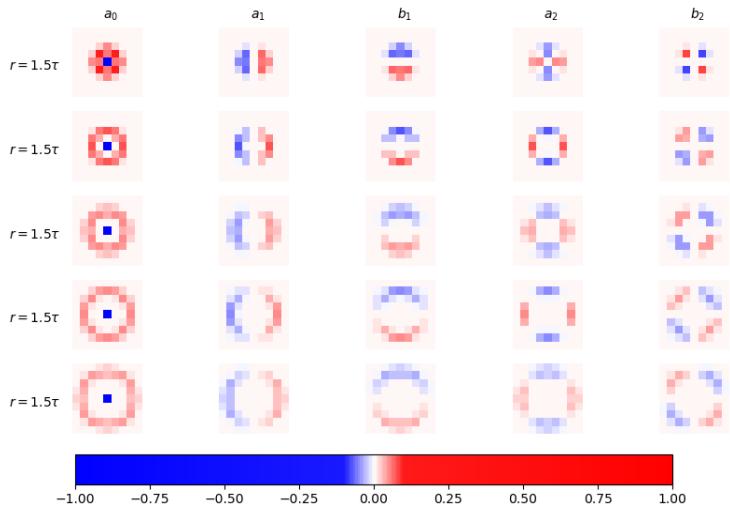

Figure 3
The convolution filters of a_0, a_1, b_1, a_2 and b_2 Fourier coefficients, with different radiuses using bilinear interpolation. (Most of the values are close to zero.)

3.3 Study the Coefficients

The azimuth-based Fourier coefficients can be calculated in each point of a DEM grid by two-dimensional discrete convolution filters, which are produced by the program described in Section 4. (Figure 2)

The following parameters can be calculated from the Fourier coefficients (a_0, a_1, b_1, a_2 and b_2):

$$
\begin{aligned}
P_0 &= 1 - \frac{a_0^2+a_1^2+b_1^2+a_2^2+b_2^2}{e^2+a_0^2+a_1^2+b_1^2+a_2^2+b_2^2} \\
P_1 &= \frac{a_1^2+b_1^2-a_0^2-a_2^2-b_2^2}{e^2+a_0^2+a_1^2+b_1^2+a_2^2+b_2^2} \\
P_2 &= \frac{2 \cdot a_0 \sqrt{a_2^2+b_2^2}}{e^2+a_0^2+a_1^2+b_1^2+a_2^2+b_2^2} \\
P_3 &= \frac{a_0^2-a_1^2-b_1^2-a_2^2-b_2^2}{e^2+a_0^2+a_1^2+b_1^2+a_2^2+b_2^2} \\
P_4 &= \frac{a_2^2+b_2^2-a_0^2-a_1^2-b_1^2}{e^2+a_0^2+a_1^2+b_1^2+a_2^2+b_2^2}
\end{aligned}
\tag{8}
$$

The e is a new parameter, it means the roughness of the terrain surface. The e has a length dimension (meter or other elevation unit) similar to the a_i and b_i Fourier coefficients. The P_i parameters have no dimension. The values of P_0 are between 0 and 1, and the values of other P_i parameters are between -1 and 1.

One hand if the P_0 value is larger (near *1*), the landform type proves plain. On the other hand the value of P_1 is large in the hillside (slope) areas. The value of P_2 is near -1 in ridges and near *1* in valleys. The P_3 value is larger in near a pit or peak; if $a_0 < 0$ it is a peak and if $a_0 > 0$ it is a pit. The P_4 is larger near the saddles.

3.4 Fuzzy Landform Classification

The fuzzy provides transitions between *true* and *false* (fuzzy logic) or an element's membership (fuzzy set). [52, 10, 46] The fuzzy classification is based on the fuzzy sets, uses fuzzy membership functions or fuzzy decision trees [43].

Fuzzy methods have been already used for landform classification in [38] with fuzzyfication of usual DEM-derived parameters. Irvin [18] uses ISODATA method for fuzzy classification of landform elements, Burrough [4] and Gorsevski [13] published a similar analysis with k-means method. MacMillan [25] also describes a fuzzy based landform classification.

The presented fuzzy based method differs from these previous methods. Azimuth based Fourier coefficients were used for creating a fuzzy landform classification.

The following fuzzy parameters can be calculated from the attributes which have been produced by the (8):

$$\mu_{flat} = P_0^{w_{flat}}$$

$$\mu_{slope} = \begin{cases} 0 & if\ P_1 < 0 \\ P_1^{w_{slope}} & if\ P_1 \geqq 0 \end{cases}$$

$$\mu_{ridge} = \begin{cases} 0 & if\ P_2 > 0 \\ (-P_2)^{w_{ridge}} & if\ P_2 \leqq 0 \end{cases}$$

$$\mu_{valley} = \begin{cases} 0 & if\ P_2 < 0 \\ P_2^{w_{valley}} & if\ P_2 \geqq 0 \end{cases} \tag{9}$$

$$\mu_{peak} = \begin{cases} 0 & if\ P_3 < 0\ or\ a_0 \geqq 0 \\ P_3^{w_{peak}} & if\ P_3 \geqq 0\ and\ a_0 < 0 \end{cases}$$

$$\mu_{pit} = \begin{cases} 0 & if\ P_3 < 0\ or\ a_0 \leqq 0 \\ P_3^{w_{pit}} & if\ P_3 \geqq 0\ and\ a_0 > 0 \end{cases}$$

$$\mu_{saddle} = \begin{cases} 0 & if\ P_4 < 0 \\ P_4^{w_{saddle}} & if\ P_4 \geqq 0 \end{cases}$$

These equations present the memberships function of the sets of different landform types. This classification defines fuzzy sets. The w_{flat}, w_{slope}, w_{ridge}, w_{valley}, w_{saddle}, w_{peak} and w_{pit} parameters can be made the related landform types stronger or weaker.

4 The Suggested Method in Practice

4.1 Implementation

Python programs were created for testing the presented method in the practice. The calccffilt function, which calculates the convolution filter has been presented in the Section 4. (Figure 2)

The calculations of (8) and (9) can be executed by simple expressions because the operators between two NumPy arrays with same size provide a third array with the results per each item. The NumPy also has a lot of functions for these calculations.

The two-dimensional discrete convolution filters need one other module. The signal module of the SciPy [3, 47] was applied to make this calculation.

The GDAL [50, 36] module provides tools for reading grid data from varied raster formats into a NumPy array. The GDAL also can write a NumPy array into a raster file.

Matplotlib [17] was used for visualizing the result and create figures (Figure 5). Besides QGIS [34] also used for these tasks.

Of course, any advanced desktop GIS software (e.g. GRASS [29, 30], SAGA [6, 31, 32], ArcGIS, etc.) include this data processing. The convolution filter and the map algebra or raster algebra are basic tools in these applications.

The process can be automated by QGIS Graphical Modeler, ArcGIS Model Builder or other graphical tools.

4.2 Results

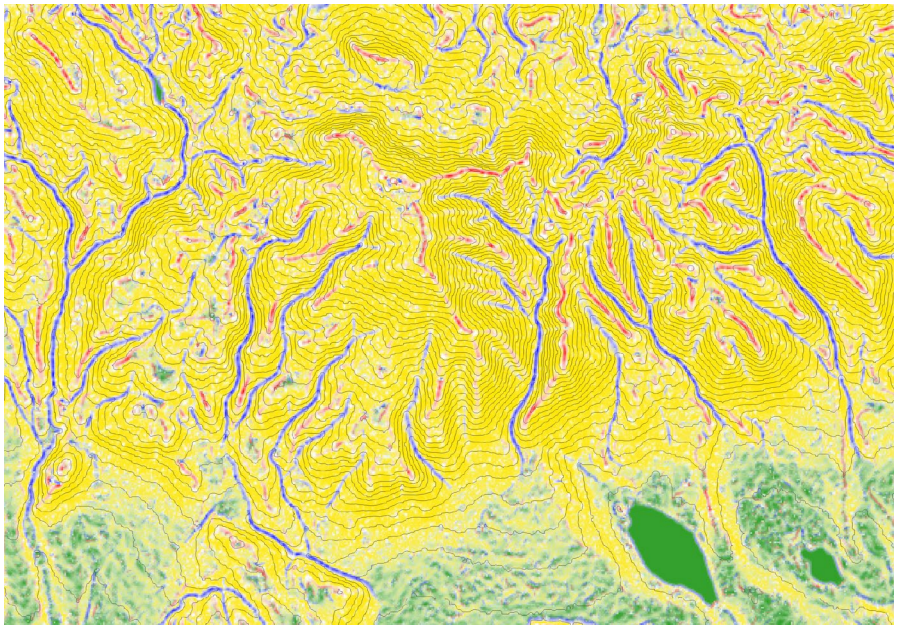

Figure 4
The fuzzy classifications of the landform in the sample area of Mátra mountains in Hungary. The contour interval is 25 meters. The landform is represented by blended colours. The blue and red areas (shaped lines) are the valleys and the ridges. The yellow areas are the slopes, and the green areas are the planes. (Created by QGIS)

After the development of the required programs (Section 7), the results derived by different parameters in different areas were studied. The elevation model was derived from the one arc second resolution SRTM [20, 35, 45] model by resampling in Standard Hungarian Projection [42] (HD72 or EOV, SRID=23700) in 25 meters resolution.

Figures 4 and 5 show the result in a sample area from Hungary. The $r=2.1$, $\tau=27.5$ *m* and $e=0.9$ *m*.

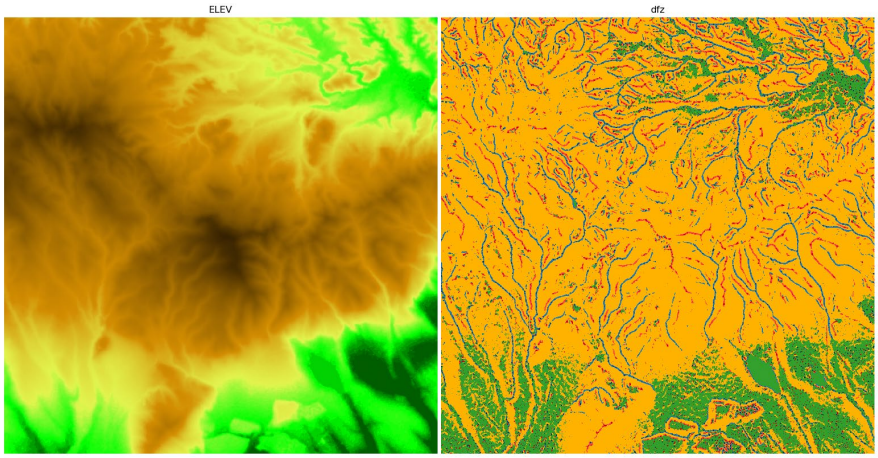

Figure 5
The defuzzified classifications of the landform in a sample area of Mátra mountains in Hungary. The elevation map with traditional colours on the left, and on the right is the defuzzified result map with colours similar to Figure 4. The darker red and blue areas are the pits and the peaks. (Created by Matplotlib)

5 Possibilities of Use in the Practice and Further Development

Landform classification is an important tool in several GIS application, for example in soil surveying [16], land consolidation [22], land valuation [23], hydrological related geomorphological analysis [1, 7], and in analysis of urban areas [27].

The landform is related to the performance of different DEM interpolation methods [5]. The optimal interpolation method can be chosen by any parameters of the introduced calculations.

The a_i and b_i parameters may be the input data of a machine learning algorithm in any analysis which uses terrain data.

Conclusion

The azimuth-based Fourier series are usable for geomorphological analysis and landform classification. The Fourier coefficients can be the base of several parameters. Fuzzy sets can be created based on the classification.

The presented methods can be used in several GIS software, because the convolution filter and raster calculator (map calculator) functions can provide the building blocks of the processing background.

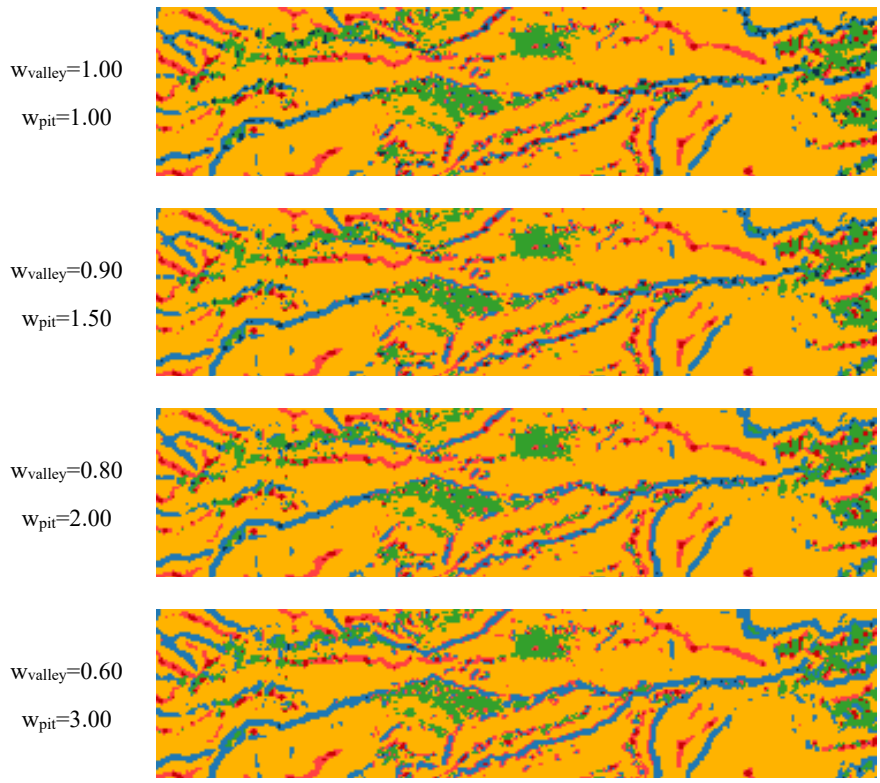

Figure 6
An enlarged part of the deffuzified map of the Figure 5 with different w_{valley} and w_{pit} parameters

The classification between valleys and pits is difficult. The *r* and *e* parameters influence the result, including this classification problem. The w_{valley} and w_{pit} parameters are very suitable for this goal. Figure 6 shows this problem and the result of the analysis with different parameters.

Acknowledgement

The paper was supported by the Bilateral Chinese-Hungarian Project No. 2019-2.1.11-TÉT-2020-00171 with the project title of *"Investigation of the characteristics of surface shapes in rural environment based on point clouds and remote sensing data"*.

References

[1] L. Bertalan, H. Sardemann, D. Mader, N. Mária Szopos, B. Nagy, and A. Eltner: Geomorphological and hydrological characterization of a meandering river by uav and uwv applications. In EGU General Assembly Conference Abstracts, p. 18069, 2020

[2] M. P. Bishop, L. A. James, J. F. Shroder, and S. J. Walsh. Geospatial technologies and digital geomorphological mapping: Concepts, issues and research. Geomorphology, 137(1):5-26, 2012. Geospatial Technologies and Geomorphological Mapping Proceedings of the 41st Annual Binghamton Geomorphology Symposium

[3] E. Bressert. Scipy and numpy: an overview for developers. 2012

[4] P. A. Burrough, P. F. van Gaans, and R. MacMillan: High-resolution landform classification using fuzzy k-means. Fuzzy sets and systems, 113(1):37-52, 2000

[5] V. Chaplot, F. Darboux, H. Bourennane, S. Leguédois, N. Silvera, and K. Phachomphon: Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density. Geomorphology, 77(1-2):126-141, 2006

[6] O. Conrad, B. Bechtel, M. Bock, H. Dietrich, E. Fischer, L. Gerlitz, J. Wehberg, V. Wichmann, and J. Böhner: System for automated geoscientific analyses (saga) v. 2.1. 4. Geoscientific Model Development, 8(7):1991-2007, 2015

[7] Z. Csatáriné Szabó, T. Mikita, G. Négyesi, O. G. Varga, P. Burai, L. Takács-Szilágyi, and S. Szabó: Uncertainty and overfitting in fluvial landform classification using laser scanned data and machine learning: A comparison of pixel and object-based approaches. Remote Sensing, 12(21):3652, 2020

[8] L. Drăguţ and C. Eisank: Automated object-based classification of topography from srtm data. Geomorphology, 141:21-33, 2012

[9] I. S. Evans. Geomorphometry and landform mapping: What is a landform? Geomorphology, 137(1):94-106, 2012

[10] J. C. Fodor and M. Roubens: Fuzzy preference modelling and multicriteria decision support. Springer Science & Business Media, 2013

[11] J. Fourier: Analytical theory of heat, by m. fourier, 1822

[12] J. Fourier: Theory analytique de la chaleur. Envres de Fourier, 1822

[13] P. V. Gorsevski, P. E. Gessler, and P. Jankowski: Integrating a fuzzy k-means classification and a bayesian approach for spatial prediction of landslide hazard. Journal of geographical systems, 5(3):223-251, 2003

[14] E. Guilbert, B. Moulin, and A. Cortés Murcia: A conceptual model for the representation of landforms using ontology design patterns. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3:15-22, 2016

[15] C. Güler, B. Beyhan, and H. Tağa. Polymorph-2d: An open-source gis plug-in for morphometric analysis of vector-based 2d polygon features. Geomorphology, 386:107755, 2021

[16] T. Hengl and D. G. Rossiter: Supervised landform classification to enhance and replace photo-interpretation in semi-detailed soil survey. Soil Science Society of America Journal, 67(6):1810-1822, 2003

[17] J. D. Hunter. Matplotlib: A 2d graphics environment. Computing In Science & Engineering, 9(3):90-95, May-Jun 2007

[18] B. J. Irvin, S. J. Ventura, and B. K. Slater: Fuzzy and isodata classification of landform elements from digital terrain data in pleasant valley, wisconsin. Geoderma, 77(2-4):137-154, 1997

[19] S. Jacek: Landform characterization with geographic information systems. Photogramm. Eng. Remote Sens, 63:183-191, 1997

[20] A. Jarvis, H. I. Reuter, A. Nelson, E. Guevara, et al: Hole-filled SRTM for the globe Version 4. available from the CGIAR-CSI SRTM 90m Database (http://srtm. csi. cgiar. org) 2008

[21] A. Karasaridis and E. Simoncelli: A filter design technique for steerable pyramid image transforms. In 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Vol. 4, pp. 2387-2390, IEEE, 1996

[22] J. Katona: The application of fuzzy logic in the field of land consolidation. In G. T. Orosz, editor, 10th International Symposium on Applied Informatics and Related Areas (AIS 2015) p. 4, Óbudai Egyetem, 2015

[23] J. Katona and M. Horoszné: Determination of factors modifying land value based on spatial data. In D. Drótos, J. Vásárhelyi, L. Czap, and P. Ivo, editors, Proceedings of the 19th International Carpathian Control Conference (ICCC 2018) pp. 625-628, IEEE, 2018

[24] R. MacMillan and P. Shary: Landforms and landform elements in geomorphometry. Developments in soil science, 33:227-254, 2009

[25] R. A. MacMillan, W. W. Pettapiece, S. C. Nolan, and T. W. Goddard: A generic procedure for automatically segmenting landforms into landform elements using dems, heuristic rules and fuzzy logic. Fuzzy sets and Systems, 113(1):81-109, 2000

[26] B. Márkus: Terrain analysis in consideration of surface curvature conditions. Periodica Polytechnica Civil Engineering, 30(1-2):71-81, 1986

[27] Q. Meng, X. Chen, Y. Sun, J. Zhang, Q. Wang, T. Jancsó, and S. Liu: Exposure opportunity index: measuring people-perceiving-greenery at floor-level effectively. Earth Science Informatics, 13:29-38, 2020

[28] G. Nagy: Interpolation methods for digital elevation models. In G. T. Orosz, editor, 9th International Symposium on Applied Informatics and Related Areas - AIS2014, pp. 72-74, Óbudai Egyetem, 2014

[29] M. Neteler, M. H. Bowman, M. Landa, and M. Metz. Grass gis: A multipurpose open source gis. Environmental Modelling & Software, 31:124-130, 2012

[30] M. Neteler and H. Mitasova: Open source GIS: a GRASS GIS approach, Vol. 689, Springer Science & Business Media, 2013

[31] V. Olaya and O. Conrad: Geomorphometry in saga. Developments in soil science, 33:293-308, 2009

[32] P. Passy and S. Théry: The use of saga gis modules in qgis. QGIS and generic tools, 1:107-149, 2018

[33] O. D. A. Prima, A. Echigo, R. Yokoyama, and T. Yoshida: Supervised landform classification of northeast honshu from dem-derived thematic maps. Geomorphology, 78(3-4):373-386, 2006

[34] QGIS Development Team: QGIS Geographic Information System. QGIS Association, 2023

[35] E. Rodriguez, C. S. Morris, and J. E. Belz: A global assessment of the SRTM performance. Photogrammetric Engineering & Remote Sensing, 72(3):249-260, 2006

[36] E. Rouault, F. Warmerdam, K. Schwehr, A. Kiselev, H. Butler, M. Loskot, T. Szekeres, E. Tourigny, M. Landa, I. Miara, et al: Gdal. Zenodo: Genève, Switzerland, 2022

[37] H. Saadat, R. Bonnell, F. Sharifi, G. Mehuys, M. Namdar, and S. Ale Ebrahim: Landform classification from a digital elevation model and satellite imagery. Geomorphology, 100(3-4):453-464, 2008

[38] J. Schmidt and A. Hewitt: Fuzzy land element classification from dtms based on geometry and terrain position. Geoderma, 121(3-4):243-256, 2004

[39] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In Proceedings., International Conference on Image Processing, Vol. 3, pp. 444-447, IEEE, 1995

[40] T. F. Stepinski and J. Jasiewicz: Geomorphons-a new approach to classification of landforms. Proceedings of geomorphometry, 2011:109-112, 2011

[41] S. Thomas: Miscallaneous tracts on some curious and very interefting subjets. J. Nourse, 1752

[42] G. Timár and G. Molnár: Map grids and datums. Eötvös Lóránd University, 84, 2013

[43] B. Tusor, M. Takács, A. R. Várkonyi-Kóczy, and J. T. Tóth: A fast fuzzy decision tree for color filtering. In 2015 IEEE 9th International Symposium on Intelligent Signal Processing (WISP) Proceedings, pp. 1-6, IEEE, 2015

[44] G. Van Rossum et al: Python Programming Language. In USENIX Annual Technical Conference, Vol. 41, 2007

[45] J. J. Van Zyl: The Shuttle Radar Topography Mission (SRTM): a breakthrough in remote sensing of topography. Acta Astronautica, 48(5):559-565, 2001

[46] A. R. Várkonyi-Kóczy: Fast anytime fuzzy fourier estimation of multisine signals. IEEE Transactions on Instrumentation and Measurement, 58(5):1763-1770, 2009

[47] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods, 17(3):261-272, 2020

[48] F. Vörös, B. van Wyk de Vries, M.-N. Guilbaud, T. Görüm, D. Karátson, and B. Székely: Dtm-based comparative geomorphometric analysis of four scoria cone areas – suggestions for additional approaches. Remote Sensing, 14(23):6152, 2022

[49] S. v. d. Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2):22-30, 2011

[50] F. Warmerdam: The geospatial data abstraction library, open source approaches in spatial data handling. pp. 87-104, Springer, 2008

[51] R. Yokoyama, M. Shirasawa, and R. J. Pike: Visualizing topography by openness: a new application of image processing to digital elevation models. Photogrammetric engineering and remote sensing, 68(3):257-266, 2002

[52] L. A. Zadeh. Fuzzy sets. Information and control, 8(3):338-353, 1965

[53] A. Weiss. Topographic Position and Landforms Analysis. Poster presentation, ESRI User Conference, San Diego, CA. 2001